# CHAPTER 2

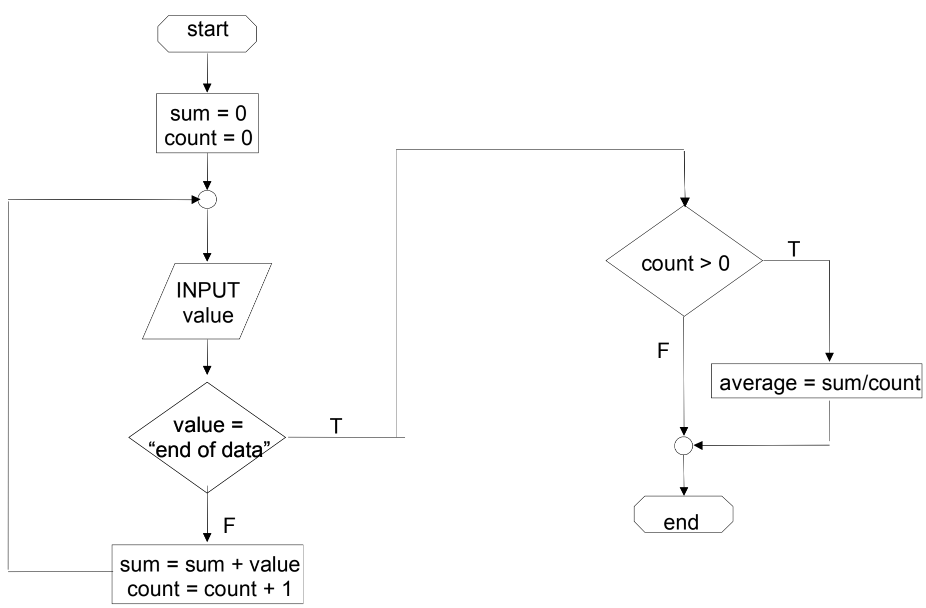
2.1

```
IF x < 10 THEN
  IF x < 5 THEN
    x = 5
  ELSE
    PRINT x
  END IF
ELSE
  DO
    IF x < 50 EXIT
    x = x - 5
  END DO
END IF
```

2.2

Step 1: Start
Step 2: Initialize ***sum*** and ***count*** to zero
Step 3: Examine top card.
Step 4: If it says "end of data" proceed to step 9; otherwise, proceed to next step.
Step 5: Add value from top card to ***sum***.
Step 6: Increase ***count*** by 1.
Step 7: Discard top card
Step 8: Return to Step 3.
Step 9: Is the ***count*** greater than zero?
If yes, proceed to step 10.
If no, proceed to step 11.
Step 10: Calculate average = ***sum***/***count***
Step 11: End

2.3



2.4

Students could implement the subprogram in any number of languages. The following Fortran 90 program is one example. It should be noted that the availability of complex variables in Fortran 90, would allow this subroutine to be made even more concise. However, we did not exploit this feature, in order to make the code more compatible with Visual BASIC, MATLAB, etc.

```
PROGRAM Rootfind
IMPLICIT NONE
INTEGER::ier
REAL::a, b, c, r1, i1, r2, i2
DATA a,b,c/1.,5.,2./
CALL Roots(a, b, c, ier, r1, i1, r2, i2)
IF (ier .EQ. 0) THEN
  PRINT *, r1,i1," i"
  PRINT *, r2,i2," i"
ELSE
  PRINT *, "No roots"
END IF
END

SUBROUTINE Roots(a, b, c, ier, r1, i1, r2, i2)
IMPLICIT NONE
INTEGER::ier
REAL::a, b, c, d, r1, i1, r2, i2
r1=0.
r2=0.
i1=0.
i2=0.
IF (a .EQ. 0.) THEN
  IF (b <> 0) THEN
    r1 = -c/b
  ELSE
    ier = 1
  END IF
ELSE
  d = b**2 - 4.*a*c
  IF (d >= 0) THEN
    r1 = (-b + SQRT(d))/(2*a)
    r2 = (-b - SQRT(d))/(2*a)
  ELSE
    r1 = -b/(2*a)
    r2 = r1
    i1 = SQRT(ABS(d))/(2*a)
    i2 = -i1
  END IF
END IF
END
```

The answers for the 3 test cases are: (*a*) −0.438, -4.56; (*b*) 0.5; (*c*) −1.25 + 2.33*i*; −1.25 − 2.33*i*.

Several features of this subroutine bear mention:

- The subroutine does not involve input or output. Rather, information is passed in and out via the arguments. This is often the preferred style, because the I/O is left to the discretion of the programmer within the calling program.
- Note that an error code is passed (IER = 1) for the case where no roots are possible.

2.5 The development of the algorithm hinges on recognizing that the series approximation of the sine can be represented concisely by the summation,

$$\sum_{i=1}^{n} \frac{x^{2i-1}}{(2i-1)!}$$

where $i$ = the order of the approximation. The following algorithm implements this summation:

Step 1: Start
Step 2: Input value to be evaluated x and maximum order n
Step 3: Set order (i) equal to one
Step 4: Set accumulator for approximation (approx) to zero
Step 5: Set accumulator for factorial product (fact) equal to one
Step 6: Calculate true value of sin(x)
Step 7: If order is greater than n then proceed to step 13
Otherwise, proceed to next step
Step 8: Calculate the approximation with the formula

$$\text{approx} = \text{approx} + (-1)^{\text{i-1}} \frac{\text{x}^{\text{2i-1}}}{\text{factor}}$$

Step 9: Determine the error

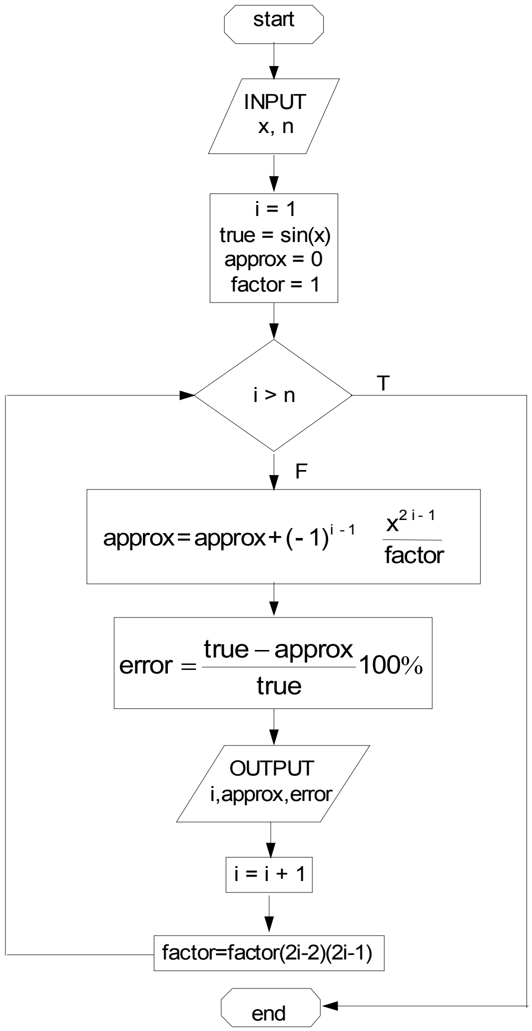
$$\%\text{error} = \left| \frac{\text{true} - \text{approx}}{\text{true}} \right| 100\%$$

Step 10: Increment the order by one
Step 11: Determine the factorial for the next iteration

$$\text{factor} = \text{factor} \bullet (2 \bullet \text{i} - 2) \bullet (2 \bullet \text{i} - 1)$$

Step 12: Return to step 7
Step 13: End

2.6



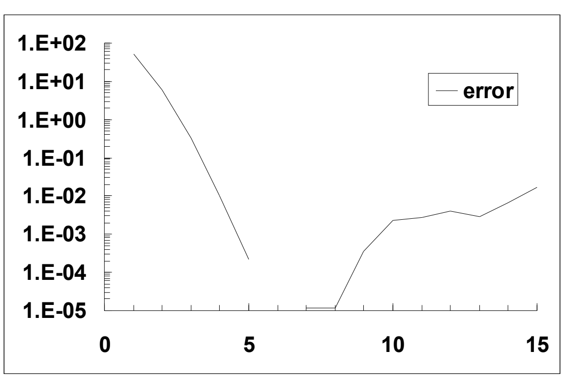

Pseudocode:

```
SUBROUTINE Sincomp(n,x)
i = 1
true = SIN(x)
approx = 0
factor = 1
DO
  IF i > n EXIT
  approx = approx + (-1)^(i-1)•x^(2•i-1) / factor
  error = Abs(true - approx) / true) * 100
  PRINT i, true, approx, error
  i = i + 1
  factor = factor•(2•i-2)•(2•i-1)
END DO
END
```

2.7 The following Fortran 90 code was developed based on the pseudocode from Prob. 2.6:

```
PROGRAM Series
IMPLICIT NONE
INTEGER::n
REAL::x
n = 15
x = 1.5
CALL Sincomp(n,x)
END

SUBROUTINE Sincomp(n,x)
IMPLICIT NONE
INTEGER::n,i,fac
REAL::x,tru,approx,er
i = 1
tru = SIN(x)
approx = 0.
fac = 1
PRINT *, "      order     true         approx          error"
DO
  IF (i > n) EXIT
  approx = approx + (-1) ** (i-1) * x ** (2*i - 1) / fac
  er = ABS(tru - approx) / tru) * 100
  PRINT *, i, tru, approx, er
  i = i + 1
  fac = fac * (2*i-2) * (2*i-1)
END DO
END

OUTPUT:
       order     true         approx          error
           1  0.9974950       1.500000      -50.37669
           2  0.9974950      0.9375000       6.014566
           3  0.9974950       1.000781     -0.3294555
           4  0.9974950      0.9973912      1.0403229E-02
           5  0.9974950      0.9974971     -2.1511559E-04
           6  0.9974950      0.9974950      0.0000000E+00
           7  0.9974950      0.9974951     -1.1950866E-05
           8  0.9974950      0.9974949      1.1950866E-05
           9  0.9974950      0.9974915      3.5255053E-04
          10  0.9974950      0.9974713      2.3782223E-03
          11  0.9974950      0.9974671      2.7965026E-03
          12  0.9974950      0.9974541      4.0991469E-03
          13  0.9974950      0.9974663      2.8801586E-03
          14  0.9974950      0.9974280      6.7163869E-03
          15  0.9974950      0.9973251      1.7035959E-02
Press any key to continue
```

The errors can be plotted versus the number of terms:



Interpretation: The absolute percent relative error drops until at n = 6, it actually yields a perfect result (pure luck!). Beyond, n = 8, the errors starts to grow. This occurs because of round-off error, which will be discussed in Chap. 3.

2.8 AQ = 442/5 = 88.4
AH = 548/6 = 91.33

without final

$$AG = \frac{30(88.4) + 30(91.33)}{30 + 30} = 89.8667$$

with final

$$AG = \frac{30(88.4) + 30(91.33) + 40(91)}{30 + 30} = 90.32$$

The following pseudocode provides an algorithm to program this problem. Notice that the input of the quizzes and homeworks is done with logical loops that terminate when the user enters a negative grade:

```
INPUT number, name
INPUT WQ, WH, WF
nq = 0
sumq = 0
DO
  INPUT quiz (enter negative to signal end of quizzes)
  IF quiz < 0 EXIT
  nq = nq + 1
  sumq = sumq + quiz
END DO
AQ = sumq / nq
PRINT AQ
nh = 0
sumh = 0
PRINT "homeworks"
DO
  INPUT homework (enter negative to signal end of homeworks)
  IF homework < 0 EXIT
  nh = nh + 1
  sumh = sumh + homework
END DO
AH = sumh / nh
PRINT "Is there a final grade (y or n)"
INPUT answer
IF answer = "y" THEN
  INPUT FE
  AG = (WQ * AQ + WH * AH + WF * FE) / (WQ + WH + WF)
ELSE
  AG = (WQ * AQ + WH * AH) / (WQ + WH)
END IF
PRINT number, name$, AG
END
```

2.9

| n | F |
|---|---|
| 0 | $100,000.00 |
| 1 | $108,000.00 |
| 2 | $116,640.00 |
| 3 | $125,971.20 |
| 4 | $136,048.90 |
| 5 | $146,932.81 |
| | • |
| | • |
| | • |
| 24 | $634,118.07 |
| 25 | $684,847.52 |

2.10 Programs vary, but results are

Bismarck = −10.842 $t = 0$ to 59
Yuma = 33.040 $t = 180$ to 242

2.11

| n | A |
|---|---|
| 1 | 40,250.00 |
| 2 | 21,529.07 |
| 3 | 15,329.19 |
| 4 | 12,259.29 |
| 5 | 10,441.04 |

2.12

| Step | $v(12)$ | $\varepsilon_t$ (%) |
|---|---|---|
| 2 | 49.96 | -5.2 |
| 1 | 48.70 | -2.6 |
| 0.5 | 48.09 | -1.3 |

Error is halved when step is halved

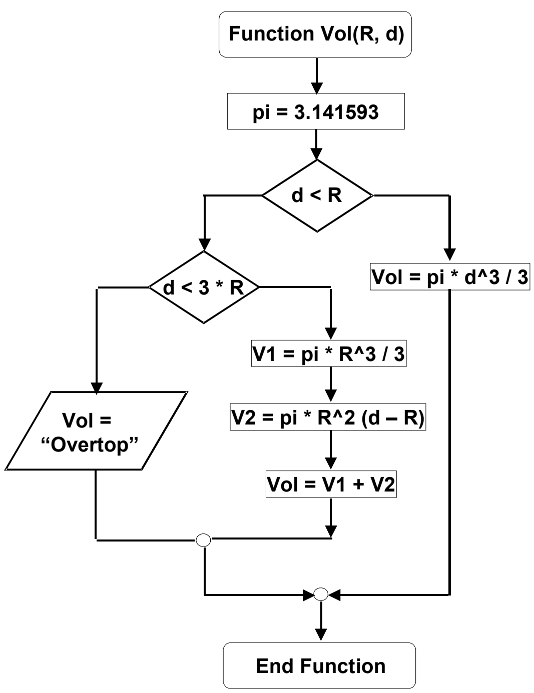
2.13

| Fortran 90 | VBA |
|---|---|

```fortran
Subroutine BubbleFor(n, b)

Implicit None

!sorts an array in ascending
!order using the bubble sort

Integer(4)::m, i, n
Logical::switch
Real::a(n),b(n),dum

m = n - 1
Do
  switch = .False.
  Do i = 1, m
    If (b(i) > b(i + 1)) Then
      dum = b(i)
      b(i) = b(i + 1)
      b(i + 1) = dum
      switch = .True.
    End If
  End Do
  If (switch == .False.) Exit
  m = m - 1
End Do

End
```

```vba
Option Explicit

Sub Bubble(n, b)

'sorts an array in ascending
'order using the bubble sort

Dim m As Integer, i As Integer
Dim switch As Boolean
Dim dum As Single

m = n - 1
Do
  switch = False
  For i = 1 To m
    If b(i) > b(i + 1) Then
      dum = b(i)
      b(i) = b(i + 1)
      b(i + 1) = dum
      switch = True
    End If
  Next i
  If switch = False Then Exit Do
  m = m - 1
Loop

End Sub
```

2.14 Here is a flowchart for the algorithm:




Here is a program in VBA:

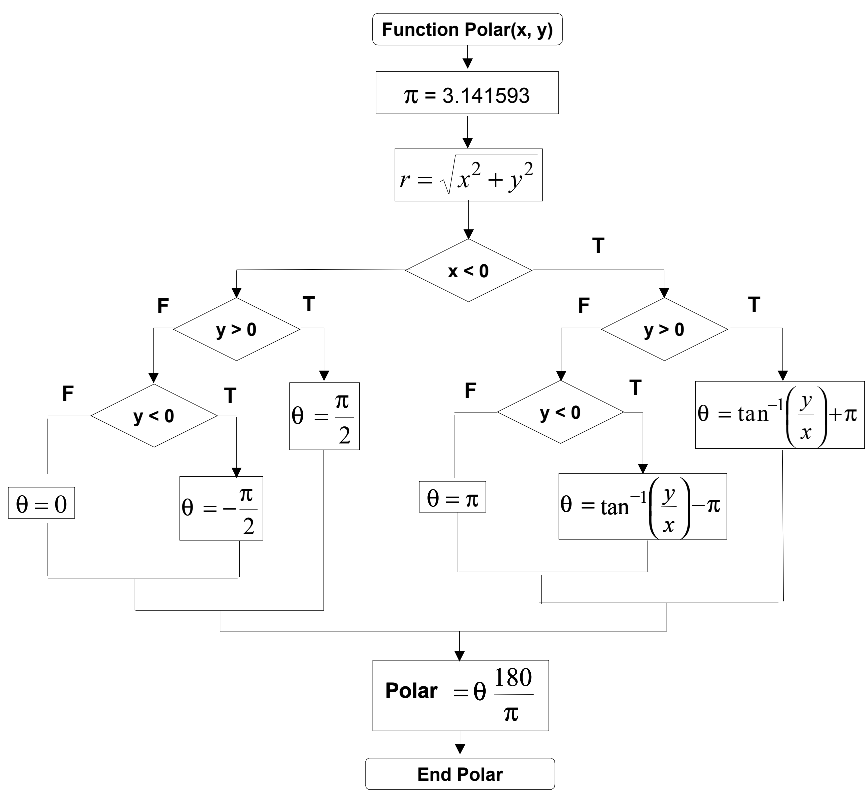
```vba
Option Explicit

Function Vol(R, d)
```

```
Dim V1 As Single, v2 As Single, pi As Single

pi = 4 * Atn(1)

If d < R Then
  Vol = pi * d ^ 3 / 3
ElseIf d <= 3 * R Then
  V1 = pi * R ^ 3 / 3
  v2 = pi * R ^ 2 * (d - R)
  Vol = V1 + v2
Else
  Vol = "overtop"
End If

End Function
```

The results are

| R | d | Volume |
|---|---|---|
| 1 | 0.3 | 0.028274 |
| 1 | 0.8 | 0.536165 |
| 1 | 1 | 1.047198 |
| 1 | 2.2 | 4.817109 |
| 1 | 3 | 7.330383 |
| 1 | 3.1 | overtop |

2.15 Here is a flowchart for the algorithm:




And here is a VBA function procedure to implement it:

```
Option Explicit

Function Polar(x, y)
```

```
Dim th As Single, r As Single
Const pi As Single = 3.141593

r = Sqr(x ^ 2 + y ^ 2)

If x < 0 Then
  If y > 0 Then
    th = Atn(y / x) + pi
  ElseIf y < 0 Then
    th = Atn(y / x) - pi
  Else
    th = pi
  End If
Else
  If y > 0 Then
    th = pi / 2
  ElseIf y < 0 Then
    th = -pi / 2
  Else
    th = 0
  End If
End If

Polar = th * 180 / pi

End Function
```

The results are:

| x | y | θ |
|---|---|---|
| 1 | 1 | 90 |
| 1 | -1 | -90 |
| 1 | 0 | 0 |
| -1 | 1 | 135 |
| -1 | -1 | -135 |
| -1 | 0 | 180 |
| 0 | 1 | 90 |
| 0 | -1 | -90 |
| 0 | 0 | 0 |

4.1 a) For this case $x_i = 0$ and $h = x$, thus

$$f(x) = f(0) + f'(0)x + f''(0)\frac{x^2}{2} + \cdots$$

$$f(0) = f'(0) = f''(0) = 1$$

$$\therefore\ f(x) = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \cdots$$

b) $$f(x_{i+1}) = e^{-x_i} - e^{-x_i}h + e^{-x_i}\frac{h^2}{2} - e^{-x_i}\frac{h^3}{6} + \cdots$$

for $x_i = 0.25$ and $x_{i+1} = 1$,

$h = 0.75$

zero order:

$$f(1) \approx e^{-0.25} = 0.778801$$

$$\text{true value} = e^{-1.0} = 0.367879$$

$$\epsilon_t = \frac{0.367879 - 0.778801}{0.367879} \times 100$$

$$\epsilon_t = -111.7\ \%$$

first order:

$$f(1) \approx 0.778801 - 0.778801(.75)$$
$$\approx 0.1947$$

$$\epsilon_t = 47.1\%$$

second order:

$$f(1) = 0.778801 - 0.778801(.75) + 0.77801\frac{(.75)^2}{2}$$

$$f(1) \approx 0.413738$$

$$\epsilon_t = -12.46\ \%$$

third order:

$$f(1) = 0.413738 - 0.77801\frac{(.75)^3}{6}$$

$$f(1) \approx 0.358978$$

$$\epsilon_t = 2.42\ \%$$

4.2 use $\epsilon_s = 0.5 \times 10^{2-2} = 0.5\%$

zero order:

$$\cos(\pi/4) \approx 1$$

$$\text{true value } \cos(\pi/4) = 0.707107$$

$$\epsilon_t = \frac{0.707107 - 1}{0.707107} \times 100 = -41.42\%$$

first order:

$$\cos(\pi/4) = 1 - \frac{(\pi/4)^2}{2}$$
$$= 0.691575$$
$$(\epsilon_t = 2.19\ \%)$$

$$\epsilon_a = \frac{0.691575 - 1}{0.691575} \times 100 = -44.6\ \%$$

second order:

$$\cos(\pi/4) \approx 0.691575 + \frac{(\pi/4)^4}{24}$$

$$\approx 0.707429$$

$$(\epsilon_t = -0.456\%)$$

$$\epsilon_a = 2.24\%$$

third order:

$$\cos(\pi/4) \approx 0.707429 - \frac{\left(\frac{\pi}{4}\right)^6}{720}$$

$$\approx 0.707130$$

$$(\epsilon_t = 0.0005\%)$$

$$\epsilon_a = -0.046\%$$

4.3 $\epsilon_s = 0.5 \times 10^{2-2} = 0.5\%$

zero order:

$$\sin\left(\frac{\pi}{4}\right) = \frac{\pi}{4} = 0.785398$$

True value $= 0.707107$

$$\epsilon_t = \frac{0.707107 - 0.785398}{0.707107} \times 100$$

$$= -11.1\%$$

first order:

$$\sin\left(\frac{\pi}{4}\right) \approx 0.785398 - \frac{\left(\frac{\pi}{4}\right)^3}{6}$$

$$= 0.704653$$

$$(\epsilon_t = 0.347\%)$$

$$\epsilon_a = \frac{(0.704653 - 0.785398)}{0.704653} \times 100$$

$$\approx -11.46\%$$

second order:

$$\sin(\pi/4) \approx 0.704653 + \frac{\left(\frac{\pi}{4}\right)^5}{60}$$

$$\approx 0.709633$$

$$(\epsilon_t = -0.57\%)$$

$$\epsilon_a = .70\%$$

third order:

$$\sin\left(\frac{\pi}{4}\right) \approx 0.709633 - \frac{\left(\frac{\pi}{4}\right)^7}{7!}$$

$$\approx 0.709597$$

$$(\epsilon_t \approx -0.35\%)$$

$$\epsilon_a \approx -0.0051\%$$

4.4 true value $f(2) = 102$

zero order:

$$f(2) \approx f(1) \approx -62$$

$$(\epsilon_t = 160.8\%)$$

first order:

$f'(1) = 75(1)^2 - 12(1) + 7 = 70$

$f(2) \approx -62 + 70(1) = 8$

$\epsilon_t = 92.1\%$

second order:

$f''(1) = 150(1) - 12 = 138$

$f(2) \approx 8 + \frac{138}{2}(1)^2 = 77$

$\epsilon_t = 24.5\%$

third order:

$f'''(1) = 150$

$f(2) \approx 77 + \frac{150}{6}(1)^3$

$\approx 102$

$\epsilon_t = 0$

as expected

4.5

true $= \ln(3) = 1.098612$

zero order:

$f(3) = f(1) = 0$

$\epsilon_t = 100\%$

first order:

$f(3) = 0 + \frac{1}{1}(2) = 2$

$\epsilon_t = -82.05\%$

second order:

$f(3) = 2 - \frac{1}{1^2}\frac{(2)^2}{2}$

$= 0$

$\epsilon_t = 100\%$

third order

$f(3) = 0 + \frac{2}{1^3}\frac{2^3}{6}$

$= 2.66666$

$(\epsilon_t = -142.7\%)$

fourth order

$f(4) = 2.6666 - \frac{6}{(1)^4}\frac{(2)^4}{24}$

$\approx -1.3333$

$\epsilon_t = 221.36\%$

diverges must use smaller step for series to converge

4.6 $f'(x) = 75x^2 - 12x + 7$

$f'(2) = 283$ true

$x_{i-1} = 1.75 \quad f(x_{i-1}) = 39.85938$
$x_i = 2.0 \quad f(x_i) = 102$
$x_{i+1} = 2.25 \quad f(x_{i+1}) = 182.1406$

forward

$$f'(2) = \frac{182.1406 - 102}{.25}$$

$$= 320.5625$$

$$\epsilon_t = -13.273$$

backward

$$f'(2) = \frac{102 - 39.85938}{.25}$$

$$= 248.5625$$
$$\epsilon_t = 12.17\%$$

central

$$f'(2) \approx \frac{182.1406 - 39.85938}{2(.25)}$$

$$= 284.5625$$
$$\epsilon_t = -0.55\%$$

Both forward and backward have errors approximately

$$|E_t| \approx \frac{f''(x_i)}{2} h$$

$$f''(2) = 150(2) - 12 = 288$$

$$\therefore |E_t| \approx \frac{288}{2}(.25) = 36$$

which is close

For central Difference

$$|E_t| \approx -\frac{f'''(x_i)}{6} h^2$$

$$\approx -\frac{150}{6}(.25)^2 = -1.5625$$

which is exactly

$$E_t = 283 - 284.5625$$

as expected

4.7 true value

$$f''(2) = 288$$

$$f''(2) \approx \frac{164.56 - 2(102) + 50.92}{(.2)^2}$$

$$= 288$$

$$f''(2) = \frac{131.765 - 2(102) + 75.115}{(.1)^2}$$

$$= 288$$

both are exact because errors are function of 4th order derivatives which are zero for 3rd order polynomial

$\Delta v = 2.77332 + 0.435734$
$= 3.209053$

$\therefore v = 30.4533 \pm 3.209053$

4.8 $$\frac{\partial v}{\partial c} = \frac{cgt\,e^{-c/m\,t} - gm(1-e^{-c/m\,t})}{c^2}$$

$= -1.38666$

$\Delta v(\tilde{c}) = \left|\frac{\partial v}{\partial c}\right| \Delta\tilde{c}$

$= 1.38666\,(2)$
$= 2.77332$

$$v(12.5) = \frac{9.8(50)}{12.5}\left(1 - e^{-\frac{12.5(6)}{50}}\right)$$

$= 30.4533$

$\therefore v = 30.4533 \pm 2.77332$

4.9 $$\Delta v(\tilde{c},\tilde{m}) = \underbrace{\left|\frac{\partial c}{\partial v}\right| \Delta\tilde{c}}_{2.77332} + \left|\frac{\partial v}{\partial m}\right| \Delta\tilde{m}$$

$$\frac{\partial v}{\partial m} = \frac{gt}{m}e^{-c/m\,t} + \frac{g}{c}(1 - e^{-c/m\,t})$$

$= 0.871467$

$\left|\frac{\partial v}{\partial m}\right| \Delta m = 0.871467\,(0.5)$

$= 0.435734$

4.10 $\Delta H(\tilde{T}) = \left|\frac{\partial H}{\partial T}\right| \Delta\tilde{T}$

$\frac{\partial H}{\partial T} = 4Ae\sigma T^3$

$= 4(.15)(.9)(5.67\times10^{-8})(650)^3$

$= 8.41$

$\Delta H \approx 8.41(25) = 210.2$

Exact Error

$$\Delta H_{true} = \frac{H(675) - H(625)}{2}$$

$$= \frac{1589 - 1167}{2}$$

$= 211$ close to $210.2$

For $\Delta T = 50$

$\Delta H \approx 8.41(50) = 420.4$

$$\Delta H_{true} = \frac{H(700) - H(600)}{2}$$

$$= \frac{1837 - 992}{2}$$

$= 422.5$ close to $420.4$

Results are good because $H(T)$ is nearly linear over range of $\Delta T$

4.11 For sphere $A = 4\pi r^2$

$$H = 4\pi r^2 e \sigma T^4$$

$$\Delta H = \left|\frac{\partial H}{\partial r}\right| \Delta \tilde{r} + \left|\frac{\partial H}{\partial e}\right| \Delta e + \left|\frac{\partial H}{\partial T}\right| \Delta T$$

$$\frac{\partial H}{\partial r} = 8\pi r e \sigma T^4$$

$$= 17604$$

$$\frac{\partial H}{\partial e} = 4\pi r^2 \sigma T^4$$

$$= 1467$$

$$\frac{\partial H}{\partial T} = 16\pi r^2 e \sigma T^3$$

$$= 9.6$$

$$\Delta H = 17604(.02) + 1467(.05) + 9.6(25)$$

$$\Delta H \approx 665.4$$

$$H(.17, .95, 575) = 2138.4$$
$$H(.13, .85, 525) = 777.6$$

$$\Delta H_{TRUE} = \frac{2138.4 - 777.6}{2}$$

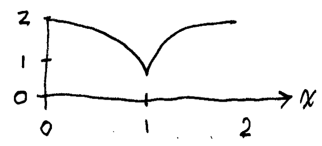
$$= 680.4$$

4.12 $$CN = \frac{\tilde{x} f'(\tilde{x})}{f(\tilde{x})}$$

a) $$CN = \frac{1.0001\left[\frac{1}{2\sqrt{1.0001-1}}\right]}{\sqrt{1.0001-1} + 1}$$

$$= 50.00$$

ill conditioned because $f'(1)$ is large near $x = 1$



b) $$CN = \frac{9(-e^{-9})}{e^{-9}} = -9$$

ill conditions because $x$ is large

c) $$CN = \frac{(200)\frac{200}{\sqrt{200^2+1}} - 1}{\sqrt{200^2+1} - 200}$$

$$\approx \frac{200(-1.2\times10^{-5})}{0.0025}$$

$$\approx -0.96$$ well conditioned

d) $$CN = \frac{x\left(\frac{-xe^{-x} - e^{-x} + 1}{x^2}\right)}{\left(\frac{e^{-x}-1}{x}\right)}$$

$$= \frac{.01\left(\frac{-.01(.99) - (.99) + 1}{(.01)^2}\right)}{\left(\frac{.99-1}{.01}\right)}$$

$$= \frac{.9}{-.1} = -.9$$ well conditioned

e) $f(x) = \dfrac{\sin(x)}{1+\cos(x)}$

$$f'(x) = \frac{(1+\cos x)(\cos(x)) + \sin x(\sin x)}{(1+\cos x)^2}$$

$$CN = \frac{1.001\pi \; f'(1.001\pi)}{f(1.001\pi)}$$

$$CN = \frac{3.144\,(202642)}{-636.6}$$

$$= -1001$$

ill conditioned because

$1 + \cos(1.001\pi) \approx 0$

4.13 addition and subtraction

$f(u,v) = u+v$

$$\Delta F = \left|\frac{\partial f}{\partial u}\right| \Delta u + \left|\frac{\partial f}{\partial v}\right| \Delta \hat{v}$$

$$\left|\frac{\partial f}{\partial u}\right| = 1 \qquad \left|\frac{\partial f}{\partial v}\right| = 1$$

$$\Delta f(\tilde{u},\tilde{v}) = \Delta\tilde{u} + \Delta\hat{v}$$

multiplication

$f(u,v) = u \cdot v$

$$\left|\frac{\partial f}{\partial u}\right| = v \qquad \left|\frac{\partial f}{\partial v}\right| = u$$

$$\Delta f(\bar{u},\bar{v}) = |\bar{v}|\,\Delta\hat{u} + |\bar{u}|\,\Delta\tilde{v}$$

division

$f(u,v) = u/v$

$$\frac{\partial f}{\partial u} = \frac{1}{v}$$

$$\frac{\partial f}{\partial v} = -u/v^2$$

$$\Delta f(u,v) = \left|\frac{1}{v}\right| \Delta u + \left|\frac{u}{v^2}\right| \Delta v$$

$$= \frac{|v|\,\Delta u + |u|\,\Delta v}{|v^2|}$$

4.14 $f(x) = ax^2+bx+c$

$f'(x) = 2ax+b$

$f''(x) = 2a$

$$a x_{i+1}^2 + b x_{i+1} + c =$$

$$a x_i^2 + b x_i + c + 2a x_i + b\,(x_{i+1} - x_i)$$

$$+ \frac{2a}{2}\left(x_{i+1}^2 - 2x_{i+1}x_i + x_i^2\right)$$

collect terms

$$a\left(x_i^2 + 2x_i x_{i+1} - 2x_i^2 + x_{i+1}^2 - 2x_{i+1}x_i + x_i^2\right) =$$

$$a\,x_{i+1}^2$$

etc

4.15

$$\Delta Q = \left|\frac{\partial Q}{\partial n}\right|\Delta n + \left|\frac{\partial Q}{\partial s}\right|\Delta s$$

$$\frac{\partial Q}{\partial n} = -\frac{1}{n^2}\frac{(BH)^{5/3}}{(B+2H)^{2/3}} s^{\frac{1}{2}}$$

$$\frac{\partial Q}{\partial s} = \frac{1}{n}\frac{(BH)^{5/3}}{(B+2H)^{2/3}}\frac{1}{2s^{1/2}}$$

$$\Delta Q = \underbrace{|-5.07|(.003)}_{0.15} + \underbrace{|2536|(.00003)}_{0.076}$$

∴ error from roughness measurement is about 2 times the error caused by uncertainty in slope, thus improve precision of roughness is best strategy

4.16 use $\epsilon_s = 0.5 \times 10^{2-2} = 0.5\%$

zero order

$$\frac{1}{1-.1} = 1.1111 \approx 1$$

$$\epsilon_t = \left(\frac{1.1111 - 1}{1.1111}\right) \times 100 = 9.99\%$$

first order

$$1.1111 \approx 1 + 0.1 = 1.1$$

$$\epsilon_t = .99\%$$

$$\epsilon_a = \frac{1.1 - 1}{1.1} \times 100 = 9.1$$

second order

$$1.1111 \approx 1 + .1 + .01 = 1.11$$

$$\epsilon_t = 0.099\%$$

$$\epsilon_a = \frac{1.11 - 1.1}{1.11} \times 100$$
$$= 0.9\%$$

third order

$$1.11111 \approx 1 + .1 + .01 + .001$$
$$\approx 1.111$$

$$\epsilon_t = 0.009\%$$

$$\epsilon_a = \frac{1.111 - 1.11}{1.111} \times 100$$

$$= 0.09\% < 0.5\%$$

4.17 $\Delta(\sin\phi_0) =$

$$\left|\frac{\partial}{\partial \alpha}\left[(1+\alpha)\sqrt{1 - \frac{\alpha}{1+\alpha}\left(\frac{v_e}{v_0}\right)^2}\right]\right| \Delta\alpha$$

$$= \left|\left\{\frac{1+\alpha}{2}\left(1 - \frac{\beta\alpha}{1+\alpha}\right)^{-\frac{1}{2}}\left(\frac{\beta\alpha}{(1+\alpha)^2} - \frac{\beta}{1+\alpha}\right) + \left(1 - \frac{\beta\alpha}{1+\alpha}\right)^{1/2}\right\}\right| \Delta\alpha$$

where $\beta = \left(\frac{v_e}{v_0}\right)^2 = 4$ and $\alpha = 0.2$

$$\Delta(\sin\phi_0) = 2.3\,\Delta\tilde{\alpha}$$

for $\Delta\delta = 0.2\,(.01) = 0.002$

$\Delta(\sin\phi_o) = 0.0046$

$\sin\phi_o = (1+.2)\sqrt{1-\frac{.2}{1.2}(4)}$

$= .6928$

Therefore

$\max \sin\phi_o = .6928 + .0046$

$= 0.69742$

$\min \sin\phi_o = .6928 - .0046$

$= 0.68822$

$\max \phi_o = \frac{0.771792}{2\pi} \times 360$

$= 44.22^\circ$

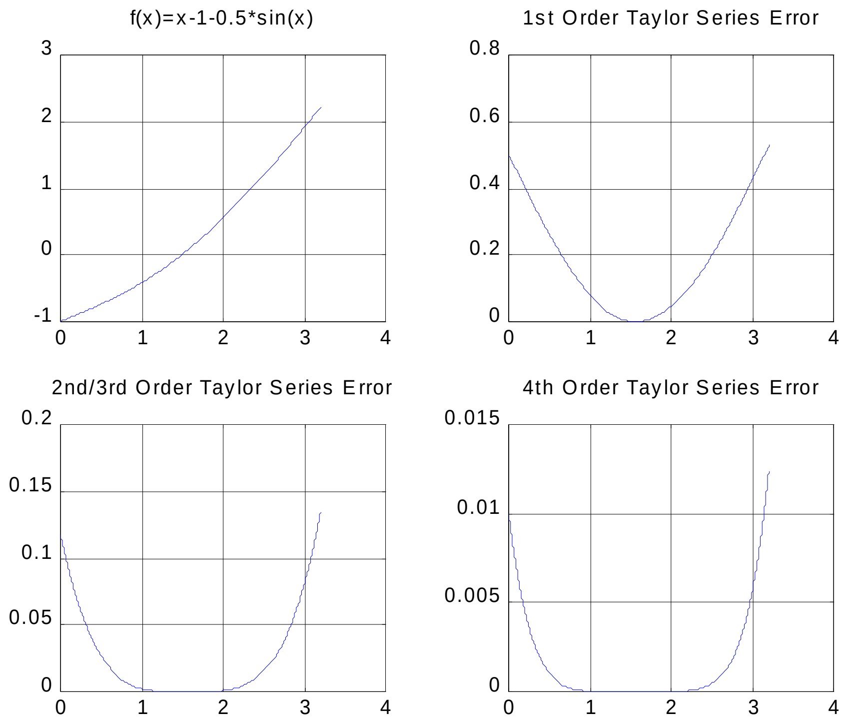
$\min \phi_o = 43.49^\circ$

4.18 f(x) = x-1-1/2*sin(x)
f '(x) = 1-1/2*cos(x)
f "(x) = 1/2*sin(x)
f "'(x) = 1/2*cos(x)
$f^{IV}$(x) = -1/2*sin(x)

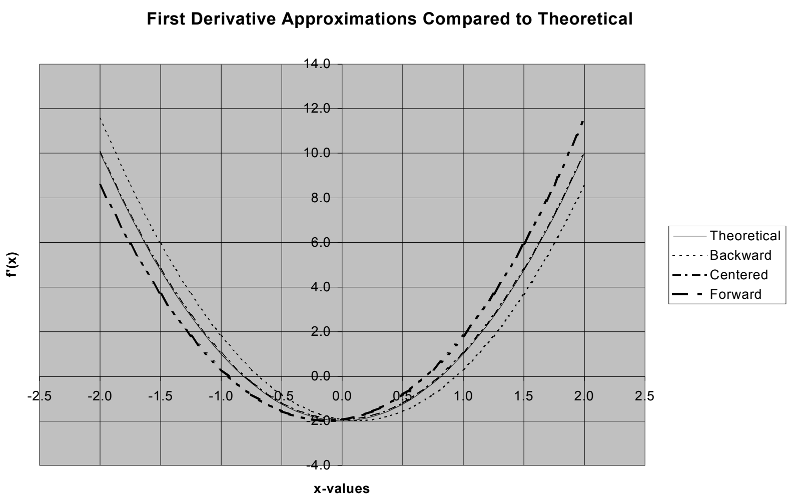
Using the Taylor Series Expansion (Equation 4.5 in the book), we obtain the following 1st, 2nd, 3rd, and 4th Order Taylor Series functions shown below in the Matlab program- f1, f2, f4. Note the 2nd and 3rd Order Taylor Series functions are the same.

From the plots below, we see that the answer is the 4th Order Taylor Series expansion.

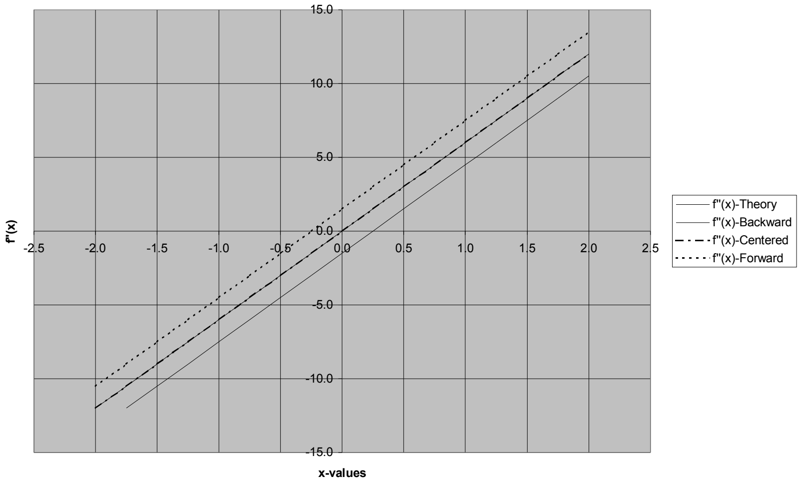
```
x=0:0.001:3.2;
f=x-1-0.5*sin(x);
subplot(2,2,1);
plot(x,f);grid;title('f(x)=x-1-0.5*sin(x)');hold on

f1=x-1.5;
e1=abs(f-f1);      %Calculates the absolute value of the
difference/error
subplot(2,2,2);
plot(x,e1);grid;title('1st Order Taylor Series Error');

f2=x-1.5+0.25.*((x-0.5*pi).^2);
e2=abs(f-f2);
subplot(2,2,3);
plot(x,e2);grid;title('2nd/3rd Order Taylor Series Error');

f4=x-1.5+0.25.*((x-0.5*pi).^2)-(1/48)*((x-0.5*pi).^4);
e4=abs(f4-f);
subplot(2,2,4);
plot(x,e4);grid;title('4th Order Taylor Series Error');hold off
```




## 4.19 EXCEL WORKSHEET AND PLOTS



Approximations of the 2nd Derivative




| x | f(x) | f(x-1) | f(x+1) | f(x-2) | f(x+2) | f''(x)-Theory | f''(x)-Back | f''(x)-Cent | f''(x)-Forw |
|---|---|---|---|---|---|---|---|---|---|
| -2.000 | 0.000 | -2.891 | 2.141 | 3.625 | 3.625 | -12.000 | 150.500 | -12.000 | -10.500 |
| -1.750 | 2.141 | 0.000 | 3.625 | -2.891 | 4.547 | -10.500 | -12.000 | -10.500 | -9.000 |
| -1.500 | 3.625 | 2.141 | 4.547 | 0.000 | 5.000 | -9.000 | -10.500 | -9.000 | -7.500 |
| -1.250 | 4.547 | 3.625 | 5.000 | 2.141 | 5.078 | -7.500 | -9.000 | -7.500 | -6.000 |
| -1.000 | 5.000 | 4.547 | 5.078 | 3.625 | 4.875 | -6.000 | -7.500 | -6.000 | -4.500 |
| -0.750 | 5.078 | 5.000 | 4.875 | 4.547 | 4.484 | -4.500 | -6.000 | -4.500 | -3.000 |
| -0.500 | 4.875 | 5.078 | 4.484 | 5.000 | 4.000 | -3.000 | -4.500 | -3.000 | -1.500 |
| -0.250 | 4.484 | 4.875 | 4.000 | 5.078 | 3.516 | -1.500 | -3.000 | -1.500 | 0.000 |
| 0.000 | 4.000 | 4.484 | 3.516 | 4.875 | 3.125 | 0.000 | -1.500 | 0.000 | 1.500 |
| 0.250 | 3.516 | 4.000 | 3.125 | 4.484 | 2.922 | 1.500 | 0.000 | 1.500 | 3.000 |
| 0.500 | 3.125 | 3.516 | 2.922 | 4.000 | 3.000 | 3.000 | 1.500 | 3.000 | 4.500 |
| 0.750 | 2.922 | 3.125 | 3.000 | 3.516 | 3.453 | 4.500 | 3.000 | 4.500 | 6.000 |
| 1.000 | 3.000 | 2.922 | 3.453 | 3.125 | 4.375 | 6.000 | 4.500 | 6.000 | 7.500 |
| 1.250 | 3.453 | 3.000 | 4.375 | 2.922 | 5.859 | 7.500 | 6.000 | 7.500 | 9.000 |
| 1.500 | 4.375 | 3.453 | 5.859 | 3.000 | 8.000 | 9.000 | 7.500 | 9.000 | 10.500 |
| 1.750 | 5.859 | 4.375 | 8.000 | 3.453 | 10.891 | 10.500 | 9.000 | 10.500 | 12.000 |
| 2.000 | 8.000 | 5.859 | 10.891 | 4.375 | 14.625 | 12.000 | 10.500 | 12.000 | 13.500 |

| x | f(x) | f(x-1) | f(x+1) | f'(x)-Theory | f'(x)-Back | f'(x)-Cent | f'(x)-Forw |
|---|---|---|---|---|---|---|---|
| -2.000 | 0.000 | -2.891 | 2.141 | 10.000 | 11.563 | 10.063 | 8.563 |
| -1.750 | 2.141 | 0.000 | 3.625 | 7.188 | 8.563 | 7.250 | 5.938 |
| -1.500 | 3.625 | 2.141 | 4.547 | 4.750 | 5.938 | 4.813 | 3.688 |
| -1.250 | 4.547 | 3.625 | 5.000 | 2.688 | 3.688 | 2.750 | 1.813 |
| -1.000 | 5.000 | 4.547 | 5.078 | 1.000 | 1.813 | 1.063 | 0.313 |
| -0.750 | 5.078 | 5.000 | 4.875 | -0.313 | 0.313 | -0.250 | -0.813 |
| -0.500 | 4.875 | 5.078 | 4.484 | -1.250 | -0.813 | -1.188 | -1.563 |
| -0.250 | 4.484 | 4.875 | 4.000 | -1.813 | -1.563 | -1.750 | -1.938 |
| 0.000 | 4.000 | 4.484 | 3.516 | -2.000 | -1.938 | -1.938 | -1.938 |
| 0.250 | 3.516 | 4.000 | 3.125 | -1.813 | -1.938 | -1.750 | -1.563 |
| 0.500 | 3.125 | 3.516 | 2.922 | -1.250 | -1.563 | -1.188 | -0.813 |
| 0.750 | 2.922 | 3.125 | 3.000 | -0.313 | -0.813 | -0.250 | 0.313 |
| 1.000 | 3.000 | 2.922 | 3.453 | 1.000 | 0.313 | 1.063 | 1.813 |
| 1.250 | 3.453 | 3.000 | 4.375 | 2.688 | 1.813 | 2.750 | 3.688 |
| 1.500 | 4.375 | 3.453 | 5.859 | 4.750 | 3.688 | 4.813 | 5.938 |
| 1.750 | 5.859 | 4.375 | 8.000 | 7.188 | 5.938 | 7.250 | 8.563 |
| 2.000 | 8.000 | 5.859 | 10.891 | 10.000 | 8.563 | 10.063 | 11.563 |

## Chapter 8

**8.1**

$$V = \frac{RT}{p} = \frac{0.082054(375)}{2.0}$$

$$V = 15.38513$$

For van der Waal

$$\left(2.0 + \frac{12.02}{v^2}\right)(v - 0.08407) = 0.082054(375)$$

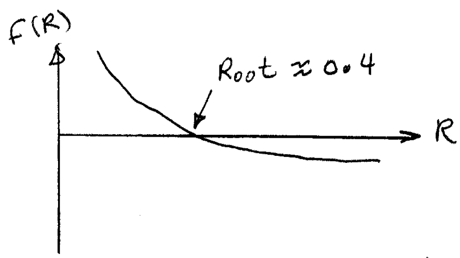
Bisection on Toolkit gives

$x_l = 10 \quad x_u = 20 \quad \epsilon_s = 0.001\%$

$x_r = 15.84641$ in 16 iterations

**8.2**

$$f(R) = \frac{R+1}{R(1+0.1R)} - \ln\left(\frac{1+0.1R}{0.1R}\right)$$



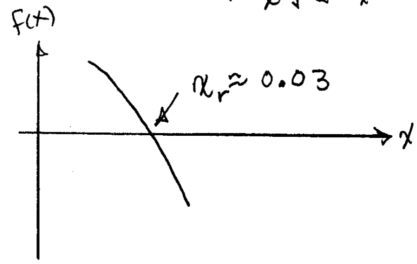

using Bisection from Toolkit

with $R_l = 0.01 \quad R_u = 2 \quad \epsilon_s = 0.001\%$

gives $R_{root} = 0.429944$ in 19 iterations

**8.3**

$$f(x) = 0.05 - \frac{x}{1-x}\sqrt{\frac{6}{2+x}}$$



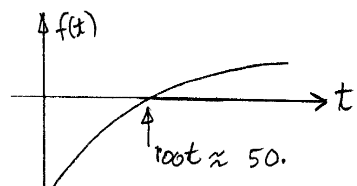

use Toolkit Bisection

$x_l = 0.01 \quad x_u = 0.05$

$\epsilon_s = 0.001\%$ gives

$x_r = 0.02824936$ in 18 iterations

**8.4** $f(t) = 10(1 - e^{-0.04t}) + 4e^{-0.04t} - 0.93(10)$




use Toolkit Bisection

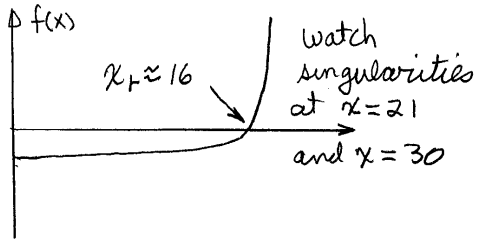
$t_l = 0 \quad t_u = 100 \quad \epsilon_s = 0.001\%$

gives $t_r = 53.71056$

in 18 iterations

8.5

$$f(x) = \frac{4+x}{(42-2x)^2\,(30-x)} - 0.015$$




Use Toolkit Bisection with

$x_l = 10 \quad x_u = 20$ with $\epsilon_s = 0.001$

gives $x_r = 16.09299$ in 16 iterations

8.6

$$K_1 = \frac{(c_0 + x_1 + x_2)}{(a_0 - 2x_1 - x_2)^2\,(b_0 - x_1)}$$

$$K_2 = \frac{(c_0 + x_1 + x_2)}{(a_0 - 2x_1 - x_2)(d_0 - x_2)}$$

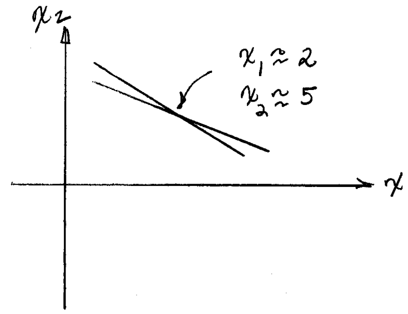
or

$$f_1(x_1, x_2) = \frac{5 + x_1 + x_2}{(50 - 2x_1 - x_2)^2\,(20 - x_1)} - 4\times10^{-4}$$

$$f_2(x_1, x_2) = \frac{5 + x_1 + x_2}{(50 - 2x_1 - x_2)(10 - x_1)} - 3.7\times10^{-2}$$

Graph functions by specifying $x_1$ and solving for $x_2$ using bisection

| 1st Equation | | 2nd Equation | |
|---|---|---|---|
| $x_1$ | $x_2$ | $x_1$ | $x_2$ |
| 0 | 8.667 | 0 | 9.854 |
| 1 | 6.862 | 1 | 7.490 |
| 2 | 5.065 | 2 | 5.105 |
| 3 | 3.277 | 3 | 2.697 |
| 4 | 1.498 | 4 | 0.265 |
| 5 | -0.27 | 5 | -2.194 |




Now apply 2 variable Newton Raphson with initial guess of $x_1 = 2$ and $x_2 = 5$

$f_1(2,5) = -3.4\times10^{-6}$

$f_2(2,5) = -4.15\times10^{-4}$

$\partial f_1/\partial x_1 = 9.4\times10^{-5} \qquad \dfrac{\partial f_2}{\partial x_1} = 9.4\times10^{-3}$

$\partial f_1/\partial x_2 = 5.2\times10^{-5} \qquad \partial f_2/\partial x_2 = 3.9\times10^{-3}$

$|J| = -1.23\times10^{-7}$

$$x_1 = 2 - \frac{-3.4\times10^{-6}(3.9\times10^{-3}) - (-4.15\times10^{-4})(5.2\times10^{-5})}{-1.23\times10^{-7}}$$

$x_1 = 2.0672$

$$x_2 = 5 + \frac{-3.4\times10^{-6}(9.4\times10^{-3}) - (-4.15\times10^{-4})\,9.4\times10^{-5}}{-1.23\times10^{-7}}$$

$x_2 = 4.9448$

Second iteration gives

$x_1 = 2.06623$

$x_2 = 4.94621$

8.7 with given values

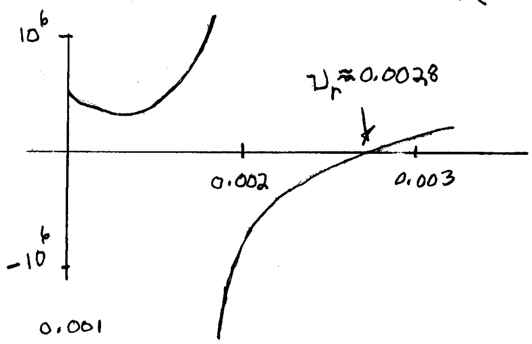
$a = 12.55778$
$b = 0.001863$

Use Excel Solver or Toolkit

gives

$h_r = 0.740015$

$$f(v) = 65000 - \frac{0.518(233)}{v - 0.001863} + \frac{12.55778}{v(v+0.001863)(15.264)}$$




use Toolkit with
$v_l = 0.002$

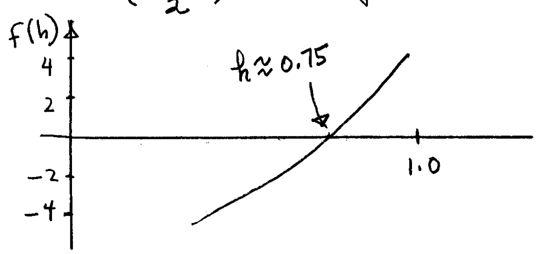
$v_u = 0.004$

gives $v_r = 0.002808$
in 17 iterations

$$\text{mass} = \frac{V}{v} = \frac{3}{0.002808} = 1068.22 \text{ kg}$$

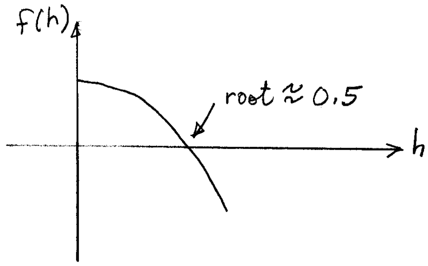
8.8

$$f(h) = 4\cos^{-1}\left(\frac{2-h}{2}\right) - (2-h)\sqrt{4h-h^2} - \frac{8}{5} = 0$$



8.9

$$f(h) = 0.5 - \frac{\pi h^2 (3-h)}{3}$$




use Toolkit with $x_l = 0$, $x_u = 1$

gives $x_r = 0.43118$

in 18 iterations with $\epsilon_s = 0.001\%$

8.10

$$g(h) = \sqrt{\frac{h^3 + 0.477}{3}}$$

$$g'(h) = \frac{\frac{1}{2} h^2}{\sqrt{\frac{h^3 + 0.477}{3}}}$$

Solve roots problem

$$f(h) = 1 - g'(h)$$

if $g'(h) > 1$ then $f(h) < 0$

using Toolkit with $x_l = 0$, $x_u = 5$

gives $x_r = 1.5159$

therefore for $h$ between 0 and 1.5159 fixed point iteration converges

If $g(h) = 3(h^2 - 0.159)^{1/3}$

$$g'(h) = \frac{2h}{(h^2 - 0.159)^{2/3}}$$

solve roots problem

$$f(h) = 1 - g'(h)$$

such that when $f(h) < 0$, $g'(h) > 1$

using Toolkit with $x_l = 1$, $x_u = 12$

gives $x_r = 8.04$

therefore for $h$ between $\sqrt{0.159} \approx 0.4$ and 8.04 fixed point iteration converges

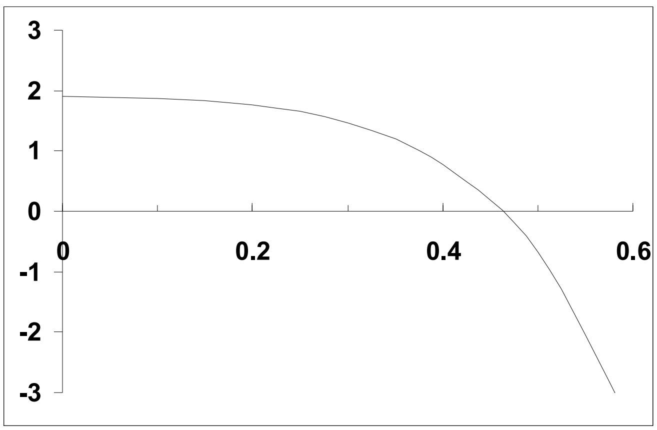
8.11 Substituting the parameter values yields

$$10\frac{\varepsilon^3}{1-\varepsilon}=150\frac{1-\varepsilon}{1000}+1.75$$

This can be rearranged and expressed as a roots problem

$$f(\varepsilon)=0.15(1-\varepsilon)+1.75-10\frac{\varepsilon^3}{1-\varepsilon}=0$$

A plot of the function suggests a root at about 0.45.




But suppose that we do not have a plot. How do we come up with a good initial guess. The void fraction (the fraction of the volume that is not solid; i.e. consists of voids) varies between 0 and 1. As can be seen, a value of 1 (which is physically unrealistic) causes a division by zero. Therefore, two physically-based initial guesses can be chosen as 0 and 0.99. Note that the zero is not physically realistic either, but since it does not cause any mathematical difficulties, it is OK. Applying bisection yields a result of $\varepsilon$ = 0.461857 in 15 iterations with an absolute approximate relative error of $6.5\times10^{-3}$ %

8.12

The total pressure is equal to the partial pressures of the components:

$$P=P_b+P_t$$

According to Antoine's equation

$$P_b=e^{A_b-\frac{B_b}{T+C_b}} \qquad P_t=e^{A_t-\frac{B_t}{T+C_t}}$$

Combining the equations yields

$$f(T)=e^{A_b-\frac{B_b}{T+C_b}}+e^{A_t-\frac{B_t}{T+C_t}}-P=0$$

The root of this equation can be evaluated to yield $T$ = 350.5.

8.13 There are a variety of ways to solve this system of 5 equations

$$K_1 = \frac{[H^+][HCO_3^-]}{[CO_2]} \tag{1}$$

$$K_2 = \frac{[H^+][CO_3^{2-}]}{[HCO_3^-]} \tag{2}$$

$$K_w = [H^+][OH^-] \tag{3}$$

$$c_T = [CO_2] + [HCO_3^-] + [CO_3^{2-}] \tag{4}$$

$$\text{Alk} = [HCO_3^-] + 2[CO_3^{2-}] + [OH^-] - [H^+] \tag{5}$$

One way is to combine the equations to produce a single polynomial. Equations 1 and 2 can be solved for

$$[H_2CO_3^*] = \frac{[H^+][HCO_3^-]}{K_1} \qquad [CO_3^{2-}] = \frac{[H^+]K_2}{[HCO_3^-]}$$

These results can be substituted into Eq. 4, which can be solved for

$$[H_2CO_3^*] = F_0 c_T \qquad [HCO_3^-] = F_1 c_T \qquad [CO_3^{2-}] = F_2 c_T$$

where $F_0$, $F_1$, and $F_2$ are the fractions of the total inorganic carbon in carbon dioxide, bicarbonate and carbonate, respectively, where

$$F_0 = \frac{[H^+]^2}{[H^+]^2 + K_1[H^+] + K_1K_2} \quad F_1 = \frac{K_1[H^+]}{[H^+]^2 + K_1[H^+] + K_1K_2} \quad F_2 = \frac{K_1K_2}{[H^+]^2 + K_1[H^+] + K_1K_2}$$

Now these equations, along with the Eq. 3 can be substituted into Eq. 5 to give

$$0 = F_1 c_T + 2F_2 c_T + K_w \big/ [H^+] - [H^+] - \text{Alk}$$

Although it might not be apparent, this result is a fourth-order polynomial in [H$^+$].

$$[H^+]^4 + (K_1 + \text{Alk})[H^+]^3 + (K_1K_2 + \text{Alk}K_1 - K_w - K_1c_T)[H^+]^2 \\ + (\text{Alk}K_1K_2 - K_1K_w - 2K_1K_2c_T)[H^+] - K_1K_2K_w = 0$$

Substituting parameter values gives

$$[H^+]^4 + 2.001\times10^{-3}[H^+]^3 - 5.012\times10^{-10}[H^+]^2 - 1.055\times10^{-19}[H^+] - 2.512\times10^{-31} = 0$$

This equation can be solved for [H$^+$] = 2.51x10$^{-7}$ (pH = 6.6). This value can then be used to compute

$$[OH^-] = \frac{10^{-14}}{2.51\times10^{-7}} = 3.98\times10^{-8}$$

$$[H_2CO_3^*] = \frac{\left(2.51\times10^{-7}\right)^2}{\left(2.51\times10^{-7}\right)^2 + 10^{-6.3}\left(2.51\times10^{-7}\right) + 10^{-6.3}10^{-10.3}} 3\times10^{-3} = 0.33304\left(3\times10^{-3}\right) = 0.001$$

$$[HCO_3^-] = \frac{10^{-6.3}\left(2.51\times10^{-7}\right)}{\left(2.51\times10^{-7}\right)^2 + 10^{-6.3}\left(2.51\times10^{-7}\right) + 10^{-6.3}10^{-10.3}} 3\times10^{-3} = 0.666562\left(3\times10^{-3}\right) = 0.002$$

$$[CO_3^{2-}] = \frac{10^{-6.3}10^{-10.3}}{\left(2.51\times10^{-7}\right)^2 + 10^{-6.3}\left(2.51\times10^{-7}\right) + 10^{-6.3}10^{-10.3}} 3\times10^{-3} = 0.000133\left(3\times10^{-3}\right) = 1.33\times10^{-4}M$$

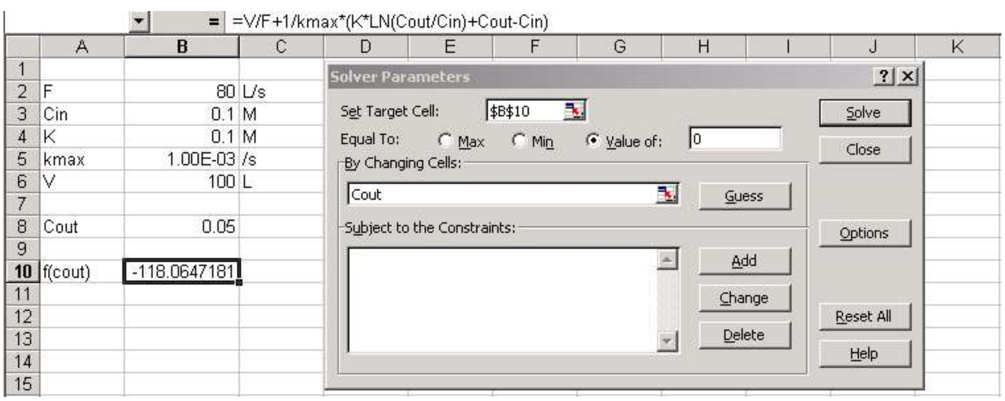
8.14 The integral can be evaluated as

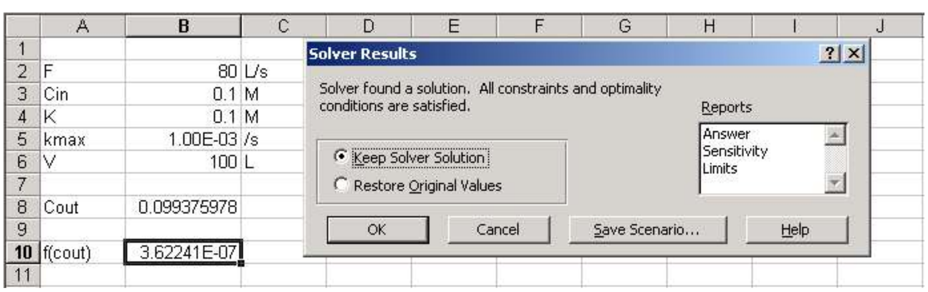
$$-\int_{C_{in}}^{C_{out}} \frac{K}{k_{max}C} + \frac{1}{k_{max}} dc = -\frac{1}{k_{max}}\left[K\ln\left(\frac{C_{out}}{C_{in}}\right) + C_{out} - C_{in}\right]$$

Therefore, the problem amounts to finding the root of

$$f(C_{out}) = \frac{V}{F} + \frac{1}{k_{max}}\left[K\ln\left(\frac{C_{out}}{C_{in}}\right) + C_{out} - C_{in}\right]$$

Excel solver can be used to find the root:

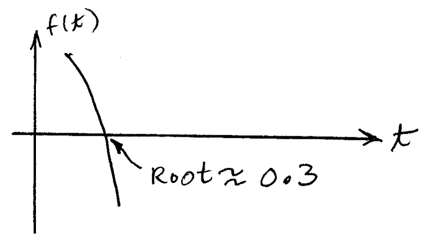


8.15

$$f(t)=8e^{-0.5t}\cos(3t)-4$$

a)




b) $f'(t)=8e^{-0.5t}(-3\sin 3t)$

$+\cos 3t\left(-4e^{-0.5t}\right)$

$$t_{i+1}=t_i-\frac{8e^{-0.5t_i}\cos 3t_i-4}{e^{-0.5t_i}(-24\sin 3t-4\cos 3t)}$$

| iter | $t_{i+1}$ | $\epsilon_a$ % |
|---|---|---|
| 1 | 0.3152936 | 4.8 |
| 2 | 0.3151661 | 0.04 |
| 3 | 0.3151661 | 0 |

c) use $t_{-1}=0.3$ $t_0=0.4$

| iter | $t_{i+1}$ | $\epsilon_a$ % |
|---|---|---|
| 1 | 0.3146946 | 4.7 |
| 2 | 0.3151525 | 27 |
| 3 | 0.3151661 | 0.15 |
| 4 | 0.3151661 | 0.004 |

8.16 let $x=P/A$

$$f(x)=\frac{40}{1+.25\sec\left[\frac{1}{2}\sqrt{\frac{x}{30{,}000}}\,100\right]}-x$$

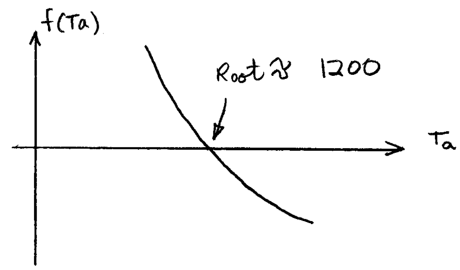
Use Toolkit Bisection with

$x_l=15$ $x_u=25$ $\epsilon_s=0.0001$

gives $x_R=20.46312$ after 19 iterations

8.17

$$f(T_a)=\frac{T_a}{20}\left(e^{500/T_a}+e^{-500/T_a}\right)-10-\frac{T_a}{10}$$




Use Toolkit Bisection with

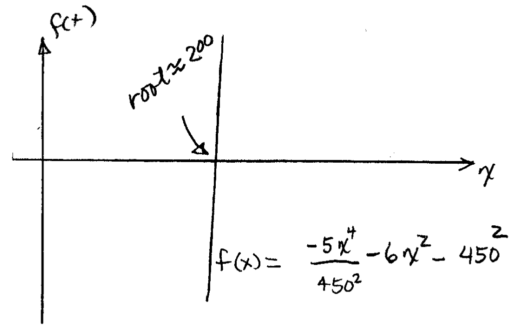
$T_{a_l}=1000$ $T_{a_u}=2000$ $\epsilon_s=0.0001$ %

gives $T_a=1266.324$

after 21 iterations

8.18

$$\frac{dy}{dx} = 0 = -5x^4 + 6(450)^2 x^2 - 450^4$$




Toolkit Bisection using

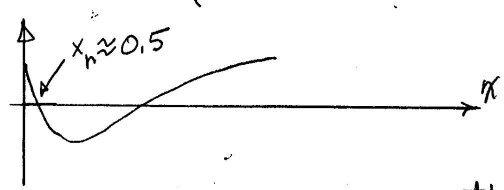
$x_l = 175$ $\quad x_u = 250$ $\quad \epsilon_s = 0.0001\ \%$

gives $x_r = 201.246$ in 19 iterations

$$y = \frac{1.75}{120(50000)(30{,}000)(450)}\left[-(201.2)^5 + 2(450)^2(201.2)^3 - (450)^4(201.2)\right] = -0.11411 \text{ cm}$$

8.19

$$f(x) = 10 - 20\left(e^{-0.2x} - e^{-0.75x}\right)$$



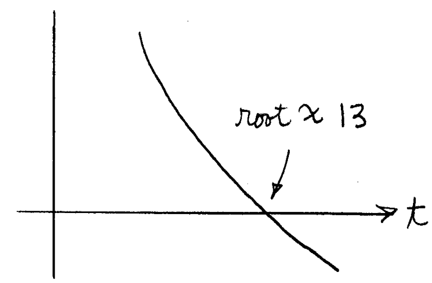

using $\qquad$ with

$x_l = 0$ $\quad x_u = 2$ $\quad \epsilon_s = 0.0001\ \%$

gives $x_r = 0.6023555$ in 22 iterations

8.20

$$f(t) = 70e^{-1.5t} + 25e^{-0.075t} - 9$$




a) $f'(t) = -105e^{-1.5t} - 1.875e^{-0.075t}$

$$t_{i+1} = t_i - \frac{(70e^{-1.5t_i} + 25e^{-0.075t} - 9)}{-105e^{-1.5t_i} - 1.875e^{-0.075t_i}}$$

| iter | $t_{i+1}$ |
|---|---|
| 1 | 13.60773 |
| 2 | 13.62201 |
| 3 | 13.62201 |

b)

| iter | $t_{i+1}$ |
|---|---|
| 1 | 13.76785 |
| 2 | 13.58012 |
| 3 | 13.6335 |
| 4 | 13.61882 |
| 5 | 13.6229 |

with $\delta = 0.5$

c) with $t_{i-1} = 12$ and $t_i = 14$

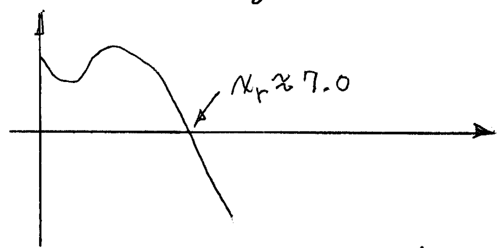
| iter | $t_{i+1}$ |
|---|---|
| 1 | 13.64464 |
| 2 | 13.62169 |
| 3 | 13.62202 |
| 4 | 13.62202 |

8.21

$$0.4 = \sin\frac{2\pi x}{16}\cos\frac{2\pi(12)(48)}{16} + e^{-x}$$

or

$$f(x) = \sin\left(\frac{\pi}{8}x\right) + e^{-x} - 0.4$$



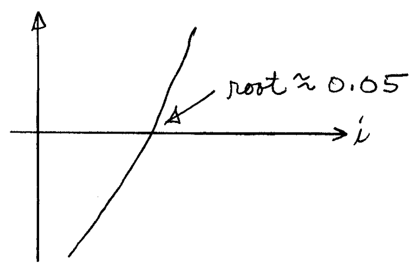

use Toolkit Bisection with

$x_l = 5$, $x_u = 10$, $\epsilon_s = 0.0001\%$

gives $x_r = 6.954732$ in 20 iterations

8.22

$$f(i) = 20000\,\frac{i(1+i)^6}{(1+i)^6 - 1} - 4000$$




Use Bisection from Toolkit with $i_l = 0.01$ $i_u = 0.10$

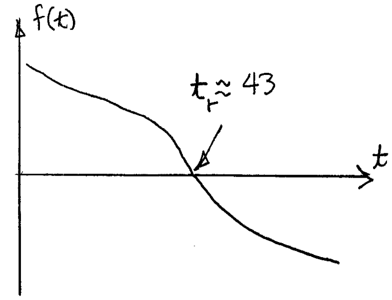
gives $i_r = 0.05471792$ with

$\epsilon_s = 0.0001\%$ in 21 iterations

8.23

$$f(t) = 75000\,e^{-0.05t} + 100000 - 1.2\left[\frac{300000}{1 + 59\,e^{-0.075t}}\right]$$

a)




b) use $t_{i-1} = 30$ $t_i = 50$

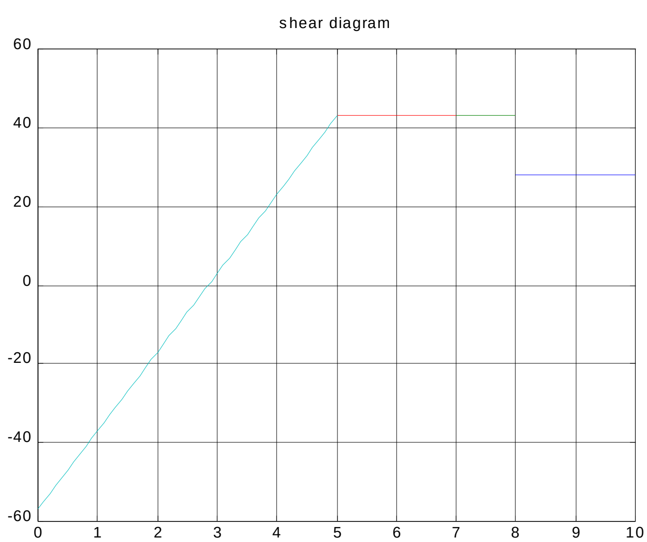
| iter | $t_{i+1}$ |
|---|---|
| 1 | 42.95521 |
| 2 | 43.17021 |
| 3 | 43.18433 |

c) use $t_0 = 40$ $\delta = 0.1$

| iter | $t_{i+1}$ |
|---|---|
| 1 | 43.15203 |
| 2 | 43.18381 |
| 3 | 43.18524 |

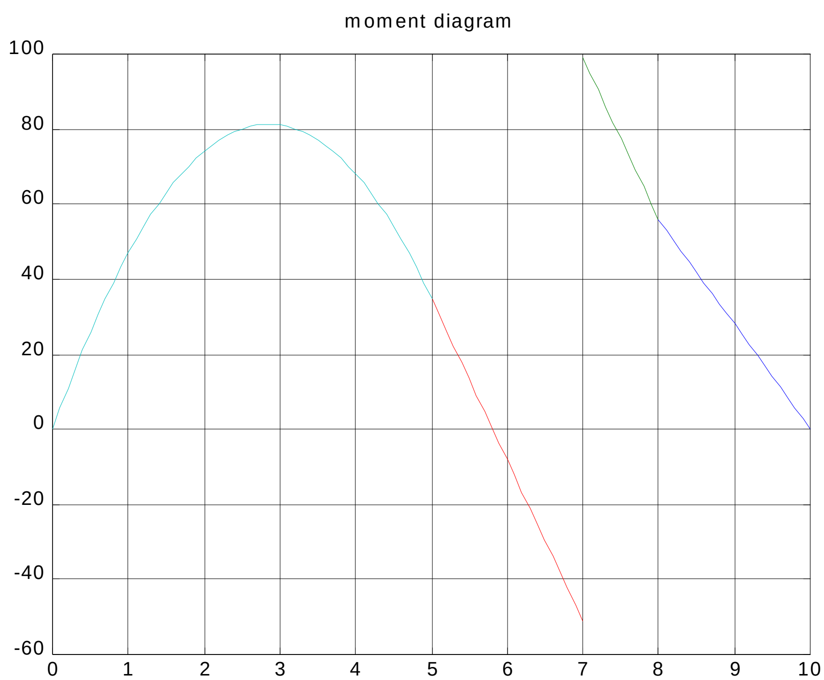
8.24

```
%Region from x=8 to x=10
x1=[8:.1:10];
y1=20*(x1-(x1-5))-15-57;
figure (1)
plot(x1,y1)
grid

%Region from x=7 to x=8
x2=[7:.1:8];
y2=20*(x2-(x2-5))-57;
figure (2)
plot(x2,y2)
grid

%Region from x=5 to x=7
x3=[5:.1:7];
y3=20*(x3-(x3-5))-57;
figure (3)
plot(x3,y3)
grid

%Region from x=0 to x=5
x4=[0:.1:5];
y4=20*x4-57;
figure (4)
plot(x4,y4)
grid

%Region from x=0 to x=10
figure (5)
plot(x1,y1,x2,y2,x3,y3,x4,y4)
grid
title('shear diagram')

a=[20 -57]
roots(a)

a =
    20   -57

ans =
    2.8500
```



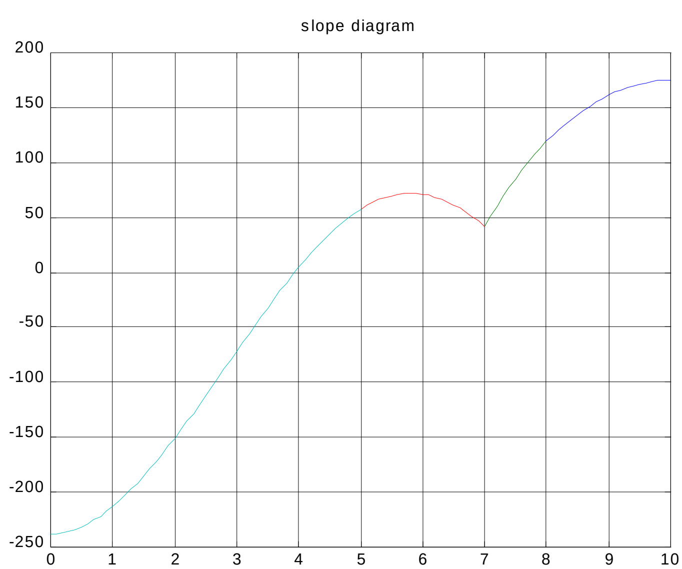
8.25

```
%Region from x=7 to x=8
x2=[7:.1:8];
y2=-10*(x2.^2-(x2-5).^2)+150+57*x2;
figure (2)
plot(x2,y2)
grid

%Region from x=5 to x=7
x3=[5:.1:7];
y3=-10*(x3.^2-(x3-5).^2)+57*x3;
figure (3)
plot(x3,y3)
grid

%Region from x=0 to x=5
x4=[0:.1:5];
y4=-10*(x4.^2)+57*x4;
figure (4)
plot(x4,y4)
grid

%Region from x=0 to x=10
figure (5)
plot(x1,y1,x2,y2,x3,y3,x4,y4)
grid
title('moment diagram')

a=[-43 250]
roots(a)

a =
   -43   250

ans =
    5.8140
```



8.26 A Matlab script can be used to determine that the slope equals zero at x = 3.94 m.

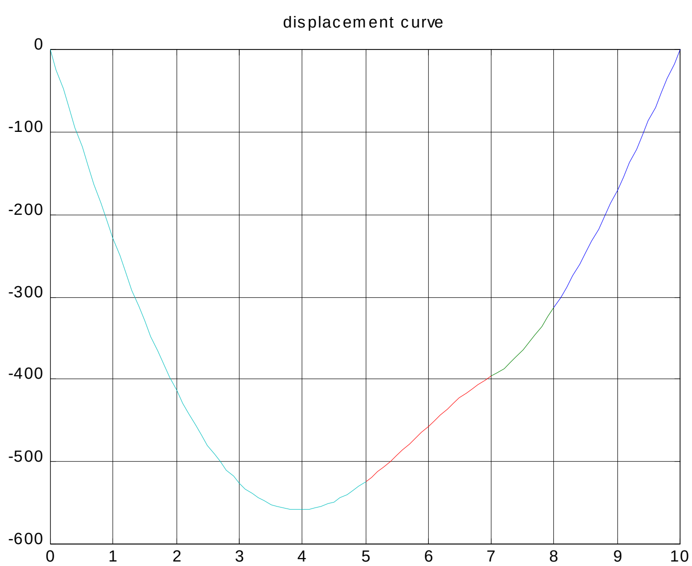
```
%Region from x=8 to x=10
x1=[8:.1:10];
y1=((-10/3)*(x1.^3-(x1-5).^3))+7.5*(x1-8).^2+150*(x1-7)+(57/2)*x1.^2-
238.25;
figure (1)
plot(x1,y1)
grid
%Region from x=7 to x=8
x2=[7:.1:8];
y2=((-10/3)*(x2.^3-(x2-5).^3))+150*(x2-7)+(57/2)*x2.^2-238.25;
figure (2)
plot(x2,y2)
grid
%Region from x=5 to x=7
x3=[5:.1:7];
y3=((-10/3)*(x3.^3-(x3-5).^3))+(57/2)*x3.^2-238.25;
figure (3)
plot(x3,y3)
grid
%Region from x=0 to x=5
x4=[0:.1:5];
y4=((-10/3)*(x4.^3))+(57/2)*x4.^2-238.25;
figure (4)
plot(x4,y4)
grid
%Region from x=0 to x=10
figure (5)
plot(x1,y1,x2,y2,x3,y3,x4,y4)
grid
title('slope diagram')
a=[-10/3 57/2 0 -238.25]
roots(a)

a =
   -3.3333   28.5000         0 -238.2500
ans =
    7.1531
    3.9357
   -2.5388
```




8.27

```
%Region from x=8 to x=10
x1=[8:.1:10];
y1=(-5/6)*(x1.^4-(x1-5).^4)+(15/6)*(x1-8).^3+75*(x1-7).^2+(57/6)*x1.^3-
238.25*x1;
figure (1)
plot(x1,y1)
grid
%Region from x=7 to x=8
x2=[7:.1:8];
y2=(-5/6)*(x2.^4-(x2-5).^4)+75*(x2-7).^2+(57/6)*x2.^3-238.25*x2;
figure (2)
plot(x2,y2)
grid
%Region from x=5 to x=7
x3=[5:.1:7];
y3=(-5/6)*(x3.^4-(x3-5).^4)+(57/6)*x3.^3-238.25*x3;
figure (3)
plot(x3,y3)
grid
%Region from x=0 to x=5
x4=[0:.1:5];
y4=(-5/6)*(x4.^4)+(57/6)*x4.^3-238.25*x4;
figure (4)
plot(x4,y4)
grid
%Region from x=0 to x=10
figure (5)
plot(x1,y1,x2,y2,x3,y3,x4,y4)
grid
title('displacement curve')

a =
   -3.3333   28.5000         0 -238.2500
ans =
    7.1531
    3.9357
   -2.5388
```

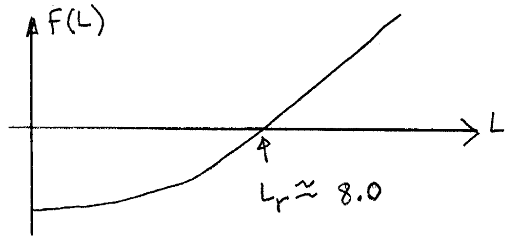
Therefore, other than the end supports, there are no points of zero displacement along the beam.



8.28

$$f(L) = e^{-Rt/2L} \cos\left[\sqrt{\frac{1}{LC} - \left(\frac{R}{2L}\right)^2}\ t\right] - \frac{q}{q_0}$$

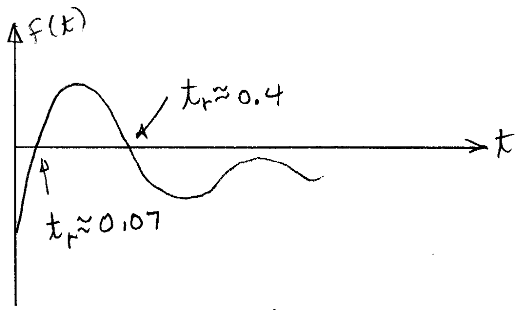
$$f(L) = e^{-7/L} \cos\left[\sqrt{\frac{10000}{L} - \frac{19600}{L^2}}\ (0.05)\right] - .01$$




Use Bisection from Toolkit with
$L_l = 5 \quad L_u = 10 \quad \epsilon_s = 0.0001\ \%$

gives $L_r = 7.841477$ in 20 iterations

8.29 $f(t) = 9e^{-t} \sin 2\pi t - 3.5$




Use Toolkit Bisection

$t_l = 0 \quad t_u = 0.2 \quad \epsilon_a = 0.0001$

$t_r = 0.0683543$ iterations = 22

$t_l = 0.2 \quad t_u = 0.5 \quad \epsilon_s = 0.0001$

$t_r = 0.4013436$ iterations = 20

8.30

$\mu = 72.19134$

$n = 1.44 \times 10^{10}$

$$f(N) = 1.44\times10^{10} - \frac{N}{2} - \frac{1}{2}\sqrt{N^2 + 4(6.21)^2\times10^{18}}$$

a) Use Toolkit Bisection

$N_l = 10^{10} \quad N_u = 2\times10^{10}$

gives $N_r = 1.172195 \times 10^{10}$

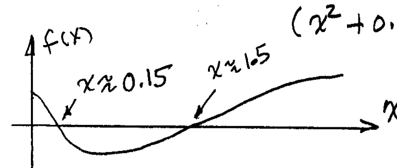
in 20 iterations

b) use $N_0 = 1\times10^9$
$\delta = 0.5$

| iter | $N_{i+1}$ |
|---|---|
| 1 | $1.494373 \times 10^{10}$ |
| 2 | $1.18959 \times 10^{10}$ |
| 3 | $1.172931 \times 10^{10}$ |
| 4 | $1.172224 \times 10^{10}$ |
| 5 | $1.172195 \times 10^{10}$ |

8.31

$$f(x) = 1 - \frac{3.5967\ x}{(x^2 + 0.64)^{3/2}}$$




Toolkit gives

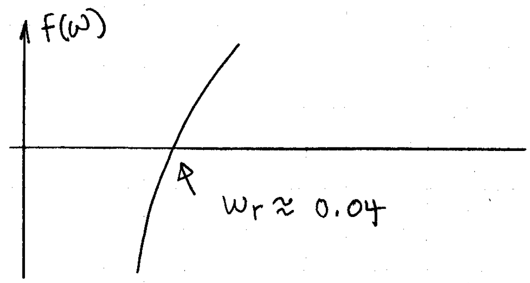
$x_r = 0.1499167$

or

$x_r = 1.606054$

8.32

$$f(\omega) = 100 - \sqrt{\frac{1}{(225)^2} + \left(0.6\times10^{-6}\,\omega - \frac{1}{\omega(.5)^2}\right)^2}$$




Use Toolkit Bisection with

$\omega_l = 0.01 \quad \omega_u = 0.1 \quad \epsilon_s = 0.0001\ \%$

gives $\omega_r = 0.03999999$ in 22 iterations

8.33 Student Specific, but for Re = 10,000

$$f(f) = \frac{1}{\sqrt{f}} - 4\log_{10}\left[10000\sqrt{f}\right] - 0.4$$

Bisection method gives

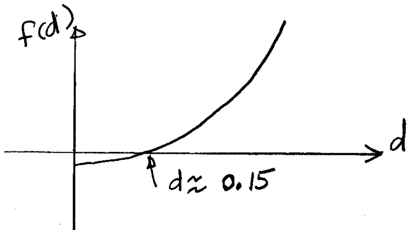
$f = 0.006860978$

for Re = 500,000
$f = 0.003032451$

in 22 iterations with
$\epsilon_s = 0.0001\ \%$

8.34

$$f(d) = 16d^{5/2} + 20000\,d^2 - 931d - 400.33$$




use Toolkit with $d_l = 0$
$d_u = 1.0$

gives $d_r = 0.166625$
in 20 iterations

8.35

$$f(T) = 0.99403 + 1.671\times10^{-4}\,T + 9.7215\times10^{-8}\,T^2 - 9.5838\times10^{-11}\,T^3 + 1.952\times10^{-14}\,T^4 - 1.2$$

use modified secant method with $T_0 = 500$
$\delta = 0.5$

| iter | $T_{i+1}$ | $\epsilon_a$ |
|---|---|---|
| 1 | 1059.383 | 52 |
| 2 | 1149.398 | 7.8 |
| 3 | 1113.68 | 3.2 |
| 4 | 1131.586 | 1.6 |
| 5 | 1123.283 | 0.7 |
| 6 | 1127.296 | 0.4 |
| 7 | 1125.393 | 0.2 |

etc

8.36 we want

$y = 1.8$ at $x = 0$
$y = 1.0$ at $x = 40$

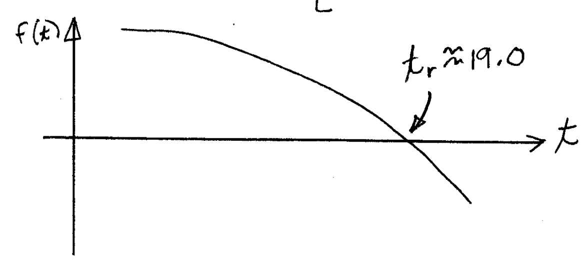
$$f(y) = \tan(2\pi\theta/360)\,x - \frac{9.8\,x^2}{800\cos^2(2\pi\theta/360)} + 0.8 = 0$$

use Toolkit gives two acceptable values

$\theta = 36.277^\circ$ or
$\theta = 52.577$

8.37

$$f(t) = 1000 - 2200\ln\left[\frac{160000}{160000 - 2680t}\right] - 9.8t$$




use Toolkit Bisection with

$t_l = 0$ $t_u = 20$ or
$t_l = 10$ $t_u = 50$

each gives $t_r = 18.54363$

in 21 or 22 iterations with

$\epsilon_s = 0.0001\%$

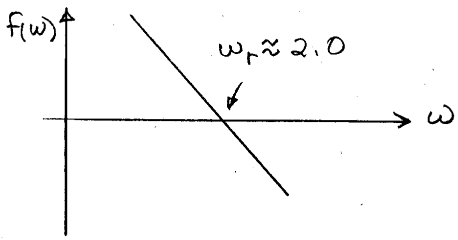
8.38

$c/c_0 = 0.1221$

$p = 34.12$

$$f(\omega) = \tan^{-1}\left[\frac{2(0.1221)\dfrac{\omega}{34.12}}{1 - \omega^2/34.12^2}\right] - \frac{\omega}{2} + 1 = 0$$

Use Toolkit Bisection

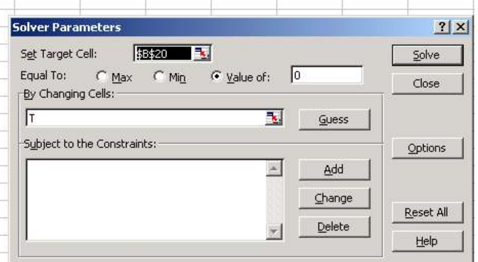



$\omega_l = 0 \quad \omega_u = 10 \quad \epsilon_s = 0.0001\,\%$

gives $\omega_r = 2.029146$ in 23 iterations

8.39 Excel Solver solution:

| | A | B | C | D | E | F | G |
|---|---|---|---|---|---|---|---|
| 1 | Prob. 8.39 | | | | | | |
| 2 | | | | | | | |
| 3 | In: | | | | | | |
| 4 | TA | 400 | | | | | |
| 5 | CpA | 9.909 | <---------- | =3.381+0.01804*TA-0.0000043*TA^2 | | | |
| 6 | FlowA | 2 | | | | | |
| 7 | | | | | | | |
| 8 | TB | 600 | | | | | |
| 9 | CpB | 71.3112 | <---------- | =8.592+0.0129*TB-0.00004078*TB^2 | | | |
| 10 | FlowB | 1 | | | | | |
| 11 | | | | | | | |
| 12 | Heatin | 50713.92 | <---------- | =FlowA*CpA*TA+FlowB*CpB*TB | | | |
| 13 | | | | | | | |
| 14 | T | 500 | | | | | |
| 15 | CpAout | 11.326 | <---------- | =3.381+0.01804*T-0.0000043*T^2 | | | |
| 16 | CpBout | 62.897 | <---------- | =8.592+0.0129*T-0.00004078*T^2 | | | |
| 17 | | | | | | | |
| 18 | Heatout: | 42774.5 | <---------- | =FlowA*CpAout*T+FlowB*CpBout*T | | | |
| 19 | | | | | | | |
| 20 | Net | 7939.42 | <---------- | =Heatin-Heatout | | | |

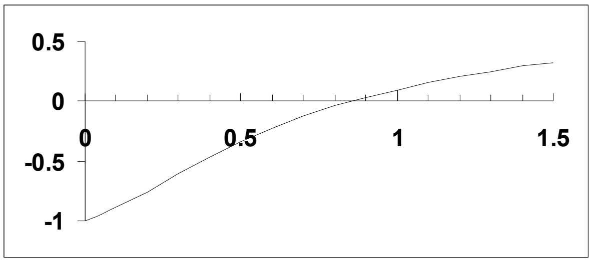



| | A | B | C | D | E | F | G |
|---|---|---|---|---|---|---|---|
| 1 | Prob. 8.39 | | | | | | |
| 2 | | | | | | | |
| 3 | In: | | | | | | |
| 4 | TA | 400 | | | | | |
| 5 | CpA | 9.909 | <---------- | =3.381+0.01804*TA-0.0000043*TA^2 | | | |
| 6 | FlowA | 2 | | | | | |
| 7 | | | | | | | |
| 8 | TB | 600 | | | | | |
| 9 | CpB | 71.3112 | <---------- | =8.592+0.0129*TB-0.00004078*TB^2 | | | |
| 10 | FlowB | 1 | | | | | |
| 11 | | | | | | | |
| 12 | Heatin | 50713.92 | <---------- | =FlowA*CpA*TA+FlowB*CpB*TB | | | |
| 13 | | | | | | | |
| 14 | T | 553.5959 | | | | | |
| 15 | CpAout | 12.05006 | <---------- | =3.381+0.01804*T-0.0000043*T^2 | | | |
| 16 | CpBout | 67.50809 | <---------- | =8.592+0.0129*T-0.00004078*T^2 | | | |
| 17 | | | | | | | |
| 18 | Heatout: | 50713.92 | <---------- | =FlowA*CpAout*T+FlowB*CpBout*T | | | |
| 19 | | | | | | | |
| 20 | Net | 4.57E-07 | <---------- | =Heatin-Heatout | | | |

8.40 The problem reduces to finding the value of n that drives the second part of the equation to 1. In other words, finding the root of

$$f(n) = \frac{n}{n-1}\left(R_c^{(n-1)/n} - 1\right) - 1 = 0$$

Inspection of the equation indicates that singularities occur at $x$ = 0 and 1. A plot indicates that otherwise, the function is smooth.




A tool such as the Excel Solver can be used to locate the root at $n$ = 0.8518.

8.41 The sequence of calculation need to compute the pressure drop in each pipe is

$$A = \pi(D/2)^2$$

$$v = \frac{Q}{A}$$

$$\text{Re} = \frac{D\rho v}{\mu}$$

$$f = \text{root}\left[4.0\log\left(\text{Re}\sqrt{f}\right) - 0.4 - \frac{1}{\sqrt{f}}\right]$$

$$\Delta P = f\,\frac{\rho v^2}{2D}$$

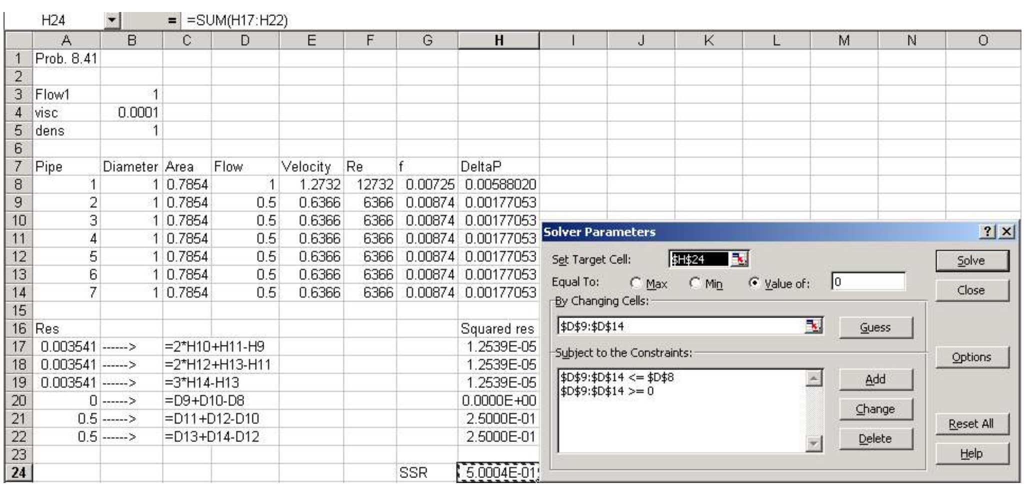
The six balance equations can then be solved for the 6 unknowns.

The root location can be solved with a technique like the modified false position method. A bracketing method is advisable since initial guesses that bound the normal range of friction factors can be readily determined. The following VBA function procedure is designed to do this

```
Option Explicit

Function FalsePos(Re)
Dim iter As Integer, imax As Integer
Dim il As Integer, iu As Integer
Dim xrold As Single, fl As Single, fu As Single, fr As Single
Dim xl As Single, xu As Single, es As Single
Dim xr As Single, ea As Single

xl = 0.00001
xu = 1
```

```
es = 0.01
imax = 40
iter = 0
fl = f(xl, Re)
fu = f(xu, Re)
Do
  xrold = xr
  xr = xu - fu * (xl - xu) / (fl - fu)
  fr = f(xr, Re)
  iter = iter + 1
  If xr <> 0 Then
    ea = Abs((xr - xrold) / xr) * 100
  End If
  If fl * fr < 0 Then
    xu = xr
    fu = f(xu, Re)
    iu = 0
    il = il + 1
    If il >= 2 Then fl = fl / 2
  ElseIf fl * fr > 0 Then
    xl = xr
    fl = f(xl, Re)
    il = 0
    iu = iu + 1
    If iu >= 2 Then fu = fu / 2
  Else
    ea = 0#
  End If
  If ea < es Or iter >= imax Then Exit Do
Loop
FalsePos = xr
End Function

Function f(x, Re)
f = 4 * Log(Re * Sqr(x)) / Log(10) - 0.4 - 1 / Sqr(x)
End Function
```

The following Excel spreadsheet can be set up to solve the problem. Note that the function call, `=falsepos(F8)`, is entered into cell G8 and then copied down to G9:G14. This invokes the function procedure so that the friction factor is determined at each iteration.


H24 = =SUM(H17:H22)

| | A | B | C | D | E | F | G | H |
|---|---|---|---|---|---|---|---|---|
| 1 | Prob. 8.41 | | | | | | | |
| 2 | | | | | | | | |
| 3 | Flow1 | 1 | | | | | | |
| 4 | visc | 0.0001 | | | | | | |
| 5 | dens | 1 | | | | | | |
| 6 | | | | | | | | |
| 7 | Pipe | Diameter | Area | Flow | Velocity | Re | f | DeltaP |
| 8 | 1 | 1 | 0.7854 | 1 | 1.2732 | 12732 | 0.00725 | 0.00588020 |
| 9 | 2 | 1 | 0.7854 | 0.5 | 0.6366 | 6366 | 0.00874 | 0.00177053 |
| 10 | 3 | 1 | 0.7854 | 0.5 | 0.6366 | 6366 | 0.00874 | 0.00177053 |
| 11 | 4 | 1 | 0.7854 | 0.5 | 0.6366 | 6366 | 0.00874 | 0.00177053 |
| 12 | 5 | 1 | 0.7854 | 0.5 | 0.6366 | 6366 | 0.00874 | 0.00177053 |
| 13 | 6 | 1 | 0.7854 | 0.5 | 0.6366 | 6366 | 0.00874 | 0.00177053 |
| 14 | 7 | 1 | 0.7854 | 0.5 | 0.6366 | 6366 | 0.00874 | 0.00177053 |
| 15 | | | | | | | | |
| 16 | Res | | | | | | | Squared res |
| 17 | 0.003541 | ------> | =2*H10+H11-H9 | | | | | 1.2539E-05 |
| 18 | 0.003541 | ------> | =2*H12+H13-H11 | | | | | 1.2539E-05 |
| 19 | 0.003541 | ------> | =3*H14-H13 | | | | | 1.2539E-05 |
| 20 | 0 | ------> | =D9+D10-D8 | | | | | 0.0000E+00 |
| 21 | 0.5 | ------> | =D11+D12-D10 | | | | | 2.5000E-01 |
| 22 | 0.5 | ------> | =D13+D14-D12 | | | | | 2.5000E-01 |
| 23 | | | | | | | | |
| 24 | | | | | | | SSR | 5.0004E-01 |



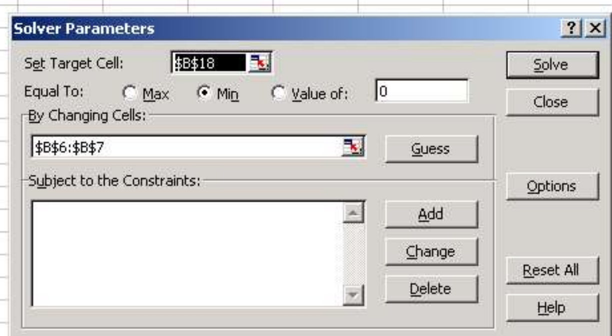
The resulting final solution is

| | A | B | C | D | E | F | G | H |
|---|---|---|---|---|---|---|---|---|
| 1 | Prob. 8.41 | | | | | | | |
| 2 | | | | | | | | |
| 3 | Flow1 | 1 | | | | | | |
| 4 | visc | 0.0001 | | | | | | |
| 5 | dens | 1 | | | | | | |
| 6 | | | | | | | | |
| 7 | Pipe | Diameter | Area | Flow | Velocity | Re | f | DeltaP |
| 8 | 1 | 1 | 0.7854 | 1 | 1.2732 | 12732 | 0.00725 | 0.00588020 |
| 9 | 2 | 1 | 0.7854 | 0.656439 | 0.8358 | 8358 | 0.00811 | 0.00283172 |
| 10 | 3 | 1 | 0.7854 | 0.343545 | 0.4374 | 4374 | 0.00973 | 0.00093051 |
| 11 | 4 | 1 | 0.7854 | 0.16715 | 0.2128 | 2128 | 0.01213 | 0.00027470 |
| 12 | 5 | 1 | 0.7854 | 0.176369 | 0.2246 | 2246 | 0.01192 | 0.00030062 |
| 13 | 6 | 1 | 0.7854 | 0.095502 | 0.1216 | 1216 | 0.01462 | 0.00010810 |
| 14 | 7 | 1 | 0.7854 | 0.080861 | 0.1030 | 1030 | 0.01550 | 0.00008216 |
| 15 | | | | | | | | |
| 16 | Res | | | | | | | Squared res |
| 17 | -0.0007 | ------> | =2*H10+H11-H9 | | | | | 4.8441E-07 |
| 18 | 0.000435 | ------> | =2*H12+H13-H11 | | | | | 1.8893E-07 |
| 19 | 0.000138 | ------> | =3*H14-H13 | | | | | 1.9152E-08 |
| 20 | -1.7E-05 | ------> | =D9+D10-D8 | | | | | 2.7258E-10 |
| 21 | -2.6E-05 | ------> | =D11+D12-D10 | | | | | 6.6336E-10 |
| 22 | -5.9E-06 | ------> | =D13+D14-D12 | | | | | 3.4572E-11 |
| 23 | | | | | | | | |
| 24 | | | | | | | SSR | 6.9346E-07 |

8.42 The following application of Excel Solver can be set up:

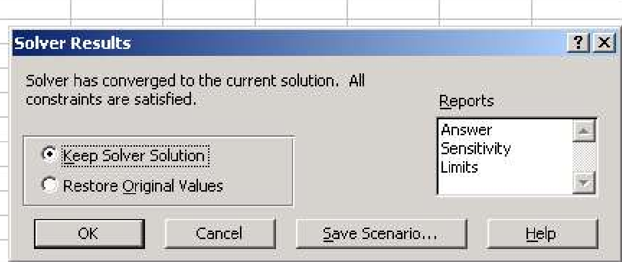
B18 = =SUM(B14:B16)

| | A | B |
|---|---|---|
| 1 | Prob0842 | |
| 2 | | |
| 3 | T0 | 500 |
| 4 | T3 | 25 |
| 5 | | |
| 6 | T1 | 200 |
| 7 | T2 | 100 |
| 8 | | |
| 9 | q1 | 306.9862 |
| 10 | q2 | 400 |
| 11 | q3 | 411.1734 |
| 12 | | |
| 13 | | |
| 14 | (q1-q2)^2 | 8651.559 |
| 15 | (q1-q3)^2 | 10854.97 |
| 16 | (q2-q3)^2 | 124.8454 |
| 17 | | |
| 18 | Sum | 19631.37 |

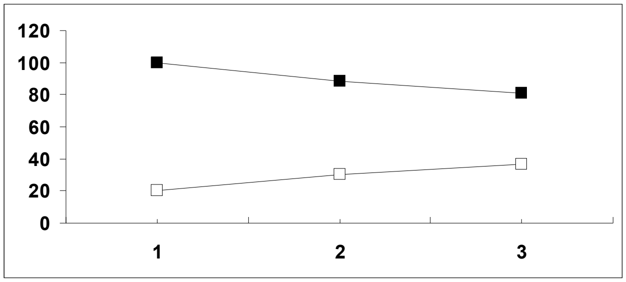



The solution is:

| | A | B |
|---|---|---|
| 1 | Prob0842 | |
| 2 | | |
| 3 | T0 | 500 |
| 4 | T3 | 25 |
| 5 | | |
| 6 | T1 | 166.9741 |
| 7 | T2 | 87.08195 |
| 8 | | |
| 9 | q1 | 319.5688 |
| 10 | q2 | 319.5685 |
| 11 | q3 | 319.5684 |
| 12 | | |
| 13 | | |
| 14 | (q1-q2)^2 | 8.75E-08 |
| 15 | (q1-q3)^2 | 1.44E-07 |
| 16 | (q2-q3)^2 | 7.05E-09 |
| 17 | | |
| 18 | Sum | 2.39E-07 |




8.43 The results are



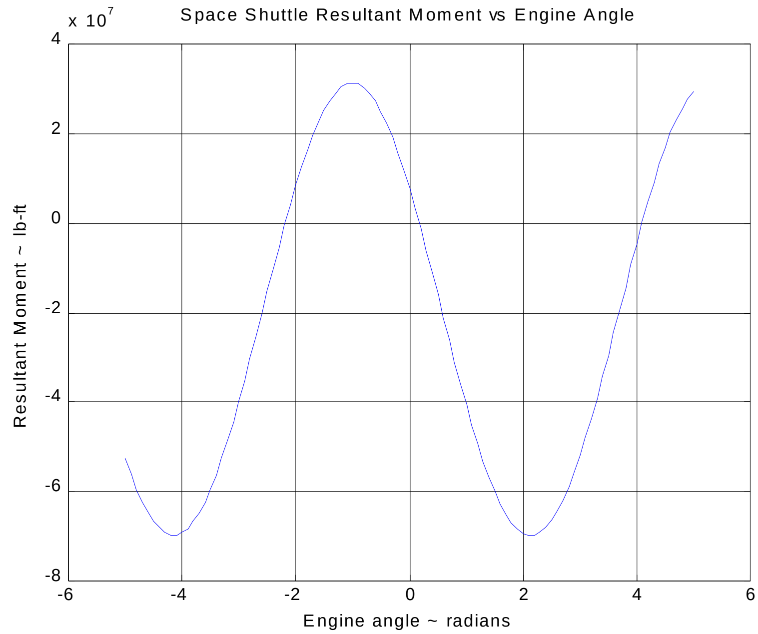

8.44

```
    %  Shuttle Liftoff Engine Angle
    %  Newton-Raphson Method of iteratively finding a single root
format long
    %     Constants
  LGB = 4.0; LGS = 24.0; LTS = 38.0;
  WS = 0.230E6; WB = 1.663E6;
  TB = 5.3E6; TS = 1.125E6;
  es = 0.5E-7; nmax = 200;
    %     Initial estimate in radians
  x = 0.25
   %Calculation loop
for i=1:nmax
    fx = LGB*WB-LGB*TB-LGS*WS+LGS*TS*cos(x)-LTS*TS*sin(x);
    dfx = -LGS*TS*sin(x)-LTS*TS*cos(x);
    xn=x-fx/dfx;
          %convergence check
    ea=abs((xn-x)/xn);
    if (ea<=es)
          fprintf('convergence: Root = %f radians \n',xn)
          theta = (180/pi)*x;
          fprintf('Engine Angle = %f  degrees \n',theta)
          break
    end
  x=xn;
              x
end
```

```
    %     Shuttle Liftoff Engine Angle
    %  Newton-Raphson Method of iteratively finding a single root
    %  Plot of Resultant Moment vs Engine Anale
format long
    %     Constants
  LGB = 4.0; LGS = 24.0; LTS = 38.0;
  WS = 0.195E6; WB = 1.663E6;
  TB = 5.3E6; TS = 1.125E6;

  x=-5:0.1:5;
  fx = LGB*WB-LGB*TB-LGS*WS+LGS*TS*cos(x)-LTS*TS*sin(x);

  plot(x,fx)
  grid
  axis([-6 6 -8e7 4e7])
  title('Space Shuttle Resultant Moment vs Engine Angle')
  xlabel('Engine angle ~ radians')
  ylabel('Resultant Moment ~ lb-ft')
```

```
x =
   0.25000000000000

x =
   0.15678173034564

x =
   0.15518504730788

x =
   0.15518449747125
```

```
convergence: Root = 0.155184 radians
Engine Angle = 8.891417  degrees
```




8.45 This problem was solved using the roots command in Matlab.

```
c =
            1          -33          -704         -1859

roots(c)

ans =

   48.3543
  -12.2041
   -3.1502
```

Therefore,

$\sigma_1 = 48.4$ Mpa $\quad \sigma_2 = -3.15$ MPa $\quad \sigma_3 = -12.20$ MPa

| | $T$ | $t$ |
|---|---|---|
| 1 | 100 | 20 |
| 2 | 88.31493 | 30.1157 |
| 3 | 80.9082 | 36.53126 |

# CHAPTER 3

3.1 Here is a VBA implementation of the algorithm:

```
Option Explicit

Sub GetEps()
Dim epsilon As Single
epsilon = 1
Do
  If epsilon + 1 <= 1 Then Exit Do
  epsilon = epsilon / 2
Loop
epsilon = 2 * epsilon
MsgBox epsilon
End Sub
```

It yields a result of $1.19209 \times 10^{-7}$ on my desktop PC.

3.2 Here is a VBA implementation of the algorithm:

```
Option Explicit

Sub GetMin()
Dim x As Single, xmin As Single
x = 1
Do
  If x <= 0 Then Exit Do
  xmin = x
  x = x / 2
Loop
MsgBox xmin
End Sub
```

It yields a result of $1.4013 \times 10^{-45}$ on my desktop PC.

3.3 The maximum negative value of the exponent for a computer that uses $e$ bits to store the exponent is

$$e_{\min} = -(2^{e-1} - 1)$$

Because of normalization, the minimum mantissa is $1/b = 2^{-1} = 0.5$. Therefore, the minimum number is

$$x_{\min} = 2^{-1} 2^{-(2^{e-1}-1)} = 2^{-2^{e-1}}$$

For example, for an 8-bit exponent

$$x_{\min} = 2^{-2^{8-1}} = 2^{-128} = 2.939 \times 10^{-39}$$

This result contradicts the value from Prob. 3.2 ($1.4013 \times 10^{-45}$). This amounts to an additional 21 divisions (i.e., 21 orders of magnitude lower in base 2). I do not know the reason for the discrepancy. However, the problem illustrates the value of determining such quantities via a program rather than relying on theoretical values.

3.4 VBA Program to compute in ascending order

```
Option Explicit

Sub Series()

Dim i As Integer, n As Integer
Dim sum As Single, pi As Single

pi = 4 * Atn(1)
sum = 0
n = 10000
For i = 1 To n
  sum = sum + 1 / i ^ 2
Next i

MsgBox sum
'Display true percent relatve error
MsgBox Abs(sum - pi ^ 2 / 6) / (pi ^ 2 / 6)

End Sub
```

This yields a result of 1.644725 with a true relative error of $6.086\times10^{-5}$.

VBA Program to compute in descending order:

```
Option Explicit

Sub Series()

Dim i As Integer, n As Integer
Dim sum As Single, pi As Single

pi = 4 * Atn(1)
sum = 0
n = 10000
For i = n To 1 Step -1
  sum = sum + 1 / i ^ 2
Next i

MsgBox sum
'Display true percent relatve error
MsgBox Abs(sum - pi ^ 2 / 6) / (pi ^ 2 / 6)

End Sub
```

This yields a result of 1.644725 with a true relative error of $1.270\times10^{-4}$

The latter version yields a superior result because summing in descending order mitigates the roundoff error that occurs when adding a large and small number.

3.5 Remember that the machine epsilon is related to the number of significant digits by Eq. 3.11

$$\xi = b^{1-t}$$

which can be solved for base 10 and for a machine epsilon of $1.19209\times10^{-7}$ for

$$t = 1 - \log_{10}(\xi) = 1 - \log_{10}(1.19209 \times 10^{-7}) = 7.92$$

To be conservative, assume that 7 significant figures is good enough. Recall that Eq. 3.7 can then be used to estimate a stopping criterion,

$$\varepsilon_s = (0.5 \times 10^{2-n})\%$$

Thus, for 7 significant digits, the result would be

$$\varepsilon_s = (0.5 \times 10^{2-7})\% = 5 \times 10^{-6}\%$$

The total calculation can be expressed in one formula as

$$\varepsilon_s = (0.5 \times 10^{2-\text{Int}(1-\log_{10}(\xi))})\%$$

It should be noted that iterating to the machine precision is often overkill. Consequently, many applications use the old engineering rule of thumb that you should iterate to 3 significant digits or better.

As an application, I used Excel to evaluate the second series from Prob. 3.6. The results are:

| | A | B | C | D | E | F | G | H |
|---|---|---|---|---|---|---|---|---|
| 1 | x | 8.3 | | | | | | |
| 2 | n | n! | x^n/n! | Series | 1/Series | True Value | et (%) | ea(%) |
| 3 | 0 | 1 | 1 | 1.000000E+00 | 1.000000E+00 | 2.485168E-04 | 4.02E+05 | |
| 4 | 1 | 1 | 8.3 | 9.300000E+00 | 1.075269E-01 | 2.485168E-04 | 4.32E+04 | 9.98E+01 |
| 5 | 2 | 2 | 34.445 | 4.374500E+01 | 2.285976E-02 | 2.485168E-04 | 9.10E+03 | 9.89E+01 |
| 6 | 3 | 6 | 95.29783 | 1.390428E+02 | 7.192028E-03 | 2.485168E-04 | 2.79E+03 | 9.65E+01 |
| 7 | 4 | 24 | 197.743 | 3.367858E+02 | 2.969246E-03 | 2.485168E-04 | 1.09E+03 | 9.16E+01 |
| 8 | 5 | 120 | 328.2534 | 6.650392E+02 | 1.503671E-03 | 2.485168E-04 | 5.05E+02 | 8.35E+01 |
| 9 | 6 | 720 | 454.0839 | 1.119123E+03 | 8.935568E-04 | 2.485168E-04 | 2.60E+02 | 7.22E+01 |
| 10 | 7 | 5040 | 538.4137 | 1.657537E+03 | 6.033049E-04 | 2.485168E-04 | 1.43E+02 | 5.88E+01 |
| 11 | 8 | 40320 | 558.6042 | 2.216141E+03 | 4.512348E-04 | 2.485168E-04 | 8.16E+01 | 4.49E+01 |
| 12 | 9 | 362880 | 515.1572 | 2.731298E+03 | 3.661263E-04 | 2.485168E-04 | 4.73E+01 | 3.21E+01 |
| 13 | 10 | 3628800 | 427.5805 | 3.158879E+03 | 3.165680E-04 | 2.485168E-04 | 2.74E+01 | 2.15E+01 |
| 14 | 11 | 39916800 | 322.6289 | 3.481508E+03 | 2.872319E-04 | 2.485168E-04 | 1.56E+01 | 1.35E+01 |
| 15 | 12 | 4.79E+08 | 223.1517 | 3.704659E+03 | 2.699304E-04 | 2.485168E-04 | 8.62E+00 | 7.93E+00 |
| 16 | 13 | 6.23E+09 | 142.4738 | 3.847133E+03 | 2.599338E-04 | 2.485168E-04 | 4.59E+00 | 4.39E+00 |
| 17 | 14 | 8.72E+10 | 84.46659 | 3.931600E+03 | 2.543494E-04 | 2.485168E-04 | 2.35E+00 | 2.29E+00 |
| 18 | 15 | 1.31E+12 | 46.73818 | 3.978338E+03 | 2.513613E-04 | 2.485168E-04 | 1.14E+00 | 1.13E+00 |
| 19 | 16 | 2.09E+13 | 24.24543 | 4.002583E+03 | 2.498386E-04 | 2.485168E-04 | 5.32E-01 | 5.29E-01 |
| 20 | 17 | 3.56E+14 | 11.83747 | 4.014421E+03 | 2.491019E-04 | 2.485168E-04 | 2.35E-01 | 2.35E-01 |
| 21 | 18 | 6.4E+15 | 5.458391 | 4.019879E+03 | 2.487637E-04 | 2.485168E-04 | 9.93E-02 | 9.92E-02 |
| 22 | 19 | 1.22E+17 | 2.384455 | 4.022264E+03 | 2.486162E-04 | 2.485168E-04 | 4.00E-02 | 4.00E-02 |
| 23 | 20 | 2.43E+18 | 0.989549 | 4.023253E+03 | 2.485551E-04 | 2.485168E-04 | 1.54E-02 | 1.54E-02 |
| 24 | 21 | 5.11E+19 | 0.391107 | 4.023644E+03 | 2.485309E-04 | 2.485168E-04 | 5.67E-03 | 5.67E-03 |
| 25 | 22 | 1.12E+21 | 0.147554 | 4.023792E+03 | 2.485218E-04 | 2.485168E-04 | 2.00E-03 | 2.00E-03 |
| 26 | 23 | 2.59E+22 | 0.053248 | 4.023845E+03 | 2.485185E-04 | 2.485168E-04 | 6.79E-04 | 6.79E-04 |
| 27 | 24 | 6.2E+23 | 0.018415 | 4.023863E+03 | 2.485174E-04 | 2.485168E-04 | 2.22E-04 | 2.22E-04 |
| 28 | 25 | 1.55E+25 | 0.006114 | 4.023870E+03 | 2.485170E-04 | 2.485168E-04 | 6.96E-05 | 6.96E-05 |
| 29 | 26 | 4.03E+26 | 0.001952 | 4.023872E+03 | 2.485169E-04 | 2.485168E-04 | 2.11E-05 | 2.11E-05 |
| 30 | **27** | **1.09E+28** | **0.0006** | **4.023872E+03** | **2.485168E-04** | **2.485168E-04** | **6.16E-06** | **6.16E-06** |
| 31 | 28 | 3.05E+29 | 0.000178 | 4.023872E+03 | 2.485168E-04 | 2.485168E-04 | 1.74E-06 | 1.74E-06 |
| 32 | 29 | 8.84E+30 | 5.09E-05 | 4.023872E+03 | 2.485168E-04 | 2.485168E-04 | 4.76E-07 | 4.76E-07 |
| 33 | 30 | 2.65E+32 | 1.41E-05 | 4.023872E+03 | 2.485168E-04 | 2.485168E-04 | 1.26E-07 | 1.26E-07 |

Notice how after summing 27 terms, the result is correct to 7 significant figures. At this point, both the true and the approximate percent relative errors are at $6.16 \times 10^{-6}$ %. At this point, the process would repeat one more time so that the error estimates would fall below the precalculated stopping criterion of $5 \times 10^{-6}$ %.

3.6 For the first series, after 25 terms are summed, the result is

| | A | B | C | D | E | F | G | H |
|---|---|---|---|---|---|---|---|---|
| 1 | x | 8.3 | | | | | | |
| 2 | n | n! | x^n/n! | Sign | Series | True Value | et (%) | ea(%) |
| 3 | 0 | 1 | 1 | 1 | 1.000000E+00 | 2.485168E-04 | 402287.2 | |
| 4 | 1 | 1 | 8.3 | -1 | -7.300000E+00 | 2.485168E-04 | 2937527 | 100.0034 |
| 5 | 2 | 2 | 34.445 | 1 | 2.714500E+01 | 2.485168E-04 | 10922702 | 99.99908 |
| 6 | 3 | 6 | 95.29783 | -1 | -6.815283E+01 | 2.485168E-04 | 27423930 | 100.0004 |
| 7 | 4 | 24 | 197.743 | 1 | 1.295902E+02 | 2.485168E-04 | 52145331 | 99.99981 |
| 8 | 5 | 120 | 328.2534 | -1 | -1.986632E+02 | 2.485168E-04 | 79939643 | 100.0001 |
| 9 | 6 | 720 | 454.0839 | 1 | 2.554206E+02 | 2.485168E-04 | 1.03E+08 | 99.9999 |
| 10 | 7 | 5040 | 538.4137 | -1 | -2.829931E+02 | 2.485168E-04 | 1.14E+08 | 100.0001 |
| 11 | 8 | 40320 | 558.6042 | 1 | 2.756111E+02 | 2.485168E-04 | 1.11E+08 | 99.99991 |
| 12 | 9 | 362880 | 515.1572 | -1 | -2.395461E+02 | 2.485168E-04 | 96390385 | 100.0001 |
| 13 | 10 | 3628800 | 427.5805 | 1 | 1.880344E+02 | 2.485168E-04 | 75662552 | 99.99987 |
| 14 | 11 | 39916800 | 322.6289 | -1 | -1.345945E+02 | 2.485168E-04 | 54159210 | 100.0002 |
| 15 | 12 | 4.79E+08 | 223.1517 | 1 | 8.855717E+01 | 2.485168E-04 | 35634175 | 99.99972 |
| 16 | 13 | 6.23E+09 | 142.4738 | -1 | -5.391659E+01 | 2.485168E-04 | 21695448 | 100.0005 |
| 17 | 14 | 8.72E+10 | 84.46659 | 1 | 3.055000E+01 | 2.485168E-04 | 12292829 | 99.99919 |
| 18 | 15 | 1.31E+12 | 46.73818 | -1 | -1.618818E+01 | 2.485168E-04 | 6514018 | 100.0015 |
| 19 | 16 | 2.09E+13 | 24.24543 | 1 | 8.057248E+00 | 2.485168E-04 | 3242034 | 99.99692 |
| 20 | 17 | 3.56E+14 | 11.83747 | -1 | -3.780226E+00 | 2.485168E-04 | 1521215 | 100.0066 |
| 21 | 18 | 6.4E+15 | 5.458391 | 1 | 1.678165E+00 | 2.485168E-04 | 675172.1 | 99.98519 |
| 22 | 19 | 1.22E+17 | 2.384455 | -1 | -7.062902E-01 | 2.485168E-04 | 284302.2 | 100.0352 |
| 23 | 20 | 2.43E+18 | 0.989549 | 1 | 2.832587E-01 | 2.485168E-04 | 113879.7 | 99.91227 |
| 24 | 21 | 5.11E+19 | 0.391107 | -1 | -1.078487E-01 | 2.485168E-04 | 43496.95 | 100.2304 |
| 25 | 22 | 1.12E+21 | 0.147554 | 1 | 3.970542E-02 | 2.485168E-04 | 15876.96 | 99.3741 |
| 26 | 23 | 2.59E+22 | 0.053248 | -1 | -1.354238E-02 | 2.485168E-04 | 5549.281 | 101.8351 |
| 27 | 24 | 6.2E+23 | 0.018415 | 1 | 4.872486E-03 | 2.485168E-04 | 1860.626 | 94.89959 |
| 28 | 25 | 1.55E+25 | 0.006114 | -1 | -1.241249E-03 | 2.485168E-04 | 599.4628 | 120.0215 |

The results are oscillating. If carried out further to n = 39, the series will eventually converge to within 7 significant digits.

In contrast the second series converges faster. It attains 7 significant digits at n = 28.

| | A | B | C | D | E | F | G | H |
|---|---|---|---|---|---|---|---|---|
| 1 | x | 8.3 | | | | | | |
| 2 | n | n! | x^n/n! | Series | 1/Series | True Value | et (%) | ea(%) |
| 3 | 0 | 1 | 1 | 1.000000E+00 | 1.000000E+00 | 2.485168E-04 | 4.02E+05 | |
| 4 | 1 | 1 | 8.3 | 9.300000E+00 | 1.075269E-01 | 2.485168E-04 | 4.32E+04 | 9.98E+01 |
| 5 | 2 | 2 | 34.445 | 4.374500E+01 | 2.285976E-02 | 2.485168E-04 | 9.10E+03 | 9.89E+01 |
| 6 | 3 | 6 | 95.29783 | 1.390428E+02 | 7.192028E-03 | 2.485168E-04 | 2.79E+03 | 9.65E+01 |
| 7 | 4 | 24 | 197.743 | 3.367858E+02 | 2.969246E-03 | 2.485168E-04 | 1.09E+03 | 9.16E+01 |
| 8 | 5 | 120 | 328.2534 | 6.650392E+02 | 1.503671E-03 | 2.485168E-04 | 5.05E+02 | 8.35E+01 |
| 9 | 6 | 720 | 454.0839 | 1.119123E+03 | 8.935568E-04 | 2.485168E-04 | 2.60E+02 | 7.22E+01 |
| 10 | 7 | 5040 | 538.4137 | 1.657537E+03 | 6.033049E-04 | 2.485168E-04 | 1.43E+02 | 5.88E+01 |
| 11 | 8 | 40320 | 558.6042 | 2.216141E+03 | 4.512348E-04 | 2.485168E-04 | 8.16E+01 | 4.49E+01 |
| 12 | 9 | 362880 | 515.1572 | 2.731298E+03 | 3.661263E-04 | 2.485168E-04 | 4.73E+01 | 3.21E+01 |
| 13 | 10 | 3628800 | 427.5805 | 3.158879E+03 | 3.165680E-04 | 2.485168E-04 | 2.74E+01 | 2.15E+01 |
| 14 | 11 | 39916800 | 322.6289 | 3.481508E+03 | 2.872319E-04 | 2.485168E-04 | 1.56E+01 | 1.35E+01 |
| 15 | 12 | 4.79E+08 | 223.1517 | 3.704659E+03 | 2.699304E-04 | 2.485168E-04 | 8.62E+00 | 7.93E+00 |
| 16 | 13 | 6.23E+09 | 142.4738 | 3.847133E+03 | 2.599338E-04 | 2.485168E-04 | 4.59E+00 | 4.39E+00 |
| 17 | 14 | 8.72E+10 | 84.46659 | 3.931600E+03 | 2.543494E-04 | 2.485168E-04 | 2.35E+00 | 2.29E+00 |
| 18 | 15 | 1.31E+12 | 46.73818 | 3.978338E+03 | 2.513613E-04 | 2.485168E-04 | 1.14E+00 | 1.13E+00 |
| 19 | 16 | 2.09E+13 | 24.24543 | 4.002583E+03 | 2.498386E-04 | 2.485168E-04 | 5.32E-01 | 5.29E-01 |
| 20 | 17 | 3.56E+14 | 11.83747 | 4.014421E+03 | 2.491019E-04 | 2.485168E-04 | 2.35E-01 | 2.35E-01 |
| 21 | 18 | 6.4E+15 | 5.458391 | 4.019879E+03 | 2.487637E-04 | 2.485168E-04 | 9.93E-02 | 9.92E-02 |
| 22 | 19 | 1.22E+17 | 2.384455 | 4.022264E+03 | 2.486162E-04 | 2.485168E-04 | 4.00E-02 | 4.00E-02 |
| 23 | 20 | 2.43E+18 | 0.989549 | 4.023253E+03 | 2.485551E-04 | 2.485168E-04 | 1.54E-02 | 1.54E-02 |
| 24 | 21 | 5.11E+19 | 0.391107 | 4.023644E+03 | 2.485309E-04 | 2.485168E-04 | 5.67E-03 | 5.67E-03 |
| 25 | 22 | 1.12E+21 | 0.147554 | 4.023792E+03 | 2.485218E-04 | 2.485168E-04 | 2.00E-03 | 2.00E-03 |
| 26 | 23 | 2.59E+22 | 0.053248 | 4.023845E+03 | 2.485185E-04 | 2.485168E-04 | 6.79E-04 | 6.79E-04 |
| 27 | 24 | 6.2E+23 | 0.018415 | 4.023863E+03 | 2.485174E-04 | 2.485168E-04 | 2.22E-04 | 2.22E-04 |
| 28 | 25 | 1.55E+25 | 0.006114 | 4.023870E+03 | 2.485170E-04 | 2.485168E-04 | 6.96E-05 | 6.96E-05 |

3.7 $f'(1.22) = \dfrac{4(1.22)}{(3-2\times 1.22^2)^2}$

using 3 digit calculations,

$x = 1.22$
$x^2 = 1.48$
$2x^2 = 2.96$

$f' = 3050$

using 4 digit calculations,

$2x^2 = 2.976$

$f' = 8471$

True value is 9066.587

Calculations have large error and are sensitive to the precision of the arithmetic

3.8 a) using 3 digit arithmetic with chopping gives

$x^3 = 20.3$
$-5x^2 = -37.2$
$6x = 16.3$
$.55 = .55$

SUM $= -.05$

TRUE value $= .011917$

$\epsilon_t = 519\%$

b)

$$\begin{array}{r} 2.73 \\ -5.00 \\ \hline -2.27 \\ \times 2.73 \\ \hline -6.19 \\ +6.00 \\ \hline -0.19 \\ \times 2.73 \\ \hline -0.518 \\ +.55 \\ \hline +.032 \end{array}$$

$\epsilon_t = -168\%$

error is significantly reduced in form (b)

3.9 Solution:

21 x 21 x 120 = 52920 words @ 64 bits/word = 8 bytes/word
52920 words @ 8 bytes/word = 423360 bytes
423360 bytes / 1024 bytes/kilobyte = 413.4 kilobytes = 0.41 M bytes

3.10 Solution:

```
%  Given: Taylor Series Approximation for cos(x) = 1 - x^2/2! + x^4/4! - ...
%  Find: number of terms needed to represent cos(x) to 8 significant
%  figures at the point where: x=0.2 pi

x=0.2*pi;
es=0.5e-08;

%approximation
cos=1;
j=1;
% j=terms counter
fprintf('j= %2.0f   cos(x)= %0.10f\n', j,cos)
fact=1;
for i=2:2:100
  j=j+1;
  fact=fact*i*(i-1);
  cosn=cos+((-1)^(j+1))*((x)^i)/fact;
  ea=abs((cosn-cos)/cosn);
  if ea<es
    fprintf('j= %2.0f   cos(x)= %0.10f   ea = %0.1e   CONVERGENCE
                es= %0.1e',j,cosn,ea,es)
       break
  end
    fprintf( 'j= %2.0f   cos(x)= %0.10f      ea = %0.1e\n',j,cosn,ea )
    cos=cosn;
end


j=  1   cos(x)= 1.0000000000
j=  2   cos(x)= 0.8026079120      ea = 2.5e-001
j=  3   cos(x)= 0.8091018514      ea = 8.0e-003
j=  4   cos(x)= 0.8090163946      ea = 1.1e-004
j=  5   cos(x)= 0.8090169970      ea = 7.4e-007
j=  6   cos(x)= 0.8090169944      ea = 3.3e-009   CONVERGENCE es = 5.0e-009»
```

Chapter 4

4.1 a) For this case $x_i = 0$ and $h = x$, thus

$$f(x) = f(0) + f'(0)x + f''(0)\frac{x^2}{2} + \cdots$$

$$f(0) = f'(0) = f''(0) = 1$$

$$\therefore\ f(x) = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \cdots$$

b) $$f(x_{i+1}) = e^{-x_i} - e^{-x_i}h + e^{-x_i}\frac{h^2}{2} - e^{-x_i}\frac{h^3}{6} + \cdots$$

for $x_i = 0.25$ and $x_{i+1} = 1$,

$h = 0.75$

zero order:

$$f(1) \simeq e^{-0.25} = 0.778801$$

true value $= e^{-1.0} = 0.367879$

$$\epsilon_t = \frac{0.367879 - 0.778801}{0.367879} \times 100$$

$$\epsilon_t = -111.7\ \%$$

first order:

$$f(1) \simeq 0.778801 - 0.778801(.75)$$
$$\simeq 0.1947$$

$$\epsilon_t = 47.1\%$$

second order:

$$f(1) = 0.778801 - 0.778801(.75) + 0.77801\frac{(.75)^2}{2}$$

$$f(1) \simeq 0.413738$$

$$\epsilon_t = -12.46\ \%$$

third order:

$$f(1) = 0.413738 - 0.77801\frac{(.75)^3}{6}$$

$$f(1) \simeq 0.358978$$

$$\epsilon_t = 2.42\ \%$$

4.2 Use $\epsilon_s = 0.5 \times 10^{2-2} = 0.5\%$

zero order:

$$\cos(\pi/4) \simeq 1$$

true value $\cos(\pi/4) = 0.707107$

$$\epsilon_t = \frac{0.707107 - 1}{0.707107} \times 100 = -41.42\%$$

first order:

$$\cos(\pi/4) = 1 - \frac{(\pi/4)^2}{2}$$
$$= 0.691575$$
$$(\epsilon_t = 2.19\ \%)$$

$$\epsilon_a = \frac{0.691575 - 1}{0.691575} \times 100 = -44.6\ \%$$

second order:

$$\cos(\pi/4) \approx 0.691575 + \frac{(\pi/4)^4}{24}$$

$$\approx 0.707429$$
$$(\epsilon_t = -0.456\%)$$

$$\epsilon_a = 2.24\%$$

third order:

$$\cos(\pi/4) \approx 0.707429 - \frac{\left(\frac{\pi}{4}\right)^6}{720}$$

$$\approx 0.707130$$
$$(\epsilon_t = 0.0005\%)$$

$$\epsilon_a = -0.046\%$$

4.3 $\epsilon_s = 0.5 \times 10^{2-2} = 0.5\%$

zero order:

$$\sin\left(\frac{\pi}{4}\right) = \frac{\pi}{4} = 0.785398$$

True value $= 0.707107$

$$\epsilon_t = \frac{0.707107 - 0.785398}{0.707107} \times 100$$

$$= -11.1\%$$

first order:

$$\sin\left(\frac{\pi}{4}\right) \approx 0.785398 - \frac{\left(\frac{\pi}{4}\right)^3}{6}$$

$$= 0.704653$$
$$(\epsilon_t = 0.347\%)$$

$$\epsilon_a = \frac{(0.704653 - 0.785398)}{0.704653} \times 100$$

$$\approx -11.46\%$$

second order:

$$\sin(\pi/4) \approx 0.704653 + \frac{\left(\frac{\pi}{4}\right)^5}{60}$$

$$\approx 0.709633$$
$$(\epsilon_t = -0.57\%)$$

$$\epsilon_a = .70\%$$

third order:

$$\sin\left(\frac{\pi}{4}\right) \approx 0.709633 - \frac{\left(\frac{\pi}{4}\right)^7}{7!}$$

$$\approx 0.709597$$
$$(\epsilon_t \approx -0.35\%)$$

$$\epsilon_a \approx -0.0051\%$$

4.4 true value $f(2) = 102$

zero order:

$$f(2) \approx f(1) \approx -62$$
$$(\epsilon_t = 160.8\%)$$

first order:

$f'(1) = 75(1)^2 - 12(1) + 7 = 70$

$f(2) \approx -62 + 70(1) = 8$

$\epsilon_t = 92.1\%$

second order:

$f''(1) = 150(1) - 12 = 138$

$f(2) \approx 8 + \frac{138}{2}(1)^2 = 77$

$\epsilon_t = 24.5\%$

third order:

$f'''(1) = 150$

$f(2) \approx 77 + \frac{150}{6}(1)^3$

$\approx 102$

$\epsilon_t = 0$

as expected

4.5

true $= \ln(3) = 1.098612$

zero order:

$f(3) = f(1) = 0$

$\epsilon_t = 100\%$

first order:

$f(3) = 0 + \frac{1}{1}(2) = 2$

$\epsilon_t = -82.05\%$

second order:

$f(3) = 2 - \frac{1}{1^2}\frac{(2)^2}{2}$

$= 0$

$\epsilon_t = 100\%$

third order

$f(3) = 0 + \frac{2}{1^3}\frac{2^3}{6}$

$= 2.66666$

$(\epsilon_t = -142.7\%)$

fourth order

$f(4) = 2.6666 - \frac{6}{(1)^4}\frac{(2)^4}{24}$

$\approx -1.3333$

$\epsilon_t = 221.36\%$

diverges must use smaller step for series to converge

4.6 $f'(x) = 75x^2 - 12x + 7$

$f'(2) = 283$ true

$x_{i-1} = 1.75 \quad f(x_{i-1}) = 39.85938$
$x_i = 2.0 \quad f(x_i) = 102$
$x_{i+1} = 2.25 \quad f(x_{i+1}) = 182.1406$

forward

$$f'(2) = \frac{182.1406 - 102}{.25}$$

$$= 320.5625$$

$$\epsilon_t = -13.273$$

backward

$$f'(2) = \frac{102 - 39.85938}{.25}$$

$$= 248.5625$$

$$\epsilon_t = 12.17\%$$

central

$$f'(2) \approx \frac{182.1406 - 39.85938}{2(.25)}$$

$$= 284.5625$$

$$\epsilon_t = -0.55\%$$

Both forward and backward have errors approximately

$$|E_t| \approx \frac{f''(x_i)}{2} h$$

$$f''(2) = 150(2) - 12 = 288$$

$\therefore$

$$|E_t| \approx \frac{288}{2}(.25) = 36$$

which is close

For central Difference

$$|E_t| \approx -\frac{f'''(x_i)}{6} h^2$$

$$\approx -\frac{150}{6}(.25)^2 = -1.5625$$

which is exactly

$$E_t = 283 - 284.5625$$

as expected

4.7 true value

$$f''(2) = 288$$

$$f''(2) \approx \frac{164.56 - 2(102) + 50.92}{(.2)^2}$$

$$= 288$$

$$f''(2) = \frac{131.765 - 2(102) + 75.115}{(.1)^2}$$

$$= 288$$

both are exact because errors are function of 4th order derivatives which are zeros for 3rd order polynomial

4.8 $$\frac{\partial v}{\partial c} = \frac{cgt e^{-c/m\,t} - gm(1 - e^{-c/m\,t})}{c^2}$$

$$= -1.38666$$

$$\Delta v(\tilde{c}) = \left|\frac{\partial v}{\partial c}\right| \Delta\tilde{c}$$

$$= 1.38666\,(2)$$
$$= 2.77332$$

$$v(12.5) = \frac{9.8(50)}{12.5}\left(1 - e^{-\frac{12.5(6)}{50}}\right)$$
$$= 30.4533$$

$\therefore\ v = 30.4533 \pm 2.77332$

4.9 $$\Delta v(\tilde{c}, \tilde{m}) = \underbrace{\left|\frac{\partial c}{\partial v}\right| \Delta\tilde{c}}_{2.77332} + \left|\frac{\partial v}{\partial m}\right| \Delta\tilde{m}$$

$$\frac{\partial v}{\partial m} = \frac{gt}{m} e^{-c/m\,t} + \frac{g}{c}(1 - e^{-c/m\,t})$$

$$= 0.871467$$

$$\left|\frac{\partial v}{\partial m}\right| \Delta m = 0.871467\,(0.5)$$

$$= 0.435734$$

$$\Delta v = 2.77332 + 0.435734$$
$$= 3.209053$$

$\therefore\ v = 30.4533 \pm 3.209053$

4.10 $$\Delta H(\tilde{T}) = \left|\frac{\partial H}{\partial T}\right| \Delta\tilde{T}$$

$$\frac{\partial H}{\partial T} = 4Ae\sigma T^3$$
$$= 4(.15)(.9)(5.67\times10^{-8})(650)^3$$
$$= 8.41$$

$$\Delta H \approx 8.41(25) = 210.2$$

Exact Error

$$\Delta H_{\text{true}} = \frac{H(675) - H(625)}{2}$$
$$= \frac{1589 - 1167}{2}$$
$$= 211 \text{ close to } 210.2$$

For $\Delta T = 50$

$$\Delta H \approx 8.41(50) = 420.4$$

$$\Delta H_{\text{true}} = \frac{H(700) - H(600)}{2}$$
$$= \frac{1837 - 992}{2}$$
$$= 422.5 \text{ close to } 420.4$$

Results are good because H(T) is nearly linear over range of $\Delta T$

4.11 For sphere $A = 4\pi r^2$

$$H = 4\pi r^2 e \sigma T^4$$

$$\Delta H = \left|\frac{\partial H}{\partial r}\right| \Delta \tilde{r} + \left|\frac{\partial H}{\partial e}\right| \Delta e + \left|\frac{\partial H}{\partial T}\right| \Delta T$$

$$\frac{\partial H}{\partial r} = 8\pi r e \sigma T^4$$
$$= 17604$$

$$\frac{\partial H}{\partial e} = 4\pi r^2 \sigma T^4$$
$$= 1467$$

$$\frac{\partial H}{\partial T} = 16\pi r^2 e \sigma T^3$$
$$= 9.6$$

$$\Delta H = 17604(.02) + 1467(.05) + 9.6(25)$$

$$\Delta H \approx 665.4$$

$$H(.17, .95, 575) = 2138.4$$
$$H(.13, .85, 525) = 777.6$$

$$\Delta H_{TRUE} = \frac{2138.4 - 777.6}{2}$$

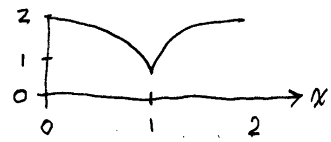
$$= 680.4$$

4.12 $$CN = \frac{\tilde{x} f'(\tilde{x})}{f(\tilde{x})}$$

a) $$CN = \frac{1.0001\left[\frac{1}{2\sqrt{1.0001-1}}\right]}{\sqrt{1.0001-1} + 1}$$

$$= 50.00$$

ill conditioned because $f'(1)$ is large near $x = 1$




b) $$CN = \frac{9(-e^{-9})}{e^{-9}} = -9$$

ill conditions because $x$ is large

c) $$CN = \frac{(200)\left(\frac{200}{\sqrt{200^2+1}} - 1\right)}{\sqrt{200^2+1} - 200}$$

$$\approx \frac{200(-1.2\times 10^{-5})}{0.0025}$$

$$\approx -0.96$$ well conditioned

d) $$CN = \frac{x\left(\frac{-xe^{-x} - e^{-x} + 1}{x^2}\right)}{\left(\frac{e^{-x} - 1}{x}\right)}$$

$$= \frac{.01\left(\frac{-.01(.99) - (.99) + 1}{(.01)^2}\right)}{\left(\frac{.99 - 1}{.01}\right)}$$

$$= \frac{.9}{-1} = -.9$$ well conditioned

e) $f(x) = \frac{\sin(x)}{1+\cos(x)}$

$$f'(x) = \frac{(1+\cos x)(\cos(x)) + \sin x (\sin x)}{(1+\cos x)^2}$$

$$CN = \frac{1.001\pi \; f'(1.001\pi)}{f(1.001\pi)}$$

$$CN = \frac{3.144\,(202642)}{-636.6}$$

$$= -1001$$

ill conditioned because

$1 + \cos(1.001\pi) \approx 0$

**division**

$$f(u,v) = u/v$$

$$\frac{\partial f}{\partial u} = \frac{1}{v}$$

$$\frac{\partial f}{\partial v} = -u/v^2$$

$$\Delta f(u,v) = \left|\frac{1}{v}\right| \Delta u + \left|\frac{u}{v^2}\right| \Delta v$$

$$= \frac{|v|\,\Delta u + |u|\,\Delta v}{|v^2|}$$

4.13 addition and subtraction

$$f(u,v) = u+v$$

$$\Delta F = \left|\frac{\partial f}{\partial u}\right| \Delta u + \left|\frac{\partial f}{\partial v}\right| \Delta \hat{v}$$

$$\left|\frac{\partial f}{\partial u}\right| = 1 \qquad \left|\frac{\partial f}{\partial v}\right| = 1$$

$$\Delta f(\tilde{u},\tilde{v}) = \Delta\tilde{u} + \Delta\tilde{v}$$

**multiplication**

$$f(u,v) = u \cdot v$$

$$\left|\frac{\partial f}{\partial u}\right| = v \qquad \left|\frac{\partial f}{\partial v}\right| = u$$

$$\Delta f(\tilde{u},\tilde{v}) = |\tilde{v}|\,\Delta\hat{u} + |\tilde{u}|\,\Delta\tilde{v}$$

4.14 $f(x) = ax^2 + bx + c$

$f'(x) = 2ax + b$

$f''(x) = 2a$

$$a x_{i+1}^2 + b x_{i+1} + c =$$

$$a x_i^2 + b x_i + c + 2a x_i + b\,(x_{i+1} - x_i)$$

$$+ \frac{2a}{2}\left(x_{i+1}^2 - 2x_{i+1}x_i + x_i^2\right)$$

collect terms

$$a\left(x_i^2 + 2x_i x_{i+1} - 2x_i^2 + x_{i+1}^2\right.$$

$$\left. - 2x_{i+1}x_i + x_i^2\right) =$$

$$a x_{i+1}^2$$

etc

4.15

$$\Delta Q = \left|\frac{\partial Q}{\partial n}\right| \Delta n + \left|\frac{\partial Q}{\partial s}\right| \Delta s$$

$$\frac{\partial Q}{\partial n} = -\frac{1}{n^2} \frac{(BH)^{5/3}}{(B+2H)^{2/3}} s^{\frac{1}{2}}$$

$$\frac{\partial Q}{\partial s} = \frac{1}{n} \frac{(BH)^{5/3}}{(B+2H)^{2/3}} \frac{1}{2s^{1/2}}$$

$$\Delta Q = \underbrace{|-5.07|(.003)}_{0.15} + \underbrace{|2536|(.00003)}_{0.076}$$

∴ error from roughness measurement is about 2 times the error caused by uncertainty in slope, thus improve precision of roughness is best strategy

4.16 use $\epsilon_s = 0.5 \times 10^{2-2} = 0.5\%$

zero order

$$\frac{1}{1-.1} = 1.1111 \approx 1$$

$$\epsilon_t = \left(\frac{1.1111 - 1}{1.1111}\right) \times 100 = 9.99\%$$

first order

$$1.1111 \approx 1 + 0.1 = 1.1$$

$$\epsilon_t = .99\%$$

$$\epsilon_a = \frac{1.1 - 1}{1.1} \times 100 = 9.1$$

second order

$$1.1111 \approx 1 + .1 + .01 = 1.11$$

$$\epsilon_t = 0.099\%$$

$$\epsilon_a = \frac{1.11 - 1.1}{1.11} \times 100$$

$$= 0.9\%$$

third order

$$1.11111 \approx 1 + .1 + .01 + .001$$

$$\approx 1.111$$

$$\epsilon_t = 0.009\%$$

$$\epsilon_a = \frac{1.111 - 1.11}{1.111} \times 100$$

$$= 0.09\% < 0.5\%$$

4.17 $\Delta(\sin\phi_0) =$

$$\left|\frac{\partial}{\partial \alpha}\left[(1+\alpha)\sqrt{1 - \frac{\alpha}{1+\alpha}\left(\frac{v_e}{v_0}\right)^2}\right]\right| \Delta\alpha$$

$$= \left|\left\{\frac{1+\alpha}{2}\left(1 - \frac{\beta\alpha}{1+\alpha}\right)^{-\frac{1}{2}}\left(\frac{\beta\alpha}{(1+\alpha)^2} - \frac{\beta}{1+\alpha}\right) + \left(1 - \frac{\beta\alpha}{1+\alpha}\right)^{1/2}\right\}\right| \Delta\alpha$$

where $\beta = \left(\frac{v_e}{v_0}\right)^2 = 4$ and $\alpha = 0.2$

$$\Delta(\sin\phi_0) = 2.3\ \Delta\tilde{\alpha}$$

for $\Delta n = 0.2\,(.01) = 0.002$

$\Delta(\sin\phi_o) = 0.0046$

$\sin\phi_o = (1+.2)\sqrt{1 - \frac{.2}{1.2}(4)}$

$= .6928$

Therefore

$\max \sin\phi_o = .6928 + .0046$
$= 0.69742$

$\min \sin\phi_o = .6928 - .0046$
$= 0.68822$

$\max \phi_o = \frac{0.771792}{2\pi} \times 360$
$= 44.22^\circ$

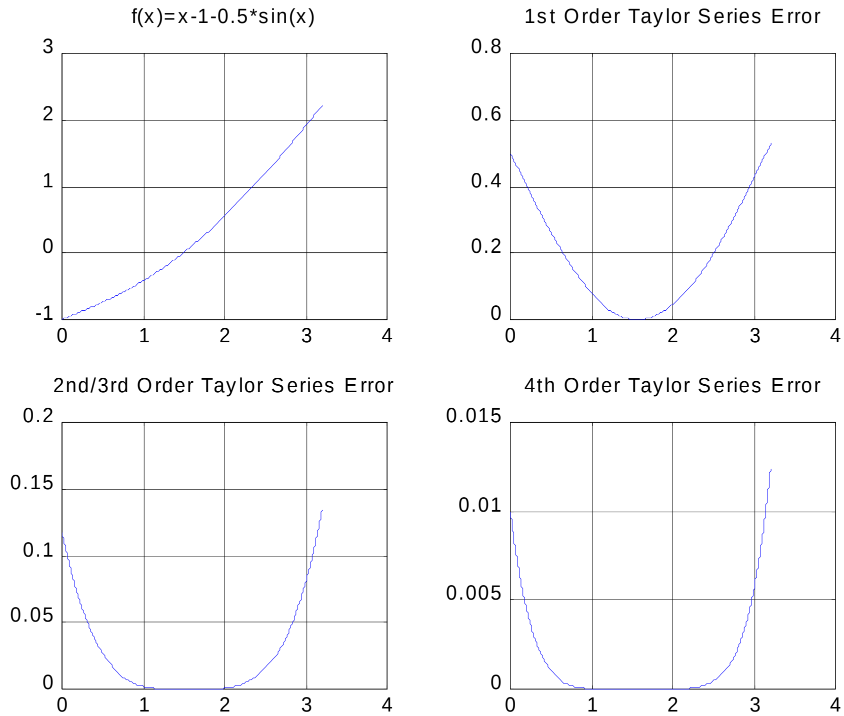
$\min \phi_o = 43.49^\circ$

4.18 f(x) = x-1-1/2*sin(x)
f '(x) = 1-1/2*cos(x)
f ''(x) = 1/2*sin(x)
f '''(x) = 1/2*cos(x)
$f^{IV}$(x) = -1/2*sin(x)

Using the Taylor Series Expansion (Equation 4.5 in the book), we obtain the following 1st, 2nd, 3rd, and 4th Order Taylor Series functions shown below in the Matlab program-f1, f2, f4. Note the 2nd and 3rd Order Taylor Series functions are the same.

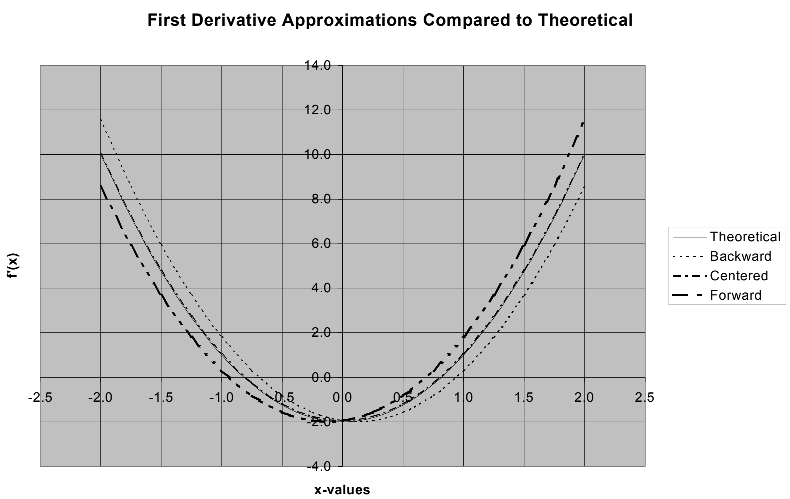
From the plots below, we see that the answer is the <u>4th Order Taylor Series expansion</u>.

```
x=0:0.001:3.2;
f=x-1-0.5*sin(x);
subplot(2,2,1);
plot(x,f);grid;title('f(x)=x-1-0.5*sin(x)');hold on

f1=x-1.5;
e1=abs(f-f1);      %Calculates the absolute value of the difference/error
subplot(2,2,2);
plot(x,e1);grid;title('1st Order Taylor Series Error');

f2=x-1.5+0.25.*((x-0.5*pi).^2);
e2=abs(f-f2);
subplot(2,2,3);
plot(x,e2);grid;title('2nd/3rd Order Taylor Series Error');

f4=x-1.5+0.25.*((x-0.5*pi).^2)-(1/48)*((x-0.5*pi).^4);
e4=abs(f4-f);
subplot(2,2,4);
plot(x,e4);grid;title('4th Order Taylor Series Error');hold off
```

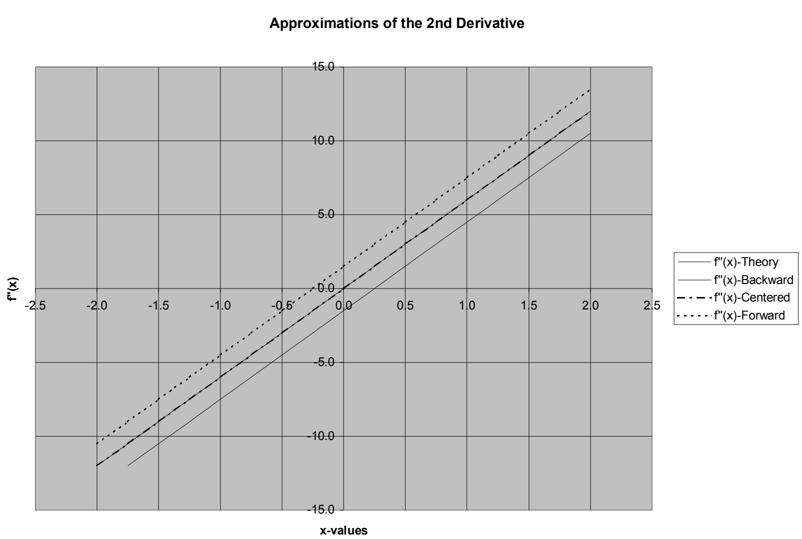



## 4.19 EXCEL WORKSHEET AND PLOTS

| x | f(x) | f(x-1) | f(x+1) | f'(x)-Theory | f'(x)-Back | f'(x)-Cent | f'(x)-Forw |
|---|---|---|---|---|---|---|---|
| -2.000 | 0.000 | -2.891 | 2.141 | 10.000 | 11.563 | 10.063 | 8.563 |
| -1.750 | 2.141 | 0.000 | 3.625 | 7.188 | 8.563 | 7.250 | 5.938 |
| -1.500 | 3.625 | 2.141 | 4.547 | 4.750 | 5.938 | 4.813 | 3.688 |
| -1.250 | 4.547 | 3.625 | 5.000 | 2.688 | 3.688 | 2.750 | 1.813 |
| -1.000 | 5.000 | 4.547 | 5.078 | 1.000 | 1.813 | 1.063 | 0.313 |
| -0.750 | 5.078 | 5.000 | 4.875 | -0.313 | 0.313 | -0.250 | -0.813 |
| -0.500 | 4.875 | 5.078 | 4.484 | -1.250 | -0.813 | -1.188 | -1.563 |
| -0.250 | 4.484 | 4.875 | 4.000 | -1.813 | -1.563 | -1.750 | -1.938 |
| 0.000 | 4.000 | 4.484 | 3.516 | -2.000 | -1.938 | -1.938 | -1.938 |
| 0.250 | 3.516 | 4.000 | 3.125 | -1.813 | -1.938 | -1.750 | -1.563 |
| 0.500 | 3.125 | 3.516 | 2.922 | -1.250 | -1.563 | -1.188 | -0.813 |
| 0.750 | 2.922 | 3.125 | 3.000 | -0.313 | -0.813 | -0.250 | 0.313 |
| 1.000 | 3.000 | 2.922 | 3.453 | 1.000 | 0.313 | 1.063 | 1.813 |
| 1.250 | 3.453 | 3.000 | 4.375 | 2.688 | 1.813 | 2.750 | 3.688 |
| 1.500 | 4.375 | 3.453 | 5.859 | 4.750 | 3.688 | 4.813 | 5.938 |
| 1.750 | 5.859 | 4.375 | 8.000 | 7.188 | 5.938 | 7.250 | 8.563 |
| 2.000 | 8.000 | 5.859 | 10.891 | 10.000 | 8.563 | 10.063 | 11.563 |



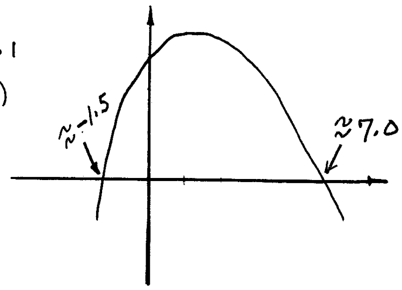

| x | f(x) | f(x-1) | f(x+1) | f(x-2) | f(x+2) | f''(x)-Theory | f''(x)-Back | f''(x)-Cent | f''(x)-Forw |
|---|---|---|---|---|---|---|---|---|---|
| -2.000 | 0.000 | -2.891 | 2.141 | 3.625 | 3.625 | -12.000 | 150.500 | -12.000 | -10.500 |
| -1.750 | 2.141 | 0.000 | 3.625 | -2.891 | 4.547 | -10.500 | -12.000 | -10.500 | -9.000 |
| -1.500 | 3.625 | 2.141 | 4.547 | 0.000 | 5.000 | -9.000 | -10.500 | -9.000 | -7.500 |
| -1.250 | 4.547 | 3.625 | 5.000 | 2.141 | 5.078 | -7.500 | -9.000 | -7.500 | -6.000 |
| -1.000 | 5.000 | 4.547 | 5.078 | 3.625 | 4.875 | -6.000 | -7.500 | -6.000 | -4.500 |
| -0.750 | 5.078 | 5.000 | 4.875 | 4.547 | 4.484 | -4.500 | -6.000 | -4.500 | -3.000 |
| -0.500 | 4.875 | 5.078 | 4.484 | 5.000 | 4.000 | -3.000 | -4.500 | -3.000 | -1.500 |
| -0.250 | 4.484 | 4.875 | 4.000 | 5.078 | 3.516 | -1.500 | -3.000 | -1.500 | 0.000 |
| 0.000 | 4.000 | 4.484 | 3.516 | 4.875 | 3.125 | 0.000 | -1.500 | 0.000 | 1.500 |
| 0.250 | 3.516 | 4.000 | 3.125 | 4.484 | 2.922 | 1.500 | 0.000 | 1.500 | 3.000 |
| 0.500 | 3.125 | 3.516 | 2.922 | 4.000 | 3.000 | 3.000 | 1.500 | 3.000 | 4.500 |
| 0.750 | 2.922 | 3.125 | 3.000 | 3.516 | 3.453 | 4.500 | 3.000 | 4.500 | 6.000 |
| 1.000 | 3.000 | 2.922 | 3.453 | 3.125 | 4.375 | 6.000 | 4.500 | 6.000 | 7.500 |
| 1.250 | 3.453 | 3.000 | 4.375 | 2.922 | 5.859 | 7.500 | 6.000 | 7.500 | 9.000 |
| 1.500 | 4.375 | 3.453 | 5.859 | 3.000 | 8.000 | 9.000 | 7.500 | 9.000 | 10.500 |
| 1.750 | 5.859 | 4.375 | 8.000 | 3.453 | 10.891 | 10.500 | 9.000 | 10.500 | 12.000 |
| 2.000 | 8.000 | 5.859 | 10.891 | 4.375 | 14.625 | 12.000 | 10.500 | 12.000 | 13.500 |



## Chapter 5

5.1

a)




b)

$$x = \frac{-2.2 \pm \sqrt{(2.2)^2 + 4(.4)(4.7)}}{2(0.4)}$$

$$x = 7.1446$$
$$= -1.6446$$

c) 1st iteration

$$x_r = \frac{10+5}{2} = 7.5$$

$$\epsilon_t = -4.97\%$$

$$\epsilon_a = \left|\frac{x_u - x_l}{x_u + x_l}\right| \times 100$$
$$= \left|\frac{5}{15}\right| \times 100 = 33.3\%$$

2nd iteration

$$x_r = \frac{7.5+5}{2} = 6.25$$

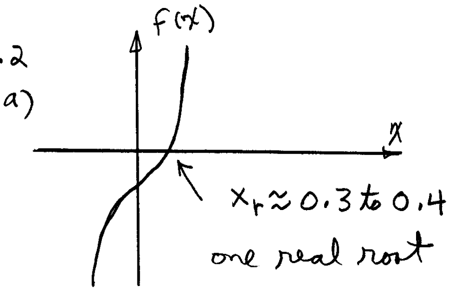
$$\epsilon_t = 12.5\%$$
$$\epsilon_a = 20\%$$

3rd iteration

$$x_r = 6.875$$
$$\epsilon_t = 3.77\%$$
$$\epsilon_a = 9.1\%$$

5.2

a)

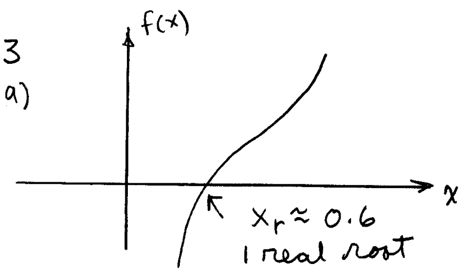



b)

| iteration | $x_r$ | $\epsilon_a$ |
|---|---|---|
| 1 | 0.5 | 100% |
| 2 | 0.25 | 100% |
| 3 | 0.375 | 33% |
| 4 | 0.3125 | 20% |
| 5 | 0.34375 | 9.1% |

5.3

a)




b)

| iteration | $x_r$ | $\epsilon_a$ |
|---|---|---|
| 1 | 0.75 | 33% |
| 2 | 0.625 | 20% |
| 3 | 0.5625 | 11.1% |
| 4 | 0.59375 | 5.2% |

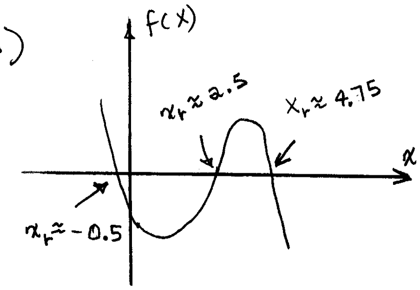
c) false position

$$x_r = x_u - \frac{f(x_u)(x_l - x_u)}{f(x_l) - f(x_u)}$$

$$\epsilon_a = \left|\frac{x_r^{new} - x_r^{old}}{x_r^{new}}\right| \times 100\%$$

| iteration | $x_r$ | $\epsilon_a$ |
|---|---|---|
| 1 | 0.62149 | — |
| 2 | 0.583727 | 6.47% |
| 3 | 0.5797806 | 0.68% |
| 4 | 0.5793734 | 0.07% |

5.4 a)



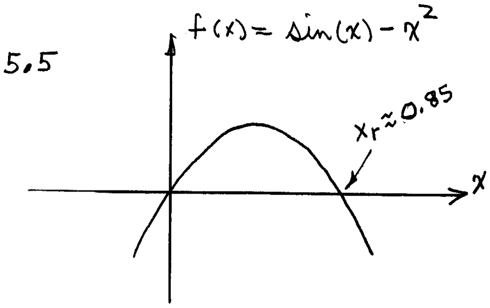

b) use $x_l = -1.0$
$x_u = 0$

$\epsilon_s = .5 \times 10^{2-3}$
$= 0.05\%$

use Eq 5.2 for $\epsilon_a$
Eq 5.7 for $x_r$

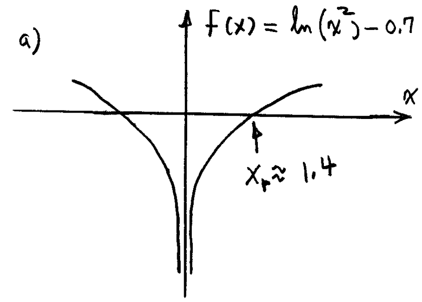
| iteration | $x_r$ | $\epsilon_a$ |
|---|---|---|
| 1 | −0.2650602 | 24% |
| 2 | −0.3489071 | 6.3% |
| 3 | −0.3725317 | 1.7% |
| 4 | −0.3789619 | .4% |
| 5 | −0.3806954 | .1% |
| 6 | −0.3811615 | .03% |
| 7 | −0.3812868 | .009% |

5.5




| iteration | $x_r$ | $\epsilon_a$ |
|---|---|---|
| 1 | 0.75 | 33.3% |
| 2 | 0.875 | 14.3 |
| 3 | 0.9375 | 6.6 |
| 4 | 0.90625 | 3.5 |
| 5 | 0.890625 | 1.7 |

5.6 a)




b) Bisection

| iteration | $x_r$ |
|---|---|
| 1 | 1.25 |
| 2 | 1.625 |
| 3 | 1.4375 |

c) False Position

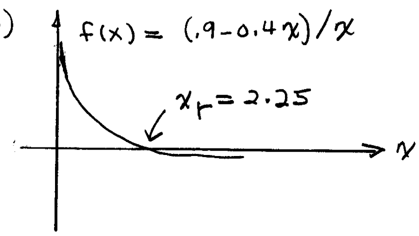
| iteration | $x_r$ |
|---|---|
| 1 | 1.628707 |
| 2 | 1.497014 |
| 3 | 1.448399 |

$x_r^{true} \approx 1.4191$

5.7

a) $x_r = \frac{0.9}{0.4} = 2.25$

b)




c)

| iter | $x_r$ | $\epsilon_a$ % | $\epsilon_t$ % |
|---|---|---|---|
| 1 | 2.6666 | — | −18.5 |
| 2 | 2.481481 | 7.46 | −10.3 |
| 3 | 2.378601 | 4.32 | −5.7 |

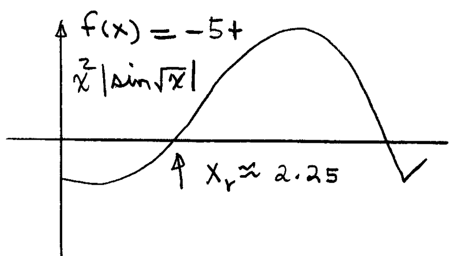
Note $|\epsilon_t| > \epsilon_a$

5.8

$$f(x) = x^2 - 15 = 0$$

| iteration | $x_r$ | $\epsilon_a$ |
|---|---|---|
| 1 | 3.857143 | — |
| 2 | 3.872727 | .4% |
| 3 | 3.872979 | .006% |

5.9

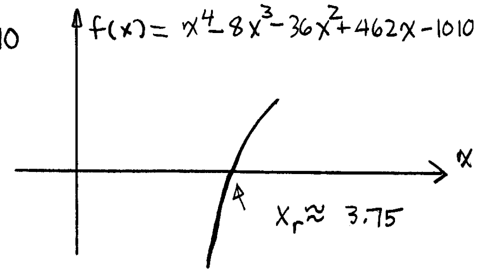



$\therefore$ use $x_l = 2$

$x_u = 3$

| iteration | $x_r$ | $\epsilon_a$ |
|---|---|---|
| 1 | 2.212672 | — |
| 2 | 2.236474 | 1.06% |
| 3 | 2.238897 | .11% |

5.10




$\therefore$ use $x_l = 3$

$x_u = 5$

| iteration | $x_r$ | $\epsilon_a$ % |
|---|---|---|
| 1 | 4.537037 | — |
| 2 | 4.219132 | 7.5 |
| 3 | 4.038489 | 4.5 |
| 4 | 3.950142 | 2.2 |
| 5 | 3.910475 | 1.0 |
| 6 | 3.893386 | .4 |

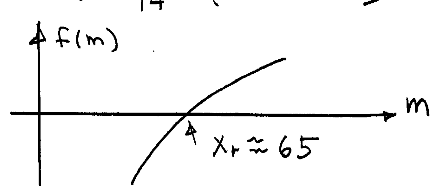
5.11 a) $x_r = 79^{\left(\frac{1}{3.3}\right)} = 3.758707$

b)

| iteration | $x_r$ | $\epsilon_a$ % |
|---|---|---|
| 1 | 3.697204 | — |
| 2 | 3.754321 | 1.5 |
| 3 | 3.758398 | .11 |
| 4 | 3.758686 | .007 |

5.12

$$f(m) = \frac{9.8\,m}{14}\left(1 - e^{-\frac{14}{m}7}\right) - 35$$




use $x_l = 60$ $\quad x_u = 70$

| iteration | $x_r$ | $\epsilon_a$ % |
|---|---|---|
| 1 | 63.85343 | — |
| 2 | 63.65618 | 0.3 |
| 3 | 63.6499 | 0.01 |

5.13 (*a*)

$$n = \frac{\log(35/.05)}{\log(2)} = 9.45 \text{ or } 10 \text{ iterations}$$

(*b*)

| iteration | $x_r$ |
|---|---|
| 1 | 17.5 |
| 2 | 26.25 |
| 3 | 30.625 |
| 4 | 28.4375 |
| 5 | 27.34375 |
| 6 | 26.79688 |
| 7 | 26.52344 |
| 8 | 26.66016 |
| 9 | 26.72852 |
| 10 | 26.76270 |

for $o_s$ = 8 mg/L, $T$ = 26.7627 °C
for $o_s$ = 10 mg/L, $T$ = 15.41504 °C
for $o_s$ = 14mg/L, $T$ = 1.538086 °C

5.14

Here is a VBA program to implement the Bisection function (Fig. 5.10) in a user-friendly program:

```
Option Explicit

Sub TestBisect()
Dim imax As Integer, iter As Integer
Dim x As Single, xl As Single, xu As Single
Dim es As Single, ea As Single, xr As Single
Dim root As Single

Sheets("Sheet1").Select
Range("b4").Select
xl = ActiveCell.Value
ActiveCell.Offset(1, 0).Select
xu = ActiveCell.Value
ActiveCell.Offset(1, 0).Select
es = ActiveCell.Value
ActiveCell.Offset(1, 0).Select
imax = ActiveCell.Value
Range("b4").Select

If f(xl) * f(xu) < 0 Then
  root = Bisect(xl, xu, es, imax, xr, iter, ea)
  MsgBox "The root is: " & root
  MsgBox "Iterations:" & iter
  MsgBox "Estimated error: " & ea
  MsgBox "f(xr) = " & f(xr)
Else
  MsgBox "No sign change between initial guesses"
End If

End Sub
```

```
Function Bisect(xl, xu, es, imax, xr, iter, ea)
Dim xrold As Single, test As Single
iter = 0
Do
  xrold = xr
  xr = (xl + xu) / 2
  iter = iter + 1
  If xr <> 0 Then
    ea = Abs((xr - xrold) / xr) * 100
  End If
  test = f(xl) * f(xr)
  If test < 0 Then
    xu = xr
  ElseIf test > 0 Then
    xl = xr
  Else
    ea = 0
  End If
  If ea < es Or iter >= imax Then Exit Do
Loop
Bisect = xr
End Function

Function f(c)
f = 9.8 * 68.1 / c * (1 - Exp(-(c / 68.1) * 10)) - 40
End Function
```

For Example 5.3, the Excel worksheet used for input looks like:

| | A | B | C | D | E |
|---|---|---|---|---|---|
| 1 | **Bisection Example** | | | | |
| 2 | | | | | |
| 3 | | | | | |
| 4 | **xl** | **12** | | Run | |
| 5 | **xu** | **16** | | | |
| 6 | **es** | **0.01** | | | |
| 7 | **imax** | **25** | | | |
| 8 | | | | | |

The program yields a root of 14.78027 after 12 iterations. The approximate error at this point is $6.63\times10^{-3}$ %. These results are all displayed as message boxes. For example, the solution check is displayed as


5.15 See solutions to Probs. 5.1 through 5.6 for results.

5.16 **Errata in Problem statement: Test the program by duplicating Example 5.5.**

Here is a VBA Sub procedure to implement the modified false position method. It is set up to evaluate Example 5.5.

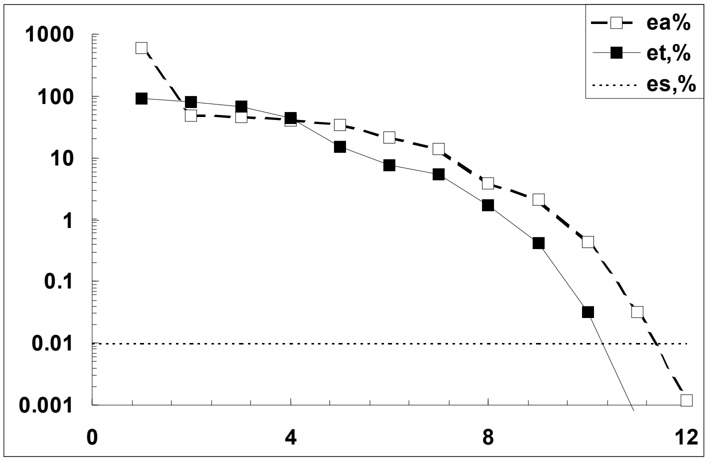
```
Option Explicit

Sub TestFP()
Dim imax As Integer, iter As Integer
Dim f As Single, FalseP As Single, x As Single, xl As Single
Dim xu As Single, es As Single, ea As Single, xr As Single

xl = 0
xu = 1.3
es = 0.01
imax = 20
MsgBox "The root is: " & FalsePos(xl, xu, es, imax, xr, iter, ea)
MsgBox "Iterations: " & iter
MsgBox "Estimated error: " & ea
End Sub

Function FalsePos(xl, xu, es, imax, xr, iter, ea)
Dim il As Integer, iu As Integer
Dim xrold As Single, fl As Single, fu As Single, fr As Single

iter = 0
fl = f(xl)
fu = f(xu)
Do
  xrold = xr
  xr = xu - fu * (xl - xu) / (fl - fu)
  fr = f(xr)
  iter = iter + 1
  If xr <> 0 Then
    ea = Abs((xr - xrold) / xr) * 100
  End If
  If fl * fr < 0 Then
    xu = xr
    fu = f(xu)
    iu = 0
    il = il + 1
    If il >= 2 Then fl = fl / 2
  ElseIf fl * fr > 0 Then
    xl = xr
    fl = f(xl)
    il = 0
    iu = iu + 1
    If iu >= 2 Then fu = fu / 2
  Else
    ea = 0#
  End If
  If ea < es Or iter >= imax Then Exit Do
Loop
FalsePos = xr
End Function

Function f(x)
f = x ^ 10 - 1
End Function
```

When the program is run for Example 5.5, it yields:

root = 14.7802
iterations = 5
error = $3.9 \times 10^{-5}$ %

5.17 **Errata in Problem statement: Use the subprogram you developed in Prob. 5.16 to duplicate the computation from Example 5.6.**

The results are plotted as




Interpretation: At first, the method manifests slow convergence. However, as it approaches the root, it approaches quadratic convergence. Note also that after the first few iterations, the approximate error estimate has the nice property that $\varepsilon_a > \varepsilon_t$.

5.18 Here is a VBA Sub procedure to implement the false position method with minimal function evaluations set up to evaluate Example 5.6.

```
Option Explicit
Sub TestFP()
Dim imax As Integer, iter As Integer, i As Integer
Dim xl As Single, xu As Single, es As Single, ea As Single, xr As
Single, fct As Single
MsgBox "The root is: " & FPMinFctEval(xl, xu, es, imax, xr, iter, ea)
MsgBox "Iterations: " & iter
MsgBox "Estimated error: " & ea
End Sub

Function FPMinFctEval(xl, xu, es, imax, xr, iter, ea)
Dim xrold, test, fl, fu, fr
iter = 0
xl = 0#
xu = 1.3
es = 0.01
imax = 50
fl = f(xl)
fu = f(xu)
xr = (xl + xu) / 2
Do
  xrold = xr
  xr = xu - fu * (xl - xu) / (fl - fu)
  fr = f(xr)
```

```
  iter = iter + 1
  If (xr <> 0) Then
    ea = Abs((xr - xrold) / xr) * 100#
  End If
  test = fl * fr
  If (test < 0) Then
    xu = xr
    fu = fr
  ElseIf (test > 0) Then
    xl = xr
    fl = fr
  Else
    ea = 0#
  End If
  If ea < es Or iter >= imax Then Exit Do
Loop
FPMinFctEval = xr
End Function

Function f(x)
f = x ^ 10 - 1
End Function
```

The program yields a root of 0.9996887 after 39 iterations. The approximate error at this point is $9.5\times10^{-3}$ %. These results are all displayed as message boxes. For example, the solution check is displayed as

The number of function evaluations for this version is 2n+2. This is much smaller than the number of function evaluations in the standard false position method (5n).

5.19 **Solve for the reactions:**

$R_1$=265 lbs.     $R_2$= 285 lbs.

**Write beam equations:**

0<x<3

$$M + (16.667x^2)\frac{x}{3} - 265x = 0$$

$$(1) \quad M = 265 - 5.55x^3$$

3<x<6

$$M + 100(x-3)(\frac{x-3}{2}) + 150(x - \frac{2}{3}(3)) - 265x = 0$$

$$(2) \quad M = -50x^2 + 415x - 150$$

6<x<10

$$M = 150(x - \frac{2}{3}(3)) + 300(x - 4.5) - 265x$$

$$(3) \quad M = -185x + 1650$$

10<x<12

$$M + 100(12 - x) = 0$$

$$(4) \quad M = 100x - 1200$$

**Combining Equations:**

Because the curve crosses the axis between 6 and 10, use (3).

(3) $M = -185x + 1650$

Set $x_L = 6; x_U = 10$

$M(x_L) = 540$
$M(x_U) = -200$ $\quad x_r = \dfrac{x_L + x_U}{2} = 8$
$M(x_R) = 170 \rightarrow$ *replaces* $x_L$

$M(x_L) = 170$
$M(x_U) = -200$ $\quad x_r = \dfrac{8+10}{2} = 9$
$M(x_R) = -15 \rightarrow$ *replaces* $x_U$

$M(x_L) = 170$
$M(x_U) = -15$ $\quad x_r = \dfrac{8+9}{2} = 8.5$
$M(x_R) = 77.5 \rightarrow$ *replaces* $x_L$

$M(x_L) = 77.5$
$M(x_U) = -15$ $\quad x_r = \dfrac{8.5+9}{2} = 8.75$
$M(x_R) = 31.25 \rightarrow$ *replaces* $x_L$

$M(x_L) = 31.25$
$M(x_U) = -15$ $\quad x_r = \dfrac{8.75+9}{2} = 8.875$
$M(x_R) = 8.125 \rightarrow$ *replaces* $x_L$

$M(x_L) = 8.125$
$M(x_U) = -15$ $\quad x_r = \dfrac{8.875+9}{2} = 8.9375$
$M(x_R) = -3.4375 \rightarrow$ *replaces* $x_U$

$M(x_L) = 8.125$
$M(x_U) = -3.4375$ $\quad x_r = \dfrac{8.875+8.9375}{2} = 8.90625$
$M(x_R) = 2.34375 \rightarrow$ *replaces* $x_L$

$M(x_L) = 2.34375$
$M(x_U) = -3.4375$ $\quad x_r = \dfrac{8.90625+8.9375}{2} = 8.921875$
$M(x_R) = -0.546875 \rightarrow$ *replaces* $x_U$

$M(x_L) = 2.34375$
$M(x_U) = -0.546875$ $\quad x_r = \dfrac{8.90625 + 8.921875}{2} = 8.9140625$

$M(x_R) = 0.8984$ **Therefore,** $x = 8.91\ feet$

5.20 $M = -185x + 1650$

Set $x_L = 6; x_U = 10$

$M(x_L) = 540$
$M(x_U) = -200$

$$x_R = x_o - \frac{M(x_U)(x_L - x_U)}{M(x_L) - M(x_U)}$$

$$x_R = 10 - \frac{-200(6-10)}{540-(-200)} = 8.9189$$

$$M(x_R) = -2 \times 10^{-7} \cong 0$$

Only one iteration was necessary.

**Therefore,** $x = 8.9189\ feet$.

# Chapter 6

**6.1** $x_0 = 0.5$

$$x_{i+1} = \sin(\sqrt{x_i})$$

| iteration | $x_{i+1}$ | $\varepsilon_a\%$ |
|---|---|---|
| 1 | 0.6496369 | 23 |
| 2 | 0.7215238 | 9.9 |
| 3 | 0.7509012 | 3.9 |
| 4 | 0.7620969 | 1.5 |
| 5 | 0.7662482 | 0.5 |
| 6 | 0.7677717 | 0.2 |
| 7 | 0.7683278 | 0.07 |

note linear convergence

**6.2**

a) $x_0 = 5$

$$x_{i+1} = (.9x_i^2 - 2.5)/1.7$$

| iter | $x_{i+1}$ |
|---|---|
| 1 | 11.76 |
| 2 | 71.80 |
| 3 | 2728.1 |

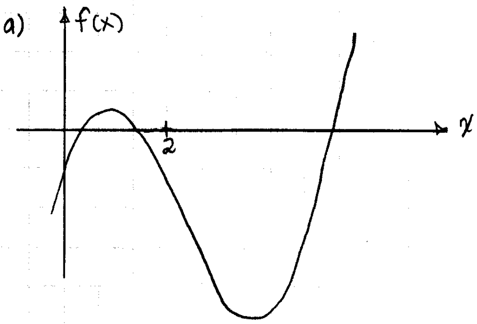
diverges

b) $$x_{i+1} = x_i - \frac{-0.9x_i^2 + 1.7x_i + 2.5}{-1.8x_i + 1.7}$$

| iter | $x_{i+1}$ | $\varepsilon_a\%$ |
|---|---|---|
| 1 | 3.424658 | 46. |
| 2 | 2.924357 | 17.1 |
| 3 | 2.861147 | 2.2 |
| 4 | 2.860105 | 0.04 |
| 5 | 2.860105 | 0 |

**6.3** $f'(x) = 1.5x^2 - 8x + 6$

a)




approx roots = 0.5, 1.5, 6.0

$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)}$$

for $x_0 = 0.5$

| iter | $x_{i+1}$ | $\varepsilon_a\%$ |
|---|---|---|
| 1 | 0.4736842 | 5.5 |
| 2 | 0.4745714 | 0.2 |
| 3 | 0.4745724 | 0.0002 |

for $x_0 = 1.5$

| iter | $x_{i+1}$ | $\varepsilon_a\%$ |
|---|---|---|
| 1 | 1.380952 | 8.6 |
| 2 | 1.369227 | 0.8 |
| 3 | 1.369103 | 0.009 |

for $x_0 = 6$

| iter | $x_{i+1}$ | $\varepsilon_a\%$ |
|---|---|---|
| 1 | 6.166667 | 2.7 |
| 2 | 6.156366 | 0.16 |
| 3 | 6.156325 | 0.0006 |

Rapid convergence with accurate initial guesses from plot

6.4 a) for $x_0 = 4.2$

| iter | $x_{i+1}$ | $\epsilon_a$ % |
|---|---|---|
| 1 | −4.849096 | 186 |
| 2 | −2.573902 | 88 |
| 3 | −1.137515 | 126 |
| 4 | −0.2727324 | 317 |
| 5 | 0.202833 | 234 |
| 6 | 0.4153548 | 51 |
| 7 | 0.4705743 | 11 |
| 8 | 0.4745519 | 0.8 |
| 9 | 0.4745724 | 0.004 |

b) for $x = 4.43$

| iter | $x_{i+1}$ | $\epsilon_a$ % |
|---|---|---|
| 1 | −3937 | 100 |
| 2 | −2623 | 50 |
| ↓ | very erratic | |
| 25 | 0.4745724 | 0.00009 |

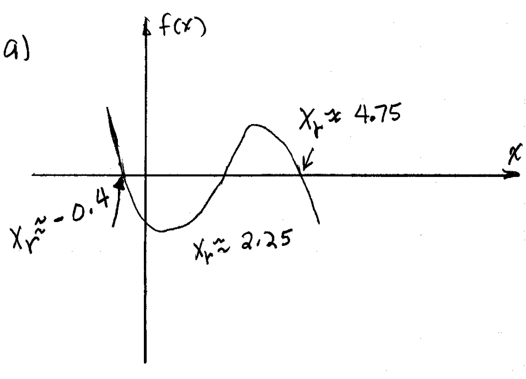
$f'(4.43) \simeq -0.00265$

$\therefore$ since $x_{i+1} = x_i - \dfrac{f(x_i)}{f'(x_i)}$

$x_{i+1}$ becomes large

6.5 $\epsilon_s = 0.5 \times 10^{2-3} = 0.05\%$

a)




b) use $x_{i-1} = -1.0$ from
$x_0 = -0.6$ part a)

$$x_{i+1} = x_i - \frac{f(x_i)(x_{i-1} - x_i)}{f(x_{i-1}) - f(x_i)}$$

| iter | $x_{i+1}$ | $\epsilon_a$ % |
|---|---|---|
| 1 | −0.4362292 | 129 |
| 2 | −0.3872544 | 55 |
| 3 | −0.3815073 | 14 |
| 4 | −0.3813334 | 1.6 |
| 5 | −0.3813328 | 0.05 |

6.6 Graph shows root near $x = 2.0$ however function is highly variable.

If $x_{i-1} = 1.0$ and $x_0 = 3.0$

method diverges.

With closer guesses

$x_{i-1} = 1.5$ and $x_0 = 2.5$

| iter | $x_{i+1}$ |
|---|---|
| 1 | 2.35693 |
| 2 | 2.547287 |
| 3 | 2.526339 |
| 4 | 2.532107 |

note the convergence at $X_r = 2.532107$ rather than
$X_r = 1.944638$

$\therefore$ If $X_{i-1} = 1.75$ and $X_0 = 2$
$X_r = 1.944608$ for iter = 4

6.7

$$f(x) = x^{3.3} - 79 = 0$$

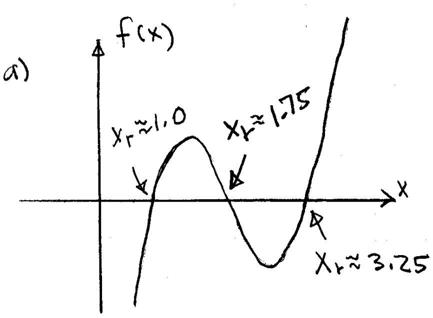
$$x_{i+1} = x_i - \frac{\delta x_i f(x_i)}{f(x_i + \delta x_i) - f(x_i)}$$

for $x_0 = 3.5$ and $\delta = 0.01$

| iter | $x_{i+1}$ | $\epsilon_a$ % |
|---|---|---|
| 1 | 3.778193 | 7.4 |
| 2 | 3.759044 | .51 |
| 3 | 3.758711 | 0.009 |

method converges for many different $x_0$ and $\delta$ because of smooth nature of function

6.8 a)




b) $f'(x) = 3x^2 - 12x + 11$

| iter | $x_{i+1}$ |
|---|---|
| 1 | 3.191304 |
| 2 | 3.068699 |
| 3 | 3.047318 |

c)

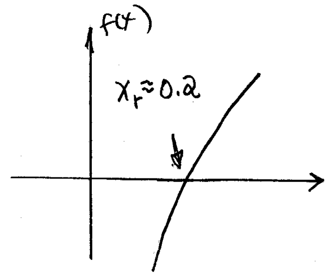
| iter | $x_{i+1}$ |
|---|---|
| 1 | 2.71111 |
| 2 | 2.87109 |
| 3 | 3.221923 |

d)

| iter | $x_{i+1}$ |
|---|---|
| 1 | 3.207573 |
| 2 | 3.082032 |
| 3 | 3.050811 |

6.9

a)




b) $f'(x) = 7\sin(x)\,e^{-x}(-1) + e^{-x}(7\cos(x))$

$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)}$$

| iter | $x_{i+1}$ |
|---|---|
| 1 | 0.144376 |
| 2 | 0.1694085 |
| 3 | 0.1701793 |

c)

| iter | $x_{i+1}$ |
|---|---|
| 1 | 0.00278 |
| 2 | 0.2182365 |
| 3 | 0.1789889 |

d)

| iter | $x_{i+1}$ |
|---|---|
| 1 | -0.1333377 |
| 2 | 0.09534793 |
| 3 | 0.1643748 |
| 4 | 0.1701739 |
| 5 | 0.17018 |

6.10 $f'(x) = 3x^2 + 2x - 5$

$f''(x) = 6x + 2$

$m = 2$

a) Regular Newton

| iter | $x_{i+1}$ | $\epsilon_a$ % |
|---|---|---|
| 1 | 0.6571428 | 69.5 |
| 2 | 0.8370023 | 21.5 |
| 3 | 0.9202698 | 9.0 |
| 4 | 0.9605444 | 4.2 |
| 5 | 0.9803708 | 2.0 |
| ⋮ | | |
| 10 | 0.9993752 | 0.05 |

note: linear convergence

b) Using $x_{i+1} = x_i - \dfrac{2f(x_i)}{f'(x_i)}$

| iter | $x_{i+1}$ | $\epsilon_a$ % |
|---|---|---|
| 1 | 1.114286 | 82 |
| 2 | 1.001565 | 11 |
| 3 | 0.9999673 | 0.16 |

c) Using $x_{i+1} = x_i - \dfrac{f(x_i)f'(x_i)}{[f'(x_i)]^2 - f(x_i)f''(x_i)}$

| iter | $x_{i+1}$ | $\epsilon_a$ % |
|---|---|---|
| 1 | 0.878788 | 77 |
| 2 | 0.9980479 | 12 |
| 3 | 0.9999452 | 0.2 |

note: b and c converge quadratically

6.11 $x_{i+1} = y_i + x_i^2 - 0.5$

a) $y_{i+1} = x_i^2 - 5x_i y_i$

| iter | $x_{i+1}$ | $y_{i+1}$ |
|---|---|---|
| 1 | 1.5 | −4 |
| 2 | −2.25 | 32.50 |
| 3 | 36.81 | 367.875 |
| 4 | 1722.535 | −66356.83 |

Diverges

b) $u = -x^2 + x - y + 0.5$

$\dfrac{\partial u}{\partial x} = -2x + 1$

$\dfrac{\partial u}{\partial y} = -1$

$v = x^2 - 5xy - y$

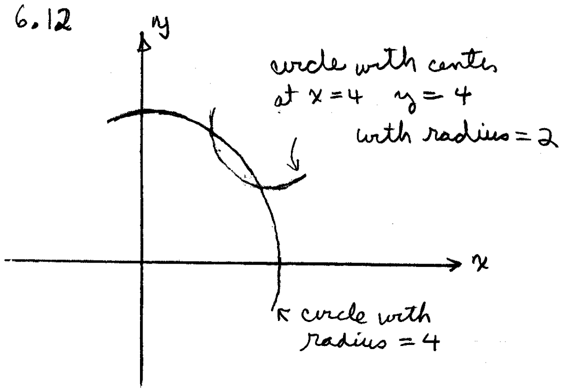
$\dfrac{\partial v}{\partial x} = 2x - 5y$

$\dfrac{\partial v}{\partial y} = -5x - 1$

| iter | $x_{i+1}$ | $y_{i+1}$ |
|---|---|---|
| 1 | 1.22222 | 0.61111 |
| 2 | 1.234179 | 0.2540807 |
| 3 | 1.233073 | 0.2015325 |
| 4 | 1.233395 | 0.2157011 |

Converges

6.12




$u = 16 - x^2 - y^2$ $\quad$ $v = 4 - (x-4)^2 - (y-4)^2$

$\frac{\partial u}{\partial x} = -2x$

$\frac{\partial u}{\partial y} = -2y$

$\frac{\partial v}{\partial x} = -2(x-4)$

$\frac{\partial v}{\partial y} = -2(y-4)$

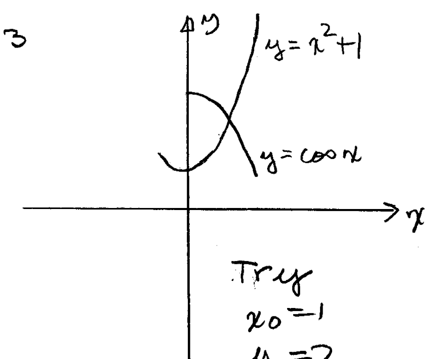
use $x_0 = 2$ $\quad y_0 = 3.5$

| iter | $x_{i+1}$ | $y_{i+1}$ |
|---|---|---|
| 1 | 2.083333 | 3.416667 |
| 2 | 2.088542 | 3.411458 |
| 3 | 2.088562 | 3.411438 |

If $x_0 = 3.5$ $\quad y_0 = 2$

| iter | $y_{i+1}$ | $x_{i+1}$ |
|---|---|---|
| 3 | 2.088562 | 3.411438 |

6.13




Try
$x_0 = 1$
$y_0 = 2$

$u = x^2 + 1 - y$

$\frac{\partial u}{\partial x} = 2x$

$\frac{\partial u}{\partial y} = -1$

$v = 3\cos x - y$

$\frac{\partial v}{\partial x} = -3\sin x$

$\frac{\partial v}{\partial y} = -1$

| iter | $x_{i+1}$ | $y_{i+1}$ |
|---|---|---|
| 1 | 0.9418962 | 1.883792 |
| 2 | 0.9221337 | 1.848913 |
| 3 | 0.9158931 | 1.83839 |

6.14

$0 = 10 - c - 2\sqrt{c}$

$c_0 = 4 \qquad \delta = 0.5$

| iter | $c_{i+1}$ | $\epsilon_a$ % |
|---|---|---|
| 1 | 5.161737 | 61 |
| 2 | 5.372652 | 3.9 |
| 3 | 5.366564 | 0.1 |
| 4 | 5.366756 | 0.003 |

6.15 $\quad$ for $c = \left(\frac{10-c}{2}\right)^2 = g(c)$

$|g'(c)| = \left|2\left(\frac{10-c}{2}\right)\right| > 1$

we expect divergence

6.15 (continued)

for $c = 10 - 2\sqrt{c} = g(c)$

$|g'(c)| = \left|\frac{1}{\sqrt{c}}\right| \leq 1$ for $c > 1$

$\therefore$ we expect convergence

for $c = \left(\frac{10-c}{2}\right)^2$ with $c_0 = 2$

| iter | $c_{i+1}$ | |
|---|---|---|
| 1 | 16 | |
| 2 | 9 | diverges |
| 3 | 0.25 | |
| 4 | 23.8 | |
| 5 | 47.4 | |

for $c = 10 - 2\sqrt{c}$

| iter | $c_{i+1}$ | |
|---|---|---|
| 1 | 7.171573 | |
| 2 | 4.644042 | converges |
| 3 | 5.689992 | |
| 4 | 5.229259 | |

6.16

Here is a VBA program to implement the Newton-Raphson algorithm and solve Example 6.3.

```
Option Explicit

Sub NewtRaph()

Dim imax As Integer, iter As Integer
Dim x0 As Single, es As Single, ea As Single

x0 = 0#
es = 0.01
imax = 20
MsgBox "Root: " & NewtR(x0, es, imax, iter, ea)
MsgBox "Iterations: " & iter
MsgBox "Estimated error: " & ea

End Sub

Function df(x)
df = -Exp(-x) - 1#
End Function

Function f(x)
f = Exp(-x) - x
End Function

Function NewtR(x0, es, imax, iter, ea)

Dim xr As Single, xrold As Single

xr = x0
iter = 0
Do
```

```
  xrold = xr
  xr = xr - f(xr) / df(xr)
  iter = iter + 1
  If (xr <> 0) Then
    ea = Abs((xr - xrold) / xr) * 100
  End If
  If ea < es Or iter >= imax Then Exit Do
Loop
NewtR = xr
End Function
```

It's application yields a root of 0.5671433 after 4 iterations. The approximate error at this point is $2.1\times10^{-5}$ %.

6.17

Here is a VBA program to implement the secant algorithm and solve Example 6.6.

```
Option Explicit

Sub SecMain()
Dim imax As Integer, iter As Integer
Dim x0 As Single, x1 As Single, xr As Single
Dim es As Single, ea As Single
x0 = 0
x1 = 1
es = 0.01
imax = 20
MsgBox "Root: " & Secant(x0, x1, xr, es, imax, iter, ea)
MsgBox "Iterations: " & iter
MsgBox "Estimated error: " & ea

End Sub

Function f(x)
f = Exp(-x) - x
End Function

Function Secant(x0, x1, xr, es, imax, iter, ea)
xr = x1
iter = 0
Do
  xr = x1 - f(x1) * (x0 - x1) / (f(x0) - f(x1))
  iter = iter + 1
  If (xr <> 0) Then
    ea = Abs((xr - x1) / xr) * 100
  End If
  If ea < es Or iter >= imax Then Exit Do
  x0 = x1
  x1 = xr
Loop
Secant = xr
End Function
```

It's application yields a root of 0.5671433 after 4 iterations. The approximate error at this point is $4.77\times10^{-3}$ %.

6.18

Here is a VBA program to implement the modified secant algorithm and solve Example 6.8.

```
Option Explicit

Sub SecMod()
Dim imax As Integer, iter As Integer
Dim x As Single, es As Single, ea As Single
x = 1
es = 0.01
imax = 20
MsgBox "Root: " & ModSecant(x, es, imax, iter, ea)
MsgBox "Iterations: " & iter
MsgBox "Estimated error: " & ea

End Sub

Function f(x)
f = Exp(-x) - x
End Function

Function ModSecant(x, es, imax, iter, ea)
Dim xr As Single, xrold As Single, fr As Single
Const del As Single = 0.01
xr = x
iter = 0
Do
  xrold = xr
  fr = f(xr)
  xr = xr - fr * del * xr / (f(xr + del * xr) - fr)
  iter = iter + 1
  If (xr <> 0) Then
    ea = Abs((xr - xrold) / xr) * 100
  End If
  If ea < es Or iter >= imax Then Exit Do
Loop
ModSecant = xr
End Function
```

It's application yields a root of 0.5671433 after 4 iterations. The approximate error at this point is $3.15\times10^{-5}$ %.

6.19

Here is a VBA program to implement the 2 equation Newton-Raphson method and solve Example 6.10.

```
Option Explicit

Sub NewtRaphSyst()

Dim imax As Integer, iter As Integer
Dim x0 As Single, y0 As Single
Dim xr As Single, yr As Single
Dim es As Single, ea As Single

x0 = 1.5
y0 = 3.5

es = 0.01
imax = 20
```

```
Call NR2Eqs(x0, y0, xr, yr, es, imax, iter, ea)

MsgBox "x, y = " & xr & ", " & yr
MsgBox "Iterations: " & iter
MsgBox "Estimated error: " & ea

End Sub
Sub NR2Eqs(x0, y0, xr, yr, es, imax, iter, ea)

Dim J As Single, eay As Single

iter = 0
Do
  J = dudx(x0, y0) * dvdy(x0, y0) - dudy(x0, y0) * dvdx(x0, y0)
  xr = x0 - (u(x0, y0) * dvdy(x0, y0) - v(x0, y0) * dudy(x0, y0)) / J
  yr = y0 - (v(x0, y0) * dudx(x0, y0) - u(x0, y0) * dvdx(x0, y0)) / J
  iter = iter + 1
  If (xr <> 0) Then
    ea = Abs((xr - x0) / xr) * 100
  End If
  If (xr <> 0) Then
    eay = Abs((yr - y0) / yr) * 100
  End If
  If eay > ea Then ea = eay
  If ea < es Or iter >= imax Then Exit Do
  x0 = xr
  y0 = yr
Loop

End Sub

Function u(x, y)
u = x ^ 2 + x * y - 10
End Function

Function v(x, y)
v = y + 3 * x * y ^ 2 - 57
End Function

Function dudx(x, y)
dudx = 2 * x + y
End Function

Function dudy(x, y)
dudy = x
End Function

Function dvdx(x, y)
dvdx = 3 * y ^ 2
End Function

Function dvdy(x, y)
dvdy = 1 + 6 * x * y
End Function
```

It's application yields roots of $x = 2$ and $y = 3$ after 4 iterations. The approximate error at this point is $1.59\times10^{-5}$ %.

6.20

The program from Prob. 6.19 can be set up to solve Prob. 6.11, by changing the functions to

```
Function u(x, y)
u = y + x ^ 2 - 0.5 - x
End Function

Function v(x, y)
v = x ^ 2 - 5 * x * y - y
End Function

Function dudx(x, y)
dudx = 2 * x - 1
End Function

Function dudy(x, y)
dudy = 1
End Function

Function dvdx(x, y)
dvdx = 2 * x ^ 2 - 5 * y
End Function

Function dvdy(x, y)
dvdy = -5 * x
End Function
```

Using a stopping criterion of 0.01%, the program yields $x = 1.233318$ and $y = 0.212245$ after 7 iterations with an approximate error of $2.2\times10^{-4}$.

The program from Prob. 6.19 can be set up to solve Prob. 6.12, by changing the functions to

```
Function u(x, y)
u = (x - 4) ^ 2 + (y - 4) ^ 2 - 4
End Function

Function v(x, y)
v = x ^ 2 + y ^ 2 - 16
End Function

Function dudx(x, y)
dudx = 2 * (x - 4)
End Function

Function dudy(x, y)
dudy = 2 * (y - 4)
End Function

Function dvdx(x, y)
dvdx = 2 * x
End Function

Function dvdy(x, y)
dvdy = 2 * y
End Function
```

Using a stopping criterion of 0.01% and initial guesses of 2 and 3.5, the program yields $x = 2.0888542$ and $y = 3.411438$ after 3 iterations with an approximate error of $9.8\times10^{-4}$.

Using a stopping criterion of 0.01% and initial guesses of 3.5 and 2, the program yields $x$ = 3.411438 and $y$ = 2.0888542 after 3 iterations with an approximate error of $9.8\times10^{-4}$.

6.21

$$x = \sqrt{a}$$

$$x^2 = a$$

$$f(x) = x^2 - a = 0$$

$$f'(x) = 2x$$

Substitute into Newton Raphson formula (Eq. 6.6),

$$x = x - \frac{x^2 - a}{2x}$$

Combining terms gives

$$x = \frac{2x(x) - x^2 + a}{2} = \frac{x^2 + a / x}{2}$$

6.22

SOLUTION:

$$f(x) = \tanh\left(x^2 - 9\right)$$
$$f'(x) = \left[\text{sech}^2\left(x^2 - 9\right)\right](2x)$$
$$x_o = 3.1$$

$$x_{i+1} = x_i - \frac{f(x)}{f'(x)}$$

| iteration | $x_{i+1}$ |
|---|---|
| 1 | 2.9753 |
| 2 | 3.2267 |
| 3 | 2.5774 |
| 4 | 7.9865 |

The solution diverges from its real root of x = 3. Due to the concavity of the slope, the next iteration will always diverge. The sketch should resemble figure 6.6(a).

6.23

SOLUTION:

$$f(x) = 0.0074x^4 - 0.284x^3 + 3.355x^2 - 12.183x + 5$$
$$f'(x) = 0.0296x^3 - 0.852x^2 + 6.71x - 12.183$$

$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)}$$

| $i$ | $x_{i+1}$ |
|---|---|
| 1 | 9.0767 |
| 2 | -4.01014 |
| 3 | -3.2726 |

The solution converged on another root. The partial solutions for each iteration intersected the x-axis along its tangent path beyond a different root, resulting in convergence elsewhere.

6.24

SOLUTION:

$f(x) = \pm\sqrt{16-(x+1)^2} + 2$

$$x_{i+1} = x_i - \frac{f(x_i)(x_{i-1} - x_i)}{f(x_{i-1}) - f(x_i)}$$

1st iteration

$$x_{i-1} = 0.5 \Rightarrow f(x_{i-1}) = -1.708$$
$$x_i = 3 \Rightarrow f(x_i) = 2$$
$$x_{i+1} = 3 - \frac{2(0.5-3)}{(-1.708-2)} = 1.6516$$

2nd iteration

$$x_i = 1.6516 \Rightarrow f(x_i) = -0.9948$$
$$x_{i-1} = 0.5 \Rightarrow f(x_{i-1}) = -1.46$$
$$x_{i+1} = 1.6516 - \frac{-0.9948(0.5-1.6516)}{(-1.46 - -0.9948)} = 4.1142$$

The solution diverges because the secant created by the two x-values yields a solution outside the functions domain.

# Chapter 7

7.1 Following Example 7.1 and pseudo code

$f_3(x) = 3 - x - 3x^2 + x^3$

$r = 0 \quad \therefore \quad x = 2$

is a root

7.2

$f_4(x) = -49 - 21x - 7x^2 - 4x^3 + x^4$

$r = -86$

7.3 Plot indicates a root at $x = 2.0$

a)

Therefore try $x_0 = 1.0$
$x_1 = 1.5$
$x_2 = 1.75$

following Example 7.2

$\delta_0 = 3.25 \qquad \delta_1 = 7.1875$

$a = 5.25 \qquad b = 8.5$

$c = -2.578$

discriminant $= 11.24$

$x_3 = 2.011$

b) Plot suggests a root at $x = 0.5$

Try $x_0 = 0.4$
$x_1 = 0.6$
$x_2 = 0.8$

following Example 7.2

$\delta_0 = 4.26 \qquad \delta_1 = 4.78$

$a = 1.299 \qquad b = 5.04$

$c = 1.392$

discriminant $= 4.26$

$x_3 = 0.5007$

7.4

a) Plot suggests a real root $x = 1$

Try $x_0 = 0.25$
$x_1 = 0.50$
$x_2 = 0.75$

$\delta_0 = 1.6875 \qquad \delta_1 = 1.9375$

$a = 0.5 \qquad b = 2.0625$

$c = -0.640625$

$x_3 = 1.0402$

Use Pseudocode in Fig 7.4
and Fortran to obtain
complex roots

$0 \pm 1.414i$

b) No real roots

Use Pseudocode in Fig 7.4
and Fortran to obtain
complex roots

$-0.5 \pm 1.323i$

$+0.5 \pm 1.323i$

c) No Real roots

Use Pseudocode in Fig 7.4
and Fortran to obtain
complex roots

$1 \pm 2i$

$0 \pm i$

7.5 Plot suggests
a) 3 real roots,

$root_1 = 0.4357$
$root_2 = 2.0$
$root_3 = 3.278$

Try $r = 1$, $s = -1$ and follow Example 7.3

<u>1st iteration</u>

$\Delta r = 1.085$ $\quad \Delta s = 0.887$

$r = 2.085$ $\quad s = -0.1129$

<u>2nd iteration</u>

$\Delta r = 0.402$ $\quad \Delta s = -0.556$

$r = 2.49$ $\quad s = -0.67$

<u>3rd iteration</u>

$\Delta r = -0.064$ $\quad \Delta s = -0.206$

$r = 2.426$ $\quad s = -0.876$

<u>4th iteration</u>

$\Delta r = 0.0096$ $\quad \Delta s = 0.0045$

$r = 2.43$ $\quad s = -0.87$

$$root_1 = \frac{r + \sqrt{r^2 + 4s}}{2}$$

$$= 1.999$$

$$root_2 = \frac{r - \sqrt{r^2+4s}}{2}$$

$$= 0.4357$$

with remaining $root_3 = 3.278$

b) plot suggests 3 real roots

root$_1$ = 2.29
root 2 = 0.95
root 3 = 1.15

Try $r = 2$ $s = -0.5$

<u>1st iteration</u>

$\Delta r = 0.23$ $\Delta s = -0.538$

$r = 2.23$ $s = -1.038$

<u>2nd iteration</u>

$\Delta r = -0.179$ $\Delta s = -0.042$

$r = 2.05$ $S = -1.08$

<u>3rd iteration</u>

$\Delta r = 0.053$ $\Delta s = -0.0165$

$r = 2.103$ $s = -1.096$

$$root_1 = \frac{r + \sqrt{r^2+4s}}{2}$$

$$= 1.14956$$

$$root_2 = \frac{r - \sqrt{r^2+4s}}{2}$$

$$= 0.95387$$

with remaining $root_3 = 2.29$

c) Try $r = 0.5$ $s = -0.5$

<u>1st iteration</u>

$\Delta r = -0.422$ $\Delta s = -0.456$

$r = 0.078$ $s = -0.956$

<u>2nd iteration</u>

$\Delta r = -0.078$ $\Delta s = -0.042$

$r = -0.000266$
$s = -0.998$

$r^2 + 4s < 0$ therefore

real part $= r/2 = -0.000133$

$$\text{imaginary part} = \frac{\sqrt{r^2+4s}}{2}\, i$$

$$= 0.9991$$

remaining quadratic gives

real = 1.0
imaginary = 2.0

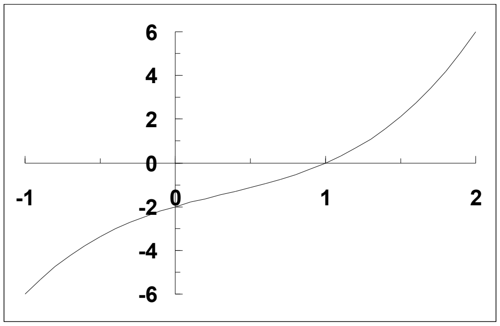
7.6 Errata in Fig. 7.4; 6th line from the bottom of the algorithm: the > should be changed to >=

*IF* ($|dx_r| < eps*x_r$ *OR iter* **>=** *maxit) EXIT*

Here is a VBA program to implement the Müller algorithm and solve Example 7.2.

```
Option Explicit

Sub TestMull()

Dim maxit As Integer, iter As Integer
Dim h As Single, xr As Single, eps As Single

h = 0.1
xr = 5
eps = 0.001
maxit = 20

Call Muller(xr, h, eps, maxit, iter)

MsgBox "root = " & xr
MsgBox "Iterations: " & iter

End Sub

Sub Muller(xr, h, eps, maxit, iter)


Dim x0 As Single, x1 As Single, x2 As Single
Dim h0 As Single, h1 As Single, d0 As Single, d1 As Single
Dim a As Single, b As Single, c As Single
Dim den As Single, rad As Single, dxr As Single

x2 = xr
x1 = xr + h * xr
x0 = xr - h * xr
Do
  iter = iter + 1
  h0 = x1 - x0
  h1 = x2 - x1
  d0 = (f(x1) - f(x0)) / h0
  d1 = (f(x2) - f(x1)) / h1
  a = (d1 - d0) / (h1 + h0)
  b = a * h1 + d1
  c = f(x2)
  rad = Sqr(b * b - 4 * a * c)
  If Abs(b + rad) > Abs(b - rad) Then
    den = b + rad
  Else
    den = b - rad
  End If
  dxr = -2 * c / den
  xr = x2 + dxr
  If Abs(dxr) < eps * xr Or iter >= maxit Then Exit Do
  x0 = x1
  x1 = x2
  x2 = xr
Loop
End Sub

Function f(x)
f = x ^ 3 - 13 * x - 12
End Function
```

7.7 The plot suggests a root at 1




Using an initial guess of 1.5 with h = 0.1 and eps = 0.001 yields the correct result of 1 in 4 iterations.

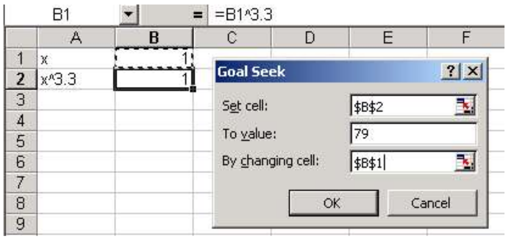
7.8 Here is a VBA program to implement the Bairstow algorithm and solve Example 7.3.

```
Option Explicit

Sub PolyRoot()

Dim n As Integer, maxit As Integer, ier As Integer, i As Integer
Dim a(10) As Single, re(10) As Single, im(10) As Single
Dim r As Single, s As Single, es As Single
n = 5
a(0) = 1.25: a(1) = -3.875: a(2) = 2.125: a(3) = 2.75: a(4) = -3.5: a(5) = 1
maxit = 20
es = 0.01
r = -1
s = -1
Call Bairstow(a(), n, es, r, s, maxit, re(), im(), ier)
For i = 1 To n
  If im(i) >= 0 Then
    MsgBox re(i) & " + " & im(i) & "i"
  Else
    MsgBox re(i) & " - " & Abs(im(i)) & "i"
  End If
Next i

End Sub

Sub Bairstow(a, nn, es, rr, ss, maxit, re, im, ier)

Dim iter As Integer, n As Integer, i As Integer
Dim r As Single, s As Single, ea1 As Single, ea2 As Single
Dim det As Single, dr As Single, ds As Single
Dim r1 As Single, i1 As Single, r2 As Single, i2 As Single
Dim b(10) As Single, c(10) As Single

r = rr
s = ss
n = nn
ier = 0
ea1 = 1
ea2 = 1
Do
  If n < 3 Or iter >= maxit Then Exit Do
  iter = 0
  Do
    iter = iter + 1
    b(n) = a(n)
    b(n - 1) = a(n - 1) + r * b(n)
    c(n) = b(n)
    c(n - 1) = b(n - 1) + r * c(n)
    For i = n - 2 To 0 Step -1
```

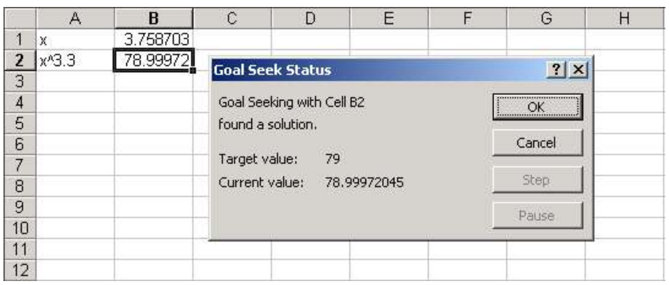
```
      b(i) = a(i) + r * b(i + 1) + s * b(i + 2)
      c(i) = b(i) + r * c(i + 1) + s * c(i + 2)
    Next i
    det = c(2) * c(2) - c(3) * c(1)
    If det <> 0 Then
      dr = (-b(1) * c(2) + b(0) * c(3)) / det
      ds = (-b(0) * c(2) + b(1) * c(1)) / det
      r = r + dr
      s = s + ds
      If r <> 0 Then ea1 = Abs(dr / r) * 100
      If s <> 0 Then ea2 = Abs(ds / s) * 100
    Else
      r = r + 1
      s = s + 1
      iter = 0
    End If
    If ea1 <= es And ea2 <= es Or iter >= maxit Then Exit Do
  Loop
  Call Quadroot(r, s, r1, i1, r2, i2)
  re(n) = r1
  im(n) = i1
  re(n - 1) = r2
  im(n - 1) = i2
  n = n - 2
  For i = 0 To n
    a(i) = b(i + 2)
  Next i
Loop
If iter < maxit Then
  If n = 2 Then
    r = -a(1) / a(2)
    s = -a(0) / a(2)
    Call Quadroot(r, s, r1, i1, r2, i2)
    re(n) = r1
    im(n) = i1
    re(n - 1) = r2
    im(n - 1) = i2
  Else
    re(n) = -a(0) / a(1)
    im(n) = 0
  End If
Else
  ier = 1
End If
End Sub

Sub Quadroot(r, s, r1, i1, r2, i2)

Dim disc
disc = r ^ 2 + 4 * s
If disc > 0 Then
  r1 = (r + Sqr(disc)) / 2
  r2 = (r - Sqr(disc)) / 2
  i1 = 0
  i2 = 0
Else
  r1 = r / 2
  r2 = r1
  i1 = Sqr(Abs(disc)) / 2
  i2 = -i1
End If
End Sub
```

7.9 See solutions to Prob. 7.5

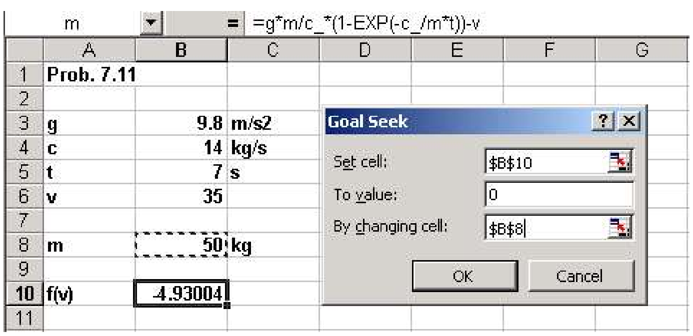
7.10 The goal seek set up is



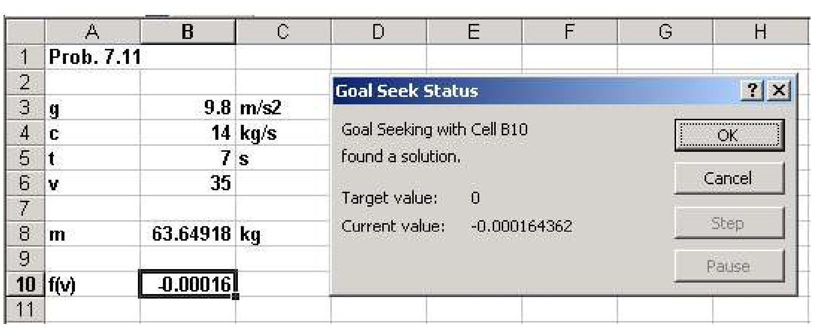

The result is



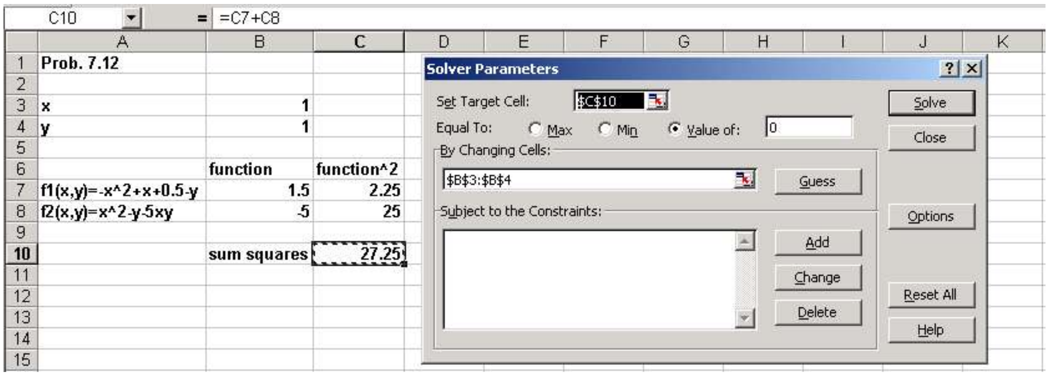

7.11 The goal seek set up is shown below. Notice that we have named the cells containing the parameter values with the labels in column A.



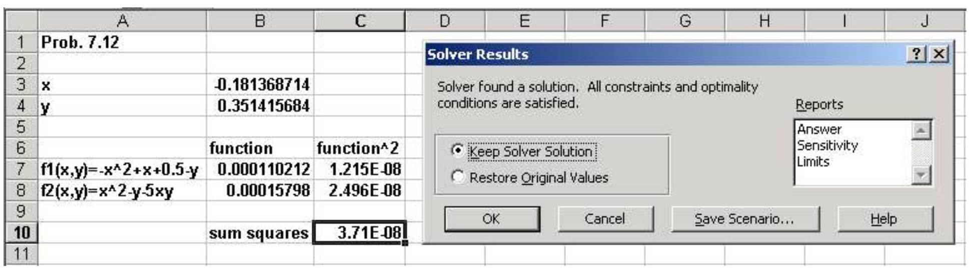

The result is 63.649 kg as shown here:



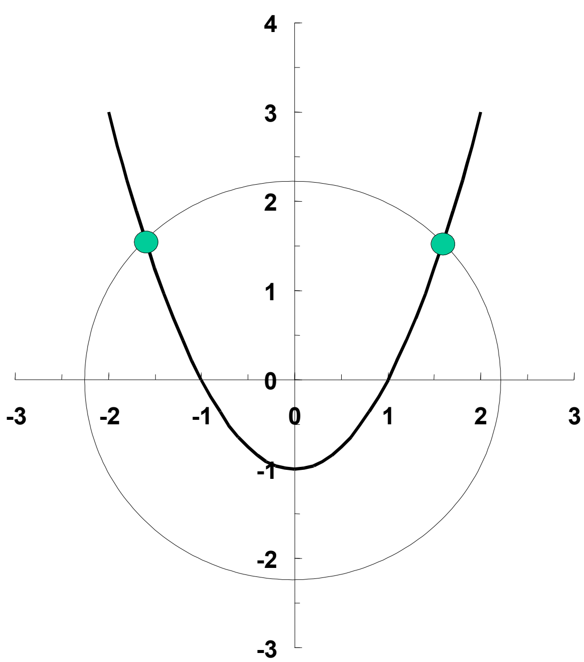
7.12 The Solver set up is shown below using initial guesses of $x = y = 1$. Notice that we have rearranged the two functions so that the correct values will drive them both to zero. We then drive the sum of their squared values to zero by varying x and y. This is done because a straight sum would be zero if $f_1(x,y) = -f_2(x,y)$.



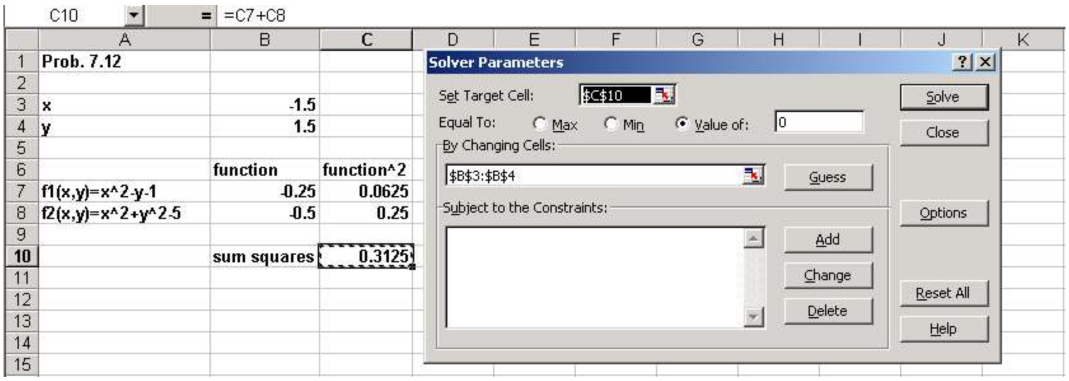

The result is



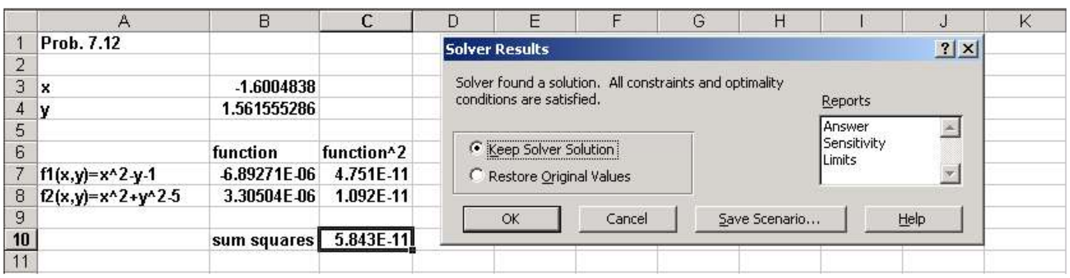

7.13 A plot of the functions indicates two real roots at about (−1.5, 1.5) and (−1.5, 1.5).



The Solver set up is shown below using initial guesses of (−1.5, 1.5). Notice that we have rearranged the two functions so that the correct values will drive them both to zero. We then drive the sum of their squared values to zero by varying x and y. This is done because a straight sum would be zero if $f_1(x,y) = -f_2(x,y)$.




The result is




For guesses of (1.5, 1.5) the result is (1.6004829, 1.561556).

7.14 First we will multiply out the polynomial so that it is in standard format

$$f_5(x) = (x+2)(x-6)(x-1)(x+4)(x-8) = x^5 - 9x^4 - 20x^3 + 204x^2 + 208x - 384$$

Note this can also be done directly in MATLAB by first setting up a vector holding the roots

```
>> v=[-2 6 1 -4 8];
```

and then using the poly function

```
>> a=poly(v)

a =
     1    -9   -20   204   208  -384
```

Next, we can evaluate this polynomial at a specific value. For example, at $x = 1$ (one of the roots), it would evaluate to zero

```
>> polyval(a,1)

ans =
     0
```

At $x = 0$, it would evaluate to

```
>> polyval(a,0)

ans =
  -384
```

The derivatives can be computed by

```
>> polyder(a)

ans =
     5   -36   -60   408   208
```

Next, a polynomial with two of the original roots can be formed

```
>> v=[-2 6];
>> b=poly(v)

b =
     1    -4   -12
```

We can divide this polynomial into the original polynomial by

```
>> [d,e]=deconv(a,b)
```

with the result being a quotient (a third-order polynomial, d) and a remainder (e)

```
d =
     1    -5   -28    32

e =
     0     0     0     0     0     0
```

Because the polynomial is a perfect divisor, the remainder polynomial has zero coefficients.

```
>> roots (d)
```

with the expected result that the remaining roots of the original polynomial are found

```
ans =
     8.0000
    -4.0000
     1.0000
```

We can now multiply d by b to come up with the original polynomial,

```
>> conv(d,b)

ans =
1    -9    -20    204    208    -384
```

Finally, we can determine all the roots of the original polynomial by

```
>> r=roots(a)

r =
   8.0000
   6.0000
  -4.0000
  -2.0000
   1.0000
```

7.15

```
p=[0.7 -4 6.2 -2];
roots(p)

ans =

    3.2786
  2.0000
    0.4357

p=[-3.704 16.3 -21.97 9.34];
roots(p)

ans =

    2.2947
    1.1525
    0.9535

p=[1 -2 6 -2 5];
roots(p)

ans =

   1.0000 + 2.0000i
   1.0000 - 2.0000i
  -0.0000 + 1.0000i
  -0.0000 - 1.0000i
```

7.16 Here is a program written in Compaq Visual Fortran 90,

```
PROGRAM Root

Use IMSL     !This establishes the link to the IMSL libraries
```

```
Implicit None       !forces declaration of all variables
Integer::nroot
Parameter(nroot=1)
Integer::itmax=50
Real::errabs=0.,errrel=1.E-5,eps=0.,eta=0.
Real::f,x0(nroot) ,x(nroot)
External f
Integer::info(nroot)

Print *, "Enter initial guess"
Read *, x0
Call ZReal(f,errabs,errrel,eps,eta,nroot,itmax,x0,x,info)
Print *, "root = ", x
Print *, "iterations = ", info

End

Function f(x)
Implicit None
Real::f,x
f = x**3-x**2+2*x-2
End
```

The output for Prob. 7.4*a* would look like

```
Enter initial guess
.5
 root =    1.000000
 iterations =            7
Press any key to continue
```

7.17

$h_o = 0.55 - 0.53 = 0.02$

$h_1 = 0.54 - 0.55 = -0.01$

$$\delta_o = \frac{58 - 19}{0.55 - 0.53} = 1950$$

$$\delta_1 = \frac{44 - 58}{0.54 - 0.55} = 1400$$

$$a = \frac{\delta_1 - \delta_o}{h_1 + h_o} = -55000$$

$b = a\,h_1 + \delta_1 = 1950$

$c = 44$

$$\sqrt{b^2 - 4ac} = 3671.85$$

$$t_o = 0.54 + \frac{-2(44)}{1950 + 3671.85} = 0.524\ \text{s}$$

Therefore, the pressure was zero at 0.524 seconds.

7.18

I) Graphically:

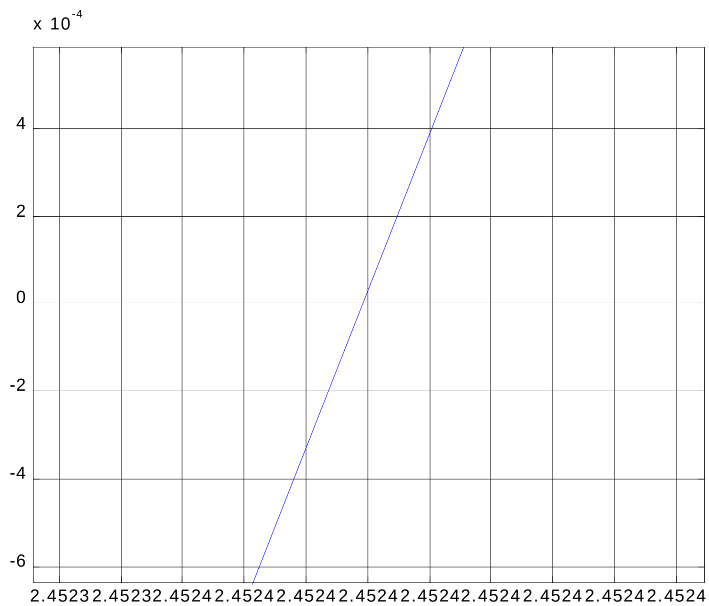
```
EDU»C=[1 3.6 0 -36.4];roots(C)

ans =  -3.0262+ 2.3843i
          -3.0262- 2.3843i
            2.4524
```

The answer is 2.4524 considering it is the only real root.

II) Using the Roots Function:

```
EDU» x=-1:0.001:2.5;f=x.^3+3.6.*x.^2-36.4;plot(x,f);grid;zoom
```

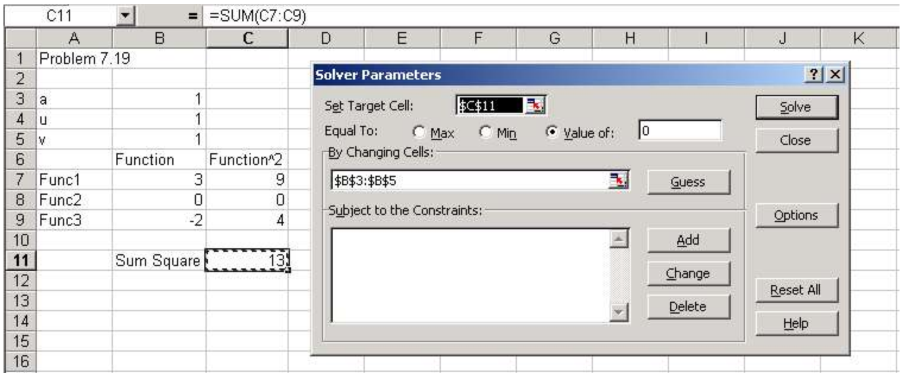
By zooming in the plot at the desired location, we get the same answer of 2.4524.

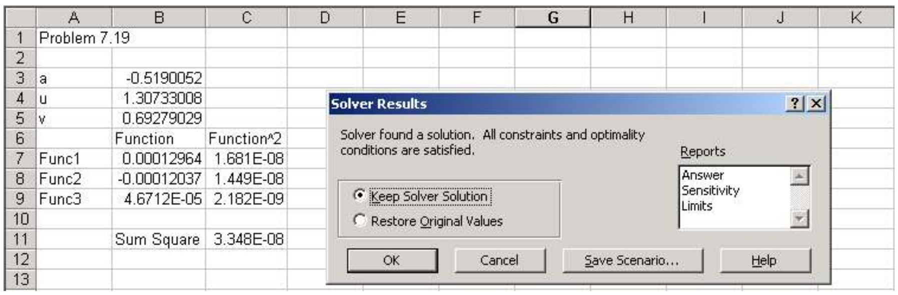



7.19

Excel Solver Solution: The 3 functions can be set up as roots problems:

$$f_1(a,u,v) = a^2 - u^2 + 3v^2 = 0$$
$$f_2(a,u,v) = u + v - 2 = 0$$
$$f_3(a,u,v) = a^2 - 2a - u = 0$$




Symbolic Manipulator Solution:

```
>>syms a u v
>>S=solve(u^2-3*v^2-a^2,u+v-2,a^2-2*a-u)

>>double (S.a)
      ans = 2.9270 + 0.3050i
            2.9270 - 0.3050i
            -0.5190
            -1.3350

>>double (S.u)
      ans = 2.6203 + 1.1753i
            2.6203 - 1.1753i
            1.3073
            4.4522

>>double (S.v)
      ans = -0.6203 + 1.1753i
            -0.6203 - 1.1753i
            0.6297
            -2.4522
```

Therefore, we see that the two real-valued solutions for $a$, $u$, and $v$ are (-0.5190,1.3073,0.6927) and (-1.3350,4.4522,-2.4522).

7.20 The roots of the numerator are: $s = -2$, $s = -3$, and $s = -4$.
The roots of the denominator are: $s = -1$, $s = -3$, $s = -5$, and $s = -6$.

$$G(s) = \frac{(s+2)(s+3)(s+4)}{(s+1)(s+3)(s+5)(s+6)}$$

# Chapter 9

## 9.1

$$\begin{bmatrix} 0 & 2 & 6 \\ 8 & 3 & 0 \\ 1 & 0 & 1 \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \end{Bmatrix} = \begin{Bmatrix} 30 \\ 20 \\ 10 \end{Bmatrix}$$

transpose

$$\begin{bmatrix} 0 & 8 & 1 \\ 2 & 3 & 0 \\ 6 & 0 & 1 \end{bmatrix}$$

## 9.2

a) A = 3x2   D = 2x4
B = 3x3   E = 3x3
C = 3x1   F = 2x3
    G = 1x3

b) square, B and E
column, C
row, G

c) $a_{12} = 5$
$b_{23} = 6$
$d_{32}$ = does not exist
$e_{22} = 1$
$f_{12} = 0$
$g_{12} = 6$

d)

(1) $A+E = \begin{bmatrix} 5 & 8 & 13 \\ 8 & 3 & 9 \\ 5 & 0 & 9 \end{bmatrix}$

(2) $B-E = \begin{bmatrix} -3 & 2 & -1 \\ 6 & -1 & -3 \\ 3 & 0 & 1 \end{bmatrix}$

(3) not possible

(4) $5B = \begin{bmatrix} 20 & 15 & 35 \\ 5 & 10 & 30 \\ 5 & 0 & 20 \end{bmatrix}$

(5) $E \times B = \begin{bmatrix} 15 & 13 & 61 \\ 32 & 23 & 67 \\ 21 & 12 & 48 \end{bmatrix}$

(6) $B \times E = \begin{bmatrix} 53 & 23 & 68 \\ 39 & 7 & 42 \\ 17 & 5 & 26 \end{bmatrix}$

(7) $[E]\{C\} = \begin{Bmatrix} 38 \\ 23 \\ 13 \end{Bmatrix}$

(8) $C^T = \begin{bmatrix} 2 & 6 & 1 \end{bmatrix}$

(9) $D^T = \begin{bmatrix} 5 & 2 \\ 4 & 1 \\ 3 & 7 \\ 6 & 5 \end{bmatrix}$

(10) $IB = B$

9.3

a) $$XY = \begin{bmatrix} 12 & 24 \\ 28 & 40 \\ 46 & 16 \end{bmatrix}$$

$$XZ = \begin{bmatrix} 37 & 49 \\ 63 & 83 \\ 31 & 39 \end{bmatrix}$$

$$YZ = \begin{bmatrix} 6 & 6 \\ 25 & 33 \end{bmatrix}$$

$$ZY = \begin{bmatrix} 7 & 4 \\ 44 & 32 \end{bmatrix}$$

b) columns of first must equal rows of second

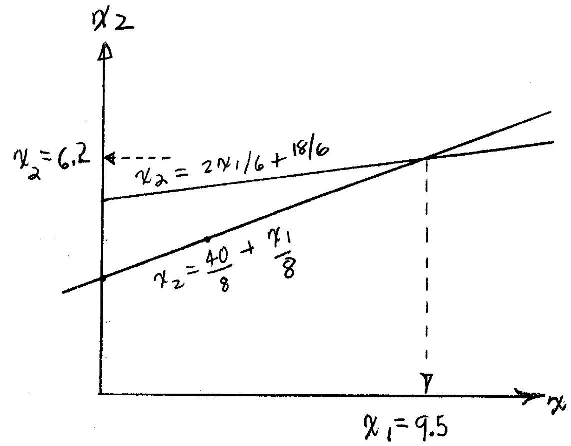
c) $YZ \neq ZY$ in part (a)

9.4 $$2x_1 - 6x_2 = -18$$

$$-x_1 + 8x_2 = 40$$

$$x_2 = \frac{2x_1}{6} + \frac{18}{6}$$

$$x_2 = \frac{40}{8} + \frac{x_1}{8}$$

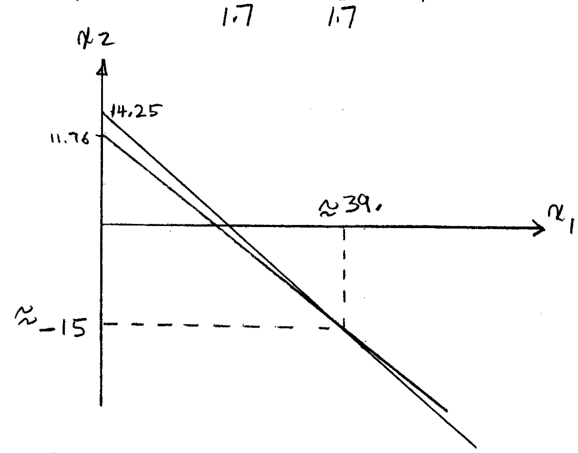



$$2(9.5) - 6(6.2) \approx -18$$

$$-9.5 + 8(6.8) \approx 40$$

9.5 $$x_2 = 14.25 - 0.77x_1$$

a) $$x_2 = \frac{20}{1.7} - \frac{1.2}{1.7}x_1$$




b) may be ill conditioned because slopes are similar

c) $x_1 = 38.76$
$x_2 = -15.60$

9.6 a)

$$A_1 = \begin{vmatrix} 1 & 2 \\ 1 & 0 \end{vmatrix} = -2$$

$$A_2 = \begin{vmatrix} 2 & 2 \\ 3 & 0 \end{vmatrix} = -6$$

$$A_3 = \begin{vmatrix} 2 & 1 \\ 3 & 1 \end{vmatrix} = -1$$

$$D = 0(-2) - 2(-6) + 5(-1)$$
$$= 7$$

b)

$$x_1 = \frac{\begin{vmatrix} 1 & 2 & 5 \\ 1 & 1 & 2 \\ 2 & 1 & 0 \end{vmatrix}}{7}$$

$$x_1 = \frac{1}{7} = 0.1428571$$

$$x_2 = \frac{\begin{vmatrix} 0 & 1 & 5 \\ 2 & 1 & 2 \\ 3 & 2 & 0 \end{vmatrix}}{7}$$

$$x_2 = \frac{11}{7} = 1.571429$$

$$x_3 = \frac{\begin{vmatrix} 0 & 2 & 1 \\ 2 & 1 & 1 \\ 3 & 1 & 2 \end{vmatrix}}{7}$$

$$x_2 = -\frac{3}{7} = -0.4285714$$

c)

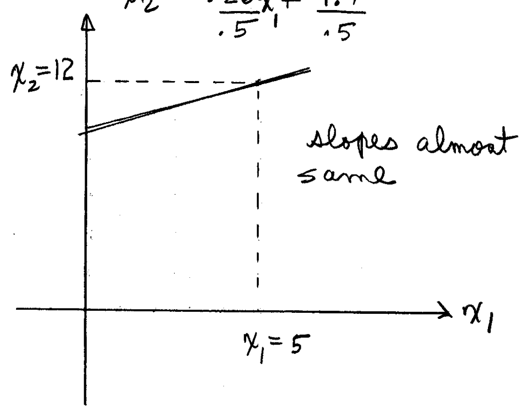
$$2\left(\frac{11}{7}\right) + 5\left(\frac{-3}{7}\right) = 1 \checkmark$$

$$2\left(\frac{1}{7}\right) + \frac{11}{7} - 2\left(\frac{3}{7}\right) = 1 \checkmark$$

$$\frac{3}{7} + \frac{11}{7} = 2 \checkmark$$

9.7 $x_2 = 0.5x_1 + 9.5$

a) $x_2 = \frac{.26}{.5}x_1 + \frac{4.7}{.5}$




b) $0.5x_1 - x_2 = -9.5$

$0.52x_1 - x_2 = -9.4$

$$D = -.5 - (1)(.52) = 0.02$$

c) Plot and D suggest ill conditioning

d) $x_1 = 5$ $x_2 = 12$

e) with $a_{11} = 0.55$

$x_1 = -3.33$ $x_2 = 7.67$

as expected small change in $a_{11}$ gives large change in solution

9.8

$$\begin{array}{cccc|c} (1) & -12 & 1 & -1 & -20 \\ (2) & -2 & -4 & 2 & 10 \\ (3) & 1 & 2 & 2 & 25 \end{array}$$

$(1)\times\frac{1}{6}$ $\quad -2 \quad 1/6 \quad -1/6 \quad -20/6 \quad (4)$

$(2)-(4)$

$$\begin{array}{cccc} -12 & 1 & -1 & -20 \\ 0 & -25/6 & 13/6 & 80/6 \\ 1 & 2 & 2 & 25 \end{array}$$

$(1)\times -\frac{1}{12}$

$$1 \quad -1/12 \quad \frac{1}{12} \quad 20/12 \quad (5)$$

$(3)-(5)$

$$\begin{array}{ccccc} -12 & 1 & -1 & -20 & \\ 0 & -25/6 & 13/6 & 80/6 & (6) \\ 0 & 25/12 & 23/12 & 280/12 & (8) \end{array}$$

$(6)\times -\frac{1}{2}$

$$0 \quad \frac{25}{12} \quad \frac{-13}{12} \quad -\frac{80}{12} \quad (7)$$

$(8)-(7)$

$$\begin{array}{ccccc} -12 & 1 & -1 & -20 & (11) \\ 0 & -25/6 & 13/6 & 80/6 & (10) \\ 0 & 0 & 36/12 & 360/12 & (9) \end{array}$$

from (9) $\quad x_3 = 10.0$

from (10) $\quad -\frac{25}{6} x_2 = \frac{80}{6} - \frac{130}{6}$

$$x_2 = 2.0$$

from (11)

$$-12 x_1 = -20 + 10 - 2$$

$$x_1 = 1.0$$

check:

$$\begin{array}{l} -12(1) + 2 - 10 = -20 \ \checkmark \\ -2 - 8 + 20 = 10 \ \checkmark \\ 1 + 4 + 20 = 25 \ \checkmark \end{array}$$

9.9 <u>pivot</u> to obtain

$$\begin{array}{ccccc} 6 & 1 & 1 & 6 & (1) \\ 5 & 1 & 2 & 4 & (2) \\ 4 & 1 & -1 & -2 & (3) \end{array}$$

(1) × 5/6 subtract from (2)
(1) × 4/6 subtract from (3)

$$\begin{array}{cccc} 6 & 1 & 1 & 6 \\ 0 & 1/6 & 7/6 & -1 \\ 0 & 2/6 & -10/6 & -6 \end{array}$$

pivot again

$$\begin{array}{ccccc} 6 & 1 & 1 & 6 & (4) \\ 0 & \frac{2}{6} & -\frac{10}{6} & -6 & (5) \\ 0 & 1/6 & \frac{7}{6} & -1 & (6) \end{array}$$

(5) × 1/2 subtract from (6)

$$\begin{array}{ccccc} 6 & 1 & 1 & 6 & (7) \\ 0 & 2/6 & -\frac{10}{6} & -6 & (8) \\ 0 & 0 & \frac{12}{6} & 2 & (9) \end{array}$$

(9) gives $x_3 = 1$
(8) gives $x_2 = -13$
(7) gives $x_1 = 3$

$$\begin{array}{l} 6(3) - 13 + 1 = 6 \ \checkmark \\ 5(3) - 13 + 2 = 4 \ \checkmark \\ 4(3) - 13 - 1 = -2 \ \checkmark \end{array}$$

9.10 normalize 1st row

| | | | | |
|---|---|---|---|---|
| 1 | 1/2 | -1/2 | 1/2 | ① |
| 5 | 2 | 2 | -4 | ② |
| 3 | 1 | 1 | 5 | ③ |

subtract ① × 5 from ②
subtract ① × 3 from ③

| | | | |
|---|---|---|---|
| 1 | 1/2 | -1/2 | 1/2 |
| 0 | -1/2 | 9/2 | -13/2 |
| 0 | -1/2 | 5/2 | 7/2 |

normalize 2nd row: ⑤/(-1/2)

| | | | | |
|---|---|---|---|---|
| 1 | 1/2 | -1/2 | 1/2 | ④ |
| 0 | 1 | -9 | 13 | ⑤ |
| 0 | -1/2 | 5/2 | 9/2 | ⑥ |

subtract ⑤ × (-1/2) from ⑥
subtract ⑤ × (1/2) from ④

| | | | | |
|---|---|---|---|---|
| 1 | 0 | 8/2 | -12/2 | ⑦ |
| 0 | 1 | -9 | 13 | ⑧ |
| 0 | 0 | -4/2 | 20/2 | ⑨ |

normalize 3rd row ⑨/(-4/2)

| | | | | |
|---|---|---|---|---|
| 1 | 0 | 8/2 | -12/2 | ⑩ |
| 0 | 1 | -9 | 13 | ⑪ |
| 0 | 0 | 1 | -5 | ⑫ |

subtract ⑫ × (-9) from ⑪
subtract ⑫ × (8/2) from ⑩

| | | | |
|---|---|---|---|
| 1 | 0 | 1 | 14 |
| 0 | 1 | 0 | -32 |
| 0 | 0 | 1 | -5 |

9.11 Use same techniques as problems 9.8–9.10

answers are:

$$x_1 = -0.25$$
$$x_2 = -0.50$$
$$x_3 = 2.25$$

9.12

$$\begin{bmatrix} 50 & 1 & 0 & 0 & 0 \\ 80 & -1 & 1 & 0 & 0 \\ 60 & 0 & -1 & 1 & 0 \\ 70 & 0 & 0 & -1 & 1 \\ 90 & 0 & 0 & 0 & -1 \end{bmatrix} \begin{Bmatrix} a \\ T_{12} \\ T_{23} \\ T_{34} \\ T_{45} \end{Bmatrix} = \begin{Bmatrix} 382 \\ 676 \\ 462 \\ 542 \\ 792 \end{Bmatrix}$$

$a = 8.15$
$T_{12} = -25.71$
$T_{23} = -2.06$
$T_{34} = -29.31$
$T_{45} = -58.11$

9.13 $A = \begin{bmatrix} 2 & 4 \\ 0 & 2 \end{bmatrix}$ $B = \begin{bmatrix} 2 & 0 \\ -1 & 0 \end{bmatrix}$

$u = \begin{bmatrix} 1 \\ 3 \end{bmatrix}$ $v = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$

$$\begin{aligned} 2x_1 + 4x_2 - 2y_1 \quad &= 1 \\ 2x_2 - y_1 \quad &= 3 \\ 2x_1 \quad + 2y_1 + 4y_2 &= 1 \\ -x_1 \quad + 2y_2 &= 0 \end{aligned}$$

$x_1 = -2.5$ $y_1 = 5.5$
$x_2 = 4.25$ $y_2 = -1.25$

$z_1 = -2.5 + 5.5j$ $z_2 = 4.25 - 1.25j$

9.14

Here is a VBA program to implement matrix multiplication and solve Prob. 9.3 for the case of $[X]\times[Y]$.

```
Option Explicit

Sub Mult()

Dim i As Integer, j As Integer
Dim l As Integer, m As Integer, n As Integer
Dim x(10, 10) As Single, y(10, 10) As Single
Dim w(10, 10) As Single

l = 2
m = 2
n = 3
x(1, 1) = 1: x(1, 2) = 6
x(2, 1) = 3: x(2, 2) = 10
x(3, 1) = 7: x(3, 2) = 4
y(1, 1) = 6: y(2, 1) = 0
y(2, 1) = 1: y(2, 2) = 4
Call Mmult(x(), y(), w(), m, n, l)

For i = 1 To n
  For j = 1 To l
    MsgBox w(i, j)
 Next j
Next i

End Sub

Sub Mmult(y, z, x, n, m, p)

Dim i As Integer, j As Integer, k As Integer
Dim sum As Single

For i = 1 To m
  For j = 1 To p
    sum = 0
    For k = 1 To n
      sum = sum + y(i, k) * z(k, j)
    Next k
    x(i, j) = sum
  Next j
Next i

End Sub
```

9.15

Here is a VBA program to implement the matrix transpose and solve Prob. 9.3 for the case of $[X]^T$.

```
Option Explicit

Sub Mult()

Dim i As Integer, j As Integer
Dim m As Integer, n As Integer
Dim x(10, 10) As Single, y(10, 10) As Single

n = 3
m = 2
x(1, 1) = 1: x(1, 2) = 6
x(2, 1) = 3: x(2, 2) = 10
```

```
x(3, 1) = 7: x(3, 2) = 4
Call MTrans(x(), y(), n, m)
For i = 1 To m
  For j = 1 To n
    MsgBox y(i, j)
  Next j
Next i

End Sub

Sub MTrans(a, b, n, m)

Dim i As Integer, j As Integer

For i = 1 To m
  For j = 1 To n
    b(i, j) = a(j, i)
  Next j
Next i

End Sub
```

9.16

Here is a VBA program to implement the Gauss elimination algorithm and solve the test case in Prob. 9.16.

```
Option Explicit

Sub GaussElim()

Dim n As Integer, er As Integer, i As Integer
Dim a(10, 10) As Single, b(10) As Single, x(10) As Single

Range("a1").Select
n = 3
a(1, 1) = 1: a(1, 2) = 1: a(1, 3) = -1
a(2, 1) = 6: a(2, 2) = 2: a(2, 3) = 2
a(3, 1) = -3: a(3, 2) = 4: a(3, 3) = 1
b(1) = 1: b(2) = 10: b(3) = 2

Call Gauss(a(), b(), n, x(), er)
If er = 0 Then
  For i = 1 To n
    MsgBox "x(" & i & ") = " & x(i)
  Next i
Else
  MsgBox "ill-conditioned system"
End If

End Sub

Sub Gauss(a, b, n, x, er)

Dim i As Integer, j As Integer
Dim s(10) As Single
Const tol As Single = 0.000001
er = 0
For i = 1 To n
  s(i) = Abs(a(i, 1))
  For j = 2 To n
    If Abs(a(i, j)) > s(i) Then s(i) = Abs(a(i, j))
  Next j
Next i
Call Eliminate(a, s(), n, b, tol, er)
If er <> -1 Then
```

```
  Call Substitute(a, n, b, x)
End If
End Sub

Sub Pivot(a, b, s, n, k)
Dim p As Integer, ii As Integer, jj As Integer
Dim factor As Single, big As Single, dummy As Single
p = k
big = Abs(a(k, k) / s(k))
For ii = k + 1 To n
  dummy = Abs(a(ii, k) / s(ii))
  If dummy > big Then
    big = dummy
    p = ii
  End If
Next ii
If p <> k Then
  For jj = k To n
    dummy = a(p, jj)
    a(p, jj) = a(k, jj)
    a(k, jj) = dummy
  Next jj
  dummy = b(p)
  b(p) = b(k)
  b(k) = dummy
  dummy = s(p)
  s(p) = s(k)
  s(k) = dummy
End If
End Sub

Sub Substitute(a, n, b, x)
Dim i As Integer, j As Integer
Dim sum As Single
x(n) = b(n) / a(n, n)
For i = n - 1 To 1 Step -1
  sum = 0
  For j = i + 1 To n
    sum = sum + a(i, j) * x(j)
  Next j
  x(i) = (b(i) - sum) / a(i, i)
Next i
End Sub

Sub Eliminate(a, s, n, b, tol, er)

Dim i As Integer, j As Integer, k As Integer
Dim factor As Single
For k = 1 To n - 1
  Call Pivot(a, b, s, n, k)
  If Abs(a(k, k) / s(k)) < tol Then
    er = -1
    Exit For
  End If
  For i = k + 1 To n
    factor = a(i, k) / a(k, k)
    For j = k + 1 To n
      a(i, j) = a(i, j) - factor * a(k, j)
    Next j
    b(i) = b(i) - factor * b(k)
  Next i
Next k
If Abs(a(k, k) / s(k)) < tol Then er = -1
End Sub
```

It’s application yields a solution of (1, 1, 1).

## Chapter 10

**10.1** Matrix multiplication is distributive

$$[L]\{[U]\{X\}-\{D\}\}=[A]\{X\}-\{C\}$$

$$[L][U]\{X\}-[L]\{D\}=[A]\{X\}-\{C\}$$

$$\therefore\ [L][U]\{X\}=[A]\{X\}$$

$$[L]\{D\}=\{C\}$$
$$[L][U]=[A]$$

**10.2**

$$\begin{bmatrix} 7 & 2 & -3 \\ 2 & 5 & -3 \\ 1 & -1 & -6 \end{bmatrix}$$

First step in forward elimination yields

$$\begin{matrix} 7 & 2 & -3 \\ 0 & 4.429 & -2.143 \\ 0 & -1.286 & -5.571 \end{matrix}$$

$$f_{21}=2/7=0.2857$$
$$f_{31}=1/7=0.1429$$
$$f_{32}=\frac{-1.286}{4.428}=-0.29$$

$\therefore$

$$[L]=\begin{bmatrix} 1 & 0 & 0 \\ 0.2857 & 1 & 0 \\ 0.1429 & -0.29 & 1 \end{bmatrix}$$

final forward elimination yields

$$[U]=\begin{bmatrix} 7 & 2 & -3 \\ 0 & 4.429 & -2.143 \\ 0 & 0 & -6.194 \end{bmatrix}$$

$$[L][U]=\begin{bmatrix} 7 & 2 & -3 \\ 2 & 5 & -3 \\ 1 & -0.999 & -6.001 \end{bmatrix}$$

with some roundoff

**10.3** $[L]\{D\}=\{B\}$

$$\begin{bmatrix} 1 & 0 & 0 \\ 0.2857 & 1 & 0 \\ 0.1429 & -0.29 & 1 \end{bmatrix}\begin{Bmatrix} d_1 \\ d_2 \\ d_3 \end{Bmatrix}=\begin{Bmatrix} -12 \\ -20 \\ -26 \end{Bmatrix}$$

Solving gives

$$\{D\}=\begin{Bmatrix} -12 \\ -16.572 \\ -29.091 \end{Bmatrix}$$

now solving

$$[U]\{X\}=\{D\}$$

$$\begin{bmatrix} 7 & 2 & -3 \\ 0 & 4.429 & -2.143 \\ 0 & 0 & -6.194 \end{bmatrix}\cdot\begin{Bmatrix} x_1 \\ x_2 \\ x_3 \end{Bmatrix}=\begin{Bmatrix} -12 \\ -16.572 \\ -29.091 \end{Bmatrix}$$

$$x_1=0.718$$
$$x_2=-1.469$$
$$x_3=4.697$$

for $B^T = [12, 18, -6]$

solving $[L]\{D\} = \{B\}$

gives $\{D\} = \begin{bmatrix} 12 \\ 14.572 \\ -3.489 \end{bmatrix}$

and solving $[U]\{X\} = \{D\}$

gives
$$x_1 = 0.938$$
$$x_2 = 3.563$$
$$x_3 = 0.563$$

10.4 use $[L]\{D\} = \{B\}$

1st column of $A^{-1}$

$$\begin{bmatrix} 1 & 0 & 0 \\ 0.2857 & 1 & 0 \\ 0.1429 & -0.29 & 1 \end{bmatrix} \begin{Bmatrix} d_1 \\ d_2 \\ d_3 \end{Bmatrix} = \begin{Bmatrix} 1 \\ 0 \\ 0 \end{Bmatrix}$$

gives $\{D\}^T = \lfloor 1 \;\; -0.286 \;\; -0.226 \rfloor$

and $[U]\{X\} = \{D\}$

$$\begin{bmatrix} 7 & 2 & -3 \\ 0 & 4.429 & -2.143 \\ 0 & 0 & -6.194 \end{bmatrix} \cdot \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \end{Bmatrix} = \begin{Bmatrix} 1 \\ -0.286 \\ -0.226 \end{Bmatrix}$$

$$\left.\begin{aligned} x_1 &= 0.172 \\ x_2 &= -0.047 \\ x_3 &= 0.036 \end{aligned}\right\} \text{1st column of } A^{-1}$$

for second column of $A^{-1}$

use $\{B\}^T = \{0 \;\; 1 \;\; 0\}$

which gives

$\{D\}^T = \{0 \quad 1 \quad 0.29\}$

$\{X\}^T = \{-0.078 \quad 0.203 \quad -0.047\}$

for 3rd column of $A^{-1}$

use $\{B\}^T = \{0 \;\; 0 \;\; 1\}$

which gives

$\{D\}^T = \{0 \quad 0 \quad 1\}$

$\{X\}^T = \{-0.047 \quad -0.078 \quad -0.161\}$

$\therefore$

$$A^{-1} = \begin{bmatrix} 0.172 & -0.078 & -0.047 \\ -0.047 & 0.203 & -0.078 \\ 0.036 & -0.047 & -0.161 \end{bmatrix}$$

$$A\,A^{-1} = I$$

10.5
$$\begin{matrix} 1 & 7 & -4 & -51 \\ 4 & -4 & 9 & 62 \\ 12 & -1 & 3 & 8 \end{matrix}$$

Pivot

$$\begin{matrix} 12 & -1 & 3 & 8 \\ 4 & -4 & 9 & 62 \\ 1 & 7 & -4 & -51 \end{matrix}$$

Now forward eliminate on A

$f_{21} = 4/12$

$f_{31} = 1/12$

$$\begin{matrix} 12 & -1 & 3 \\ 0 & -3.667 & 8 \\ 0 & 7.083 & -4.25 \end{matrix}$$

Pivot again

$$\begin{matrix} 12 & -1 & 3 \\ 0 & 7.083 & -4.25 \\ 0 & -3.667 & 8 \end{matrix} \quad \{B\} = \begin{Bmatrix} 8 \\ -51 \\ 62 \end{Bmatrix}$$

$f_{21} = 1/12 \qquad f_{31} = 4/12$

$f_{32} = \dfrac{-3.667}{7.083} = -0.51765$

$$[L] = \begin{bmatrix} 1 & 0 & 0 \\ 0.0833 & 1 & 0 \\ 0.3333 & -0.51765 & 1 \end{bmatrix}$$

$$[U] = \begin{bmatrix} 12 & -1 & 3 \\ 0 & 7.083 & -4.25 \\ 0 & 0 & 5.8 \end{bmatrix}$$

$[L]\{D\} = \{B\}$ gives

$\{D\}^T = 8 \quad -51.667 \quad 32.591$

and finally

$[U]\{X\} = \{D\}$ gives

$x_1 = -1.065$
$x_2 = -3.923$
$x_3 = 5.619$

10.6

$$\begin{matrix} -5 & 0 & 12 \\ 4 & -1 & -1 \\ 1 & -2 & 12 \end{matrix}$$

forward eliminate with

$f_{21} = -4/5 \qquad f_{31} = -1/5$

gives

$$\begin{matrix} -5 & 0 & 12 \\ 0 & -1 & 8.6 \\ 0 & -2 & 14.4 \end{matrix}$$

Pivot

$$\begin{matrix} -5 & 0 & 12 \\ 0 & -2 & 14.4 \\ 0 & -1 & 8.6 \end{matrix}$$ with

$f_{21} = -\frac{1}{5}, \quad f_{31} = -4/5$

$f_{32} = -1/-2 = 0.5 \qquad \{B\} = \begin{Bmatrix} 60 \\ -86 \\ -2 \end{Bmatrix}$

gives

$$L = \begin{bmatrix} 1 & 0 & 0 \\ -0.2 & 1 & 0 \\ -0.8 & 0.5 & 1 \end{bmatrix}$$

$$U = \begin{bmatrix} -5 & 0 & 12 \\ 0 & -2 & 14.4 \\ 0 & 0 & 1.4 \end{bmatrix}$$

$[L]\{D\} = \{B\}$ gives

$\{D\}^T = 60 \quad -74 \quad 83$

and $[U]\{X\} = \{D\}$ gives

$x_1 = 130.286$
$x_2 = 463.857$
$x_3 = 59.286$

10.7

$$A^{-1} = \begin{matrix} 1 & 1.7143 & -0.8571 \\ 3.5 & 5.143 & -3.071 \\ 0.5 & 0.7143 & -0.357 \end{matrix}$$

$$\{X\} = A^{-1}\{b\} = \begin{Bmatrix} 130.286 \\ 463.857 \\ 59.286 \end{Bmatrix}$$

where $\{b\} = \begin{Bmatrix} 60 \\ -2 \\ -86 \end{Bmatrix}$,

for $\{b\} = \begin{Bmatrix} 110 \\ 55 \\ -105 \end{Bmatrix}$

$$\{X\} = \begin{Bmatrix} 294.286 \\ 990.357 \\ 131.786 \end{Bmatrix}$$

10.8 $\quad l_{11} = 2 \quad l_{21} = -1 \quad l_{31} = 3$

$$u_{12} = \frac{a_{12}}{l_{11}} = -2.5$$

$$u_{13} = \frac{a_{13}}{l_{11}} = 0.5$$

$$l_{22} = a_{23} - l_{21}u_{12}$$
$$= 3 - (-1)(-2.5) = 0.5$$

$$l_{32} = a_{32} - l_{31}u_{12}$$
$$= -4 - (3)(-2.5) = 3.5$$

$$u_{23} = \frac{a_{23} - l_{21}u_{13}}{l_{22}}$$
$$= \frac{-1 - (-1)(0.5)}{0.5} = -1$$

$$l_{33} = a_{33} - l_{31}u_{13} - l_{32}u_{23}$$
$$= 2 - 3(0.5) - 3.5(-1) = 4$$

$\therefore$

$$[L] = \begin{matrix} 2 & 0 & 0 \\ -1 & 0.5 & 0 \\ 3 & 3.5 & 4 \end{matrix}$$

$$[U] = \begin{matrix} 1 & -2.5 & 0.5 \\ 0 & 1 & -1 \\ 0 & 0 & 1 \end{matrix}$$

$L \cdot U$ = original matrix

10.9 using Mathcad

a)

$$A^{-1} = \begin{bmatrix} 0.0641 & 0.00836 & 0.00950 \\ 0.0170 & 0.0509 & 0.00695 \\ 0.0184 & 0.0135 & 0.0492 \end{bmatrix}$$

b) $\quad A^{-1} \cdot \{b\} = \{x\}$

where $\{b\} = \begin{Bmatrix} 500 \\ 200 \\ 30 \end{Bmatrix}$

gives $\{c\} = \begin{Bmatrix} 33.996 \\ 18.893 \\ 13.384 \end{Bmatrix}$

c) $\quad i = 1 \quad j = 3 \quad$ we want

$$(\bar{a}'_{13})(b_3) = 5$$

or $b_3 = 526.4$

solving for $\{c\}$ with

$$\{b\} = \begin{Bmatrix} 500 \\ 200 \\ 556.4 \end{Bmatrix}$$

gives $$\{c\} = \begin{Bmatrix} 38.996 \\ 22.549 \\ 39.278 \end{Bmatrix}$$

d) $$a_{31}^{-1}(b_1) = \Delta c_1$$

$$0.01843(50) = 0.9215$$

$$a_{32}^{-1}(b_2) = \Delta c_1$$

$$0.0135(100) = 1.35$$

or total $\Delta c_1 = 2.27$

check by solving for $\{c\}$ with

$$\{b\} = \begin{Bmatrix} 450 \\ 100 \\ 30 \end{Bmatrix}$$

gives $$\{c\} = \begin{Bmatrix} 29.956 \\ 12.953 \\ 11.116 \end{Bmatrix}$$

Note that $13.384 - 2.27$

$$= 11.114 \approx 11.116$$

as expected

10.10 Scale so max element in each row is one.

$$A = \begin{matrix} 1 & -0.333 & 0.833 \\ 1 & 0.129 & 0.294 \\ 0.155 & 0.0971 & 1 \end{matrix}$$

$$\sum_i \sum_j a_{ij}^2 = 3.94$$

$$\|A\|_e = \sqrt{3.94} = 1.985$$

| column | $\Sigma \lvert a \rvert$ |
|---|---|
| 1 | 2.155 |
| 2 | 0.559 |
| 3 | 2.127 |

$$\|A\|_1 = 2.155$$

| row | $\Sigma \lvert a \rvert$ |
|---|---|
| 1 | 2.166 |
| 2 | 1.423 |
| 3 | 1.252 |

$$\|A\|_\infty = 2.166$$

10.11 Following procedure of Prob 10.10 without scaling gives

$$\|A\|_e = 18.25$$
$$\|A\|_\infty = 17$$

$$\|A\|_e = 18.33$$
$$\|A\|_\infty = 17$$

10.12

$$A^{-1} = \begin{bmatrix} -1.9\times10^{14} & 5.6\times10^{14} \\ 5.6\times10^{14} & -1.7\times10^{15} \\ -5.6\times10^{14} & 1.7\times10^{15} \\ 1.9\times10^{14} & -5.6\times10^{14} \end{bmatrix}$$

using math cad,

$$A^{-1} = \begin{bmatrix} 17.64 & -69.5 & 95 & -42.5 \\ -133.8 & 680.0 & -1030 & 483.9 \\ 269.8 & -1523 & 2442 & -1188 \\ -158.3 & 946.5 & -1574 & 785.3 \end{bmatrix}$$

$$\|A\|_\infty = 3.04$$

$$\|A^{-1}\|_\infty = 5423$$

$$\text{Cond } A = 1647$$

$$\log_{10}(1647) = 3.2$$

therefore 3 or 4 significant figures may be lost

10.14

```
Option Explicit

Sub LUDTest()
Dim n As Integer, er As Integer, i As Integer, j As Integer
Dim a(3, 3) As Single, b(3) As Single, x(3) As Single
Dim tol As Single

n = 3
a(1, 1) = 3: a(1, 2) = -0.1: a(1, 3) = -0.2
a(2, 1) = 0.1: a(2, 2) = 7: a(2, 3) = -0.3
a(3, 1) = 0.3: a(3, 2) = -0.2: a(3, 3) = 10
b(1) = 7.85: b(2) = -19.3: b(3) = 71.4
tol = 0.000001

Call LUD(a(), b(), n, x(), tol, er)

'output results to worksheet
Sheets("Sheet1").Select
Range("a3").Select
For i = 1 To n
  ActiveCell.Value = x(i)
  ActiveCell.Offset(1, 0).Select
```

```
Next i
Range("a3").Select

End Sub

Sub LUD(a, b, n, x, tol, er)
Dim i As Integer, j As Integer
Dim o(3) As Single, s(3) As Single
Call Decompose(a, n, tol, o(), s(), er)
If er = 0 Then
  Call Substitute(a, o(), n, b, x)
Else
  MsgBox "ill-conditioned system"
  End
End If
End Sub

Sub Decompose(a, n, tol, o, s, er)
Dim i As Integer, j As Integer, k As Integer
Dim factor As Single
For i = 1 To n
  o(i) = i
  s(i) = Abs(a(i, 1))
  For j = 2 To n
    If Abs(a(i, j)) > s(i) Then s(i) = Abs(a(i, j))
  Next j
Next i
For k = 1 To n - 1
  Call Pivot(a, o, s, n, k)
  If Abs(a(o(k), k) / s(o(k))) < tol Then
    er = -1
    Exit For
  End If
 For i = k + 1 To n
    factor = a(o(i), k) / a(o(k), k)
    a(o(i), k) = factor
    For j = k + 1 To n
      a(o(i), j) = a(o(i), j) - factor * a(o(k), j)
    Next j
  Next i
Next k
If (Abs(a(o(k), k) / s(o(k))) < tol) Then er = -1
End Sub

Sub Pivot(a, o, s, n, k)
Dim ii As Integer, p As Integer
Dim big As Single, dummy As Single
p = k
big = Abs(a(o(k), k) / s(o(k)))
For ii = k + 1 To n
  dummy = Abs(a(o(ii), k) / s(o(ii)))
  If dummy > big Then
    big = dummy
    p = ii
  End If
Next ii
dummy = o(p)
o(p) = o(k)
o(k) = dummy
End Sub

Sub Substitute(a, o, n, b, x)
Dim k As Integer, i As Integer, j As Integer
Dim sum As Single, factor As Single
For k = 1 To n - 1
  For i = k + 1 To n
    factor = a(o(i), k)
```

```
    b(o(i)) = b(o(i)) - factor * b(o(k))
  Next i
Next k
x(n) = b(o(n)) / a(o(n), n)
For i = n - 1 To 1 Step -1
  sum = 0
  For j = i + 1 To n
    sum = sum + a(o(i), j) * x(j)
  Next j
  x(i) = (b(o(i)) - sum) / a(o(i), i)
Next i
End Sub
```

10.15

```
Option Explicit

Sub LUGaussTest()
Dim n As Integer, er As Integer, i As Integer, j As Integer
Dim a(3, 3) As Single, b(3) As Single, x(3) As Single
Dim tol As Single, ai(3, 3) As Single
n = 3
a(1, 1) = 3: a(1, 2) = -0.1: a(1, 3) = -0.2
a(2, 1) = 0.1: a(2, 2) = 7: a(2, 3) = -0.3
a(3, 1) = 0.3: a(3, 2) = -0.2: a(3, 3) = 10
tol = 0.000001
Call LUDminv(a(), b(), n, x(), tol, er, ai())
If er = 0 Then
  Range("a1").Select
  For i = 1 To n
    For j = 1 To n
      ActiveCell.Value = ai(i, j)
      ActiveCell.Offset(0, 1).Select
    Next j
    ActiveCell.Offset(1, -n).Select
  Next i
  Range("a1").Select
Else
  MsgBox "ill-conditioned system"
End If
End Sub


Sub LUDminv(a, b, n, x, tol, er, ai)
Dim i As Integer, j As Integer
Dim o(3) As Single, s(3) As Single
Call Decompose(a, n, tol, o(), s(), er)
If er = 0 Then
  For i = 1 To n
    For j = 1 To n
      If i = j Then
        b(j) = 1
      Else
        b(j) = 0
      End If
    Next j
    Call Substitute(a, o, n, b, x)
    For j = 1 To n
      ai(j, i) = x(j)
    Next j
  Next i
End If
End Sub

Sub Decompose(a, n, tol, o, s, er)
Dim i As Integer, j As Integer, k As Integer
```

```
Dim factor As Single
For i = 1 To n
  o(i) = i
  s(i) = Abs(a(i, 1))
  For j = 2 To n
    If Abs(a(i, j)) > s(i) Then s(i) = Abs(a(i, j))
  Next j
Next i
For k = 1 To n - 1
  Call Pivot(a, o, s, n, k)
  If Abs(a(o(k), k) / s(o(k))) < tol Then
    er = -1
    Exit For
  End If
  For i = k + 1 To n
    factor = a(o(i), k) / a(o(k), k)
    a(o(i), k) = factor
    For j = k + 1 To n
      a(o(i), j) = a(o(i), j) - factor * a(o(k), j)
    Next j
  Next i
Next k
If (Abs(a(o(k), k) / s(o(k))) < tol) Then er = -1
End Sub

Sub Pivot(a, o, s, n, k)
Dim ii As Integer, p As Integer
Dim big As Single, dummy As Single
p = k
big = Abs(a(o(k), k) / s(o(k)))
For ii = k + 1 To n
  dummy = Abs(a(o(ii), k) / s(o(ii)))
  If dummy > big Then
    big = dummy
    p = ii
  End If
Next ii
dummy = o(p)
o(p) = o(k)
o(k) = dummy
End Sub

Sub Substitute(a, o, n, b, x)
Dim k As Integer, i As Integer, j As Integer
Dim sum As Single, factor As Single
For k = 1 To n - 1
  For i = k + 1 To n
    factor = a(o(i), k)
    b(o(i)) = b(o(i)) - factor * b(o(k))
  Next i
Next k
x(n) = b(o(n)) / a(o(n), n)
For i = n - 1 To 1 Step -1
  sum = 0
  For j = i + 1 To n
    sum = sum + a(o(i), j) * x(j)
  Next j
  x(i) = (b(o(i)) - sum) / a(o(i), i)
Next i
End Sub
```

10.16 Solve

$2\Delta x_1 + 4\Delta x_2 + \Delta x_3 = -5-(-4)$
$5\Delta x_1 + 2\Delta x_2 + \Delta x_3 = 12-12=0$
$\Delta x_1 + 2\Delta x_2 + \Delta x_3 = 3-4$

gives $\Delta x_1 = 0.25$
$\Delta x_2 = -0.125$
$\Delta x_3 = -1$

and $x_1 = 2+0.25 = 2.25$
$x_2 = -5-0.125 = -5.125$
$x_3 = 12-1 = 11$
exact

10.17

$$\vec{A}\cdot\vec{B} = 0 \Rightarrow -4a+2b = 3 \quad (1)$$
$$\vec{A}\cdot\vec{C} = 0 \Rightarrow 2a-3c = -6 \quad (2)$$
$$\vec{B}\cdot\vec{C} = 2 \Rightarrow 3b+c = 10 \quad (3)$$

Solve the three equations using Matlab:

```
>> A=[-4 2 0; 2 0 -3; 0 3 1]
b=[3; -6; 10]
x=inv(A)*b

x = 0.525
    2.550
    2.350
```

Therefore, $a = 0.525$, $b = 2.550$, and $c = 2.350$.

10.18

$$(\vec{A}\times\vec{B}) = \begin{vmatrix} \vec{i} & \vec{j} & \vec{k} \\ a & b & c \\ -2 & 1 & -4 \end{vmatrix} = (-4b-c)\vec{i} - (-4a+2c)\vec{j} + (a+2b)\vec{k}$$

$$(\vec{A}\times\vec{C}) = \begin{vmatrix} \vec{i} & \vec{j} & \vec{k} \\ a & b & c \\ 1 & 3 & 2 \end{vmatrix} = (2b-3c)\vec{i} - (2a-c)\vec{j} + (3a+b)\vec{k}$$

$$(\vec{A}\times\vec{B}) + (\vec{A}\times\vec{C}) = (-2b-4c)\vec{i} - (-2a+c)\vec{j} + (4a+b)\vec{k}$$

Therefore,

$$(-2b-4c)\vec{i} + (-2a-c)\vec{j} + (4a+b)\vec{r} = (5a+6)\vec{i} + (3b-2)\vec{j} + (-4c+1)\vec{k}$$

We get the following set of equations $\Rightarrow$

$$-2b-4c = 5a+6 \quad \Rightarrow \quad -5a-2b-4c = 6 \qquad (1)$$

$$2a-c = 3b-2 \quad \Rightarrow \quad 2a-3b-c = -2 \qquad (2)$$

$$4a+b = -4c+1 \quad \Rightarrow \quad 4a+b-4c = 1 \qquad (3)$$

In Matlab:

```
A=[-5  -5  -4 ; 2  -3  -1 ; 4  1  -4]
B=[ 6 ; -2 ; 1] ; x = inv (A) * b
```

Answer $\Rightarrow$
```
x  = -3.6522
     -3.3478
  4.7391
```

$a$ = -3.6522, $b$ = -3.3478, $c$ = 4.7391

10.19

(I) $f(0) = 1 \Rightarrow a(0) + b = 1 \Rightarrow b = 1$

$f(2) = 1 \Rightarrow c(2) + d = 1 \Rightarrow 2c + d = 1$

(II) If $f$ is continuous, then at $x$ = 1

$$ax + b = cx + d \Rightarrow a(1) + b = c(1) + d \Rightarrow a + b - c - d = 0$$

(III) $a + b = 4$

$$\begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 2 & 2 \\ 1 & 1 & -1 & -1 \\ 1 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 4 \end{bmatrix}$$

Solve using Matlab $\Rightarrow$

$\boxed{\begin{array}{l} a = 3 \\ b = 1 \\ c = \text{-}3 \\ d = 7 \end{array}}$

10.20 MATLAB provides a handy way to solve this problem.

(*a*)

```
>> a=hilb(3)

a =
    1.0000    0.5000    0.3333
    0.5000    0.3333    0.2500
    0.3333    0.2500    0.2000

>> b=[1 1 1]'

b =
     1
     1
     1

>> c=a*b

c =
    1.8333
    1.0833
    0.7833

>> format long e

>> x=a\b

>> x =

    9.999999999999991e-001
    1.000000000000007e+000
    9.999999999999926e-001
```

(*b*)

```
>> a=hilb(7);
>> b=[1 1 1 1 1 1 1]';
>> c=a*b;
>> x=a\b
x =

    9.999999999914417e-001
    1.000000000344746e+000
    9.999999966568566e-001
    1.000000013060454e+000
    9.999999759661609e-001
    1.000000020830062e+000
    9.999999931438059e-001
```

(*c*)

```
>> a=hilb(10);
>> b=[1 1 1 1 1 1 1 1 1 1]';
>> c=a*b;
>> x=a\b

x =
    9.999999987546324e-001
    1.000000107466305e+000
    9.999977129981819e-001
    1.000020777695979e+000
    9.999009454847158e-001
    1.000272183037448e+000
    9.995535966572223e-001
    1.000431255894815e+000
    9.997736605804316e-001
    1.000049762292970e+000
```

## Chapter 11

11.1

$e_2 = -1/2 = -0.5$
$f_2 = 2 - (-0.5)(-1) = 1.50$

$e_3 = -1/1.5 = -0.667$
$f_3 = 2 - (-0.667)(-1) = 1.333$

Transformed system is

$$\begin{bmatrix} 2 & -1 & 0 \\ -0.5 & 1.5 & -1 \\ 0 & -0.667 & 1.333 \end{bmatrix}$$

which is decomposed as

$$L = \begin{bmatrix} 1 & 0 & 0 \\ -0.5 & 1 & 0 \\ 0 & -0.667 & 1 \end{bmatrix}$$

$$U = \begin{bmatrix} 2 & -1 & 0 \\ 0 & 1.5 & -1 \\ 0 & 0 & 1.333 \end{bmatrix}$$

The right hand side becomes

$r_2 = 4 - (-0.5)(124) = 66$
$r_3 = 14 - (-0.667)(66) = 58.$

$$\{b\} = \begin{Bmatrix} 124 \\ 66 \\ 58 \end{Bmatrix}$$

and $[U]\{X\} = \{b\}$ gives

$x_1 = 98.5$
$x_2 = 73.0$
$x_3 = 43.5$

11.2 use $[L]\{D\} = \begin{Bmatrix} 1 \\ 0 \\ 0 \end{Bmatrix}$

for first col of $A^{-1}$.

Gives $\{D\} = \begin{Bmatrix} 1 \\ 0.5 \\ 0.334 \end{Bmatrix}$

and $[U]\{X\} = \{D\}$

gives $\left.\begin{array}{l} x_1 = 0.75 \\ x_2 = 0.5 \\ x_3 = 0.25 \end{array}\right\}$ 1st col of $A^{-1}$

for 2nd column of $A^{-1}$

use $\{D\} = \begin{Bmatrix} 0 \\ 1 \\ 0 \end{Bmatrix}$

which gives

$x_1 = 0.5$
$x_2 = 1.0$
$x_3 = 0.5$

for 3rd column of $A^{-1}$

use $\{D\} = \begin{Bmatrix} 0 \\ 0 \\ 1 \end{Bmatrix}$

which gives

$x_1 = 0.25$
$x_2 = 0.50$
$x_3 = 0.75$

$$A^{-1} = \begin{matrix} 0.75 & 0.5 & 0.25 \\ 0.5 & 1 & 0.5 \\ 0.25 & 0.5 & 0.75 \end{matrix}$$

11.3 Following Prob 11.1 and Figure 11.2 gives

$$e_2 = -0.02875/2.01475 = -0.01427$$
$$f_2 = 2.01434$$

$$e_3 = -0.0142727$$
$$f_3 = 2.01434$$

$$e_4 = -0.0142727$$
$$f_4 = 2.01434$$

$$r_1 = 4.175$$
$$r_2 = 0.01521$$
$$r_3 = 0.02979$$
$$r_4 = 2.0726$$

and

$$T_4 = 1.036$$
$$T_3 = 0.0152$$
$$T_2 = 0.0298$$
$$T_1 = 2.0726$$

11.4

$$\begin{bmatrix} 2.4495 & 6.1237 & 22.454 \\ & 4.1833 & 20.916 \\ & & 6.1106 \end{bmatrix}$$

$$\begin{bmatrix} 2.4495 & & \\ 6.1237 & 4.1833 & \\ 22.454 & 20.916 & 6.1106 \end{bmatrix} \begin{bmatrix} 6 & 15 & 55 \\ 15 & 55 & 225 \\ 55 & 225 & 979 \end{bmatrix}$$

11.5 Following Ex 11.2 Cholesky yields

$$L = \begin{bmatrix} 2.4495 & 0 & 0 \\ 6.7361 & 5.5565 & 0 \\ 5.7155 & 1.710 & 4.291 \end{bmatrix}$$

multiplying $L L^T$

gives original symmetric matrix

11.6

$$x_1 = \frac{124 + x_2}{2}$$

$$x_2 = \frac{4 + x_1 + x_3}{2}$$

$$x_3 = \frac{14 + x_2}{2}$$

use initial guess

$x_1 = 62$, $x_2 = 2$, $x_3 = 7$

1st iteration gives

$$x_1 = \frac{124+2}{2} = 63$$

$$x_2 = \frac{4+63+7}{2} = 37$$

$$x_3 = \frac{14+37}{2} = 25.5$$

fifth iteration gives

$x_1 = 96.25$ $\quad \epsilon_a = 2.3\%$
$x_2 = 70.75$ $\quad \epsilon_a = 3.2\%$
$x_3 = 42.38$ $\quad \epsilon_a = 2.6\%$

11.7

$c_1 = (500 + 2c_2 + 3c_3)/17$
$c_2 = (200 + 5c_1 + 2c_3)/21$
$c_3 = (30 + 5c_1 + 5c_2)/22$

try $c_1 = 0$, $c_2 = 0$, $c_3 = 0$

1st iteration

$c_1 = 29.41$
$c_2 = 16.53$ etc
$c_3 = 11.80$

3rd iteration

$c_1 = 33.93$ $\epsilon_a = 1.4\%$
$c_2 = 18.86$ $\epsilon_a = 1.3\%$
$c_3 = 13.36$ $\epsilon_a = 1.3\%$

11.8 try $c_1 = 0$ $c_2 = 0$ $c_3 = 0$

$$\left.\begin{aligned} c_1 &= 500/17 = 29.41 \\ c_2 &= 200/21 = 9.52 \\ c_3 &= 30/22 = 1.36 \end{aligned}\right\} 1^{st}$$

$$\left.\begin{aligned} c_1 &= (500 + 2(9.52) + 3(1.36))/17 = 30.77 \\ c_2 &= (200 + 5(29.41) + 2(1.36))/21 = 16.66 \\ c_3 &= (30 + 5(29.41) + 5(9.52))/22 = 10.21 \end{aligned}\right\} 2^{nd}$$

etc

$$\left.\begin{aligned} c_1 &= 33.88 \quad \epsilon_a = 0.7\% \\ c_2 &= 18.77 \quad \epsilon_a = 1.0\% \\ c_3 &= 13.23 \quad \epsilon_a = 2.1\% \end{aligned}\right\} 5^{th}$$

11.9 Rearrange

$x_1 = (-2 + x_2 + x_3)/4$
$x_2 = (45 - 6x_1)/8$
$x_3 = (80 + 5x_1)/12$

assume $x_1 = x_2 = x_3 = 0$

First iteration

$x_1^{new} = -2/4 = -0.5$

$x_1 = 0.9(-0.5) + (1-0.9)(0)$
$= -0.45$

$x_2^{new} = (45 - 6(-0.45))/8 = 5.96$

$x_2 = 0.9(5.96) + (1-0.9)(0) = 5.36$

$x_3^{new} = (80 + 5(-0.45))/12 = 6.48$

$x_3 = 0.9(6.48) + (1-0.9)(0) = 5.83$

Second Iteration

$x_1^{new} = (-2 + 5.36 + 5.83)/4 = 2.298$

$x_1 = 0.9(2.298) + 0.1(-0.45) = 2.02$

$x_2^{new} = (45 - 6(2.02))/8 = 4.11$

$x_2 = 0.9(4.11) + 0.1(5.36) = 4.235$

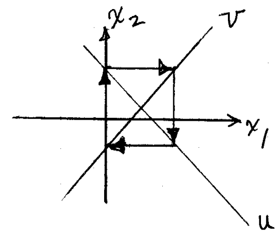
$x_3^{new} = (80 + 5(2.02))/12 = 7.51$

$x_3 = 0.9(7.51) + 0.1(5.83) = 7.34$

etc

$$\left.\begin{aligned} x_1 &= 2.375 \\ x_2 &= 3.844 \\ x_3 &= 7.656 \end{aligned}\right\} \text{final values}$$

11.10 Diverges




11.11

a) use lsolve from Mathcad
all $x$'s = 1 for both (i) and (ii)

b)

(i) $$A^{-1} = \begin{bmatrix} 3.875 & -5.5 & 2.125 \\ -5.5 & 7 & -2.5 \\ 2.125 & -2.5 & 0.875 \end{bmatrix}$$

(ii) $$A^{-1} = \begin{bmatrix} -1.9\times10^{14} & 5.7\times10^{14} & -5.7\times10^{14} & 1.9\times10^{14} \\ 5.6\times10^{14} & -1.7\times10^{15} & 1.7\times10^{15} & -5.7\times10^{14} \\ -5.6\times10^{14} & 1.7\times10^{15} & -1.7\times10^{15} & 5.7\times10^{14} \\ 1.9\times10^{14} & -5.7\times10^{14} & 5.7\times10^{14} & -1.9\times10^{14} \end{bmatrix}$$

c) (i) use cond$e(a)$ = 455.462

(ii) singular matrix message displayed

Excel solution to 11.11 (ii)

Sheet1

| | | | | |
|---|---|---|---|---|
| 1 | 4 | 9 | 16 | 30 |
| 4 | 9 | 16 | 25 | 54 |
| 9 | 16 | 25 | 36 | 86 |
| 16 | 25 | 36 | 49 | 126 |
| | | | | |
| 1.67E+14 | -5E+14 | 5E+14 | -1.7E+14 | 108 |
| -5E+14 | 1.5E+15 | -1.5E+15 | 5E+14 | -56 |
| 5E+14 | -1.5E+15 | 1.5E+15 | -5E+14 | -16 |
| -1.7E+14 | 5E+14 | -5E+14 | 1.67E+14 | 0 |

Matlab solution to Prob. 11.11 (ii):

```
a=[1 4 9 16;4 9 16 25;9 16 25 36;16 25 36 49]

a =

     1     4     9    16
     4     9    16    25
     9    16    25    36
    16    25    36    49

b=[30 54 86 126]

b =

    30    54    86   126

b=b'

b =

    30
    54
    86
   126

x=a\b
Warning: Matrix is close to singular or badly scaled.
         Results may be inaccurate. RCOND = 2.092682e-018.

x =

    1.1053
    0.6842
    1.3158
    0.8947

x=inv(a)*b
Warning: Matrix is close to singular or badly scaled.
         Results may be inaccurate. RCOND = 2.092682e-018.

x =

     0
     0
     0
     0

cond(a)

ans =

  4.0221e+017
```

11.12

```
Program Linsimp
Use IMSL
Implicit None
Integer::ipath,lda,n,ldfac
Parameter(ipath=1,lda=3,ldfac=3,n=3)
Integer::ipvt(n),i,j
Real::A(lda,lda),Rcond,Res(n)
Real::Rj(n),B(n),X(n)
Data A/1.0,0.5,0.3333333,0.5,0.3333333,0.25,0.3333333,0.25,0.2/
Data B/1.833333,1.083333,0.783333/

Call linsol(n,A,B,X,Rcond)
Print *, 'Condition number = ', 1.0E0/Rcond
Print *
Print *, 'Solution:'
Do I = 1,n
  Print *, X(i)
End Do
End Program

Subroutine linsol(n,A,B,X,Rcond)
Implicit none
Integer::n, ipvt(3)
Real::A(n,n), fac(n,n), Rcond, res(n)
Real::B(n), X(n)
Call lfcrg(n,A,3,fac,3,ipvt,Rcond)
Call lfirg(n,A,3,fac,3,ipvt,B,1,X,res)
End
```

11.13

```
Option Explicit

Sub TestChol()

Dim i As Integer, j As Integer
Dim n As Integer
Dim a(10, 10) As Single

n = 3
a(1, 1) = 6: a(1, 2) = 15: a(1, 3) = 55
a(2, 1) = 15: a(2, 2) = 55: a(2, 3) = 225
a(3, 1) = 55: a(3, 2) = 225: a(3, 3) = 979

Call Cholesky(a(), n)

'output results to worksheet
Sheets("Sheet1").Select
Range("a3").Select
For i = 1 To n
  For j = 1 To n
    ActiveCell.Value = a(i, j)
    ActiveCell.Offset(0, 1).Select
  Next j
  ActiveCell.Offset(1, -n).Select
Next i
Range("a3").Select

End Sub
Sub Cholesky(a, n)

Dim i As Integer, j As Integer, k As Integer
Dim sum As Single
```

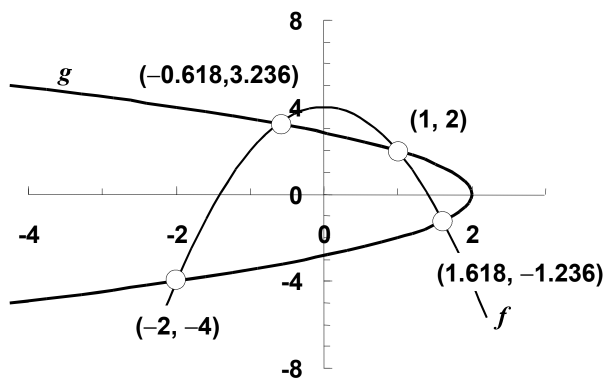
```
For k = 1 To n
  For i = 1 To k - 1
    sum = 0
    For j = 1 To i - 1
      sum = sum + a(i, j) * a(k, j)
    Next j
    a(k, i) = (a(k, i) - sum) / a(i, i)
  Next i
  sum = 0
  For j = 1 To k - 1
    sum = sum + a(k, j) ^ 2
  Next j
  a(k, k) = Sqr(a(k, k) - sum)
Next k

End Sub
```

11.14

```
Option Explicit

Sub Gausseid()
Dim n As Integer, imax As Integer, i As Integer
Dim a(3, 3) As Single, b(3) As Single, x(3) As Single
Dim es As Single, lambda As Single
n = 3
a(1, 1) = 3: a(1, 2) = -0.1: a(1, 3) = -0.2
a(2, 1) = 0.1: a(2, 2) = 7: a(2, 3) = -0.3
a(3, 1) = 0.3: a(3, 2) = -0.2: a(3, 3) = 10
b(1) = 7.85: b(2) = -19.3: b(3) = 71.4
es = 0.1
imax = 20
lambda = 1#
Call Gseid(a(), b(), n, x(), imax, es, lambda)
For i = 1 To n
  MsgBox x(i)
Next i
End Sub

Sub Gseid(a, b, n, x, imax, es, lambda)
Dim i As Integer, j As Integer, iter As Integer, sentinel As Integer
Dim dummy As Single, sum As Single, ea As Single, old As Single
For i = 1 To n
  dummy = a(i, i)
  For j = 1 To n
    a(i, j) = a(i, j) / dummy
  Next j
  b(i) = b(i) / dummy
Next i
For i = 1 To n
  sum = b(i)
  For j = 1 To n
    If i <> j Then sum = sum - a(i, j) * x(j)
  Next j
  x(i) = sum
Next i
iter = 1
Do
  sentinel = 1
  For i = 1 To n
    old = x(i)
    sum = b(i)
    For j = 1 To n
      If i <> j Then sum = sum - a(i, j) * x(j)
    Next j
```

```
      x(i) = lambda * sum + (1# - lambda) * old
      If sentinel = 1 And x(i) <> 0 Then
        ea = Abs((x(i) - old) / x(i)) * 100
        If ea > es Then sentinel = 0
      End If
    Next i
    iter = iter + 1
    If sentinel = 1 Or iter >= imax Then Exit Do
  Loop
  End Sub
```

11.15 As shown, there are 4 roots, one in each quadrant.




It might be expected that if an initial guess was within a quadrant, the result wouls be the root in the quadrant. However a sample of initial guesses spanning the range yield the following roots:

| | | | | | |
|---|---|---|---|---|---|
| 6 | (-2, -4) | (-0.618,3.236) | (-0.618,3.236) | (1,2) | (-0.618,3.236) |
| 3 | (-0.618,3.236) | (-0.618,3.236) | (-0.618,3.236) | (1,2) | (-0.618,3.236) |
| 0 | (1,2) | (1.618, -1.236) | (1.618, -1.236) | (1.618, -1.236) | (1.618, -1.236) |
| -3 | (-2, -4) | (-2, -4) | (1.618, -1.236) | (1.618, -1.236) | (1.618, -1.236) |
| -6 | (-2, -4) | (-2, -4) | (-2, -4) | (1.618, -1.236) | (-2, -4) |
| | -6 | -3 | 0 | 3 | 6 |

We have highlighted the guesses that converge to the roots in their quadrants. Although some follow the pattern, others jump to roots that are far away. For example, the guess of (-6, 0) jumps to the root in the first quadrant.

This underscores the notion that root location techniques are highly sensitive to initial guesses and that open methods like the Solver can locate roots that are not in the vicinity of the initial guesses.

11.16

x = transistors
y = resistors
z = computer chips

System equations:

$$\begin{aligned} 3x + 3y + 2z &= 810 \\ x + 2y + z &= 410 \\ 2x + y + 2z &= 490 \end{aligned}$$

Let A = $\begin{bmatrix} 3 & 3 & 2 \\ 1 & 2 & 1 \\ 2 & 1 & 2 \end{bmatrix}$ and B = $\begin{bmatrix} 810 \\ 410 \\ 490 \end{bmatrix}$

Plug into Excel and use two functions- minverse mmult

Apply $Ax = B$
$x = A^{-1} * B$

Answer: $x = 100$, $y = 110$, $z = 90$

11.17 As ordered, none of the sets will converge. However, if Set 1 and 3 are reordered so that they are diagonally dominant, they will converge on the solution of (1, 1, 1).

Set 1:
$$\begin{aligned} 8x + 3y + z &= 12 \\ 2x + 4y - z &= 5 \\ -6x \qquad + 7z &= 1 \end{aligned}$$

Set 3:
$$\begin{aligned} 3x + y - z &= 3 \\ x + 4y - z &= 4 \\ x + y + 5z &= 7 \end{aligned}$$

At face value, because it is not diagonally dominant, Set 2 would seem to be divergent. However, since it is close to being diagonally dominant, a solution can be obtained by the following ordering:

Set 3:
$$\begin{aligned} -2x + 2y - 3z &= -3 \\ 2y - z &= 1 \\ -x + 4y + 5z &= 8 \end{aligned}$$

11.18

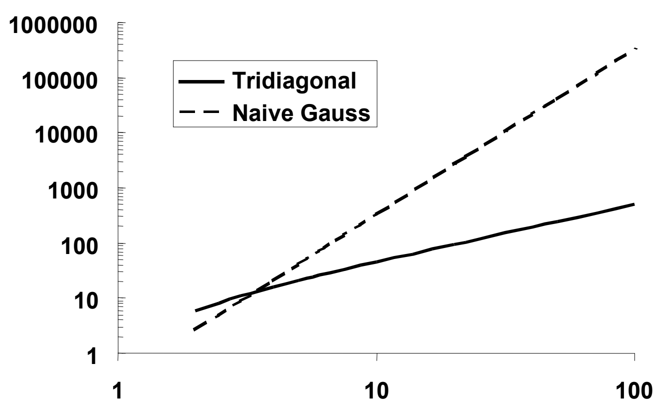
```
Option Explicit

Sub TriDiag()
Dim i As Integer, n As Integer
Dim e(10) As Single, f(10) As Single, g(10) As Single
Dim r(10) As Single, x(10) As Single
n = 4
e(2) = -1.2: e(3) = -1.2: e(4) = -1.2
f(1) = 2.04: f(2) = 2.04: f(3) = 2.04: f(4) = 2.04
g(1) = -1: g(2) = -1: g(3) = -1
r(1) = 40.8: r(2) = 0.8: r(3) = 0.8: r(4) = 200.8
Call Thomas(e(), f(), g(), r(), n, x())
For i = 1 To n
  MsgBox x(i)
Next i
End Sub

Sub Thomas(e, f, g, r, n, x)
Call Decomp(e, f, g, n)
Call Substitute(e, f, g, r, n, x)
End Sub
```

```
Sub Decomp(e, f, g, n)
Dim k As Integer
For k = 2 To n
  e(k) = e(k) / f(k - 1)
  f(k) = f(k) - e(k) * g(k - 1)
Next k
End Sub

Sub Substitute(e, f, g, r, n, x)
Dim k As Integer
For k = 2 To n
  r(k) = r(k) - e(k) * r(k - 1)
Next k
x(n) = r(n) / f(n)
For k = n - 1 To 1 Step -1
  x(k) = (r(k) - g(k) * x(k + 1)) / f(k)
Next k
End Sub
```

11.19 The multiplies and divides are noted below

```
Sub Decomp(e, f, g, n)
Dim k As Integer
For k = 2 To n
  e(k) = e(k) / f(k - 1)                   '(n - 1)
  f(k) = f(k) - e(k) * g(k - 1)            '(n - 1)
Next k
End Sub

Sub Substitute(e, f, g, r, n, x)
Dim k As Integer
For k = 2 To n
  r(k) = r(k) - e(k) * r(k - 1)            '(n - 1)
Next k
x(n) = r(n) / f(n)                         '   1
For k = n - 1 To 1 Step -1
  x(k) = (r(k) - g(k) * x(k + 1)) / f(k)       '2(n - 1)
Next k
End Sub

Sum =                                      5(n-1) + 1
```

They can be summed to yield $5(n - 1) + 1$ as opposed to $n^3/3$ for naive Gauss elimination. Therefore, a tridiagonal solver is well worth using.



Chapter 12

12.1 forcing function becomes

$$\begin{Bmatrix} 100 \\ 0 \\ 48 \\ 0 \\ 0 \end{Bmatrix}$$

solving with the coefficients gives

$c_1 = 17.89$
$c_2 = 17.89$
$c_3 = 7.32$
$c_4 = 10.20$
$c_5 = 17.89$

12.2 Use coefficents of matrix inverse

$a_{21}^{-1} = 0.1698$
$a_{31}^{-1} = 0.01887$

$\Delta c_2 = 0.1698 \times .25 \times 50$

$= \dfrac{2.12}{11.51} \times 100 = 18.4\%$

$\Delta c_3 = 0.01887 \times 0.25 \times 50$

$= \dfrac{0.236}{19.06} \times 100 = 1.24\%$

12.3 $Q_{01} + Q_{03} = Q_{44} + Q_{55}$

12.4

Reactor 1:

$50 - (3+4)c_1 + 2c_3 = 0$

Reactor 2:

$4c_1 - 4c_2 = 0$

Reactor 3:

$200 + c_2 - 11c_3 = 0$

Reactor 4:

$9c_3 + 2c_5 - 11c_4 = 0$

Reactor 5:

$3c_1 + 3c_2 - 6c_5 = 0$

or

$$\begin{aligned}
7c_1 \quad - 2c_3 &= 50 \\
4c_1 - 4c_2 &= 0 \\
-c_2 + 11c_3 &= 200 \\
9c_3 - 11c_4 + 2c_5 &= 0 \\
3c_1 + 3c_2 \quad - 6c_5 &= 0
\end{aligned}$$

gives $c_1 = 12.67$
$c_2 = 12.67$
$c_3 = 19.33$
$c_4 = 18.12$
$c_5 = 12.67$

12.5 Write flow balance for each reactor and entire system, gives

$$
\begin{aligned}
Q_{31} - Q_{15} &= -5 \\
-Q_{25} - Q_{24} - Q_{23} &= 0 \\
-Q_{31} + Q_{23} - Q_{34} &= -10 \\
+Q_{24} + Q_{34} - Q_{44} &= 0 \\
Q_{15} + Q_{25} - Q_{55} &= 0 \\
Q_{44} + Q_{55} &= 15
\end{aligned}
$$

6 equations with 8 unknowns
∴ no unique solution. more information is needed

12.6 mass balance equations

$$
\begin{aligned}
400 + 20c_2 &= 80c_1 + 40c_1 \\
80c_1 &= 20c_2 + 60c_2 \\
40c_1 + 60c_2 &= 10c_3
\end{aligned}
$$

or

$$
\begin{bmatrix} 140 & -20 & 0 \\ -80 & 80 & 0 \\ -40 & -60 & 120 \end{bmatrix}
\begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix} =
\begin{bmatrix} 400 \\ 0 \\ 200 \end{bmatrix}
$$

solving $c_1 = 3.33$
$c_2 = 3.33$
$c_3 = 4.44$

12.7 Superior, $c_1$

$$180 = 67c_1$$

Michigan, $c_2$

$$710 = 36c_2$$

Huron, $c_3$

$$67c_1 + 36c_2 + 740 = 161c_3$$

erie, $c_4$

$$161c_3 + 3850 = 182c_4$$

ontario, $c_5$

$$182c_4 + 4720 = 215c_5$$

$$
\begin{bmatrix} 67 & 0 & 0 & 0 & 0 \\ 0 & 36 & 0 & 0 & 0 \\ -67 & -36 & 161 & 0 & 0 \\ 0 & 0 & -161 & 182 & 0 \\ 0 & 0 & 0 & -182 & 212 \end{bmatrix}
\begin{Bmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \\ c_5 \end{Bmatrix} =
\begin{Bmatrix} 180 \\ 710 \\ 740 \\ 3850 \\ 4720 \end{Bmatrix}
$$

gives $c_1 = 2.7$
$c_2 = 19.7$
$c_3 = 10.1$
$c_4 = 30.1$ $c_5 = 48.1$

12.8

1st stage

$$F_1 Y_{in} + F_2 X_2 = F_2 X_1 + F_1 Y_1$$

$$X = KY$$

$$-\left(1+\frac{F_2}{F_1}K\right)Y_1 + \frac{F_2}{F_1}KY_2 = -Y_{in}$$

last stage

$$F_1 Y_4 + F_2 X_{in} = F_2 X_5 + F_1 Y_5$$

$$Y_4 - \left(1+\frac{F_2}{F_1}K\right)Y_5 = -\frac{F_2}{F_1}X_{in}$$

$$F_2/F_1 = 800/400 = 2$$

$$1+\frac{F_2}{F_1}K = 1+2(5) = 11$$

$$\begin{aligned}
11Y_1 - 2Y_2 &= 0.1\\
-2Y_1 + 11Y_2 - 2Y_3 &= 0\\
-2Y_2 + 11Y_3 - 2Y_4 &= 0\\
-2Y_3 + 11Y_4 - 2Y_5 &= 0\\
-2Y_4 + 11Y_5 &= 0
\end{aligned}$$

gives

$$\begin{aligned}
Y_1 &= 0.00941\\
Y_2 &= 0.00177\\
Y_3 &= 0.000334\\
Y_4 &= 0.0000627\\
Y_5 &= 0.0000114
\end{aligned}$$

$$X_i = 5Y_i$$

12.9

$$\rightarrow + \quad \sum F_x = 0: \quad P\cos 21.5^\circ - M\cos 37^\circ - M\cos 80^\circ = 0$$

$$0.93042P - 0.9723M = 0 \quad (1)$$

$$\uparrow + \quad \sum F_y = 0: \quad P\sin 21.5^\circ + M\sin 37^\circ - M\sin 80^\circ = 0$$

$$0.3665P - 0.383M = 0 \quad (2)$$

Use any method to solve equations (1) and (2):

$$A = \begin{bmatrix} \cos 21.5^\circ & -(\cos 37^\circ + \cos 80^\circ) \\ \sin 21.5^\circ & (\sin 37^\circ - \sin 80^\circ) \end{bmatrix}$$

$$B = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

Apply Ax = B where x = $\begin{bmatrix} P \\ M \end{bmatrix}$

Use Matlab or calculator for results

P = 314 lb
M = 300 lb

12.10 Mass balances can be written for each reactor as

$$
\begin{aligned}
0 &= Q_{\text{in}} c_{A,\text{in}} - Q_{\text{in}} c_{A,1} - k_1 V_1 c_{A,1} \\
0 &= Q_{\text{in}} c_{B,1} + k_1 V_1 c_{A,1} \\
0 &= Q_{\text{in}} c_{A,1} + Q_{32} c_{A,3} - (Q_{\text{in}} + Q_{32}) c_{A,2} - k_2 V_2 c_{A,2} \\
0 &= Q_{\text{in}} c_{B,1} + Q_{32} c_{B,3} - (Q_{\text{in}} + Q_{32}) c_{B,2} + k_2 V_2 c_{A,2} \\
0 &= (Q_{\text{in}} + Q_{32}) c_{A,2} + Q_{43} c_{A,4} - (Q_{\text{in}} + Q_{43}) c_{A,3} - k_3 V_3 c_{A,3} \\
0 &= (Q_{\text{in}} + Q_{32}) c_{B,2} + Q_{43} c_{B,4} - (Q_{\text{in}} + Q_{43}) c_{B,3} + k_3 V_3 c_{A,3} \\
0 &= (Q_{\text{in}} + Q_{43}) c_{A,3} - (Q_{\text{in}} + Q_{43}) c_{A,4} - k_4 V_4 c_{A,4} \\
0 &= (Q_{\text{in}} + Q_{43}) c_{B,3} - (Q_{\text{in}} + Q_{43}) c_{B,4} + k_4 V_4 c_{A,4}
\end{aligned}
$$

Collecting terms, the system can be expresses in matrix form as

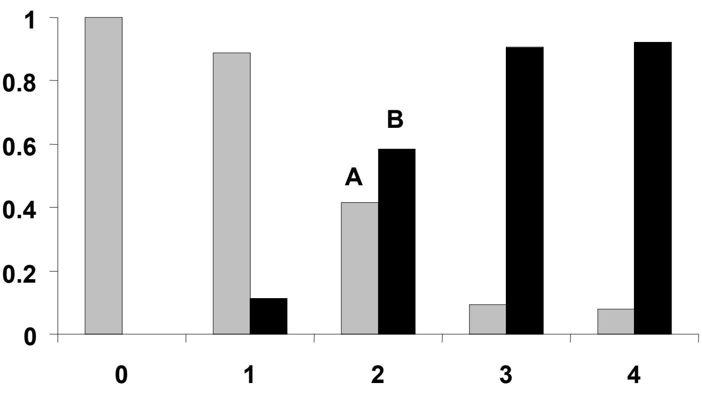
$[A]\{C\} = \{B\}$

where

$$
[A] = \begin{bmatrix}
11.25 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-1.25 & 10 & 0 & 0 & 0 & 0 & 0 & 0 \\
-10 & 0 & 22.5 & 0 & -5 & 0 & 0 & 0 \\
0 & -10 & -7.5 & 15 & 0 & -5 & 0 & 0 \\
0 & 0 & -15 & 0 & 68 & 0 & -3 & 0 \\
0 & 0 & 0 & -15 & -50 & 18 & 0 & -3 \\
0 & 0 & 0 & 0 & -13 & 0 & 15.5 & 0 \\
0 & 0 & 0 & 0 & 0 & -13 & -2.5 & 13
\end{bmatrix}
$$

$[B]^{\mathrm{T}} = [10\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0]$ $\qquad$ $[C]^{\mathrm{T}} = [c_{A,1}\ c_{B,1}\ c_{A,2}\ c_{B,2}\ c_{A,3}\ c_{B,3}\ c_{A,4}\ c_{B,4}]$

The system can be solved for $[C]^{\mathrm{T}}$ = [0.889 0.111 0.416 0.584 0.095 0.905 0.080 0.920].




12.11 Assuming a unit flow for $Q_1$, the simultaneous equations can be written in matrix form as

$$\begin{bmatrix} -1 & 2 & 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 2 & 1 & 0 \\ 0 & 0 & 0 & 0 & -1 & 3 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & -1 \end{bmatrix} \begin{Bmatrix} Q_2 \\ Q_3 \\ Q_4 \\ Q_5 \\ Q_6 \\ Q_7 \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{Bmatrix}$$

These equations can be solved to give $[Q]^T = [0.7321\ 0.2679\ 0.1964\ 0.0714\ 0.0536\ 0.0178]$.

12.12

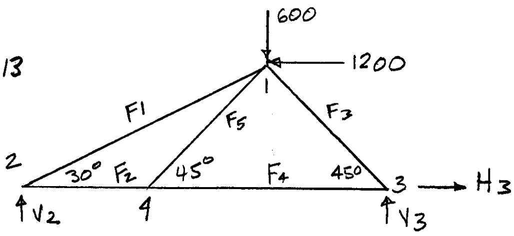
let $x_1$ = $m^3$ from pit 1
$x_2$ = $m^3$ from pit 2
$x_3$ = $m^3$ from pit 3

$$0.52x_1 + 0.2x_2 + 0.25x_3 = 4800$$
$$0.3x_1 + 0.5x_2 + 0.2x_3 = 5810$$
$$0.18x_1 + 0.3x_2 + 0.55x_3 = 5690$$

solving gives

$x_1 = 4011.6$
$x_2 = 7162.8$
$x_3 = 5125.6$

12.13




Node 4:

$\sum F_H = 0 = -F_2 + F_4 + F_5 \cos 45°$

$\sum F_V = 0 = F_5 \sin 45°$

$\therefore F_5 = 0$

$\therefore$ 7 equations with 7 unknowns

Node 1:

$\sum F_H = 0 = -F_1 \cos 30° - F_5 \cos 45° + F_3 \cos 45° - 1200$

$\sum F_V = 0 = -F_1 \sin 30° - F_5 \sin 45° - F_3 \sin 45° - 600$

Node 2:

$\sum F_H = 0 = F_2 + F_1 \cos 30°$

$\sum F_V = 0 = F_1 \sin 30° + V_2$

Node 3:

$\sum F_H = 0 = H_3 - F_4 - F_3 \cos 45°$

$\sum F_V = 0 = V_3 + F_3 \sin 45°$

$$\begin{bmatrix} 0.866 & 0 & -0.707 & 0 & 0 & 0 & 0 \\ 0.5 & 0 & 0.707 & 0 & 0 & 0 & 0 \\ -0.866 & -1 & 0 & 0 & 0 & 0 & 0 \\ -0.5 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0.707 & 1 & 0 & -1 & 0 \\ 0 & 0 & -0.707 & 0 & 0 & 0 & -1 \\ 0 & 1 & 0 & -1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} F_1 \\ F_2 \\ F_3 \\ F_4 \\ V_2 \\ H_3 \\ V_3 \end{bmatrix} = \begin{bmatrix} -1200 \\ -600 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

gives

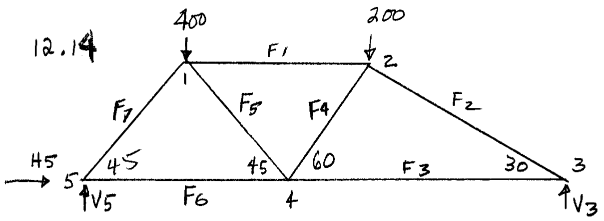
$F_1 = -1318$ $\quad V_2 = 659$

$F_2 = 1141$ $\quad H_3 = 1200$

$F_3 = 83$ $\quad V_3 = -59$

$F_4 = 1141$

12.14




Node 1:

$$\sum F_H = 0 = F_1 - F_7 \cos 45^\circ + F_5 \cos 45^\circ$$
$$\sum F_V = 0 = -400 - F_7 \sin 45^\circ - F_5 \sin 45^\circ$$

Node 2:

$$\sum F_H = 0 = -F_1 - F_4 \cos 60^\circ + F_2 \cos 30^\circ$$
$$\sum F_V = 0 = -200 - F_4 \sin 60^\circ - F_2 \sin 30^\circ$$

Node 3:

$$\sum F_H = 0 = -F_3 - F_2 \cos 30^\circ$$
$$\sum F_V = 0 = V_3 + F_2 \sin 30^\circ$$

Node 4:

$$\sum F_H = 0 = -F_6 + F_3 - F_5 \cos 45^\circ + F_4 \cos 60^\circ$$
$$\sum F_V = 0 = F_5 \sin 45^\circ + F_4 \sin 60^\circ$$

Node 5:

$$\sum F_H = 0 = H_5 + F_6 + F_7 \cos 45^\circ$$
$$\sum F_V = 0 = V_5 + F_7 \sin 45^\circ$$

In matrix form

$$\begin{bmatrix} -1 & 0 & 0 & 0 & -0.707 & 0 & 0.707 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.707 & 0 & 0.707 & 0 & 0 & 0 \\ 1 & -0.866 & 0 & 0.5 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.5 & 0 & 0.866 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.866 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -0.5 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & -0.5 & 0.707 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -0.866 & -0.707 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & -0.707 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -0.707 & 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} F_1 \\ F_2 \\ F_3 \\ F_4 \\ F_5 \\ F_6 \\ F_7 \\ V_3 \\ H_5 \\ V_5 \end{bmatrix} = \begin{bmatrix} 0 \\ -400 \\ 0 \\ -200 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

which can be solved for

$F_1 = -375.1$
$F_2 = -424.9$
$F_3 = 367.9$
$F_4 = 14.4$
$F_5 = -17.6$
$F_6 = 387.6$
$F_7 = -548.2$

$V_3 = 212.4$
$H_5 = 0$
$V_5 = 387.6$

12.15

use the first 2 columns of the inverse

| | $F_1H$ | $F_1V$ |
|---|---|---|
| $F_1$ | 0.866 | 0.5 |
| $F_2$ | 0.25 | -0.433 |
| $F_3$ | -0.5 | 0.866 |
| $H_2$ | -1 | 0 |
| $V_2$ | -0.433 | -0.25 |
| $V_3$ | 0.433 | -0.75 |

$F_1 = 1800\,(0.866) - 2200\,(.5) = 458.8$

$F_2 = 1800\,(0.25) - 2200\,(-0.433) = 1402.6$

$F_3 = 1800\,(-0.5) - 2200\,(0.866) = -2805.2$

$H_2 = 1800\,(-1) - 2200\,(0) = -1800$

$V_2 = 1800\,(-0.433) - 2200\,(-0.25) = -229.4$

$V_3 = 1800\,(0.433) - 2200\,(-0.75) = 2429.4$

12.16

$\sum F_y = 0 \qquad V_2 + V_3 = 1000$

$\sum M = 0 \qquad 1000\,(\cos 30^\circ) L_1 - V_3 L_2$

Geometry $\qquad \cos 30^\circ L_1 + \cos 60^\circ L_3 = L_2$

$V_2 = 250 \qquad V_3 = 750$ , thus

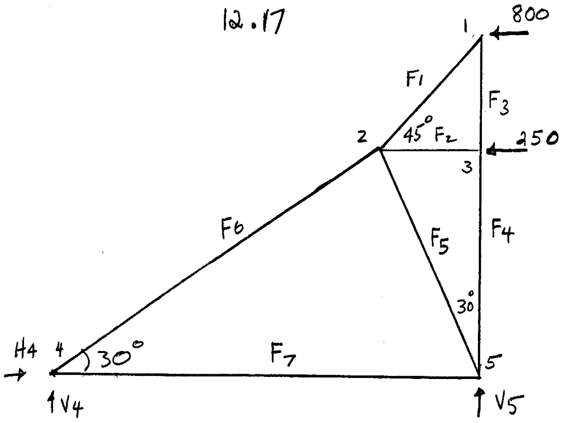
$866\, L_1 - 750 L_2 = 0$

$0.866\, L_1 + 0.5\, L_3 = L_2$

solving

$$L_3 = \frac{L_2 - 0.866\, L_1}{0.5}$$

12.17




Node 1:

$\sum F_H = 0 = -F_1 \cos 45^\circ - 800$

$\sum F_V = 0 = -F_1 \sin 45^\circ - F_3$

Node 2:

$\sum F_H = 0 = F_1 \cos 45^\circ + F_2 - F_6 \cos 30^\circ + F_5 \cos 60^\circ$

$\sum F_V = 0 = F_1 \sin 45^\circ - F_6 \sin 30^\circ - F_5 \sin 60^\circ$

Node 3:

$\sum F_H = 0 = -F_2 - 250$

$\sum F_V = 0 = F_3 - F_4$

Node 4:

$\sum F_H = 0 = F_6 \cos 30^\circ + F_7 + H_4$

$\sum F_V = 0 = F_6 \sin 30^\circ + V_4$

Node 5:

$\sum F_H = 0 = -F_7 - F_5 \cos 60^\circ$

$\sum F_V = 0 = F_4 + F_5 \sin 60^\circ + V_5$

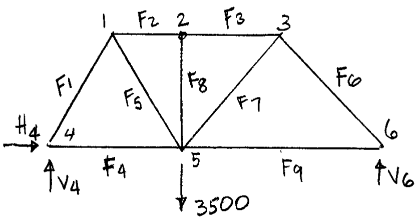
$$\begin{bmatrix} 0.707 & 0 & 0 & 0 & 0 & & & & & \\ 0.707 & 0 & 1 & 0 & 0 & & & & & \\ -.707 & -1 & 0 & 0 & -0.5 & 0.866 & & & & \\ -.707 & 0 & 0 & 0 & 0.866 & 0.5 & & & & \\ 0 & +1 & 0 & 0 & 0 & & & & & \\ 0 & 0 & -1 & 1 & 0 & & & & & \\ 0 & 0 & 0 & 0 & 0 & -0.866 & -1 & +1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -0.5 & & & -1 & \\ 0 & 0 & 0 & 0 & 0.5 & & & 1 & & \\ 0 & 0 & 0 & -1 & -0.866 & 0 & 0 & 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} F_1 \\ F_2 \\ F_3 \\ F_4 \\ F_5 \\ F_6 \\ F_7 \\ H_4 \\ V_4 \\ V_5 \end{bmatrix} = \begin{bmatrix} -800 \\ 0 \\ 0 \\ 0 \\ -250 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Solving gives

$F_1 = -1132.5$
$F_2 = -250$
$F_3 = 800$
$F_4 = 800$
$F_5 = -167.8$
$F_6 = -1309.3$
$F_7 = 89.9$

$H_4 = 1050$
$V_4 = 654.7$
$V_5 = -654.7$

12.18




node 1:

$\sum F_H = 0 = F_2 - F_1 \cos 60^\circ + F_5 \cos 60^\circ$
$\sum F_V = 0 = -F_1 \sin 60^\circ - F_5 \sin 60^\circ$

Node 2:

$\sum F_H = 0 = -F_2 + F_3$
$\sum F_V = 0 = -F_8$

Node 3:

$\sum F_H = 0 = -F_3 - F_7 \cos 45^\circ + F_6 \cos 45^\circ$
$\sum F_V = 0 = -F_7 \sin 45^\circ - F_6 \sin 45^\circ$

Node 4:

$\sum F_H = 0 = F_1 \cos 60^\circ + F_4 + H_4$
$\sum F_V = 0 = F_1 \sin 60^\circ + V_4$

Node 5:

$\sum F_H = 0 = -F_4 - F_5 \cos 60^\circ + F_7 \cos 45^\circ + F_9$

$\sum F_V = 0 = F_5 \sin 60^\circ + F_8 + F_7 \sin 45^\circ - 3500$

Node 6:

$\sum F_H = 0 = -F_9 - F_6 \cos 45^\circ$
$\sum F_V = 0 = F_6 \sin 45^\circ + V_6$

Note that $F_8 = 0$ thus the middle member is unnecessary unless there is a load with a non zero vertical component at node 2.

Solve for 11 unknowns

$$
\begin{bmatrix}
0.5 & -1 & 0 & 0 & -0.5 & 0 & 0 & 0 & 0 & 0 & 0 \\
0.866 & 0 & 0 & 0 & 0.866 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & -0.707 & 0.707 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0.707 & 0.707 & 0 & 0 & 0 & 0 \\
-0.5 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\
-0.866 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\
0 & 0 & 0 & 1 & 0.5 & 0 & -0.707 & -1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & -0.866 & 0 & -0.707 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0.707 & 0 & +1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & -0.707 & 0 & 0 & 0 & 0 & -1
\end{bmatrix}
\begin{bmatrix} F_1 \\ F_2 \\ F_3 \\ F_4 \\ F_5 \\ F_6 \\ F_7 \\ F_9 \\ H_4 \\ V_4 \\ V_6 \end{bmatrix}
=
\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ -3500 \\ 0 \\ 0 \end{bmatrix}
$$

solving gives

$F_1 = -2562.2$ $\quad F_5 = 2562.2$ $\quad H_4 = 0$

$F_2 = -2562.2$ $\quad F_6 = -1812.0$ $\quad V_4 = 2218.9$

$F_3 = -2562.2$ $\quad F_7 = 1812.0$ $\quad V_6 = 1281.1$

$F_4 = 1281.1$ $\quad F_9 = 1281.1$

12.19

a)

Room 1

$$0 = W_{smoker} + Q_a C_a - Q_a C_1 + E_{13}(C_3 - C_1)$$

Room 2

$$0 = Q_b C_b + \frac{Q_a}{2} C_4 - Q_c C_2 + E_{24}(C_4 - C_2)$$

Room 3

$$0 = W_{grill} + Q_a C_1 + E_{13}(C_1 - C_3) + E_{34}(C_4 - C_3) - Q_a C_3$$

Room 4

$$0 = Q_a C_3 + E_{34}(C_3 - C_4) + E_{24}(C_2 - C_4) - Q_d C_4 - \frac{Q_a}{2} C_4$$

where $Q_c = Q_b + \frac{Q_a}{2}$ and $Q_d = \frac{Q_a}{2}$

$$\begin{bmatrix} 225 & 0 & -25 & 0 \\ & 175 & & -125 \\ -225 & & 275 & -50 \\ & -25 & -250 & 275 \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \end{bmatrix} = \begin{bmatrix} 1400 \\ 100 \\ 2000 \\ 0 \end{bmatrix}$$

gives $c_1 = 8.10$
$c_2 = 12.34$
$c_3 = 16.90$
$c_4 = 16.48$

check: $150(12.34) + 100(16.48) = 3499$
$100 + 400 + 1000 + 2000 = 3500$ } check ✓

b) smokers RHS $= \begin{Bmatrix} 1000 \\ 0 \\ 0 \\ 0 \end{Bmatrix}$

gives $c_2 = 3.45$

grill RHS $= \begin{Bmatrix} 0 \\ 0 \\ 200 \\ 0 \end{Bmatrix}$

gives $c_2 = 6.90$

vent RHS $= \begin{Bmatrix} 400 \\ 100 \\ 0 \\ 0 \end{Bmatrix}$

$c_2 = 2.0$

smoker $= 3.45/12.34 \times 100 = 27.6\%$
grill $= 6.9/12.34 \times 100 = 55.9\%$
vent $= 2/12.34 \times 100 = 16.2\%$
$\approx 100\%$

12.20 Find the unit vectors:

$$A\left(\frac{1\hat{i} - 2\hat{j} - 4\hat{k}}{\sqrt{1^2 + 2^2 + 4^2}}\right) = 0.218\hat{i} - 0.436\hat{j} - 0.873\hat{k}$$

$$B\left(\frac{2\hat{i} + 1\hat{j} - 4\hat{k}}{\sqrt{1^2 + 2^2 + 4^2}}\right) = 0.436\hat{i} + 0.218\hat{j} - 0.873\hat{k}$$

Sum moments about the origin:

$$\sum M_{ox} = 50(2) - 0.436B(4) - 0.218A(4) = 0$$

$$\sum M_{oy} = 0.436A(4) - 0.218B(4) = 0$$

Solve for A & B using equations 9.10 and 9.11:

In the form
$$a_{11}x_1 + a_{12}x_2 = b_1$$
$$a_{21}x_2 + a_{22}x_2 = b_2$$

$$-0.872A + -1.744B = -100$$
$$1.744A + -0.872B = 0$$

Plug into equations 9.10 and 9.11:

$$x_1 = \frac{a_{22}b_1 - a_{12}b_2}{a_{11}a_{22} - a_{12}a_{21}} = \frac{87.2}{3.80192} = 22.94 \quad N$$

$$x_2 = \frac{a_{11}b_2 - a_{21}b_1}{a_{11}a_{22} - a_{12}a_{21}} = \frac{174.4}{3.80192} = 45.87 \quad N$$

12.21

$$T\left(\frac{1\hat{i} + 6\hat{j} - 4\hat{k}}{\sqrt{1^2 + 6^2 + 4^2}}\right) = 0.1374\hat{i} + 0.824\hat{j} - 0.549\hat{k}$$

$$\sum M_y = -5(1) + -0.549T(1) = 0$$

$$T = 9.107\ kN$$

$$\therefore T_x = 1.251\ kN\,, T_y = 7.50\ kN\,, T_z = -5\ kN$$

$$\sum M_x = -5(3) + -7.5(4) + -5(3) + B_z(3) = 0 \qquad B_z = 20\ kN$$
$$\sum M_z = 7.5(3) + 1.251(3) + B_x(3) = 0$$
$$B_x = -3.751\ kN$$
$$\sum F_z = -5 + -5 + A_z + 20 = 0$$
$$A_z = -10\ kN$$
$$\sum F_x = A_x + -3.751 + 1.251 = 0 \qquad A_x = 2.5\ kN$$
$$\sum F_y = 7.50 + A_y = 0 \qquad Ay = -7.5\ kN$$

12.22 This problem was solved using Matlab.

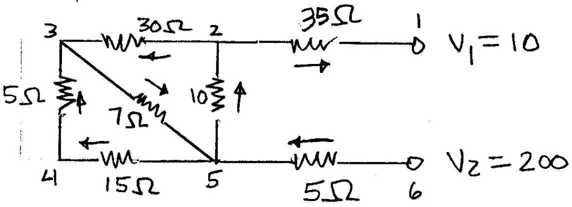
```
A = [1  0  0  0  0  0  0  0  1  0
     0  0  1  0  0  0  0  1  0  0
     0  1  0  3/5  0  0  0  0  0  0
     -1  0  0  -4/5  0  0  0  0  0  0
     0  -1  0  0  0  0  3/5  0  0  0
     0  0  0  0  -1  0  -4/5  0  0  0
     0  0  -1  -3/5  0  1  0  0  0  0
     0  0  0  4/5  1  0  0  0  0  0
     0  0  0  0  0  -1  -3/5  0  0  0
     0  0  0  0  0  0  4/5  0  0  1];

b = [0  0  -54  0  0  24  0  0  0  0];

x=inv(A)*b

x =
```

```
 24.0000
-36.0000
 54.0000
-30.0000
 24.0000
 36.0000
-60.0000
-54.0000
-24.0000
 48.0000
```

Therefore, in kN

| | | | | |
|---|---|---|---|---|
| AB = 24 | BC = −36 | AD = 54 | BD = −30 | CD = 24 |
| DE = 36 | CE = −60 | $A_x$ = −54 | $A_y$ = −24 | $E_y$ = 48 |

12.23




current equations:

$$-i_{12} - i_{23} + i_{52} = 0$$
$$i_{23} - i_{53} + i_{43} = 0$$
$$-i_{43} + i_{54} = 0$$
$$i_{35} - i_{52} + i_{65} - i_{54} = 0$$

voltage equations:

$$i_{21} = \frac{V2 - 10}{35} \qquad i_{54} = \frac{V5 - V4}{15}$$

$$i_{23} = \frac{V_2 - V_3}{30} \qquad i_{35} = \frac{V_3 - V_5}{7}$$

$$i_{43} = \frac{V_4 - V_3}{5} \qquad i_{52} = \frac{V_5 - V_2}{10}$$

$$i_{65} = \frac{200 - V_5}{5}$$

$$\begin{bmatrix} -1 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 & -1 & 1 & 0 & 0 & 0 & 0 \\ 35 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 30 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 5 & 0 & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 15 & 0 & 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 7 & 0 & 0 & 0 & 0 & -1 & 0 & 1 \\ 0 & 0 & 10 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 5 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} i_{21} \\ i_{23} \\ i_{52} \\ i_{35} \\ i_{43} \\ i_{54} \\ i_{65} \\ V_2 \\ V_3 \\ V_4 \\ V_5 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ -10 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 200 \end{bmatrix}$$

$i_{21} = 3.98$ $\quad i_{54} = 0.23$ $\quad V_5 = 180$

$i_{23} = -0.88$ $\quad i_{65} = 3.98$

$i_{52} = 3.10$ $\quad V_2 = 149$

$i_{35} = -0.65$ $\quad V_3 = 176$

$i_{43} = 0.23$ $\quad V_4 = 177$

12.24 The simultaneous equations are

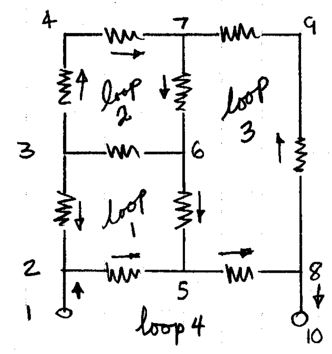
$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 1 & -1 & 0 \\ 0 & 0 & -1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & -1 \\ 0 & 10 & -10 & 0 & -15 & -5 \\ 5 & -10 & 0 & -50 & 0 & 0 \end{bmatrix} \begin{bmatrix} i_{12} \\ i_{52} \\ i_{32} \\ i_{65} \\ i_{54} \\ i_{43} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 180 \end{bmatrix}$$

$i_{12} = 2.88$ $\quad i_{65} = -2.88$

$i_{52} = -2.16$ $\quad i_{54} = -0.72$

$i_{32} = -0.72$ $\quad i_{43} = -0.72$

12.25




write 8 current node equations and 4 loop voltage balances

$$i_{32} - i_{25} + i_{12} = 0$$
$$-i_{32} - i_{34} + i_{63} = 0$$
$$i_{34} - i_{47} = 0$$
$$i_{25} + i_{65} - i_{58} = 0$$
$$i_{76} - i_{63} - i_{65} = 0$$
$$i_{47} - i_{76} + i_{79} = 0$$
$$i_{58} - i_{89} - i_{810} = 0$$
$$i_{89} - i_{79} = 0$$

$$-20 i_{25} + 10 i_{65} - 5 i_{63} - 5 i_{32} = 0$$
$$5 i_{63} + 10 i_{76} + 5 i_{47} + 20 i_{34} = 0$$
$$-50 i_{58} - 15 i_{89} - 0 i_{79} - 10 i_{76} - 10 i_{65} = 0$$
$$110 - 20 i_{25} - 50 i_{58} = 40$$

Express in matrix form

$$\begin{bmatrix}
1 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-1 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & -1 & 0 & -1 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & -1 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & -1 & -1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 \\
5 & 20 & 0 & 0 & 5 & 0 & -10 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & -20 & -5 & -5 & 0 & 0 & -10 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 10 & 50 & 10 & 0 & 15 & 0 \\
0 & 20 & 0 & 0 & 0 & 0 & 0 & 50 & 0 & 0 & 0 & 0
\end{bmatrix}
\begin{bmatrix}
i_{32} \\ i_{25} \\ i_{12} \\ i_{34} \\ i_{63} \\ i_{47} \\ i_{65} \\ i_{58} \\ i_{76} \\ i_{79} \\ i_{89} \\ i_{810}
\end{bmatrix}$$

gives

$$\begin{matrix} i_{32} = & -2.40 \\ i_{25} = & 0.99 \\ i_{12} = & 3.38 \\ i_{34} = & 0.89 \\ i_{63} = & -1.50 \\ i_{47} = & 0.89 \\ i_{65} = & 0.023 \\ i_{58} = & 1.00 \\ i_{76} = & -1.48 \\ i_{79} = & -2.38 \\ i_{89} = & -2.38 \\ i_{810} = & 3.38 \end{matrix} \quad = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 70 \end{bmatrix}$$

12.26

$$15c_1 + 17c_2 + 19c_3 = 2.12$$
$$0.25c_1 + 0.33c_2 + 0.42c_3 = 0.0434$$
$$c_1 + 1.2c_2 + 1.6c_3 = 0.164$$

$$c_1 = 20$$
$$c_2 = 40$$
$$c_3 = 60$$

12.27 This problem can be solved directly on a calculator capable of doing matrix operations or on Matlab.

```
a=[60 -40 0
   -40 150 -100
   0 -100 130];
b=[200
   0
   230];

x=inv(a)*b

x =

     7.7901
     6.6851
     6.9116
```

Therefore,

$I_1 = 7.79$ A
$I_2 = 6.69$ A
$I_3 = 6.91$ A

12.28 This problem can be solved directly on a calculator capable of doing matrix operations or on Matlab.

```
a=[17 -8 -3
```

```
   -2 6 -3
   -1 -4 13];
b=[480
   0
   0];

x=inv(a)*b

x =

   37.3585
   16.4151
    7.9245
```

Therefore,

$V_1 = 37.4$ V
$V_2 = 16.42$ V
$V_3 = 7.92$ V

12.29 This problem can be solved directly on a calculator capable of doing matrix operations or on Matlab.

```
a=[6 0 -4 1
   0 8 -8 -1
   -4 -8 18 0
   -1 1 0 0];

b=[0
   -20
   0
   10];

x=inv(a)*b

x =

   -7.7778
    2.2222
   -0.7407
   43.7037
```

Therefore,

$I_1 = -7.77$ A
$I_2 = 2.22$ A
$I_3 = -.741$ A
$V_s = 43.7$ V

12.30 This problem can be solved directly on a calculator capable of doing matrix operations or on Matlab.

```
a=[55 0 -25
   0 37 -4
   -25 -4 29];
b=[-200
   -250
    100];

x=inv(a)*b
```

```
x =

   -4.1103
   -6.8695
   -1.0426
```

Therefore,

$I_1$ = -4.11 A
$I_3$ = -6.87 A
$I_4$ = -1.043 A

12.31 at steady state

$$4kx_1 - 3kx_2 = m_1 g$$
$$-3kx_1 + 4kx_2 - kx_3 = m_2 g$$
$$-kx_2 + kx_3 = m_3 g$$

and substituting parameter values

$$\begin{bmatrix} 80 & -60 & 0 \\ -60 & 40 & -10 \\ 0 & -10 & 10 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{Bmatrix} 19.6 \\ 29.4 \\ 24.5 \end{Bmatrix}$$

solving gives

$x_1 = 2.303$
$x_2 = 2.744$
$x_3 = 5.194$

12.32

$$\{W\} = \begin{Bmatrix} 15 \times 9.8 \\ 3 \times 9.8 \\ 2 \times 9.8 \end{Bmatrix} = \begin{Bmatrix} 147 \\ 29.4 \\ 19.6 \end{Bmatrix}$$

multiply by the inverse stiffness matrix to yield

$x_1 = 19.6$
$x_2 = 22.05$
$x_3 = 24.01$

12.33 $50(x_2 - x_1) = 150x_1$

$75(x_3 - x_2) = 50(x_2 - x_1)$

$225(x_4 - x_3) = 75(x_3 - x_2)$

$2000 = 225(x_4 - x_3)$

$$\begin{bmatrix} -200 & 50 & 0 & 0 \\ 50 & -125 & 75 & 0 \\ 0 & 75 & -300 & 225 \\ 0 & 0 & -225 & 225 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 2000 \end{bmatrix}$$

solving gives

$x_1 = 13.33$ $\quad x_3 = 80.0$

$x_2 = 53.33$ $\quad x_4 = 88.89$

12.34) $100a + T = 519.72$

$50a - T + R = 216.55$

$25a - R = 108.27$

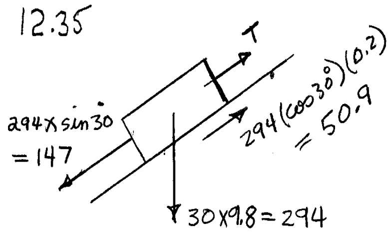
solving gives

$a = 4.826$

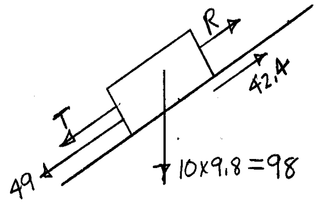
$T = 37.126$

$R = 12.379$

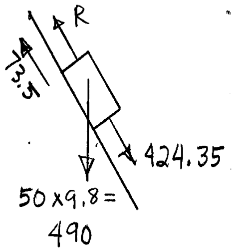
12.35




$$147 - 50.9 - T = 30a$$




$$49 + T - R - 42.4 = 10a$$

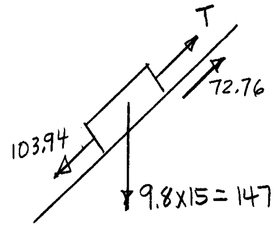



$$-424.35 + 73.5 + R = 50a$$

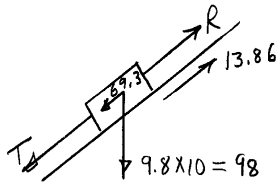
$$\begin{bmatrix} 30a + T \\ 10a - T + R \\ 50a - R \end{bmatrix} = \begin{bmatrix} 96.1 \\ 6.6 \\ -350.85 \end{bmatrix}$$

gives $a = -2.757$
$T = 178.82$
$R = 212.99$

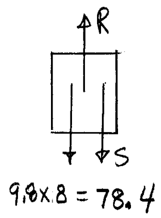
12.36



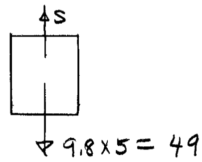

$$103.94 - T - 72.76 = 15a$$




$$T + 69.3 - R - 13.86 = 10a$$




$$R - 78.4 - S = 8a$$




$$S - 49 = 5a$$

$$\begin{bmatrix} 15 & 1 & 0 & 0 \\ 10 & -1 & 1 & 0 \\ 8 & 0 & -1 & 1 \\ 5 & 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} a \\ T \\ R \\ S \end{bmatrix} = \begin{bmatrix} 31.18 \\ 55.44 \\ -78.4 \\ -49 \end{bmatrix}$$

Solving gives

$a = -1.073$
$T = 47.277$
$R = 113.449$
$S = 43.634$

12.37

```
%massspring34.m
```

```
k1=10;
k2=40;
k3=40;
k4=10;
m1=1;
m2=1;
m3=1;
km=[(1/m1)*(k2+k1), -(k2/m1),0; -(k2/m2), (1/m2)*(k2+k3), -(k3/m2);
0, -(k3/m3),(1/m3)*(k3+k4)];
X=[0.05;0.04;0.03];
kmx=km*X

kmx =

0.9000
0.0000
-0.1000
```

Therefore, $\ddot{x}_1$ = -0.9, $\ddot{x}_2 = 0$ , and $\ddot{x}_3 = 0.1$ m/s$^2$.

# CHAPTER 16

16.1

$A = \pi r^2 + 2\pi r h$

$V = \pi r^2 h = .2$

Using Excel Solver

$r = 0.399481$
$h = 0.398916$
$A = 1.502636$

16.2

$A = \pi r \sqrt{r^2 + h^2} + \pi r^2$

$V = \pi r^2 h / 3$

Excel Solver gives

$r = 0.407156$
$h = 1.152068$
$A = 2.083753$

16.3 Solver gives

$C = 1.567889$

$g_{max} = 0.369635$

16.4 (*a*) The total LP formulation is given by

Maximize $C = 0.15X + 0.025Y + 0.05Z$ {Maximize profit}

subject to

$X + Y + Z \geq 6$ {Material constraint}
$X + Y < 3$ {Time constraint}
$X - Y \geq 0$ {Storage constraint}

$Z - 0.5Y \geq 0$ {Positivity constraints}

(*b*) The simplex tableau for the problem can be set up and solved as

($c$) An Excel spreadsheet can be set up to solve the problem as

The Solver can be called and set up as

The resulting solution is

In addition, a sensitivity report can be generated as

($d$) The high shadow price for storage from the sensitivity analysis from ($c$) suggests that increasing storage will result in the best increase in profit.

16.5 An LP formulation for this problem can be set up as

Maximize $C = 0.15X + 0.025Y + 0.05Z$ {Maximize profit}

subject to

$X + Y + Z \geq 6$ {$X$ material constraint}
$X + Y < 3$ {$Y$ material constraint}
$X - Y \geq 0$ {Waste constraint}

$Z - 0.5Y \geq 0$ {Positivity constraints}

($b$) An Excel spreadsheet can be set up to solve the problem as

The Solver can be called and set up as

The resulting solution is

This is an interesting result which might seem counterintuitive at first. Notice that we create some of the unprofitable $z_2$ while producing none of the profitable $z_3$. This occurred because we used up all of $Y$ in producing the highly profitable $z_1$. Thus, there was none left to produce $z_3$.

16.6 Substitute $x_B = 1 - x_T$ into the pressure equation,

$$(1 - x_T)P_{sat_B} + x_T P_{sat_T} = P$$

and solve for $x_T$,

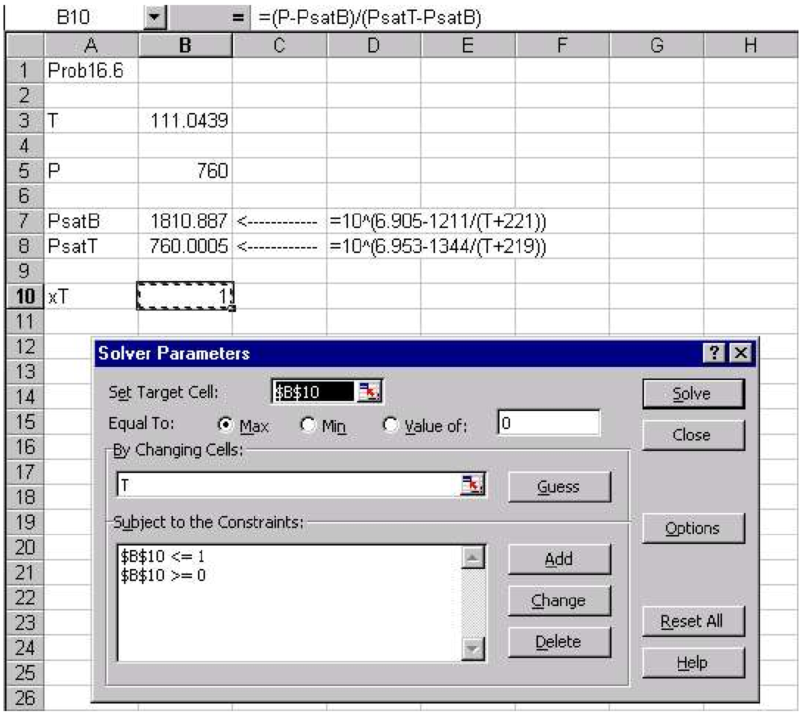
$$x_T = \frac{P - P_{sat_B}}{P_{sat_T} - P_{sat_B}} \tag{1}$$

where the partial pressures are computed as

$$P_{sat_B} = 10^{\left(6.905 - \frac{1211}{T+221}\right)}$$

$$P_{sat_B} = 10^{\left(6.953 - \frac{1344}{T+219}\right)}$$

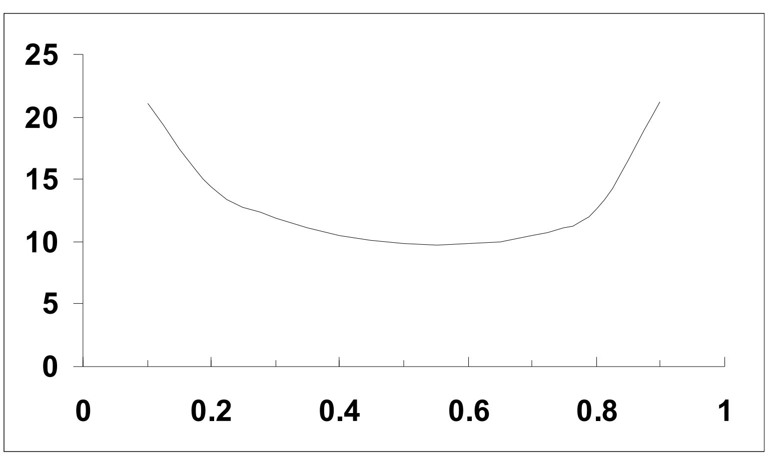
The solution then consists of maximizing Eq. 1 by varying *T* subject to the constraint that $0 \le x_T \le 1$. The Excel solver can be used to obtain the solution of $T = 111.04$.




16.7 This is a straightforward problem of varying $x_A$ in order to minimize

$$f(x_A) = \left( \frac{1}{(1 - x_A)^2} \right)^{0.6} + 5\left( \frac{1}{x_A} \right)^{0.6}$$

(*a*) The function can be plotted versus $x_A$



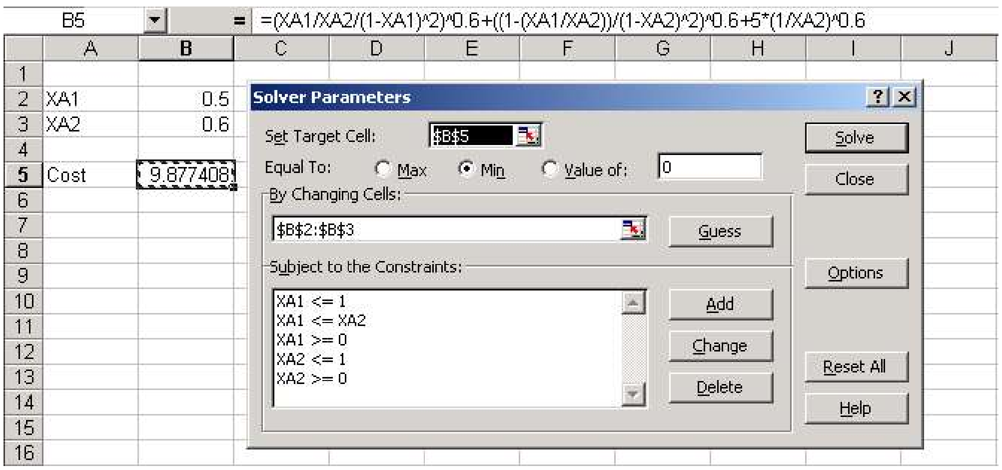

(*b*) The result indicates a minimum between 0.5 and 0.6. Using Golden Section search or a package like Excel or MATLAB yields a minimum of 0.564807.

16.8 This is a case of constrained nonlinear optimization. The conversion factors range between 0 and 1. In addition, the cost function can not be evaluated for certain combinations of XA1 and XA2. The problem is the second term,

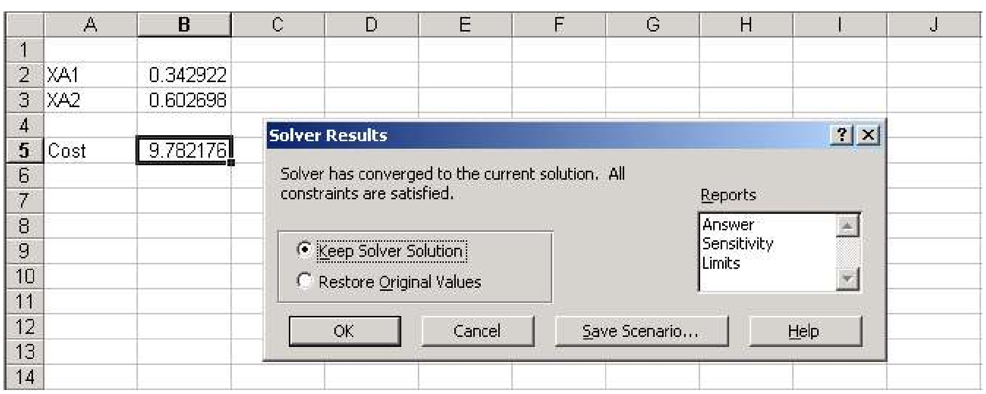
$$\left(\frac{1-\dfrac{x_{A1}}{x_{A2}}}{(1-x_{A2})^2}\right)^{0.6}$$

If $x_{A1} > x_{A2}$, the numerator will be negative and the term cannot be evaluated.

Excel Solver can be used to solve the problem:

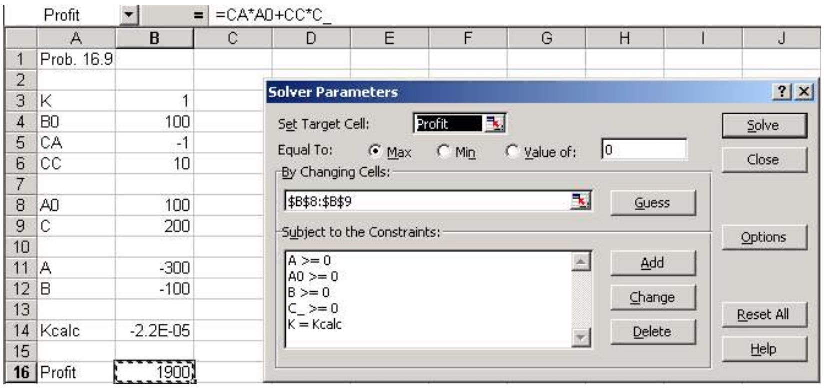



The result is



16.9 **Errata: Change $B_0$ to 100.**

This problem can be set up on Excel and the answer generated with Solver:

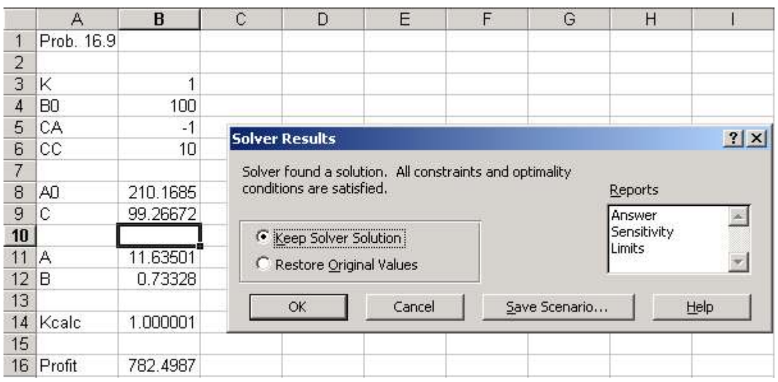


Profit = =CA*A0+CC*C_

| | A | B |
|---|---|---|
| 1 | Prob. 16.9 | |
| 2 | | |
| 3 | K | 1 |
| 4 | B0 | 100 |
| 5 | CA | -1 |
| 6 | CC | 10 |
| 7 | | |
| 8 | A0 | 100 |
| 9 | C | 200 |
| 10 | | |
| 11 | A | -300 |
| 12 | B | -100 |
| 13 | | |
| 14 | Kcalc | -2.2E-05 |
| 15 | | |
| 16 | Profit | 1900 |



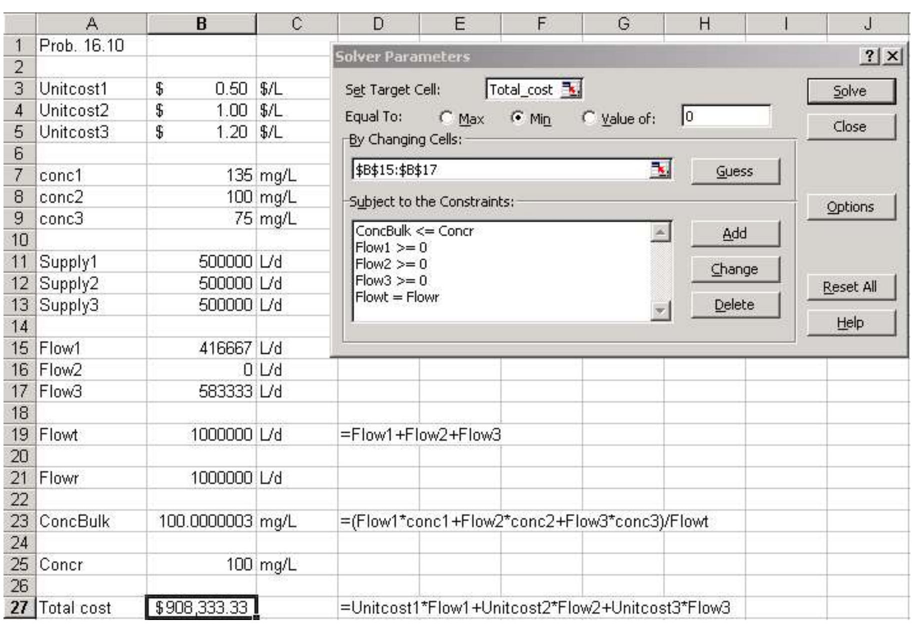
The solution is



| | A | B |
|---|---|---|
| 1 | Prob. 16.9 | |
| 2 | | |
| 3 | K | 1 |
| 4 | B0 | 100 |
| 5 | CA | -1 |
| 6 | CC | 10 |
| 7 | | |
| 8 | A0 | 210.1685 |
| 9 | C | 99.26672 |
| 10 | | |
| 11 | A | 11.63501 |
| 12 | B | 0.73328 |
| 13 | | |
| 14 | Kcalc | 1.000001 |
| 15 | | |
| 16 | Profit | 782.4987 |



16.10 The problem can be set up in Excel Solver.

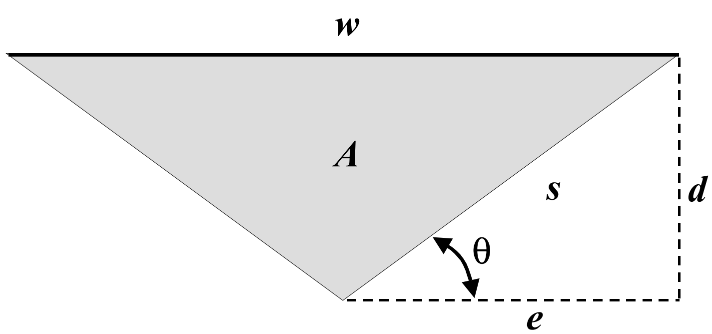


| | A | B | C | D |
|---|---|---|---|---|
| 1 | Prob. 16.10 | | | |
| 2 | | | | |
| 3 | Unitcost1 | $ 0.50 | $/L | |
| 4 | Unitcost2 | $ 1.00 | $/L | |
| 5 | Unitcost3 | $ 1.20 | $/L | |
| 6 | | | | |
| 7 | conc1 | 135 | mg/L | |
| 8 | conc2 | 100 | mg/L | |
| 9 | conc3 | 75 | mg/L | |
| 10 | | | | |
| 11 | Supply1 | 500000 | L/d | |
| 12 | Supply2 | 500000 | L/d | |
| 13 | Supply3 | 500000 | L/d | |
| 14 | | | | |
| 15 | Flow1 | 416667 | L/d | |
| 16 | Flow2 | 0 | L/d | |
| 17 | Flow3 | 583333 | L/d | |
| 18 | | | | |
| 19 | Flowt | 1000000 | L/d | =Flow1+Flow2+Flow3 |
| 20 | | | | |
| 21 | Flowr | 1000000 | L/d | |
| 22 | | | | |
| 23 | ConcBulk | 100.0000003 | mg/L | =(Flow1*conc1+Flow2*conc2+Flow3*conc3)/Flowt |
| 24 | | | | |
| 25 | Concr | 100 | mg/L | |
| 26 | | | | |
| 27 | Total cost | $908,333.33 | | =Unitcost1*Flow1+Unitcost2*Flow2+Unitcost3*Flow3 |

The solution is

| | | | | |
|---|---|---|---|---|
| 15 | Flow1 | 416667 | L/d | |
| 16 | Flow2 | 0 | L/d | |
| 17 | Flow3 | 583333 | L/d | |
| 18 | | | | |
| 19 | Flowt | 1000000 | L/d | =Flow1+Flow2+Flow3 |
| 20 | | | | |
| 21 | Flowr | 1000000 | L/d | |
| 22 | | | | |
| 23 | ConcBulk | 100.0000003 | mg/L | =(Flow1*conc1+Flow2*conc2+Flow3*conc3)/Flowt |
| 24 | | | | |
| 25 | Concr | 100 | mg/L | |
| 26 | | | | |
| 27 | Total cost | $908,333.33 | | =Unitcost1*Flow1+Unitcost2*Flow2+Unitcost3*Flow3 |

16.11




The following formulas can be developed:

$$e = \frac{w}{2} \tag{1}$$

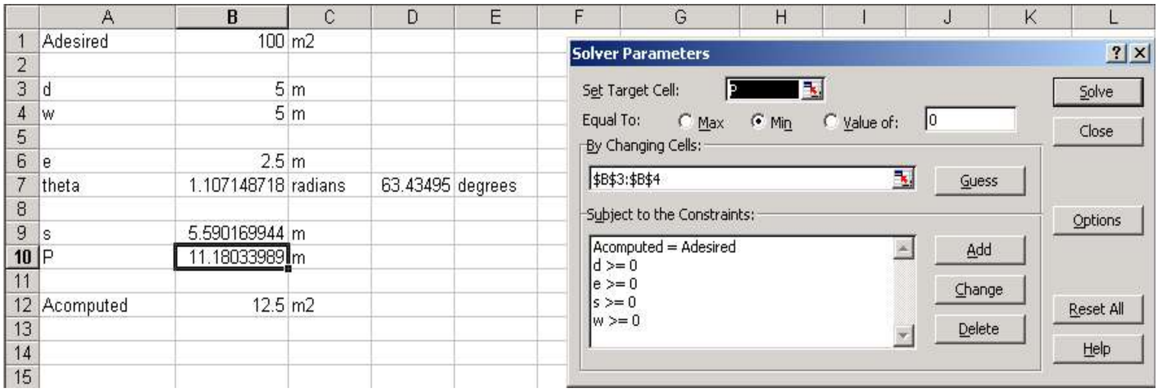
$$\theta = \tan^{-1}\frac{d}{e} \tag{2}$$

$$s = \sqrt{d^2 + e^2} \tag{3}$$

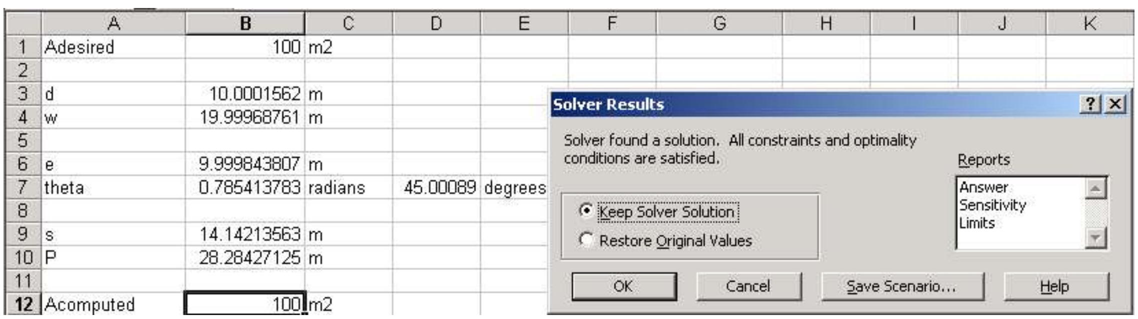
$$P = 2s \tag{4}$$

$$A = \frac{wd}{2}$$
(5)

Then the following Excel worksheet and Solver application can be set up:




Note that we have named the cells with the labels in the adjacent left columns. Our goal is to minimize the wetted perimeter by varying the depth and width. We apply positivity constraints along with the constraint that the computed area must equal the desired area. The result is




Thus, this specific application indicates that a 45° angle yields the minimum wetted perimeter.

The verification of whether this result is universal can be attained inductively or deductively. The inductive approach involves trying several different desired areas in conjunction with our solver solution. As long as the desired area is greater than 0, the result for the optimal design will be 45°.

The deductive verification involves calculus. The minimum wetted perimeter should occur when the derivative of the perimeter with respect to one of the primary dimensions (i.e., *w* or *d*) flattens out. That is, the slope is zero. In the case of the width, this would be expressed by:

$$\frac{dP}{dw} = 0$$

If the second derivative at this point is positive, the value of *w* is at a minimum. To formulate *P* in terms of *w*, substitute Eqs. 1 and 5 into 3 to yield

$$s = \sqrt{(2A/w)^2 + (w/2)^2}$$

(6)

Substitute this into Eq. 4 to give

$$P = 2\sqrt{(2A/w)^2 + (w/2)^2}$$
(7)

Differentiating Eq. 7 yields

$$\frac{dP}{dw} = \frac{-8A^2/w^3 + w/2}{\sqrt{(2A/w)^2 + (w/2)^2}} = 0$$
(8)

Therefore, at the minimum

$$-8A^2/w^3 + w/2 = 0$$
(9)

which can be solved for

$$w = 2\sqrt{A}$$
(10)

This can be substituted back into Eq. 5 to give

$$d = \sqrt{A}$$
(11)

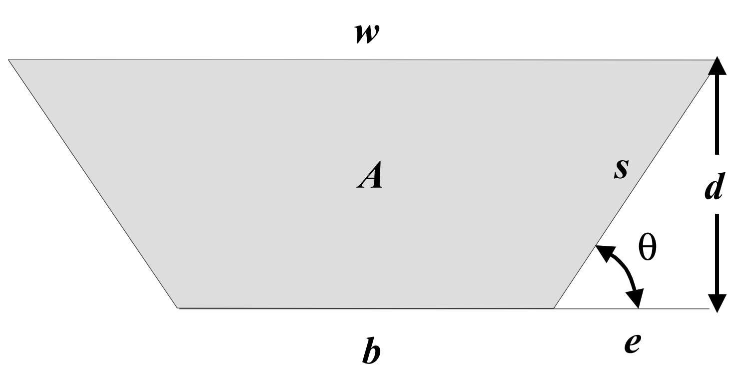
Thus, we arrive at the general conclusion that the optimal channel occurs when $w = 2d$. Inspection of Eq. 2 indicates that this corresponds to $\theta = 45^{\circ}$.

The development of the second derivative is tedious, but results in

$$\frac{d^2P}{dw^2} = 32\frac{A^2}{w^4}\sqrt{(2A/w)^2 + (w/2)^2}$$
(12)

Since A and w are by definition positive, the second derivative will always be positive.

16.12



The following formulas can be developed:

$$e = \frac{d}{\tan\theta} \tag{1}$$

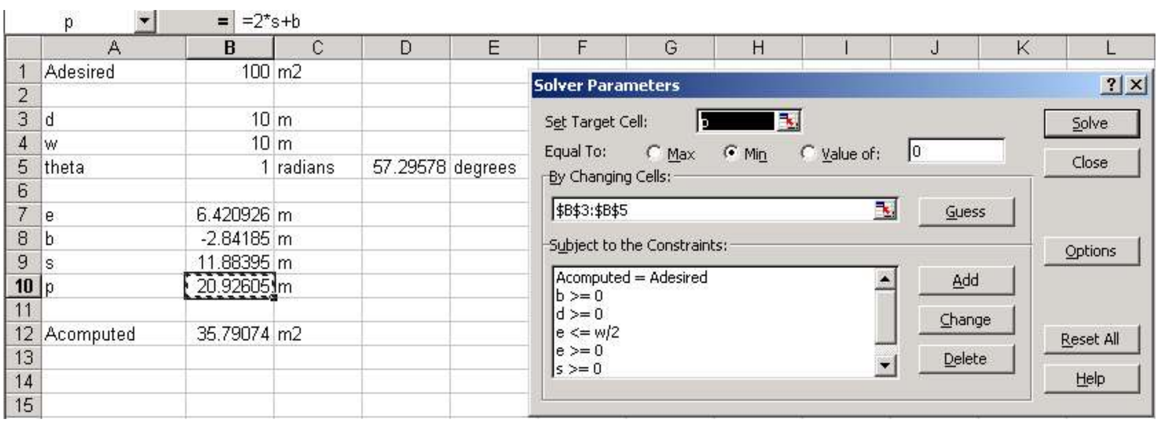
$$b = w - 2e \tag{2}$$

$$s = \sqrt{d^2 + e^2} \tag{3}$$

$$P = 2s + b \tag{4}$$

$$A = \frac{w+b}{2}d \tag{5}$$

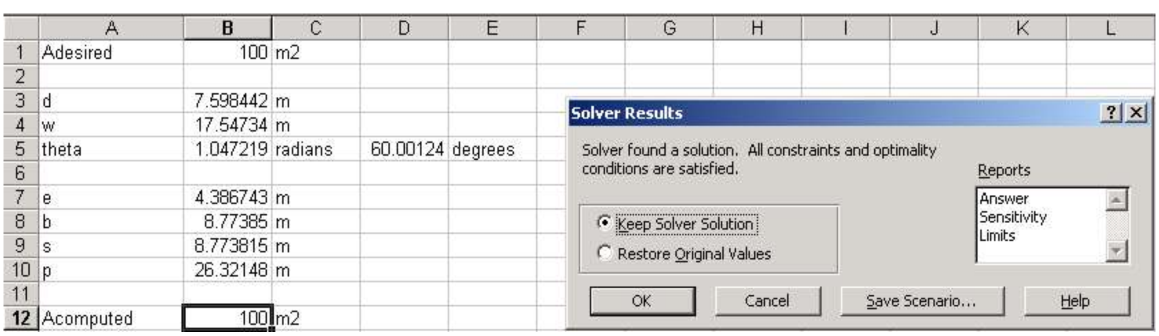
Then the following Excel worksheet and Solver application can be set up:



| | A | B | C | D | E |
|---|---|---|---|---|---|
| 1 | Adesired | 100 | m2 | | |
| 2 | | | | | |
| 3 | d | 10 | m | | |
| 4 | w | 10 | m | | |
| 5 | theta | 1 | radians | 57.29578 | degrees |
| 6 | | | | | |
| 7 | e | 6.420926 | m | | |
| 8 | b | -2.84185 | m | | |
| 9 | s | 11.88395 | m | | |
| 10 | p | 20.92605 | m | | |
| 11 | | | | | |
| 12 | Acomputed | 35.79074 | m2 | | |



Note that we have named the cells with the labels in the adjacent left columns. Our goal is to minimize the wetted perimeter by varying the depth, width and theta (the angle). We apply positivity constraints along with the constraint that the computed area must equal the desired area. We also constrain e that it cannot be greater than $w/2$. The result is



| | A | B | C | D | E |
|---|---|---|---|---|---|
| 1 | Adesired | 100 | m2 | | |
| 2 | | | | | |
| 3 | d | 7.598442 | m | | |
| 4 | w | 17.54734 | m | | |
| 5 | theta | 1.047219 | radians | 60.00124 | degrees |
| 6 | | | | | |
| 7 | e | 4.386743 | m | | |
| 8 | b | 8.77385 | m | | |
| 9 | s | 8.773815 | m | | |
| 10 | p | 26.32148 | m | | |
| 11 | | | | | |
| 12 | Acomputed | 100 | m2 | | |



Thus, this specific application indicates that a 60° angle yields the minimum wetted perimeter.

16.13

$$A_{ends} = 2\pi r^2$$

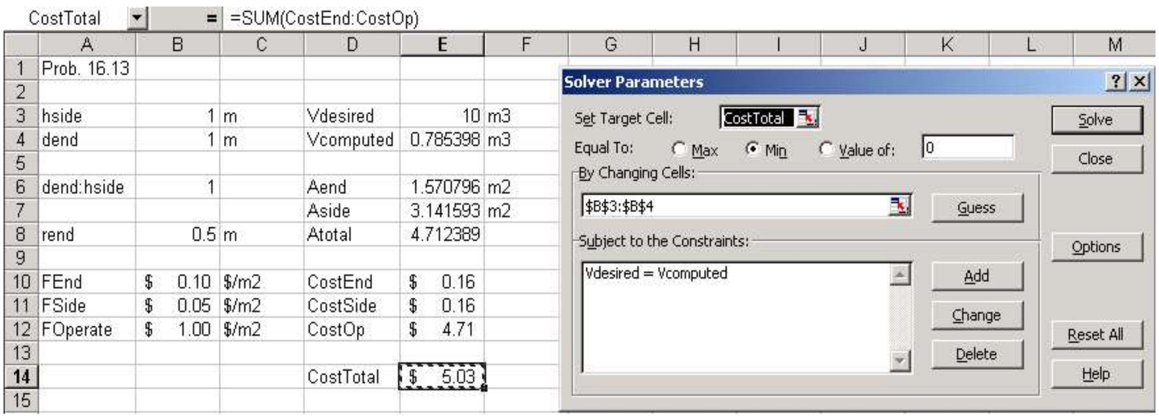
$$A_{side} = 2\pi rh$$

$$A_{total} = A_{ends} + A_{side}$$

$$V_{computed} = 2\pi r^2 h$$

$$Cost = F_{ends} A_{ends} + F_{side} A_{side} + F_{operate} A_{operate}$$

Then the following Excel worksheet and Solver application can be set up:

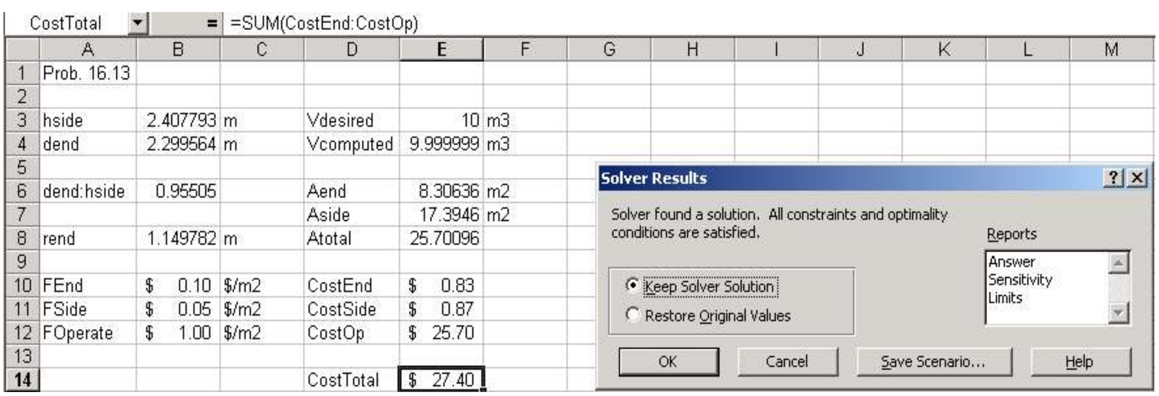


CostTotal = =SUM(CostEnd:CostOp)

| | A | B | C | D | E | F |
|---|---|---|---|---|---|---|
| 1 | Prob. 16.13 | | | | | |
| 2 | | | | | | |
| 3 | hside | 1 | m | Vdesired | 10 | m3 |
| 4 | dend | 1 | m | Vcomputed | 0.785398 | m3 |
| 5 | | | | | | |
| 6 | dend:hside | 1 | | Aend | 1.570796 | m2 |
| 7 | | | | Aside | 3.141593 | m2 |
| 8 | rend | 0.5 | m | Atotal | 4.712389 | |
| 9 | | | | | | |
| 10 | FEnd | $ 0.10 | $/m2 | CostEnd | $ 0.16 | |
| 11 | FSide | $ 0.05 | $/m2 | CostSide | $ 0.16 | |
| 12 | FOperate | $ 1.00 | $/m2 | CostOp | $ 4.71 | |
| 13 | | | | | | |
| 14 | | | | CostTotal | $ 5.03 | |
| 15 | | | | | | |



which results in the following solution:



CostTotal = =SUM(CostEnd:CostOp)

| | A | B | C | D | E | F |
|---|---|---|---|---|---|---|
| 1 | Prob. 16.13 | | | | | |
| 2 | | | | | | |
| 3 | hside | 2.407793 | m | Vdesired | 10 | m3 |
| 4 | dend | 2.299564 | m | Vcomputed | 9.999999 | m3 |
| 5 | | | | | | |
| 6 | dend:hside | 0.95505 | | Aend | 8.30636 | m2 |
| 7 | | | | Aside | 17.3946 | m2 |
| 8 | rend | 1.149782 | m | Atotal | 25.70096 | |
| 9 | | | | | | |
| 10 | FEnd | $ 0.10 | $/m2 | CostEnd | $ 0.83 | |
| 11 | FSide | $ 0.05 | $/m2 | CostSide | $ 0.87 | |
| 12 | FOperate | $ 1.00 | $/m2 | CostOp | $ 25.70 | |
| 13 | | | | | | |
| 14 | | | | CostTotal | $ 27.40 | |



16.14 Excel Solver gives: $x = 0.5$, $y = 0.8$ and $f_{min} = -0.85$.

16.15

$$\text{Cost} = 4(.0025)(275)(\pi)dt + 2d$$

Constraints

$$\frac{3000}{\pi dt} - 550 \le 0$$

$$\frac{3000}{\pi dt} - \frac{\pi^2 \times 900{,}000 \times (d^2+t^2)}{8(275^2)} \le 0$$

$$1 \le d \le 10$$
$$0.1 \le t \le 1$$

using Excel Solver gives

$d = 6.113922$ cm
$t = 0.283981$ cm

$\text{cost}_{min} = 27.2278$

16.16 using Excel solver

$t_c = 3.391236$

$O_c = 3.322621$

16.17 using Excel Solver

$x = 0.57919$
$y = 10.29728$

$C_{max} = 9.018862$

16.18 Excel Solver Gives

a) $P = B + 2H$

$$Q = \frac{(BH)^{5/3}(.003)^{1/2}}{.035\,(B+2H)^{2/3}} = 1$$

$B = 40.6992$
$H = 20.3496$
$P_{max} = 81.39839$

b) same B, H, and $P_{max}$ as a)

cost = \$ 86891.14

c) note that above constraint equation can be written as,

$$\underbrace{bh}_{A} = \text{constant} \times \underbrace{(B+2H)}_{P}^{2/5}$$

$\therefore$ A and P are minimized at the same time

thus the same B and H satisfy both a) and b)

16.19

$$100 = \frac{\pi^3 (29) r^4}{4L^2} \qquad 35 = \pi r^2 L$$

$$4L^2 = \pi^3 (.29) r^4 \qquad L = \frac{35}{\pi r^2}$$

$$L = 1.499 r_2$$

$$r = 1.65\ m$$

$$L = 4.08\ m$$

16.20 $I_1 = 4$ $I_2 = 2$ $I_3 = 2$ $I_4 = 0$ $I_5 = 2$ $P = 80$

16.21 Using Loop and Node Balances

$$i_1 + i_2 = 10$$
$$i_3 = i_1 + i_5$$
$$i_2 = i_4 + i_5$$

$$i_1 R_1 - i_5 R_5 - i_2 R_2 = 0$$
$$i_3 R_3 - i_4 R_4 + i_5 R_5 = 0$$

$$\text{power} = i_2^2 R_2 + i_3^2 R_3$$

Using Excel Solver

$R_1 = 10.0$
$R_2 = 8.0$
$R_3 = 8.0$
$R_4 = 10$
$R_5 = 1$

$i_1 = 4.5$
$i_2 = 5.5$
$i_3 = 5.5$
$i_4 = 4.5$
$i_5 = 1$

Power = 484

16.22

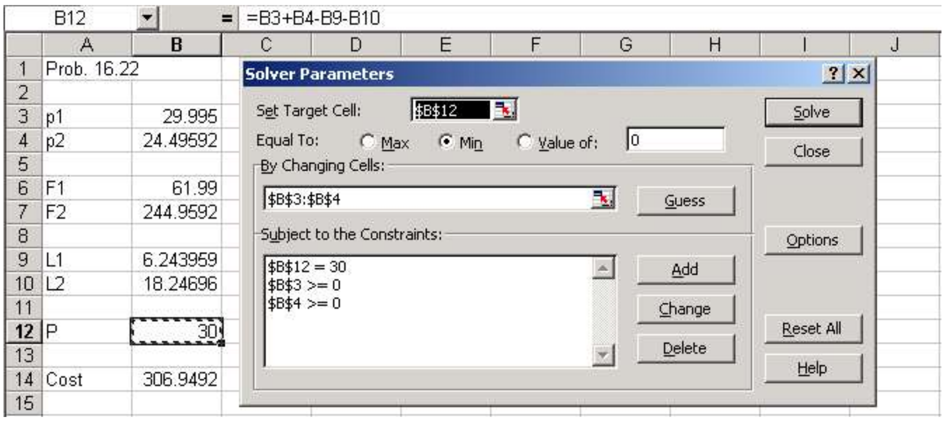
Total cost is

$$C = 2p_1 + 10p_2 + 2$$

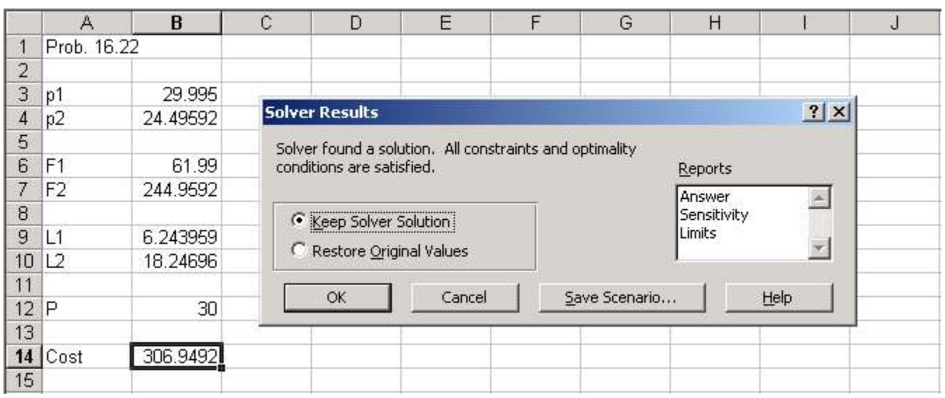
Total power delivered is

$$P = 0.6p_1 + 0.4p_2$$

Using the Excel Solver:




which yields the solution




16.23 This is a trick question. Because of the presence of $(1 - s)$ in the denominator, the function will experience a division by zero at the maximum. This can be rectified by merely canceling the $(1 - s)$ terms in the numerator and denominator to give

$$T = \frac{15s}{4s^2 - 3s + 4}$$

Any of the optimizers described in this section can then be used to determine that the maximum of $T$ = 3 occurs at $s$ = 1.

16.24 Using Solver from Excel

| W | V | $D_{min}$ |
|---|---|---|
| 12,000 | 483.66 | 2339.23 |
| 13,000 | 503.41 | 2534.17 |
| 14,000 | 522.41 | 2729.10 |
| 15,000 | 540.74 | 2924.04 |
| 16,000 | 558.48 | 3118.97 |
| 17,000 | 575.67 | 3313.91 |
| 18,000 | 592.36 | 3508.85 |

16.25 Using Solver from Excel

$d = 0.87358$

$D = 2$

$N = 3$

$W_{min} = 3.2052$

16.26 Solver gives

$x = 0.786151$

$f_{max} = 0.30028$

16.27 An LP formulation for this problem can be set up as

Maximize $C = 0.15X + 0.025Y + 0.05Z$ {Minimize cost}

subject to

$X + Y + Z \geq 6$ {Performance constraint}
$X + Y < 3$ {Safety constraint}
$X - Y \geq 0$ {$X$-$Y$ Relationship constraint}
$Z - 0.5Y \geq 0$ {$Y$-$Z$ Relationship constraint}

(*b*) An Excel spreadsheet can be set up to solve the problem as

The Solver can be called and set up as

The resulting solution is

16.28

$$\tau = \frac{Tc}{J} \Rightarrow 20000000 = \frac{500 r_o}{\pi/2\,(r_o^4 - r_i^4)}$$

$$r_i = \sqrt[4]{r_o^4 - 1.5915 \text{ x } 10^{-5} r_o}$$

$$\phi = \frac{TL}{JG} \Rightarrow 2.5\left(\frac{\pi}{180}\right) = \frac{500(5)}{77 \text{ x } 10^9 \left(\frac{\pi}{2}\right)\left(r_o^4 - r_i^4\right)}$$

$$r_i = \sqrt[4]{r_o^4 - 2.8422 \text{ x } 10^{-7}}$$

$$r_o = 29.76 \text{ mm}$$

$$r_i = 23.61 \text{ mm} \qquad \text{but } r_o - r_i \geq 8 \text{ mm}$$

$$\therefore r_o = 29.76 \text{ mm}, \; r_i = 21.76 \text{ mm}$$

16.29

$$L = \frac{\text{Re}\,\mu}{\rho V} = 0.567$$

$$h = \frac{2F}{C_D \rho V^2 b} = .0779$$

$$h = L = 0.567 \text{ cm}$$

| | A | B | C | D | E | F |
|---|---|---|---|---|---|---|
| 1 | | X | Y | Z | Total | Constraint |
| 2 | Amount | 1.5 | 1.5 | 3 | | |
| 3 | Performance | 1 | 1 | 1 | 6 | 6 |
| 4 | Safety | 1 | 1 | 0 | 3 | 3 |
| 5 | X-Y | 1 | -1 | 0 | 0 | 0 |
| 6 | Z-Y | 0 | -0.5 | 1 | 2.25 | 0 |
| 7 | Cost | 0.15 | 0.025 | 0.05 | 0.4125 | |

Set target cell: E7

Equal to ❍ max ● min ❍ value of 0

By changing cells
B2:D2

Subject to constraints:
E3≥F3
E4≤F4
E5≥F5
E6≥F6

| | A | B | C | D | E | F |
|---|---|---|---|---|---|---|

| | | | | | | |
|---|---|---|---|---|---|---|
| 1 | | X | Y | Z | Total | Constraint |
| 2 | Amount | 0 | 0 | 0 | | |
| 3 | Performance | 1 | 1 | 1 | 0 | 6 |
| 4 | Safety | 1 | 1 | 0 | 0 | 3 |
| 5 | X-Y | 1 | -1 | 0 | 0 | 0 |
| 6 | Z-Y | 0 | -0.5 | 1 | 0 | 0 |
| 7 | Cost | 0.15 | 0.025 | 0.05 | 0 | |

| | A | B | C | D | E | F | G |
|---|---|---|---|---|---|---|---|
| 1 | | Z1 | Z2 | Z3 | W | total | constraint |
| 2 | amount | 4000 | 3500 | 0 | 500 | | |
| 3 | amount X | 1 | 1 | 0 | 0 | 7500 | 7500 |
| 4 | amount Y | 2.5 | 0 | 1 | 0 | 10000 | 10000 |
| 5 | amount W | 1 | -1 | -1 | -1 | 0 | 0 |
| 6 | profit | 2500 | -50 | 200 | -300 | 9675000 | |

Set target cell: F6

Equal to ● max ❍ min ❍ value of 0

By changing cells

B2:E2

Subject to constraints:

B2≥0

C2≥0

F3≤G3

F4≤G4

F5=G5

| | A | B | C | D | E | F | G |
|---|---|---|---|---|---|---|---|
| 1 | | Z1 | Z2 | Z3 | W | total | constraint |
| 2 | amount | 0 | 0 | 0 | 0 | | |
| 3 | amount X | 1 | 1 | 0 | 0 | 0 | 7500 |
| 4 | amount Y | 2.5 | 0 | 1 | 0 | 0 | 10000 |
| 5 | amount W | 1 | -1 | -1 | -1 | 0 | 0 |
| 6 | profit | 2500 | -50 | 200 | -300 | 0 | |

**Microsoft Excel 5.0c Sensitivity Report**
**Worksheet: [PROB1605.XLS]Sheet3**
**Report Created: 12/12/97 9:47**

Changing Cells

| Cell | Name | Final Value | Reduced Cost | Objective Coefficient | Allowable Increase | Allowable Decrease |
|---|---|---|---|---|---|---|
| $B$2 | amount Product 1 | 150 | 0 | 30 | 0.833333333 | 2.5 |
| $C$2 | amount Product 2 | 125 | 0 | 30 | 1.666666667 | 1 |
| $D$2 | amount Product 3 | 175 | 0 | 35 | 35 | 5 |

Constraints

| Cell | Name | Final Value | Shadow Price | Constraint R.H. Side | Allowable Increase | Allowable Decrease |
|---|---|---|---|---|---|---|
| $E$3 | material total | 3000 | 0.625 | 3000 | 1E+30 | 1E+30 |
| $E$4 | time total | 55 | 12.5 | 55 | 1E+30 | 1E+30 |
| $E$5 | storage total | 450 | 26.25 | 450 | 1E+30 | 1E+30 |

| | A | B | C | D | E | F |
|---|---|---|---|---|---|---|
| 1 | | Product 1 | Product 2 | Product 3 | total | constraint |
| 2 | amount | 150 | 125 | 175 | | |
| 3 | material | 5 | 4 | 10 | 3000 | 3000 |

| | | | | | | |
|---|---|---|---|---|---|---|
| 4 | time | 0.05 | 0.1 | 0.2 | 55 | 55 |
| 5 | storage | 1 | 1 | 1 | 450 | 450 |
| 6 | profit | 30 | 30 | 35 | 14375 | |

Set target cell: E6
Equal to ● max ❍ min ❍ value of 0
By changing cells
B2:D2
Subject to constraints:
E3≤F3
E4≤F4
E5≤F5

| | A | B | C | D | E | F |
|---|---|---|---|---|---|---|
| 1 | | Product 1 | Product 2 | Product 3 | total | constraint |
| 2 | amount | 0 | 0 | 0 | | |
| 3 | material | 5 | 4 | 10 | 0 | 3000 |
| 4 | time | 0.05 | 0.1 | 0.2 | 0 | 55 |
| 5 | storage | 1 | 1 | 1 | 0 | 450 |
| 6 | profit | 30 | 30 | 35 | 0 | |

| Basis | P | x1 | x2 | x3 | S1 | S2 | S3 | Solution | Intercept |
|---|---|---|---|---|---|---|---|---|---|
| P | 1 | -30 | -30 | -35 | 0 | 0 | 0 | 0 | |
| S1 | 0 | 5 | 4 | 10 | 1 | 0 | 0 | 3000 | 300 |
| S2 | 0 | 0.05 | 0.1 | 0.2 | 0 | 1 | 0 | 55 | 275 |
| S3 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 450 | 450 |

| Basis | P | x1 | x2 | x3 | S1 | S2 | S3 | Solution | Intercept |
|---|---|---|---|---|---|---|---|---|---|
| P | 1 | -21.25 | -12.5 | 0 | 0 | 175 | 0 | 9625 | |
| S1 | 0 | 2.5 | -1 | 0 | 1 | -50 | 0 | 250 | 100 |
| x3 | 0 | 0.25 | 0.5 | 1 | 0 | 5 | 0 | 275 | 1100 |
| S3 | 0 | 0.75 | 0.5 | 0 | 0 | -5 | 1 | 175 | 233.3333 |

| Basis | P | x1 | x2 | x3 | S1 | S2 | S3 | Solution | Intercept |
|---|---|---|---|---|---|---|---|---|---|
| P | 1 | 0 | -21 | 0 | 8.5 | -250 | 0 | 11750 | |
| x1 | 0 | 1 | -0.4 | 0 | 0.4 | -20 | 0 | 100 | -250 |
| x3 | 0 | 0 | 0.6 | 1 | -0.1 | 10 | 0 | 250 | 416.6667 |
| S3 | 0 | 0 | 0.8 | 0 | -0.3 | 10 | 1 | 100 | 125 |

| Basis | P | x1 | x2 | x3 | S1 | S2 | S3 | Solution |
|---|---|---|---|---|---|---|---|---|
| P | 1 | 0 | 0 | 0 | 0.625 | 12.5 | 26.25 | 14375 |
| x1 | 0 | 1 | 0 | 0 | 0.25 | -15 | 0.5 | 150 |
| x3 | 0 | 0 | 0 | 1 | 0.125 | 2.5 | -0.75 | 175 |
| x2 | 0 | 0 | 1 | 0 | -0.375 | 12.5 | 1.25 | 125 |

## Chapter 20

20.1 after transforming the data, the following line is obtained

$$\log k = -0.83 + 0.422 \log f$$

or

$$k = 0.1479\, X^{0.422}$$

20.2 Try linear

$$C = 1333.761 + 2.655963\, T$$

with $S_{y/x} = 34.7$

However for quadratic

$$C = 1311.5 + 1.7807\, T + 0.010941\, T^2$$

reduces $S_{y/x}$ to 19.98

∴ use quadratic for high accuracy

20.3 use linear and $x_0 = 15$ and $x_1 = 20$

gives $DO_{18} = 8.56$

quadratic with $x_0 = 15$ $x_1 = 20$ and $x_2 = 25$

gives $DO_{18} = 8.548$

cubic with $x_0 = 10$ $x_1 = 15$ $x_2 = 20$ $x_3 = 25$

gives $DO_{18} = 8.5368$

5th order using all data

gives $DO_{18} = 8.535994$

cubic is a good compromise that gives good accuracy and reasonable computation effort.

20.4

| T | Dissolved Oxygen |
|---|---|
| 5 | 10.5 |
| 10 | 9.2 |
| 15 | 8.2 |

the quadratic $DO = a_0 + a_1 T + a_2 T^2$ must satisfy

$$\begin{bmatrix} 1 & 5 & 25 \\ 1 & 10 & 100 \\ 1 & 15 & 225 \end{bmatrix} \begin{pmatrix} a_0 \\ a_1 \\ a_2 \end{pmatrix} = \begin{matrix} 10.5 \\ 9.2 \\ 8.2 \end{matrix}$$

solving gives

$$DO = 12.1 - 0.35T + 0.006T^2$$

and $DO_8 = 9.684$

20.5

$$DO = a_0 + a_1 T + a_2 C$$

Solve Normal Equations

$$\begin{bmatrix} 18 & 315 & 180000 \\ 315 & 6825 & 3150000 \\ 180000 & 3150000 & 3\times 10^9 \end{bmatrix} \begin{Bmatrix} a_0 \\ a_1 \\ a_2 \end{Bmatrix} = \begin{Bmatrix} 160.2 \\ 2550.5 \\ 1496000 \end{Bmatrix}$$

gives

$$a_0 = 13.1567$$
$$a_1 = -0.1928$$
$$a_2 = -8.83\times 10^{-5}$$

$$DO_{12,15000} = 13.1567 - 0.1928(12) - 8.83\times 10^{-5}(15000)$$

$$= 9.5186$$

20.6

$$Os = a_0 + a_1 T + a_2 T^2 + a_3 T^3 + a_4 C$$

$$Z = \begin{bmatrix} 1 & 5 & 25 & 125 & 0 \\ 1 & 10 & 100 & 1000 & 0 \\ 1 & 15 & 225 & 3375 & 0 \\ 1 & 20 & 400 & 8000 & 0 \\ 1 & 25 & 625 & 15625 & 0 \\ 1 & 30 & 900 & 27000 & 0 \\ 1 & 5 & 25 & 125 & 10000 \\ 1 & 10 & 100 & 1000 & 10000 \\ 1 & 15 & 225 & 3375 & 10000 \\ 1 & 20 & 400 & 8000 & 10000 \\ 1 & 25 & 625 & 15625 & 10000 \\ 1 & 30 & 900 & 27000 & 10000 \\ 1 & 5 & 25 & 125 & 20000 \\ 1 & 10 & 100 & 1000 & 20000 \\ 1 & 15 & 225 & 3375 & 20000 \\ 1 & 20 & 400 & 8000 & 20000 \\ 1 & 25 & 625 & 15625 & 20000 \\ 1 & 30 & 900 & 27000 & 20000 \end{bmatrix} \quad Y = \begin{bmatrix} 12.8 \\ 11.3 \\ 10.0 \\ 9.0 \\ 8.2 \\ 7.4 \\ 11.6 \\ 10.3 \\ 9.1 \\ 8.2 \\ 7.4 \\ 6.8 \\ 10.5 \\ 9.2 \\ 8.2 \\ 7.4 \\ 6.7 \\ 6.1 \end{bmatrix}$$

$$[Z^T][Z]\{A\} = \{Z^T\}\{Y\}$$

$$\begin{bmatrix} 18 & 315 & 6.825\times10^{3} & 1.654\times10^{5} & 1.8\times10^{5} \\ 315 & 6.825\times10^{3} & 1.634\times10^{5} & 4.266\times10^{6} & 3.15\times10^{6} \\ 6.825\times10^{3} & 1.654\times10^{5} & 4.266\times10^{6} & 1.144\times10^{8} & 6.825\times10^{7} \\ 1.654\times10^{5} & 4.266\times10^{6} & 1.144\times10^{8} & 3.149\times10^{9} & 1.654\times10^{9} \\ 1.8\times10^{5} & 3.15\times10^{6} & 6.825\times10^{7} & 1.654\times10^{9} & 3.\times10^{9} \end{bmatrix} \begin{Bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \end{Bmatrix} = \begin{Bmatrix} 160.2 \\ 2.551\times10^{3} \\ 5.214\times10^{4} \\ 1.221\times10^{6} \\ 1.496\times10^{6} \end{Bmatrix}$$

$\{A\} = [Z^T Z]^{-1}\{Z^T Y\}$ gives $a_0 = 14.217$, $a_1 = -0.373$, $a_2 = 7.143\times10^{-3}$, $a_3 = -6.6667\times10^{-5}$, $a_4 = -8.833\times10^{-5}$

$$DO_{30,20000} = 14.217 - 0.373(30) + 7.143\times10^{-3}(30^2) - 6.6667\times10^{-5}(30^3) - 8.833\times10^{-5}(20000)$$

$$= 5.8915$$

20.7 $y = 2.410326 + 1.055163x$

$S_{xy} = 12.97$

$r^2 = 0.794$

$y(30) = 34.065$

20.8 Regression gives

$P = 8100.47 + 30.3164\,T_{°C}$

$r^2 = 0.999$

$R = \left(\frac{P}{T}\right)\frac{V}{n}$

$P/T = 30.3164$

$n = \frac{1\text{ kg}}{28\text{ gm/mole}}$

$R = 30.3164\left(\frac{10}{10^3/28}\right)$

$= 8.487$ which is close to $8.314$ J/gmole

20.9 using all data obtain $4^{th}$ order interpolating poly

$y = a_0 + a_1T + a_2T^2 + a_3T^3 + a_4T^4$

where $a_0 = 4.966$
$a_1 = -3.551\times10^{-2}$
$a_2 = 8.273\times10^{-5}$
$a_3 = -7.811\times10^{-8}$
$a_4 = 2.604\times10^{-11}$

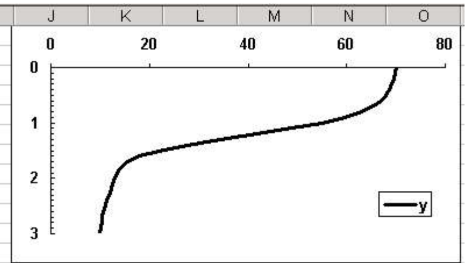
$y(750) = 0.153765$

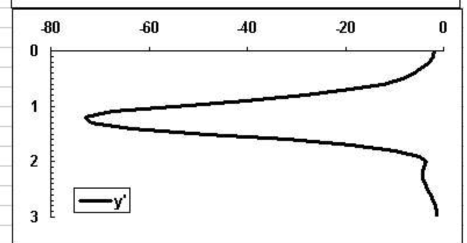
A VBA code to do this with the computer is

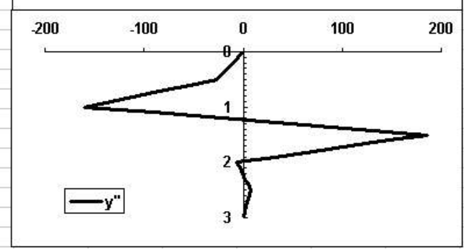
```
Sub Splines()
Dim i As Integer, n As Integer
Dim x(100) As Single, y(100) As Single, xu As Single, yu As Single
Dim xint(100) As Single
Dim dy As Single, d2y As Single
Sheets("Sheet1").Select
Range("a5").Select
n = ActiveCell.Row
Selection.End(xlDown).Select
n = ActiveCell.Row - n
Range("a5").Select
For i = 0 To n
  x(i) = ActiveCell.Value
  ActiveCell.Offset(0, 1).Select
  y(i) = ActiveCell.Value
  ActiveCell.Offset(1, -1).Select
Next i
Range("d5").Select
nint = ActiveCell.Row
Selection.End(xlDown).Select
nint = ActiveCell.Row - nint
Range("d5").Select
For i = 0 To nint
  xint(i) = ActiveCell.Value
  ActiveCell.Offset(1, 0).Select
Next i
Range("e5").Select
For i = 0 To nint
  Call Spline(x(), y(), n, xint(i), yu, dy, d2y)
  ActiveCell.Value = yu
  ActiveCell.Offset(0, 1).Select
  ActiveCell.Value = dy
  ActiveCell.Offset(0, 1).Select
  ActiveCell.Value = d2y
  ActiveCell.Offset(1, -2).Select
Next i
Range("a5").Select
End Sub

Sub Spline(x, y, n, xu, yu, dy, d2y)
Dim e(10) As Single, f(10) As Single, g(10) As Single, r(10) As Single, d2x(10) As Single
Call Tridiag(x, y, n, e, f, g, r)
Call Decomp(e(), f(), g(), n - 1)
Call Substit(e(), f(), g(), r(), n - 1, d2x())
Call Interpol(x, y, n, d2x(), xu, yu, dy, d2y)
End Sub

Sub Tridiag(x, y, n, e, f, g, r)
Dim i As Integer
f(1) = 2 * (x(2) - x(0))
g(1) = x(2) - x(1)
r(1) = 6 / (x(2) - x(1)) * (y(2) - y(1))
r(1) = r(1) + 6 / (x(1) - x(0)) * (y(0) - y(1))
For i = 2 To n - 2
  e(i) = x(i) - x(i - 1)
  f(i) = 2 * (x(i + 1) - x(i - 1))
  g(i) = x(i + 1) - x(i)
  r(i) = 6 / (x(i + 1) - x(i)) * (y(i + 1) - y(i))
  r(i) = r(i) + 6 / (x(i) - x(i - 1)) * (y(i - 1) - y(i))
Next i
e(n - 1) = x(n - 1) - x(n - 2)
f(n - 1) = 2 * (x(n) - x(n - 2))
r(n - 1) = 6 / (x(n) - x(n - 1)) * (y(n) - y(n - 1))
r(n - 1) = r(n - 1) + 6 / (x(n - 1) - x(n - 2)) * (y(n - 2) - y(n - 1))
End Sub

Sub Interpol(x, y, n, d2x, xu, yu, dy, d2y)
Dim i As Integer, flag As Integer
Dim c1 As Single, c2 As Single, c3 As Single, c4 As Single
Dim t1 As Single, t2 As Single, t3 As Single, t4 As Single
flag = 0
i = 1
Do
  If xu >= x(i - 1) And xu <= x(i) Then
```

```
    c1 = d2x(i - 1) / 6 / (x(i) - x(i - 1))
    c2 = d2x(i) / 6 / (x(i) - x(i - 1))
    c3 = y(i - 1) / (x(i) - x(i - 1)) - d2x(i - 1) * (x(i) - x(i - 1)) / 6
    c4 = y(i) / (x(i) - x(i - 1)) - d2x(i) * (x(i) - x(i - 1)) / 6
    t1 = c1 * (x(i) - xu) ^ 3
    t2 = c2 * (xu - x(i - 1)) ^ 3
    t3 = c3 * (x(i) - xu)
    t4 = c4 * (xu - x(i - 1))
    yu = t1 + t2 + t3 + t4
    t1 = -3 * c1 * (x(i) - xu) ^ 2
    t2 = 3 * c2 * (xu - x(i - 1)) ^ 2
    t3 = -c3
    t4 = c4
    dy = t1 + t2 + t3 + t4
    t1 = 6 * c1 * (x(i) - xu)
    t2 = 6 * c2 * (xu - x(i - 1))
    d2y = t1 + t2
    flag = 1
  Else
    i = i + 1
  End If
  If i = n + 1 Or flag = 1 Then Exit Do
Loop
If flag = 0 Then
  MsgBox "outside range"
  End
End If
End Sub

Sub Decomp(e, f, g, n)
Dim k As Integer
For k = 2 To n
  e(k) = e(k) / f(k - 1)
  f(k) = f(k) - e(k) * g(k - 1)
Next k
End Sub

Sub Substit(e, f, g, r, n, x)
Dim k As Integer
For k = 2 To n
  r(k) = r(k) - e(k) * r(k - 1)
Next k
x(n) = r(n) / f(n)
For k = n - 1 To 1 Step -1
  x(k) = (r(k) - g(k) * x(k + 1)) / f(k)
Next k
End Sub
```

| | A | B | C | D | E | F | G | H |
|---|---|---|---|---|---|---|---|---|
| 1 | Problem 20.10 | | | | | | | RUN |
| 2 | | | | | | | | |
| 3 | Data: | | | Interpolate: | | | | |
| 4 | x | y | | x | y | y' | y'' | |
| 5 | 0 | 70 | | 0 | 70 | -1.83076 | 0 | |
| 6 | 0.5 | 68 | | 0.1 | 69.80825 | -2.09107 | -5.20615 | |
| 7 | 1 | 55 | | 0.2 | 69.56443 | -2.872 | -10.4123 | |
| 8 | 1.5 | 22 | | 0.3 | 69.21649 | -4.17353 | -15.6185 | |
| 9 | 2 | 13 | | 0.4 | 68.71237 | -5.99569 | -20.8246 | |
| 10 | 2.5 | 11 | | 0.5 | 68 | -8.33846 | -26.0308 | |
| 11 | 3 | 10 | | 0.6 | 66.99139 | -12.28 | -52.8 | |
| 12 | | | | 0.7 | 65.45477 | -18.8985 | -79.5692 | |
| 13 | | | | 0.8 | 63.12246 | -28.1939 | -106.338 | |
| 14 | | | | 0.9 | 59.72677 | -40.1662 | -133.108 | |
| 15 | | | | 1 | 55 | -54.8154 | -159.877 | |
| 16 | | | | 1.1 | 48.83421 | -67.3489 | -90.7938 | |
| 17 | | | | 1.2 | 41.76049 | -72.9742 | -21.7107 | |
| 18 | | | | 1.3 | 34.46967 | -71.6911 | 47.37228 | |
| 19 | | | | 1.4 | 27.65256 | -63.4997 | 116.4554 | |
| 20 | | | | 1.5 | 22 | -48.4 | 185.5385 | |
| 21 | | | | 1.6 | 18.02375 | -31.7643 | 147.1754 | |
| 22 | | | | 1.7 | 15.51926 | -18.9649 | 108.8123 | |
| 23 | | | | 1.8 | 14.10289 | -10.0018 | 70.44926 | |
| 24 | | | | 1.9 | 13.39102 | -4.87508 | 32.08617 | |
| 25 | | | | 2 | 13 | -3.58461 | -6.27692 | |
| 26 | | | | 2.1 | 12.61477 | -4.07385 | -3.5077 | |
| 27 | | | | 2.2 | 12.19446 | -4.28615 | -0.73846 | |
| 28 | | | | 2.3 | 11.76677 | -4.22154 | 2.030768 | |
| 29 | | | | 2.4 | 11.35938 | -3.88 | 4.800004 | |
| 30 | | | | 2.5 | 11 | -3.26154 | 7.569232 | |
| 31 | | | | 2.6 | 10.70917 | -2.58031 | 6.055387 | |
| 32 | | | | 2.7 | 10.47889 | -2.05046 | 4.541538 | |
| 33 | | | | 2.8 | 10.29403 | -1.672 | 3.027694 | |
| 34 | | | | 2.9 | 10.13945 | -1.44492 | 1.513845 | |
| 35 | | | | 3 | 10 | -1.36923 | 0 | |
| 36 | | | | | | | | |

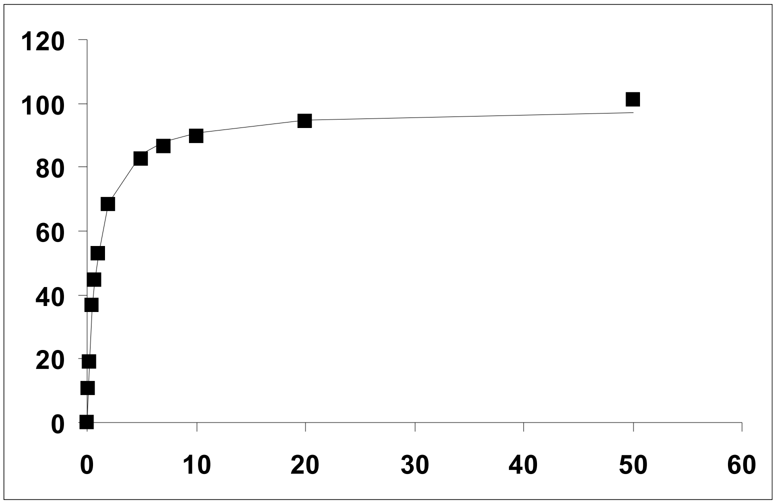


20.11 The best fit equation can be determined by nonlinear regression as

$$[B] = \frac{98.84[F]}{0.8766 + [F]}$$



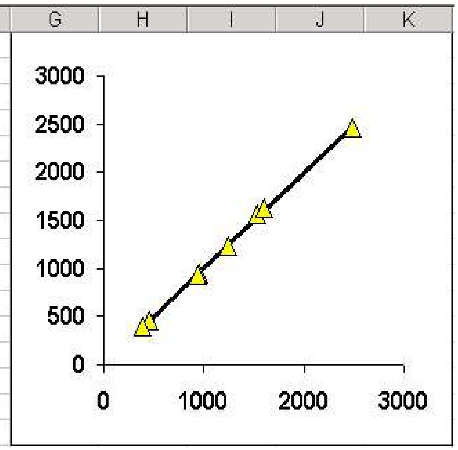

Disregarding the point (0, 0), The $r^2$ can be computed as

$S_t = 9902.274$
$S_r = 23.36405$

$$r^2 = \frac{9902.274 - 23.364}{9902.274} = 0.9976$$

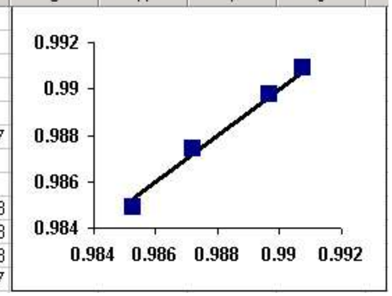
20.12 The Excel Solver can be used to develop a nonlinear regression to fit the parameters. The result (along with a plot of –dA/dt calculated with the model versus the data estimates) are shown below. Note that the 1:1 line is also displayed on the plot.

| | A | B | C | D | E |
|---|---|---|---|---|---|
| 1 | Prob 20.12 | | | | |
| 2 | | | | | |
| 3 | R | 0.00198 | | | |
| 4 | | | | | |
| 5 | k01 | 7653.477 | | | |
| 6 | E1 | 4.498061 | | SSR | 3623.813 |
| 7 | | | | | |
| 8 | -dA/dt data | A | T | -dA/dt calculated | SR |
| 9 | 460 | 200 | 280 | 458.448491 | 2.40718 |
| 10 | 385 | 100 | 300 | 393.701529 | 75.71661 |
| 11 | 960 | 150 | 320 | 947.9852747 | 144.3536 |
| 12 | 940 | 80 | 350 | 929.1165318 | 118.4499 |
| 13 | 1530 | 60 | 400 | 1568.524387 | 1484.128 |
| 14 | 2485 | 50 | 450 | 2456.787933 | 795.9207 |
| 15 | 1600 | 20 | 500 | 1628.072627 | 788.0724 |
| 16 | 1245 | 10 | 550 | 1230.345152 | 214.7646 |



20.13 The Excel Solver can be used to develop a nonlinear regression to fit the parameters. The result (along with a plot of the model versus the data estimates) are shown below. Note that the 1:1 line is also displayed on the plot.

| | A | B | C | D | E | F |
|---|---|---|---|---|---|---|
| 1 | Prob. 20.13 | | | | | |
| 2 | | | | | | |
| 3 | R | 82.05 | | | | |
| 4 | | | | | | |
| 5 | A1 | -226.742 | | | | |
| 6 | A2 | 0.987181 | | | SSR | 2.12245E-07 |
| 7 | | | | | | |
| 8 | P (atm) | T(K) | Vol(mL) | PV/(RT) | 1 + A1/V +A2/V^2 | SR |
| 9 | 0.969 | 298 | 25000 | 0.990761 | 0.990930303 | 2.86376E-08 |
| 10 | 1.09 | 298 | 22200 | 0.989657 | 0.989786377 | 1.67844E-08 |
| 11 | 1.341 | 298 | 18000 | 0.987203 | 0.987403199 | 4.01096E-08 |
| 12 | 1.606 | 298 | 15000 | 0.98524 | 0.984883839 | 1.26713E-07 |

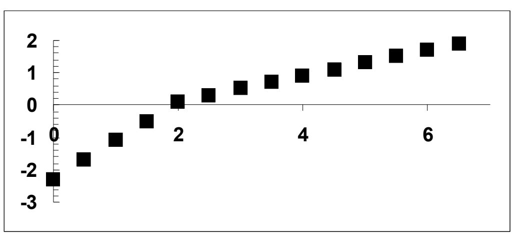


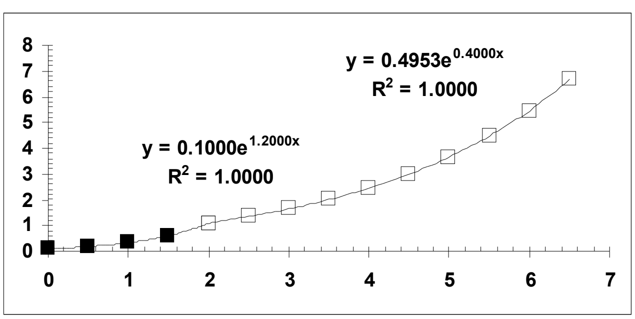

20.14 The standard errors can be computed via Eq. 17.9

$$s_{y/x} = \sqrt{\frac{S_r}{n-2}}$$

Thus, Model C seems best because its standard error is lower.

20.15 A plot of the natural log of cells versus time indicates two straight lines with a sharp break at 2. Trendline can be used to fit each range separately with the exponential model as shown in the second plot.

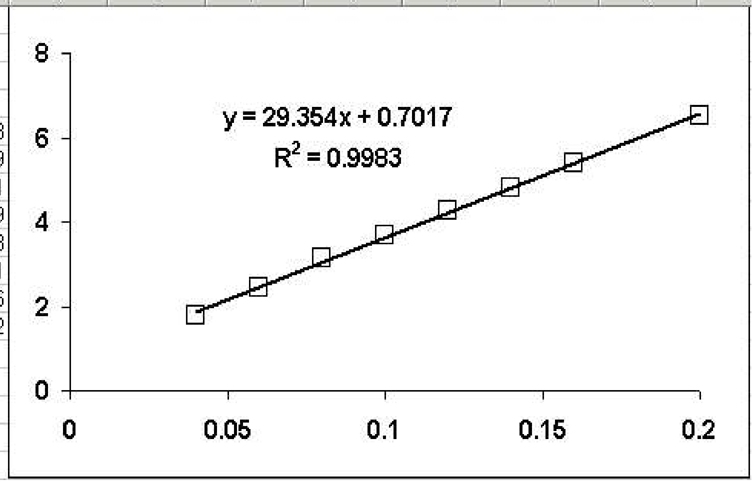



20.16 (This problem was designed by Theresa Good of Texas A&M.) The problem can be solved with Microsoft Excel:

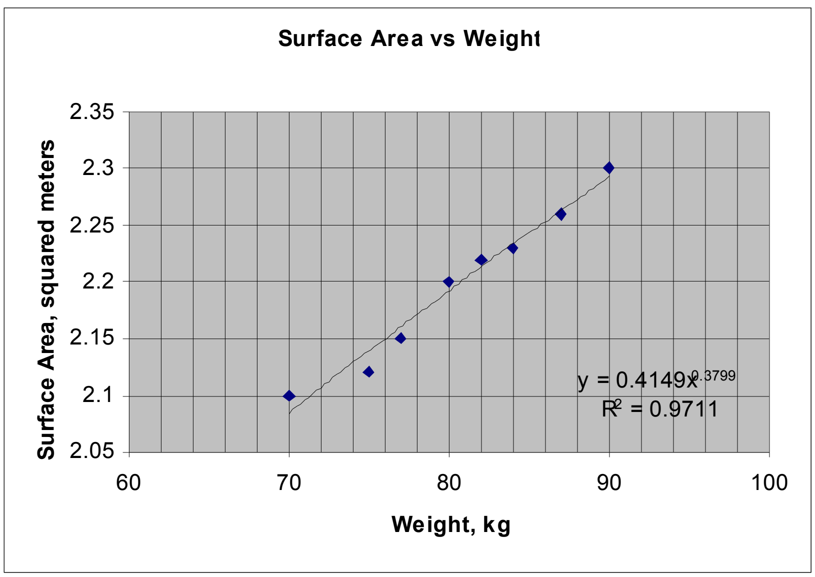
B17 = =(INTERCEPT(C5:C12,A5:A12)/K)^(1/a)

| | A | B | C |
|---|---|---|---|
| 1 | | | |
| 2 | | | |
| 3 | c | t | (t/t0-1)/c |
| 4 | 0 | 83 | |
| 5 | 0.04 | 88.9 | 1.777108 |
| 6 | 0.06 | 95.1 | 2.429719 |
| 7 | 0.08 | 104 | 3.162651 |
| 8 | 0.1 | 113.5 | 3.674699 |
| 9 | 0.12 | 125.5 | 4.267068 |
| 10 | 0.14 | 138.9 | 4.810671 |
| 11 | 0.16 | 154.6 | 5.391566 |
| 12 | 0.2 | 191.2 | 6.518072 |
| 13 | | | |
| 14 | K | 3.90E-04 | |
| 15 | a | 0.58 | |
| 16 | | | |
| 17 | M | 409417.4 | |



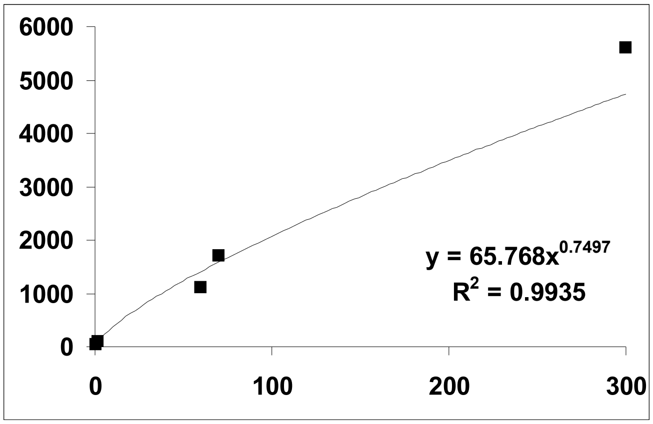

20.17

Plot the data using Excel:




$A = 0.4149W^{0.3799}$ with R-squared = 0.9711. Therefore, a = 0.4149 and b = 0.3799. The predicted surface area for a 95 kg human is approximately: $A = 0.4149\,(95)^{0.3799}$ = **2.34 m²**

20.18 The Excel Trend Line tool can be used to fit a power law to the data:




The logarithmic slope relating the mass and metabolism is 0.75.

20.19

The solution consists of three separate Matlab programs.

**1.** Polynomial Regression

```
g=[105 126 215 315 402];
st=[3.44 4.12 7.02 10.21 13.01];
gf=0:1:450;

%Mean and St
stmean=mean(st);
St=sum( (st-stmean).*(st-stmean) );


% Linear Fit
c1=polyfit(g,st,1);
st1=polyval(c1,g);
Sr1=sum((st-st1).*(st-st1));
r=sqrt((St-Sr1)/St);
stf1=polyval(c1,gf);
plot(g,st,'+',gf,stf1); grid; axis([0 450 0 14]);
title('Shear Rate vs. Shear Stress for 40% Hct Blood')
xlabel('Shear Rate - 1/sec');ylabel('Shear Stress - N/m^2')

fprintf('Correlation Coefficient = %f\n', r)
fprintf('Newtonian slope or Viscosity = %f\n', c1(1))
K = sqrt(c1(1));
fprintf('Consistency index K = %f', K)
```

2. %Polynomial Regression

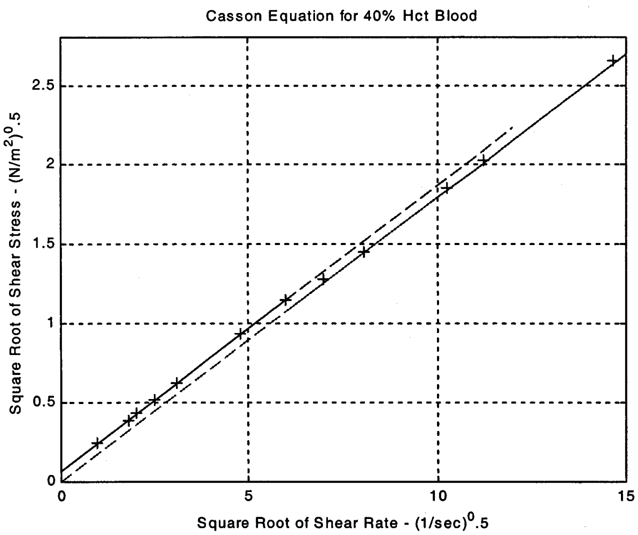
```
g=[105 126 215 315 402];
st=[3.44 4.12 7.02 10.21 13.01];
gf=0:1:450;

%Mean and St
stmean=mean(st);
St=sum( (st-stmean).*(st-stmean) );


% Linear Fit
c1=polyfit(g,st,1);
st1=polyval(c1,g);
Sr1=sum((st-st1).*(st-st1));
r=sqrt((St-Sr1)/St);
stf1=polyval(c1,gf);
plot(g,st,'+',gf,stf1); grid; axis([0 450 0 14]);
title('Shear Rate vs. Shear Stress for 40% Hct Blood')
xlabel('Shear Rate - 1/sec');ylabel('Shear Stress - N/m^2')
fprintf('Correlation Coefficient = %f\n', r)

 fprintf('Newtonian slope or Viscosity = %f\n', c1(1))
 K = sqrt(c1(1));
 fprintf('Consistency index K = %f', K)
```

3. % Casson Equation curve fit.

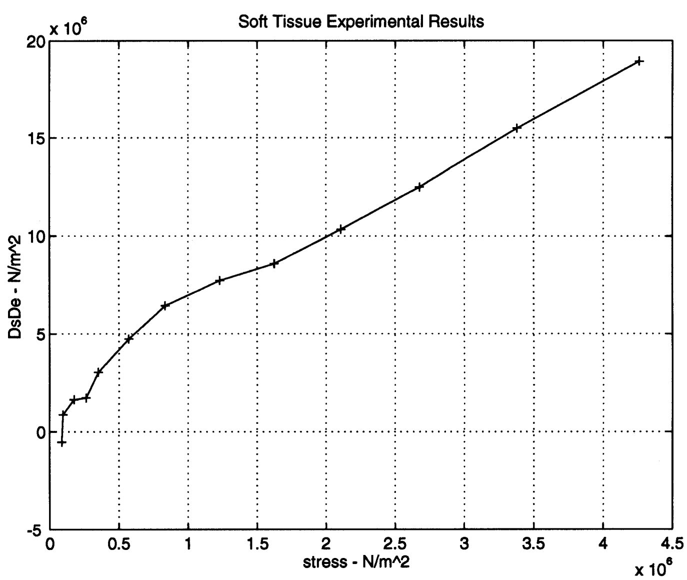
```
g=[.91 3.3 4.1 6.3 9.6 23 36 49 65 105 126 215 315 402];
st=[.059 .15 .19 .27 .39 .87 1.33 1.65 2.11 3.44 4.12 7.02 10.21
13.01];
gsq=sqrt(g);
stsq=sqrt(st);


% Casson Equation segment overall curve
gasqc=0:1:6; gasqcd=6:1:12;
tysq = 0.065818; Kc = 0.180922;
stsqc=tysq+Kc.*gasqc;
stsqcd=tysq+Kc.*gasqcd

% Newtonian segement
gasqn=6:1:15;gasqnd=0:1:10;
K=0.179474
stsqn=K*gasqn;
stsqnd=K*gasqnd;

plot(gasqc,stsqc,gasqcd,stsqcd,'--',gasqn,stsqn,gasqnd,stsqnd,'--
',gsq,stsq,'+');
axis([0 15 0 2.8]);
title('Casson Equation for 40% Hct Blood');
xlabel('Square Root of Shear Rate - (1/sec)^0.5');
ylabel('Square Root of Shear Stress - (N/m^2)^0.5');
grid
```

**Final Results:**

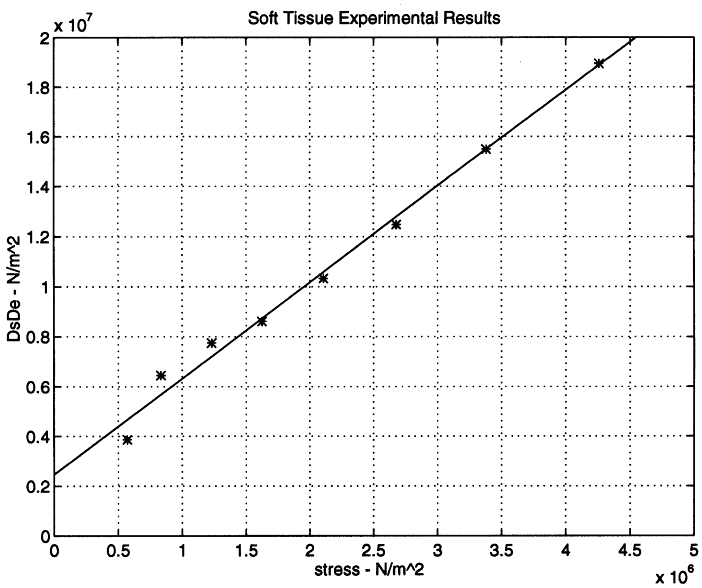
```
Correlation Coefficient = 0.999925
Casson slope Kc = 0.180922
SqRt Yield Stress, y-intrcept = 0.065818
Yield Stress = 0.004332

Correlation Coefficient = 0.999993
Newtonian slope or Viscosity = 0.032211
Consistency index K = 0.179474
```



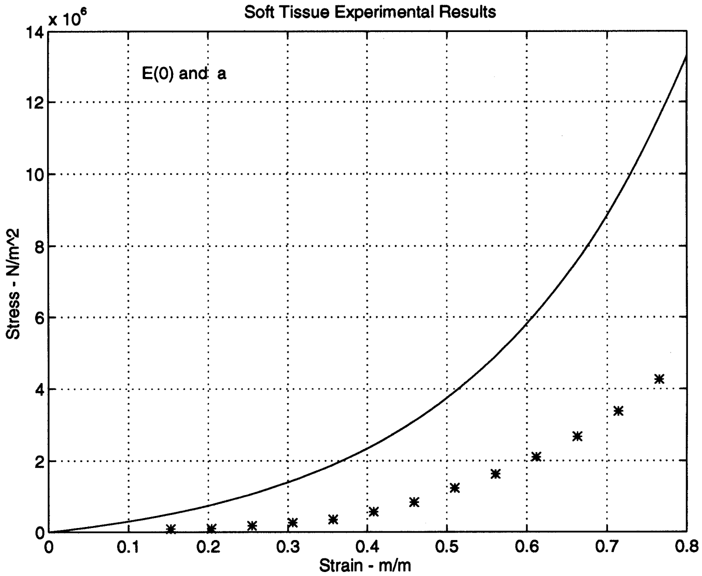

20.20

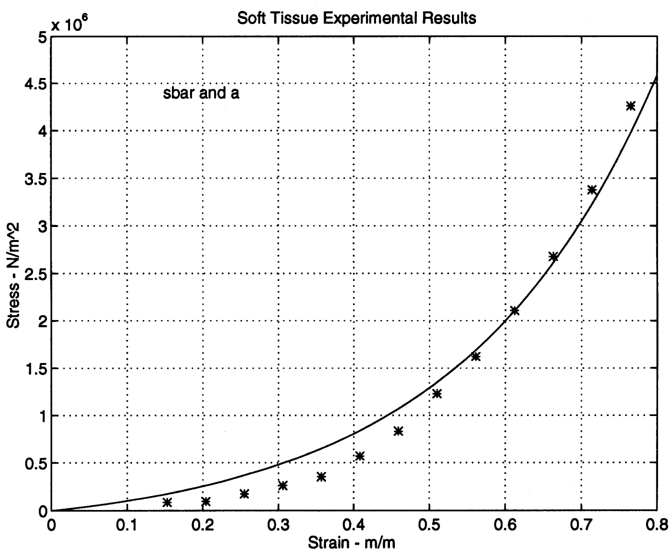


```
    % Raw data input
s=[87.8 96.6 176 263 351 571 834 1229 1624 2107 2678 3380 4258]*1e+3;
e=[153 204 255 306 357 408 459 510 561 612 663 714 765]*1e-3;
de=51e-3; dde=2*de;
    % Finite Differences
dsde(1)=  (-s(3)+4*s(2)-3*s(1))/dde;    %  forward difference
for i=2:12
     dsde(i)=(s(i+1)-s(i-1))/dde;       %   centered difference
end
dsde(13)=(3*s(13)-4*s(12)+s(11))/dde;  %   backward difference

plot(s,dsde,'-',s,dsde,'+')
title('Soft Tissue Experimental Results')
xlabel('stress - N/m^2 '); ylabel('DsDe - N/m^2'); grid
```



```
    %Raw data input
s=[87.8 96.6 176 263 351 571 834 1229 1624 2107 2678 3380 4258]*1e+3;
e=[153 204 255 306 357 408 459 510 561 612 663 714 765]*1e-3;

   %Regression analysis
   %Elimination of early data
idx=5;  % idx=starting point for data exclusion (points with subscribt above idx will be included in s)
       % With this data the range idx can be    idx=3 to idx=8
np=length(s)-idx;
for i=1:np
    sr(i)=s(idx+i);      %sr = regression values for s
end
   %Constants
de=51e-3; dde=2*de;
   % Finite difference
dsder(1)=  (-sr(3)+4*sr(2)-3*sr(1))/dde;          % forward difference
for i=2:np-1
   dsder(i)=(sr(i+1)-sr(i-1))/dde;                % centered difference
end
dsder(np)=(3*sr(np)-4*sr(np-1)+sr(np-2))/dde;    % backward difference
   %Linear Fit
c1=polyfit(sr,dsder,1);
a=c1(1);  Eo=c1(2);
sp=0:1e6:5e6;
dsde1=polyval(c1,sp);
plot(sp,dsde1,sr,dsder,'*')
  title('Soft Tissue Experimental Results')
  xlabel('stress - N/m^2 '); ylabel('DsDe - N/m^2');
  axis([0 5e6 0 20e6]); grid; pause

       % Stress-Strain Curve Plot
       % Plot the analytic expression for s vs e
          % Using Eo and a
  ep=0:.005:0.8;              %  ep=curve plot value of e
  sp=(Eo/a)*(exp(a*ep)-1);    %  sp=curve plot value of s
  plot(ep,sp,e,s,'*')
    title(' Soft Tissue Experimental Results');
    xlabel('Strain - m/m'); ylabel('Stress - N/m^2')
    grid; gtext('E(0) and  a'); pause
        % Using sStar and eStar
  sStar=s(10); eStar=e(10);
  sbar=sStar/(exp(a*eStar)-1);
  sp2=sbar*(exp(a*ep)-1);
  plot(ep,sp2,e,s,'*')
    title(' Soft Tissue Experimental Results');
    xlabel('Strain - m/m'); ylabel('Stress - N/m^2')
    grid; gtext('sbar and a');
```



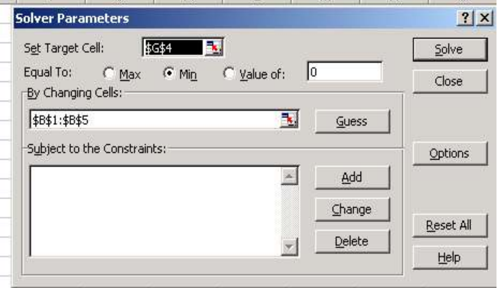

20.21 The problem is set up as the following Excel Solver application. Notice that we have assumed that the model consists of a constant plus two bell-shaped curves:

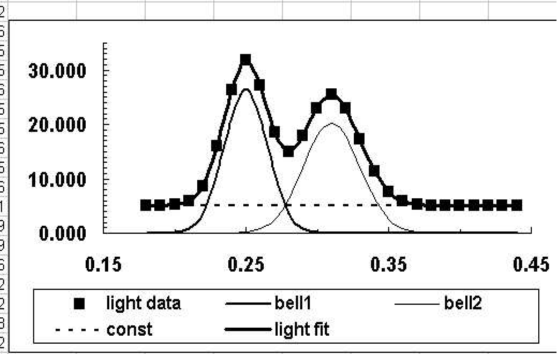
$$f(x) = c + \frac{k_1 e^{-k_1^2 (x-a_1)}}{\sqrt{\pi}} + \frac{k_2 e^{-k_2^2 (x-a_2)}}{\sqrt{\pi}}$$

The resulting solution is

G4 = =SUM(G7:G33)

| | A | B | C | D | E | F | G |
|---|---|---|---|---|---|---|---|
| 1 | **kkk1** | **47.0006** | | | | | |
| 2 | **aaa1** | **0.2500** | | | | | |
| 3 | **kkk2** | **36.0095** | | | | | |
| 4 | **aaa2** | **0.3100** | | | | **SSR** | **0.01301** |
| 5 | **const** | **5.0948** | | | | | |
| 6 | position | light data | bell1 | bell2 | const | light fit | SR |
| 7 | 0.18 | 5.055 | 0.000526 | 6.14E-09 | 5.094819 | 5.095345 | 0.001666 |
| 8 | 0.19 | 5.066 | 0.009298 | 1.57E-07 | 5.094819 | 5.104117 | 0.001428 |
| 9 | 0.2 | 5.166 | 0.105668 | 3.1E-06 | 5.094819 | 5.20049 | 0.00119 |
| 10 | 0.21 | 5.837 | 0.771991 | 4.72E-05 | 5.094819 | 5.866857 | 0.000891 |
| 11 | 0.22 | 8.699 | 3.625812 | 0.000555 | 5.094819 | 8.721186 | 0.000492 |
| 12 | 0.23 | 16.036 | 10.94768 | 0.005035 | 5.094819 | 16.04754 | 0.000133 |
| 13 | 0.24 | 26.374 | 21.25021 | 0.035232 | 5.094819 | 26.38026 | 3.92E-05 |
| 14 | 0.25 | 31.790 | 26.51723 | 0.190211 | 5.094819 | 31.80226 | 0.000153 |
| 15 | 0.26 | 27.140 | 21.27243 | 0.792331 | 5.094819 | 27.15958 | 0.000375 |
| 16 | 0.27 | 18.597 | 10.97059 | 2.546538 | 5.094819 | 18.61195 | 0.000212 |
| 17 | 0.28 | 15.045 | 3.637198 | 6.314885 | 5.094819 | 15.0469 | 4.85E-06 |
| 18 | 0.29 | 17.958 | 0.775225 | 12.08238 | 5.094819 | 17.95242 | 3.11E-05 |
| 19 | 0.3 | 23.042 | 0.106222 | 17.83656 | 5.094819 | 23.0376 | 1.93E-05 |
| 20 | 0.31 | 25.418 | 0.009357 | 20.31615 | 5.094819 | 25.42033 | 5.43E-06 |
| 21 | 0.32 | 22.943 | 0.00053 | 17.85433 | 5.094819 | 22.94968 | 4.46E-05 |
| 22 | 0.33 | 17.197 | 1.93E-05 | 12.10646 | 5.094819 | 17.2013 | 2.21E-05 |
| 23 | 0.34 | 11.430 | 4.51E-07 | 6.333776 | 5.094819 | 11.4286 | 1.97E-06 |
| 24 | 0.35 | 7.659 | 6.79E-09 | 2.5567 | 5.094819 | 7.651519 | 5.6E-05 |
| 25 | 0.36 | 5.904 | 6.57E-11 | 0.796285 | 5.094819 | 5.891104 | 0.000166 |
| 26 | 0.37 | 5.302 | 4.08E-13 | 0.19135 | 5.094819 | 5.286169 | 0.000251 |
| 27 | 0.38 | 5.150 | 1.63E-15 | 0.035478 | 5.094819 | 5.130297 | 0.000369 |
| 28 | 0.39 | 5.122 | 4.2E-18 | 0.005075 | 5.094819 | 5.099894 | 0.000489 |
| 29 | 0.4 | 5.121 | 6.93E-21 | 0.00056 | 5.094819 | 5.095379 | 0.000656 |
| 30 | 0.41 | 5.123 | 7.36E-24 | 4.77E-05 | 5.094819 | 5.094866 | 0.000792 |
| 31 | 0.42 | 5.126 | 5.03E-27 | 3.13E-06 | 5.094819 | 5.094822 | 0.000972 |
| 32 | 0.43 | 5.129 | 2.21E-30 | 1.59E-07 | 5.094819 | 5.094819 | 0.001168 |
| 33 | 0.44 | 5.132 | 6.23E-34 | 6.22E-09 | 5.094819 | 5.094819 | 0.001382 |




Thus, the retina thickness is estimated as 0.31 – 0.25 = 0.06.

20.22 use regression

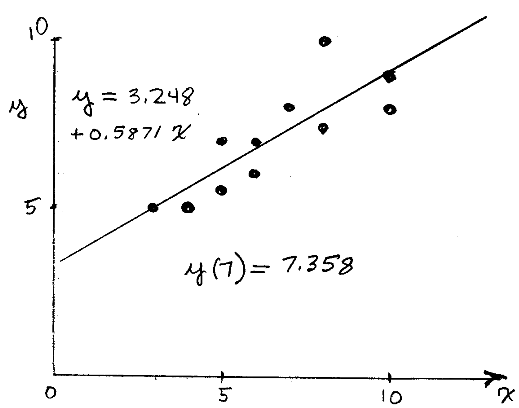
$y = -0.03302 + 0.159875x$

$S_{x/y} = 0.173$

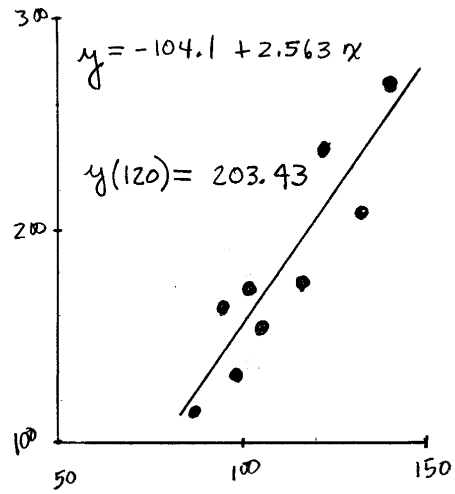
$r^2 = 0.88$

$y(4.5) = 0.6864$

20.23



20.24




20.25

$$y = -0.629 + 0.288\,x$$

$$s_{xy} = 0.46$$

$$r^2 = 0.98$$

$$y(15) = 3.69$$

20.26 $n = 16/10 = 1.6$
$m = 4.8/10 = 0.48$

a) use $x_0 = 0.3$
$x_1 = 0.4$
$x_2 = 0.5$
$x_3 = 0.6$

Lagrange interpolation gives

$$f(0.48) = 0.128454$$

b) use all data and lspline function for mathcad spline interpolation

$$\text{interp}(v_s, x, y, 0.48) = 0.128$$

$$q = \frac{200}{4.8(16)} = 2.604$$

$$\sigma_z = 2.604\,(0.128454) = 0.3345\ \frac{t}{m^2}$$

20.27 use matrix approach

$$\{z^T\}\{z\} = \begin{bmatrix} 0.2754 & 0.6005 & 0.7215 \\ 0.6005 & 1.9517 & 3.123 \\ 0.7215 & 3.123 & 6.594 \end{bmatrix}$$

for multiple linear regression

$$\{z^T\}\{y\} = \begin{Bmatrix} 4.667 \\ 15.149 \\ 25.545 \end{Bmatrix}$$

$$\begin{Bmatrix} A \\ B \\ C \end{Bmatrix} = \left[\{z^T\}\{z\}\right]^{-1} \cdot \{z^T\}\{y\}$$

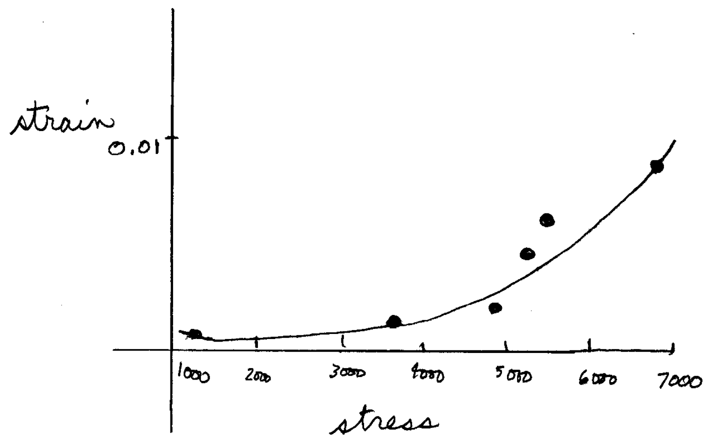
solving gives

$A = 3.68$
$B = 4.44$
$C = 1.37$

which are the initial populations at $t = 0$

20.28

$$\text{stress} = \frac{25069}{10.65} = 2353.8$$




linear regression (not shown) gives

$$\text{strain} = -0.00253 + 1.3865 \times 10^{-6}(\text{stress})$$

and $\text{strain}(2353.8) = 7.37 \times 10^{-4}$

but $r^2 = 0.76$

with $2^{nd}$ order regression

$$\text{strain} = 1.507 \times 10^{-3} - 1.272 \times 10^{-6}(\text{stress}) + 3.379 \times 10^{-10}(\text{stress})^2$$

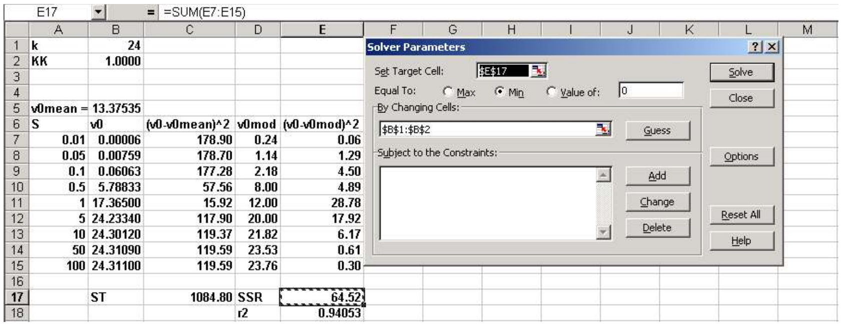
with $r^2 = 0.91$

and $\text{strain}(2353.8) = 3.848 \times 10^{-4}$

Note the large difference between the $1^{st}$ and $2^{nd}$ order results

$$\Delta L = 3.848 \times 10^{-4}(9.14) = 0.00352 \text{ m}$$

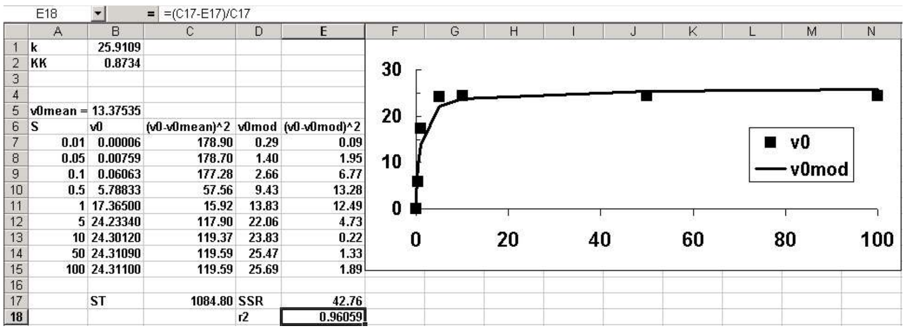
20.29 Clearly the linear model is not adequate. The second model can be fit with the Excel Solver:


E17 = =SUM(E7:E15)

| | A | B | C | D | E |
|---|---|---|---|---|---|
| 1 | k | 24 | | | |
| 2 | KK | 1.0000 | | | |
| 3 | | | | | |
| 4 | | | | | |
| 5 | v0mean = | 13.37535 | | | |
| 6 | S | v0 | (v0-v0mean)^2 | v0mod | (v0-v0mod)^2 |
| 7 | 0.01 | 0.00006 | 178.90 | 0.24 | 0.06 |
| 8 | 0.05 | 0.00759 | 178.70 | 1.14 | 1.29 |
| 9 | 0.1 | 0.06063 | 177.28 | 2.18 | 4.50 |
| 10 | 0.5 | 5.78833 | 57.56 | 8.00 | 4.89 |
| 11 | 1 | 17.36500 | 15.92 | 12.00 | 28.78 |
| 12 | 5 | 24.23340 | 117.90 | 20.00 | 17.92 |
| 13 | 10 | 24.30120 | 119.37 | 21.82 | 6.17 |
| 14 | 50 | 24.31090 | 119.59 | 23.53 | 0.61 |
| 15 | 100 | 24.31100 | 119.59 | 23.76 | 0.30 |
| 16 | | | | | |
| 17 | | ST | 1084.80 | SSR | 64.52 |
| 18 | | | | r2 | 0.94053 |



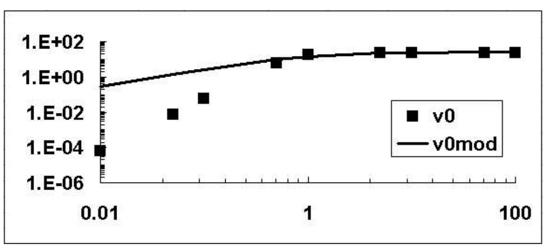
Notice that we have reexpressed the initial rates by multiplying them by $1\times10^5$. We did this so that the sum of the squares of the residuals would not be miniscule. Sometimes this will lead the Solver to conclude that it is at the minimum, even though the fit is poor. The solution is:


E18 = =(C17-E17)/C17

| | A | B | C | D | E |
|---|---|---|---|---|---|
| 1 | k | 25.9109 | | | |
| 2 | KK | 0.8734 | | | |
| 3 | | | | | |
| 4 | | | | | |
| 5 | v0mean = | 13.37535 | | | |
| 6 | S | v0 | (v0-v0mean)^2 | v0mod | (v0-v0mod)^2 |
| 7 | 0.01 | 0.00006 | 178.90 | 0.29 | 0.09 |
| 8 | 0.05 | 0.00759 | 178.70 | 1.40 | 1.95 |
| 9 | 0.1 | 0.06063 | 177.28 | 2.66 | 6.77 |
| 10 | 0.5 | 5.78833 | 57.56 | 9.43 | 13.28 |
| 11 | 1 | 17.36500 | 15.92 | 13.83 | 12.49 |
| 12 | 5 | 24.23340 | 117.90 | 22.06 | 4.73 |
| 13 | 10 | 24.30120 | 119.37 | 23.83 | 0.22 |
| 14 | 50 | 24.31090 | 119.59 | 25.47 | 1.33 |
| 15 | 100 | 24.31100 | 119.59 | 25.69 | 1.89 |
| 16 | | | | | |
| 17 | | ST | 1084.80 | SSR | 42.76 |
| 18 | | | | r2 | 0.96059 |



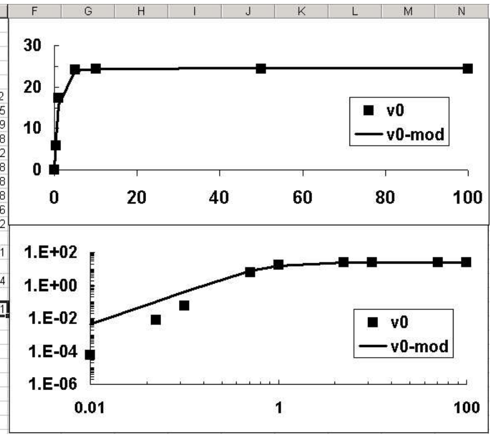
Although the fit might appear to be OK, it is biased in that it underestimates the low values and overestimates the high ones. The poorness of the fit is really obvious if we display the results as a log-log plot:




Notice that this view illustrates that the model actually overpredicts the very lowest values.

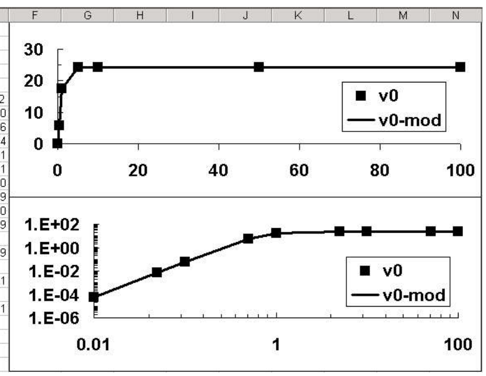
The third and fourth models provide a means to rectify this problem. Because they raise [*S*] to powers, they have more degrees of freedom to follow the underlying pattern of the data. For example, the third model gives:

| | A | B | C | D | E |
|---|---|---|---|---|---|
| 1 | k | 24.58563146 | | | |
| 2 | KK | 0.5638 | | | |
| 3 | | | | | |
| 4 | | | | | |
| 5 | mean | 13.37535 | | | |
| 6 | S | v0 | (v0 - v0mean)^2 | v0-mod | (v0 - v0mod)^2 |
| 7 | 0.01 | 0.00006 | 178.8982473 | 0.00436 | 1.848E-05 |
| 8 | 0.05 | 0.00759 | 178.6967647 | 0.108528 | 0.010187589 |
| 9 | 0.1 | 0.06063 | 177.2817639 | 0.42844 | 0.135287038 |
| 10 | 0.5 | 5.78833 | 57.56280753 | 7.552345 | 3.111747332 |
| 11 | 1 | 17.36500 | 15.91734128 | 15.72131 | 2.701712948 |
| 12 | 5 | 24.23340 | 117.8973428 | 24.04337 | 0.036112918 |
| 13 | 10 | 24.30120 | 119.3742918 | 24.44778 | 0.021487108 |
| 14 | 50 | 24.31090 | 119.5863474 | 24.58009 | 0.072462046 |
| 15 | 100 | 24.31100 | 119.5885345 | 24.58425 | 0.074662992 |
| 16 | | | | | |
| 17 | | ST | 1084.803441 | SSR | 6.163678451 |
| 18 | | | | | |
| 19 | | | | log(SSR) | 0.789839974 |
| 20 | | | | | |
| 21 | | | | r2 | 0.994318161 |
| 22 | | | | | |
| 23 | | | | | |
| 24 | | | | | |
| 25 | | | | | |
| 26 | | | | | |
| 27 | | | | | |
| 28 | | | | | |
| 29 | | | | | |




Finally, the cubic model results in a perfect fit:

| | A | B | C | D | E |
|---|---|---|---|---|---|
| 1 | k | 24.31096419 | | | |
| 2 | KK | 0.4000 | | | |
| 3 | | | | | |
| 4 | | | | | |
| 5 | mean | 13.37535 | | | |
| 6 | S | v0 | (v0 - v0mean)^2 | v0-mod | (v0 - v0mod)^2 |
| 7 | 0.01 | 0.00006 | 178.8982473 | 6.07775E-05 | 2.61873E-20 |
| 8 | 0.05 | 0.00759 | 178.6967647 | 0.007594828 | 8.02552E-16 |
| 9 | 0.1 | 0.06063 | 177.2817639 | 0.060626048 | 2.1946E-14 |
| 10 | 0.5 | 5.78833 | 57.56280753 | 5.788339559 | 9.13676E-11 |
| 11 | 1 | 17.36500 | 15.91734128 | 17.36499102 | 8.06953E-11 |
| 12 | 5 | 24.23340 | 117.8973428 | 24.23341751 | 3.06725E-10 |
| 13 | 10 | 24.30120 | 119.3742918 | 24.30124373 | 1.9119E-09 |
| 14 | 50 | 24.31090 | 119.5863474 | 24.3108864 | 1.85078E-10 |
| 15 | 100 | 24.31100 | 119.5885345 | 24.31095447 | 2.07336E-09 |
| 16 | | | | | |
| 17 | | ST | 1084.803441 | SSR | 4.64914E-09 |
| 18 | | | | | |
| 19 | | | | log(SSR) | -8.332627101 |
| 20 | | | | | |
| 21 | | | | r2 | 1 |
| 22 | | | | | |
| 23 | | | | | |
| 24 | | | | | |
| 25 | | | | | |




Thus, the best fit is

$$v_0 = \frac{2.4311\times10^{-5}[S]}{0.4+[S]}$$

20.30 FFT gives all zero for both real and imaginary parts except

| frequency | real | imaginary |
|---|---|---|
| 0 | 6 | 0 |
| $4/2\pi$ | 0 | 1 |
| $7/2\pi$ | 2.5 | 0 |
| $-7/2\pi$ | 2.5 | 0 |
| $-4/2\pi$ | 0 | $-1$ |

20.31 use $x_0 = 0.75$

$x_1 = 1.25$

$x_2 = 1.5$

$x_3 = 0.25$

$x_4 = 2$

| order | f(1.1) |
|---|---|
| 1 | 0.31 |
| 2 | 0.1616 |
| 3 | 0.1604 |
| 4 | 0.1762 |

using Lagrange interpolation

clearly 1st order estimate is inadequate.

other values are acceptable

20.32 Use Polynomial Regression from TOOLKIT

| order | f(1.1) | $S_{xy}$ |
|---|---|---|
| 1 | 1.329 | 1.48 |
| 2 | 0.0856 | 0.13 |
| 3 | 0.115 | 0.062 |
| 4 | 0.1762 | — |

note that results are sensitive to order because there are few data points

Interpolation is perhaps a better way to estimate f(1.1) is the measurements are reasonably well known

20.33 $x_0 = 0.25$
$x_1 = 0.125$
$x_2 = 0.375$
$x_3 = 0.50$
$x_4 = 0$

| order | f(0.22) |
|---|---|
| 1 | 7.4165 |
| 2 | 7.825 |
| 3 | 7.921 |
| 4 | 7.871 |

20.34 Linear regression

gives $y = -0.592 + 2.8082\,x$

$r^2 = 0.999 \quad y(6) = 16.25$

If the theoretical aspects of the problem demand that $i = 0$ when $V = 0$ we may require that intercept $= 0$.

model becomes $y = a_1 x$

Use matrix method

$$z = \begin{bmatrix} 2 \\ 3 \\ 4 \\ 5 \\ 7 \\ 10 \end{bmatrix} \qquad y = \begin{bmatrix} 5.2 \\ 7.8 \\ 10.7 \\ 13 \\ 19.3 \\ 27.5 \end{bmatrix}$$

$$\{Z^T\}\{Z\} = 203$$

$$a_1 = [Z^T \cdot Z]^{-1} \cdot (Z^T \cdot y)$$

gives $a_1 = 2.718$

which is not a line that minimizes

$$\sum (\text{measured } y - \text{model } y)^2$$

$y(6) = 16.308$

20.35 Linear regression gives

$y = 0.509 + 4.901\,x$

with $r^2 = 0.99$

$L \approx 4.901$

non zero intercept suggests data violates Faraday's Law or have measurement uncertainties

20.36 Use Lagrange interpolation

$$f(0.1) = 4.648$$

20.37 $y = a_0 + a_1 x + a_2 x^2 + a_3 x^3 + a_4 x^4 + a_5 x^5$

$$\begin{bmatrix} 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & -0.5 & 0.25 & -0.125 & 0.0625 & -0.03125 \\ 1 & -0.25 & 0.0625 & -0.015625 & 0.0039063 & -0.0009766 \\ 1 & 0.25 & 0.0625 & 0.015625 & 0.0039063 & 0.0009766 \\ 1 & 0.50 & 0.25 & 0.0125 & 0.0625 & 0.03125 \\ 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \end{bmatrix} = \begin{bmatrix} -193 \\ -41 \\ -13.5625 \\ 13.5625 \\ 41 \\ 193 \end{bmatrix}$$

solving gives

$$a_0 = -2.1 \times 10^{-14}$$
$$a_1 = 45$$
$$a_2 = -6.4 \times 10^{-14}$$
$$a_3 = 148$$
$$a_4 = 8.5 \times 10^{-14}$$
$$a_5 = 2.3 \times 10^{-13}$$

or $v \approx 45 i + 148 i^3$

which was used to generate the data

20.38 Use Regression because measurement are likely to have some error

Linear regression gives

$$y = 4.85571 + 0.029281\,x$$

with $r^2 = 0.98$

$$y(400) = 16.558$$

20.39 a)

| order | $J_0(2.1)$ |
|---|---|
| 1 | 0.1671501 |
| 2 | 0.1668251 |
| 3 | 0.1666376 |
| 4 | 0.1666376 |

b) Use Mathcad lspline and interp functions

$$x = \begin{bmatrix} 1.2 \\ 2.0 \\ 2.2 \\ 2.4 \\ 2.6 \end{bmatrix} \qquad y = \begin{bmatrix} 0.34 \\ 0.2239 \\ 0.1104 \\ 0.0025 \\ 0.0968 \end{bmatrix}$$

$$vs = \text{lspline}(x, y)$$

$$\text{interp}(vs, x, y, 2.1) = 0.17057$$

Note that simple interpolation gives more accurate results

20.40 let $p = a e^{bt}$

$$\ln p = \ln a + bt$$

| $t$ | $\ln p$ |
|---|---|
| 0 | 4.605 |
| 5 | 5.357 |
| 10 | 6.105 |
| 15 | 6.855 |
| 20 | 7.605 |

Regression gives

$$\ln p = 4.6058 + 0.14996\,t$$

$$\therefore \ln a = 4.6058$$
$$a = 100.06$$

$$b = 0.14996$$

$$\therefore p = 100.06\, e^{0.14996\,t}$$

check

$$p(15) = 100.06\, e^{0.14996(15)} = 948.77$$

$$p(5) = 100.06\, e^{0.14996(5)} = 211.78$$

20.41 use

$x_0 = 1 \quad f(x_0) = 4.7$
$x_1 = 2 \quad f(x_1) = 28.9$
$x_2 = 3 \quad f(x_2) = 84$

| order | $f(1.23)$ |
|---|---|
| 1 | 10.266 |
| 2 | 7.53 |

From case study 20.4

$$Q = 55.9\,(1.23)^{2.62}\,(0.01)^{0.54}$$

$= 8$

20.42 The equation to be fit is given by

$$\log D = \log a_0 + a_1 \log S + a_2 \log Q$$

multiple linear regression can be used to obtain

$a_0 = -0.668 \quad a_1 = -0.205$
$a_2 = 0.382$

or

$$\log D = -0.668 - 0.205 \log S + 0.382 \log Q$$

or

$$D = 0.2148\, S^{-0.205}\, Q^{0.382}$$

or solving for Q

$$Q = 55.9\, D^{2.62}\, S^{0.54}$$ which is the same as Case Study 20.4

20.43 a) use $x_0 = 8$
$x_1 = 4$
$x_2 = 12$
$x_3 = 0$

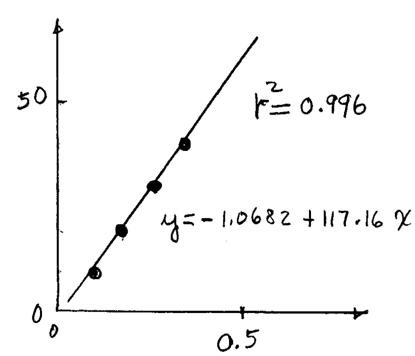
| order | $f(7.5)$ |
|---|---|
| 1 | 1.409925 |
| 2 | 1.408153 |
| 3 | 1.407905 |

b) use all data to perform 2nd order regression, gives

$$y = 1.7841 - 0.0551x + 0.0008062x^2$$

and $y(7.5) = 1.4161$

20.44 For the first four points



For the last four points

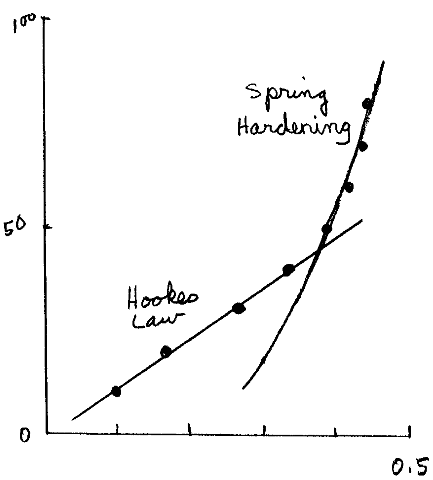
| $x$ | $y$ | $\log x$ | $\log y$ |
|---|---|---|---|
| 0.39 | 50 | −0.409 | 1.699 |
| 0.42 | 60 | −0.377 | 1.778 |
| 0.43 | 70 | −0.366 | 1.845 |
| 0.44 | 80 | −0.357 | 1.903 |

Regression gives

$$\log y = 3.221 + 3.75\,x$$

$$r^2 = 0.938$$

$$\therefore \quad y = 1663.41\,x^{3.75}$$



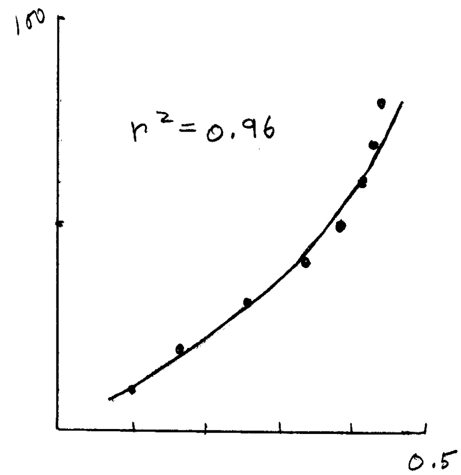

20.45 additional values

| $x$ | $y$ | $\log x$ | $\log y$ |
|---|---|---|---|
| 0.1 | 10 | −1. | 1 |
| 0.17 | 20 | −0.7695 | 1.301 |
| 0.27 | 30 | −0.5686 | 1.477 |
| 0.35 | 40 | −0.456 | 1.602 |

gives $\log y = 2.268 + 1.288 \log x$

or

$$y = 185.35\,x^{1.288}$$




20.46

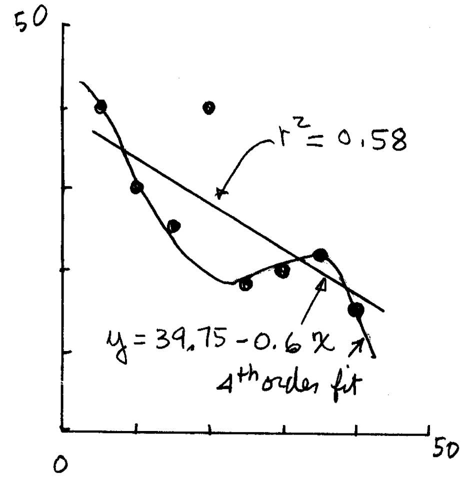
Plot of data suggests polynomial regression would be best approach

| order | $r^2$ | $f(45)$ |
|---|---|---|
| 1 | 0.98 | 79.14 |
| 2 | 0.9974 | 74.87 |
| 3 | 0.9974 | 74.65 |

Note no improvement in $r^2$ for 3rd order compared to 2nd order

2nd order is good compromise giving small error with reasonable calculation effort

20.47




$y(17) = 29.55$

Some students may reject point at $x=20$, $y=40$ and use polynomical regression

| order | $r^2$ | $f(17)$ |
|---|---|---|
| 1 | 0.78 | 27.6 |
| 2 | 0.89 | 24.74 |
| 3 | 0.96 | 22.3 |
| 4 | 0.99 | 21.3 |

4th fits data without $x=20$, $y=40$ almost exactly

with $a_0 = 43.75$
$a_1 = 0.02195$
$a_2 = -0.2183$
$a_3 = 0.01056$
$a_4 = -1.39 \times 10^{-4}$

20.48

Linear Regression gives

$y(55{,}000) = 9.643183$
with $r^2 = 0.99997$

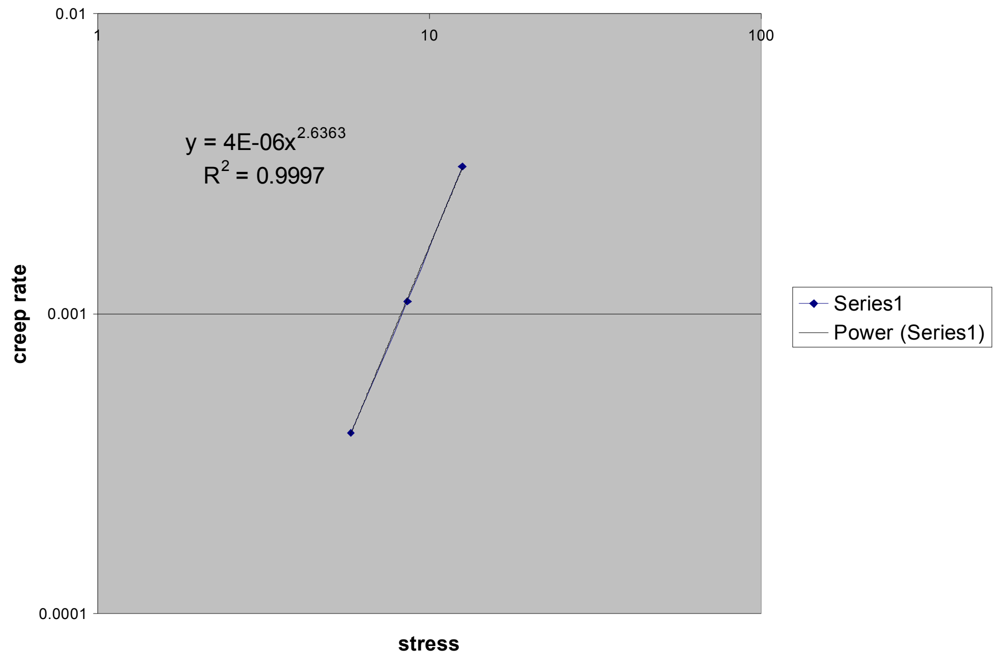
Linear Interpolation gives

$y_1(55{,}000) = 9.642825$

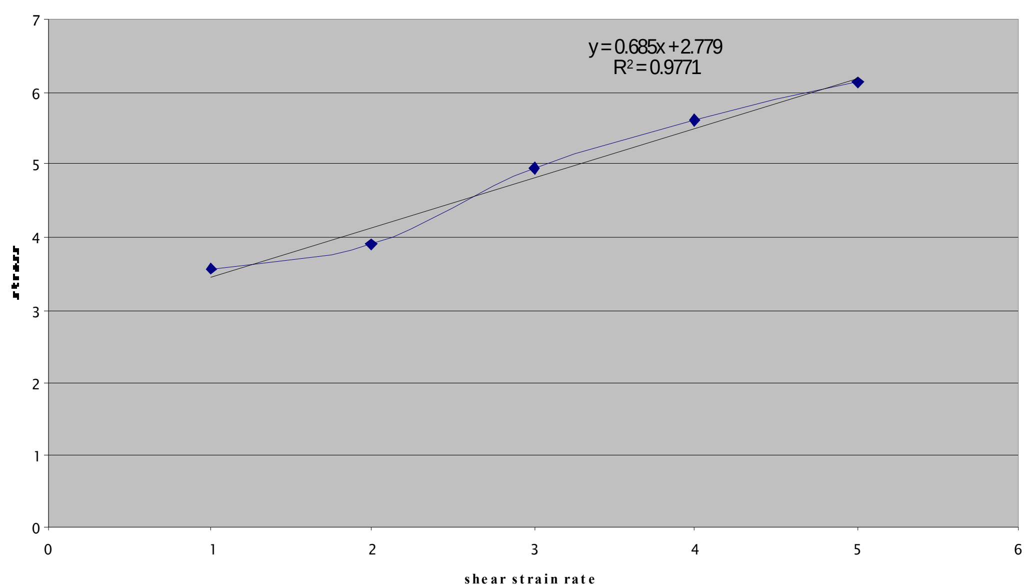
2nd order interpolation

$y_2(55{,}000) = 9.642787$

20.49 This problem was solved using an Excel spreadsheet.




20.50 This problem was solved using an Excel spreadsheet.

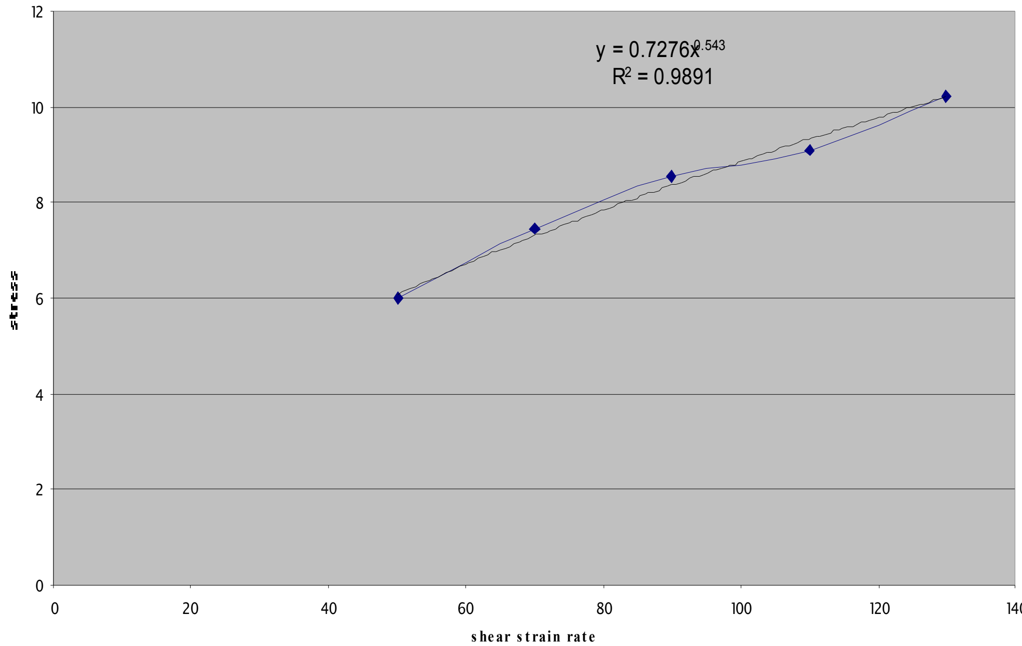



**$\mu$ = 0.685**

**$\tau_y$ = 2.779 N/m$^2$**

$r^2$ = **0.9771**

20.51 This problem was solved using an Excel spreadsheet.




$\mu$ = **0.7276**
***n*** **= 0.543**

| | | | |
|---|---|---|---|
| *n* | 15 | | |
| | Model A | Model B | Model C |
| $S_r$ | 135 | 90 | 72 |
| Number of model | | | |
| parameters fit | 2 | 3 | 4 |
| $s_{y/x}$ | 3.222517 | 2.738613 | 2.558409 |

20.10 use mathcad

vs = lspline (x, y)

The second derivatives at the nodes become

| x | $f''(x)$ |
|---|---|
| 0.5 | -68.49 |
| 1.0 | -110.03 |
| 1.5 | 220.62 |
| 2.0 | -100.43 |
| 2.5 | 61.11 |

inflection point here

use interp (x, y, vs, 1.0) = 50
interp (x, y, vs, 1.16) = 39.44
interp (x, y, vs, 1.33) = 28.15
interp (x, y, vs, 1.5) = 20.0

The cubic equation that exactly fits these points is unique and given by

$$f = a_0 + a_1 x + a_2 x^2 + a_3 x^3$$

where $a_0 = -56.92$
$a_1 = 384.92$
$a_2 = -389.15$
$a_3 = 111.15$

$$f''(x) = 2a_2 + 6a_3 x$$

The inflection point occurs when $f''(x) = 0$ or,

$$x = \frac{-2a_2}{6a_3} = \frac{-2(389.15)}{6(111.15)}$$

$$= 1.167$$

$$f' = a_1 + 2a_2 T + 3a_3 T^2$$

$$f'(1.167) = 384.92 + 2(-389.15)(1.167) + 3(111.15)(1.167)^2$$

$$= -69.23$$

Therefore

$$J = -0.01\left(-69.23 \frac{\text{cal/cm}^3}{\text{m} \times 10^2\,\text{cm/m}}\right)$$

$$= 6.923 \times 10^{-3} \frac{\text{cal}}{\text{cm}^2\text{s}}$$

# CHAPTER 13

13.1 $\frac{df}{dx} = 2x - 8 = 0$

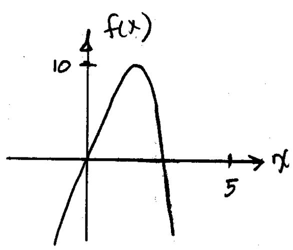
a) $x_{max} = 4$

b) $f(0) = 12$
$f(2) = 0$
$f(6) = 0$

$$x_3 = \frac{12(4-36)}{2(12)(2-6)} = 4$$

13.2

a)




b) $f''(x) = -45x^4 - 24x^2 < 0$
for all $x$, $\therefore$ concave

c) $f' = -9x^5 - 8x^3 + 12 = 0$

root $= 0.916915$

using BISECTION Method

$f(.916915) = 8.69729$

13.3 Follow Example 13.1

for $x_l = 0$ $x_u = 2$

$d = 0.618 \times 2 = 1.236$

$x_1 = 0 + 1.236 = 1.236$
$x_2 = 2 - 1.236 = 0.764$

The follow table can be generated

| $i$ | $x_l$ | $x_2$ | $x_1$ | $x_u$ |
|---|---|---|---|---|
| 1 | 0 | 0.764 | 1.236 | 2 |
| 2 | 0 | 0.472 | 0.764 | 1.236 |
| 3 | 0.472 | 0.764 | 0.944 | 1.236 |
| 4 | 0.764 | 0.994 | 1.056 | 1.236 |
| 5 | 0.764 | 0.875 | 0.944 | 1.056 |
| ⋮ | | | | |
| 11 | 0.9117 | 0.9180 | 0.9218 | 0.9280 |

with $\varepsilon_a = 0.677\ \%$

and

$x_{opt} = 0.9179$

and

$f(x_{opt}) = 8.6979$

13.4 Follow Example 13.2

with $x_0 = 0$, $x_1 = 1$ and $x_2 = 2$

gives

$x_3 = 0.5702$

the follow table is generated

| $i$ | $x_0$ | $x_1$ | $x_2$ | $x_3$ |
|---|---|---|---|---|
| 1 | 0 | 1 | 2 | 0.5702 |
| 2 | 0 | 0.5702 | 1 | 1.101 |
| 3 | 0.5702 | 1.101 | 1 | 0.8738 |
| 4 | 0.5702 | 0.8738 | 1.101 | 0.8802 |
| 5 | 0.8738 | 0.8802 | 1.101 | 0.9096 |
| 6 | 0.8802 | 0.9096 | 1.101 | 0.9126 |
| 7 | 0.9096 | 0.9126 | 1.101 | 0.9158 |
| 8 | 0.9126 | 0.9158 | 1.101 | 0.9164 |

xopt = 0.9164

f(xopt) = 8.6979

13.5 $$x_{i+1} = x_i - \frac{f_i'}{f_i''}$$

$$f = -1.5x^6 - 2x^4 + 12$$
$$f' = -9x^5 - 8x^3 + 12$$
$$f'' = -45x^4 - 24x^2$$

for $x_0 = 2$

$x_1 = 1.9800$

$x_2 = 1.5677$

⋮

$x_7 = 0.916916$

where $\text{abs}\left[\frac{x_{i+1} - x_i}{x_{i+1}}\right] \times 100 < 1$

13.6 Student answers may vary but Golden Section Search is inefficient but always converges if $x_l$ and $x_u$ bracket the max or min of a unimodal function.

Quadratic interpolation and Newtons method may converge rapidly for well-behaved functions and good initial values, otherwise they may diverge.

Newtons method has the disadvantage that it requires evaluation of $f''$.

13.7 a)

| $i$ | $x_l$ | $x_2$ | $x_1$ | $x_u$ |
|---|---|---|---|---|
| 1 | -2.0 | 0.292 | 1.708 | 4.0 |
| 2 | 0.292 | 1.708 | 2.584 | 4.0 |
| 3 | 0.292 | 1.167 | 1.708 | 2.584 |
| 4 | 1.167 | 1.708 | 2.043 | 2.584 |
| 5 | 1.708 | 2.043 | 2.249 | 2.584 |
| ⋮ | | | | |
| 11 | 2.043 | 2.0612 | 2.073 | 2.091 |

xopt = 2.073

f(xopt) = 1.808

with $\epsilon_a = 0.9$ %

13.7 b)

| $i$ | $x_0$ | $x_1$ | $x_2$ | $x_3$ |
|---|---|---|---|---|
| 1 | 1.75 | 2.0 | 2.25 | 2.0617 |
| 2 | 2.0 | 2.0617 | 2.25 | 2.0741 |
| 3 | 2.0617 | 2.0741 | 2.25 | 2.0779 |
| 4 | 2.0741 | 2.0779 | 2.25 | 2.0791 |
| 5 | 2.0791 | 2.0791 | 2.25 | 2.0786 |

xopt = 2.0791

f(xopt) = 1.8082

13.7 c)

$f' = 2 - 3.5x + 3.3x^2 - x^3$

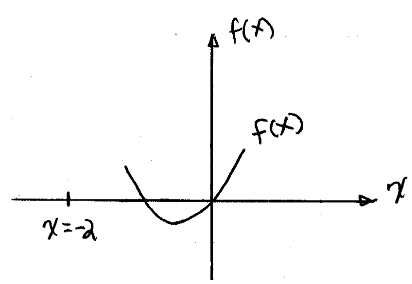
$f'' = -3.5 + 6.6x - 3x^2$

| $i$ | $x_i$ | $\epsilon_a$ |
|---|---|---|
| 0 | 2.5 | — |
| 1 | 2.19565 | 13.9 |
| 2 | 2.0917 | 5.0 |
| 3 | 2.07951 | 0.6 |

13.8 $f' = 6 + 15x + 9x^2 + 4x^3$

$f'' = 15 + 18x + 12x^2$

which is >0 for $-2 \le x \le 1$




13.10 answers vary depending choice for initial points

For example for

$x_0 = 0.1$, $x_1 = 0.5$, $x_2 = 5.0$

13.9 a) function has minimum

| $i$ | $x_l$ | $x_2$ | $x_1$ | $x_u$ |
|---|---|---|---|---|
| 1 | -2.0 | -0.854 | -0.146 | 1. |
| 2 | -2.0 | -1.292 | -0.854 | -0.146 |
| 3 | -1.292 | -0.854 | -0.584 | -0.146 |
| 4 | -0.854 | -0.584 | -0.416 | -0.146 |
| 5 | -0.854 | -0.687 | -0.584 | -0.416 |
| ⋮ | | | | |
| 13 | -0.535 | -0.531 | -0.529 | -0.526 |

xopt = -0.52905

f(xopt) = -1.440989

$\epsilon_a = 0.7\ \%$

b)

| $i$ | $x_0$ | $x_1$ | $x_2$ | $x_3$ |
|---|---|---|---|---|
| 1 | -2 | -1 | 1 | -0.692 |
| 2 | -2 | -0.692 | -1 | -0.621 |
| 3 | -2 | -0.622 | -0.692 | -0.556 |
| 4 | -2 | -0.556 | -0.622 | -0.540 |
| 5 | -2 | -0.540 | -0.556 | -0.531 |

xopt = -0.53969

f(xopt) = -1.4404

c) with $x_0 = -1$

| $i$ | $x_i$ | $\epsilon_a$ |
|---|---|---|
| 0 | -1 | — |
| 1 | -0.555 | 80 |
| 2 | -0.5278 | 5.2 |
| 3 | -0.52803 | 0.04 |

13.12

a) Newton method

| $i$ | $x_i$ |
|---|---|
| 0 | -1 |

13.13 Because of multiple local minima and maxima, there is no really simple means to test whether a single maximum occurs within an interval without actually performing a search. However, if we assume that the function has one maximum and no minima within the interval, a check can be included. Here is a VBA program to implement the Golden section search algorithm for maximization and solve Example 13.1.

```
Option Explicit

Sub GoldMax()
Dim ier As Integer
Dim xlow As Double, xhigh As Double
Dim xopt As Double, fopt As Double
xlow = 0
xhigh = 4
Call GoldMx(xlow, xhigh, xopt, fopt, ier)
If ier = 0 Then
  MsgBox "xopt = " & xopt
  MsgBox "f(xopt) = " & fopt
Else
  MsgBox "Does not appear to be maximum in [xl, xu]"
End If
End Sub

Sub GoldMx(xlow, xhigh, xopt, fopt, ier)
Dim iter As Integer, maxit As Integer, ea As Double, es As Double
Dim xL As Double, xU As Double, d As Double, x1 As Double
Dim x2 As Double, f1 As Double, f2 As Double
Const R As Double = (5 ^ 0.5 - 1) / 2

ier = 0
maxit = 50
es = 0.001
xL = xlow
xU = xhigh
iter = 1
d = R * (xU - xL)
x1 = xL + d
x2 = xU - d
f1 = f(x1)
f2 = f(x2)
If f1 > f2 Then
  xopt = x1
  fopt = f1
Else
  xopt = x2
  fopt = f2
End If
If fopt > f(xL) And fopt > f(xU) Then
  Do
    d = R * d
    If f1 > f2 Then
      xL = x2
      x2 = x1
      x1 = xL + d
      f2 = f1
      f1 = f(x1)
    Else
      xU = x1
      x1 = x2
      x2 = xU - d
      f1 = f2
      f2 = f(x2)
    End If
    iter = iter + 1
    If f1 > f2 Then
```

```
      xopt = x1
      fopt = f1
    Else
      xopt = x2
      fopt = f2
    End If
    If xopt <> 0 Then ea = (1 - R) * Abs((xU - xL) / xopt) * 100
    If ea <= es Or iter >= maxit Then Exit Do
  Loop
Else
  ier = 1
End If

End Sub

Function f(x)
f = -(2 * Sin(x) - x ^ 2 / 10)
End Function
```

13.14 The easiest way to set up a maximization algorithm so that it can do minimization is the realize that minimizing a function is the same as maximizing its negative. Therefore, the following algorithm minimizes or maximizes depending on the value of a user input variable, ind, where ind = -1 and 1 correspond to minimization and maximization, respectively.

```
Option Explicit

Sub GoldMinMax()
Dim ind As Integer       'Minimization (ind = -1); Maximization (ind = 1)
Dim xlow As Double, xhigh As Double
Dim xopt As Double, fopt As Double
xlow = 0
xhigh = 4
Call GoldMnMx(xlow, xhigh, -1, xopt, fopt)
MsgBox "xopt = " & xopt
MsgBox "f(xopt) = " & fopt
End Sub

Sub GoldMnMx(xlow, xhigh, ind, xopt, fopt)
Dim iter As Integer, maxit As Integer, ea As Double, es As Double
Dim xL As Double, xU As Double, d As Double, x1 As Double
Dim x2 As Double, f1 As Double, f2 As Double
Const R As Double = (5 ^ 0.5 - 1) / 2

maxit = 50
es = 0.001
xL = xlow
xU = xhigh
iter = 1
d = R * (xU - xL)
x1 = xL + d
x2 = xU - d
f1 = f(ind, x1)
f2 = f(ind, x2)
If f1 > f2 Then
  xopt = x1
  fopt = f1
Else
  xopt = x2
  fopt = f2
End If

Do
  d = R * d
  If f1 > f2 Then
```

```
      xL = x2
      x2 = x1
      x1 = xL + d
      f2 = f1
      f1 = f(ind, x1)
    Else
      xU = x1
      x1 = x2
      x2 = xU - d
      f1 = f2
      f2 = f(ind, x2)
    End If
    iter = iter + 1
    If f1 > f2 Then
      xopt = x1
      fopt = f1
    Else
      xopt = x2
      fopt = f2
    End If
    If xopt <> 0 Then ea = (1 - R) * Abs((xU - xL) / xopt) * 100
    If ea <= es Or iter >= maxit Then Exit Do
  Loop
  fopt = ind * fopt
  End Sub

  Function f(ind, x)
  f = ind * (1.1333 * x ^ 2 - 6.2667 * x + 1)
  End Function
```

13.15 Because of multiple local minima and maxima, there is no really simple means to test whether a single maximum occurs within an interval without actually performing a search. However, if we assume that the function has one maximum and no minima within the interval, a check can be included. Here is a VBA program to implement the Quadratic Interpolation algorithm for maximization and solve Example 13.2.

```
Option Explicit

Sub QuadMax()
Dim ier As Integer
Dim xlow As Double, xhigh As Double
Dim xopt As Double, fopt As Double
xlow = 0
xhigh = 4
Call QuadMx(xlow, xhigh, xopt, fopt, ier)
If ier = 0 Then
  MsgBox "xopt = " & xopt
  MsgBox "f(xopt) = " & fopt
Else
  MsgBox "Does not appear to be maximum in [xl, xu]"
End If
End Sub

Sub QuadMx(xlow, xhigh, xopt, fopt, ier)
Dim iter As Integer, maxit As Integer, ea As Double, es As Double
Dim x0 As Double, x1 As Double, x2 As Double
Dim f0 As Double, f1 As Double, f2 As Double
Dim xoptOld As Double

ier = 0
maxit = 50
es = 0.01
x0 = xlow
x2 = xhigh
x1 = (x0 + x2) / 2
```

```
f0 = f(x0)
f1 = f(x1)
f2 = f(x2)
If f1 > f0 Or f1 > f2 Then
  xoptOld = x1
  Do
    xopt = f0 * (x1^2 - x2^2) + f1 * (x2^2 - x0^2) + f2 * (x0^2 - x1^2)
    xopt = xopt / (2*f0 * (x1 - x2) + 2*f1 * (x2 - x0) + 2*f2 * (x0 - x1))
    fopt = f(xopt)
    iter = iter + 1
    If xopt > x1 Then
      x0 = x1
      f0 = f1
      x1 = xopt
      f1 = fopt
    Else
      x2 = x1
      f2 = f1
      x1 = xopt
      f1 = fopt
    End If
    If xopt <> 0 Then ea = Abs((xopt - xoptOld) / xopt) * 100
    xoptOld = xopt
    If ea <= es Or iter >= maxit Then Exit Do
  Loop
Else
  ier = 1
End If
End Sub

Function f(x)
f = -(2 * Sin(x) - x ^ 2 / 10)
End Function
```

13.16 Here is a VBA program to implement the Newton-Raphson method for maximization.

```
Option Explicit

Sub NRMax()
Dim xguess As Double
Dim xopt As Double, fopt As Double
xguess = 2.5
Call NRMx(xguess, xopt, fopt)
MsgBox "xopt = " & xopt
MsgBox "f(xopt) = " & fopt
End Sub

Sub NRMx(xguess, xopt, fopt)
Dim iter As Integer, maxit As Integer, ea As Double, es As Double
Dim x0 As Double, x1 As Double, x2 As Double
Dim f0 As Double, f1 As Double, f2 As Double
Dim xoptOld As Double

maxit = 50
es = 0.01
Do
  xopt = xguess - df(xguess) / d2f(xguess)
  fopt = f(xopt)
  If xopt <> 0 Then ea = Abs((xopt - xguess) / xopt) * 100
  xguess = xopt
  If ea <= es Or iter >= maxit Then Exit Do
Loop
End Sub

Function f(x)
f = -(2 * Sin(x) - x ^ 2 / 10)
```

```
End Function

Function df(x)
df = 2 * Cos(x) - x / 5
End Function

Function d2f(x)
d2f = -2 * Sin(x) - 1 / 5
End Function
```

13.17 Here is a VBA program to implement the Newton-Raphson method for maximization.

$$d_1 = \left(\frac{\sqrt{5}-1}{2}\right)(4-2) = 1.23606$$

$$x_1 = 2 + d_1 = 3.23606$$

$$x_2 = 4 - d_1 = 2.76394$$

$$f(x_1) = -4.69808$$

$$f(x_2) = -5.55333$$

$$f(x_2) < f(x_1) \Rightarrow x_1 \textit{ is new } x_U$$

$$d_2 = \left(\frac{\sqrt{5}-1}{2}\right)(3.23606-2) = 0.763927$$

$$x_1 = 2 + d_2 = 2.7639$$

$$x_2 = 3.23606 - d_2 = 2.472133$$

$$f(x_1) = -5.55331$$

$$f(x_2) = -4.82656$$

$$f(x_1) < f(x_2) \Rightarrow x_2 \text{ is new } x_L$$

$$d_3 = \left(\frac{\sqrt{5}-1}{2}\right)(3.23606 - 2.472133) = 0.4721$$

$$x_1 = 2.472133 + d_3 = 2.9442$$

$$x_2 = 3.23606 - d_3 = 2.7639$$

$$f(x_1) = -4.9353$$

$$f(x_2) = -5.55331$$

$$f(x_2) < f(x_1) \Rightarrow x_1 \text{ is new } x_U$$

$$d_4 = \left(\frac{\sqrt{5}-1}{2}\right)(2.9442 - 2.472133) = 0.29175$$

$$x_1 = 2.472133 + d_4 = 2.7638$$

$$x_2 = 2.9442 - d_4 = 2.6524$$

$$f(x_1) = -5.55331$$

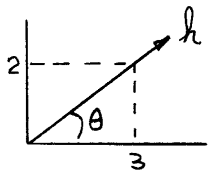
$$f(x_2) = -5.4082$$

$\therefore$ at time $t = 2.76$, minimum pressure is –5.55331

# CHAPTER 14

**14.1** $\dfrac{\partial f}{\partial x} = 8 \qquad \dfrac{\partial f}{\partial y} = 4$

at $x = 2$, $y = 2$




$\theta = \tan^{-1}(2/3)$

$= 33.69°$

$g'(0) = 8\cos(33.69) + 4\sin(33.69) = 8.875$

**14.2**

a) $\nabla f = \begin{bmatrix} 2y^2 + 3ye^{xy} \\ 4xy + 3xe^{xy} \end{bmatrix}$

$$H = \begin{bmatrix} 3y^2 e^{xy} & 4y + 3xye^{xy} + 3e^{xy} \\ 4y + 3yxe^{xy} + 3e^{xy} & 4x + 3x^2 e^{xy} \end{bmatrix}$$

b) $\nabla f = \begin{bmatrix} 2x \\ 2y \\ 4z \end{bmatrix}$

$$H = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 4 \end{bmatrix}$$

c) $\nabla f \begin{bmatrix} \dfrac{2x + 2y}{x^2 + 2xy + 3y^2} \\ \dfrac{2x + 6y}{x^2 + 2xy + 3y^2} \end{bmatrix}$

c) $$H = \frac{\begin{bmatrix} -2x^2 + 2y^2 - 4xy & -2x^2 - 6y^2 - 12xy \\ -2x^2 - 6y^2 - 12xy & 2x^2 - 24y^2 - 12xy \end{bmatrix}}{(x^2 + 2xy + 3y^2)^2}$$

**14.3** Set $\dfrac{\partial f}{\partial x} = \dfrac{\partial f}{\partial y} = 0$

$-2.5x + 2y = 0$
$2x - 4y = -1.5$

solving $x = 0.5$
$y = 0.625$

14.4

a)

$$\frac{\partial f}{\partial x} = 2y - 2.5x$$

$$\frac{\partial f}{\partial y} = 2x + 1.5 - 4y$$

at $x_0 = 1$, $y_0 = 1$

$$\frac{\partial f}{\partial x} = -.5 \qquad \frac{\partial f}{\partial y} = -.5$$

Search Direction $= -.5i - .5j$

$f(1-.5h, 1-.5h)$

$$= 1.5 - .75h - 1.25(1 - h + .25h^2)$$

$$g(h) = .25 + .5h - .3125h^2$$

setting $g'(h) = 0$ gives

$$h^* = .8$$

$\therefore$ $x_1 = .6$ $\quad y_1 = .6$

$$\frac{\partial f}{\partial x} = -.3 \qquad \frac{\partial f}{\partial y} = .3$$

Search Direction $= -.3i + .3j$

$f(.6 - .3h, .6 + .3h)$

$$g(h) = .45 + .18h - .1125h^2$$

setting $g'(h) = 0$ gives

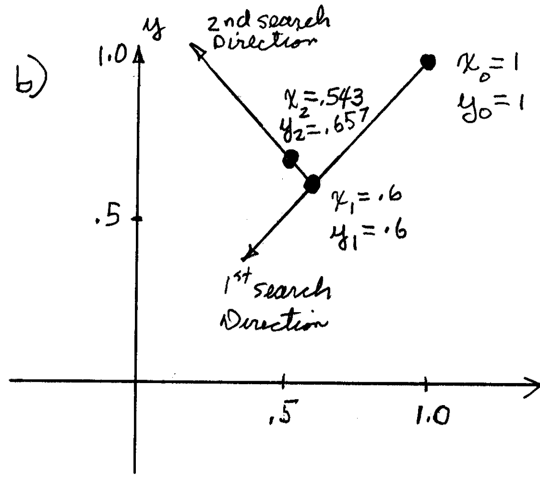
$$h^* = 0.19$$

$\therefore$ $x_2 = .6 + (.19)(-.3) = .543$

$y_2 = .6 + (.19)(.3) = .657$

etc

b)




14.5 $\quad \dfrac{\partial f}{\partial x} = 2(x-2) \qquad \dfrac{\partial f}{\partial y} = 2(y-3)$

at $x = 1$ and $y = 1$

$$\frac{\partial f}{\partial x} = -2 \qquad \frac{\partial f}{\partial y} = -4$$

$f(1-2h, 1-4h)$

$$= (1 - 2h - 2)^2 + (1 - 4h - 3)^2$$

$$g(h) = (-2h-1)^2 + (-4h-2)^2$$

setting $g'(h) = 0$ gives

$$h^* = -1/2$$

$$x_1 = 1 - (2)(-\tfrac{1}{2}) = 2$$

$$y_1 = 1 - (4)(-\tfrac{1}{2}) = 3$$

converges exactly in 1 iteration

14.6 $\frac{\partial f}{\partial x} = 3.5 + 2x - 4x^3 - 2y$

$\frac{\partial f}{\partial y} = 2 - 2x - 2y$

at $x=0$ , $y=0$

$\frac{\partial f}{\partial x} = 3.5 \qquad \frac{\partial f}{\partial y} = 2$

$f(0+3.5h,\ 0+2h) =$

$(3.5)^2 h + 4h + (3.5)^2 h^2 - (3.5)^4 h^4$
$- 2(3.5)(2)h^2 - 4h^2$

$\frac{\partial f}{\partial h} = 0 = 16.25 - 11.5h - 600.25h^3$

Using Bisection gives
$h^* = 0.279$

$\therefore$

$x_1 = 0 + 0.279(3.5)$

$= 0.9765$

$y_1 = 0 + 0.279(2)$

$= 0.558$

$f(0-7h, 0+11h) = g(h)$

$g(h) = 454.8h^2 + 170h$

at $g'(h) = 0$ , $h^* = -0.1869$

$\therefore$

$x_1 = 0 + 7(0.1869) = +1.31$
$y_1 = 0 - 11(0.1869) = -2.056$

14.7 $\frac{\partial f}{\partial x} = -7 + 2.4x - 2y$

$\frac{\partial f}{\partial y} = 11 + 4y - 2x$

at $x=0$, $y=0$, $\quad \frac{\partial f}{\partial x} = -7$

$\frac{\partial f}{\partial y} = 11$

14.8 **Errata: p. 357; The initial value of the variable maxf must be set to some ridiculously small value before the iterations are begun. Add the following line to the beginning of the VBA code:**

```
maxf = -999E9
```

The following code implements the random search algorithm in VBA:

```
Option Explicit
Sub RandSearch()

Dim n As Long
Dim xmin As Single, xmax As Single, ymin As Single, ymax As Single
Dim maxf As Single, maxx As Single, maxy As Single

xmin = -2
xmax = 2
ymin = -2
ymax = 2

n = InputBox("n=")
Call RndSrch(n, xmin, xmax, ymin, ymax, maxy, maxx, maxf)

MsgBox maxf
MsgBox maxx
MsgBox maxy

End Sub
Sub RndSrch(n, xmin, xmax, ymin, ymax, maxy, maxx, maxf)
Dim j As Long
Dim x As Single, y As Single, fn As Single
maxf = -999E9
For j = 1 To n
  x = xmin + (xmax - xmin) * Rnd
  y = ymin + (ymax - ymin) * Rnd
  fn = f(x, y)
  If fn > maxf Then
    maxf = fn
    maxx = x
    maxy = y
   End If
Next j
End Sub

Function f(x, y)
f = 3.5 * x + 2 * y + x ^ 2 - x ^ 4 - 2 * x * y - y ^ 2
End Function
```

14.9 The following code implements the grid search algorithm in VBA:

```
Option Explicit
Sub GridSearch()

Dim nx As Long, ny As Long
Dim xmin As Single, xmax As Single, ymin As Single, ymax As Single
Dim maxf As Single, maxx As Single, maxy As Single

xmin = -2
xmax = 2
ymin = -2
ymax = 2
nx = 1000
ny = 1000
```

```
Call GridSrch(nx, ny, xmin, xmax, ymin, ymax, maxy, maxx, maxf)

MsgBox maxf
MsgBox maxx
MsgBox maxy

End Sub
Sub GridSrch(nx, ny, xmin, xmax, ymin, ymax, maxy, maxx, maxf)
Dim i As Long, j As Long
Dim x As Single, y As Single, fn As Single
Dim xinc As Single, yinc As Single
xinc = (xmax - xmin) / nx
yinc = (ymax - ymin) / ny
maxf = -999000000000#
x = xmin
For i = 0 To nx
  y = ymin
  For j = 0 To ny
    fn = f(x, y)
    If fn > maxf Then
      maxf = fn
      maxx = x
      maxy = y
     End If
     y = y + yinc
  Next j
  x = x + xinc
Next i
End Sub

Function f(x, y)
f = y - x - 2 * x ^ 2 - 2 * x * y - y ^ 2
End Function
```

14.10

$$f(x,y) = 5x^2y - 8y^2 - 7x^2$$

$$\frac{\partial f}{\partial x} = 10xy - 14x \Rightarrow 10(2)(4) - 14(4) = 24$$

$$\frac{\partial f}{\partial y} = 5x^2 - 16y \Rightarrow 5(4)^2 - 16(2) = 48$$

$$\nabla f = 24\hat{i} + 48\hat{j}$$

$$f\left(x_o + \frac{\partial f}{\partial x}h, y_o + \frac{\partial f}{\partial y}h\right) = f(4 + 24h, 2 + 48h)$$

$$= 5(4+24h)^2(2+48h) - 8(2+48h)^2 - 7(4+24h)^2$$

$$g(x) = 138240h^3 + 29376h^2 + 2880h + 16$$

14.11

$$f(x,y) = 2x^3y^2 - 6yx + x^2 + 4y$$

$$\frac{\partial f}{\partial x} = 6x^2y^2 - 6y + 2x \Rightarrow 6(1)(1) - 6(1) + 2(1) = 2$$

$$\frac{\partial f}{\partial x} = 4x^3y - 6y + 4 \Rightarrow 4(1)(1) - 6(1) + 4 = 2$$

$$\nabla f = 2\hat{i} + 2\hat{j}$$

$$f(x_o + \frac{2f}{2x}h\,,\, y_o + \frac{2f}{2y}h) = f(1+2h\,, 1+2h)$$

$$= 2(1+2h)^3(1+2h)^2 - 6(1+2h)(1+2h) + (1+2h)^2 + 4(1+2h)$$

$$g(x) = 64h^5 + 160h^4 + 160h^3 + 60h^2 + 8h + 1$$

# CHAPTER 15

15.1 (Note: Although it is not really clear from the problem statement, it should be assumed that each unit of product is equivalent to a kg.)

(*a*) Define $x_a$ = amount of product A produced, and $x_b$ = amount of product B produced. The objective function is to maximize profit,

$$P = 45x_a + 30x_b$$

Subject to the following constraints

$$\begin{array}{ll} 20x_a + 5x_b \le 10000 & \{\text{raw materials}\} \\ 0.05x_a + 0.15x_b \le 40 & \{\text{production time}\} \\ x_a + x_b \le 550 & \{\text{storage}\} \\ x_a, x_b \ge 0 & \{\text{positivity}\} \end{array}$$

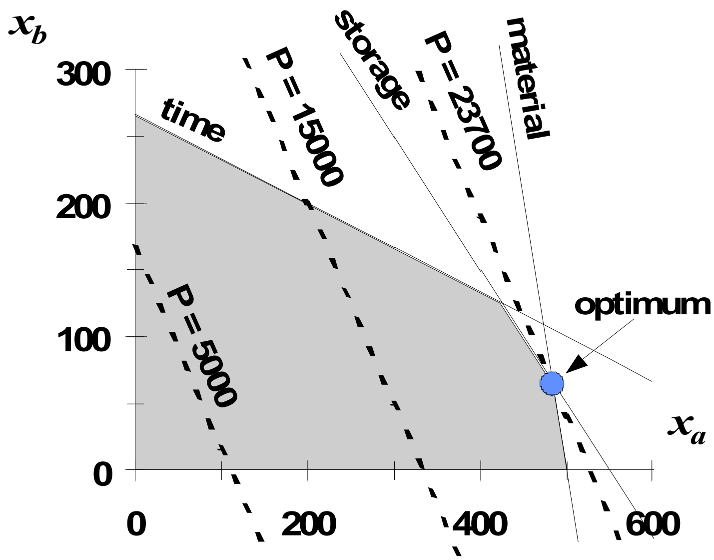
(*b*) To solve graphically, the constraints can be reformulated as the following straight lines

$$\begin{array}{ll} x_b = 2000 - 4x_a & \{\text{raw materials}\} \\ x_b = 266.667 - 0.3333x_a & \{\text{production time}\} \\ x_b = 550 - x_a & \{\text{storage}\} \end{array}$$

The objective function can be reformulated as

$$x_b = (1/30)P - 1.5x_a$$

The constraint lines can be plotted on the $x_b$-$x_a$ plane to define the feasible space. Then the objective function line can be superimposed for various values of $P$ until it reaches the boundary. The result is $P \cong 23700$ with $x_a \cong 483$ and $x_b \cong 67$. Notice also that material and storage are the binding constraints and that there is some slack in the time constraint.




(*c*) The simplex tableau for the problem can be set up and solved as

| | Basis | P | xa | xb | S1 | S2 | S3 | Solution | Intercept |
|---|---|---|---|---|---|---|---|---|---|
| | P | 1 | -45 | -30 | 0 | 0 | 0 | 0 | |
| material | S1 | 0 | 20 | 5 | 1 | 0 | 0 | 10000 | 500 |
| time | S2 | 0 | 0.05 | 0.15 | 0 | 1 | 0 | 40 | 800 |
| storage | S3 | 0 | 1 | 1 | 0 | 0 | 1 | 550 | 550 |

| | Basis | P | xa | xb | S1 | S2 | S3 | Solution | Intercept |
|---|---|---|---|---|---|---|---|---|---|
| | P | 1 | 0 | -18.75 | 2.25 | 0 | 0 | 22500 | |
| xa | xa | 0 | 1 | 0.25 | 0.05 | 0 | 0 | 500 | 2000 |
| time | S2 | 0 | 0 | 0.1375 | -0 | 1 | 0 | 15 | 109.0909 |
| storage | S3 | 0 | 0 | 0.75 | -0.05 | 0 | 1 | 50 | 66.66667 |

| | Basis | P | xa | xb | S1 | S2 | S3 | Solution | Intercept |
|---|---|---|---|---|---|---|---|---|---|
| | P | 1 | 0 | 0 | 1 | 0 | 25 | 23750 | |
| xa | xa | 0 | 1 | 0 | 0.067 | 0 | -0.333 | 483.33333 | |
| time | S2 | 0 | 0 | 0 | 0.007 | 1 | -0.183 | 5.8333333 | |
| xb | xb | 0 | 0 | 1 | -0.07 | 0 | 1.333 | 66.666667 | |

(*d*) An Excel spreadsheet can be set up to solve the problem as

| | A | B | C | D | E |
|---|---|---|---|---|---|
| 1 | | xA | xB | total | constraint |
| 2 | amount | 0 | 0 | | |
| 3 | time | 0.05 | 0.15 | 0 | 40 |
| 4 | storage | 1 | 1 | 0 | 550 |
| 5 | raw material | 20 | 5 | 0 | 10000 |
| 6 | profit | 45 | 30 | 0 | |

The Solver can be called and set up as

Set target cell: D6
Equal to ● max ❍ min ❍ value of 0

By changing cells
B2:C2
Subject to constraints:
D3≤E3
D4≤E4
D5≤E5

The resulting solution is

| | A | B | C | D | E |
|---|---|---|---|---|---|
| 1 | | xA | xB | total | constraint |
| 2 | amount | 483.3333 | 66.66667 | | |
| 3 | time | 0.05 | 0.15 | 34.16667 | 40 |
| 4 | storage | 1 | 1 | 550 | 550 |
| 5 | raw material | 20 | 5 | 10000 | 10000 |
| 6 | profit | 45 | 30 | 23750 | |

In addition, a sensitivity report can be generated as

**Microsoft Excel 5.0c Sensitivity Report**
**Worksheet: [PROB1501.XLS]Sheet2**
**Report Created: 12/8/97 17:06**

Changing Cells

| Cell | Name | Final Value | Reduced Cost | Objective Coefficient | Allowable Increase | Allowable Decrease |
|---|---|---|---|---|---|---|
| $B$2 | amount xA | 483.3333333 | 0 | 45 | 75 | 15 |
| $C$2 | amount xB | 66.66666667 | 0 | 30 | 15 | 18.75 |

Constraints

| Cell | Name | Final Value | Shadow Price | Constraint R.H. Side | Allowable Increase | Allowable Decrease |
|---|---|---|---|---|---|---|
| $D$3 | time | 34.16666667 | 0 | 40 | 1E+30 | 5.833333333 |
| $D$4 | storage | 550 | 25 | 550 | 31.81818182 | 1E+30 |
| $D$5 | raw material | 10000 | 1 | 10000 | 1E+30 | 875 |

(*e*) The high shadow price for storage from the sensitivity analysis from (*d*) suggests that increasing storage will result in the best increase in profit.

15.2 (*a*) The total LP formulation is given by

$$\text{Maximize } Z = 150x_1 + 175x_2 + 250x_3 \quad \{\text{Maximize profit}\}$$

subject to

$$7x_1 + 11x_2 + 15x_3 \le 154 \quad \{\text{Material constraint}\}$$
$$10x_1 + 8x_2 + 12x_3 \le 80 \quad \{\text{Time constraint}\}$$
$$x_1 \le 9 \quad \{\text{“Regular” storage constraint}\}$$
$$x_2 \le 6 \quad \{\text{“Premium” storage constraint}\}$$
$$x_2 \le 5 \quad \{\text{“Supreme” storage constraint}\}$$

$x_1, x_2, x_3 \geq 0$ {Positivity constraints}

(*b*) The simplex tableau for the problem can be set up and solved as

| Basis | P | x1 | x2 | x3 | S1 | S2 | S3 | S4 | S5 | Solution | Intercept |
|---|---|---|---|---|---|---|---|---|---|---|---|
| P | 1 | -150 | -175 | -250 | 0 | 0 | 0 | 0 | 0 | 0 | |
| S1 | 0 | 7 | 11 | 15 | 1 | 0 | 0 | 0 | 0 | 154 | 10.2667 |
| S2 | 0 | 10 | 8 | 12 | 0 | 1 | 0 | 0 | 0 | 80 | 6.66667 |
| S3 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 9 | ∞ |
| S4 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 6 | ∞ |
| S5 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 5 | 5 |

| Basis | P | x1 | x2 | x3 | S1 | S2 | S3 | S4 | S5 | Solution | Intercept |
|---|---|---|---|---|---|---|---|---|---|---|---|
| P | 1 | -150 | -175 | 0 | 0 | 0 | 0 | 0 | 250 | 1250 | |
| S1 | 0 | 7 | 11 | 0 | 1 | 0 | 0 | 0 | -15 | 79 | 7.18182 |
| S2 | 0 | 10 | 8 | 0 | 0 | 1 | 0 | 0 | -12 | 20 | 2.5 |
| S3 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 9 | ∞ |
| S4 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 6 | 6 |
| x3 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 5 | ∞ |

| Basis | P | x1 | x2 | x3 | S1 | S2 | S3 | S4 | S5 | Solution | Intercept |
|---|---|---|---|---|---|---|---|---|---|---|---|
| P | 1 | 68.75 | 0 | 0 | 0 | 21.88 | 0 | 0 | -12.5 | 1687.5 | |
| S1 | 0 | -6.75 | 0 | 0 | 1 | -1.375 | 0 | 0 | 1.5 | 51.5 | 34.3333 |
| x2 | 0 | 1.25 | 1 | 0 | 0 | 0.125 | 0 | 0 | -1.5 | 2.5 | -1.66667 |
| S3 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 9 | ∞ |
| S4 | 0 | -1.25 | 0 | 0 | 0 | -0.125 | 0 | 1 | 1.5 | 3.5 | 2.33333 |
| x3 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 5 | 5 |

| Basis | P | x1 | x2 | x3 | S1 | S2 | S3 | S4 | S5 | Solution |
|---|---|---|---|---|---|---|---|---|---|---|
| P | 1 | 58.3333 | 0 | 0 | 0 | 20.83 | 0 | 8.33 | 0 | 1716.7 |
| S1 | 0 | -5.5 | 0 | 0 | 1 | -1.25 | 0 | -1 | 0 | 48 |
| x2 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 6 |
| S3 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 9 |
| S5 | 0 | -0.8333 | 0 | 0 | 0 | -0.083 | 0 | 0.67 | 1 | 2.3333 |
| x3 | 0 | 0.83333 | 0 | 1 | 0 | 0.083 | 0 | -0.67 | 0 | 2.6667 |

(*c*) An Excel spreadsheet can be set up to solve the problem as

| | A | B | C | D | E | F |
|---|---|---|---|---|---|---|
| 1 | | regular | premium | supreme | total | constraint |
| 2 | amount | 0 | 0 | 0 | | |
| 3 | material | 7 | 11 | 15 | 0 | 154 |
| 4 | time | 10 | 8 | 12 | 0 | 80 |
| 5 | reg stor | 1 | 0 | 0 | 0 | 9 |
| 6 | prem stor | 0 | 1 | 0 | 0 | 6 |
| 7 | sup stor | 0 | 0 | 1 | 0 | 5 |
| 8 | profit | 150 | 175 | 250 | 0 | |

The Solver can be called and set up as

Set target cell: E8
Equal to ● max ❍ min ❍ value of 0

By changing cells
B2:D2
Subject to constraints:
E3≤F3
E4≤F4
E5≤F5
E6≤F6
E7≤F7
B2≥0
C2≥0
D2≥0

The resulting solution is

| | A | B | C | D | E | F |
|---|---|---|---|---|---|---|
| 1 | | regular | premium | supreme | total | constraint |
| 2 | amount | 0 | 6 | 2.666667 | | |
| 3 | material | 7 | 11 | 15 | 106 | 154 |
| 4 | time | 10 | 8 | 12 | 80 | 80 |
| 5 | reg stor | 1 | 0 | 0 | 0 | 9 |
| 6 | prem stor | 0 | 1 | 0 | 6 | 6 |
| 7 | sup stor | 0 | 0 | 1 | 2.666667 | 5 |
| 8 | profit | 150 | 175 | 250 | 1716.667 | |

In addition, a sensitivity report can be generated as

**Microsoft Excel 5.0c Sensitivity Report**
**Worksheet: [PROB1502.XLS]Sheet4**
**Report Created: 12/12/97 9:53**

Changing Cells

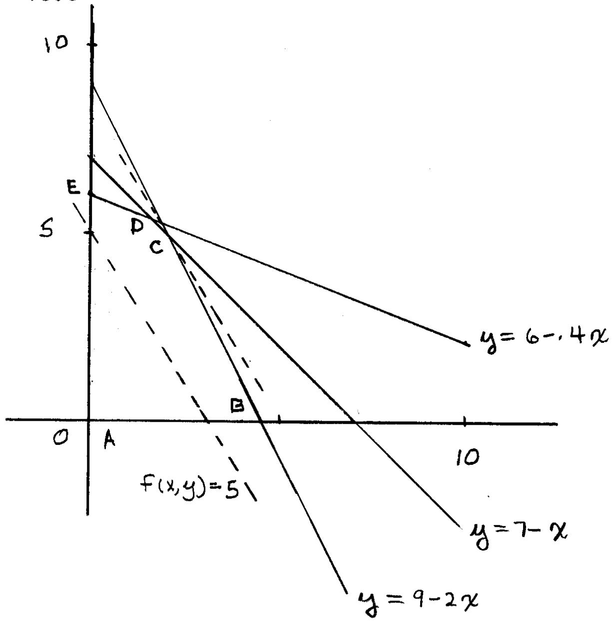
| Cell | Name | Final Value | Reduced Cost | Objective Coefficient | Allowable Increase | Allowable Decrease |
|---|---|---|---|---|---|---|
| $B$2 | amount regular | 0 | -58.33333333 | 150 | 58.33333333 | 1E+30 |
| $C$2 | amount premium | 6 | 0 | 175 | 1E+30 | 8.333333333 |
| $D$2 | amount supreme | 2.666666667 | 0 | 250 | 12.5 | 70 |

Constraints

| Cell | Name | Final Value | Shadow Price | Constraint R.H. Side | Allowable Increase | Allowable Decrease |
|---|---|---|---|---|---|---|
| $E$3 | material total | 106 | 0 | 154 | 1E+30 | 48 |
| $E$4 | time total | 80 | 20.83333333 | 80 | 28 | 32 |
| $E$5 | reg stor total | 0 | 0 | 9 | 1E+30 | 9 |
| $E$6 | prem stor total | 6 | 8.333333333 | 6 | 4 | 3.5 |
| $E$7 | sup stor total | 2.666666667 | 0 | 5 | 1E+30 | 2.333333333 |

(*d*) The high shadow price for time from the sensitivity analysis from (*c*) suggests that increasing time will result in the best increase in profit.

15.3 a)




we observe f(x,y) occurs at point where

$$x + y = 7$$
$$2x + y = 9$$

are both satisfied,
solving $x = 2 \quad y = 5$

and $z = f(2,5) = 8.33$

b) maximize $z = 5/3\,x + y$
Subject to

$$\begin{aligned} x + 2.5y + S_1 \qquad\quad &= 15 \\ x + \quad y \quad + S_2 \quad &= 7 \\ 2x + \quad y \qquad + S_3 &= 9 \end{aligned}$$

$$x, y, S_1, S_2, S_3 \geq 0$$

at A $\quad x = 0 \quad y = 0$
$S_1 = 15$
$S_2 = 7$
$S_3 = 9$
$z = 0$

at B $\quad y = 0 \quad S_3 = 0$

$$\begin{aligned} x \quad + S_1 \quad &= 15 \\ x \quad\quad + S_2 &= 7 \\ 2x \quad\quad &= 9 \end{aligned}$$

$$x = 4.5$$
$$S_1 = 10.5$$
$$S_2 = 2.5$$

$$Z = 7.5$$

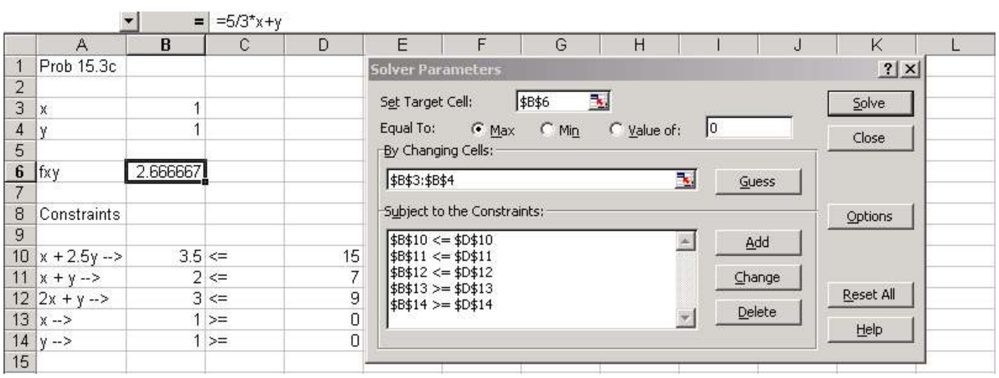
at C $\quad S_3 = 0 \quad S_2 = 0$

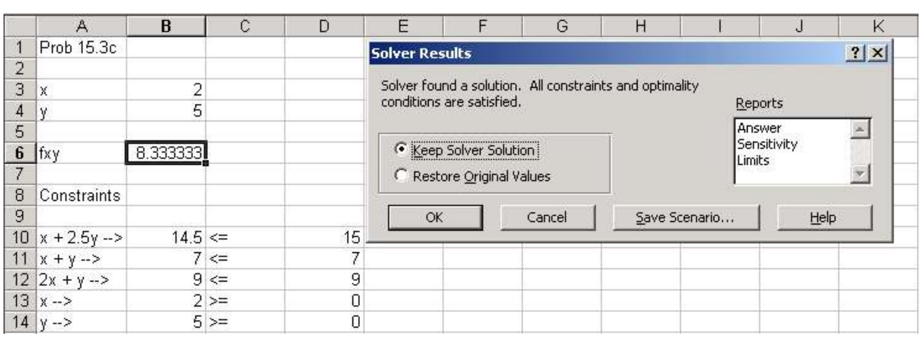
$$\begin{aligned} x + 2.5y + S_1 &= 15 \\ x + y &= 7 \\ 2x + y &= 9 \end{aligned}$$

$$x = 2$$
$$y = 5$$
$$S_1 = .5$$

$$Z = 8.33$$

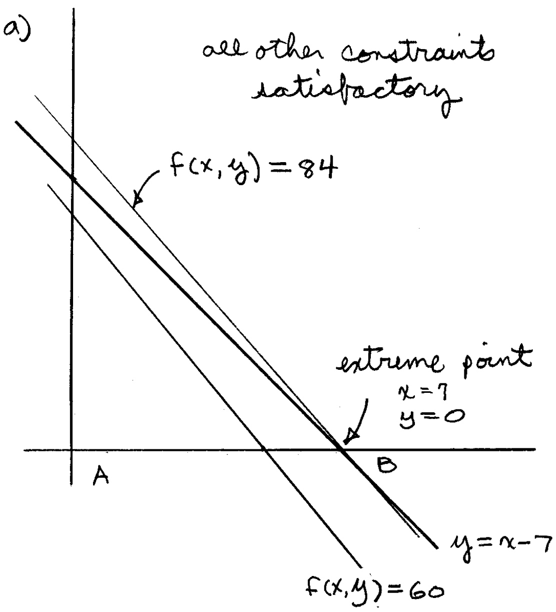
(*c*) Using the Excel Solver



15.4

a)




b) maximize $Z = 12x + 10y$

subject to:

$$\begin{aligned} 5x + 4y + S_1 \quad\quad\quad &= 1700 \\ x + y \quad + S_2 \quad\quad &= 7 \\ 4.5x + 3.5y \quad\quad + S_3 \quad &= 1600 \\ x + 2y \quad\quad\quad + S_4 &= 500 \end{aligned}$$

$x, y, S_1, S_2, S_3, S_4 \geq 0$

at A,
$x = 0$
$y = 0$
$S_1 = 1700$
$S_2 = 7$
$S_3 = 1600$
$S_4 = 500$

$Z = 0$

at B,

$$y = 0$$
$$s_2 = 0$$

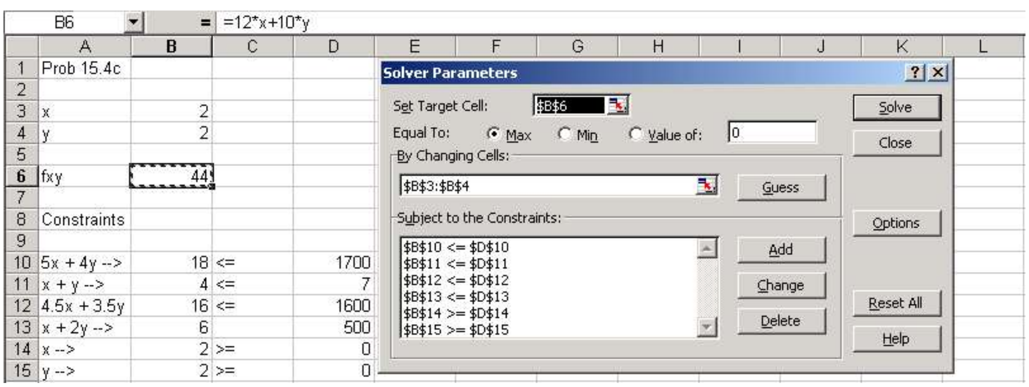
$$5x + s_1 = 1700$$
$$x = 7$$
$$4.5x + s_3 = 1600$$
$$x + s_4 = 500$$

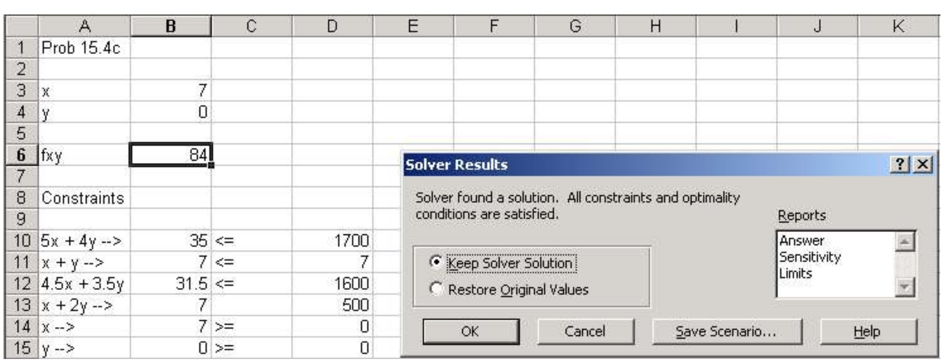
solving

$$x = 7$$
$$s_1 = 1665$$
$$s_3 = 1568.5$$
$$s_4 = 493$$

$$z = 84$$

(*c*) Using the Excel Solver



15.5 Student Specific

Excel gives results as

$x = 0.383602$
$y = 0.516398$

$\max f(x,y) = 1.265412$

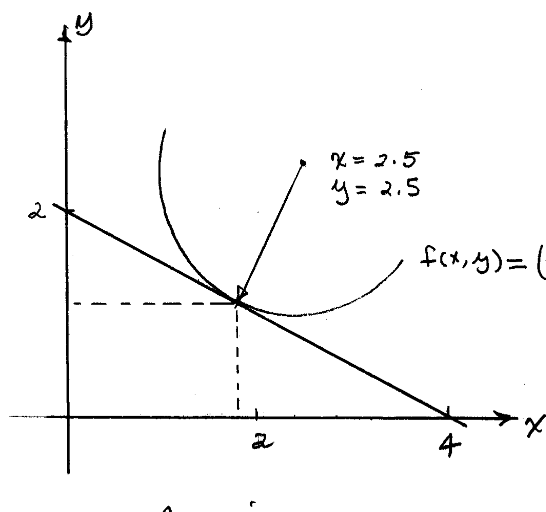
15.6 Student Specific

Excel gives results as

$x = 0.843539$
$y = 0.488307$

$\max f(x,y) = 21.39861$

15.7




Excel gives

$x = 1.8$
$y = 1.1$

$\max f(x,y) = 2.45$

15.8 Student specific
Excel gives

$x = 0.5$
$y = 0.625$
$f(x,y) = 0.46875$

15.9 Student specific
Excel gives

$x = 1.151388$
$y = -0.15139$

$f(x,y) = 3.621005$

15.10 a) Student specific
b) Excel gives

$x = 1.07143$
$y = -2.21429$

c) $f(x,y) = -15.9286$

d) $\dfrac{\partial^2 f}{\partial x^2} = 2.4 \quad \dfrac{\partial^2 f}{\partial y^2} = 4$

$\dfrac{\partial}{\partial y}\left(\dfrac{\partial f}{\partial x}\right) = -2$

$H = \begin{bmatrix} 2.4 & -2 \\ -2 & 4 \end{bmatrix} \quad |H| = 5.4$

minimum because

$|H| > 0$ and $\dfrac{\partial^2 f}{\partial x^2} > 0$

15.11

Total surface area = $\pi DH + 2\left(\frac{\pi D^2}{4}\right)$

Minimize $f(D,H) = \pi DH + \frac{\pi D^2}{2}$

Constraints:

$$\frac{\pi D^2 H}{4} \geq 320$$
$$3 \leq D \leq 10$$
$$2 \leq H \leq 10$$

$$H = \frac{A}{\pi D} - \frac{D}{2}$$
$$H = 407.43 / D^2$$

when $A \approx 260\text{cm}^2$

$\text{H} = 8.41\text{cm}$

$\text{D} = 6.96\text{cm}$

15.12

Profit: $z = 13{,}000x_1 + 15{,}000x_2$

Constraints:

1. $17.5x_1 + 21x_2 \leq 8000$
2. $680x_1 + 500x_2 \leq 240000$
3. $x_1 \leq 400$
4. $x_2 \leq 350$

5,6. $x_1, x_2 \leq 0$

$x_2 = 224.2$ cars

$\text{x}_1 = 188.1$ cars

$\text{z} = \$5{,}810{,}000$

# Chapter 17

17-1

$$\frac{\sum_{i=1}^{25} y_i}{25} = \text{mean}$$

a) mean = 1.6244

b) $s_y = 0.33939$

c) $s_y^2 = 0.115184$

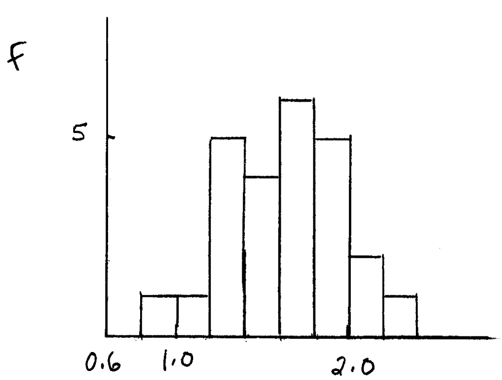
d) c.v. = 20.471 %

e) $t_{0.05/2,\,25-1} = 2.0639$

L = 1.6017134
U = 1.6470866

17.2

| Interval | Frequency |
|---|---|
| 0.6 – 0.8 | 0 |
| 0.8 – 1.0 | 1 |
| 1.0 – 1.2 | 1 |
| 1.2 – 1.4 | 5 |
| 1.4 – 1.6 | 4 |
| 1.6 – 1.8 | 6 |
| 1.8 – 2.0 | 5 |
| 2.0 – 2.2 | 2 |
| 2.2 – 2.4 | 1 |




17.3

$$\text{mean} = \frac{\sum_{i=1}^{27} y_i}{27}$$

a) = 27.67857

b) $s_y = 10.801$

c) $s_y^2 = 116.671$

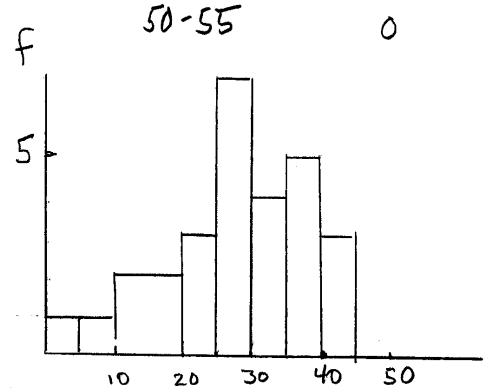
d) c.v = 38.32 %

e) $t_{0.1/2,\,27-1} = 1.7056$

L = 27.032
U = 28.325

f)

| Interval | Frequency |
|---|---|
| 0-5 | 1 |
| 5-10 | 1 |
| 10-15 | 2 |
| 15-20 | 2 |
| 20-25 | 3 |
| 25-30 | 7 |
| 30-35 | 4 |
| 35-40 | 5 |
| 40-45 | 3 |
| 45-50 | 0 |
| 50-55 | 0 |



17.3

a) 68% should fall between

$27.68 - 10.8 = 16.88$

and

$27.68 + 10.8 = 38.48$

actually $\frac{19}{28} = 67.8$

fall within range

17.4 use Toolkit with x vs y

$y = 3.3888 + 0.3725(x)$

$S_{x/y} = 1.232$

$r^2 = 0.81066$

$r = 0.90$

with y vs x

$x = -5.3869 + 2.1763(y)$

$S_{y/x} = 2.977$

$r^2 = 0.81066$

$r = 0.90$

therefore "best" lines, and $S_{x/y}$ and $S_{y/x}$ differ.

$r^2$ and $r$ stay the same

17.5 Using Toolkit best line becomes

$y = 30.74 - 0.7719\,x$

at $x = 5$ the predicted value $y(5) = 26.88$

compared to measured

$y(5) = 5$

The original standard error of the estimate $= 4.39$

$26.88 - (2)(4.39) = 18.1$

measured value of 5 is greater than 2 times standard error and therefore probably erroneous.

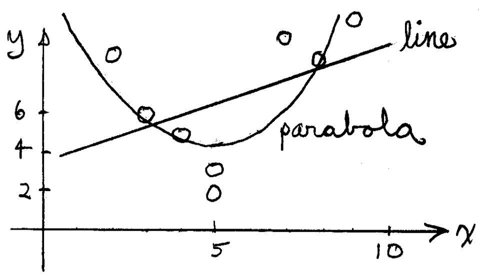
17.6

a) $y = 3.48955 + 0.62985(x)$

Standard error $= 3.221$

correlation coeff $= 0.4589$

b) $y = 16.027 - 4.807\,x + 0.4889\,x^2$



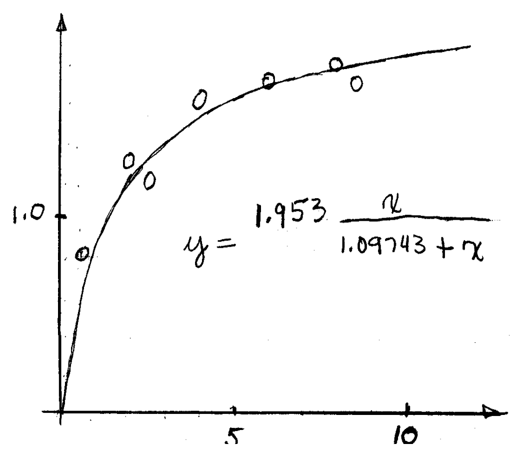

17.7 Regress $1/x$ vs $1/y$ using Toolkit gives

$\frac{1}{y} = 0.5120351 + 0.561923\left(\frac{1}{x}\right)$

standard error $= 0.0487$

$\frac{1}{0.512035} = 1.953$ and

$0.561923(1.953) = 1.09743$

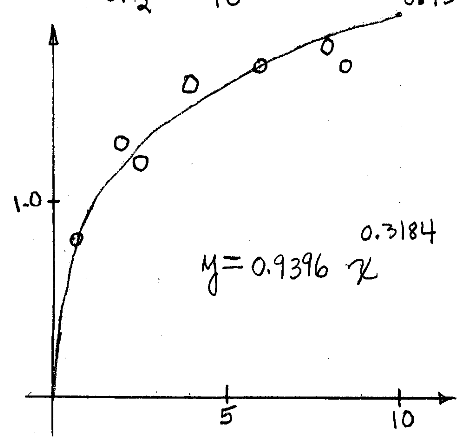



17.8 Regress $\log x$ vs $\log y$ using Toolkit

$\log y = -0.02705 + 0.3184 \log x$

standard Error $= 0.0361$

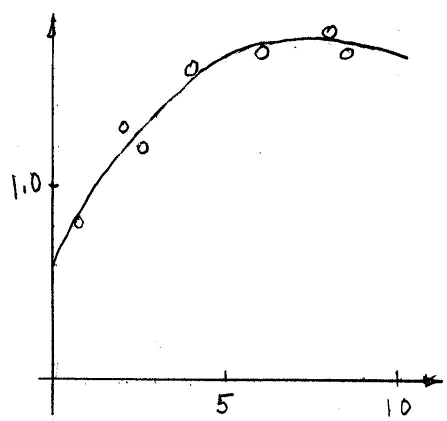
$a_2 = 10^{-0.02705} = 0.9396$




17.9 Use Toolkit with $n = 2$

$y = 0.5967 + 0.3391\,(x) - 0.02433\,(x^2)$

standard Error $= 0.08946$



17.10

| $\log x$ | $\log y$ |
|---|---|
| 0.398 | 0.845 |
| 0.544 | 0.740 |
| 0.699 | 0.591 |
| 0.778 | 0.556 |
| 0.875 | 0.491 |
| 1.0 | 0.447 |
| 1.097 | 0.415 |
| 1.176 | 0.380 |
| 1.243 | 0.362 |
| 1.301 | 0.362 |

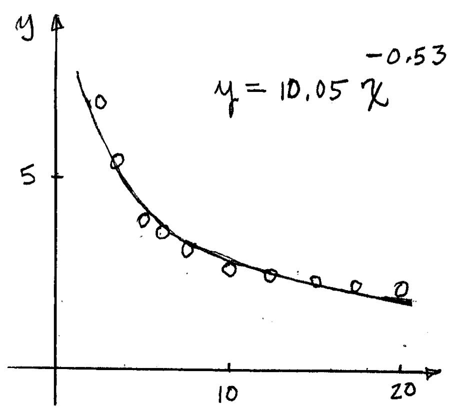
fit straight line

$a_0 = 1.002 \quad a_1 = -0.53$

Standard Error = 0.039

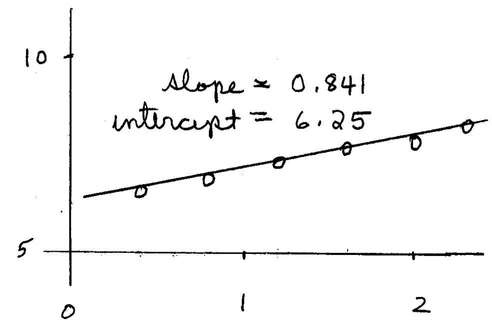
$b_2 = a_1 \quad \log a_2 = a_0$

$a_2 = 10^{1.002} = 10.05$



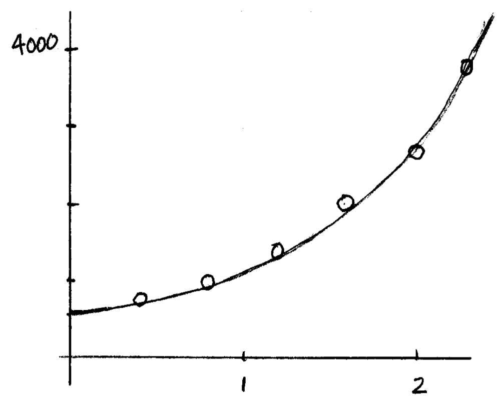

17.11

| $x$ | $\ln y$ |
|---|---|
| 0.4 | 6.62 |
| 0.8 | 6.91 |
| 1.2 | 7.24 |
| 1.6 | 7.60 |
| 2.0 | 7.90 |
| 2.3 | 8.23 |




$e^{6.25} = 518$

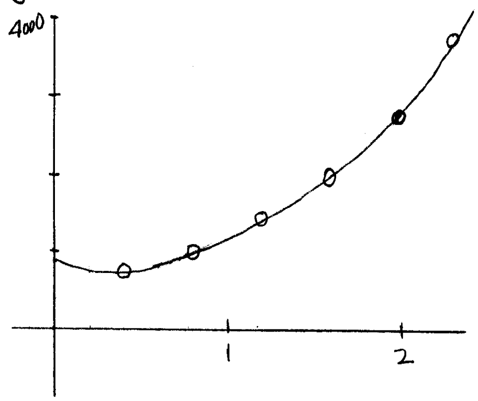
model $y = 518\, e^{0.841x}$



17.12

Standard Error = 112.7

$y = 854.66 - 453.99x + 726.77x^2$




d) $y = 12.17 + 1.35x - 0.01545x^2$

standard error = 2.15

c) is best because standard error is smallest. also model form may be best as well because both data and c) model as $x \to$ large

$y \to$ constant

17.13

a) $y = 20.667 + 0.496x$

standard error = 3.728

b) fit $\log x$ vs $\log y$ gives

$\log y = 0.9848 + 0.395 \log x$

with standard error = 0.036

or $y = 9.656\, x^{0.395}$

c) fit $\frac{1}{x}$ vs $\frac{1}{y}$ gives

$1/y = 0.01914 + 0.213 \frac{1}{x}$

with standard error = 0.0011

or $y = 52.2 \dfrac{x}{11.1 + x}$

17.14

$\sum y = 242.7$

$\sum x_1 = 20$

$\sum x_2 = 12$

$\sum x_1^2 = 60$

$\sum x_2^2 = 20$

$\sum x_1 x_2 = 30$

$\sum x_1 y = 661$

$\sum x_2 y = 331.2$

$$\begin{bmatrix} 9 & 20 & 12 \\ 20 & 60 & 30 \\ 12 & 30 & 20 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} 242.7 \\ 661 \\ 331.2 \end{bmatrix}$$

$a_0 = 14.404$

$a_1 = 9.026$

$a_2 = -5.62$

$S_t = 1060.26$

$S_r = 4.81$

$$S_{y/x_1,x_2} = \sqrt{\frac{4.81}{9-3}} = 0.895$$

$$r^2 = \frac{1060.26 - 4.81}{1060.26} = 0.995$$

$r = 0.997$

17.15

$\sum y = 142$
$\sum x_1 = 9$
$\sum x_2 = 27$
$\sum x_1^2 = 15$
$\sum x_2^2 = 117$
$\sum x_1 x_2 = 33$
$\sum x_1 y = 115$
$\sum x_2 y = 453$

$$\begin{bmatrix} 9 & 9 & 27 \\ 9 & 15 & 33 \\ 27 & 33 & 117 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} 142 \\ 115 \\ 453 \end{bmatrix}$$

$a_0 = 16.67$
$a_1 = -6.3$
$a_2 = 1.8$

$S_t = 281.55$

$S_r = 62.85$

$$S_{y/x_1,x_2} = \sqrt{\frac{62.85}{9-3}} = 3.24$$

$$r^2 = \frac{281.55 - 62.85}{281.55} = 0.777$$

$r = 0.881$

17.16 From Toolkit

$$y = 685.25 - 218.89x + 643.6x^2$$

$$\partial f/\partial a_0 = 1 \qquad \frac{\partial f}{\partial a_1} = x \qquad \frac{\partial f}{\partial a_2} = x^2$$

$$[Z_0] = \begin{bmatrix} 1 & 0.075 & 0.005625 \\ 1 & 0.5 & 0.25 \\ 1 & 1.0 & 1.0 \\ 1 & 1.2 & 1.44 \\ 1 & 1.7 & 2.89 \\ 1 & 2. & 4 \\ 1 & 2.3 & 5.29 \end{bmatrix}$$

$$[Z_0]^T[Z_0] = \begin{bmatrix} 7 & 8.77 & 14.87 \\ 8.77 & 14.87 & 27.93 \\ 14.87 & 27.93 & 55.47 \end{bmatrix}$$

TRY Initial guess $a_0 = 675$
$a_1 = -200$
$a_2 = 650$

$$\{D\} = \begin{bmatrix} -63.6 \\ 62.5 \\ 75.0 \\ 29.0 \\ -163.5 \\ -225.0 \\ 96.5 \end{bmatrix}$$

$$\Delta A = [Z^T Z]^{-1} Z^T \{D\} \quad \text{gives}$$

$$[\Delta A] = \begin{bmatrix} 10.248 \\ -18.88 \\ -6.46 \end{bmatrix}$$

and therefore new a's

$a_0 = 685.248$
$a_1 = -218.88$
$a_2 = 643.60$

almost exact in one iteration

$$\{D\} = \begin{bmatrix} -0.667 \\ 0 \\ 2.0 \\ -0.333 \\ 2.286 \\ -1.5 \\ 0.111 \\ 0 \\ 1.091 \\ 0.333 \end{bmatrix}$$

17.17 use initial guess
$a_3 = 50 \quad b_3 = 10$

$$\frac{\partial f}{\partial a_3} = \frac{x}{x+b_3} \qquad \frac{\partial f}{\partial b_3} = \frac{-a_3 x}{(x+b_3)^2}$$

at the data points

$$[Z_0] = \begin{bmatrix} 0.333 & -1.111 \\ 0.5 & -1.25 \\ 0.6 & -1.2 \\ 0.667 & -1.111 \\ 0.714 & -1.020 \\ 0.75 & -0.938 \\ 0.777 & -0.864 \\ 0.80 & -0.80 \\ 0.8182 & -0.744 \\ 0.833 & -0.694 \end{bmatrix}$$

$$[Z^T][Z] = \begin{bmatrix} 4.847 & -6.387 \\ -6.387 & 9.814 \end{bmatrix}$$

$$\left[[Z^T]\cdot[Z]\right]^{-1} = \begin{bmatrix} 1.45 & 0.944 \\ 0.944 & 0.7161 \end{bmatrix}$$

$$\Delta A = \left[Z^T Z\right]^{-1} Z^T\{D\}$$

$$= \begin{matrix} 0.4887 \\ -0.0237 \end{matrix}$$

and therefore

$a_3 = 50 + 0.4887 = 50.4887$
$b_3 = 10 - 0.0237 = 9.976$

etc with additional iterations

17.18

a) Prob 17.4 gives

$y = 3.3888 + 0.3725\,x$

$s_{x/y} = 1.232$

$$Z = \begin{bmatrix} 1 & 1 \\ 1 & 3 \\ 1 & 5 \\ 1 & 7 \\ 1 & 10 \\ 1 & 12 \\ 1 & 13 \\ 1 & 16 \\ 1 & 18 \\ 1 & 20 \end{bmatrix} \qquad \{Y\} = \begin{bmatrix} 4 \\ 5 \\ 6 \\ 5 \\ 8 \\ 7 \\ 6 \\ 9 \\ 12 \\ 11 \end{bmatrix}$$

$$[Z^T][Z] \quad \{A\} = \{Z^T\}\{Y\}$$

$$\begin{bmatrix} 10 & 105 \\ 105 & 1477 \end{bmatrix} \begin{Bmatrix} a_0 \\ a_1 \end{Bmatrix} = \begin{Bmatrix} 73 \\ 906 \end{Bmatrix}$$

$$\{A\} = [Z^T Z]^{-1} \{Z^T \{Y\}\}$$

$$= \begin{bmatrix} 0.3944 & -0.0280 \\ -0.0280 & 0.00267 \end{bmatrix} \begin{Bmatrix} 73 \\ 906 \end{Bmatrix}$$

$$\{A\} = \begin{Bmatrix} 3.3888 \\ 0.3725 \end{Bmatrix}$$

$$s(a_0) = \sqrt{0.3944\,(1.232)^2} = 0.774$$

$$s(a_1) = \sqrt{0.00267\,(1.232)^2} = 0.0637$$

$$\text{TINV}(0.10, 8) = 1.86$$

$$a_0 = 3.3888 \pm 1.86(0.774)$$

$$a_0 = 3.3888 \pm 1.440$$

$$a_0 = 1.949 \text{ to } 4.828$$

$$a_1 = 0.3725 \pm 1.86\,(0.0637)$$

$$a_1 = 0.2540 \text{ to } 0.4909$$

b) Prob 17.13 gives

$$y = 20.667 + 0.496x$$

$$s_{xy} = 3.728$$

$$\{A\} = [Z^T Z]^{-1} \{Z^T Y\}$$

$$= \begin{bmatrix} 0.4667 & -0.0133 \\ -0.0133 & 0.000425 \end{bmatrix} \begin{Bmatrix} 343 \\ 10455 \end{Bmatrix}$$

$$\{A\} = \begin{Bmatrix} 20.6667 \\ 0.4957 \end{Bmatrix}$$

$$s(a_0) = \sqrt{0.4667\,(3.728)^2} = 2.55$$

$$s(a_1) = \sqrt{0.0004585\,(3.728)^2} = 0.0798$$

$$a_0 = 20.667 \pm 1.86\,(2.55)$$

$$a_0 = 20.667 \pm 4.743$$

$$a_1 = 0.496 \pm 1.86\,(0.0798)$$

$$a_1 = 0.496 \pm 0.1484$$

17.19 Here's VBA code to implement linear regression:

```
Option Explicit

Sub Regres()

Dim n As Integer
Dim x(20) As Single, y(20) As Single, a1 As Single, a0 As Single
Dim syx As Single, r2 As Single

n = 7
x(1) = 1: x(2) = 2: x(3) = 3: x(4) = 4: x(5) = 5
x(6) = 6: x(7) = 7
y(1) = 0.5: y(2) = 2.5: y(3) = 2: y(4) = 4: y(5) = 3.5
y(6) = 6: y(7) = 5.5

Call Linreg(x(), y(), n, a1, a0, syx, r2)
MsgBox "slope= " & a1
MsgBox "intercept= " & a0
MsgBox "standard error= " & syx
MsgBox "coefficient of determination= " & r2
MsgBox "correlation coefficient= " & Sqr(r2)

End Sub

Sub Linreg(x, y, n, a1, a0, syx, r2)

Dim i As Integer
Dim sumx As Single, sumy As Single, sumxy As Single
Dim sumx2 As Single, st As Single, sr As Single
Dim xm As Single, ym As Single

sumx = 0
sumy = 0
sumxy = 0
sumx2 = 0
st = 0
sr = 0
For i = 1 To n
  sumx = sumx + x(i)
  sumy = sumy + y(i)
  sumxy = sumxy + x(i) * y(i)
  sumx2 = sumx2 + x(i) ^ 2
Next i
xm = sumx / n
ym = sumy / n
a1 = (n * sumxy - sumx * sumy) / (n * sumx2 - sumx * sumx)
a0 = ym - a1 * xm
For i = 1 To n
  st = st + (y(i) - ym) ^ 2
  sr = sr + (y(i) - a1 * x(i) - a0) ^ 2
Next i
syx = (sr / (n - 2)) ^ 0.5
r2 = (st - sr) / st

End Sub
```

17.20

| log N | log Stress |
|---|---|
| 0 | 3.053463 |
| 1 | 3.024486 |
| 2 | 2.996949 |
| 3 | 2.903633 |
| 4 | 2.813581 |
| 5 | 2.749736 |
| 6 | 2.630428 |

n =7

$\sum x_i y_i = 58.514$

$\sum x_i^2 = 91$

$\sum x_i = 21$

$\sum y_i = 20.17228$

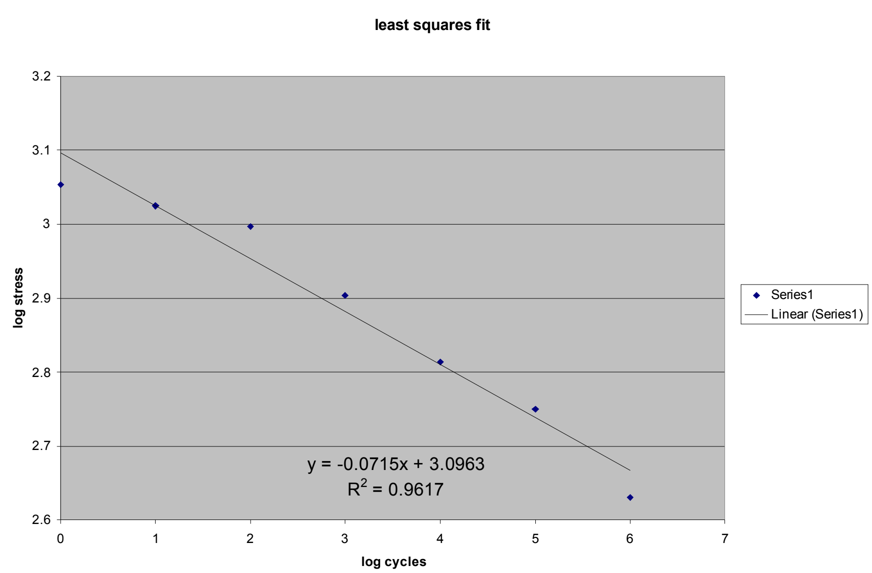
$\bar{x} = 3$

$\bar{y} = 2.8817$

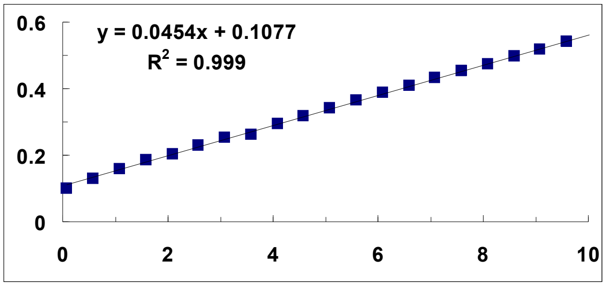
$$a_1 = \frac{n\sum x_i y_i - \sum x_i y_i}{n\sum x_i^2 - (\sum x_i)^2} = \frac{7(58.514) - (21)(20.17228)}{7(91) - (21)^2} = -0.07153$$

$$a_o = \bar{y} - a_1 \bar{x} = 2.8817 - (-0.07153)(3) = 3.09629$$

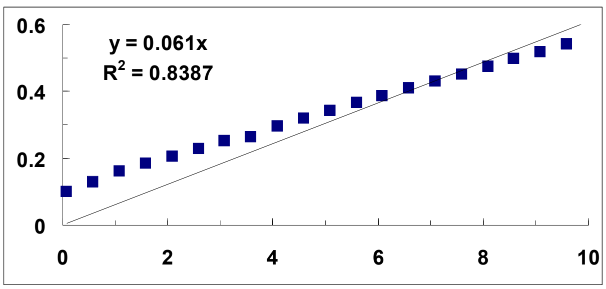
Therefore, $y = -0.07153x + 3.0963$. Excel spreadsheet solution:



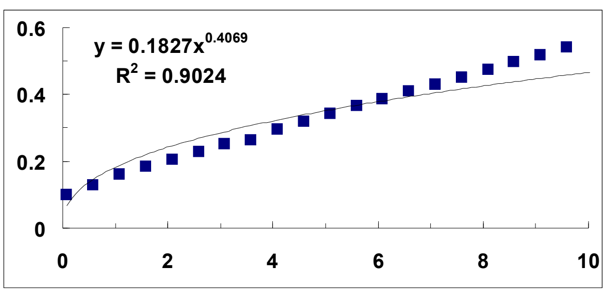
17.21 This problem was solved using an Excel spreadsheet and TrendLine. Linear regression gives




Forcing a zero intercept yields

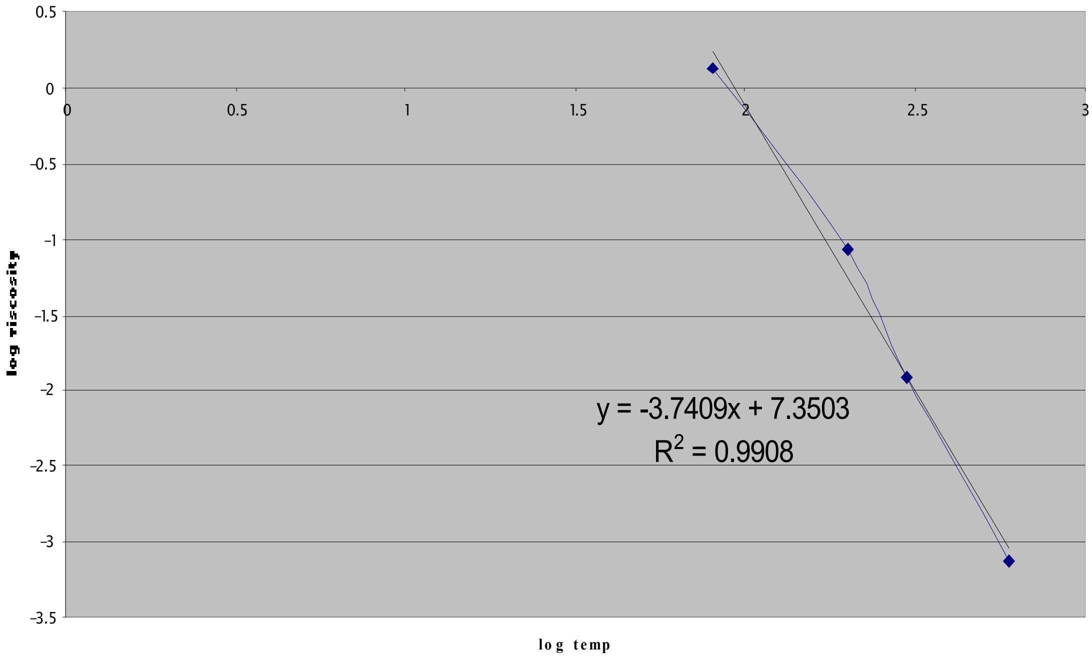



One alternative that would force a zero intercept is a power fit




However, this seems to represent a poor compromise since it misses the linear trend in the data. An alternative approach would to assume that the physically-unrealistic non-zero intercept is an artifact of the measurement method. Therefore, if the linear slope is valid, we might try $y = 0.0454x$.

17.22 This problem was solved using an Excel spreadsheet.

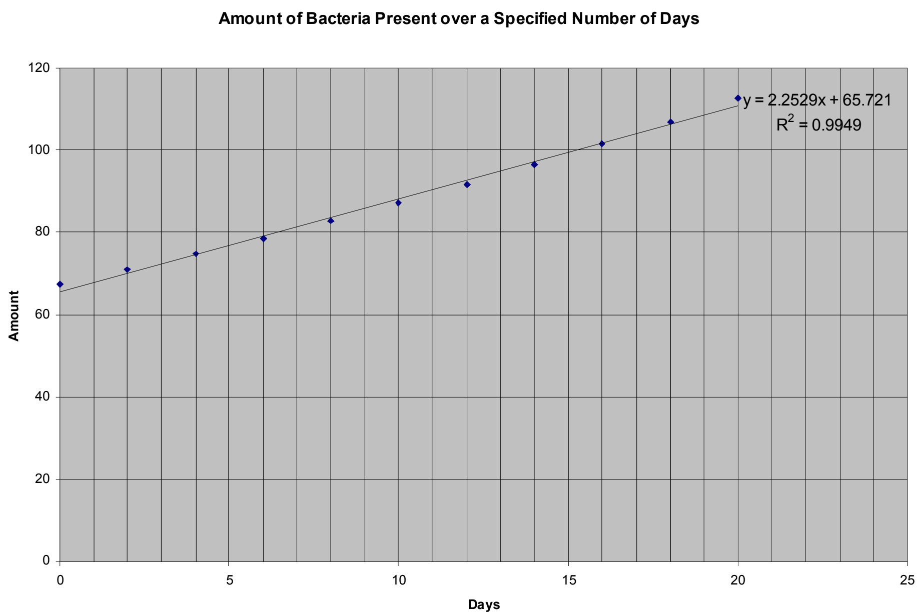


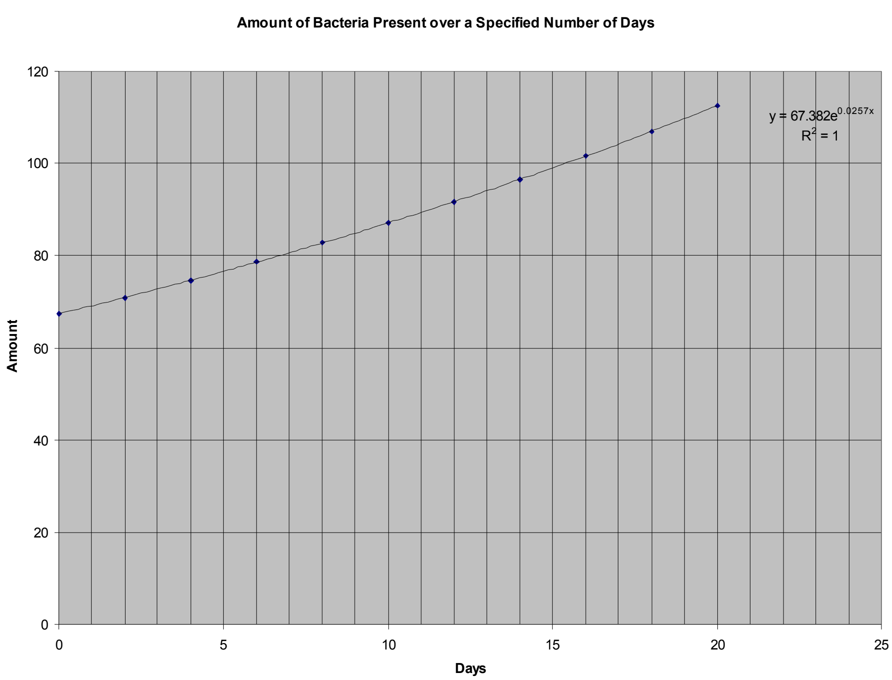

17.23 Using Excel, plot a linear fit which results in $R^2 = 0.9949$. Using an exponential fit results in $R^2 = 1$, which implies a perfect fit. Therefore, use the exponential solution.

The amount of bacteria after 30 days:

$$y = 67.382e^{0.0257x}$$

$$y(30) = 145.67 \times 10^6$$


Amount of Bacteria Present over a Specified Number of Days
y = 2.2529x + 65.721
R2 = 0.9949
Amount
Days
120
100
80
60
40
20
0
0
5
10
15
20
25


Amount of Bacteria Present over a Specified Number of Days
y = 67.382e0.0257x
R2 = 1
Amount
Days
120
100
80
60
40
20
0
0
5
10
15
20
25

# Chapter 18

18.1 a)

$\log(4) = 0.60206$
$\log(4.5) = 0.6532125$
$\log(5) = 0.69897$ (true)
$\log(5.5) = 0.7403627$
$\log(6) = 0.7781513$

$$f_1(5) = 0.60206 + \frac{0.7781513 - 0.60206}{6-4}(5-4)$$

$$f_1(5) = 0.690106$$

$$\epsilon_t = \frac{0.69897 - 0.690106}{0.69897} \times 100$$
$$= 1.27\%$$

b) $$f_1(5) = 0.6532125 + \frac{0.7403627 - 0.6532125}{5.5-4.5}(5-4.5)$$

$$f_1(5) = 0.696788$$

$$\epsilon_t = \frac{0.69897 - 0.696788}{0.69897} \times 100$$

$= 0.3\%$ smaller than part (a) as expected

18.2 add second order term to results from Prob 18.1 a)

$$b_2 = \frac{\frac{f(5.5) - f(6)}{5.5-6} - \frac{f(6) - f(4)}{6-4}}{5.5 - 4}$$

$$f_2(5) = 0.690106 - 0.008312\,(5-4)(5-6)$$
$$= 0.698419 \qquad (\epsilon_t = 0.08\%)$$

18.3

$$b_3 = f[x_3, x_2, x_1, x_0]$$

$$b_3 = \frac{f[x_3,x_2,x_1] - f[x_2,x_1,x_0]}{x_3 - x_0}$$

use $x_0 = 4$
$x_1 = 6$
$x_2 = 5.5$
$x_3 = 4.5$

$f[x_2, x_1, x_0] =$

$-0.008312$ from prob 18.2

$f[x_3, x_2, x_1] =$

$$\frac{\frac{f(x_3) - f(x_2)}{x_3 - x_2} - \frac{f(x_2) - f(x_1)}{x_2 - x_1}}{x_3 - x_1}$$

$= -0.0077153$

$$b_3 = \frac{-0.0077153 - -0.008312}{4.5 - 4}$$

$= 0.0011934$

$f_3(5) = 0.698419 +$

$0.0011934\,(5-4)(5-6)(5-5.5)$

$= 0.6990157 \quad (\epsilon_t = 0.006\%)$

18.4 select best points

a)

$x_0 = 3 \qquad f(x_0) = 8$

$x_1 = 4 \qquad f(x_1) = 2$

$x_2 = 2.5 \qquad f(x_2) = 7$

$x_3 = 5 \qquad f(x_3) = 1$

$x_3 = 2$ is also acceptable

$f[x_1, x_0] = -6$

$f[x_2, x_1] = -3.333$

$f[x_3, x_2] = -2.4$

$f[x_2, x_1, x_0] = -5.333$

$f[x_3, x_2, x_1] = 0.9333$

$f[x_3, x_2, x_1, x_0] = 3.1333$

$$f_1(3.4) = 8 + (-6)(3.4-3) = 5.6$$

$$f_2(3.4) = 5.6 + (-5.33)(3.4-3)(3.4-4) = 6.8799$$

$$f_3(3.4) = 6.8799 + 3.1333(3.4-3)(3.4-4)(3.4-2.5) = 6.20311$$

b)

$R_1 = -5.33(0.4)(-0.6) = 1.2792$

$R_2 = 3.1333(0.4)(-0.6)(0.9) = 0.67673$

$$R_3 = f[x_4, x_3, x_2, x_1, x_0](3.4-3)(3.4-4)(3.4-2.5)(3.4-5)$$

let $x_4 = 2 \qquad f(x_4) = 5$

$$f[x_4, x_3] = \frac{5-1}{2-5} = -1.333$$

$$f[x_4, x_3, x_2] = \frac{-1.333 - -2.4}{2-2.5} = -2.134$$

$$f[x_4, x_3, x_2, x_1] = \frac{-2.134 - 0.9333}{2-4} = 1.5336$$

$$f[x_4, x_3, x_2, x_1, x_0] = \frac{1.5336 - 3.1333}{2-3} = 1.5997$$

$$R_3 = 1.5997(3.4-3)(3.4-4)(3.4-2.5)(3.4-5) = 0.553$$

18.5

$x_0 = 3 \qquad f[x_1, x_0] = 7.25$

$x_1 = 5 \qquad f[x_2, x_1] = 5.25$

$x_2 = 2 \qquad f[x_3, x_2] = 8$

$x_3 = 6 \qquad f[x_4, x_3] = 6.25$

$x_4 = 1$

$f[x_2, x_1, x_0] = 2$

$f[x_3, x_2, x_1] = 2.75$

$f[x_4, x_3, x_2] = 1.75$

$f[x_3, x_2, x_1, x_0] = 0.25$

$f[x_4, x_3, x_2, x_1] = 0.25$

$f[x_4, x_3, x_2, x_1, x_0] = 0$

∴ data generated with cubic equation

$$f_1(4) = 5.25 + 7.25(4-3) = 12.5$$

$$f_2(4) = 12.5 + 2(4-3)(4-5) = 10.5$$

$$f_3(4) = 10.5 + 0.25(4-3)(4-5)(4-2) = 10.0$$

$$f_4(4) = 10.0$$

18.6

$$f_1(5) = \left(\frac{5-6}{4-6}\right)0.60206 + \left(\frac{5-4}{6-4}\right)0.7781513$$

$$= 0.6901057$$

$$f_2(5) = \frac{(5-6)(5-5.5)}{(4-6)(4-5.5)}0.60206 + \frac{(5-4)(5-5.5)}{(6-4)(6-5.5)}0.7781513$$

$$+ \frac{(5-4)(5-6)}{(5.5-4)(5.5-6)}0.7403627$$

$$= 0.698418$$

$$f_3(5) = \frac{(5-6)(5-5.5)(5-4.5)}{(4-6)(4-5.5)(4-4.5)}0.60206 = -0.1003433$$

$$+ \frac{(5-4)(5-5.5)(5-4.5)}{(6-4)(6-5.5)(6-4.5)}0.7781513 = -0.129692$$

$$+ \frac{(5-4)(5-6)(5-4.5)}{(5.5-4)(5.5-6)(5.5-4.5)}0.7403627 = +0.493575$$

$$+ \frac{(5-4)(5-6)(5-5.5)}{(4.5-4)(4.5-6)(4.5-5.5)}0.6532125 = +0.435475$$

$$= 0.6990147$$

18.7

$$f_1(4) = \frac{(4-5)}{(3-5)}5.25 + \frac{(4-3)}{(5-3)}19.75 = 12.5$$

$$f_2(4) = \frac{(4-5)(4-2)}{(3-5)(3-2)}5.25 = 5.25$$

$$+ \frac{(4-3)(4-2)}{(5-3)(5-2)}19.75 = 6.58333$$

$$+ \frac{(4-3)(4-5)}{(2-3)(2-5)}4 = -1.3333$$

$$= 10.5000$$

$$f_3(4) = \frac{(4-5)(4-2)(4-6)}{(3-5)(3-2)(3-6)} 5.25 = 3.5$$

$$\frac{(4-3)(4-2)(4-6)}{(5-3)(5-2)(5-6)} 19.75 = 13.1666$$

$$\frac{(4-3)(4-5)(4-6)}{(2-3)(2-5)(2-6)} 4 = -0.6666$$

$$\frac{(4-3)(4-5)(4-2)}{(6-3)(6-5)(6-2)} 36 = -6.0$$

$$+ \quad 10.00$$

solving quadratic for $x$ when $f(x) = 0.93$ gives

$$x = \frac{0.24706 \pm \sqrt{(0.24706)^2 - 4(0.29412)}}{2(0.029412)}$$

$(0.5064717)$

gives $x = 3.5519$

18.8 use $x_0 = 2$ $x_1 = 3$ $x_2 = 4$ $x_3 = 5$

$$\begin{bmatrix} 1 & 2 & 4 & 8 \\ 1 & 3 & 9 & 27 \\ 1 & 4 & 16 & 64 \\ 1 & 5 & 25 & 125 \end{bmatrix} \begin{Bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{Bmatrix} = \begin{Bmatrix} 0.5 \\ 0.3333 \\ 0.25 \\ 0.20 \end{Bmatrix}$$

Solve using Gauss Elimination

$$f(x) = 1.284 - 0.5923x + 0.11685x^2 - 0.00835x^3$$

Now use Toolkit Bisection to solve

$$0 = -0.3 + 1.284 - 0.5923x + 0.11685x^2 - 0.00835x^3$$

gives $x = 3.319672$

18.9 a) $\frac{x^2}{1+x^2} = 0.93$ $\quad x = \sqrt{\frac{0.93}{0.07}} = 3.6449574$

b) use $x_0 = 2$ $x_1 = 3$ $x_2 = 4$

$$\begin{bmatrix} 1 & 2 & 4 \\ 1 & 3 & 9 \\ 1 & 4 & 16 \end{bmatrix} \begin{Bmatrix} a_0 \\ a_1 \\ a_2 \end{Bmatrix} = \begin{Bmatrix} 0.8 \\ 0.9 \\ 0.941176 \end{Bmatrix}$$

solving gives

$$f_2(x) = 0.4235283 + 0.24706x - 0.029412x^2$$

c)

$$\begin{bmatrix} 1 & 2 & 4 & 8 \\ 1 & 3 & 9 & 27 \\ 1 & 4 & 16 & 64 \\ 1 & 5 & 25 & 125 \end{bmatrix} \begin{Bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{Bmatrix} = \begin{Bmatrix} 0.8 \\ 0.9 \\ 0.941176 \\ 0.961538 \end{Bmatrix}$$

solving gives

$a_0 = 0.271488$
$a_1 = 0.41177$
$a_2 = -0.086427$
$a_3 = 0.006335$

$$0 = -0.93 + 0.271488 + 0.41177x - 0.086427x^2 + 0.006335x^3$$

Solving using Bisection from Toolkit gives

$x = 3.618866$

Note as order of interpolation increases accuracy and agreement with analytical solution improves

18.10

| $i$ | $x$ | $f(x)$ |
|---|---|---|
| 0 | 1 | 1 |
| 1 | 2 | 5 |
| 2 | 2.5 | 7 |
| 3 | 3 | 8 |
| 4 | 4 | 2 |

<u>match interior points</u>

$4a_1 + 2b_1 + c_1 = 5$
$4a_2 + 2b_2 + c_2 = 5$
$6.25a_2 + 2.5b_2 + c_2 = 7$
$6.25a_3 + 2.5b_3 + c_3 = 7$
$9a_3 + 3b_3 + c_3 = 8$
$9a_4 + 3b_4 + c_4 = 8$

<u>End Points</u>

$a_1 + b_1 + c_1 = 1$
$16a_4 + 4b_4 + c_4 = 2$

<u>Match slopes</u>

$4a_1 + b_1 = 4a_2 + b_2$
$5a_2 + b_2 = 5a_3 + b_3$
$6a_3 + b_3 = 6a_4 + b_4$

<u>assume</u> $a_1 = 0$

$$\begin{bmatrix} 2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 4 & 2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 6.25 & 2.5 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 6.25 & 2.5 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 9 & 3 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 9 & 3 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 16 & 4 & 1 \\ 1 & 0 & -4 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 5 & 1 & 0 & -5 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 6 & 1 & 0 & -6 & -1 & 0 \end{bmatrix} \begin{bmatrix} b_1 \\ c_1 \\ a_2 \\ b_2 \\ c_2 \\ a_3 \\ b_3 \\ c_3 \\ a_4 \\ b_4 \\ c_4 \end{bmatrix} = \begin{bmatrix} 5 \\ 5 \\ 7 \\ 7 \\ 8 \\ 8 \\ 1 \\ 2 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

which can be solved

$$\underbrace{\begin{matrix} a_1 = 0 & a_2 = 0 \\ b_1 = 4 & b_2 = 4 \\ c_1 = -3 & c_2 = -3 \end{matrix}}_{} \quad \begin{matrix} a_3 = -4 & a_4 = -6 \\ b_3 = 24 & b_4 = 36 \\ c_3 = -28 & c_4 = -46 \end{matrix}$$

first 3 points form straight line

$$f_3(3.4) = -6(3.4)^2 + 36(3.4) - 46 = 7.04$$

$$f_3(2.5) = -4(2.2)^2 + 24(2.2) - 28 = 5.44$$

18.11 a)

| | |
|---|---|
| $x_0 = 1$ | $f(x_0) = 4.75$ |
| $x_1 = 2$ | $f(x_1) = 4$ |
| $x_2 = 3$ | $f(x_2) = 5.25$ |
| $x_3 = 5$ | $f(x_3) = 19.75$ |
| $x_4 = 6$ | $f(x_4) = 36$ |

Use Eq. 18.37, for $i=1$

$$(2-1)f''(1) + 2(3-1)f''(2) + (3-2)f''(3) =$$

$$\frac{6}{3-2}\left[5.25 - 4\right] + \frac{6}{2-1}\left[4.75 - 4\right]$$

or $4f''(2) + f''(3) = 12$ since $f''(1) = 0$

for $i=2$

$$(3-2)f''(2) + 2(5-2)f''(3) + (5-3)f''(4) =$$

$$\frac{6}{5-3}\left[19.75 - 5.25\right] + \frac{6}{3-2}\left[4 - 5.25\right] \quad \text{or,}$$

$$f''(2) + 6f''(3) + 2f''(4) = 36$$

for $i=3$

$$(5-3)f''(3) + 2(6-3)f''(5) + (6-5)f''(6) =$$

$$\frac{6}{6-5}\left[36 - 19.75\right] + \frac{6}{5-3}\left[5.25 - 19.75\right] \quad \text{or,}$$

$$2f''(3) + 6f''(5) + f''(6) = 54$$

or, because $f''(6) = 0$

$$2f''(3) + 6f''(5) = 54$$

Solve $\begin{bmatrix} 4 & 1 & 0 \end{bmatrix} \begin{bmatrix} f''(2) \end{bmatrix} \quad \begin{bmatrix} 12 \end{bmatrix}$

4th interval $f_4(x) = \frac{8.0164}{6(6-5)}(6-x)^3$

$$+ \left[\frac{19.75}{6-5} - \frac{8.0164(6-5)}{6}\right](6-x)$$

$$+ \frac{36}{\phantom{0}}(x-5) \quad \text{or,}$$

18.12

| $x$ | $f(x)$ |
|---|---|
| 3 | 8 |
| 4 | 2 |
| 5 | 1 |

$$a_0 + 3a_1 + 9a_2 = 8$$
$$a_0 + 4a_1 + 16a_2 = 2$$
$$a_0 + 5a_1 + 25a_2 = 1$$

solving $a_0 = 56$
$a_1 = -23.5$
$a_2 = 2.5$

18.13

| $x$ | $f(x)$ |
|---|---|
| 1 | 4.75 |
| 2 | 4 |
| 3 | 5.25 |
| 5 | 19.75 |

$$a_0 + a_1 + a_2 + a_3 = 4.75$$
$$a_0 + 2a_1 + 4a_2 + 8a_3 = 4$$
$$a_0 + 3a_1 + 9a_2 + 27a_3 = 5.25$$
$$a_0 + 5a_1 + 25a_2 + 125a_3 = 19.75$$

solving $a_0 = 6$ $a_2 = -0.5$
$a_1 = -1$ $a_3 = 0.25$

18.14 Here is a VBA program to implement Newton interpolation. It is set up to solve Example 18.5:

```
Option Explicit

Sub Newt()

Dim n As Integer, i As Integer
Dim yint(10) As Single, x(10) As Single, y(10) As Single
Dim ea(10) As Single, xi As Single

Range("a5").Select
n = ActiveCell.Row
Selection.End(xlDown).Select
n = ActiveCell.Row - n
Range("a5").Select
For i = 0 To n
  x(i) = ActiveCell.Value
  ActiveCell.Offset(0, 1).Select
  y(i) = ActiveCell.Value
  ActiveCell.Offset(1, -1).Select
Next i
Range("e3").Select
xi = ActiveCell.Value
```

```
Call Newtint(x(), y(), n, xi, yint, ea)

Sheets("Sheet1").Select
Range("d5:f25").ClearContents
Range("d5").Select
For i = 0 To n
  ActiveCell.Value = i
  ActiveCell.Offset(0, 1).Select
  ActiveCell.Value = yint(i)
  ActiveCell.Offset(0, 1).Select
  ActiveCell.Value = ea(i)
  ActiveCell.Offset(1, -2).Select
Next i
Range("a5").Select

End Sub

Sub Newtint(x, y, n, xi, yint, ea)

Dim i As Integer, j As Integer, order As Integer
Dim fdd(10, 10) As Single, xterm As Single
Dim yint2 As Single

For i = 0 To n
  fdd(i, 0) = y(i)
Next i
For j = 1 To n
  For i = 0 To n - j
    fdd(i, j) = (fdd(i + 1, j - 1) - fdd(i, j - 1)) / (x(i + j) - x(i))
  Next i
Next j
xterm = 1#
yint(0) = fdd(0, 0)
For order = 1 To n
  xterm = xterm * (xi - x(order - 1))
  yint2 = yint(order - 1) + fdd(0, order) * xterm
  ea(order - 1) = yint2 - yint(order - 1)
  yint(order) = yint2
Next order

End Sub
```

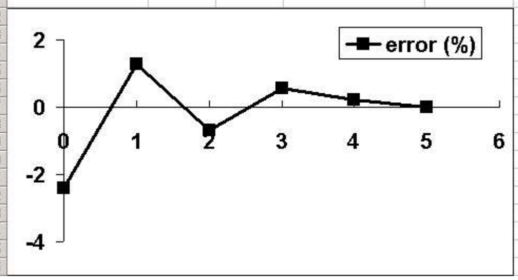
18.15 Here is the solution when the program from Prob. 18.14 is run.

| | A | B | C | D | E | F | G | H | I |
|---|---|---|---|---|---|---|---|---|---|
| 1 | Problem 18.15 | | | | | | | | |
| 2 | | | | | | | | | |
| 3 | Data: | | | Interpolation at x = | 2 | | | | |
| 4 | x | y | | order | f(x) | error (%) | | | |
| 5 | 1 | 0 | | 0 | 0 | 0.4620981 | | | |
| 6 | 4 | 1.386294 | | 1 | 0.462098122 | 0.1037462 | | RUN | |
| 7 | 6 | 1.791759 | | 2 | 0.565844357 | 0.0629242 | | | |
| 8 | 5 | 1.609438 | | 3 | 0.628768563 | 0.0469534 | | | |
| 9 | 3 | 1.098612 | | 4 | 0.675721943 | 0.0217922 | | | |
| 10 | 1.5 | 0.405465 | | 5 | 0.697514176 | -0.003616 | | | |
| 11 | 2.5 | 0.916291 | | 6 | 0.693897784 | -0.000459 | | | |
| 12 | 3.5 | 1.252763 | | 7 | 0.693438709 | 0 | | | |
| 13 | | | | | | | | | |

18.16 See solutions for Probs. 18.1 through 18.3.

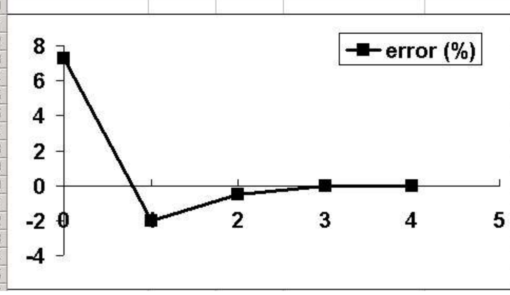
18.17 A plot of the error can easily be added to the Excel application. The following shows the solution for Prob. 18.4:

| | A | B | C | D | E | F | G | H |
|---|---|---|---|---|---|---|---|---|
| 1 | Problem 18.4 | | | | | | | |
| 2 | | | | | | | | |
| 3 | Data: | | | Interpolation at x = | 3.4 | | | |
| 4 | x | y | | order | f(x) | error (%) | | |
| 5 | 3 | 8 | | 0 | 8 | -2.400001 | | |
| 6 | 4 | 2 | | 1 | 5.599999428 | 1.2800002 | | RUN |
| 7 | 2.5 | 7 | | 2 | 6.879999638 | -0.6768 | | |
| 8 | 5 | 1 | | 3 | 6.203199387 | 0.5529599 | | |
| 9 | 2 | 5 | | 4 | 6.756159306 | 0.2204161 | | |
| 10 | 1 | 1 | | 5 | 6.976575375 | 0 | | |




The following shows the solution for Prob. 18.5:

| | A | B | C | D | E | F | G | H |
|---|---|---|---|---|---|---|---|---|
| 1 | Problem 18.5 | | | | | | | |
| 2 | | | | | | | | |
| 3 | Data: | | | Interpolation at x = | 4 | | | |
| 4 | x | y | | order | f(x) | error (%) | | |
| 5 | 3 | 5.25 | | 0 | 5.25 | 7.25 | | |
| 6 | 5 | 19.75 | | 1 | 12.5 | -2 | | RUN |
| 7 | 2 | 4 | | 2 | 10.5 | -0.5 | | |
| 8 | 6 | 36 | | 3 | 10 | 0 | | |
| 9 | 1 | 4.75 | | 4 | 10 | 0 | | |




18.18

```
Option Explicit

Sub LagrInt()

Dim n As Integer, i As Integer, order As Integer
Dim x(10) As Single, y(10) As Single, xi As Single

Range("a5").Select
n = ActiveCell.Row
Selection.End(xlDown).Select
n = ActiveCell.Row - n
Range("a5").Select
For i = 0 To n
  x(i) = ActiveCell.Value
  ActiveCell.Offset(0, 1).Select
  y(i) = ActiveCell.Value
  ActiveCell.Offset(1, -1).Select
Next i
Range("e3").Select
order = ActiveCell.Value
ActiveCell.Offset(1, 0).Select
xi = ActiveCell.Value
```

```
ActiveCell.Offset(2, 0).Select
ActiveCell.Value = Lagrange(x(), y(), order, xi)

End Sub

Function Lagrange(x, y, order, xi)

Dim i As Integer, j As Integer
Dim sum As Single, prod As Single

sum = 0#
For i = 0 To order
  prod = y(i)
  For j = 0 To order
    If i <> j Then
      prod = prod * (xi - x(j)) / (x(i) - x(j))
    End If
  Next j
  sum = sum + prod
Next i
Lagrange = sum

End Function
```

Application to Example 18.7:

| | A | B | C | D | E | F | G | H |
|---|---|---|---|---|---|---|---|---|
| 1 | Example 18.7 | | | | | | | |
| 2 | | | | | | | | |
| 3 | Data: | | | Order = | 2 | | RUN | |
| 4 | x | y | | Interpolation at x = | 10 | | | |
| 5 | 13 | 4755 | | | | | | |
| 6 | 7 | 3940 | | f(x) = | 4672.813 | | | |
| 7 | 5 | 3090 | | | | | | |
| 8 | 3 | 2310 | | | | | | |
| 9 | 1 | 800 | | | | | | |
| 10 | | | | | | | | |

18.19 The following VBA program uses cubic interpolation for all intervals:

```
Option Explicit

Sub Newt()

Dim n As Integer, i As Integer
Dim yint(10) As Single, x(10) As Single, y(10) As Single
Dim ea(10) As Single, xi As Single

Range("a5").Select
n = ActiveCell.Row
Selection.End(xlDown).Select
n = ActiveCell.Row - n
Range("a5").Select
For i = 0 To n
  x(i) = ActiveCell.Value
  ActiveCell.Offset(0, 1).Select
  y(i) = ActiveCell.Value
  ActiveCell.Offset(1, -1).Select
Next i
Range("e4").Select
xi = ActiveCell.Value

ActiveCell.Offset(2, 0).Select
ActiveCell.Value = Interp(x(), y(), n, xi)
Range("a5").Select
```

```
End Sub

Function Interp(x, y, n, xx)

Dim ii As Integer

If xx < x(0) Or xx > x(n) Then
  Interp = "out of range"
Else
  If xx <= x(ii + 1) Then
    Interp = Lagrange(x, y, 0, 3, xx)
  ElseIf xx <= x(n - 1) Then
  For ii = 0 To n - 2
    If xx >= x(ii) And xx <= x(ii + 1) Then
      Interp = Lagrange(x, y, ii - 1, 3, xx)
      Exit For
    End If
  Next ii
  Else
    Interp = Lagrange(x, y, n - 3, 3, xx)
  End If
End If
End Function


Function Lagrange(x, y, i0, order, xi)

Dim i As Integer, j As Integer
Dim sum As Single, prod As Single

sum = 0#
For i = i0 To i0 + order
  prod = y(i)
  For j = i0 To i0 + order
    If i <> j Then
      prod = prod * (xi - x(j)) / (x(i) - x(j))
    End If
  Next j
  sum = sum + prod
Next i
Lagrange = sum

End Function
```

Application to evaluate ln(2.5):

| | A | B | C | D | E | F | G | H | I |
|---|---|---|---|---|---|---|---|---|---|
| 1 | Problem 18.4 | | | | | | | | |
| 2 | | | | | | | | | |
| 3 | Data: | | | | | | | | |
| 4 | x | y | | Interpolation at x = | 2.5 | | | | |
| 5 | 1 | 0 | | | | | | | |
| 6 | 2 | 0.693147 | | f(x) = | 0.921221316 | | | RUN | |
| 7 | 3 | 1.098612 | | | | | | | |
| 8 | 4 | 1.386294 | | True value | 0.916290732 | | | | |
| 9 | 5 | 1.609438 | | | | | | | |
| 10 | 6 | 1.791759 | | Error | 0.53810 | % | | | |
| 11 | 7 | 1.94591 | | | | | | | |
| 12 | 8 | 2.079442 | | | | | | | |
| 13 | 9 | 2.197225 | | | | | | | |
| 14 | 10 | 2.302585 | | | | | | | |
| 15 | | | | | | | | | |

18.20

```
Sub Splines()
Dim i As Integer, n As Integer
```

```
Dim x(7) As Single, y(7) As Single, xu As Single, yu As Single
Dim dy As Single, d2y As Single

Range("a5").Select
n = ActiveCell.Row
Selection.End(xlDown).Select
n = ActiveCell.Row - n
Range("a5").Select
For i = 0 To n
  x(i) = ActiveCell.Value
  ActiveCell.Offset(0, 1).Select
  y(i) = ActiveCell.Value
  ActiveCell.Offset(1, -1).Select
Next i
Range("e4").Select
xu = ActiveCell.Value

Call Spline(x(), y(), n, xu, yu, dy, d2y)

ActiveCell.Offset(2, 0).Select
ActiveCell.Value = yu

End Sub

Sub Spline(x, y, n, xu, yu, dy, d2y)
Dim e(10) As Single, f(10) As Single, g(10) As Single, r(10) As Single,
d2x(10) As Single

Call Tridiag(x, y, n, e, f, g, r)
Call Decomp(e(), f(), g(), n - 1)
Call Substit(e(), f(), g(), r(), n - 1, d2x())
Call Interpol(x, y, n, d2x(), xu, yu, dy, d2y)

End Sub

Sub Tridiag(x, y, n, e, f, g, r)
Dim i As Integer

f(1) = 2 * (x(2) - x(0))
g(1) = x(2) - x(1)
r(1) = 6 / (x(2) - x(1)) * (y(2) - y(1))
r(1) = r(1) + 6 / (x(1) - x(0)) * (y(0) - y(1))
For i = 2 To n - 2
  e(i) = x(i) - x(i - 1)
  f(i) = 2 * (x(i + 1) - x(i - 1))
  g(i) = x(i + 1) - x(i)
  r(i) = 6 / (x(i + 1) - x(i)) * (y(i + 1) - y(i))
  r(i) = r(i) + 6 / (x(i) - x(i - 1)) * (y(i - 1) - y(i))
Next i
e(n - 1) = x(n - 1) - x(n - 2)
f(n - 1) = 2 * (x(n) - x(n - 2))
r(n - 1) = 6 / (x(n) - x(n - 1)) * (y(n) - y(n - 1))
r(n - 1) = r(n - 1) + 6 / (x(n - 1) - x(n - 2)) * (y(n - 2) - y(n - 1))

End Sub

Sub Interpol(x, y, n, d2x, xu, yu, dy, d2y)
Dim i As Integer, flag As Integer
Dim c1 As Single, c2 As Single, c3 As Single, c4 As Single
Dim t1 As Single, t2 As Single, t3 As Single, t4 As Single

flag = 0
i = 1
Do
  If xu >= x(i - 1) And xu <= x(i) Then
    c1 = d2x(i - 1) / 6 / (x(i) - x(i - 1))
    c2 = d2x(i) / 6 / (x(i) - x(i - 1))
```

```
    c3 = y(i - 1) / (x(i) - x(i - 1)) - d2x(i - 1) * (x(i) - x(i - 1)) / 6
    c4 = y(i) / (x(i) - x(i - 1)) - d2x(i) * (x(i) - x(i - 1)) / 6
    t1 = c1 * (x(i) - xu) ^ 3
    t2 = c2 * (xu - x(i - 1)) ^ 3
    t3 = c3 * (x(i) - xu)
    t4 = c4 * (xu - x(i - 1))
    yu = t1 + t2 + t3 + t4
    t1 = -3 * c1 * (x(i) - xu) ^ 2
    t2 = 3 * c2 * (xu - x(i - 1)) ^ 2
    t3 = -c3
    t4 = c4
    dy = t1 + t2 + t3 + t4
    t1 = 6 * c1 * (x(i) - xu)
    t2 = 6 * c2 * (xu - x(i - 1))
    d2y = t1 + t2
    flag = 1
  Else
    i = i + 1
  End If
  If i = n + 1 Or flag = 1 Then Exit Do
Loop
If flag = 0 Then
 MsgBox "outside range"
  End
End If
End Sub

Sub Decomp(e, f, g, n)

Dim k As Integer
For k = 2 To n
  e(k) = e(k) / f(k - 1)
  f(k) = f(k) - e(k) * g(k - 1)
Next k

End Sub

Sub Substit(e, f, g, r, n, x)
Dim k As Integer

For k = 2 To n
  r(k) = r(k) - e(k) * r(k - 1)
Next k
x(n) = r(n) / f(n)
For k = n - 1 To 1 Step -1
  x(k) = (r(k) - g(k) * x(k + 1)) / f(k)
Next k

End Sub
```

| | A | B | C | D | E | F | G | H |
|---|---|---|---|---|---|---|---|---|
| 1 | **Example 18.10** | | | | | | | |
| 2 | | | | | | | | |
| 3 | **Data:** | | | | | | | |
| 4 | **x** | **y** | | **Interpolation at x =** | **5** | | **RUN** | |
| 5 | **3** | **2.5** | | | | | | |
| 6 | **4.5** | **1** | | **f(x) =** | **1.10289** | | | |
| 7 | **7** | **2.5** | | | | | | |
| 8 | **9** | **0.5** | | | | | | |
| 9 | | | | | | | | |

18.21 The following shows the solution for Prob. 18.4:

| | A | B | C | D | E | F | G | H |
|---|---|---|---|---|---|---|---|---|
| 1 | Problem 18.4 | | | | | | | |
| 2 | | | | | | | | |
| 3 | Data: | | | | | | | |
| 4 | x | y | | Interpolation at x = | 2.25 | | RUN | |
| 5 | 1 | 1 | | | | | | |
| 6 | 2 | 5 | | f(x) = | 6.013615 | | | |
| 7 | 2.5 | 7 | | | | | | |
| 8 | 3 | 8 | | | | | | |
| 9 | 4 | 2 | | | | | | |
| 10 | 5 | 1 | | | | | | |
| 11 | | | | | | | | |

The following shows the solution for Prob. 18.5:

| | A | B | C | D | E | F | G | H |
|---|---|---|---|---|---|---|---|---|
| 1 | Problem 18.5 | | | | | | | |
| 2 | | | | | | | | |
| 3 | Data: | | | | | | | |
| 4 | x | y | | Interpolation at x = | 2.25 | | RUN | |
| 5 | 1 | 4.75 | | | | | | |
| 6 | 2 | 3 | | f(x) = | 3.204406 | | | |
| 7 | 3 | 5.25 | | | | | | |
| 8 | 5 | 19.75 | | | | | | |
| 9 | 6 | 36 | | | | | | |
| 10 | | | | | | | | |

18.22

$$f_1(x) = f(x_o) + \frac{f(x_1) - f(x_o)}{x_1 - x_o}(x - x_o)$$

$$f_1(x) = 6.5453 + \left(\frac{6.7664 - 6.5453}{0.12547 - 0.11144}\right)(x - 0.11144)$$

$$f_1(x) = 4.789107 + 15.579x$$

$$x = 0.118, f_1(x) = 6.6487$$

$$s = 6.6487 \ \frac{kJ}{kg \circ K}$$

# CHAPTER 19

19.1 The normal equations can be derived as

$$\begin{bmatrix} 11 & 2.416183 & 2.018098 \\ 2.416183 & 6.004565 & 0.017037 \\ 2.018098 & 0.017037 & 4.995435 \end{bmatrix} \begin{Bmatrix} A_0 \\ A_1 \\ B_1 \end{Bmatrix} = \begin{Bmatrix} 83.9 \\ 15.43934 \\ 10.81054 \end{Bmatrix}$$

which can be solved for

$A_0 =$ 7.957538
$A_1 =$ -0.6278
$B_1 =$ -1.04853

The mean is 7.958 and the amplitude and the phase shift can be computed as

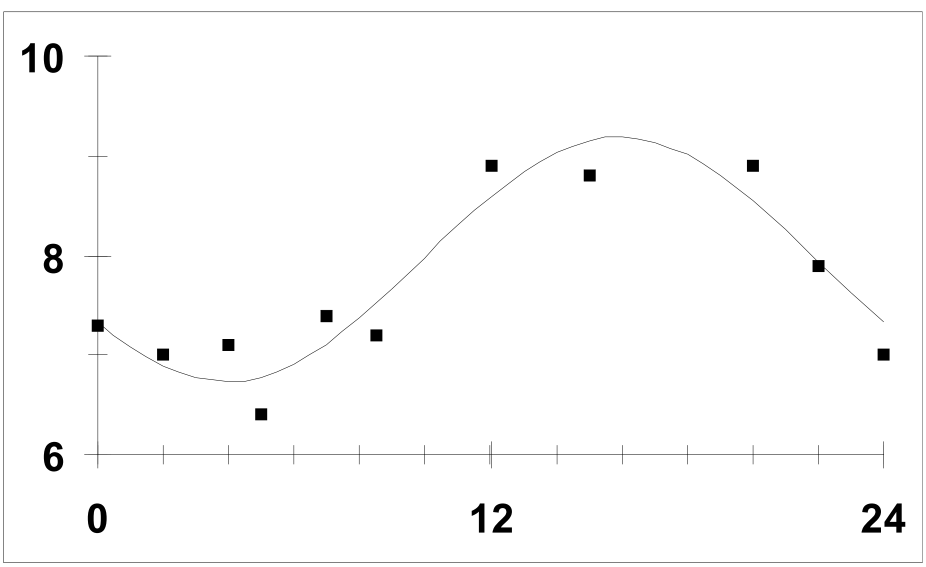
$$C_1 = \sqrt{(-0.6278)^2 + (-1.04853)^2} = 1.222$$

$$\theta = \tan^{-1}\left(\frac{-1.04853}{-0.6278}\right) + \pi = 2.11 \text{ radians} \times \frac{12 \text{ hrs}}{\pi} = 8.06 \text{ hr}$$

Thus, the final model is

$$f(t) = 7.958 + 1.222 \cos\left(\frac{2\pi}{24}(t + 8.06)\right)$$

The data and the fit are displayed below:




Note that the peak occurs at 24 − 8.06 = 15.96 hrs.

19.2 The normal equations can be derived as

$$\begin{bmatrix} 1890 & 127.279 & -568.187 \\ 0 & 5 & -1 \\ 0 & -1 & 5 \end{bmatrix} \begin{Bmatrix} A_0 \\ A_1 \\ B_1 \end{Bmatrix} = \begin{Bmatrix} 350{,}265 \\ -381.864 \\ 156.281 \end{Bmatrix}$$

which can be solved for

$A_0 =$ 195.2491
$A_1 =$ -73.0433
$B_1 =$ 16.64745

The mean is 195.25 and the amplitude and the phase shift can be computed as

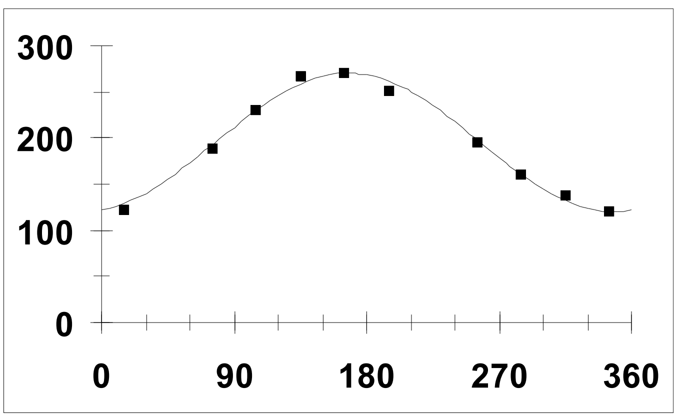
$$C_1 = \sqrt{(-73.0433)^2 + (16.6475)^2} = 74.916$$

$$\theta = \tan^{-1}\left(\frac{16.6475}{--73.0433}\right) + \pi = 3.366 \text{ radians} \times \frac{180 \text{ d}}{\pi} = 192.8 \text{ d}$$

Thus, the final model is

$$f(t) = 195.25 + 74.916 \cos\left(\frac{2\pi}{360}(t + 192.8)\right)$$

The data and the fit are displayed below:




19.3 In the following equations, $\omega_0 = 2\pi/T$

$$\frac{\int_0^T \sin(\omega_0 t)\, dt}{T} = \frac{-\omega_0\left[\cos(\omega_0 t)\right]_0^T}{T} = \frac{-\omega_0}{T}\left(\cos 2\pi - \cos 0\right) = 0$$

$$\frac{\int_0^T \cos(\omega_0 t)\, dt}{T} = \frac{\omega_0\left[\sin(\omega_0 t)\right]_0^T}{T} = \frac{\omega_0}{T}\left(\sin 2\pi - \sin 0\right) = 0$$

$$\frac{\int_0^T \sin^2(\omega_0 t)\,dt}{T} = \frac{\left[\frac{t}{2} - \frac{\sin(2\omega_0 t)}{4\omega_0}\right]_0^T}{T} = \frac{\frac{T}{2} - \frac{\sin 4\pi}{4\omega_0} - 0 + 0}{T} = \frac{1}{2}$$

$$\frac{\int_0^T \cos^2(\omega_0 t)\,dt}{T} = \frac{\left[\frac{t}{2} + \frac{\sin(2\omega_0 t)}{4\omega_0}\right]_0^T}{T} = \frac{\frac{T}{2} + \frac{\sin 4\pi}{4\omega_0} - 0 - 0}{T} = \frac{1}{2}$$

$$\frac{\int_0^T \cos(\omega_0 t)\sin(\omega_0 t)\,dt}{T} = \left[\frac{\sin^2(\omega_0 t)}{2T\omega_0}\right]_0^T = \frac{\sin^2 2\pi}{2\omega_0 T} - 0 = 0$$

19.4 $a_0 = 0$

$$a_k = \frac{2}{T}\int_{-T/2}^{T/2} -t\cos(k\omega_0 t)\;dt$$

$$= -\frac{2}{T}\left[\frac{1}{(k\omega_0)^2}\cos(k\omega_0 t) + \frac{t}{k\omega_0}\sin(k\omega_0 t)\right]_{T/2}^{T/2}$$

$$b_k = \frac{2}{T}\int_{-T/2}^{T/2} -t\sin(k\omega_0 t)\;dt$$

$$= -\frac{2}{T}\left[\frac{1}{(k\omega_0)^2}\sin(k\omega_0 t) - \frac{t}{k\omega_0}\cos(k\omega_0 t)\right]_{T/2}^{T/2}$$

On the basis of these, all $a$'s = 0. For $k$ = odd,

$$b_k = \frac{2}{k\pi}$$

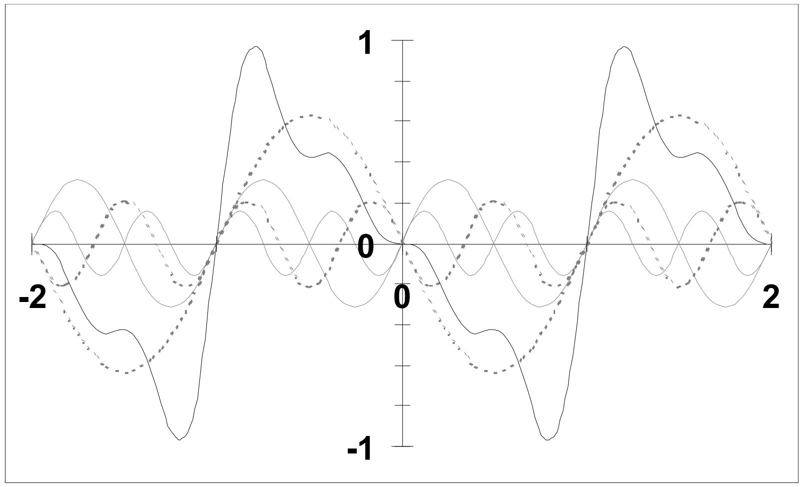
For $k$ = even,

$$b_k = -\frac{2}{k\pi}$$

Therefore, the series is

$$f(t) = -\frac{2}{\pi}\sin(\omega_0 t) + \frac{1}{\pi}\sin(2\omega_0 t) - \frac{2}{3\pi}\sin(3\omega_0 t) + \frac{1}{2\pi}\sin(4\omega_0 t) + \cdots$$

The first 4 terms are plotted below along with the summation:




19.5 $a_0 = 0.5$

$$a_k = \frac{2}{2}\left[\int_{-1}^{0} -t\cos(k\pi t)\;dt + \int_{0}^{1} t\cos(k\pi t)\;dt\right]$$

$$= 1\left\{\left[-\frac{\cos(k\pi t)}{(k\pi)^2} - \frac{t\sin(k\pi t)}{k\pi}\right]_{-1}^{0} + \left[\frac{\cos(k\pi t)}{(k\pi)^2} + \frac{t\sin(k\pi t)}{k\pi}\right]_{0}^{1}\right\}$$

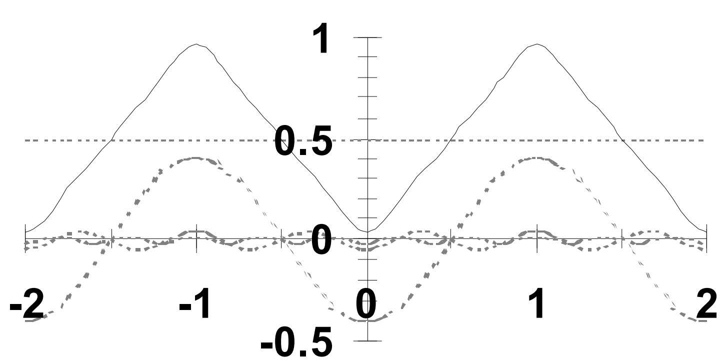
$$= \frac{2}{(k\pi)^2}(\cos k\pi - 1)$$

$b_k = 0$

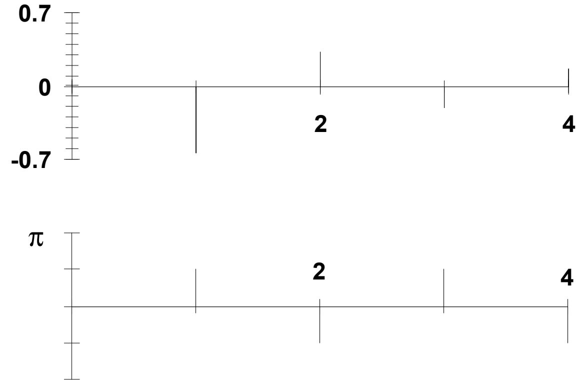
Substituting these coefficients into Eq. (19.17) gives

$$f(t) = \frac{1}{2} - \frac{12}{\pi^2}\cos(\pi t) - \frac{12}{9\pi^2}\cos(3\pi t) - \frac{12}{25\pi^2}\cos(5\pi t) + \cdots$$

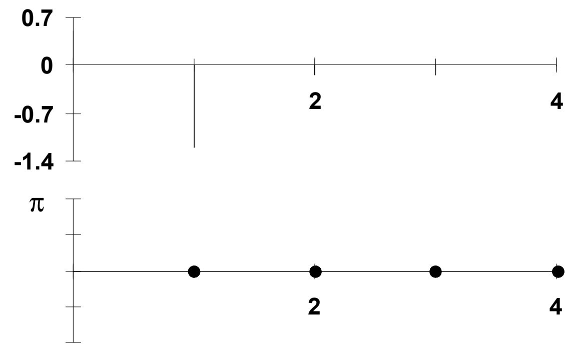
This function for the first 4 terms is displayed below:



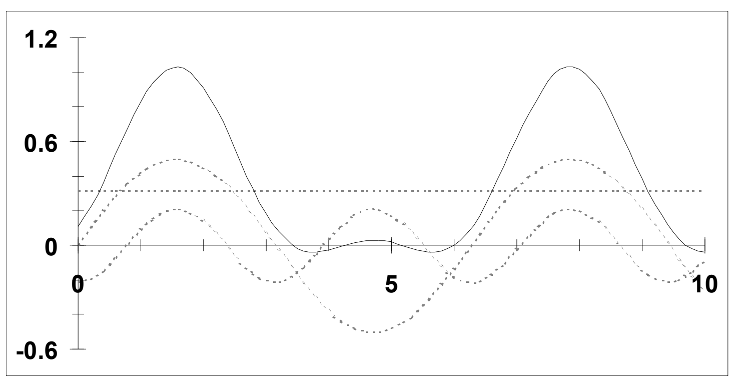
19.6



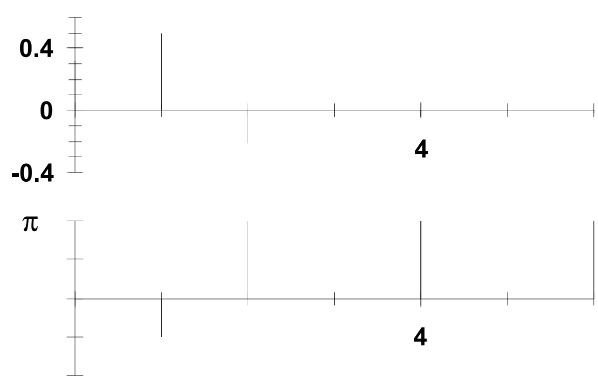

19.7




19.8




19.9



19.10 Here's a Fortran 90 code that implements the DFT. It is set up to solve Prob. 19.11.

```
PROGRAM DFourier
IMPLICIT NONE
INTEGER i,N
REAL f(0:127),re(0:127),im(0:127),omega,pi,t,Tp,dt
pi=4.*atan(1.)
N=32
omega=2.*pi/N
t=0.
Tp=2.*pi
dt=4.*Tp/N
DO i=0,N-1
  f(i)=sin(t)
  if (f(i).LT.0.) f(i)=0.
  t=t+dt
END DO
CALL DFT(f,N,re,im,omega)
OPEN (UNIT=1,FILE='Prob1911.dat',STATUS='unknown')
DO i=0,N-1
  WRITE(1,*) i,f(i),re(i),im(i)
END DO
CLOSE(1)
END

SUBROUTINE DFT(f,N,re,im,omega)
IMPLICIT NONE
INTEGER k,nn,N
REAL f(0:127),re(0:127),im(0:127),angle,omega
DO k=0,N-1
  DO nn=0,N-1
    angle=k*omega*nn
    re(k)=re(k)+f(nn)*cos(angle)/N
    im(k)=im(k)-f(nn)*sin(angle)/N
  END DO
END DO
END
```

19.11 The results for the $n = 32$ case are displayed below:

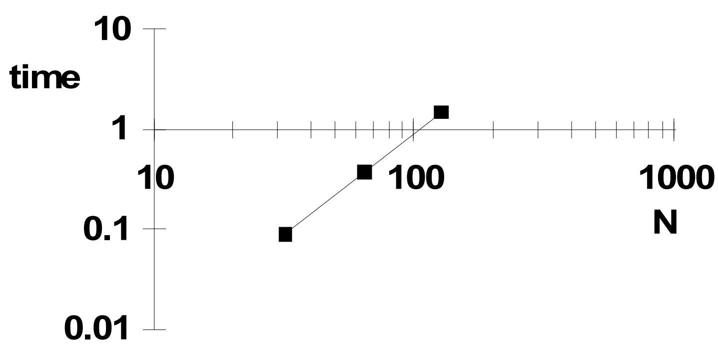
| index | f(t) | real | imaginary |
|---|---|---|---|
| 0 | 0 | 0.3018 | 0 |
| 1 | 0.7071 | 0 | 0 |
| 2 | 1 | 0 | 0 |
| 3 | 0.7071 | 0 | 0 |
| 4 | 0 | 0 | -0.25 |
| 5 | 0 | 0 | 0 |
| 6 | 0 | 0 | 0 |
| 7 | 0 | 0 | 0 |
| 8 | 0 | -0.125 | 0 |
| 9 | 0.7071 | 0 | 0 |
| 10 | 1 | 0 | 0 |
| 11 | 0.7071 | 0 | 0 |
| 12 | 0 | 0 | 0 |
| 13 | 0 | 0 | 0 |
| 14 | 0 | 0 | 0 |
| 15 | 0 | 0 | 0 |
| 16 | 0 | -0.0518 | 0 |
| 17 | 0.7071 | 0 | 0 |
| 18 | 1 | 0 | 0 |
| 19 | 0.7071 | 0 | 0 |
| 20 | 0 | 0 | 0 |

| | | | |
|---|---|---|---|
| 21 | 0 | 0 | 0 |
| 22 | 0 | 0 | 0 |
| 23 | 0 | 0 | 0 |
| 24 | 0 | -0.125 | 0 |
| 25 | 0.7071 | 0 | 0 |
| 26 | 1 | 0 | 0 |
| 27 | 0.7071 | 0 | 0 |
| 28 | 0 | 0 | 0.25 |
| 29 | 0 | 0 | 0 |
| 30 | 0 | 0 | 0 |
| 31 | 0 | 0 | 0 |

The runs for $N = 32$, 64 and 128 were performed with the following results obtained. (Note that even though we used a slow PC, we had to call the function numerous times to obtain measurable times. These times were then divided by the number of function calls to determine the time per call shown below)

| N | time (s) |
|---|---|
| 32 | 0.09 |
| 64 | 0.37 |
| 128 | 1.48 |

A power (log-log) model was fit (see plot below) to this data to yield log(time) = –4.08 + 2.02 log($N$). Thus, the result verifies that the execution time $\propto N^2$.




19.12 Here's a Fortran 90 code that implements the FFT. It is set up to solve Prob. 19.13.

```
PROGRAM FFourier
IMPLICIT NONE
INTEGER i,N
REAL f(0:127),re(0:127),im(0:127),omega,pi,t,Tp,dt
pi=4.*ATAN(1.)
N=32
t=0.
Tp=2.*pi
dt=4.*Tp/N
DO i=0,N-1
  re(i)=sin(t)
  if (re(i).LT.0.) re(i)=0.
  f(i)=re(i)
  t=t+dt
END DO
CALL FFT(N,re,im)
DO i=0,N-1
```

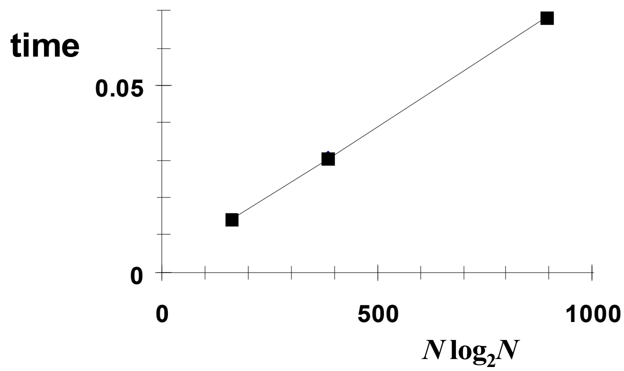
```
  PRINT *, i,f(i),re(i),im(i)
END DO
CLOSE(1)
END

SUBROUTINE FFT (N, x, y)
IMPLICIT NONE
INTEGER :: i,j,N,m,N2,N1,k,l
REAL :: f(0:127),re(0:127),im(0:127),omega,pi,t,Tp,dt,xN,angle
REAL :: arg,c,s,xt,x(0:n),y(0:n),yt
xN=N
m = INT(LOG(xN) / LOG(2.))
pi = 4. * ATAN(1.)
N2 = N
DO k = 1, m
  N1 = N2
  N2 = N2 / 2
  angle = 0.
  arg = 2 * pi / N1
  DO j = 0, N2 - 1
    c = COS(angle)
    s = -SIN(angle)
    DO i = j, N - 1, N1
      l = i + N2
      xt = x(i) - x(l)
      x(i) = x(i) + x(l)
      yt = y(i) - y(l)
      y(i) = y(i) + y(l)
      x(l) = xt * c - yt * s
      y(l) = yt * c + xt * s
    END DO
    angle = (j + 1) * arg
  END DO
END DO
j = 0
DO i = 0, N - 2
  IF (i.LT.j) THEN
    xt = x(j)
    x(j) = x(i)
    x(i) = xt
    yt = y(j)
    y(j) = y(i)
    y(i) = yt
  END IF
  k = N / 2
  DO
    IF (k.GE.j+1) EXIT
    j = j - k
    k = k / 2
  END DO
  j = j + k
END DO
DO i = 0, N - 1
  x(i) = x(i) / N
  y(i) = y(i) / N
END DO
END
```

19.13 Note that the results for the $n = 32$ case should be the same as for the DFT as in the first part of the solution of Prob. 19.11 as shown above. The runs for $N = 32$, 64 and 128 were performed with the following results obtained. (Note that even though we used a slow PC, we had to call the function numerous times to obtain measurable times. These times were then divided by the number of function calls to determine the time per call shown below)

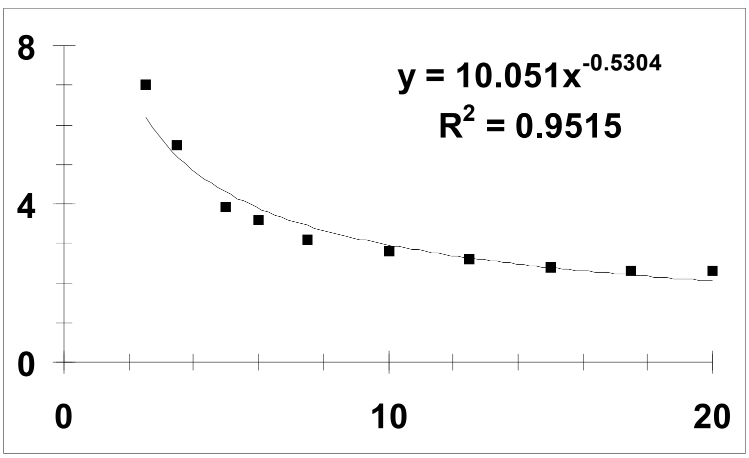
| *N* | time (s) |
|---|---|

| | |
|---|---|
| 32 | 0.0135 |
| 64 | 0.031 |
| 128 | 0.068 |

A plot of time versus $N \log_2 N$ yielded a straight line (see plot below). Thus, the result verifies that the execution time $\propto N \log_2 N$.




19.14 Using a similar approach to that described in Example 19.3, the Excel Chart Wizard and the Trendline tool can be used to create the following fit:




19.15 Using a similar approach to Example 19.4, the following spreadsheet can be set up:

| *T* | *T^2* | *T^3* | *T^4* | *o* |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 14.621 |
| 8 | 64 | 512 | 4096 | 11.843 |
| 16 | 256 | 4096 | 65536 | 9.87 |
| 24 | 576 | 13824 | 331776 | 8.418 |
| 32 | 1024 | 32768 | 1048576 | 7.305 |
| 40 | 1600 | 64000 | 2560000 | 6.413 |

The Data Analysis Toolpack can then be used to generate

SUMMARY OUTPUT

| *Regression Statistics* | |
|---|---|
| Multiple R | 0.99999994 |

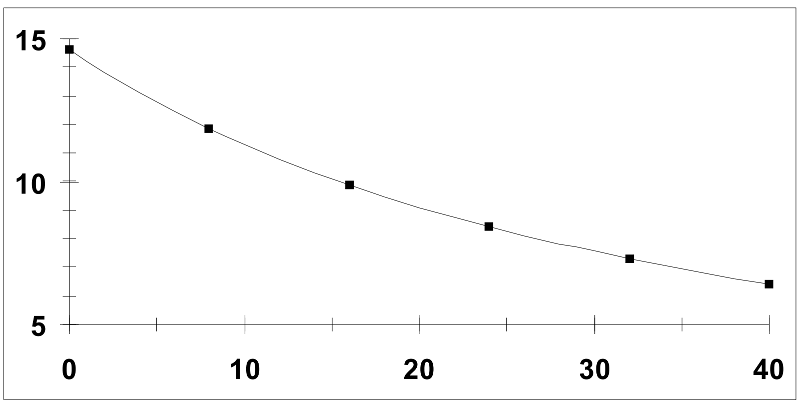
| | |
|---|---|
| R Square | 0.99999988 |
| Adjusted R Square | 0.99999939 |
| Standard Error | 0.00239377 |
| Observations | 6 |

ANOVA

| | *df* | *SS* | *MS* | *F* | *Significance F* |
|---|---|---|---|---|---|
| Regression | 4 | 47.0093523 | 11.75234 | 2050962 | 0.0005237 |
| Residual | 1 | 5.7302E-06 | 5.73E-06 | | |
| Total | 5 | 47.009358 | | | |

| | *Coefficients* | *Std Error* | *t Stat* | *P-value* | *Lower 95%* | *Upper 95%* |
|---|---|---|---|---|---|---|
| Intercept | 14.6208492 | 0.00238902 | 6120.018 | 0.000104 | 14.59049395 | 14.6512 |
| X Variable 1 | -0.4113267 | 0.0011012 | -373.527 | 0.001704 | -0.425318685 | -0.39733 |
| X Variable 2 | 0.0090115 | 0.00013149 | 68.53234 | 0.009289 | 0.007340736 | 0.010682 |
| X Variable 3 | -0.0001302 | 5.1867E-06 | -25.1078 | 0.025342 | -0.000196129 | -6.4E-05 |
| X Variable 4 | 8.4432E-07 | 6.4426E-08 | 13.10526 | 0.048483 | 2.57132E-08 | 1.66E-06 |

The polynomial along with the data can be plotted as




19.16 Linear regression can be implemented with the Data Analysis Toolpack in a fashion similar to Example 19.4. After setting the data ranges, the confidence interval box should be checked an set to 90% in order to generate 90% confidence intervals for the coefficients. The result is

SUMMARY OUTPUT

| Regression Statistics | |
|---|---|
| Multiple R | 0.98465 |
| R Square | 0.969535 |
| Adjusted R Square | 0.963442 |
| Standard Error | 1.625489 |
| Observations | 7 |

ANOVA

| | df | SS | MS | F | Significance F |
|---|---|---|---|---|---|
| Regression | 1 | 420.4375 | 420.4375 | 159.1232 | 5.56E-05 |

| | | | |
|---|---|---|---|
| Residual | 5 | 13.21107 | 2.642214 |
| Total | 6 | 433.6486 | |

| | Coefficients | Standard Error | t Stat | P-value | Lower 95% | Upper 95% | Lower 90.0% | Upper 90.0% |
|---|---|---|---|---|---|---|---|---|
| Intercept | 0.714286 | 1.373789 | 0.519938 | 0.625298 | -2.81715 | 4.245717 | -2.05397 | 3.482538 |
| X Variable 1 | 1.9375 | 0.153594 | 12.6144 | 5.56E-05 | 1.542674 | 2.332326 | 1.628 | 2.247 |

The 90% confidence interval for the intercept is from -2.05 to 3.48, which encompasses zero. The regression can be performed again, but with the "Constant is Zero" box checked on the regression dialogue box. The result is

SUMMARY OUTPUT

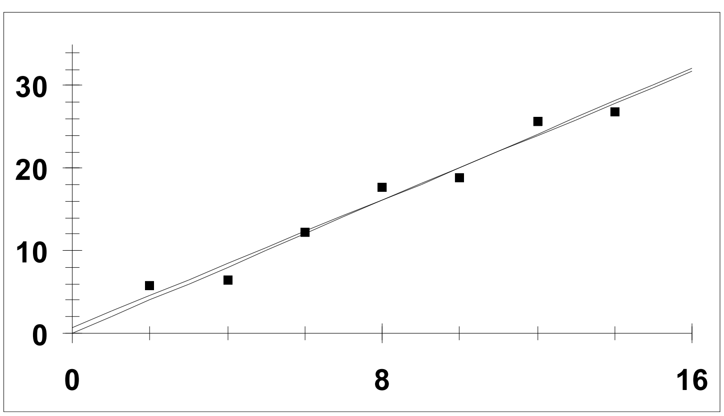
| Regression Statistics | |
|---|---|
| Multiple R | 0.983813 |
| R Square | 0.967888 |
| Adjusted R Square | 0.801221 |
| Standard Error | 1.523448 |
| Observations | 7 |

ANOVA

| | df | SS | MS | F | Significance F |
|---|---|---|---|---|---|
| Regression | 1 | 419.7232 | 419.7232 | 180.8456 | 4.07E-05 |
| Residual | 6 | 13.92536 | 2.320893 | | |
| Total | 7 | 433.6486 | | | |

| | Coefficients | Standard Error | t Stat | P-value | Lower 95% | Upper 95% | Lower 90.0% | Upper 90.0% |
|---|---|---|---|---|---|---|---|---|
| Intercept | 0 | #N/A | #N/A | #N/A | #N/A | #N/A | #N/A | #N/A |
| X Variable 1 | 2.008929 | 0.064377 | 31.20549 | 7.19E-08 | 1.851403 | 2.166455 | 1.883832 | 2.134026 |

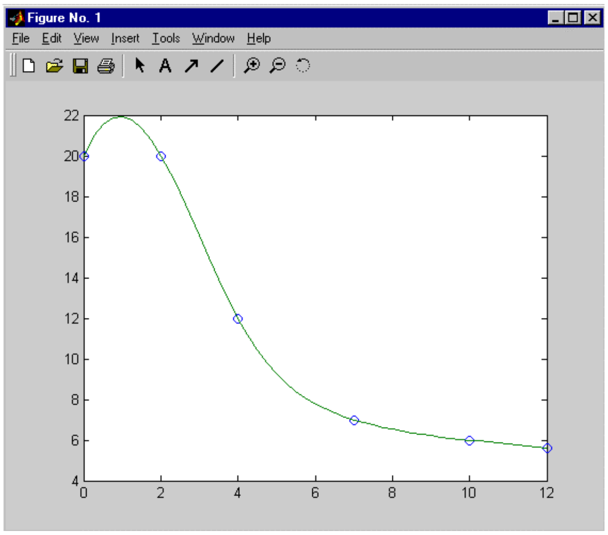
The data along with both fits is shown below:




19.17 Using MATLAB:

```
>> x=[0 2 4 7 10 12];
>> y=[20 20 12 7 6 5.6];
```

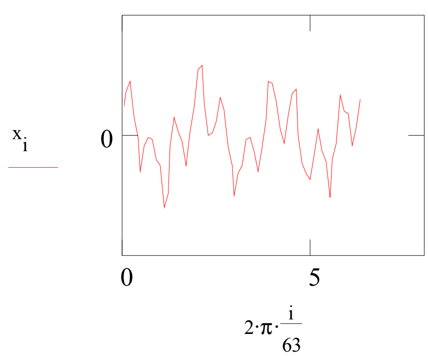
```
>> xi=0:.25:12;
>> yi=spline(x,y,xi);
>> plot(x,y,'o',xi,yi)
```




```
>> spline(x,y,3)

ans =
   16.0669
```

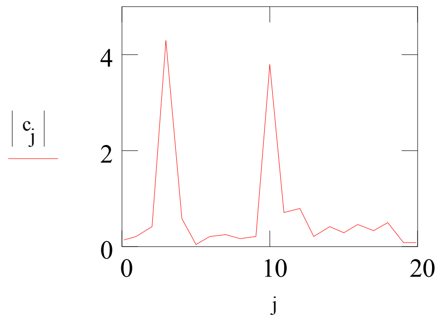
## 19.18 Using Mathcad

$i := 0..63$




$c := \mathrm{fft}(x)$

$j := 0..20$




19.19 As in Example 19.5, the data can be entered as

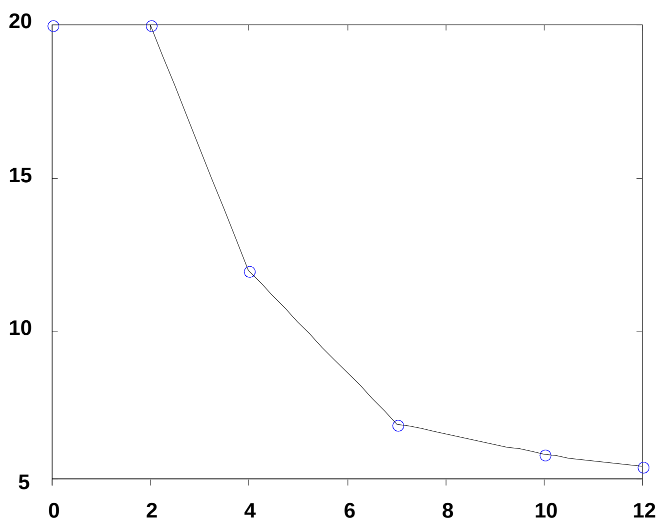
```
>> x=[0 2 4 7 10 12];
>> y=[20 20 12 7 6 5.6];
```

Then, a set of x values can be generated and the interp1 function used to generate the linear interpolation

```
>> xi=0:.25:12;
>> yi=interp1(x,y,xi);
```

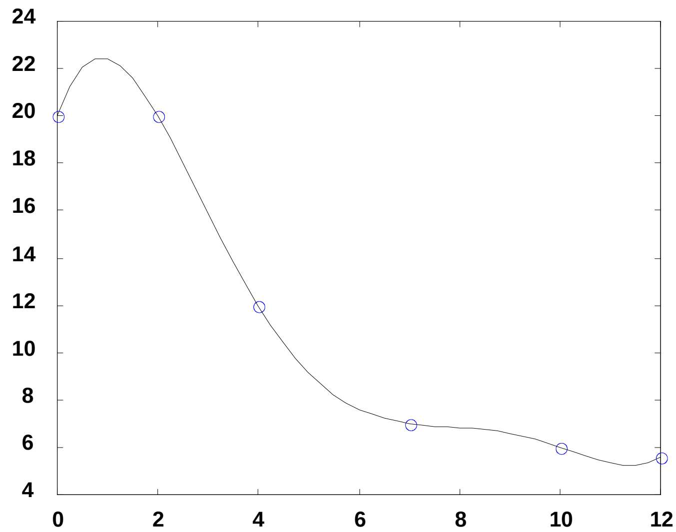
These points can then be plotted with

```
>> plot(x,y,'o',xi,yi)
```



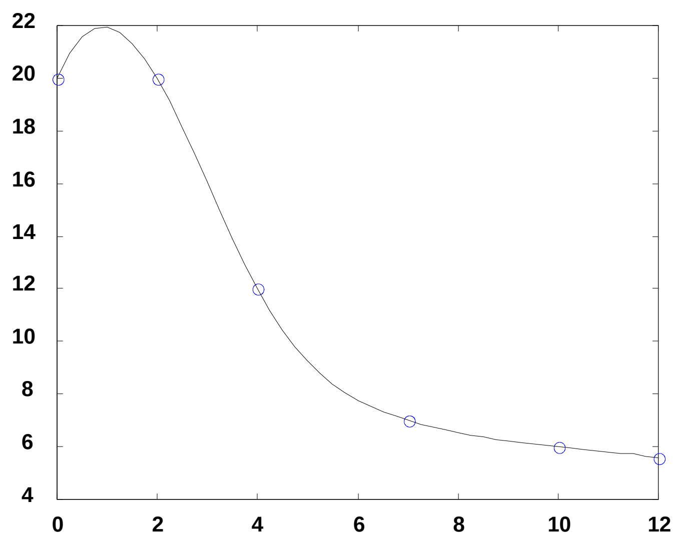

The 5th-order interpolating polynomial and plot can be generated with

```
>> p=polyfit(x,y,5)
p =
    0.0021   -0.0712    0.8909   -4.5982    6.1695   20.0000
>> yi=polyval(p,xi);
>> plot(x,y,'o',xi,yi)
```



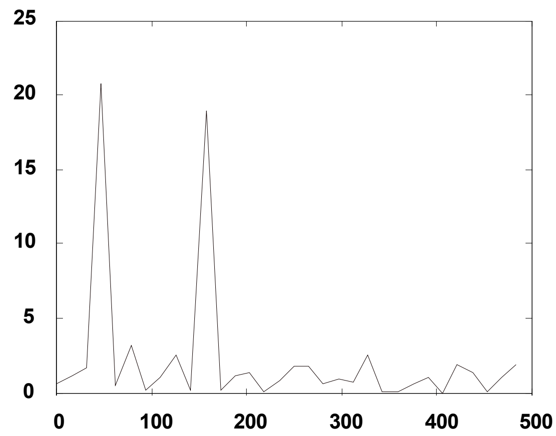

The cubic spline and plot can be generated with

```
>> yi=spline(x,y,xi);
>> plot(x,y,'o',xi,yi)
```




19.20 The following MATLAB session develops the fft along with a plot of the power spectral density versus frequency.

```
>> t=0:63;
>> y=cos(3*2*pi*t/63)+sin(10*2*pi*t/63)+randn(size(t));
>> Y=fft(y,64);
>> Pyy=Y.*conj(Y)/64;
>> f=1000*(0:31)/64;
>> plot(f,Pyy(1:32))
```



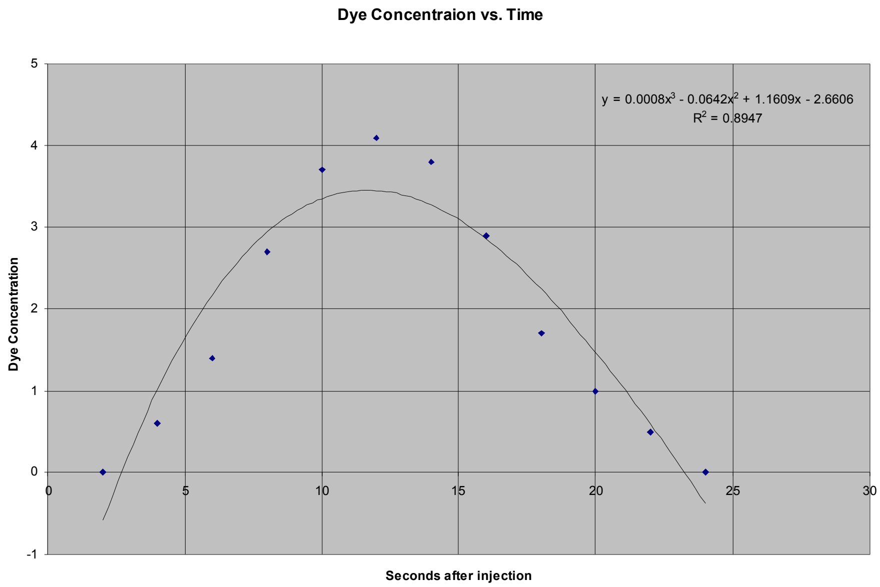

19.21

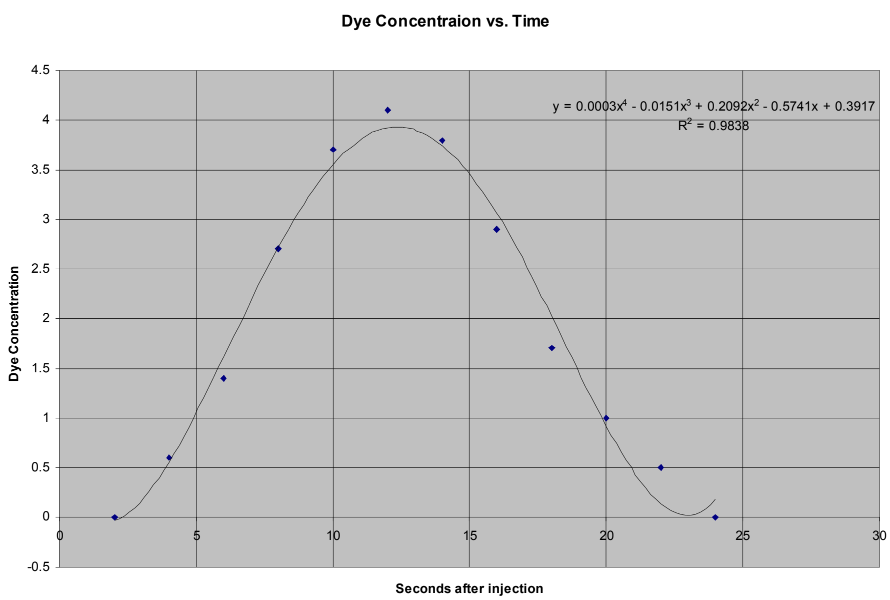
```
PROGRAM Fitpoly
Use IMSL
Implicit NONE
Integer::ndeg,nobs,i,j
Parameter (ndeg=4, nobs=6)
Real:: b (ndeg + 1), sspoly(ndeg + 1), stat(10), X(nobs), y(nobs), ycalc
(nobs)
Data x/0,8,16,24,32,40/
Data y/14.621,11.843,9.870,8.418,7.305,6.413/
Call Rcurv(nobs, X, y, ndeg, b, sspoly, stat)
Print *, 'Fitted polynomial is'
Do i = 1,ndeg+1
  Print 10, i - 1, b(i)
End Do
Print *
Print 20, stat(5)
Print *
Print *, '      No.           X            Y           YCALC'
Do i = 1,nobs
  ycalc = 0
  Do j = 1,ndeg+1
    ycalc(i) = ycalc(i) + b(j)*x(i)**(j-1)
  End Do
  Print 30, i, X(i), y(i), ycalc(i)
End Do
10 Format(1X, 'X^',I1,' TERM: ',F8.4)
20 Format(1X,'R^2: ',F8.3,'%')
30 Format(1X,I8,3(5X,F8.4))
End
```

```
Output:

Fitted polynomial is
 X^0 TERM:  14.6208
 X^1 TERM:  -0.4113
 X^2 TERM:   0.0090
 X^3 TERM:  -0.0001
 X^4 TERM:   0.0000

 R^2:  100.000%

       No.          X           Y          YCALC
        1       0.0000      14.6210      14.6208
        2       8.0000      11.8430      11.8438
        3      16.0000       9.8700       9.8685
        4      24.0000       8.4180       8.4195
        5      32.0000       7.3050       7.3042
```

6 40.0000 6.4130 6.4132

19.22 Using Excel, plot the data and use the trend line function to fit a polynomial of specific order. Obtain the R – squared value to determine the goodness of fit.




Use the 4th order polynomial:

$$C = 0.0003t^4 - 0.0151t^3 + 0.2092t^2 - 0.5741t + 0.3917$$

Integrate to find the area under the curve:

$$\int_{2}^{24} 0.0003t^4 - 0.0151t^3 + 0.2092t^2 - 0.5741t + 0.3917 \;\; dt = 33.225$$

Area under curve:

33.225 mg sec/L

Cardiac output = $\dfrac{5\,mg}{33.225\;mg\;\sec/\,L} = 0.15049\;L/\sec = 9\;L/\min$

Cardiac output ≅ 9 L/min

19.23 Plug in $A_o = 1$ and $T = \frac{1}{4} \Rightarrow$

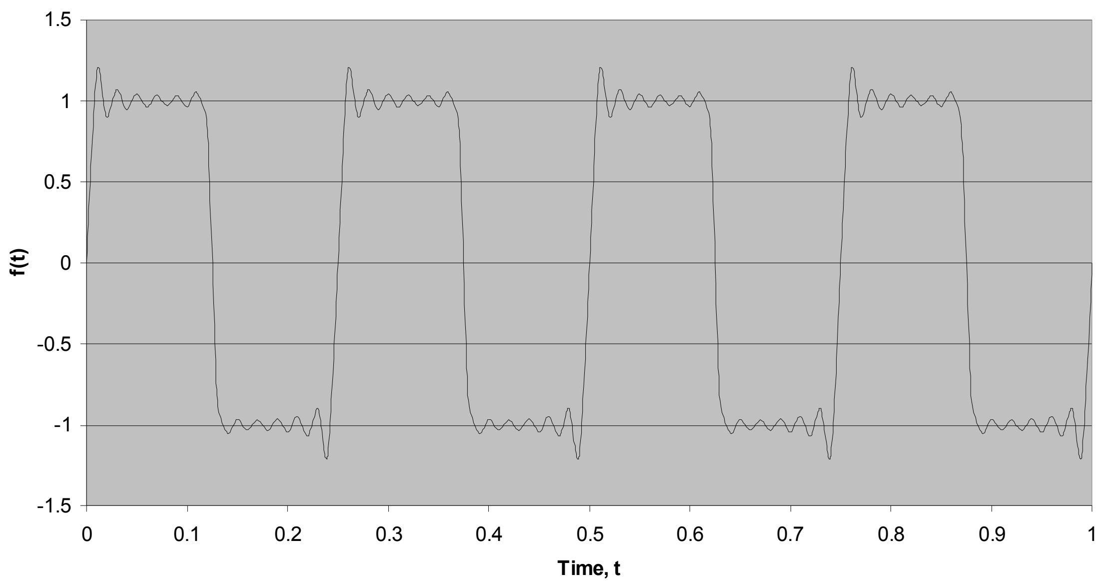
$$f(t) = \sum_{n=1}^{\infty} \left( \frac{4A_o}{(2n-1)\pi} \right) \sin\left( \frac{2\pi(2n-1)t}{T} \right)$$

Make table and plot in Excel $\Rightarrow$

Shown on the following pages

| **time->** | 0 | 0.01 | 0.02 | 0.03 | 0.04 | 0.05 | 0.06 | 0.07 | 0.08 | 0.09 | 0.1 | 0.11 | 0.12 | 0.13 | 0.14 | 0.15 | 0.16 | 0.17 | 0.18 | 0.19 | 0.2 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **n** | | | | | | | | | | | | | | | | | | | | | |
| 1 | 0 | 0.317 | 0.613 | 0.872 | 1.075 | 1.211 | 1.271 | 1.251 | 1.152 | 0.981 | 0.748 | 0.469 | 0.160 | -0.160 | -0.469 | -0.748 | -0.981 | -1.152 | -1.251 | -1.271 | -1.211 |
| 2 | 0 | 0.291 | 0.424 | 0.327 | 0.053 | -0.249 | -0.417 | -0.358 | -0.106 | 0.204 | 0.404 | 0.384 | 0.156 | -0.156 | -0.384 | -0.404 | -0.204 | 0.106 | 0.358 | 0.417 | 0.249 |
| 3 | 0 | 0.242 | 0.150 | -0.150 | -0.242 | 0.000 | 0.242 | 0.150 | -0.150 | -0.242 | 0.000 | 0.242 | 0.150 | -0.150 | -0.242 | 0.000 | 0.242 | 0.150 | -0.150 | -0.242 | 0.000 |
| 4 | 0 | 0.179 | -0.067 | -0.154 | 0.125 | 0.107 | -0.165 | -0.045 | 0.182 | -0.023 | -0.173 | 0.088 | 0.140 | -0.140 | -0.088 | 0.173 | 0.023 | -0.182 | 0.045 | 0.165 | -0.107 |
| 5 | 0 | 0.109 | -0.139 | 0.068 | 0.052 | -0.135 | 0.119 | -0.018 | -0.097 | 0.141 | -0.083 | -0.035 | 0.128 | -0.128 | 0.035 | 0.083 | -0.141 | 0.097 | 0.018 | -0.119 | 0.135 |
| 6 | 0 | 0.043 | -0.079 | 0.105 | -0.116 | 0.110 | -0.089 | 0.056 | -0.015 | -0.029 | 0.068 | -0.098 | 0.114 | -0.114 | 0.098 | -0.068 | 0.029 | 0.015 | -0.056 | 0.089 | -0.110 |
| Sum | 0 | 1.180 | 0.901 | 1.068 | 0.947 | 1.044 | 0.962 | 1.035 | 0.967 | 1.033 | 0.964 | 1.050 | 0.847 | -0.847 | -1.050 | -0.964 | -1.033 | -0.967 | -1.035 | -0.962 | -1.044 |

**Sum of the First Six Terms of the Fourier Series**



# Chapter 21

21.1

a) $$\int_0^3 (1-e^{-x})\,dx = x+e^{-x}\Big]_0^3$$

$$= 2.049787$$

b) $$\int_{-2}^{4} (1-x-4x^3+x^5)\,dx$$

$$= x-\frac{x^2}{2}-x^4+\frac{x^6}{6}\Big]_{-2}^{4}$$

$$= 432$$

c) $$\int_0^{\pi/2} (8+4\sin x)\,dx$$

$$= 8x-4\cos x\Big]_0^{\pi/2}$$

$$= 16.56637$$

21.2

a) $a=0 \qquad f(a)=0$
$b=3 \qquad f(b)=0.950213$

$$I=(3-0)\ \frac{0+0.950213}{2}$$

$$= 1.425319$$

b) $a=-2 \qquad f(a)=3$
$b=4 \qquad f(b)=765$

$$I=(4--2)\ \frac{3+765}{2}$$

$$= 2304$$

c) $a=0 \qquad f(a)=8$
$b=\pi/2 \qquad f(b)=12$

$$I=(\pi/2-0)\left[\frac{8+12}{2}\right]$$

$$= 15.708$$

21.3

a) $\underline{n=2}$

$$I=(3-0)\left[\frac{0+2(0.77687)+0.950213}{4}\right]$$

$$= 1.877964$$

$\underline{n=4}$

$$I=(3-0)\left[\frac{0+2(0.5276+0.7769+0.8946)+0.950213}{8}\right]$$

$$= 2.005658$$

$\underline{n=6}$

$$I=(3-0)\left[\frac{0+2(0.3935+0.63212+0.7769+0.8646+0.918)+0.950213}{12}\right]$$

$$= 2.030073$$

b) $\underline{n=2}$

$$I=6\left[\frac{3+2(-3)+765}{4}\right]$$

$$= 1143$$

$\underline{n=4}$

$$I = 6\left[\frac{3+2(1.987-3+33.66)+765}{8}\right]$$

$$= 624.94$$

$\underline{n=6}$

$$I = 6\left[\frac{3+2(5+1-3-1+133)+765}{12}\right]$$

$$= 519$$

c) $\underline{n=2}$

$$I = \left(\frac{\pi}{2}-0\right)\left[\frac{8+2(10.82)+12}{4}\right]$$

$$= 16.3586$$

$\underline{n=4}$

$$I = \left(\frac{\pi}{2}-0\right)\left[\frac{8+2(9.53+10.82+11.69)+12}{8}\right]$$

$$= 16.515$$

$\underline{n=6}$

$$I = \left(\frac{\pi}{2}-0\right)\left[\frac{8+2(9.035+10+10.83+11.46+11.86)+12}{12}\right]$$

$$= 16.5435$$

21.4

a)
$x_0 = 0 \qquad f(x_0) = 0$
$x_1 = 1.5 \qquad f(x_1) = 0.7769$
$x_2 = 3 \qquad f(x_2) = 0.950$

$$I = (3-0)\ \frac{0+4(.7769)+0.950}{6}$$
$$= 2.0288$$

b)
$x_0 = -2 \qquad f(x_0) = 3$
$x_1 = 1 \qquad f(x_1) = -3$
$x_2 = 4 \qquad f(x_2) = 765$

$$I = (4--2)\left[\frac{3+4(-3)+765}{6}\right]$$

$$= 756$$

c)
$x_0 = 0 \qquad f(x_0) = 8$
$x_1 = \pi/4 \qquad f(x_1) = 10.828$
$x_2 = \pi/2 \qquad f(x_2) = 12$

$$I = \left(\frac{\pi}{2}-0\right)\left[\frac{8+4(10.828)+12}{6}\right]$$
$$= 16.5755$$

21.5 $\underline{n=4}$

a)
$$I = 3\left[\frac{0+4(0.5276+0.8946)+2(0.777)+0.9502}{12}\right]$$

$$= 2.0482$$

$\underline{n=6}$

$$I = 3\left[\frac{0+4(0.393+0.777+0.918)+2(0.632+0.865)+0.9502}{18}\right]$$

$$= 2.0495$$

b) $\underline{n=4}$

$$I=(4--2)\left[\frac{3+4(1.97+33.66)+2(-3)+765}{12}\right]$$

$$=452.25$$

$\underline{n=6}$

$$I=(4--2)\left[\frac{3+4(5-3+133)+2(1-1)+765}{18}\right]$$

$$=436$$

c) $\underline{n=4}$

$$I=\left(\frac{\pi}{2}-0\right)\left[\frac{8+4(9.531+11.696)+2(10.828)+12}{12}\right]$$

$$=16.5669$$

$\underline{n=6}$

$$I=\left(\frac{\pi}{2}-0\right)\left[\frac{8+4(9.03+10.83+11.86)+2(10+11.46)+12}{18}\right]$$

$$=16.56648$$

21.6 a)

$$\begin{array}{ll} x_0=0 & f(x_0)=0 \\ x_1=1 & f(x_1)=0.632 \\ x_2=2 & f(x_2)=0.865 \\ x_3=3 & f(x_3)=0.950 \end{array}$$

$$I=(3-0)\left[\frac{0+3(0.632+0.865)+0.95}{8}\right]$$

$$=2.0402$$

b)

$$\begin{array}{ll} x_0=-2 & f(x_0)=3 \\ x_1=0 & f(x_1)=1 \\ x_2=2 & f(x_2)=-1 \\ x_3=4 & f(x_3)=765 \end{array}$$

$$I=(4--2)\left[\frac{3+3(1-1)+765}{8}\right]$$

$$=576$$

c)

$$\begin{array}{ll} x_0=0 & f(x_0)=8 \\ x_1=\pi/6 & f(x_1)=10 \\ x_2=\pi/3 & f(x_2)=11.46 \\ x_3=\pi/2 & f(x_3)=12 \end{array}$$

$$I=\left(\frac{\pi}{2}-0\right)\left[\frac{8+3(10+11.46)+12}{8}\right]$$

$$=16.5704$$

21.7 5 segments

$$\underbrace{\bullet\ \bullet\ \bullet}_{\text{use 1/3 Rule}}\underbrace{\ \bullet\ \bullet\ \bullet}_{\text{use 3/8}}$$

a)

$$\left.\begin{array}{l} x_0=0 \\ x_1=.6 \\ x_2=1.2 \end{array}\right\}\ \text{1/3 Rule}$$

$$\left.\begin{array}{l} x_0=1.2 \\ x_1=1.8 \\ x_2=2.4 \\ x_3=3. \end{array}\right\}\ \text{3/8 Rule}$$

$$I = (1.2-0)\frac{[0+4(0.4512)+0.699]}{6}$$

$$+ (3-1.2)\frac{[0.699+3(0.835+0.909)+0.950]}{8}$$

$$= 0.5007119 + 1.54822$$
$$= 2.048932$$

b) $\left.\begin{array}{l} x_0 = -2 \\ x_1 = -0.8 \\ x_2 = 0.4 \end{array}\right\} \frac{1}{2}$ Rule $\quad \left.\begin{array}{l} x_0 = 0.4 \\ x_1 = 1.6 \\ x_2 = 2.8 \\ x_3 = 4.0 \end{array}\right\} \frac{3}{8}$ Rule

$$I = (0.4--2)\frac{[3+4(3.52)+0.354]}{6}$$

$$+ (4-0.4)\frac{[0.354+3(-6.50+82.50)+765]}{8}$$

$$= 6.97421 + 447.0056$$
$$= 453.9798$$

c) $\left.\begin{array}{l} x_0 = 0 \\ x_1 = 0.314 \\ x_2 = 0.628 \end{array}\right\} \frac{1}{3}$ Rule $\quad \left.\begin{array}{l} x_0 = 0.628 \\ x_1 = 0.942 \\ x_2 = 1.257 \\ x_3 = 1.571 \end{array}\right\} \frac{3}{8}$ Rule

$$I = (0.628-0)\frac{[8+4(9.236)+10.351]}{6}$$

$$+ (1.5708-0.628)\frac{[10.35+3(11.24+11.80)+12]}{8}$$

$$I = 5.790522 + 10.77629$$
$$= 16.566812$$

21.8 as an example for $n=4$

| | |
|---|---|
| $x_0 = 1$ | $f(x_0) = 4.0$ |
| $x_1 = 1.25$ | $f(x_1) = 4.2025$ |
| $x_2 = 1.5$ | $f(x_2) = 4.6944$ |
| $x_3 = 1.75$ | $f(x_3) = 5.389$ |
| $x_4 = 2.0$ | $f(x_4) = 6.25$ |

$$I = (2-1)\frac{[4+2(4.2025+4.6944+5.389)+6.25]}{8}$$

$$= 4.852744$$

$$\epsilon_t = \frac{4.8333-4.8527}{4.8333} \times 100$$
$$= -0.40$$

| $n$ | $I$ | $\epsilon_t$ % |
|---|---|---|
| 1 | 5.125 | -6.06 |
| 2 | 4.9097 | -1.58 |
| 3 | 4.8677 | -0.71 |
| 4 | 4.8527 | -0.40 |

21.9 $\int_{-3}^{5} (4x+5)^3\,dx$

$$= \left.\frac{(4x+5)^4}{16}\right]_{-3}^{5} = 24264$$

$n=4$

$$I = (5-(-3))\frac{[-343+4(1+4913)+2(729)+15625]}{12}$$

$$= 24264$$

$n = 5$ use one application of 1/3 and 3/8 Rules

$$I = (.2-(-3))\left[\frac{-343+4(-0.2156)+195.1}{6}\right]$$

$$+ (5-0.2)\left[\frac{195.1+3(1815.8+6434.9)+15625}{8}\right]$$

$$= -79.3344 + 24343.34$$
$$= 24264.006$$

Both are exact (except for roundoff error) because f(x) is 3rd order

21.10 $$\int_0^3 x e^{2x}\,dx = \frac{e^{2x}}{4}(2x-1)\Big]_0^3 = 504.536$$

$x_0 = 0$ $\quad f(x_0) = 0$
$x_1 = 0.75$ $\quad f(x_1) = 3.36$
$x_2 = 1.5$ $\quad f(x_2) = 30.13$
$x_3 = 2.25$ $\quad f(x_3) = 202.54$
$x_4 = 3.$ $\quad f(x_4) = 1210.29$

Trap Rule

$$I = (3-0)\left[\frac{0+2(3.36+30.13+202.54)+1210.29}{8}\right]$$

$$= 630.8784 \qquad \epsilon_t = -25.04\%$$

Simp 1/3 Rule

$$I = (3-0)\left[\frac{0+4(3.36+202.54)+2(30.13)+1210.29}{12}\right]$$

$$= 523.5356 \qquad \epsilon_t = -3.77\%$$

21.11
$$\int_0^1 15^{2x}\,dx = \frac{15^{2x}}{2\ln(15)}\Big]_0^1$$

$$= 41.35817$$

a)
$$I = (1-0)\left[\frac{1+225}{2}\right]$$
$$= 113 \qquad \epsilon_t = -173\%$$

b)
$$I = (1-0)\left[\frac{1+4(15)+225}{6}\right]$$
$$= 47.6667 \qquad \epsilon_t = -15.2\%$$

c)
$$I = (1-0)\left[\frac{1+3(6.08+37.0)+225}{8}\right]$$
$$= 44.4033 \qquad \epsilon_t = -7.4\%$$

d)
$$I = (1-0)\left[\frac{7(1+225)+32(3.873+58.095)+12(15)}{90}\right]$$
$$= 41.61075 \qquad \epsilon_t = -0.61\%$$

e)
$$I = (1-0)(15)$$
$$= 15 \qquad \epsilon_t = 63.7\%$$

f)

$$I = (1-0)\left[\frac{6.082 + 36.993}{2}\right]$$

$$= 21.53769 \qquad \epsilon_t = 47.9\%$$

g) $$I = (1-0)\left[\frac{2(3.873+58.095) - 1(15)}{3}\right]$$

$$= 36.31182 \qquad \epsilon_t = 12.2\%$$

21.12

$$\int_0^{\pi} (5+3\sin x)\,dx$$

$$= 5x - 3\cos x\Big]_0^{\pi} = 21.707963$$

a) $$I = (\pi-0)\left[\frac{5+5}{2}\right]$$

$$= 15.70796 \qquad \epsilon_t = 27.64\%$$

b) $$I = (\pi-0)\left[\frac{5+4(8)+5}{6}\right]$$

$$= 21.99115 \qquad \epsilon_t = -1.30\%$$

c) $$I = (\pi-0)\left[\frac{5+3(7.6+7.6)+5}{8}\right]$$

$$= 21.82954 \qquad \epsilon_t = -0.56\%$$

d) $$I = (1.2566-0)\left[\frac{5+4(6.763)+7.853}{6}\right] + (3.14159-1.2566)\left[\frac{7.853+3(7.853+6.763)+5}{8}\right]$$

$$= 8.357726 + 13.36056$$
$$= 21.718286 \qquad \epsilon_t = -0.05\%$$

e) $$I = (\pi-0)\left[\frac{7(5+5)+32(7.12+7.12)+12(8)}{90}\right]$$

$$= 21.70367 \qquad \epsilon_t = 0.02\%$$

f) $$I = (\pi-0)\,[8]$$

$$= 25.13274 \qquad \epsilon_t = -15.8\%$$

g) $$I = (\pi-0)\left[\frac{7.598+7.598}{2}\right]$$

$$= 23.87006 \qquad \epsilon_t = -10.00\%$$

h) $$I = (\pi-0)\left[\frac{2(7.12+7.12) - 1(8)}{3}\right]$$

$$= 21.45214 \qquad \epsilon_t = 1.2\%$$

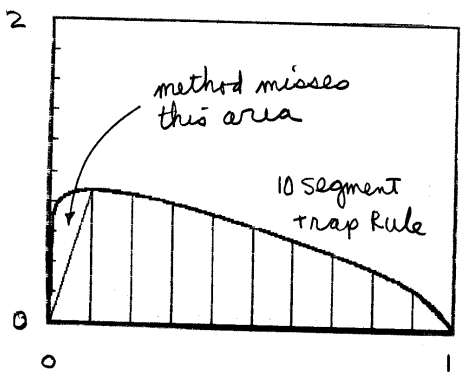
21.13 with 1000 segments

$$I = 0.6020474$$

which has 3 digit accuracy

a plot of the function shows that function rises steeply from $x=0$.

The trapezoidal rule tends to under estimate true value.




21.14

$$I = (0.5-0)\left[\frac{1+2(7+4+3+5)+2}{10}\right]$$

$$= 2.05$$

21.15

$$I = (0.2-0)\left[\frac{1+4(7)+4}{6}\right]$$

$$+ (0.5-0.2)\left[\frac{4+3(3+5)+2}{8}\right]$$

$$= 1.1 + 1.125 = 2.225$$

21.16

$$I = (11-(-3))\left[\frac{1+2(-4-9+2+4+2+6)-3}{14}\right]$$

$$= 0$$

21.17

$$I = (5-(-3))\left[\frac{1+4(-4+2)+2(-9)+4}{12}\right]$$

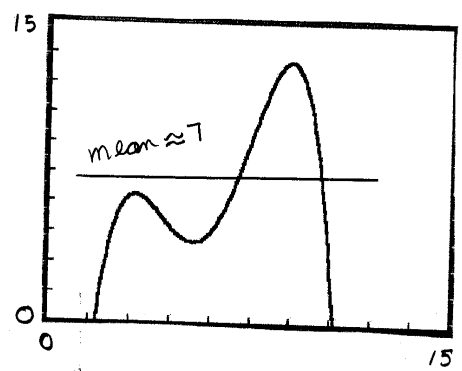
$$+ (11-5)\left[\frac{4+3(2+6)-3}{8}\right]$$

$$= -14 + 18.75$$

$$= 4.75$$

21.18

a)

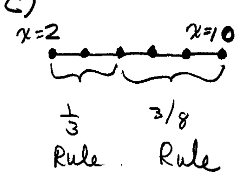



b)

$$\overline{f(x)} = \frac{\int_2^{10}\left[-46+45.4x-13.8x^2+1.71x^3-0.0729x^4\right]dx}{8}$$

$$= \frac{58.62656}{8} = 7.32832$$

c)



$$I = (5.2-2)\left[\frac{2.114 + 4(6.129) + 4.066}{6}\right]$$

$$+ (10-5.2)\left[\frac{4.066 + 3(6.416 + 12.208) + 9}{8}\right]$$

$$= 16.37178 + 41.36316$$

$$= 57.73494$$

$$\bar{f} = 57.73494/8 = 7.21687$$

$$\epsilon_t = \frac{7.32832 - 7.21687}{7.32832} \times 100 = 1.5\%$$

21.19

a) $$\int_0^{1.2} e^{-x}\,dx = -e^{-x}\Big]_0^{1.2} = 0.6988$$

b) $$I = (0.1)\left(\frac{1 + 0.9048}{2}\right)$$

$$+ (0.6)\left[\frac{0.9048 + 2(0.7408 + 0.6065) + 0.4966}{6}\right]$$

$$+ (0.5)\left[\frac{0.4966 + 2(0.3867) + 0.3012}{4}\right]$$

$$= 0.09524 + 0.4096 + 0.1964$$

$$= 0.7012 \qquad \epsilon_t = -0.34\%$$

c) $$I = 0.09524 + (0.6)\left[\frac{0.9048 + 3(0.7408 + 0.6065) + 0.4966}{8}\right]$$

$$+ 0.1964$$

$$I = 0.09524 + 0.4082 + 0.1964$$

$$= 0.6998 \qquad \epsilon_t = -0.14\%$$

21.20

$$\int_{-2}^{2}\int_{0}^{4} (x^2 - 3y^2 + xy^3)\,dx\,dy$$

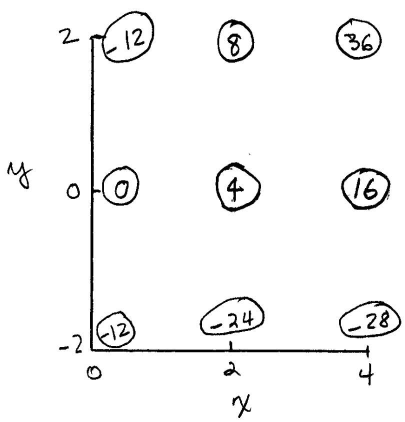
$$= \int_{-2}^{2} \left[\frac{x^3}{3} - 3y^2x + y^3\frac{x^2}{2}\right]_0^4 dy$$

$$= \int_{-2}^{2} \left(\frac{64}{3} - 12y^2 + 8y^3\right) dy$$

$$= \frac{64}{3}y - 4y^3 + 2y^4\Big]_{-2}^{2}$$

$$= 21.332$$

b)




at $y=-2$

$$I = 4\left[\frac{-12 + 2(-24) - 28}{4}\right] = -88$$

at $y=0$

$$I = 4\left[\frac{0 + 2(4) + 16}{4}\right] = 24$$

at $y=2$

$$I = 4\left[\frac{-12 + 2(8) + 36}{4}\right] = 40$$

Now integrate with respect to y

$$I = 4\left[\frac{-88 + 2(24) + 40}{4}\right]$$

$I = 0$ $\qquad \epsilon_t = 100\%$

c) $$I = 4\left[\frac{-12 + 4(-24) - 28}{6}\right] = -90.67$$

$$I = 4\left[\frac{0 + 4(4) + 16}{6}\right] = 21.33$$

$$I = 4\left[\frac{-12 + 4(8) + 36}{6}\right] = 37.33$$

With respect to y

$$I = 4\left[\frac{-90.67 + 4(21.33) + 37.33}{6}\right]$$

$= 21.333$

$\epsilon_t \approx 0$

as expected

21.21 a)

$$\int_{-4}^{4}\int_{0}^{6}\int_{-1}^{3} (x^3 + 2yz)\,dx\,dy\,dz$$

$$= \int_{-4}^{4}\int_{0}^{6} \left[\frac{x^4}{4} + 2yzx\right]_{-1}^{3} dy\,dz$$

$$= \int_{-4}^{4}\int_{0}^{6} (20 + 8yz)\,dy\,dz$$

$$= \int_{-4}^{4} \left[20y + 4y^2 z\right]_0^6 dz$$

$$= \int_{-4}^{4} (120 + 144z)\,dz$$

$$= \left[120z + 72z^2\right]_{-4}^{4} = 960$$

b) $x_0 = -1 \quad x_1 = 1 \quad x_2 = 3$

$z = -4 \quad y = 0$

$$I = 4\left[\frac{-1 + 4(1) + 27}{6}\right] = 20$$

$z = -4 \quad y = 3$

$$I = 4\left[\frac{-25 + 4(-23) + 3}{6}\right] = -76$$

$z = -4 \quad y = 6$

$$I = 4\left[\frac{-49 + 4(-47) - 21}{6}\right] = -172$$

$z = 0 \quad y = 0$

$$I = 4\left[\frac{-1 + 4(1) + 27}{6}\right] = 20$$

$z = 0 \quad y = 3$

$$I = 4\left[\frac{-1 + 4(1) + 27}{6}\right] = 20$$

$z = 0 \quad y = 6$

$$I = 4\left[\frac{-1 + 4(1) + 27}{6}\right] = 20$$

$z = 4 \quad y = 0$

$$I = 4\left[\frac{-1 + 4(1) + 27}{6}\right] = 20$$

$z = 4 \quad y = 3$

$$I = 4\left[\frac{23 + 4(25) + 51}{6}\right] = 116$$

$z = 4 \quad y = 6$

$$I = 4\left[\frac{47 + 4(49) + 75}{6}\right] = 212$$

middle integrals

$z = -4$

$$I = 6\left[\frac{20 + 4(-76) - 172}{6}\right] = -456$$

$z = 0$

$$I = 6\left[\frac{20 + 4(20) + 20}{6}\right] = 120$$

$z = 4$

$$I = 6\left[\frac{20 + 4(116) + 212}{6}\right] = 696$$

Outer Integral

$$I = 8\left[\frac{-456 + 4(120) + 696}{6}\right]$$

$$I = 960$$

$\epsilon_t = 0$ as expected

21.22 Here is a VBA code to implement the multi-segment trapezoidal rule for equally-spaced segments:

```
Option Explicit

Sub TestTrapm()

Dim n As Integer, i As Integer, ind As Integer
Dim label As String
Dim a As Single, b As Single, h As Single
Dim x(100) As Single, f(100) As Single

'Enter data and integration parameters
ind = InputBox("Functional (1) or Tabulated (2) data?")
a = InputBox("Lower bound = ")
b = InputBox("Upper bound = ")
n = InputBox("Number of segments = ")
h = (b - a) / n
If ind = 1 Then
  'generate data from function
  x(0) = a
  f(0) = fx(a)
  For i = 1 To n
    x(i) = x(i - 1) + h
    f(i) = fx(x(i))
  Next i
Else
  'user input table of data
  x(0) = a
  label = "f(" & x(0) & ") = "
  f(i) = Val(InputBox(label))
  For i = 1 To n
    x(i) = x(i - 1) + h
    label = "f(" & x(i) & ") = "
    f(i) = InputBox(label)
  Next i
End If

'invoke function to determine and display integral
MsgBox "The integral is " & Trapm(h, n, f())

End Sub

Function Trapm(h, n, f)
Dim i As Integer
Dim sum As Single
sum = f(0)
For i = 1 To n - 1
  sum = sum + 2 * f(i)
Next i
sum = sum + f(n)
Trapm = h * sum / 2
End Function

Function fx(x)
fx = 0.2 + 25 * x - 200 * x ^ 2 + 675 * x ^ 3 - 900 * x ^ 4 + 400 * x ^ 5
End Function
```

21.23 Here is a VBA code to implement the multi-segment Simpson's 1/3 rule algorithm from Fig. 21.13*c*:

```
Option Explicit

Sub TestSimpm()
```

```
Dim n As Integer, i As Integer
Dim label As String
Dim a As Single, b As Single, h As Single
Dim x(100) As Single, f(100) As Single

'Enter data and integration parameters
a = InputBox("Lower bound = ")
b = InputBox("Upper bound = ")
n = InputBox("Number of segments = ")
h = (b - a) / n

'generate data from function fx
x(0) = a
f(0) = fx(a)
For i = 1 To n
  x(i) = x(i - 1) + h
  f(i) = fx(x(i))
Next i

'invoke function Simp13m to determine and display integral
MsgBox "The integral is " & Simp13m(h, n, f())

End Sub

Function Simp13m(h, n, f)
Dim i As Integer
Dim sum As Single
sum = f(0)
For i = 1 To n - 2 Step 2
  sum = sum + 4 * f(i) + 2 * f(i + 1)
Next i
sum = sum + 4 * f(n - 1) + f(n)
Simp13m = h * sum / 3
End Function

Function fx(x)
fx = 0.2 + 25 * x - 200 * x ^ 2 + 675 * x ^ 3 - 900 * x ^ 4 + 400 * x ^ 5
End Function
```

21.24

```
Option Explicit

Sub TestUneven()

Dim n As Integer, i As Integer
Dim label As String
Dim a As Single, b As Single, h As Single
Dim x(100) As Single, f(100) As Single

'Enter data
Range("a6").Select
n = ActiveCell.Row
Selection.End(xlDown).Select
n = ActiveCell.Row - n

'Input data from sheet
Range("a6").Select

For i = 0 To n
 x(i) = ActiveCell.Value
  ActiveCell.Offset(0, 1).Select
  f(i) = ActiveCell.Value
  ActiveCell.Offset(1, -1).Select
Next i

'invoke function to determine and display integral
```

```
MsgBox "The integral is " & Uneven(n, x(), f())

End Sub

Function Uneven(n, x, f)
Dim k As Integer, j As Integer
Dim h As Single, sum As Single, hf As Single
h = x(1) - x(0)
k = 1
sum = 0#
For j = 1 To n
  hf = x(j + 1) - x(j)
 If Abs(h - hf) < 0.000001 Then
    If k = 3 Then
      sum = sum + Simp13(h, f(j - 3), f(j - 2), f(j - 1))
      k = k - 1
    Else
      k = k + 1
     End If
  Else
    If k = 1 Then
      sum = sum + Trap(h, f(j - 1), f(j))
    Else
      If k = 2 Then
        sum = sum + Simp13(h, f(j - 2), f(j - 1), f(j))
      Else
        sum = sum + Simp38(h, f(j - 3), f(j - 2), f(j - 1), f(j))
      End If
      k = 1
    End If
  End If
  h = hf
Next j
Uneven = sum
End Function

Function Trap(h, f0, f1)
Trap = h * (f0 + f1) / 2
End Function

Function Simp13(h, f0, f1, f2)
Simp13 = 2 * h * (f0 + 4 * f1 + f2) / 6
End Function

Function Simp38(h, f0, f1, f2, f3)
Simp38 = 3 * h * (f0 + 3 * (f1 + f2) + f3) / 8
End Function

Function fx(x)
fx = 0.2 + 25 * x - 200 * x ^ 2 + 675 * x ^ 3 - 900 * x ^ 4 + 400 * x ^ 5
End Function
```

21.25 (*a*)

$$M = (b-a)[\frac{f(x_o) + 2\sum_{i=1}^{n-1} f(x_i) + f(x_n)}{2n}]$$

$$M = (11-0)\left[\frac{4+2(4.15+4.6+5.35+6.4+7.75+9.4+11.35+13.6+16.15+19)+22.15}{2(11)}\right]$$

$$= 110.825 \text{ lb - ft}$$

(*b*) The 1/3 rule can only be applied to the first 10 panels. The trapezoidal rule can be applied to the 11th

$$M = (10-0)[\frac{4+4(4.15+5.35+7.75+11.35+16.15)+2(4.6+6.4+9.4+13.6)+19}{3(10)}]$$

$$+(12-11)\frac{19+22.15}{2} = 110.825 \text{ lb-ft}$$

(*c*) The 3/8 rule can only be applied to the first 9 panels and the 1/3 rule applied to the last 2:

$$M = (3-0)[\frac{4+3(4.15+4.6)+5.35}{8}] + (6-3)[\frac{5.35+3(6.4+7.75)+9.4}{8}]$$

$$+(9-6)[\frac{9.4+3(11.35+13.6)+16.15}{8}] + (11-9)[\frac{16.15+4(19)+22.15}{6}] = 110.55 \text{ lb-ft}$$

This result is exact because we're integrating a quadratic. The results of (*a*) and (*b*) are not exact because they include trapezoidal rule evaluations.

21.26

Divide the curve into sections according to $dV$ changes and use appropriate rules.

$$I_1 = 1.5(\frac{420+368}{2}) = 591$$

$$I_2 = \frac{1}{3}(368+4(333)+326) = 675.33$$

$$I_3 = \frac{3}{8}(326+3(326)+3(312)+242) = 930.75$$

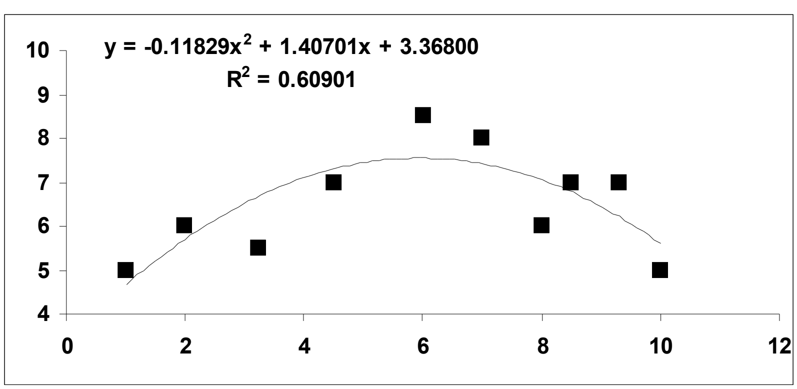
$$I_4 = 1(\frac{242+207}{2}) = 224.5$$

$$W = I_1 + I_2 + I_3 + I_4 = 2421.583$$

**Therefore, the work done is 2420 kJ.**

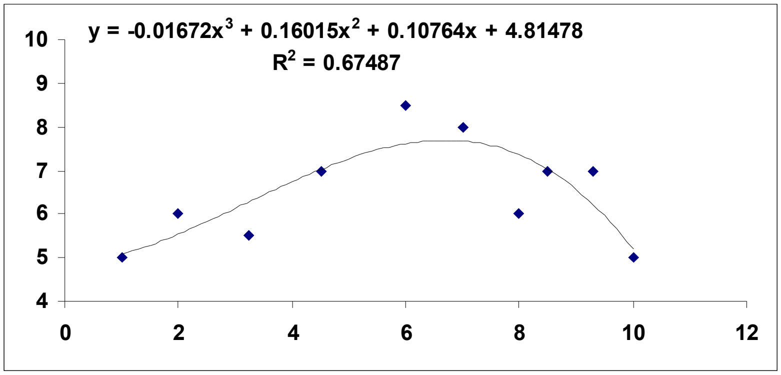
21.27 (*a*) The trapezoidal rule yields 60.425.

(*b*) A parabola can be fit to the data to give




The parabola can be integrated and evaluated from 1 to 10 to give 60.565.

($c$) A cubic can be fit to the data to give




The cubic can be integrated and evaluated from 1 to 10 to give 60.195.

Although it's not asked in the problem statement, the algorithm from Fig. 21.15*b* can also be applied (see Solution to Prob. 20.24 for code) to yield 60.258.

21.28 ($a$) The following 2 equations must hold:

$$f(a) = Qe^{ra} \tag{1}$$

$$f(b) = Qe^{rb} \tag{2}$$

Take the natural log of Eq. 1 and solve for

$$\ln Q = \ln f(a) - ra \tag{3}$$

or

$$Q = f(a)e^{-ra} \tag{4}$$

Substituting (3) into the natural log of Eq. 2 gives

$$\ln f(b) = \ln f(a) - ra + rb \tag{5}$$

and solve for

$$r = \frac{\ln\left(f(a)/f(b)\right)}{a-b} \tag{6}$$

These results can be verified for the case where $Q = 3$ and $r = -0.5$. If $a = 2$ and $b = 4$, $f(a) = 1.1036$ and $f(b) = 0.406$. Substituting these values into Eqs. 6 and 4 gives

$$r = \frac{\ln(1.0136/0.406)}{2-4} = -0.5$$

$$Q = e^{\ln(1.1036)-(-0.5)(2)} = 1.1036e^{-(-0.5)2} = 3$$

(*b*)

$$I = \int_a^b Qe^{rx}\,dx = \frac{Q}{r}\left(e^{rb} - e^{ra}\right)$$

Substituting Eq. 4

$$I = \frac{f(a)e^{-ra}}{r}\left(e^{rb} - e^{ra}\right) = \frac{f(a)}{r}\left(e^{r(b-a)} - 1\right)$$

Substituting Eq. 6

$$I = \frac{f(a)}{\dfrac{\ln\left(f(a)/f(b)\right)}{a-b}}\left(e^{\frac{\ln\left(f(a)/f(b)\right)}{a-b}(b-a)} - 1\right)$$

Simplifying

$$I = \frac{(b-a)\left(f(b) - f(a)\right)}{\ln\left(f(b)/f(a)\right)}$$

This result can be verified for the case where $Q = 3$ and $r = -0.5$. If $a = 2$ and $b = 4$, $f(a) = 1.1036$ and $f(b) = 0.406$. Substituting these values into the integral equation gives

$$I = \frac{(4-2)\left(0.406 - 1.1036\right)}{\ln\left(0.406/1.1036\right)} = 1.9353$$

which matches the analytical integral

$$I == \frac{3}{-0.5}\left(e^{-0.5(4)} - e^{-0.5(2)}\right) = 1.9353$$

## Chapter 22

22.1

| | | |
|---|---|---|
| 5.125 | 4.837963 | $\underline{\underline{4.83347}}$ |
| 4.909722 | 4.833751 | |
| 4.852744 | | |

$\varepsilon_a = 5.3\%$ (between 5.125 and 4.837963); $-0.09\%$ (between 4.837963 and 4.83347)

$$\varepsilon_t = \left[\frac{4.8333 - 4.83347}{4.8333}\right] \times 100 = -0.0097\%$$

22.2

| | | | |
|---|---|---|---|
| 1815.43 | 665.3998 | 514.0778 | $\underline{\underline{504.6932}}$ |
| 952.9072 | 523.5355 | 504.8398 | |
| 630.8784 | 506.0083 | | |
| 537.2258 | | | |

$$\varepsilon_a = \left[\frac{504.6932 - 514.0778}{504.6932}\right] \times 100 = -1.86\%$$

exact $I \simeq 504.536$ (see Prob 21.10)

$$\varepsilon_t = \frac{504.536 - 504.6932}{504.536} \times 100 = -0.03\%$$

22.3

| | | | |
|---|---|---|---|
| 0.4251706 | 2.892777 | 2.900309 | $\underline{\underline{2.881037}}$ |
| 2.275876 | 2.899838 | 2.881338 | |
| 2.743848 | 2.882494 | | |
| 2.847833 | | | |

22.4

<u>two point:</u>

$$f(x) = x + \frac{1}{x}$$

$$y = \frac{(2+1)}{2} + \frac{(2-1)}{2}x$$

$$dy = \left(\frac{2-1}{2}\right)dx$$

$$\int_1^2 f(x)\,dx = \int_{-1}^{1} f(y)\,dy$$

$$I = 2.074414\,(1) + 2.75596\,(1)$$

$$= 4.830374 \qquad \epsilon_t = 0.06\%$$

<u>three point:</u>

$$I = 2.022896\,(0.555555) + 2.347222\,(0.88888) + 2.921322\,(0.555555)$$

$$= 4.833208 \qquad \epsilon_t = 0.0019\%$$

<u>four point:</u>

$$I = 2.009026\,(0.3478548) + 2.16712\,(0.6521452) + 2.9977\,(0.3478548) + 2.573719\,(0.6521452)$$

$$= 4.833329 \qquad \epsilon_t = -0.0006$$

22.5

$$f(x) = x e^{2x}\,dx$$

$$y = \frac{3+0}{2} + \frac{3-0}{2}x$$

$$dy = \frac{3-0}{2}\,dx$$

$$\int_0^3 f(x)\,dx = \int_{-1}^{1} f(y)\,dy$$

<u>Two point:</u>

$$I = 3.379298\,(1) + 402.9158\,(1)$$

$$= 406.2951 \qquad \epsilon_t = 19.5\%$$

<u>Three point:</u>

$$I = .9972798\,(.5555556) + 45.19246\,(.8888889) + 819.1716\,(.5555556)$$

$$= 495.8205$$

$$\epsilon_t = 1.7\%$$

Four Point:

$$I = 0.4739085(0.3478548) + 10.75639(0.6521452) + 1113.793(0.3478548) + 167.9269(0.6521452)$$

$$= 504.1305 \qquad \varepsilon_t = 0.08\%$$

22.6 $f(x) = \dfrac{e^x \sin x}{1+x^2}$

$$y = \frac{3+0}{2} + \frac{3-0}{2}x$$

$$dy = \frac{3-0}{2}dx$$

$$\int_0^3 f(x)\,dx = \int_{-1}^{1} f(y)\,dy$$

Six Point:

$$I = 0.1661607(0.1713245)$$
$$0.9642544(0.3607616)$$
$$1.854998(0.4679139)$$
$$0.696431(0.1713245)$$
$$1.521079(0.3607616)$$
$$2.071476(0.4679139)$$

$$= 2.881647$$

22.7

| | | | |
|---|---|---|---|
| 224.3657 | 288.5603 | 289.4308 | $\underline{\underline{289.4351}}$ |
| 272.5117 | 289.3764 | 289.435 | |
| 285.1602 | 289.4314 | | |
| 288.3636 | | | |

$\varepsilon_a = 0.0015\%$

22.8

$$y = \frac{3-3}{2} + \frac{3-(-3)}{2}x = 3x$$

$$dy = \frac{3-(-3)}{2}dx = 3\,dx$$

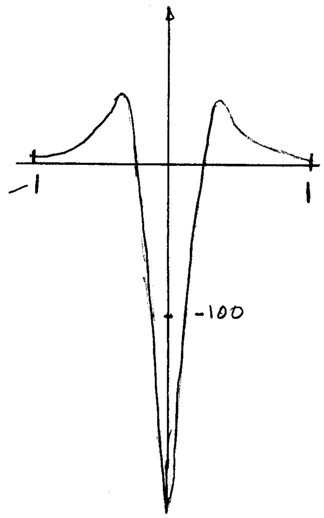
$$\int_{-3}^{3} \frac{2}{1+x^2}dx = \int_{-1}^{1} \frac{6}{1+18y^2}dy$$

| Points | I |
|---|---|
| 2 | 1.714286 |
| 3 | 5.898305 |
| 4 | 2.831287 |
| 5 | 4.517031 |
| 6 | 3.384514 |

not converging

Insight can be gained by looking at the nature of f(y) and its derivatives

$$f''(y) = \frac{(1+18y^2)^2(-216) + 216y\left[2(1+18y^2)36y\right]}{(1+18y^2)^4}$$




$f''(y)$ and higher derivatives are large. Thus integral evaluation is inaccurate because $E_t$ is related to magnitude of derivatives

22.9

a) $$\int_1^\infty \frac{dx}{x(x+2)} = \int_0^1 \frac{1}{t^2}(t)\left(\frac{1}{1/t+2}\right)dt$$

$$= \int_0^1 \frac{1}{1+2t}\,dt$$

use 8 applications of midpoint rule

$$I = \frac{1}{8}\big(0.8888 + 0.72727 + 0.61538 + 0.5333 + 0.47059 + 0.42105 + 0.38095 + 0.3478\big)$$

$= 0.5481623$

vs true value $= 0.5493061$

b) $$\int_0^\infty e^{-y}\sin^2 y\,dy = \int_0^2 e^{-y}\sin^2 y\,dy + \int_2^\infty e^{-y}\sin^2 y\,dy$$

For first part use 4 applications of Simpsons $\frac{1}{3}$ Rule

$$I = (2-0)\frac{\left[0 + 4(0.048+0.219+0.258+0.168) + 2(0.139+0.26+0.222) + 0.112\right]}{24}$$

$= 0.3438$

$$\int_2^\infty e^{-y}\sin^2 y\,dy = \int_0^{1/2} \frac{1}{t^2}e^{-1/t}\sin^2(1/t)\,dt$$

Use midpoint rule with $h = 1/8$

$$I = \frac{1}{8}(0 + 0.91 + 0.001 + 0.303) = 0.049$$

$I = 0.3438 + 0.049$

$= 0.3928$

compare with true $= 0.4$

c) $\int_0^2 \frac{dy}{(1+y^2)(1+y^2/2)}$

Use Simpsons $\frac{1}{3}$ Rule

$$I = 2\frac{\left[1+4(0.913+0.5+0.219+0.97) + 2(0.711+0.333+0.145)+0.067\right]}{24}$$

$= 0.863$

$$\int_2^\infty \frac{dy}{(1+y^2)(1+y^2/2)} = \int_0^{1/2} \frac{1}{t^2}\frac{dt}{(1+(1/t)^2)(1+\frac{1}{2t^2})}$$

Use mid point Rule

$$I = \frac{1}{8}(0.004+0.33+0.85+0.147) = 0.034$$

$$I = 0.863+0.034 = 0.897 \quad true \cong 0.92$$

d) $\int_{-2}^{2} y e^{-y} dy$ Use Simpsons Rule

$$I = (2-(-2))\frac{\left[-14.78+4(-6.72-0.824+0.303+0.335) + 2(-2.72+0+0.368)+0.2707\right]}{24}$$

$= -7.807$

$$\int_2^\infty y e^{-y} dy = \int_0^{1/2} \frac{1}{t^2}\frac{1}{t}e^{-1/t} dt$$

$$I = \frac{1}{8}(0+0.732+1.336+1.214) = 0.410$$

$$I = -7.807+0.410 = -7.397$$

e) Integrate

$$\underbrace{\frac{2}{\sqrt{2\pi}}\int_0^2 e^{-x^2/2} dx + \frac{2}{\sqrt{2\pi}}\int_2^\infty e^{-x^2/2} dx}_{\text{because of symmetry}} = I$$

Use Simpson's Rule for $\int_0^2 e^{-x^2/2} dx$

$$I = 2\frac{\left[1+4(0.969+0.755+0.458+0.216) + 2(0.882+0.607+0.325) + 0.135\right]}{24}$$

$= 1.196 \times 2 = 2.392$

$$\int_2^\infty e^{-x^2/2} dx = \int_0^{1/2} \frac{1}{t^2} e^{-(1/2t^2)} dt$$

$$I = \frac{1}{8}(0+0+0.061+0.383)$$

$= 0.056$

$$I = 2.392 + 2(0.056) = \frac{2.503}{\sqrt{2\pi}} = 0.999 \quad (true = 1.0)$$

22.10

Results vary, however for 4 applications of Simpsons 1/3 Rule

$$I = (1-0)\frac{\left[0 + 4(0.8732 + 0.7479 + 0.5483 + 0.2944) + 2(0.8270 + 0.6531 + 0.4343) + 0\right]}{24}$$

$$I = 0.57016 \qquad \epsilon_t = 5.3\%$$

22.11

```
Option Explicit

Sub RhombTest()
Dim maxit As Integer
Dim a As Single, b As Single, es As Single
a = 0
b = 0.8
maxit = 3
es = 0.001
MsgBox Rhomberg(a, b, maxit, es)
End Sub

Function Rhomberg(a, b, maxit, es)
Dim n As Integer, j As Integer, k As Integer, iter As Integer
Dim i(10, 10) As Single, ea As Single
n = 1
i(1, 1) = TrapEq(n, a, b)
iter = 0
Do
  iter = iter + 1
  n = 2 ^ iter
  i(iter + 1, 1) = TrapEq(n, a, b)
  For k = 2 To iter + 1
    j = 2 + iter - k
    i(j, k) = (4 ^ (k - 1) * i(j + 1, k - 1) - i(j, k - 1)) / (4 ^ (k - 1)
- 1)
  Next k
  ea = Abs((i(1, iter + 1) - i(1, iter)) / i(1, iter + 1)) * 100
  If (iter >= maxit Or ea <= es) Then Exit Do
Loop
Rhomberg = i(1, iter + 1)
End Function

Function TrapEq(n, a, b)
Dim i As Integer
Dim h As Single, x As Single, sum As Single
h = (b - a) / n
x = a
sum = f(x)
For i = 1 To n - 1
  x = x + h
  sum = sum + 2 * f(x)
Next i
sum = sum + f(b)
```

```
TrapEq = (b - a) * sum / (2 * n)
End Function

Function f(x)
f = 0.2 + 25 * x - 200 * x ^ 2 + 675 * x ^ 3 - 900 * x ^ 4 + 400 * x ^ 5
End Function
```

22.12

```
Option Explicit

Sub GaussQuadTest()
Dim i As Integer, j As Integer, k As Integer
Dim a As Single, b As Single, a0 As Single, a1 As Single, sum As Single
Dim c(11) As Single, x(11) As Single, j0(5) As Single, j1(5) As Single

'set constants
c(1) = 1#: c(2) = 0.888888889: c(3) = 0.555555556: c(4) = 0.652145155
c(5)  =  0.347854845:  c(6)  =  0.568888889:  c(7)  =  0.478628671:  c(8)  =
0.236926885
c(9) = 0.467913935: c(10) = 0.360761573: c(11) = 0.171324492
x(1) = 0.577350269: x(2) = 0: x(3) = 0.774596669: x(4) = 0.339981044
x(5) = 0.861136312: x(6) = 0: x(7) = 0.53846931: x(8) = 0.906179846
x(9) = 0.238619186: x(10) = 0.661209386: x(11) = 0.932469514
j0(1) = 1: j0(2) = 3: j0(3) = 4: j0(4) = 7: j0(5) = 9
j1(1) = 1: j1(2) = 3: j1(3) = 5: j1(4) = 8: j1(5) = 11

a = 0
b = 0.8
Sheets("Sheet1").Select
Range("a1").Select
For i = 1 To 5
  ActiveCell.Value = GaussQuad(i, a, b, c(), x(), j0(), j1())
  ActiveCell.Offset(1, 0).Select
Next i
End Sub

Function GaussQuad(n, a, b, c, x, j0, j1)

Dim k As Integer, j As Integer
Dim a0 As Single, a1 As Single
Dim sum As Single

a0 = (b + a) / 2
a1 = (b - a) / 2
sum = 0
If Int(n / 2) - n / 2# = 0 Then
  k = (n - 1) * 2
  sum = sum + c(k) * a1 * f(fc(x(k), a0, a1))
End If
For j = j0(n) To j1(n)
  sum = sum + c(j) * a1 * f(fc(-x(j), a0, a1))
  sum = sum + c(j) * a1 * f(fc(x(j), a0, a1))
Next j
GaussQuad = sum
End Function

Function fc(xd, a0, a1)
fc = a0 + a1 * xd
End Function

Function f(x)
f = 0.2 + 25 * x - 200 * x ^ 2 + 675 * x ^ 3 - 900 * x ^ 4 + 400 * x ^ 5
End Function
```

22.13

See solutions for Probs. 22.1, 22.2 and 22.3 for answers

22.14

See solutions for Probs. 22.4, 22.5 and 22.6 for answers

22.15

```
Option Explicit

Sub TestMidPoint()

Dim i As Integer, j As Integer, d As Integer
Dim a As Single, b As Single, h As Single, x As Single
Dim sum As Single, ea As Single, es As Single
Dim integral As Single, integralold As Single

a = -0.5
b = 0
es = 0.01

Range("a5").Select
Sheets("Sheet1").Range("a5:d25").ClearContents
Do
  integralold = integral
  d = 3 ^ i
  h = (b - a) / d
  x = a - h / 2
  sum = 0
  ActiveCell.Value = d
  ActiveCell.Offset(0, 1).Select
  ActiveCell.Value = h
  ActiveCell.Offset(0, 1).Select
  For j = 1 To d
    x = x + h
    sum = sum + f(x)
  Next j
  integral = sum * h
  i = i + 1
  ActiveCell.Value = integral
  ActiveCell.Offset(0, 1).Select
  ea = Abs((integral - integralold) / integral) * 100
  ActiveCell.Value = ea
  ActiveCell.Offset(1, -3).Select
  If ea < es Then Exit Do
Loop

End Sub

Function f(x)
f = 1 / x ^ 2 * Exp(-1 / (2 * x ^ 2))
End Function
```

| | A | B | C | D | E | F |
|---|---|---|---|---|---|---|
| 1 | Prob. 22.15 | | | | | |
| 2 | | | | | | |
| 3 | d | h | Integral | ea(%) | | RUN |
| 4 | 1 | 0.5 | 0.002683701 | 100 | | |
| 5 | 3 | 0.166667 | 0.054783922 | 95.1013 | | |
| 6 | 9 | 0.055556 | 0.056747258 | 3.45979 | | |
| 7 | 27 | 0.018519 | 0.056995228 | 0.435072 | | |
| 8 | 81 | 0.006173 | 0.057022724 | 0.04822 | | |
| 9 | 243 | 0.002058 | 0.057025783 | 0.005363 | | |
| 10 | | | | | | |

## Chapter 23

**23.1**

| | $x$ | $f(x)$ |
|---|---|---|
| $x_{i-2}$ | $\pi/12$ | 0.259 |
| $x_{i-1}$ | $\pi/6$ | 0.5 |
| $x_i$ | $\pi/4$ | 0.707 |
| $x_{i+1}$ | $\pi/3$ | 0.866 |
| $x_{i+2}$ | $5\pi/12$ | 0.966 |

true $= \cos(\pi/4) = 0.707$

| | first order | second order |
|---|---|---|
| Forward | 0.607 (14.5%) | 0.720 (−1.8%) |
| Back | 0.791 (−10.6%) | 0.726 (−2.6%) |
| Center | 0.699 (1.1%) | 0.7069 (0.014%) |

**23.2**

| | $x$ | $f(x)$ |
|---|---|---|
| $x_{i-2}$ | 16 | 1.204 |
| $x_{i-1}$ | 18 | 1.256 |
| $x_i$ | 20 | 1.301 |
| $x_{i+1}$ | 22 | 1.342 |
| $x_{i+2}$ | 24 | 1.380 |

| | first order | second order |
|---|---|---|
| Foward | 0.0205 (5.5%) | 0.02125 (2.1%) |
| Back | 0.0225 (−3.7%) | 0.02075 (4.4%) |
| Center | 0.0215 (0.9%) | 0.02133 (1.7%) |

true $= \dfrac{\log_{10} e}{20} = 0.02171$

**23.3**

| | $x$ | $f(x)$ |
|---|---|---|
| $x_{i-2}$ | 0.8 | 2.226 |
| $x_{i-1}$ | 0.9 | 2.460 |
| $x_i$ | 1 | 2.718 |
| $x_{i+1}$ | 1.1 | 3.004 |
| $x_i$ | 1.2 | 3.320 |

true = 2.718

First Derivative

| $Oh^2$ | $Oh^4$ |
|---|---|
| 2.72 (−0.07%) | 2.715 (0.1%) |

Second Derivative

| $Oh^2$ | $Oh^4$ |
|---|---|
| 2.80 (−3%) | 2.817 (−3.6%) |

Note large roundoff when 4 digit arithmetic is used.

When 8 digits are used, $Oh^4$ 2nd derivative is

2.718279
(0.0001%)

23.4

$$D(\pi/6) = \frac{0.866 - 0.5}{\pi/6} = 0.699$$

$$D(\pi/3) = \frac{0.966 - 0.259}{\pi/3} = 0.675$$

$$D = \frac{4}{3}(0.699) - \frac{1}{3}(0.675) = 0.707$$

23.5

| | $x$ | $f(x)$ |
|---|---|---|
| $x_{i-2}$ | 2 | 0.693 |
| $x_{i-1}$ | 3 | 1.099 |
| $x_i$ | 4 | 1.386 |
| $x_{i+1}$ | 5 | 1.609 |
| $x_{i+2}$ | 6 | 1.792 |

$$D(1) = \frac{1.609 - 1.099}{2} = 0.255$$

$$D(2) = \frac{1.792 - 0.693}{4} = 0.274$$

$$D = \frac{4}{3}(0.255) - \frac{1}{3}(0.274) = 0.2487$$

23.6 $x_{i-1} = -0.5$ $f(-0.5) = -1.9375$
$x_i = 1.0$ $f(1.0) = -22$
$x_{i+1} = 2.0$ $f(2.0) = -36$

$$y' = 12x^3 - 21x^2 - 10$$

$$y'(0) = -10 \quad (\text{true})$$

Eq 23.9 gives

$$f'(0) = -1.9375 \frac{(-1-2)}{(-0.5-1)(-0.5-2)} - 22 \frac{(-(-0.5))-2)}{(1-(-0.5))(1-2)} - 36 \frac{(-(-0.5))-1)}{(2-(-0.5))(2-1)}$$

$$= -13.25$$

centered difference gives

$$f'(0) = \frac{f(1) - f(-1)}{2}$$

$$= \frac{-22 - 12}{2} = -17$$

23.7 Equation 23.9 becomes at $x = x_i$

$$f'(x_i) = f(x_{i-1}) \frac{x_i - x_{i+1}}{(x_{i-1} - x_i)(x_{i-1} - x_{i+1})}$$

$$+ f(x_i) \frac{2x_i - x_{i-1} - x_{i+1}}{(x_i - x_{i-1})(x_i - x_{i+1})}$$

$$+ f(x_{i+1}) \frac{x_i - x_{i-1}}{(x_{i+1} - x_{i-1})(x_{i+1} - x_i)}$$

for equispaced points $h$ distance apart,

$$f'(x_i) = f(x_{i-1})\frac{-h}{-h(-2h)}$$

$$+ f(x_i)\frac{2x_i-(x_i-h)-(x_i+h)}{h(-h)}$$

$$+ f(x_{i+1})\frac{h}{2h(h)}$$

$$= -\frac{f(x_{i-1})}{2h} + 0 + \frac{f(x_{i+1})}{2h}$$

$$= \frac{f(x_{i+1}) - f(x_{i-1})}{2h}$$

23.8

a)

| | $x$ | $f(x)$ |
|---|---|---|
| $x_{i-2}$ | -1 | -20 |
| $x_{i-1}$ | -0.5 | -17.125 |
| $x_i$ | 0 | -15 |
| $x_{i+1}$ | .5 | -12.875 |
| $x_{i+2}$ | 1 | -10 |

Central Difference $Oh^4$

D = 4

b)

| | $x$ | $f(x)$ |
|---|---|---|
| $x_{i-2}$ | 0.1 | 0.00995 |
| $x_{i-1}$ | 0.3 | 0.08598 |
| $x_i$ | 0.5 | 0.219396 |
| $x_{i+1}$ | 0.7 | 0.374773 |
| $x_{i+2}$ | 0.9 | 0.503504 |

D = 0.756994

c)

| | $x$ | $f(x)$ |
|---|---|---|
| $x_{i-2}$ | 2.8 | 1.350242 |
| $x_{i-1}$ | 2.9 | 1.448821 |
| $x_i$ | 3.0 | 1.557408 |
| $x_{i+1}$ | 3.1 | 1.677891 |
| $x_{i+2}$ | 3.2 | 1.812656 |

D = 1.141791

d)

| | | |
|---|---|---|
| $x_{i-2}$ | 0.8 | 0.540596 |
| $x_{i-1}$ | 0.9 | 0.507503 |
| $x_i$ | 1.0 | 0.479426 |
| $x_{i+1}$ | 1.1 | 0.45518 |
| $x_{i+2}$ | 1.2 | 0.433954 |

D = -0.25998

e)

| | | |
|---|---|---|
| $x_{i-2}$ | 0.5 | 2.148721 |
| $x_{i-1}$ | 0.75 | 2.867 |
| $x_i$ | 1 | 3.718282 |
| $x_{i+1}$ | 1.25 | 4.740343 |
| $x_{i+2}$ | 1.5 | 5.981689 |

D = 3.717925

23.9 Use $Oh^2$ formulas

| $t$ | $y$ | $dy/dt$ | $d^2y/dt^2$ |
|---|---|---|---|
| 0 | 0 | 0 | 4 |
| 1 | 2 | 4 | 4 |
| 2 | 8 | 8 | 4 |
| 3 | 18 | 12 | 4 |
| 4 | 32 | 16 | 4 |
| 5 | 50 | 20 | 4 |

Note data are described by

$y = 2t^2 \quad y' = 4t \quad y'' = 4$

23.10

```
Option Explicit

Sub RhombTest()
Dim maxit As Integer
Dim a As Single, b As Single, es As Single
Dim x As Single
x = 0.5
maxit = 3
es = 0.001
MsgBox RhomDiff(x, maxit, es)
End Sub

Function RhomDiff(x, maxit, es)
Dim n As Integer, j As Integer, k As Integer, iter As Integer
Dim i(10, 10) As Single, ea As Single, del As Single, a As Single, b As Single
n = 1
i(1, 1) = DyDx(x, n)
iter = 0

Do
  iter = iter + 1
  n = 2 ^ iter
  i(iter + 1, 1) = DyDx(x, n)
  For k = 2 To iter + 1
    j = 2 + iter - k
    i(j, k) = (4 ^ (k - 1) * i(j + 1, k - 1) - i(j, k - 1)) / (4 ^ (k - 1) - 1)
  Next k
  ea = Abs((i(1, iter + 1) - i(1, iter)) / i(1, iter + 1)) * 100
  If (iter >= maxit Or ea <= es) Then Exit Do
Loop
RhomDiff = i(1, iter + 1)
End Function

Function DyDx(x, n)
Dim a As Single, b As Single
a = x - x / n
b = x + x / n
DyDx = (f(b) - f(a)) / (b - a)
End Function

Function f(x)
f = -0.1 * x ^ 4 - 0.15 * x ^ 3 - 0.5 * x ^ 2 - 0.25 * x + 1.2
End Function
```

23.11 The following program implements Eq. 23.9.

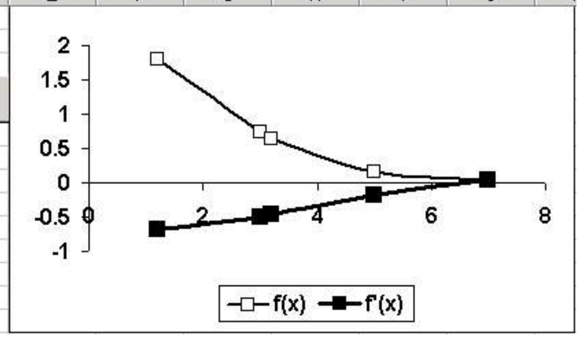
```
Option Explicit

Sub TestDerivUnequal()

Dim n As Integer, i As Integer
Dim x(100) As Single, y(100) As Single, dy(100) As Single

Range("a5").Select
n = ActiveCell.Row
Selection.End(xlDown).Select
n = ActiveCell.Row - n
Range("a5").Select
For i = 0 To n
  x(i) = ActiveCell.Value
  ActiveCell.Offset(0, 1).Select
  y(i) = ActiveCell.Value
```

```
  ActiveCell.Offset(1, -1).Select
Next i

For i = 0 To n
  dy(i) = DerivUnequal(x(), y(), n, x(i))
Next i

Range("c5").Select
For i = 0 To n
  ActiveCell.Value = dy(i)
  ActiveCell.Offset(1, 0).Select
Next i
End Sub

Function DerivUnequal(x, y, n, xx)

Dim ii As Integer

If xx < x(0) Or xx > x(n) Then
  DerivUnequal = "out of range"
Else
  If xx < x(1) Then
    DerivUnequal = DyDx(xx, x(0), x(1), x(2), y(0), y(1), y(2))
  ElseIf xx > x(n - 1) Then
    DerivUnequal = DyDx(xx, x(n - 2), x(n - 1), x(n), y(n - 2), y(n - 1),
y(n))
  Else
    For ii = 1 To n - 2
      If xx >= x(ii) And xx <= x(ii + 1) Then
        If xx - x(ii - 1) < x(ii) - xx Then
          'If the unknown is closer to the lower end of the range,
          'x(ii) will be chosen as the middle point
           DerivUnequal = DyDx(xx, x(ii - 1), x(ii), x(ii + 1), y(ii - 1),
y(ii), y(ii + 1))
        Else
          'Otherwise, if the unknown is closer to the upper end,
          'x(ii+1) will be chosen as the middle point
          DerivUnequal = DyDx(xx, x(ii), x(ii + 1), x(ii + 2), y(ii), y(ii
+ 1), y(ii + 2))
        End If
        Exit For
      End If
    Next ii
  End If
End If

End Function

Function DyDx(x, x0, x1, x2, y0, y1, y2)
DyDx = y0 * (2 * x - x1 - x2) / (x0 - x1) / (x0 - x2) _
      + y1 * (2 * x - x0 - x2) / (x1 - x0) / (x1 - x2) _
       + y2 * (2 * x - x0 - x1) / (x2 - x0) / (x2 - x1)
End Function
```

The result is

| | | |
|---|---|---|
| Prob 23.11 | | |
| x | f(x) | f'(x) |
| 1.2 | 1.807 | -0.6835 |
| 3 | 0.7468 | -0.50564 |
| 3.2 | 0.65 | -0.46236 |
| 5 | 0.1684 | -0.17314 |
| 7 | 0.03192 | 0.036663 |

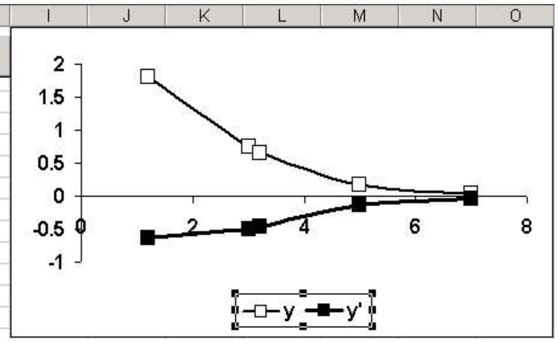
Run




An even more elegant approach is to put cubic splines through the data (recall Sec. 20.2 and the solution for Prob. 20.10) to evaluate the derivatives.

| | | | | | |
|---|---|---|---|---|---|
| Problem 23.10 | | | | | |
| Data: | | Interpolate: | | | |
| x | y | x | y | y' | y" |
| 1.2 | 1.807 | 1.2 | 1.807 | -0.63211 | 0 |
| 3 | 0.7468 | 3 | 0.7468 | -0.50278 | 0.143695 |
| 3.2 | 0.65 | 3.2 | 0.65 | -0.4608 | 0.276099 |
| 5 | 0.1684 | 5 | 0.1684 | -0.12955 | 0.091963 |
| 7 | 0.03192 | 7 | 0.03192 | -0.03759 | 0 |

RUN




23.12

(*a*) Create the following M function:

```
function y=f(x)
y=9.8*68.1/12.5*(1-exp(-12.5/68.1*x));
```

Then implement the following MATLAB session:

```
>> Q=quad('f',0,10)

Q =
  289.4351
```

(*b*)

$$d(t)=\frac{gm}{c}\int_0^t\left(1-e^{-(c/m)t}\right)dt$$

$$d(t)=\frac{gm}{c}\left[t+\frac{m}{c}e^{-(c/m)t}\right]_0^t$$

$$d(t)=\frac{9.8(68.1)}{12.5}\left[10+\frac{68.1}{12.5}e^{-(12.5/68.1)10}-0-\frac{68.1}{12.5}\right]_0^{10}=289.4351$$

(*c*) Create the following M function:

```
>> function y=f(x)
```

```
>> y=9.8*68.1/12.5*(1-exp(-12.5/68.1*x));
```

Then implement the following MATLAB session:

```
>> x=[9.99 10.01]
>> y=f(x)
>> d=diff(y)./diff(x)

d =
    1.5634
```

(*d*)

$$a(t) = \frac{gm}{c}\frac{d}{dt}\left(1 - e^{-(c/m)t}\right)$$

$$a(t) = ge^{-(c/m)t}$$

$$a(t) = 9.8e^{-(12.5/68.1)10} = -1.56337$$

23.13 (*a*) Create the following M function:

```
function y=fn(x)
y=1/sqrt(2*pi)*exp(-(x.^2)/2);
```

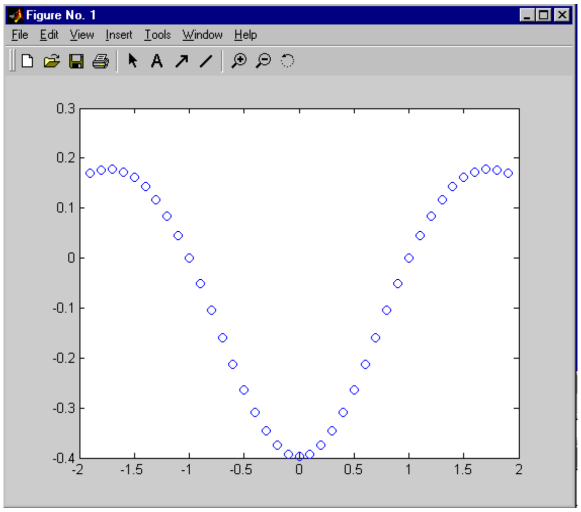
Then implement the following MATLAB session:

```
>> x=-2:.1:2;
>> y=fn(x);
>> Q=quad('fn',-1,1)

Q =
    0.6827

>> Q=quad('fn',-2,2)

Q =
    0.9545
```

Thus, about 68.3% of the area under the curve falls between –1 and 1 and about 95.45% falls between –2 and 2.

(*b*)

```
>> x=-2:.1:2
>> y=fn(x)
>> d=diff(y)./diff(x)
>> x=-1.95:.1:1.95
>> d2=diff(d)./diff(x)
>> x=-1.9:.1:1.9
>> plot(x,d2,'o')
```




Thus, inflection points ($d^2y/dx^2 = 0$) occur at –1 and 1.

23.14 (*a*) Create the following M function:

```
function y=fn(x)
y=1/sqrt(2*pi)*exp(-(x.^2)/2);
```

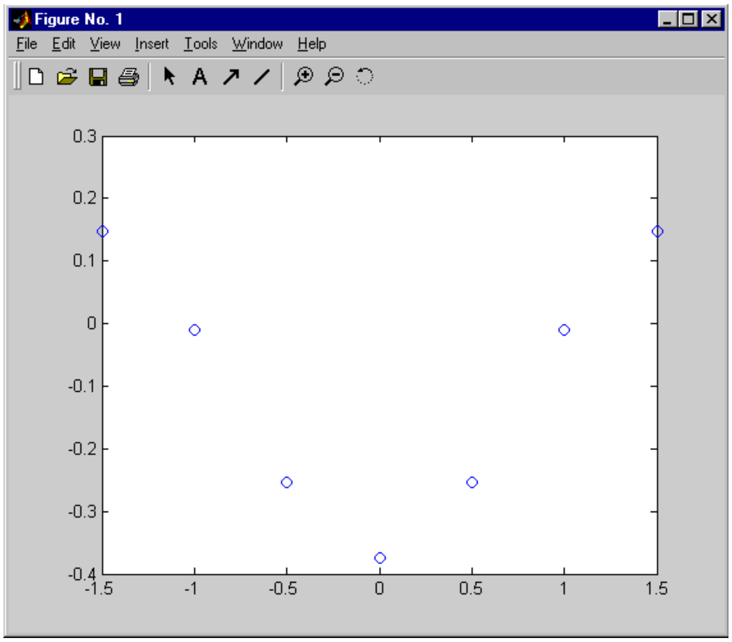
Then implement the following MATLAB session:

```
>> x=-2:.5:2;
>> y=fn(x);
>> Q=quad('fn',-1,1)

Q =
    0.6827

>> Q=quad('fn',-2,2)

Q =
    0.9545
```

Thus, about 68.3% of the area under the curve falls between –1 and 1 and about 95.45% falls between –2 and 2.

(*b*)

```
>> d=diff(y)./diff(x);
>> x=-1.75:.5:1.75;
>> d2=diff(d)./diff(x);
>> x=-1.5:.5:1.5;
>> plot(x,d2,'o')
```




Thus, inflection points ($d^2y/dx^2 = 0$) occur at –1 and 1.

23.15

```
Program Integrate
Use imsl
Implicit None
Integer::irule=1
Real::a=-1.,b=1,errabs=0.0,errrel=0.001
Real::errest,res,f
External f
Call QDAG(f,a,b,errabs,errrel,irule,res,errest)
Print '('' Computed = '',F8.4)',res
Print '('' Error estimate ='',1PE10.3)',errest
End Program

Function f(x)
Implicit None
Real:: x , f
Real::pi
Parameter(pi=3.1415927)
f=1/sqrt(2*pi)*exp(-x**2/2)
End Function

Answers:

x = -1 to 1: Computed =   0.6827 Error estimate = 4.069E-06
x = -2 to 2: Computed =   0.9545 Error estimate = 7.975E-06
x = -3 to 3: Computed =   0.9973 Error estimate = 5.944E-06
```

23.16 MATLAB Script:

```
% Prob2316 Integration program
a=0;
b=pi/2;
integral=quad('ff',a,b)
end
```

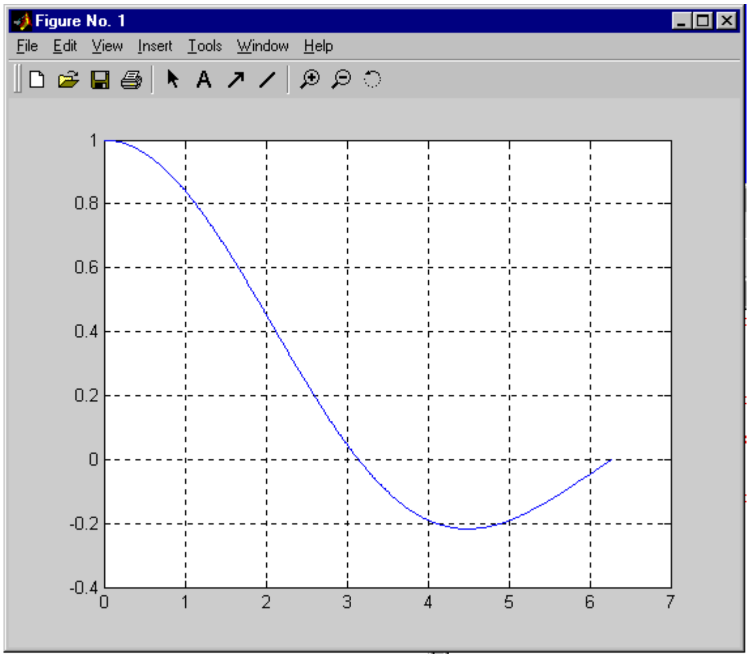
```
function y=ff(x);
y=sin(sin(x));

>> prob2316

integral =
    0.8932
```

23.17 MATLAB Script:

```
%Numerical Integration of sin(t)/t = function sint(t)
%Limits: a=0, b=2pi
%Using the "quad" and "quadl" function for numerical integration
%Plot of function
t=0.01:0.01:2*pi;
y=ff2(t);
plot(t,y); grid
%Integration
format long
a=0.01;
b=2*pi;
Iquad=quad('ff2',a,b)
Iquadl=quadl('ff2',a,b)

function y=ff2(t);
y=sin(t)./t;
```

MATLAB execution:

```
>> prob2317

Iquad =
   1.40815164305082


Iquadl =
   1.40815163168846
```



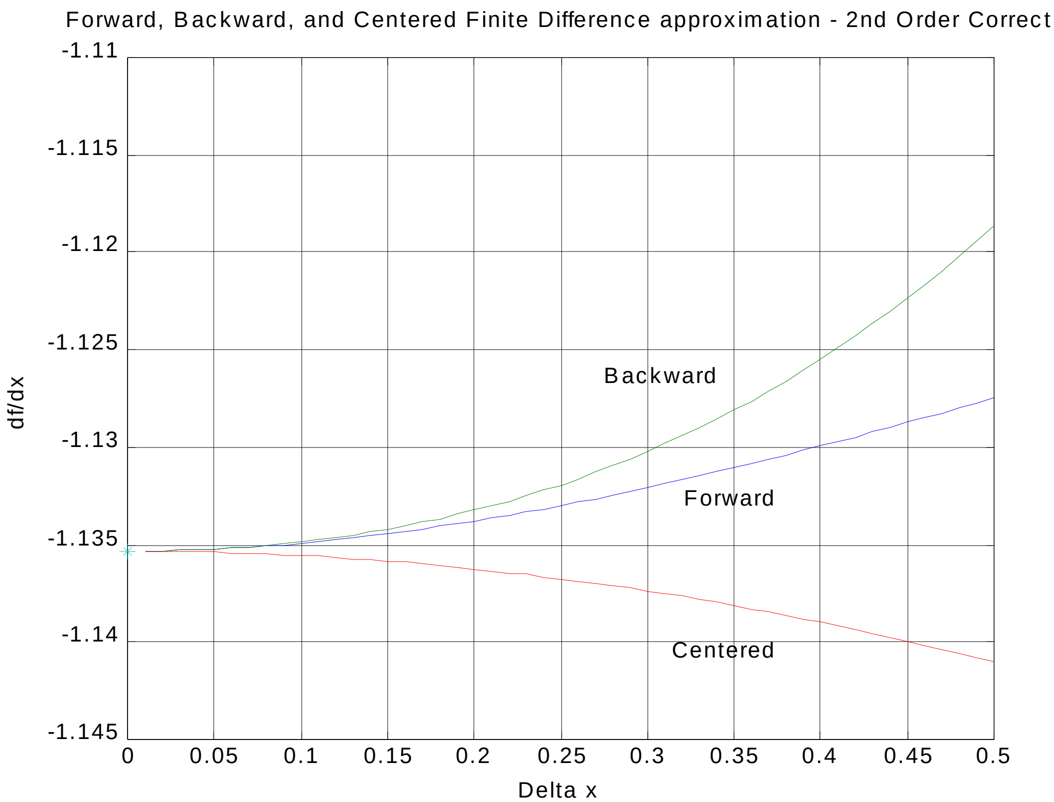

23.18

```
%Centered Finite Difference First & Second Derivatives of Order O(dx^2)
%Using diff(y)

dx=0.5;
y=[1.4 2.1 3.3 4.7 7.1 6.4 8.8 7.2 8.9 10.7 9.8];
dyf=diff(y);

% First Derivative Centered FD using diff
n=length(y);
for i=1:n-2
    dydxc(i)=(dyf(i+1)+dyf(i))/(2*dx);
end

%Second Derivative Centered FD using diff
dy2dx2c=diff(dyf)/(dx*dx);

fprintf('first derivative \n');  fprintf('%f\n', dydxc)
fprintf('second derivative \n'); fprintf('%f\n', dy2dx2c)
```

```
first derivative
1.900000
2.600000
3.800000
1.700000
1.700000
0.800000
0.100000
3.500000
0.900000
second derivative
2.000000
0.800000
4.000000
-12.400000
12.400000
-16.000000
13.200000
0.400000
-10.800000
```

23.19

```
%  Finite Difference Approximation of slope
%  For f(x)=exp(-x)-x
%      f'(x)=-exp(-x)-1
%  Centered diff. df/dx=(f(i+1)-f(i-1))/2dx         + O(dx^2)
%  Fwd. diff.     df/dx=(-f(i+2)+4f(i+1)-3f(i))/2dx + O(dx^2)
%  Bkwd. diff.    df/dx=(3f(i)-4f(i-1)+f(i-2))/2dx  + O(dx^2)


x=2;
fx=exp(-x)-x;
dfdx2=-exp(-x)-1;

%approximation
dx=0.5:-0.01:.01;
for i=1:length(dx)
       %x-values at i+-dx and +-2dx
    xp(i)=x+dx(i);
    x2p(i)=x+2*dx(i);
    xn(i)=x-dx(i);
    x2n(i)=x-2*dx(i);
       %f(x)-values at i+-dx and +-2dx
    fp(i)=exp(-xp(i))-xp(i);
    f2p(i)=exp(-x2p(i))-x2p(i);
    fn(i)=exp(-xn(i))-xn(i);
    f2n(i)=exp(-x2n(i))-x2n(i);
       %Finite Diff. Approximations
    Cdfdx(i)=(fp(i)-fn(i))/(2*dx(i));
     Fdfdx(i)=(-f2p(i)+4*fp(i)-3*fx)/(2*dx(i));
    Bdfdx(i)=(3*fx-4*fn(i)+f2n(i))/(2*dx(i));
end
dx0=0;
plot(dx,Fdfdx,'--',dx,Bdfdx,'-.',dx,Cdfdx,'-',dx0,dfdx2,'*')
 grid
title('Forward, Backward, and Centered Finite Difference approximation - 2nd
Order Correct')
xlabel('Delta x')
ylabel('df/dx')
gtext('Centered'); gtext('Forward');  gtext('Backward')
```



Forward, Backward, and Centered Finite Difference approximation - 2nd Order Correct
-1.11
-1.115
-1.12
-1.125
-1.13
-1.135
-1.14
-1.145
df/dx
0
0.05
0.1
0.15
0.2
0.25
0.3
0.35
0.4
0.45
0.5
Delta x
Backward
Forward
Centered

23.20

FIND: $\dfrac{d^3f}{dx^3}$ CENTERED FINITE-DIFFERENCE APPROXIMATION of $O(\Delta x^2)$

SOLN.: ALL $\Delta x$s ARE EQUAL

FWD <u>TAYLOR'S SERIES EXPANSION ABOUT $a = x_i$ & $x = x_{i+2}$ (2 $\Delta x$ STEPS FWD.)</u>

$$f(x_{i+2}) = f(x_i) + f'(x_i)\,2\Delta x + \tfrac{1}{2} f''(x_i)(2\Delta x)^2 + \tfrac{1}{6} f'''(x_i)(2\Delta x)^3 + \tfrac{1}{24} f^{(4)}_{(x_i)}(2\Delta x)^4 + \tfrac{1}{120} f^{(5)}_{(x_i)}(2\Delta x)^5 + \dots$$

$$f(x_{i+2}) = f(x_i) + 2f'(x_i)\Delta x + 2f''(x_i)\,\Delta x^2 + \tfrac{8}{6} f'''(x_i)\,\Delta x^3 + \tfrac{16}{24} f^{(4)}(x_i)\,\Delta x^4 + \tfrac{32}{120} f^{(5)}(x_i)\,\Delta x^5 + \dots \qquad (1)$$

<u>EXPANSION ABOUT $a = x_i$ $x = x_{i+1}$</u>

$$f(x_{i+1}) = f(x_i) + f'(x_i)\Delta x + \tfrac{1}{2} f''(x_i)\Delta x^2 + \tfrac{1}{6} f'''(x_i)\Delta x^3 + \tfrac{1}{24} f^{(4)}(x_i)\Delta x^4 + \tfrac{1}{120} f^{(5)}(x_i)\Delta x^5 + \dots \qquad (2)$$

$$2f(x_{i+1}) = 2f(x_i) + 2f'(x_i)\Delta x + f''(x_i)\Delta x^2 + \tfrac{2}{6} f'''(x_i)\Delta x^3 + \tfrac{2}{24} f^{(4)}(x_i)\Delta x^4 + \tfrac{2}{120} f^{(5)}(x_i)\Delta x^5 + \dots \qquad 2(2)$$

SUBTRACT 2(2) from (1)

$$f(x_{i+2}) - 2f(x_{i+1}) = -f(x_i) + f''(x_i)\Delta x^2 + \tfrac{6}{6} f'''(x_i)\Delta x^3 + \tfrac{12}{24} f^{(4)}(x_i)\Delta x^4 + \tfrac{30}{120} f^{(5)}(x_i)\Delta x^5 + \dots \qquad (3)$$

BKWD <u>EXPANSION ABOUT $a = x_i$ $x = x_{i-2}$ (2 $\Delta x$ STEPS BKWD.)</u>

$$f(x_{i-2}) = f(x_i) + f'(x_i)(-2\Delta x) + \tfrac{1}{2} f''(x_i)(-2\Delta x)^2 + \tfrac{1}{6} f'''(x_i)(-2\Delta x)^3 + \tfrac{1}{24} f^{(4)}(x_i)(-2\Delta x)^4 + \tfrac{1}{120} f^{(6)}(x_i)(-2\Delta x)^5 + \dots$$

$$f(x_{i-2}) = f(x_i) - 2f'(x_i)\Delta x + 2f''(x_i)\Delta x^2 - \tfrac{8}{6} f'''(x_i)\Delta x^3 + \tfrac{16}{24} f^{(4)}(x_i)\Delta x^4 - \tfrac{32}{120} f^{(6)}(x_i)\Delta x^5 + \dots \qquad (4)$$

<u>EXPANSION ABOUT $a = x_i$ $x = x_{i-1}$</u>

$$f(x_{i-1}) = f(x_i) - f'(x_i)\Delta x + \tfrac{1}{2} f''(x_i)\Delta x^2 - \tfrac{1}{6} f'''(x_i)\Delta x^3 + \tfrac{1}{24} f^{(4)}(x_i)\Delta x^4 - \tfrac{1}{120} f^{(5)}(x_i)\Delta x^5 + \dots \qquad (5)$$

$$2f(x_{i-1}) = 2f(x_i) - 2f'(x_i)\Delta x + f''(x_i)\Delta x^2 - \tfrac{2}{6} f'''(x_i)\Delta x^3 + \tfrac{2}{24} f^{(4)}(x_i)\Delta x^4 - \tfrac{2}{120} f^{(5)}(x_i)\Delta x^5 + \dots \qquad 2(5)$$

SUBTRACT (4) from 2(5)

$$2f(x_{i-1}) - f(x_{i-2}) = f(x_i) - f''(x_i)\Delta x^2 + \tfrac{6}{6} f'''(x_i)\Delta x^3 - \tfrac{12}{24} f^{(4)}(x_i)\Delta x^4 + \tfrac{30}{120} f^{(5)}(x_i)\Delta x^5 + \dots \qquad (6)$$

<u>ADD (3) AND (6)</u>

$$f(x_{i+2}) - 2f(x_{i+1}) + 2f(x_{i-1}) - f(x_{i-2}) = 2f'''(x_i)\Delta x^3 + \tfrac{60}{120} f^{(5)}(x_i)\,\Delta x^5 + \dots$$

$$f'''(x_i) = \frac{f(x_{i+2}) - 2f(x_{i+1}) + 2f(x_{i-1}) - f(x_{i-2})}{2\Delta x^3} - \frac{\left[\tfrac{1}{2} f^{(5)}(x_i)\,\Delta x^5\right]}{2\Delta x^3} + \dots$$

$$\frac{d^3f}{dx}(x_i) = \frac{f(x_{i+2}) - 2f(x_{i+1}) + 2f(x_{i-1}) - f(x_{i-2})}{2\Delta x^3} - \underbrace{\tfrac{1}{4} f^{(5)}(x_i)\,\Delta x^2}_{O(\Delta x^2)} + \dots$$

23.21

**a)**

$$v=\frac{dx}{dt}=x'(t_i)=\frac{x(t_{i+1})-x(t_{i-1})}{2h}=\frac{7.3-5.1}{2}=1.1\ \ m/s$$

$$a=\frac{d^2x}{dt^2}=x''(t_i)=\frac{x(t_{i+1})-2x(t_i)+x(t_{i-1})}{h^2}=\frac{7.3-2(6.5)+5.1}{1^2}=-0.6\ \ m/s^2$$

**b)**

$$v=\frac{-x(t_{i+2})+4x(t_{i+1})-3x(t_i)}{2h}=\frac{-8+4(7.3)-3(6.5)}{2}=0.85\ \ m/s$$

$$a=\frac{-x(t_{i+3})+4x(t_{i+2})-5x(t_{i+1})+2x(t_i)}{h^2}=\frac{-8.4+4(8)-5(7.3)+2(6.3)}{1^2}=-0.3\ \ m/s^2$$

**c)**

$$v=\frac{3x(t_i)-4x(t_{i-1})+x(t_{i-2})}{2h}=\frac{3(6.5)-4(5.1)+3.4}{2}=1.25\ \ m/s$$

$$a=\frac{2x(t_i)-5x(t_{i-1})+4x(t_{i-2})-x(t_{i-3})}{h^2}=\frac{2(6.5)-5(5.1)+4(3.4)-1.8}{1^2}=-0.7\ \ m/s^2$$

23.22

$$\dot{\theta}=\frac{d\theta}{dt}=\frac{\theta(t_{i+1})-\theta(t_{i-1})}{2h}=\frac{0.67-0.70}{2}=-0.015\ \text{rad/s}$$

$$\dot{r}=\frac{dr}{dt}=\frac{r(t_{i+1})-r(t_{i-1})}{2h}=\frac{6030-5560}{2}=235\ \text{ft/s}$$

$$\ddot{\theta}=\frac{d^2\theta}{dt^2}=\frac{\theta(t_{i+1})-2\theta(t_i)+\theta(t_{i-1})}{h^2}=\frac{0.67-2(0.68)+0.70}{(1)^2}=0.01\ \text{rad/s}^2$$

$$\ddot{r}=\frac{d^2r}{dt^2}=\frac{r(t_{i+1})-2r(t_i)+r(t_{i-1})}{h^2}=\frac{6030-2(5800)+5560}{(1)^2}=-10\ \text{ft/s}^2$$

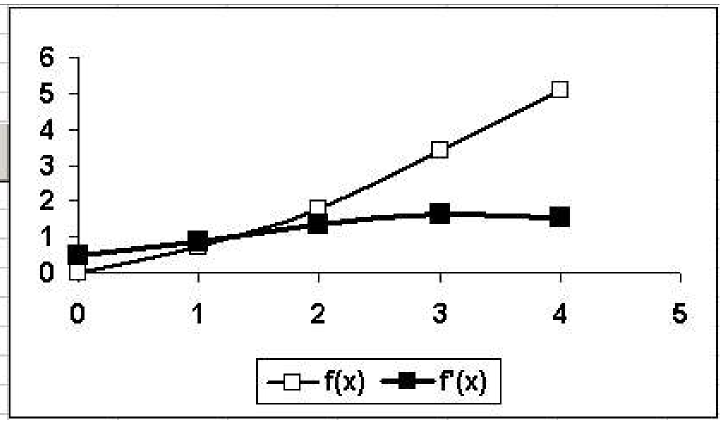
$$\vec{v}=235\,\vec{e}_r-87\,\vec{e}_\theta$$

$$\vec{a}=-8.695\,\vec{e}_r+50.95\,\vec{e}_\theta$$

23.23 Use the same program as was developed in the solution of Prob. 23.11

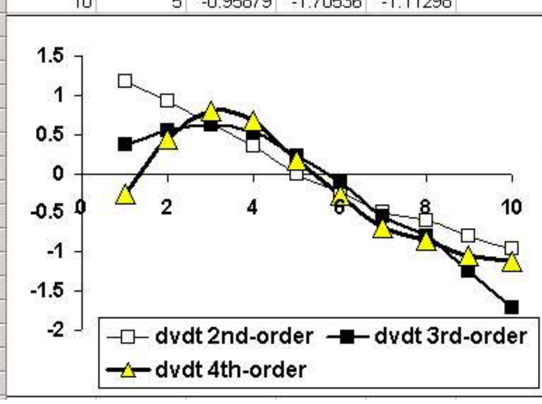
| Prob 23.23 | | |
|---|---|---|
| x | f(x) | f'(x) |
| 0 | 0 | 0.500 |
| 1 | 0.7 | 0.850 |
| 2 | 1.8 | 1.350 |
| 3 | 3.4 | 1.650 |
| 4 | 5.1 | 1.550 |
| 5 | 6.5 | 1.100 |
| 6 | 7.3 | 0.750 |
| 7 | 8 | 0.550 |
| 8 | 8.4 | 0.250 |

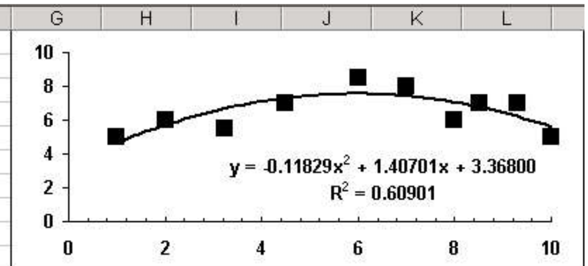
Run

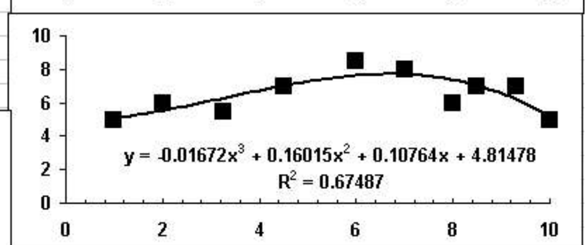


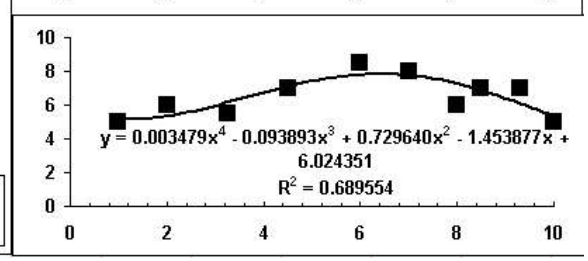

23.24

| | A | B | C | D | E |
|---|---|---|---|---|---|
| 1 | Prob. 23.24 | | | | |
| 2 | | | | | |
| 3 | | | dvdt | dvdt | dvdt |
| 4 | t | v | 2nd-order | 3rd-order | 4th-order |
| 5 | 1 | 5 | 1.17043 | 0.37778 | -0.26236 |
| 6 | 2 | 6 | 0.93385 | 0.5476 | 0.449295 |
| 7 | 3.25 | 5.5 | 0.638125 | 0.6188 | 0.791259 |
| 8 | 4.5 | 7 | 0.3424 | 0.53325 | 0.676979 |
| 9 | 6 | 8.5 | -0.01247 | 0.22368 | 0.167215 |
| 10 | 7 | 8 | -0.24905 | -0.1081 | -0.268 |
| 11 | 8 | 6 | -0.48563 | -0.5402 | -0.6821 |
| 12 | 8.5 | 7 | -0.60392 | -0.79387 | -0.85514 |
| 13 | 9.3 | 7 | -0.79318 | -1.25191 | -1.05156 |
| 14 | 10 | 5 | -0.95879 | -1.70536 | -1.11298 |



## Chapter 24

24.1

$$I = 300\,\frac{\left[0.11454 + 4(0.11904 + 0.132 + 0.15024) + 2(0.12486 + 0.14046) + 0.16134\right]}{18}$$

$= 40.194$ note that using only 3 points also gives exact result because $C(T)$ is 2nd order

$\Delta H = 40.194 \times 1500 = 60{,}291$

24.2

$\varepsilon_a = -2.9\%$ $\quad$ $\varepsilon_a = 0$

| | | |
|---|---|---|
| 41.382 | 40.194 | 40.194 |
| 40.491 | 40.194 | |
| 40.26825 | | |

note exact answer occurs in second column as expected

$\Delta H = 40.194 \times 1500 = 60{,}291$

24.3

$$x = \frac{150 + (-150)}{2} + \frac{150 - (-150)}{2}\,x_d$$

$= 150\,x_d$

$dx = 150\,dx_d$

2 point formula gives

| $x$ | $f(x)$ |
|---|---|
| −86.60254 | 18.0705 |
| 86.60254 | 22.1235 |

$I = 18.0705 + 22.1235$

$= 40.194$

3 point formula gives

$I = (17.61576 + 23.05343) \times (0.5555556) + 19.8 \times (0.8888889)$

$= 40.19395$

both exact because errors are

$E_t \propto f^{(4)}(\xi)$ for 2 point

$E_t \propto f^{6}(\xi)$ for 3 point

which are zeros for 2nd order polynomial

24.4 use Simpson's $\frac{1}{3}$

$$I = (50-0)\,\frac{\left[10 + 4(22+47+58+40+32) + 2(35+55+52+37) + 34\right]}{30}$$

$= 1996.667$

$M = 4\,\frac{m^3}{min} \times 1996.667\ \frac{mg\ min}{m^3} = 7986.667$ mg

24.5 Use combination $\frac{1}{3}$ and $\frac{3}{8}$ Simpson's Rules

$$I = (8-0)\left[\frac{10+4(20+40)+2(30)+60}{12}\right]$$

$$+ (20-8)\left[\frac{60+3(72+70)+50}{8}\right]$$

$$= 246.6667 + 804 = 1050.6667$$

$$M = 1050.6667 \times 12 = 12608$$

24.6 Use Equation 23.9

| | $x$ | $f(x)$ |
|---|---|---|
| $x_{i-1}$ | 0 | 0.1 |
| $x_i$ | 1 | 0.4 |
| $x_{i+1}$ | 3 | 0.9 |

at $x = 0$

$$f'(0) = 0.1\frac{(-1-3)}{(0-1)(0-3)} + 0.4\frac{(-3)}{(1-0)(1-3)}$$

$$+ 0.9\frac{(-1)}{(3-0)(3-1)}$$

$$= -0.1333 + 0.6 - 0.15$$
$$= 0.3167 \times 10^{-6} \frac{gm}{cm4}$$

$$\text{mass flux} = 0.3167 \times 10^{-6} \times 2 \times 10^{-6}$$
$$= 0.6334 \times 10^{-12} \frac{g}{cm^2 s}$$

$$\text{mass} = 0.6334 \times 10^{-12} (3 \times 10^{6}) \times 10^{4} \frac{cm^2}{m^2} \times 60 \times 60 \times 24 \times 365$$
$$= 5.9925 \times 10^{5} \ g/yr$$

24.7

at $t = 0$ use $Oh^2$ forward divided difference formula

$$f'(0) = \frac{-0.73 + 4(0.65) - 3(0.5)}{30}$$

$$= 0.0123$$

use centered divided difference formula at $t = 15, 30, 45$, and 90

$$f'(15) = \frac{0.73 - 0.5}{30} = 0.0076667$$

$$f'(30) = \frac{0.88 - 0.65}{30} = 0.0076667$$

$$f'(45) = \frac{1.03 - 0.73}{30} = 0.01$$

$$f'(90) = \frac{1.3 - 1.03}{60} = 0.0045$$

use backward divided difference formula at $t = 120$

$$f'(120) = \frac{3(1.3) - 4(1.14) + 1.03}{60}$$

$$= 0.0061667$$

Use Equation 23.9 at $t = 60$

| | $x$ | $f(x)$ |
|---|---|---|
| $x_{i-1}$ | 45 | 0.88 |
| $x_i$ | 60 | 1.03 |
| $x_{i+1}$ | 90 | 1.14 |

$$f'(60) = 0.88\,\frac{(120-60-90)}{(45-60)(45-90)}$$

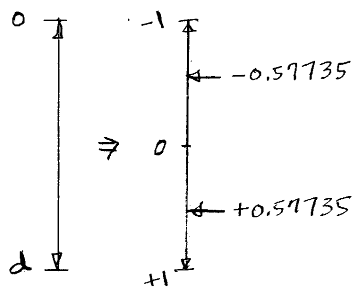
$$+\ 1.03\,\frac{(120-45-90)}{(60-45)(60-90)}$$

$$+\ 1.14\,\frac{(120-45-60)}{(90-45)(90-60)}$$

$$= -0.03911 + 0.03433 + 0.01267$$

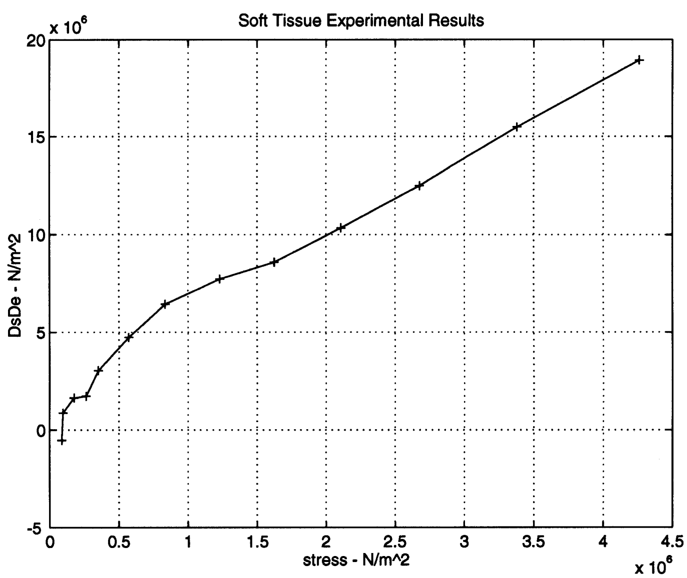
$$= 0.00789$$

24.8 Use two point Gauss Quadrature

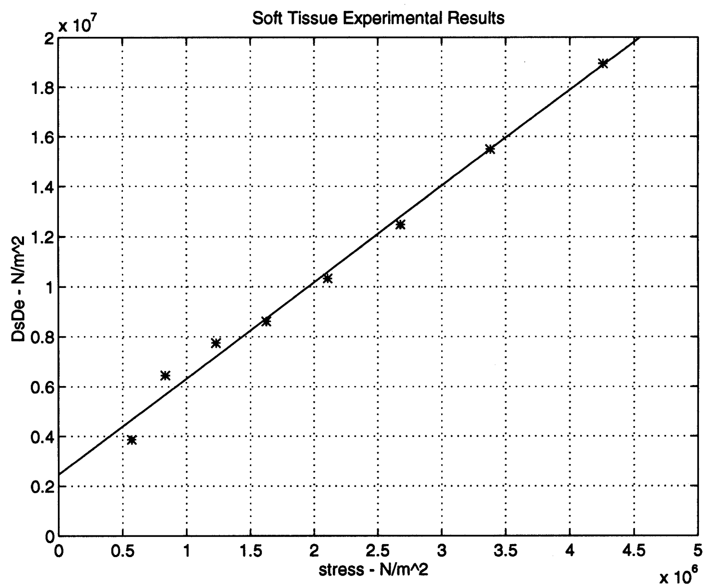



which is 21.1 % d and
78.9 % d

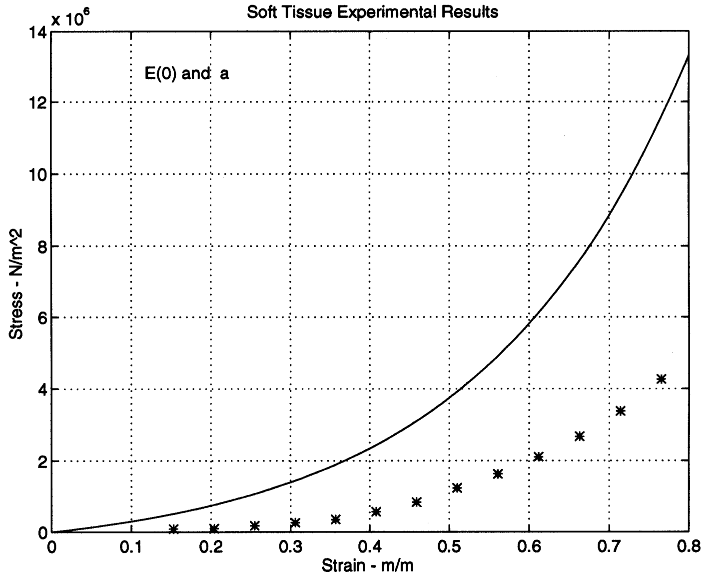
24.9

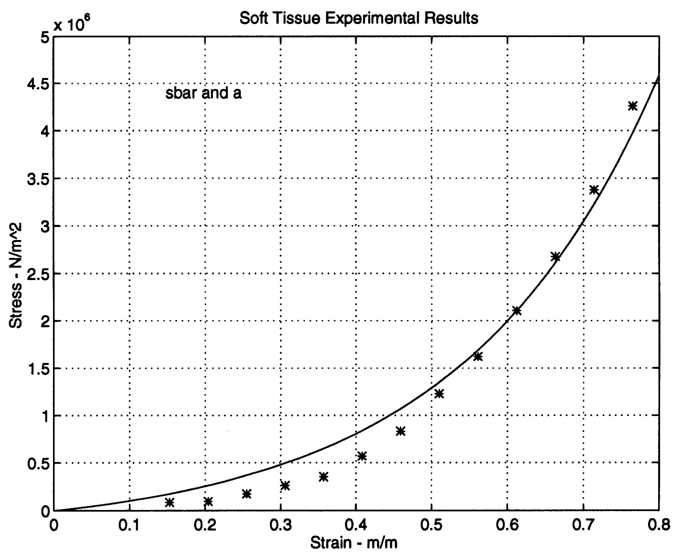


```
    % Raw data input
s=[87.8 96.6 176 263 351 571 834 1229 1624 2107 2678 3380 4258]*1e+3;
e=[153 204 255 306 357 408 459 510 561 612 663 714 765]*1e-3;
de=51e-3; dde=2*de;
    % Finite Differences
dsde(1)=  (-s(3)+4*s(2)-3*s(1))/dde;    %  forward difference
for i=2:12
     dsde(i)=(s(i+1)-s(i-1))/dde;       %   centered difference
end
dsde(13)=(3*s(13)-4*s(12)+s(11))/dde;  %   backward difference

plot(s,dsde,'-',s,dsde,'+')
title('Soft Tissue Experimental Results')
xlabel('stress - N/m^2 '); ylabel('DsDe - N/m^2'); grid
```



```
   %Raw data input
s=[87.8 96.6 176 263 351 571 834 1229 1624 2107 2678 3380 4258]*1e+3;
e=[153 204 255 306 357 408 459 510 561 612 663 714 765]*1e-3;

   %Regression analysis
   %Elimination of early data
idx=5;  % idx=starting point for data exclusion (points with subscribt above idx will be included in s)
        % With this data the range idx can be    idx=3 to idx=8
np=length(s)-idx;
for i=1:np
    sr(i)=s(idx+i);     %sr = regression values for s
end
   %Constants
de=51e-3; dde=2*de;
   % Finite difference
dsder(1)= (-sr(3)+4*sr(2)-3*sr(1))/dde;          %  forward difference
for i=2:np-1
   dsder(i)=(sr(i+1)-sr(i-1))/dde;               %  centered difference
end
dsder(np)=(3*sr(np)-4*sr(np-1)+sr(np-2))/dde;    %  backward difference
   %Linear Fit
c1=polyfit(sr,dsder,1);
a=c1(1);  Eo=c1(2);
sp=0:1e6:5e6;
dsde1=polyval(c1,sp);
plot(sp,dsde1,sr,dsder,'*')
  title('Soft Tissue Experimental Results')
  xlabel('stress - N/m^2 '); ylabel('DsDe - N/m^2');
  axis([0 5e6 0 20e6]); grid; pause

      % Stress-Strain Curve Plot
      % Plot the analytic expression for s vs e
         % Using Eo and a
 ep=0:.005:0.8;              %  ep=curve plot value of e
 sp=(Eo/a)*(exp(a*ep)-1);    %  sp=curve plot value of s
 plot(ep,sp,e,s,'*')
   title(' Soft Tissue Experimental Results');
   xlabel('Strain - m/m'); ylabel('Stress - N/m^2')
   grid; gtext('E(0) and  a'); pause
       % Using sStar and eStar
 sStar=s(10); eStar=e(10);
 sbar=sStar/(exp(a*eStar)-1);
 sp2=sbar*(exp(a*ep)-1);
 plot(ep,sp2,e,s,'*')
   title(' Soft Tissue Experimental Results');
   xlabel('Strain - m/m'); ylabel('Stress - N/m^2')
   grid; gtext('sbar and a');
```




24.10

| Time After Injection (sec) | Semilog Dye Concentration | Constant | Product |
|---|---|---|---|
| 9 | 0.11 | 1 | 0.11 |
| 9.5 | 0.14 | 2 | 0.28 |
| 10 | 0.18 | 2 | 0.36 |
| 10.5 | 0.25 | 2 | 0.5 |
| 11 | 0.4 | 2 | 0.8 |
| 11.5 | 0.7 | 2 | 1.4 |
| 12 | 1.4 | 2 | 2.8 |
| 12.5 | 2.4 | 2 | 4.8 |
| 13 | 4 | 2 | 8 |
| 13.5 | 5.5 | 2 | 11 |
| 14 | 6.85 | 2 | 13.7 |
| 14.5 | 8 | 2 | 16 |

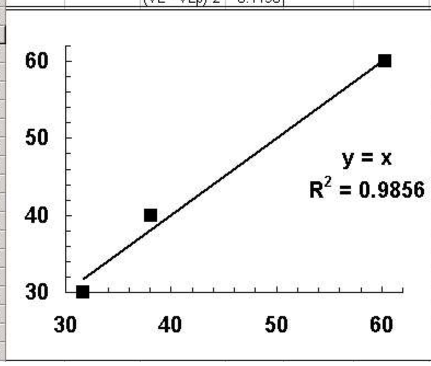
| | | | |
|---|---|---|---|
| 15 | 9 | 2 | 18 |
| 15.5 | 9.35 | 2 | 18.7 |
| 16 | 9.2 | 2 | 18.4 |
| 16.5 | 8.7 | 2 | 17.4 |
| 17 | 7.95 | 2 | 15.9 |
| 17.5 | 7 | 2 | 14 |
| 18 | 5.95 | 2 | 11.9 |
| 18.5 | 4.85 | 2 | 9.7 |
| 19 | 4.1 | 2 | 8.2 |
| 19.5 | 3.5 | 2 | 7 |
| 20 | 3 | 2 | 6 |
| 20.5 | 2.55 | 2 | 5.1 |
| 21 | 2.2 | 2 | 4.4 |
| 21.5 | 1.8 | 2 | 3.6 |
| 22 | 1.5 | 2 | 3 |
| 22.5 | 1.3 | 2 | 2.6 |
| 23 | 1.1 | 2 | 2.2 |
| 23.5 | 0.9 | 2 | 1.8 |
| 24 | 0.8 | 2 | 1.6 |
| 24.5 | 0.64 | 2 | 1.28 |
| 25 | 0.55 | 2 | 1.1 |
| 25.5 | 0.47 | 2 | 0.94 |
| 26 | 0.4 | 2 | 0.8 |
| 26.5 | 0.34 | 2 | 0.68 |
| 27 | 0.29 | 2 | 0.58 |
| 27.5 | 0.24 | 2 | 0.48 |
| 28 | 0.2 | 2 | 0.4 |
| 28.5 | 0.16 | 2 | 0.32 |
| 29 | 0.14 | 2 | 0.28 |
| 29.5 | 0.125 | 2 | 0.25 |
| 30 | 0.1 | 1 | 0.1 |
| | **Sum of Products** | **=** | **236.46** |
| | **Trapezoidal Approximation** | **=** | **59.115** |

Cardiac Output = [56 mg/59.115 mg*sec/L]*60 = **5.68 L/min**

24.11 The following Excel Solver application can be used to estimate: $k$ = 0.09915 and $A$ =6.98301.

L20 = =D18+H18+L18

| | A | B | C | D | E | F | G | H | I | J | K | L |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | **k** | **0.009915** | | | | | | | | | | |
| 2 | **A** | **6.983009** | | | | | | | | | | |
| 3 | | | | | | | | | | | | |
| 4 | Patient | A | | | Patient | B | | | Patient | C | | |
| 5 | Age | 65 | | | | 43 | | | | 80 | | |
| 6 | VL | 60 | | | | 40 | | | | 30 | | |
| 7 | age(yrs) | P (mmHG) | P - 13 | Trap | Age (yrsr) | P (mmHG) | P - 13 | Trap | Age (yrsr) | P (mmHG) | P - 13 | Trap |
| 8 | 25 | 13 | 0 | | 25 | 11 | -2 | | 25 | 13 | 0 | |
| 9 | 40 | 15 | 2 | 15 | 40 | 30 | 17 | 112.5 | 40 | 14 | 1 | 7.5 |
| 10 | 50 | 22 | 9 | 55 | 41 | 32 | 19 | 18 | 50 | 15 | 2 | 15 |
| 11 | 60 | 23 | 10 | 95 | 42 | 33 | 20 | 19.5 | 60 | 17 | 4 | 30 |
| 12 | 65 | 24 | 11 | 52.5 | 43 | 35 | 22 | 21 | 80 | 19 | 6 | 100 |
| 13 | | | | | | | | | | | | |
| 14 | | | Integral | 217.5 | | | Integral | 171 | | | Integral | 152.5 |
| 15 | | | | | | | | | | | | |
| 16 | | | VLp | 60.3459 | | | VLp | 38.0547 | | | VLp | 31.6769 |
| 17 | | | | | | | | | | | | |
| 18 | | | (VL - VLp)^2 | 0.1196 | | | (VL - VLp)^2 | 3.7844 | | | (VL - VLp)^2 | 2.8120 |
| 19 | | | | | | | | | | | | |
| 20 | | | | | | | | | | | **SumSq** | **6.7159** |




24.12 The following Excel spreadsheet is set up to use (left) a combination of the trapezoidal and Simpsons rules and (right) just the trapezoidal rule:

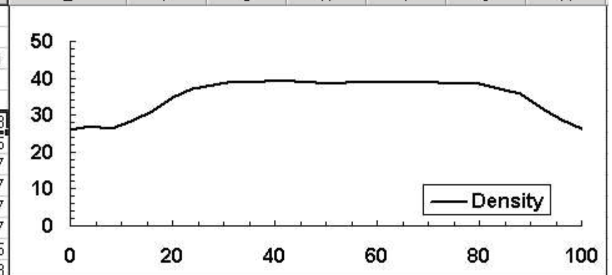
C20 = =C16*24*10

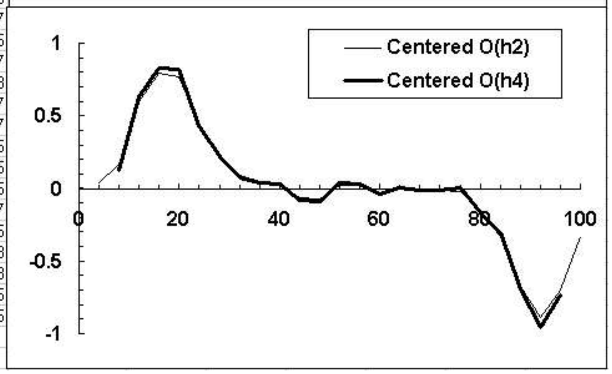
| | A | B | C | D | E | F | G | H |
|---|---|---|---|---|---|---|---|---|
| 1 | **Prob 24.12** | | | | | | | |
| 2 | | | | | | | | |
| 3 | Simp1/3-3/8-Trap | | | | | Trap | | |
| 4 | Time | Flux | Integral | Rule | | Time | Flux | Trapezoidal |
| 5 | 0 | 15 | | | | 0 | 15 | |
| 6 | 1 | 14 | | | | 1 | 14 | 14.5 |
| 7 | 2 | 12 | 27.66667 | 1/3 rule | | 2 | 12 | 13 |
| 8 | 3 | 11 | | | | 3 | 11 | 11.5 |
| 9 | 4 | 9 | | | | 4 | 9 | 10 |
| 10 | 5 | 8 | 30 | 3/8 rule | | 5 | 8 | 8.5 |
| 11 | 10 | 5 | | | | 10 | 5 | 32.5 |
| 12 | 15 | 2.5 | | | | 15 | 2.5 | 18.75 |
| 13 | 20 | 2 | 60.9375 | 3/8 rule | | 20 | 2 | 11.25 |
| 14 | 24 | 1 | 6 | Trap rule | | 24 | 1 | 6 |
| 15 | | | | | | | | |
| 16 | | Integral | 124.6042 | | | | | 111.5 |
| 17 | | | | | | | | |
| 18 | | Average | 5.19184 | | | | Average | 4.64583333 |
| 19 | | | | | | | | |
| 20 | Mass delivered | | **29905** | | | | | **26760** |

24.13

D6 = =(-B8+8*B7-8*B5+B4)/12/(A6-A5)

| | A | B | C | D |
|---|---|---|---|---|
| 1 | **Prob. 24.13** | | | |
| 2 | | | | |
| 3 | Distance | Density | Centered O(h2) | Centered O(h4) |
| 4 | 0 | 26.035 | | |
| 5 | 4 | 26.968 | 0.038 | |
| 6 | 8 | 26.339 | 0.16975 | 0.121270833 |
| 7 | 12 | 28.326 | 0.592375 | 0.6293125 |
| 8 | 16 | 31.078 | 0.793375 | 0.830729167 |
| 9 | 20 | 34.673 | 0.77025 | 0.819729167 |
| 10 | 24 | 37.24 | 0.45025 | 0.433479167 |
| 11 | 28 | 38.275 | 0.230875 | 0.217541667 |
| 12 | 32 | 39.087 | 0.0915 | 0.0758125 |
| 13 | 36 | 39.007 | 0.04625 | 0.042833333 |
| 14 | 40 | 39.457 | 0.0215 | 0.031729167 |
| 15 | 44 | 39.179 | -0.064625 | -0.0780625 |
| 16 | 48 | 38.94 | -0.070125 | -0.086729167 |
| 17 | 52 | 38.618 | 0.024 | 0.039708333 |
| 18 | 56 | 39.132 | 0.023875 | 0.032291667 |
| 19 | 60 | 38.809 | -0.02675 | -0.039166667 |
| 20 | 64 | 38.918 | -0.002875 | 0.00275 |
| 21 | 68 | 38.786 | -0.01275 | -0.013875 |
| 22 | 72 | 38.816 | -0.015875 | -0.0154375 |
| 23 | 76 | 38.659 | -0.021625 | 0.001791667 |
| 24 | 80 | 38.643 | -0.167875 | -0.1645 |
| 25 | 84 | 37.316 | -0.334375 | -0.305645833 |
| 26 | 88 | 35.968 | -0.67325 | -0.693333333 |
| 27 | 92 | 31.93 | -0.891625 | -0.9595 |
| 28 | 96 | 28.835 | -0.70275 | -0.7328125 |
| 29 | 100 | 26.308 | -0.3335 | |
| 30 | 104 | 26.167 | | |




The extremes for both cases are at 16 and 92

24.14 $\int_0^{30} 200\, z \left[\frac{z}{5+z}\right] e^{-z/15}\, dz$

| | | | |
|---|---|---|---|
| 10440.15 | 20034.62 | 19520.57 | 19345.74 |
| 17636.01 | 19552.69 | 19348.47 | |
| 19073.52 | 19361.24 | | |
| 19289.31 | | | |

$$\int_0^{30} 200\left(\frac{3}{5+z}\right)e^{-z/15}\,dz$$

348.005

1001.731

1320.585

1435.007

1219.64

1426.87

1473.148

1440.685

1476.233

1476.797

$$d = \frac{19345.74}{1476.797} = 13.0998$$

$$V = \frac{1476.797\,(13.0998)}{3}$$

$$= 6448.58$$

$$T = \frac{6448.58}{0.995} = 6480.985$$

$$H = 1476.797 - 648.985\,(0.0995)$$

$$= 831.94$$

24.15 Use 5 point Gauss Quadrature

$$\int_0^{30} 200\, z \left(\frac{z}{5+z}\right) e^{-z/15}\, dz$$

$I = 0.2369269(8144.2578 + 10852.84)$

$+ 0.4786287(7601.172 + 12217.47)$

$+ 0.5688889(12415.93)$

$= 19320.41$

$$\int_0^{30} 200 \left(\frac{z}{5+z}\right) e^{-z/15}\, dz$$

$= 0.2369269(599.9124 + 379.5668)$

$+ 0.4786287(1097.966 + 529.4212)$

$+ 0.5688889(827.7288)$

$= 1481.865$

$d = 19320.41 / 1481.865 = 13.038$

$V = 1481.865(13.038)/3 = 6440.185$

$T = 6440.185 / 0.995 = 6472.548$

$H = 1481.865 - 6472.548(0.0995)$

$= 837.846$

24.16

Trapezoidal Rule gives

$$I = (30-0)\frac{[0 + 2(68.92 + 57.48 + 39.84 + 26.03 + 16.55) + 10.37]}{12}$$

$= 1070.057$

Simpsons 1/3 Rule

$$I = (30-0)\frac{[0 + 4(68.92 + 39.84 + 16.55) + 2(57.48 + 26.03) + 10.37]}{18}$$

$= 1131.098$

Simpson's 3/8 Rule

$$I = (30-0)\frac{[0 + 3(68.92 + 57.48 + 26.02 + 16.55) + 2(39.84) + 10.37]}{16}$$

$= 1119.381$

24.17

With $h=4$, (Trap Rule)

$$I = 20\left[\frac{0+2(2+4+4+3.4)+0}{10}\right]$$

$= 53.6$

With $h=2$, (Trap Rule)

$$I = 20\,\frac{\left[0+2(1.8+2+4+4+6+4+3.6+3.4+2.8)+0\right]}{20}$$

$= 63.2$

With $h=2$, (Simp 1/3 Rule)

$$I = 20\,\frac{\left[0+4(1.8+4+6+3.6+2.8)+2(2+4+4+3.4)+0\right]}{30}$$

$= 66.4$

24.18

| Distance (N to S) | Ordinate (W to E) |
|---|---|
| 0 | 0 |
| 200 | 600 |
| 400 | 880 |
| 600 | 950 |
| 800 | 1100 |
| 1000 | 1600 |
| 1200 | 1900 |
| 1400 | 2100 |
| 1600 | 2300 |
| 1800 | 2600 |
| 2000 | 2950 |
| 2200 | 3150 |
| 2400 | 3200 |
| 2600 | 3200 |
| 2800 | 3150 |
| 3000 | 3000 |
| 3200 | 2800 |
| 3400 | 2700 |
| 3600 | 2600 |
| 3800 | 2525 |
| 4000 | 2525 |
| 4200 | 2430 |

Use Simp 1/3 Rule on first 18 segments and Simp 3/8 Rule on last 3 segments

$$I = (3600-0)\frac{\left[0+4(600+950+1600+2100+2600+3150+3200+3000+2700)+2(880+1100+1900+2300+2950+3200+3150+2800+2600)\right]}{54}$$

$= 7917333.33$

$$I = (4200-3600)\frac{\left[2600+3(2525+2525)+2430\right]}{8}$$

$= 1513500$

TOTAL $= 9{,}430{,}833$ $ft^2$

24.19 Use combination of Trapezoidal and Simpsons Rule because data are unequally spaced

| | Time after midnight | | Rate |
|---|---|---|---|
| $I_1$ | 0 | 1/3 Rule | 2 |
| | 2 | | 2 |
| | 4 | | 0 |
| $I_2$ | 5 | Trap Rule | 2 |
| | 6 | | 5 |
| | 7 | | 8 |
| $I_3$ | 7.5 | trap | 25 |
| | 8 | | 12 |
| | 8.5 | | 5 |
| $I_4$ | 10 | Trap | 10 |
| $I_5$ | 12.5 | Trap | 12 |
| $I_6$ | 14 | Trap | 7 |
| $I_7$ | 16 | Trap | 9 |
| $I_8$ | 17 | simp $\frac{1}{3}$ | 28 |
| | 18 | | 22 |
| | 19 | | 10 |
| | 20 | | 9 |
| | 21 | | 11 |
| | 22 | | 8 |
| | 23 | | 9 |
| | 24 | | 3 |

$$I_1 = (4-0)\left[\frac{2+4(2)+0}{6}\right] = 6.67$$

$$I_2 = (7-4)\left[\frac{8+2(5+2)+0}{6}\right] = 11$$

Note could also use Simp 3/8 Rule which gives $I = 10.875$

$$I_3 = (8.5-7)\left[\frac{5+2(12+25)+8}{6}\right] = 21.75$$

m Simp 3/8 Rule

$$I_4 = (10-8.5)\left(\frac{10+5}{2}\right) = 11.25$$

$$I_5 = (12.5-10)\left(\frac{12+10}{2}\right) = 27.5$$

$$I_6 = (14-12.5)\left(\frac{7+12}{2}\right) = 14.25$$

$$I_7 = (16-14)\left(\frac{9+7}{2}\right) = 16.$$

$$I_8 = (24-16)\frac{\left[9+4(28+10+11+9) + 2(22+9+8)+3\right]}{24}$$

$$= 107.33$$

$$I_{total} = 215.75 \ \frac{\text{cars}}{\text{min}}\text{hr}$$

$$C = 215.75 \ \frac{\text{cars hr}}{\text{min}} \times 1440$$

$$C = 310896 \text{ cars/day}$$

24.20

| $l$ | $F(l)$ | $l\,F(l)$ |
|---|---|---|
| 0 | 0 | 0 |
| 30 | 350 | 10,500 |
| 60 | 1000 | 60,000 |
| 90 | 1500 | 135,000 |
| 120 | 2600 | 312,000 |
| 150 | 3000 | 450,000 |
| 180 | 3300 | 594,000 |
| 210 | 3500 | 735,000 |
| 240 | 3600 | 864,000 |

Use 4 applications of Simpsons 1/3 Rule

$$I = (240-0)\frac{\left[0+4(10{,}500+135000+450000+735{,}000)+2(60{,}000+312{,}000+594{,}000)+864000\right]}{24}$$

$$= 73{,}404{,}000$$

$$I = (240-0)\frac{\left[0+4(350+1500+3000+3500)+2(1000+2600+3300)+3600\right]}{24}$$

$$I = 508{,}000$$

$$d = \frac{73{,}404{,}000}{508{,}000} = 144.496 \text{ m}$$

24.21

| $z$ | $w(z)$ | $\rho g w(z)(60-z)$ (a) | $z \times (a)$ |
|---|---|---|---|
| 60 | 200 | 0 | 0 |
| 50 | 190 | $1.862 \times 10^7$ | $9.31 \times 10^8$ |
| 40 | 175 | $3.43 \times 10^7$ | $1.372 \times 10^9$ |
| 30 | 160 | $4.704 \times 10^7$ | $1.4112 \times 10^9$ |
| 20 | 135 | $5.292 \times 10^7$ | $1.0584 \times 10^9$ |
| 10 | 130 | $6.37 \times 10^7$ | $6.37 \times 10^8$ |
| 0 | 122 | $7.1736 \times 10^7$ | 0 |

$$f_t = 60\,\frac{\left[7.1736+4(6.37+4.704+1.862)+2(5.292+3.43)+0\right]}{18} \times 10^7$$

$$= 2.54539 \times 10^9 \text{ N}$$

$$I = (60-0)\frac{\left[0+4(6.37+14.112+9.31)+2(10.584+13.72)+0\right]\times 10^8}{24}$$

$$= 5.59253 \times 10^{10} \text{ Nm}$$

$$d = \frac{5.59253 \times 10^{10}}{2.54539 \times 10^9} = 21.97 \text{ m}$$

24.22 assume
1 year = 12 30day months

also assume flow on 1 Jan and 31 Dec is average of mid Jan and Dec flows, that is,

$$\text{Flow (1 Jan = 31 Dec)} = \frac{31+27}{2} = 29$$

| Day | Flow |
|---|---|
| 0 | 29 |
| 15 | 31 |
| 45 | 37 |
| 75 | 80 |
| 105 | 119 |
| 165 | 102 |
| 255 | 20 |
| 285 | 21 |
| 315 | 23 |
| 345 | 27 |
| 360 | 29 |

$$I = (15-0)\left[\frac{31+29}{2}\right] = 450$$

$$+ (105-15)\left[\frac{31+2(37+80)+119}{6}\right] = 5760$$

$$+ (165-105)\left[\frac{102+119}{2}\right] = 6630$$

$$+ (255-165)\left[\frac{20+102}{2}\right] = 5490$$

$$+ (345-255)\left[\frac{20+2(21+23)+27}{6}\right] = 2025$$

$$+ (360-345)\left[\frac{29+27}{2}\right] = 420$$

$$\text{TOTAL} = 20{,}775$$

$$20{,}775 \frac{m^3}{s} \times 60 \times 60 \times 24 \times 365 = 6.5516 \times 10^{11} \ m^3/yr$$

24.23 $H = e_{ab} A \int_0^t q(t)\,dt$

Use Trapezoidal Rule

$$\int_0^{14} q(t)\,dt = (14-0)\left[\frac{0.10 + 2(1.62+5.32+6.29+7.80+8.81+8.00+8.57+8.03+7.04+6.27+5.56+3.54+1) + 0.20}{28}\right]$$

$$= 78 \ cal/cm^2$$

$$H = 78 \times 0.45 \times 150000 = 5265000 \ cal$$

24.24 Use 3 point Forward Difference approximation for $f'(0) = \left.\frac{dT}{dx}\right|_{x=0}$

$$f'(0) = \frac{-15 + 4(17) - 3(20)}{2(0.1)} = -35 \ \frac{°C}{m}$$

$$60\left(\frac{W}{m^2}\right) = -k\left(\frac{W}{°C\,m}\right)\left(-35\right)\left(\frac{°C}{m}\right)$$

$$k = 1.7143 \ \frac{W}{°C\,m}$$

24.25 $\int_0^1 i^2(t)\,dt$

$$= 0.2369269(0.02064727 + 0.08038712) + 0.4786287(0.5547844 + 1.244244) + 0.5688889(1.889466)$$

$$= 1.959901$$

and $I_{RMS} = 1.399965$

24.26

$$I = (1/2 - 0)\left[\frac{0 + 4(1.315045 + 6.677329 + 7.557865 + 3.664595 + 0.3960831) + 2(4.095143 + 7.942476 + 5.883931 + 1.66496) + 0}{30}\right]$$

$$= 1.960278$$

$$I_{RMS} = 1.4000993$$

24.27

$$\overbrace{\quad}^{\epsilon_a = 0.85\%}$$

| | | | |
|---|---|---|---|
| 0 | 2.519288 | 1.943064 | 1.959828 |
| 1.889466 | 1.979078 | 1.959566 | |
| 1.956675 | 1.960785 | | |
| 1.959758 | | | |

$I = 1.959828$

$I_{RMS} = 1.39999386$

24.28

| $t$ | $i$ | $di/dt$ | $V_L$ |
|---|---|---|---|
| 0 | 0 | 1.5 | 6 |
| 0.1 | 0.15 | 1.5 | 6 |
| 0.2 | 0.30 | 2.0 | 8 |
| 0.3 | 0.55 | 2.35 | 9.4 |
| 0.5 | 0.8 | 3.375 | 13.5 |
| 0.7 | 1.9 | 7.625 | 30.5 |

$$i'(0) = \frac{-0.3 + 4(0.15) - 3(0)}{2(0.1)}$$

$$= 1.5$$

$$i'(0.7) = \frac{3(1.9) - 4(0.8) + 0.55}{2(0.2)}$$

$$= 7.625$$

$$i'(0.3) = 0.3\,\frac{2(0.3) - 0.3 - 0.5}{(0.2-0.3)(0.2-0.5)} + 0.55\,\frac{2(0.3) - 0.2 - 0.5}{(0.3-0.2)(0.3-0.5)} + 0.8\,\frac{2(0.3) - 0.2 - 0.3}{(0.5-0.2)(0.5-0.3)}$$

$$= 2.35$$

24.29

$$\overline{V} = \frac{1}{60}\int_0^{60} i(t)\cdot R(i)\,dt$$

$$I = (60-0)\left[\frac{1.313\times10^{8} + 4\left(8.27\times10^{7} + 3.03\times10^{7} + 7.88\times10^{6} + 1.1\times10^{6} + 18000\right) + 2\left(5.32\times10^{7} + 1.62\times10^{7} + 3.32\times10^{6} + 2.37\times10^{5}\right) + 0}{30}\right]$$

$$= 1.57484\times10^{9}$$

$$\overline{V} = 2.625\times10^{7}$$

24.30

| $x$ | $F(x)$ | $\theta(x)$ | $F(x)\cos\theta(x)$ |
|---|---|---|---|
| 0 | 0 | 0.5 | 0 |
| 5 | 6.5 | 1.4 | 1.104786 |
| 10 | 11 | 0.75 | 8.048578 |
| 15 | 13.5 | 0.90 | 8.391735 |
| 20 | 14 | 1.3 | 3.744984 |
| 25 | 12.5 | 1.48 | 1.133395 |
| 30 | 9 | 1.50 | 0.636635 |

$$I = (30-0)\left[\frac{0 + 4(1.104786 + 8.391735 + 1.133395) + 2(8.048578 + 3.744984) + 0.636635}{18}\right]$$

$I = 111.239$

24.31

$$W = \int_0^{30} (1.5x - 0.04x^2)$$

$$\cos(0.8 + 0.125x - 0.009x^2 + 0.0002x^3)\,dx$$

$$I = (30-0)\frac{[0 + 2(2.2717 + 3.284938 + 3.162033) - 2.48032]}{8}$$

$I = 56.08883$

| Segments | W |
|---|---|
| 4 | 56.08883 |
| 8 | 62.82859 |
| 16 | 64.49269 |

24.32

| Segments | W |
|---|---|
| 4 | 64.56122 |
| 8 | 65.07517 |
| 16 | 65.04739 |

24.33

$\epsilon_a = -0.3\%$

| | | | |
|---|---|---|---|
| −37.2048 | 53.29716 | 65.31216 | 65.10621 |
| 30.67167 | 64.56122 | 65.10942 | |
| 56.08883 | 65.07516 | | |
| 62.82859 | | | |

24.34 Gauss Quadrature

$I = 33.7911 + 39.51317$

| points | W |
|---|---|
| 2 | 73.30428 |
| 3 | 64.95193 |
| 4 | 64.89586 |
| 5 | 65.03461 |
| 6 | 65.04463 |

24.35 $W = Fd$

$$d = \int_0^t v(t)\,dt$$

$$\int_0^6 4t\,dt = 72$$ (16 applications of trap rule)

Exact because $v(t)$ is linear

$$\int_6^{14} (24 + (6-t)^2)\,dt = 363$$

true = 362.667

$$W = 200(72 + 363) = 87000$$

24.36 Use $Oh^2$ central differences for middle points and $Oh^2$ forward and backward differences for 1st and last points

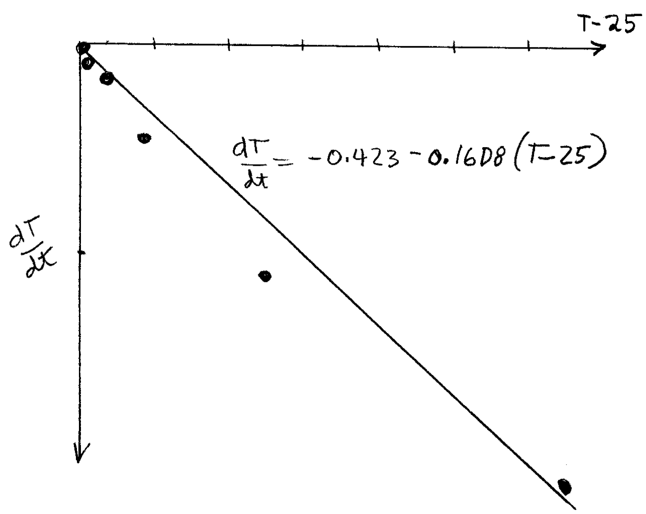
| $t$ | $I$ | $dT/dt$ | $T-25$ |
|---|---|---|---|
| 0 | 90 | -10.4 | 65 |
| 5 | 49.9 | -5.6 | 24.9 |
| 10 | 33.8 | -2.2 | 8.8 |
| 15 | 28.4 | -0.8 | 3.4 |
| 20 | 26.2 | -0.3 | 1.2 |
| 25 | 25.4 | -0.02 | 0.4 |

$$f'(0) = \frac{-33.8 + 4(49.9) - 3(90)}{10}$$

$$= -10.4$$

$$f'(25) = \frac{3(25.4) - 4(26.2) + 28.4}{10}$$

$$= -0.02$$



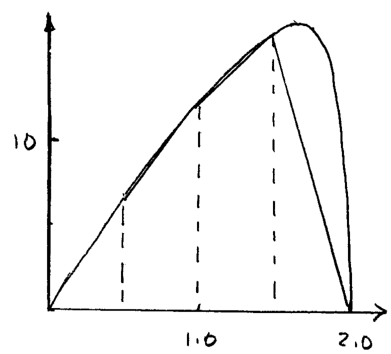

$\therefore\ k \cong 0.161$

24.37

$$I = (0.02-0)\frac{0+40}{2} + (0.05-0.02)\frac{40+37.5}{2} + (0.25-0.05)\frac{[37.5+4(43+60)+2(52)+60]}{12}$$

$$I = 11.72$$

24.38 $\quad Q = \int_0^2 2\left(1-r/2\right)^{1/6}(2\pi r)\,dr$




a rough sketch shows that Trap Rule underestimates area of last section. Error is large because of the nature of the function. There we need many applications for good accuracy.

| applications | Q |
|---|---|
| 2 | 11.1953 |
| 4 | 16.0727 |
| 8 | 18.2054 |
| 16 | 19.1425 |
| 32 | 19.5560 |
| 64 | 19.7390 |
| 128 | 19.820 |
| 500 | 19.87201 |

24.39 $W = \int_0^x F\,dx$

use Trap Rule

$$I = \int_0^{0.35} F\,dx + \int_{0.35}^{0.45} F\,dx$$

$$I = (0.35-0)\frac{\left[0+2(0.01+0.028+0.046+0.063+0.082+0.11)+0.13\right]}{14}$$

$$+ (0.45-0.35)\left[\frac{0.13+0.20}{2}\right]$$

$$= 0.0367$$

24.40

| $t$ | $x$ | $dx/dt = v$ | $dv/dt$ |
|---|---|---|---|
| 0 | 154 | 71.91 | -50.27 |
| 0.51 | 186 | 53.58 | -21.61 |
| 1.03 | 209 | 49.94 | -11.60 |
| 1.74 | 250 | 37.25 | -26.59 |
| 2.36 | 262 | 16.05 | -24.50 |
| 3.24 | 272 | 6.59 | -10.81 |
| 3.82 | 274 | 0.30 | -10.88 |

The above are calculated using Eq 23.9

24.41

$$d = \int_0^{10}(10t^2 - 5t)\,dt + \int_{10}^{20}(1000-5t)\,dt + \int_{20}^{30}(45t+2(t-20)^2)\,dt$$

Use 10 seg trapezoidal rule to generate points for each section.

$d = 3100 + 9250 + 11920$

$= 24{,}270$

24.42

<u>6 Seg Trap Rule</u>

$$I = (30-0)\frac{\left[0+2(132.31+282.71+455.21+655.34+890.97)+1173.94\right]}{12}$$

$= 15017.5$

<u>6 Seg Simpsons 1/3 Rule</u>

$I = 14939.95$

<u>6 Point Gauss Quad</u>

$I = 14939.39$

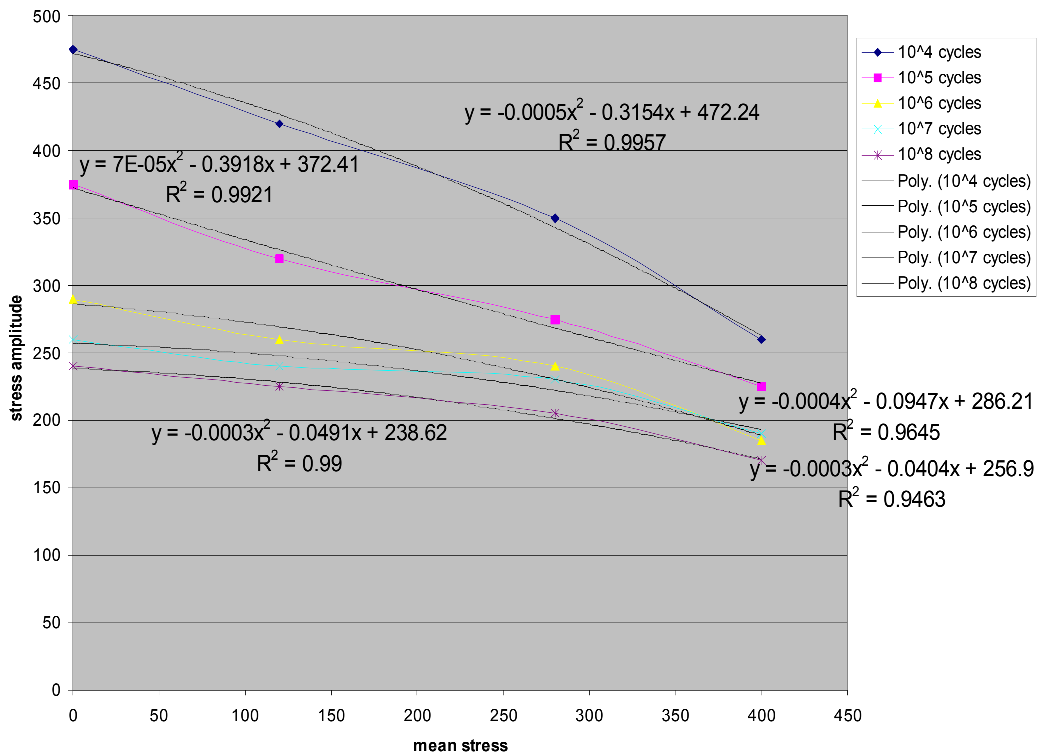
<u>$Oh^8$ Romberg</u>

| | | |
|---|---|---|
| 17609.08 | 14973.9 | 14939.94 |
| 15632.69 | 14942.06 | 14939.41 |
| 15114.72 | 14939.58 | |
| 14983.36 | | |

14939.4

$\epsilon_a = -0.0036\,\%$

24.43



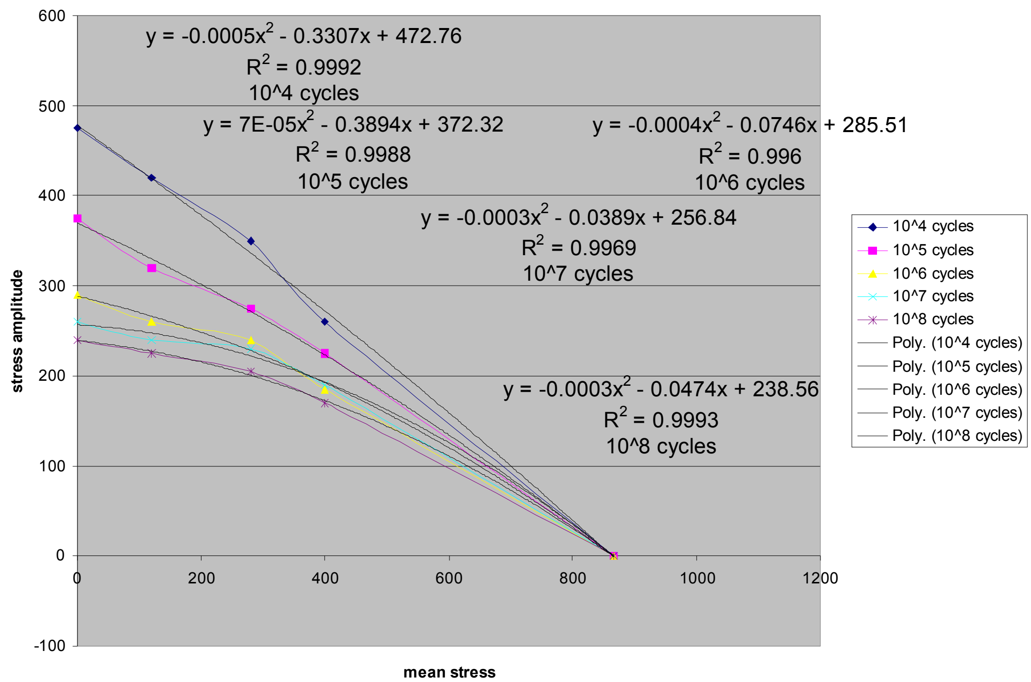

Finding roots in Matlab:

```
a=[-0.0005 -0.3154 472.24];
roots(a)

b=[7E-05 -0.3918  372.41];
roots(b)

c=[-0.0004 -0.0947 286.21];
roots(c)

d=[-0.0003 -0.0404 256.9];
roots(d)

e=[ -0.0003 -0.0491 238.62];
roots(e)
```

Roots: 706.3, 1213.7, 735.75, 860.5, 813.77

Using the AVERAGE command in Excel, the ultimate strength, $\sigma_u$ , was **866 MPa.**

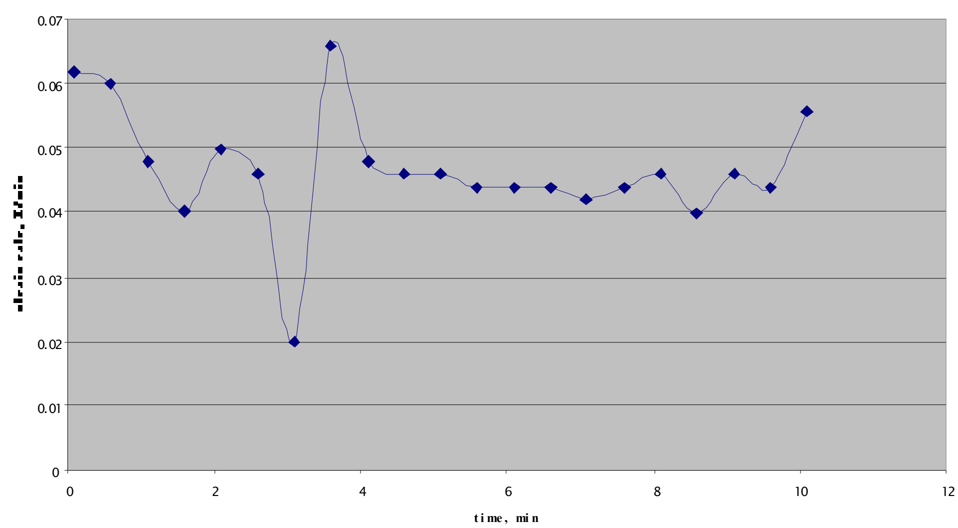
Plot with $\sigma_\mu$ included:




It can be seen from the higher $R^2$ values that the polynomial fit including the ultimate stress, $\sigma_u$, is more accurate than the fit without including $\sigma_u$.

24.44

This problem was solved using Excel.

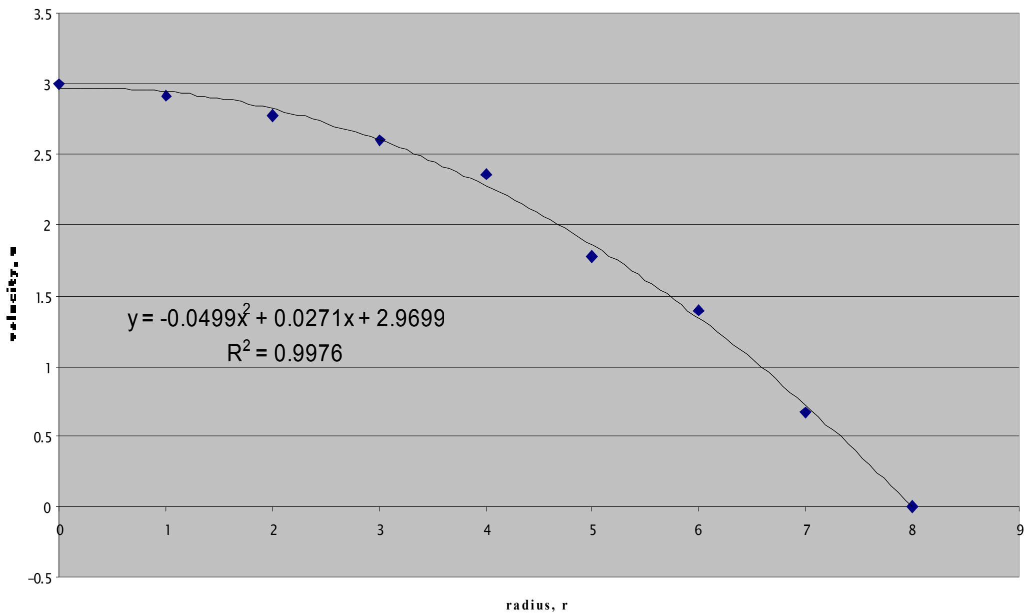



The following values were calculated beginning with the ninth data point of the series.

**Mean = 0.045305**
**Standard Deviation = 0.003716**

24.45 This problem was solved using Excel.

**a)** Find the equation for velocity using Excel.




$$Q = \int_0^R 2\pi \; r \, u \, dr$$

Integrate according to the equation above, using $R = 8$ in.

$\boldsymbol{Q}$ **= 2.12 ft$^3$/s**

**b)**

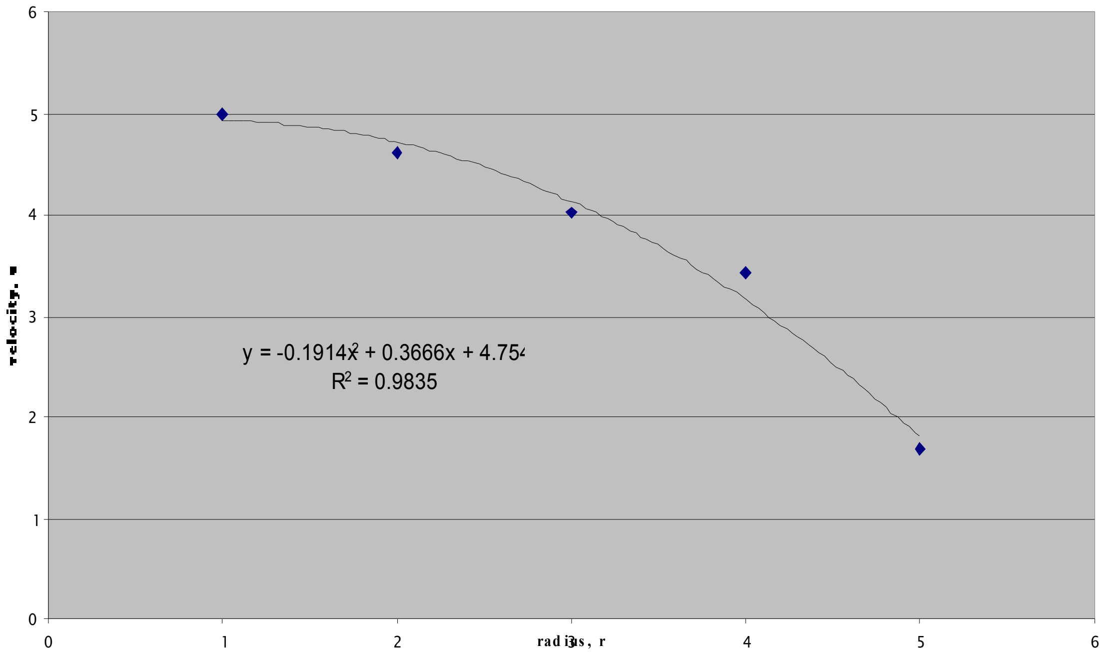
$$I \cong (b-a)\frac{f(x_0) + 4\sum_{i=1,3,5}^{n-1} f(x_i) + 2\sum_{j=2,4,6}^{n-2} f(x_j) + f(x_n)}{3n}$$

$$I \cong 2\pi * (8)\frac{0*3 + 4(1*2.92 + 3*2.61 + 5*1.78 + 7*0.67) + 2(2*2.78 + 4*2.36 + 6*1.4) + 8*0}{3*8}$$

$\boldsymbol{I} \cong$ **2.097 ft$^3$/s**

**c) % error** $= \left|\dfrac{2.097 - 2.12}{2.12} * 100\right| = 1.10\%$

24.46 a) Find the equation for velocity using Excel.




$$Q = \int_{r_1}^{r_2} 2\pi \; r\, u\, dr$$

To find the volume flow rate in the region around the plug, integrate according to the equation above, using $r_1$=1 in. and $r_2$= 6 in.

$Q_1$ = 2.073 ft$^3$/s

To find the volume flow rate of the plug, use $Q_2 = u_c A_c$

$Q_2$ = 0.1091 ft$^3$/s

$Q = Q_1 + Q_2$ = 2.182 ft$^3$/s

**$Q$ = 2.182 ft$^3$/s**

**b)** Integral for the outer region:

$$I \cong (b-a)\frac{f(x_0)+4\sum_{i=1,3,5}^{n-1} f(x_i)+2\sum_{j=2,4,6}^{n-2} f(x_j)+f(x_n)}{3n}$$

$$I \cong 2\pi * (5)\frac{1*5+4(2*4.62+4*3.42)+2(3*4.01+5*1.69)+0*6}{3*5}$$

$I \cong$ 2.002 ft$^3$/s

Inner region $Q2$ = 0.1091 ft$^3$/s remains the same.

**Therefore, the volume flow rate $Q$ = 2.111 ft$^3$/s.**

**c) % error** $= \left| \dfrac{2.111 - 2.182}{2.111} \right| = 3.36\%$

24.47 The following Excel worksheet solves the problem. Note that the derivative is calculated with a centered difference,

$$\frac{dV}{dT} = \frac{V_{450K} - V_{350K}}{100K}$$

F4 = =C4-450*E4

| | A | B | C | D | E | F | G | H |
|---|---|---|---|---|---|---|---|---|
| 1 | Prob. 24.47 | | | | | | | |
| 2 | | | | | | | | |
| 3 | P,atm | T=350K | T=400K | T=450K | dVdT | (V - T (dV/dT)p) | Integral | |
| 4 | 0.1 | 220 | 250 | 282.5 | 0.625 | -31.25 | | |
| 5 | 5 | 4.1 | 4.7 | 5.23 | 0.0113 | -0.385 | -77.5058 | Trap |
| 6 | 10 | 2.2 | 2.5 | 2.7 | 0.005 | 0.25 | -0.3375 | Trap |
| 7 | 20 | 1.35 | 1.49 | 1.55 | 0.002 | 0.59 | 4.2 | Trap |
| 8 | 25 | 1.1 | 1.2 | 1.24 | 0.0014 | 0.57 | | |
| 9 | 30 | 0.9 | 0.99 | 1.03 | 0.0013 | 0.405 | 5.458333 | Simp1/3 |
| 10 | 40 | 0.68 | 0.75 | 0.78 | 0.001 | 0.3 | 3.525 | Trap |
| 11 | 45 | 0.61 | 0.675 | 0.7 | 0.0009 | 0.27 | 1.425 | Trap |
| 12 | 50 | 0.54 | 0.6 | 0.62 | 0.0008 | 0.24 | 1.275 | Trap |
| 13 | | | | | | | | |
| 14 | | | | | | Total Integral = | **-61.9599** | |

24.48 A single application of the trapezoidal rule yields:

$$I = (22 - 2)\frac{12.2 + 1.11}{2} = 133.1$$

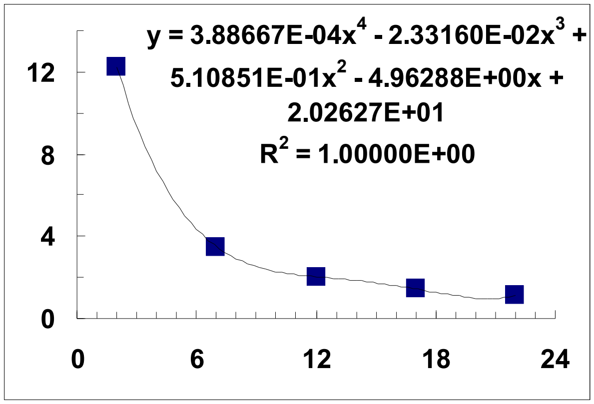
A 2-segment trapezoidal rule gives

$$I = (22 - 2)\frac{12.2 + 2(2.04) + 1.11}{4} = 86.95$$

A 4-segment trapezoidal rule gives

$$I = (22 - 2)\frac{12.2 + 2(3.49 + 2.04 + 1.44) + 1.11}{8} = 68.125$$

Because we do not know the true value, it it would seem impossible to estimate the error. However, we can try to fit different order polynomials to see if we can get a decent fit to the data. This yields the surprising result that a 4$^{th}$-order polynomial results in almost a perfect fit. For example, using the Excel trend line gives:




This can be integrated analytically to give 61.20365. Note that the same result would result from using Boole's rule, Rhomberg integration or Gauss quadrature.[1]

Therefore, we can estimate the errors as

$$I = \left| \frac{61.20365 - 133.1}{61.20365} \right| \times 100\% = 117.47\%$$

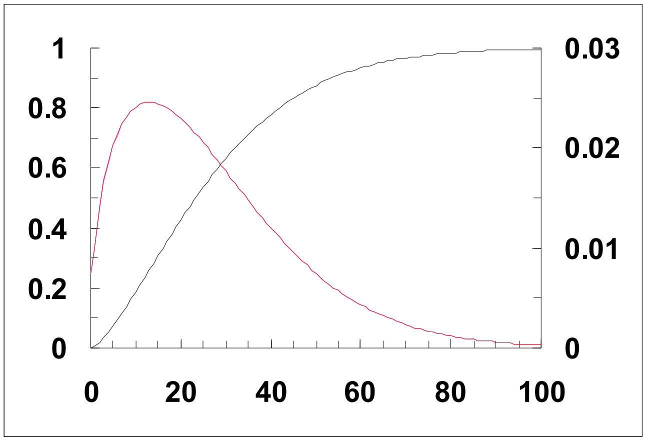
$$I = \left| \frac{61.20365 - 86.95}{61.20365} \right| \times 100\% = 42.07\%$$

$$I = \left| \frac{61.20365 - 68.125}{61.20365} \right| \times 100\% = 11.31\%$$

The ratio of these is 117.47:42.07:11.31 = 10.4:3.7:1. Thus it approximates the quartering of the error that we would expect according to Eq. 21.13.

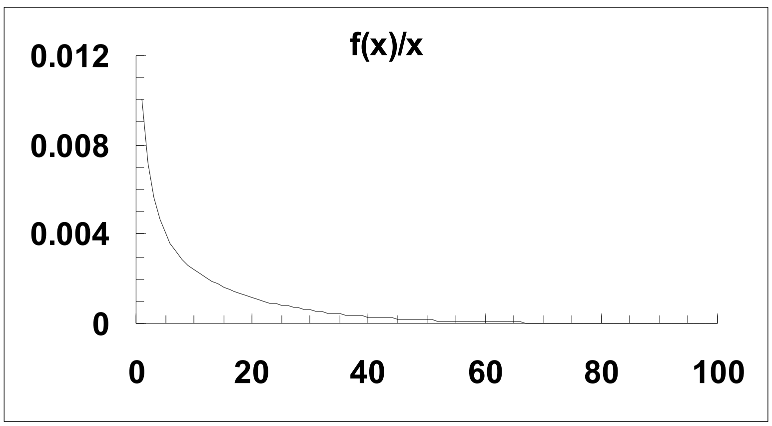
24.49 (*b*) This problem can be solved in a number of ways. One approach is to set up Excel with a series of equally-spaced *x* values from 0 to 100. Then one of the formulas described in this Part of the book can be used to numerically compute the derivative. For example, I used x values with an interval of 1 and Eq. 23.9. The resulting plot of the function and its derivative is

[1] There might be a slight discrepancy due to roundoff.




(*b*) Inspection of this plot indicates that the maximum derivative occurs at about a diameter of 13.3.

(*c*) The function to be integrated looks like




This can be integrated from 1 to a high number using any of the methods provided in this book. For example, the Trapezoidal rule can be used to integrate from 1 to 100, 1 to 200 and 1 to 300 using $h = 1$. The results are:

| $h$ | $I$ |
|---|---|
| 100 | 0.073599883 |
| 200 | 0.073632607 |
| 300 | 0.073632609 |

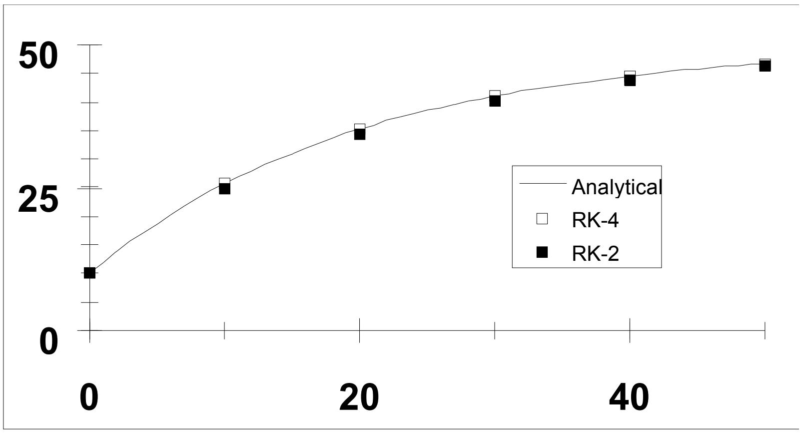
Thus, the integral seems to be converging to a value of 0.073633. $S_m$ can be computed as $6 \times 0.073633 = 0.4418$.

**CHAPTER 28**

28.1 The solution with the 2nd-order RK (Heun without corrector) can be laid out as

For the 4th-order RK, the solution is

A plot of both solutions along with the analytical result is displayed below:




28.2 The mass-balance equations can be written as

$$\frac{dc_1}{dt} = -0.14c_1 + 0.04c_3$$

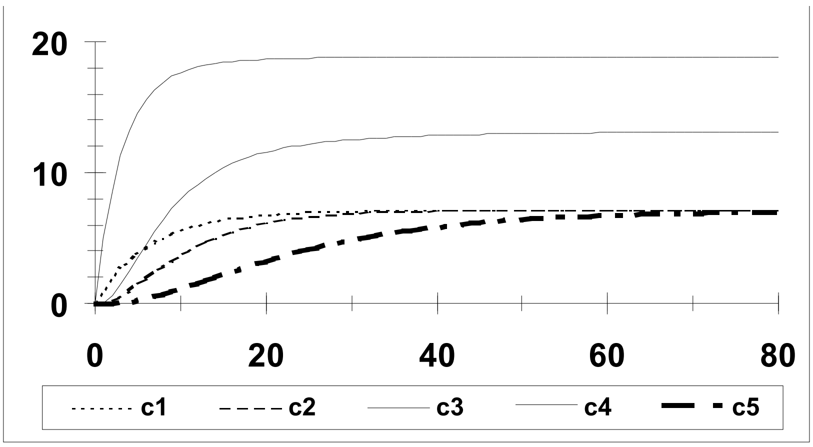
$$\frac{dc_2}{dt} = 0.2c_1 - 0.2c_2$$

$$\frac{dc_3}{dt} = 0.025c_2 - 0.275c_3$$

$$\frac{dc_4}{dt} = 0.1125c_3 - 0.175c_4 + 0.025c_5$$

$$\frac{dc_5}{dt} = 0.03c_1 + 0.03c_2 - 0.06c_5$$

Selected solution results (Euler's method) are displayed below, along with a plot of the results.




Finally, MATLAB can be used to determine the eigenvalues and eigenvectors:

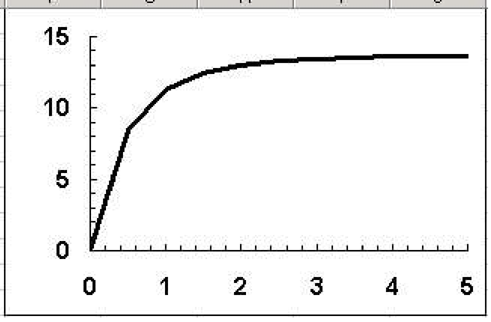
```
>> a=[.14 -.04 0 0 0;-.2 .2 0 0 0;0 -.025 .275 0 0;0 0 -.1125 .175 -.025;-.03
-.03 0 0 .06]

a =
    0.1400   -0.0400         0         0         0
   -0.2000    0.2000         0         0         0
         0   -0.0250    0.2750         0         0
         0         0   -0.1125    0.1750   -0.0250
   -0.0300   -0.0300         0         0    0.0600

>> [v,d]=eig(a)

v =
         0         0         0   -0.1836    0.0826
         0         0         0   -0.2954   -0.2567
         0    0.6644         0   -0.0370   -0.6021
    1.0000   -0.7474    0.2124    0.1890    0.7510
         0         0    0.9772    0.9176    0.0256

d =
    0.1750         0         0         0         0
         0    0.2750         0         0         0
         0         0    0.0600         0         0
         0         0         0    0.0757         0
         0         0         0         0    0.2643
```

28.3 Substituting the parameters into the differential equation gives

$$\frac{dc}{dt} = 20 - 0.1c - 0.1c^2$$

The mid-point method can be applied with the result:

| | A | B | C | D | E |
|---|---|---|---|---|---|
| 1 | t | c | | | |
| 2 | 0 | 0 | | Run | |
| 3 | 0.5 | 8.5 | | | |
| 4 | 1 | 11.33498 | | | |
| 5 | 1.5 | 12.45029 | | | |
| 6 | 2 | 12.99069 | | | |
| 7 | 2.5 | 13.27729 | | | |
| 8 | 3 | 13.43624 | | | |
| 9 | 3.5 | 13.52653 | | | |
| 10 | 4 | 13.57851 | | | |
| 11 | 4.5 | 13.60866 | | | |
| 12 | 5 | 13.62623 | | | |

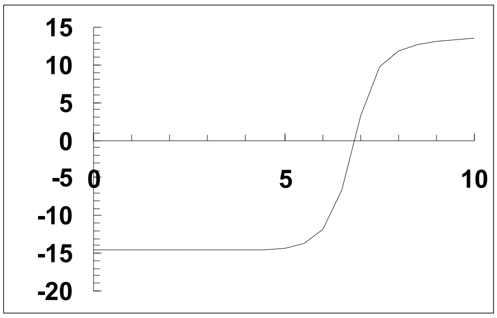



The results are approaching a value of 13.6351

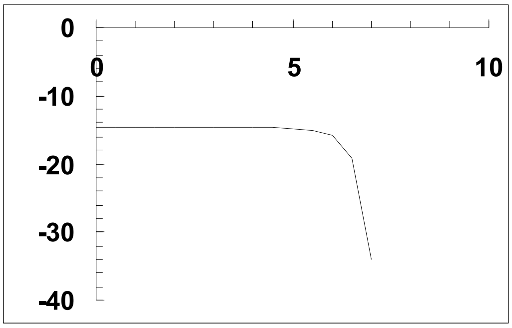
**Challenge question:**

The steady state form (i.e., $dc/dt = 0$) of the equation is $0 = 200 - c - c^2$, which can be solved for 13.65097141 and -14.65097141. Thus, there is a negative root.

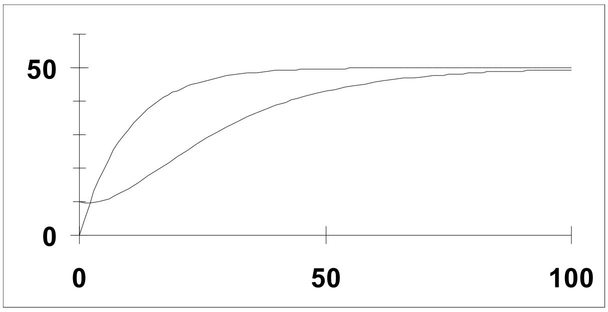
If we put in the initial y value as –14.650971 (or higher precision, the solution will stay at the negative root. However, if we pick a value that is slightly higher (a per machine precision), it will gravitate towards the positive root. For example if we use –14.65097



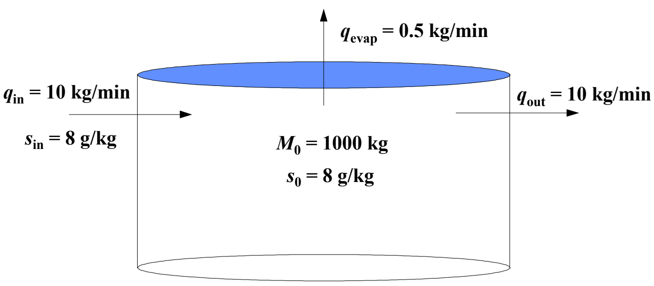

Conversely, if we use a slightly lower value, it will go unstable




28.4 The first steps of the solution are shown below along with a plot. Notice that the a value of the inflow concentration at the end of the interval (***cin-end***) is required to calculate the $k_2$'s correctly.




28.5 The system is as depicted below:




(*a*) The mass of water in the tank can be modeled with a simple mass balance

$$\frac{dM}{dt} = q_{\text{in}} - q_{\text{out}} - q_{\text{evap}} = 10 - 10 - 0.5 = -0.5$$

With the initial condition that $M = 1000$ at $t = 0$, this equation can be integrated to yield,

$$M = 1000 - 0.5t$$

Thus, the time to empty the tank ($M$ = 0) can be calculated as $t$ = 1000/0.5 = 2000 minutes.

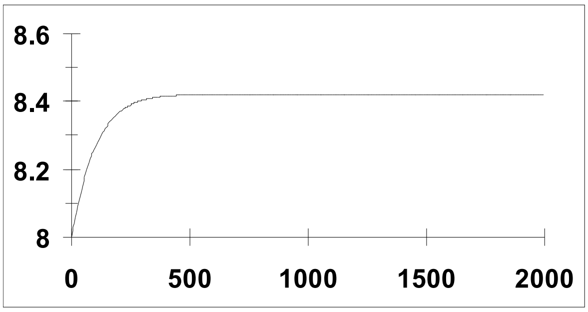
($b$) The concentration in the tank over this period can be computed in several ways. The simplest is to compute the mass of salt in the tank over time by solving the following differential equation:

$$\frac{dm}{dt} = q_{\text{in}} s_{\text{in}} - q_{\text{out}} s$$

where $m$ = the mass of salt in the tank. The salt concentration in the tank, $s$, is the ratio of the mass of salt to the mass of water

$$s = \frac{m}{M} = \frac{m}{1000 - 0.5t}$$

The first few steps of the solution of this ODE with Euler's method is tabulated below. In addition, a graph of the entire solution is also displayed.




Recognize that a singularity occurs at $t$ = 2000, because the tank would be totally empty at this point.

28.6 A heat balance for the sphere can be written as

$$\frac{dH}{dt} = hA(T_a - T)$$

The heat gain can be transformed into a volume loss by considering the latent heat of fusion. Thus,

$$\frac{dV}{dt} = -\frac{hA}{\rho L_f}(T_a - T) \qquad (1)$$

where ρ = density ≅ 1 kg/m$^3$ and $L_f$ = latent heat of fusion ≅ 333 kJ/kg. The volume and area of a sphere are computed by

$$V = \frac{4}{3}\pi r^3 \qquad A = 4\pi r^2$$

(2)

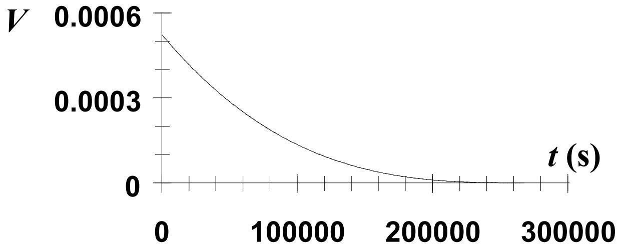
These can be combined with (1) to yield,

$$\frac{dV}{dt} = \frac{h 4\pi \left( \frac{3}{4} \frac{V}{\pi} \right)^{2/3}}{\rho L_f} (T_a - T)$$

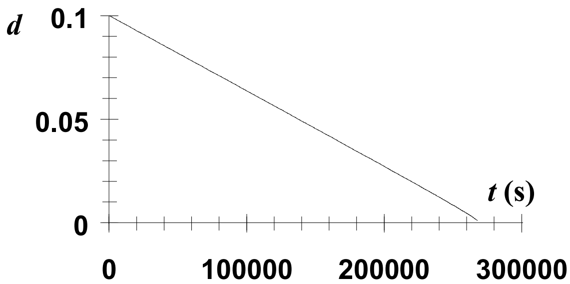
This equation can be integrated along with the initial condition,

$$V_0 = \frac{4}{3} \pi (0.05)^3 = 0.000524 \text{ m}^3$$

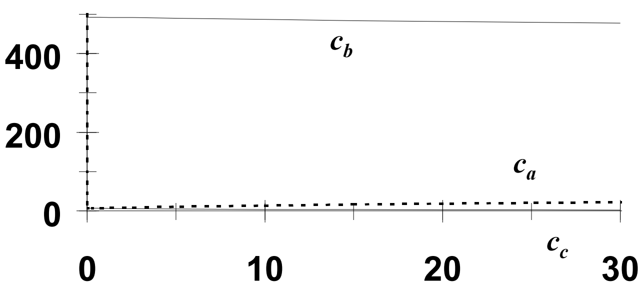
to yield the resulting volume as a function of time.



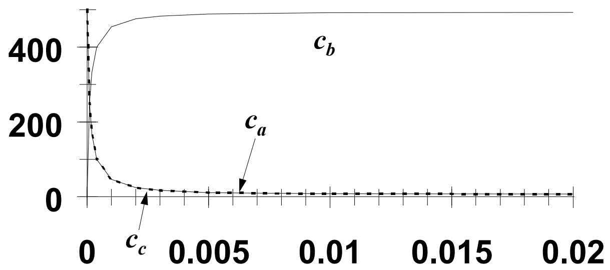

This result can be converted into diameter using (2)




28.7 The system for this problem is stiff. Thus, the use of a simple explicit Runge-Kutta scheme would involve using a very small time step in order to maintain a stable solution. A solver designed for stiff systems was used to generate the solution shown below. Two views of the solution are given. The first is for the entire solution domain.




In addition, we can enlarge the initial part of the solution to illustrate the fast transients that occur as the solution moves from its initial conditions to its dominant trajectories.



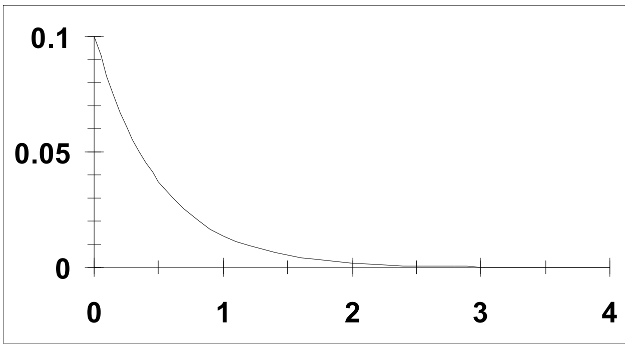

28.8 Several methods could be used to obtain a solution for this problem (e.g., finite-difference, shooting method, finite-element). The finite-difference approach is straightforward:

$$D\frac{A_{i-1} - 2A_i + A_{i+1}}{\Delta x^2} - kA_i = 0$$

Substituting parameter values and collecting terms gives

$$-1\times 10^{-6} A_{i-1} + (2\times 10^{-6} + 4\times 10^{-6}\Delta x^2) - 1\times 10^{-6} A_{i+1} = 0$$

Using a $\Delta x = 0.2$ cm this equation can be written for all the interior nodes. The resulting linear system can be solved with an approach like the Gauss-Seidel method. The following table and graph summarize the results.




28.9 The ODE to be solved is

$$\frac{dP}{dt} = -\frac{b}{a}P + \frac{A\sin\omega t}{a}$$

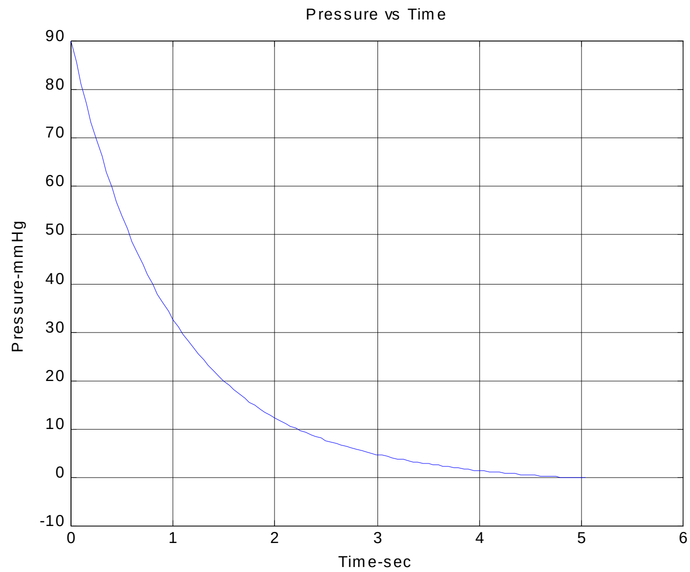
Substituting the parameters, it becomes

$$\frac{dP}{dt} = \sin t - P$$

The following Matlab script uses Euler's method to solve the problem.

```
dt=0.05;
max=5;
n=max/dt+1;
t=zeros(1,n);
p=zeros(1,n);
t(1)=0;
p(1)=90;
for i=1:n
   p(i+1)=p(i)+dydt(t(i),p(i))*dt;
   t(i+1)=t(i)+dt;
end
```

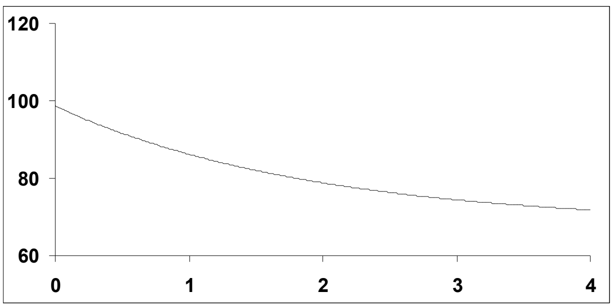
```
plot(t,p)
grid
xlabel('Time-sec')
ylabel('Pressure-mmHg')
title('Pressure vs Time')
zoom

function s=dydt(t,p);
A=1;
w=1;
s=A*sin(w*t)-p;
```




28.10 Excel can be used to compute the basic results. As can be seen, the person died 1.13 hrs prior to being discovered. The non-self-starting Heun yielded the following time series of temperature:

B11 = =-1/K*LN((Tempd-Tempa)/(Temp0-Tempa))

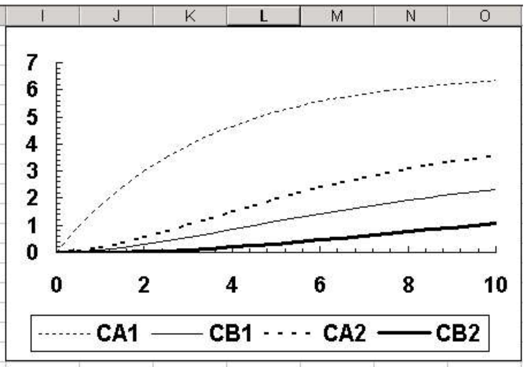
| | A | B | C | D | E | F | G |
|---|---|---|---|---|---|---|---|
| 1 | Prob. 28.10 | | | | | | |
| 2 | | | | | | | |
| 3 | Tempa | 68 | oF | | | | |
| 4 | Temp0 | 85 | oF | | | | |
| 5 | t1 | 2 | hr | | | | |
| 6 | Temp1 | 74 | oF | | | | |
| 7 | Tempd | 98.6 | oF | | | | |
| 8 | | | | | | | |
| 9 | K | 0.520727 | /hr | | | | |
| 10 | | | | | | | |
| 11 | td | -1.12878 | hr | | | | |



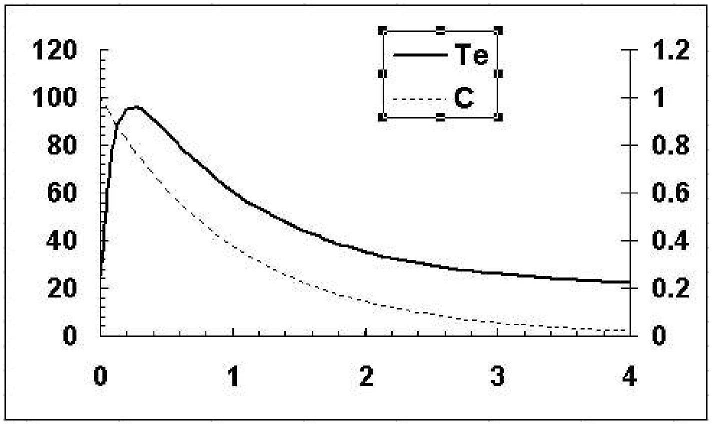

28.11 The classical 4th order RK method yields

| t | CA1 | CB1 | CA2 | CB2 |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 0.5 | 0.928609 | 0.023016 | 0.045281 | 0.00075 |
| 1 | 1.727872 | 0.08482 | 0.164179 | 0.00546 |
| 1.5 | 2.415803 | 0.176013 | 0.335308 | 0.016718 |
| 2 | 3.007912 | 0.288885 | 0.541809 | 0.035962 |
| 2.5 | 3.517545 | 0.417146 | 0.770508 | 0.063783 |
| 3 | 3.956191 | 0.55569 | 1.011215 | 0.100165 |
| 3.5 | 4.333736 | 0.700407 | 1.256147 | 0.144668 |
| 4 | 4.658694 | 0.848013 | 1.499457 | 0.19657 |
| 4.5 | 4.938387 | 0.995913 | 1.736845 | 0.254981 |
| 5 | 5.179121 | 1.142081 | 1.965239 | 0.318922 |
| 5.5 | 5.386324 | 1.284962 | 2.18254 | 0.387384 |
| 6 | 5.564665 | 1.42339 | 2.387406 | 0.459374 |
| 6.5 | 5.718164 | 1.556515 | 2.579086 | 0.533944 |
| 7 | 5.850283 | 1.683745 | 2.757278 | 0.610212 |
| 7.5 | 5.963998 | 1.804698 | 2.922021 | 0.687375 |
| 8 | 6.061873 | 1.919159 | 3.073603 | 0.764714 |
| 8.5 | 6.146116 | 2.027046 | 3.212496 | 0.841596 |
| 9 | 6.218624 | 2.128384 | 3.339293 | 0.917476 |
| 9.5 | 6.281033 | 2.223279 | 3.454669 | 0.991889 |
| 10 | 6.334748 | 2.311897 | 3.559345 | 1.064449 |

RUN

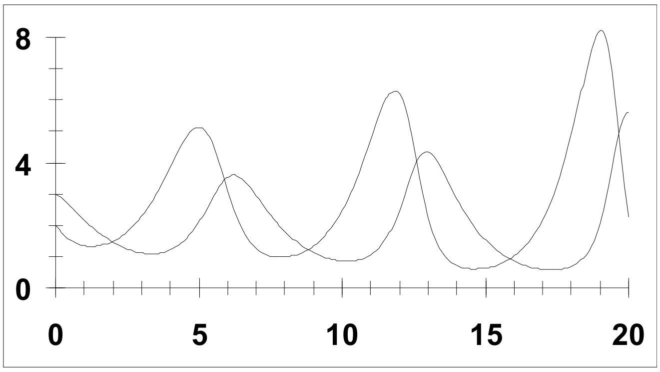



28.12 The classical 4th order RK method yields

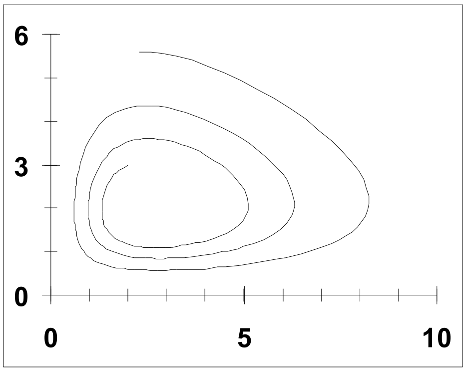



28.13 (*a*) The first few steps of Euler's method are shown in the following table

A plot of the entire simulation is shown below:

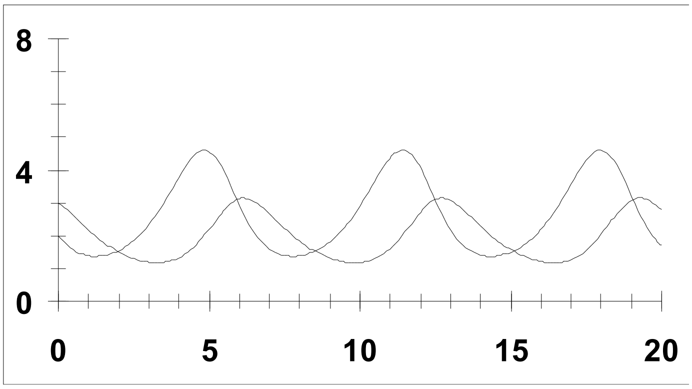



Notice that because the Euler method is lower order, the peaks are increasing, rather than repeating in a stable manner as time progresses. This result is reinforced when a state-space plot of the calculation is displayed.

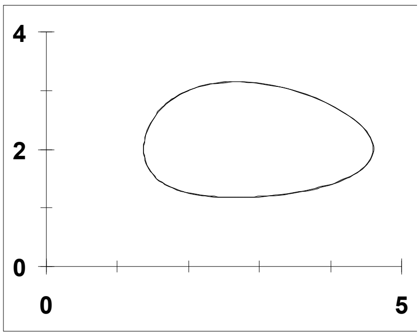



($b$) The first few steps of the Heun method is shown in the following table

A plot of the entire simulation is shown below:



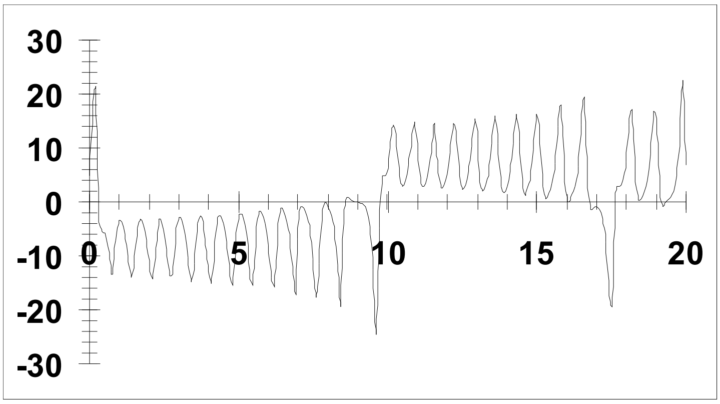

Notice that in contrast to the Euler method, the peaks are stable manner as time progresses. This result is also reinforced when a state-space plot of the calculation is displayed.




($c$) The first few steps of the 4$^{th}$-order RK method is shown in the following table

The results are quite close to those obtained with the Heun method in part ($b$). In fact, both the time series and state-space plots are indistinguishable from each other.

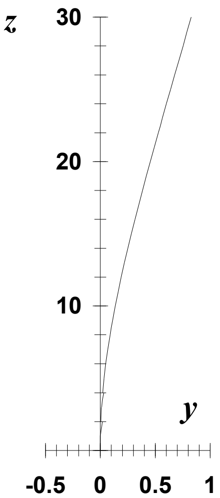
28.14 Using the step size of 0.1, ($a$) and ($b$) both give unstable results. The 4$^{th}$-order RK method yields a stable solution. The first few values are shown in the following table. A plot of the result for $x$ is also shown below. Notice how after about $t$ = 6, this solution diverges from the double precision version in Fig. 28.9.




28.15 The second-order equation can be reexpressed as a pair of first-order equations,

$$\frac{dy}{dz} = w$$
$$\frac{dw}{dz} = \frac{f}{2EI}(L-z)^2$$

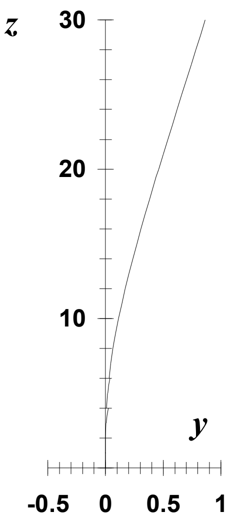
We used Euler's method with $h = 1$ to obtain the solution:



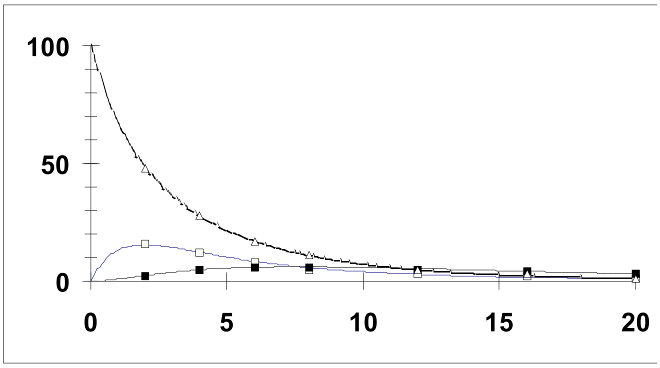

28.16 The second-order equation can be reexpressed as a pair of first-order equations,

$$\frac{dy}{dz} = w \qquad \frac{dw}{dz} = \frac{200ze^{-2z/30}}{(5+z)2EI}(L-z)^2$$

We used Euler's method with $h = 1$ to obtain the solution:



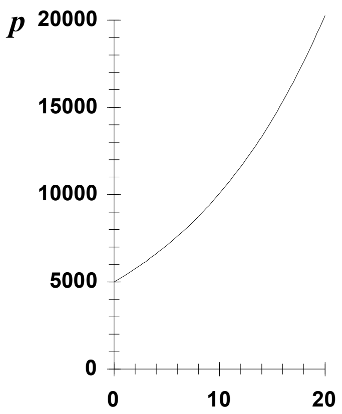

28.17 This problem was solved using the Excel spreadsheet in a fashion similar to the last example in Sec. 28.1. We set up Euler's method to solve the 3 ODEs using guesses for the diffusion coefficients. Then we formed a column containing the squared residuals between our predictions and the measured values. Adjusting the diffusion coefficients with the Solver tool minimized the sum of the squares. At first, we assumed that the diffusion coefficients were zero. For this case the Solver did not converge on a credible answer. We then made guesses of $1\times10^7$ for both. This magnitude was based on the fact that the volumes were of this order of magnitude. The resulting simulation did not fit the data very well, but was much better than when we had guessed zero. When we used Solver, it converged on $E_{12} = 9.22\times10^5$ and $E_{13} = 2.19\times10^6$ which corresponded to a sum of the squares of residuals of 2.007. Some of the Euler calculations are displayed below along with a plot of the fit.

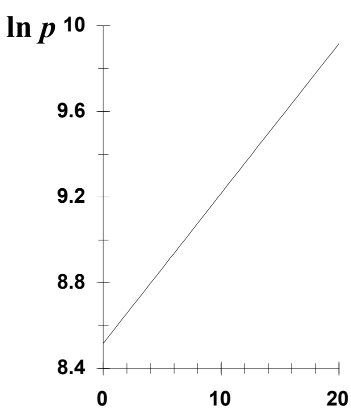



It should be noted that we made up the "measurements" for this problem using the 4th-order RK method with values for diffusive mixing of $E_{12} = 1\times10^6$ and $E_{13} = 2\times10^6$. We then used a random number generator to add some error to this "data."

28.18 The Heun method can be used to compute

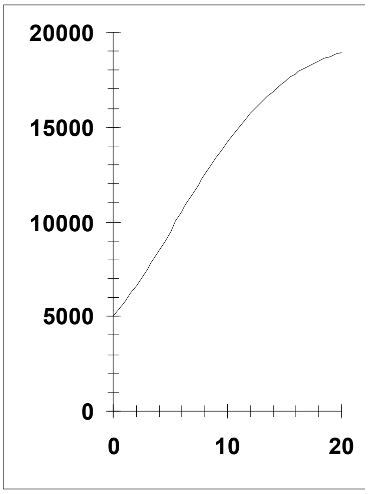
The results can be plotted. In addition, linear regression can be used to fit a straight line to $\ln p$ versus $t$ to give $\ln p = 8.52 + 0.07t$. Thus, as would be expected from a first-order model, the slope is equal to the growth rate of the population.



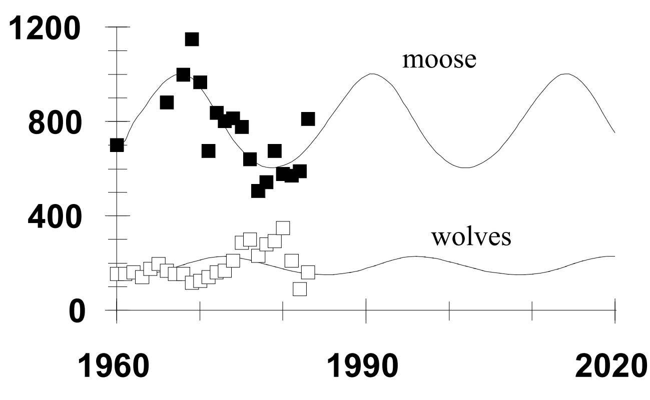

28.19 The Heun method can be used to compute

The results can be plotted.

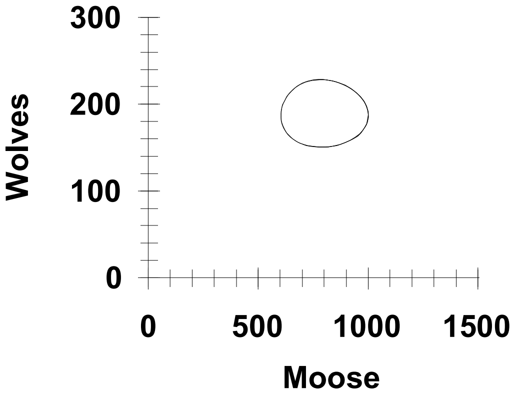



The curve is s-shaped. This shape occurs because initially the population is increasing exponentially since $p$ is much less than $p_{max}$. However, as $p$ approaches $p_{max}$, the growth rate decreases and the population levels off.

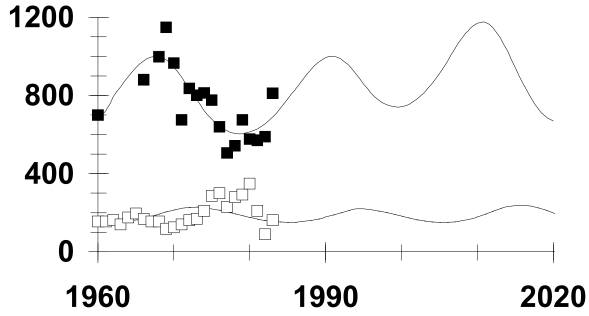
28.20 (*a*) Nonlinear regression (e.g., using the Excel solver option) can be used to minimize the sum of the squares of the residuals between the data and the simulation. The resulting estimates are: $a = 0.32823$, $b = 0.01231$, $c = 0.22445$, and $d = 0.00029$. The fit is:

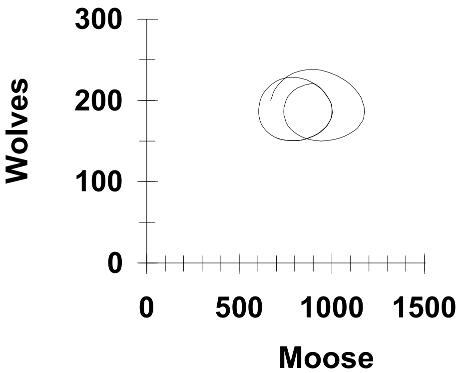



(*b*) The results in state space are,

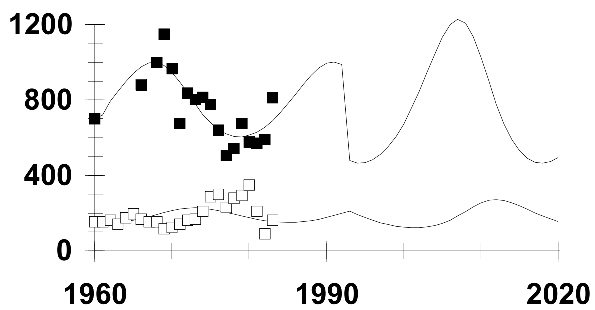

300
200
100
0
Wolves
0
500
1000
1500
Moose

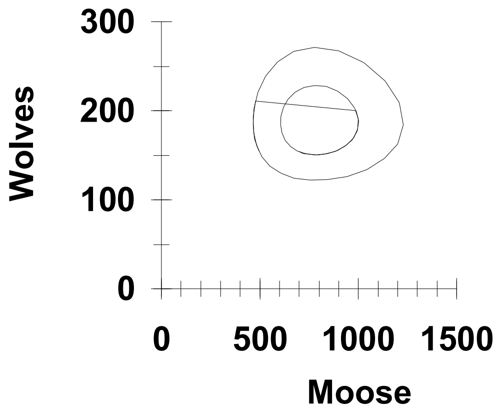
(*c*)


1200
800
400
0
1960
1990
2020


300
200
100
0
Wolves
0
500
1000
1500
Moose

(*d*)


1200
800
400
0
1960
1990
2020

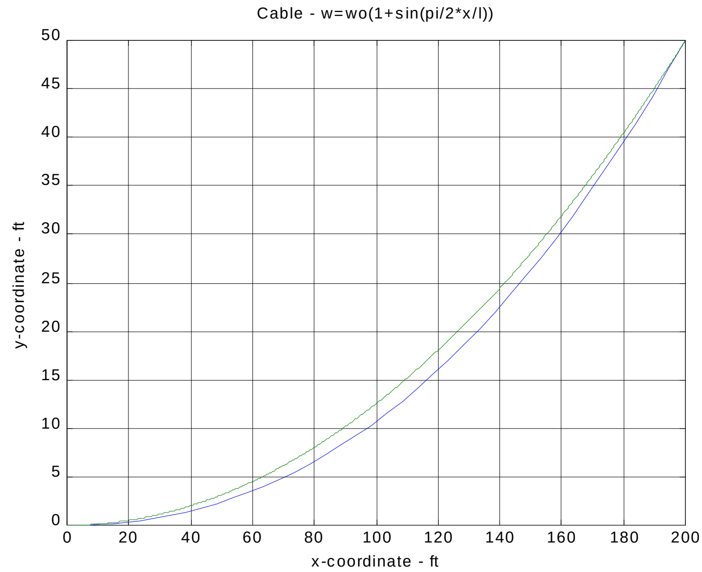



28.21 Main Program:

```
% Hanging static cable - w=w(x)
% Parabolic solution w=w(x)
% CUS Units (lb,ft,s)
% w = wo(1+sin(pi/2*x/l)
es=0.5e-7
% Independent Variable x, xs=start x, xf=end x xs=0; xf=200;
%initial conditions [y(1)=cable y-coordinate, y(2)=cable slope];
ic=[0 0];

global wToP

wToP=0.0025;
[x,y]=ode45('slp',xs,xf,ic,.5e-7);
yf(1)=y(length(x));
wTo(1)=wToP;
ea(1)=1;

wToP=0.002;
[x,y]=ode45('slp',xs,xf,ic,.5e-7);
yf(2)=y(length(x));
wTo(2)=wToP;
ea(2)=abs( (yf(2)-yf(1))/yf(2) );

for k=3:10
   wTo(k)=wTo(k-1)+(wTo(k-1)-wTo(k-2))/(yf(k-1)-yf(k-2))*(50-yf(k-1));
  wToP=wTo(k);
  [x,y]=ode45('slp',xs,xf,ic,.5e-7);
  yf(k)=y(length(x));
  ea(k)=abs( (yf(k)-yf(k-1))/yf(k) );
  if (ea(k)<=es)
    %Analytic Solution with constant w (for Comparison)
    xa=xs:.01:xf;
    ya=(0.00125)*(xa.*xa);
    plot(x,y(:,1),xa,ya,'--'); grid;
    xlabel('x-coordinate - ft'); ylabel('y-coordinate - ft');
    title('Cable - w=wo(1+sin(pi/2*x/l))');
    fprintf('wTo %f\n', wTo)
    fprintf('yf  %f\n', yf)
    fprintf('ea  %f\n', ea)
    break
 end
end
```

Function ‘slp’:

```
function dxy=slp(x,y)
global wToP
dxy=[y(2);(wToP)*(1+sin(pi/2*x/200))]
```




28.22

<u>Analytic Solution for the case where $b = 0$</u>

Substituting into the analytic solution the end point dimensions gives

$$\frac{25}{T_o} - \cosh\left(\frac{50}{T_o}\right) + 1 = 0$$

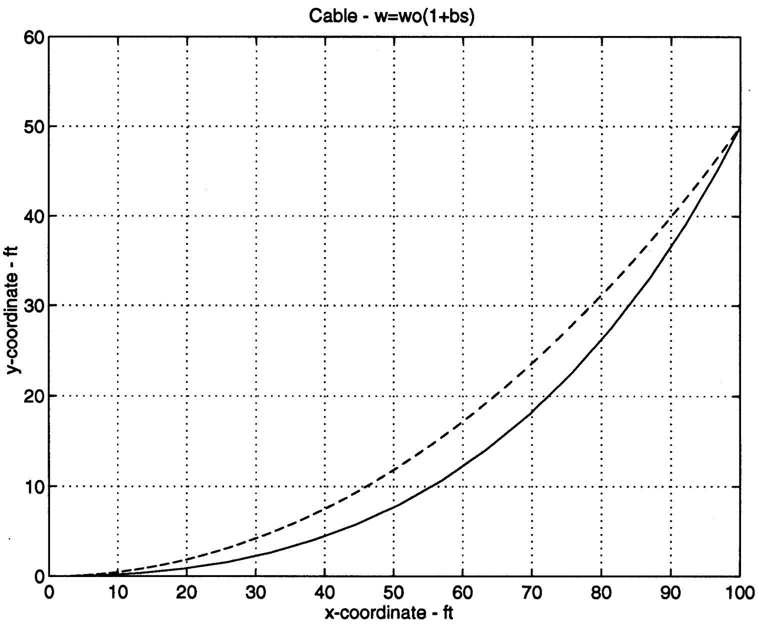
This equation can be solved using a root finding numerical method which gives $T_o = 53.7$ lbs.

<u>Numerical Solution</u>

In this solution the ratio of $w_o/T_o$ is first estimated for two values and these results are used to make the next estimate using a method similar to the shooting method. The convergence goal is to make the final y-value $y_A=l_A=50$ ft. The MATLAB program for execution is listed below.

```
  %  Hanging static cable
  %  Catenary w=w(s)
  %  Weight/unit length w=wo(1+b*s)
es=0.5e-5;
  %  Independent Variable x, xs=start x, xf=end x
xs=0; xf=100;

  %initial conditions
  %[y(1)=cable y-coordinate, y(2)=cable slope, y(3)=cable length];
ic=[0 0 0];

global wToP

wToP=0.0093;
[x,y]=ode45('slc',xs,xf,ic,.5e-7);
yf(1)=y(length(x));
wTo(1)=wToP;
ea(1)=1;

wToP=0.002;
[x,y]=ode45('slc',xs,xf,ic,.5e-7);
yf(2)=y(length(x));
wTo(2)=wToP;
ea(2)=abs( (yf(2)-yf(1))/yf(2) );

for k=3:10
   wTo(k)=wTo(k-1)+(wTo(k-1)-wTo(k-2))/(yf(k-1)-yf(k-2))*(50-yf(k-1));
   wToP=wTo(k);
   [x,y]=ode45('slc',xs,xf,ic,.5e-7);
   yf(k)=y(length(x));
   ea(k)=abs( (yf(k)-yf(k-1))/yf(k) );
   if (ea(k)<=es)
       %Analytic Solution with constant w
       xa=xs:.01:xf;
       ya=(107.432018)*(cosh(0.009308212*xa)-1);
       plot(x,y(:,1),xa,ya,'--'); grid;
          xlabel('x-coordinate - ft'); ylabel('y-coordinate - ft');
          title('Cable - w=wo(1+bs)');
        format long
      fprintf('wTo = %e\n', wTo)
      fprintf('yf = %f\n', yf)
      fprintf('ea = %e\n', ea)
     break
   end
end

function dxy=slc(x,y)
global wToP
dxy(1)=y(2);
dxy(2)=(wToP)*(1+0.05*y(3))*sqrt(1+y(2).*y(2));
dxy(3)=sqrt(1+y(2).*y(2));
```

```
>>
wTo = 9.300000e-03
wTo = 2.000000e-03
wTo = 2.167663e-03
wTo = 3.447873e-03
wTo = 3.302391e-03
wTo = 3.317200e-03
wTo = 3.317416e-03
wTo = 3.317415e-03
yf = 992.781477
yf = 27.837645
yf = 30.404032
yf = 52.512367
yf = 49.715274
yf = 49.995924
yf = 50.000007
yf = 50.000000
ea = 1.000000e+00
ea = 3.466327e+01
ea = 8.440943e-02
ea = 4.210120e-01
ea = 5.626225e-02
ea = 5.613466e-03
ea = 8.164681e-05
ea = 1.328803e-07
>>
```

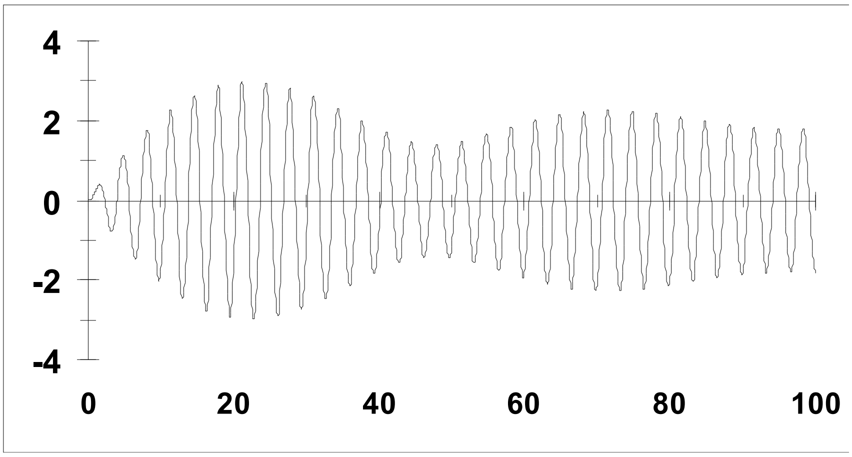



28.23 The second-order equation can be reexpressed as a pair of first-order equations,

$$\frac{dq}{dt} = i$$

$$\frac{di}{dz} = -0.05i - 4q + \sin 1.8708t$$

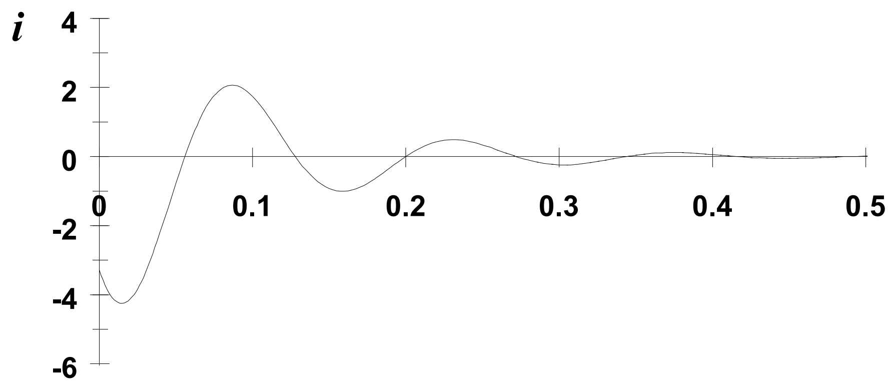
The parameters can be substituted and the system solved with the 4th-order RK method in double-precision with $h = 0.1$. A table showing the first few steps and a graph of the entire solution are shown below.

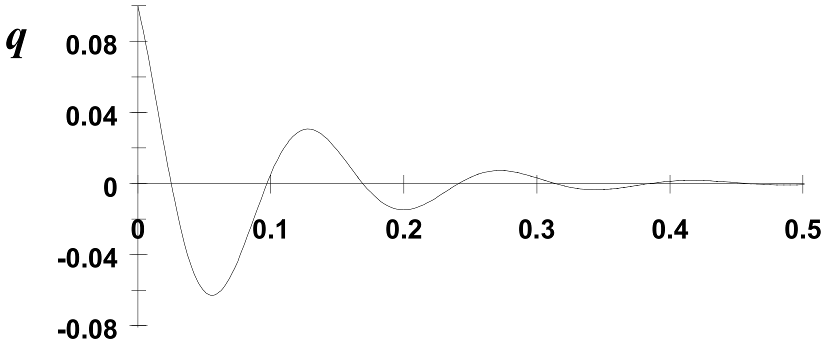



28.24 The second-order equation can be reexpressed as a pair of first-order equations,

$$\frac{dq}{dt} = i$$
$$\frac{di}{dt} = -\frac{R}{L}i - \frac{q}{CL}$$

The parameters can be substituted and the system solved with the 4th-order RK method with $h = 0.005$. A table showing the first few steps and a graph of the entire solution are shown below.



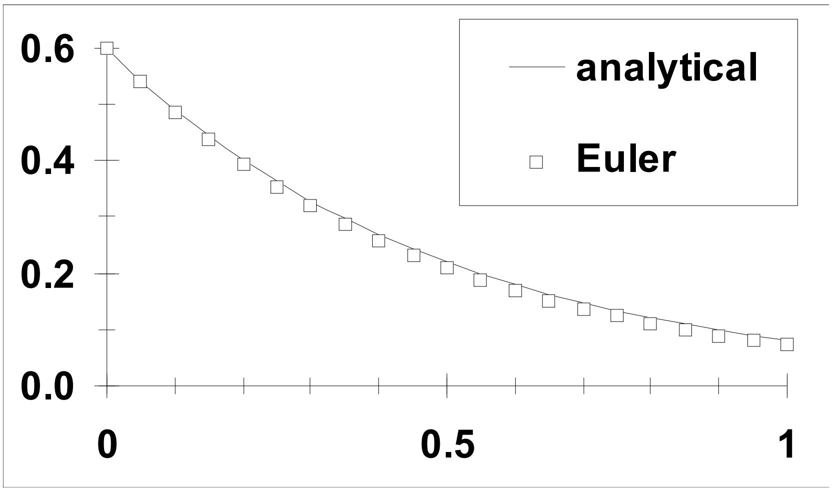

28.25 The equation can be solved analytically as

$$\frac{di}{dt} = -\frac{R}{L}i$$
$$\frac{di}{i} = -\frac{R}{L}dt$$
$$\ln i = -(R / L)t + C$$
$$C = 0.001$$
$$i = 0.001e^{-2t}$$

The numerical solution can be obtained by expressing the equation as

$$\frac{di}{dt} = -2i$$

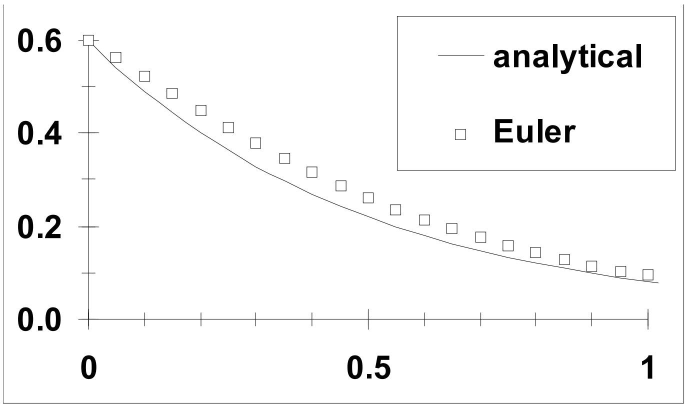
and using Euler's method with $h = 0.05$ to solve for the current. Some selected values are shown, along with a plot of both the analytical and numerical result. A better match would be obtained by using a smaller step or a higher-order method.




28.26 The numerical solution can be obtained by expressing the equation as

$$\frac{di}{dt} = -(-i + i^3)2$$

and using Euler's method with $h = 0.05$ to solve for the current. Some selected values are shown, along with a plot of the numerical result. Note that the table and plot also show the analytical solution for the linear case computed in Prob. 28.19.



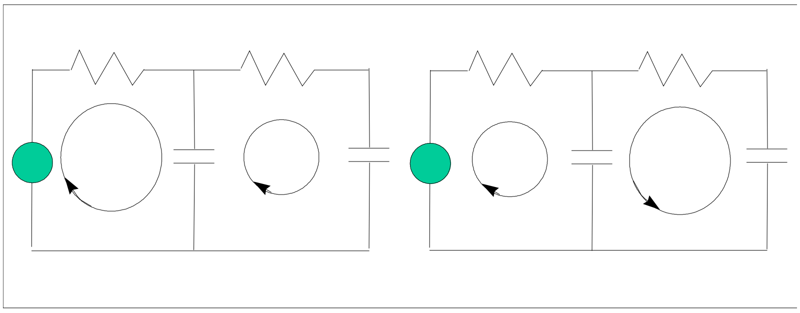
28.27 Using an approach similar to Sec. 28.3, the system can be expressed in matrix form as

$$\begin{bmatrix} 1-\lambda & -1 \\ -1 & 2-\lambda \end{bmatrix} \begin{Bmatrix} i_1 \\ i_2 \end{Bmatrix} = \{0\}$$

A package like MATLAB can be used to evaluate the eigenvalues and eigenvectors as in

```
>> a=[1 -1;-1 2];
>> [v,d]=eig(a)

v =
    0.8507   -0.5257
    0.5257    0.8507

d =
    0.3820         0
         0    2.6180
```

Thus, we can see that the eigenvalues are $\lambda = 0.382$ and 2.618 or natural frequencies of $\omega = 0.618/\sqrt{LC}$ and $1.618/\sqrt{LC}$. The eigenvectors tell us that these correspond to oscillations that coincide (0.8507 0.5257) and which run counter to each other (−0.5257 0.8507).



28.28 The differential equations to be solved are

linear:

$$\frac{d\theta}{dt} = v$$

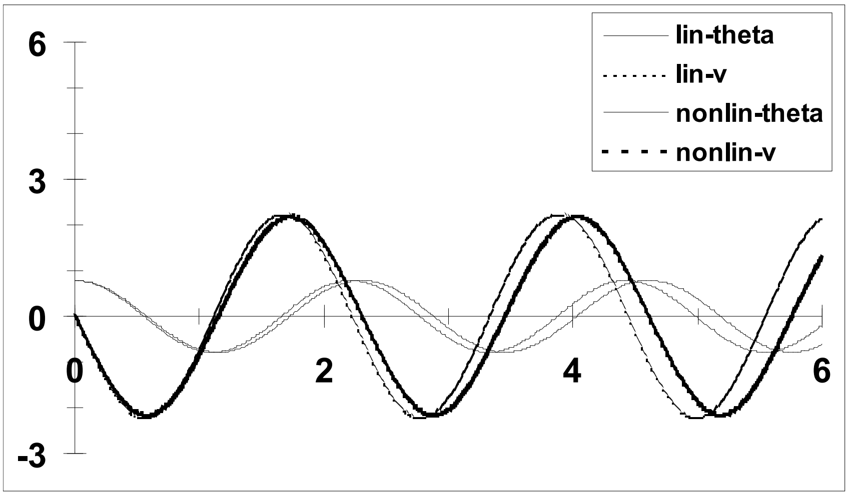
$$\frac{dv}{dt} = -\frac{32.2}{4}\theta$$

nonlinear:

$$\frac{d\theta}{dt} = v$$

$$\frac{dv}{dt} = -\frac{32.2}{4}\sin\theta$$

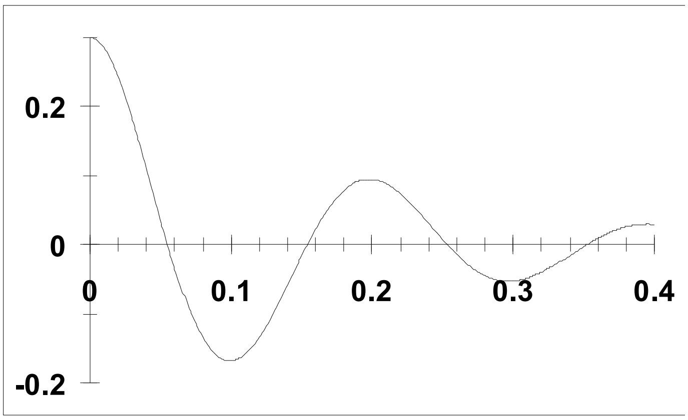
A few steps for the 4$^{th}$-order RK solution of the nonlinear system are contained in the following table and a plot of both solutions is shown below.




28.29 The differential equations to be solved are

$$\frac{dx}{dt} = v$$
$$\frac{dv}{dt} = -\frac{c}{m}v - \frac{k}{m}v$$

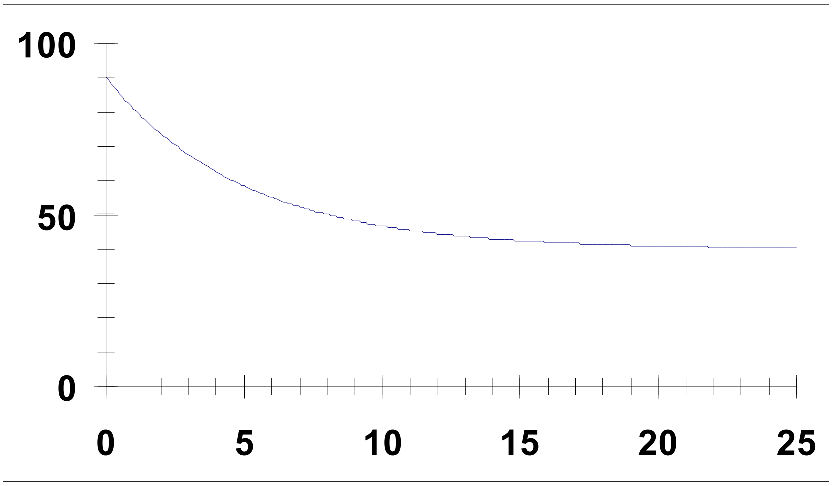
A few steps for the 2nd-order RK solution (Heun without iteration) are shown in the following table and a plot of displacement is shown below.




28.30 The differential equation to be solved is

$$\frac{dT}{dt} = 0.2(40 - T)$$

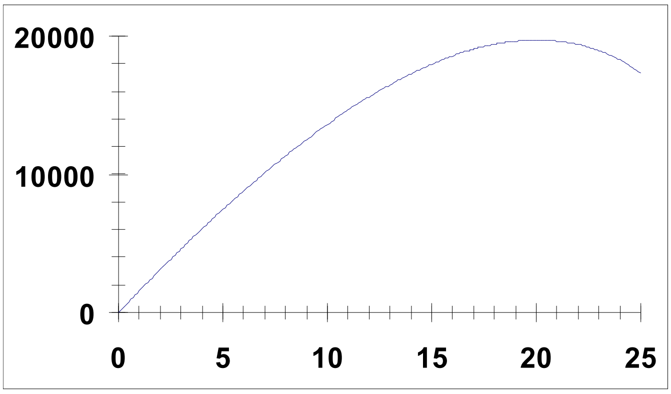
A few steps for the 2nd-order RK solution (Heun without iteration) are shown in the following table and a plot of temperature versus time is shown below. The temperature will drop 95% of the way to the new temperature in 3/0.2 = 15 minutes.




28.31 The differential equation to be solved is

$$\frac{dQ}{dt} = 0.4(10)\frac{100(20-2.5)(20-t)}{100-2.5t}$$

A few steps for the 2nd-order RK solution (Heun without iteration) are shown in the following table and a plot of heat flow versus time is shown below.




28.32 The differential equations to be solved are

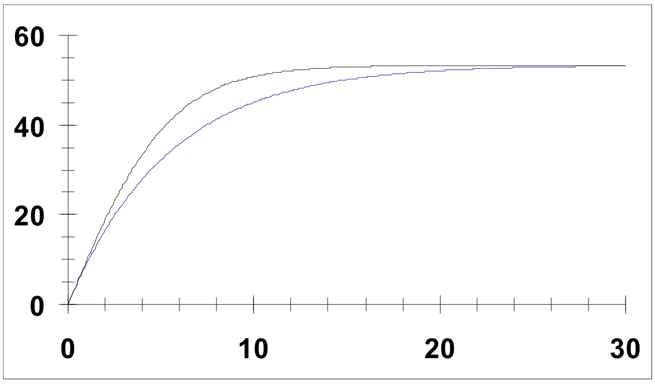
nonlinear:

$$\frac{dv}{dt} = 9.8 - \frac{0.235}{68.1}v^2$$

linear:

$$\frac{dv}{dt} = 9.8 - \frac{12.5}{68.1}v$$

A few steps for the solution (Euler) are shown in the following table, which also includes the analytical solution from Example 1.1. A plot of the result is also shown below. Note, the nonlinear solution is the bolder line.

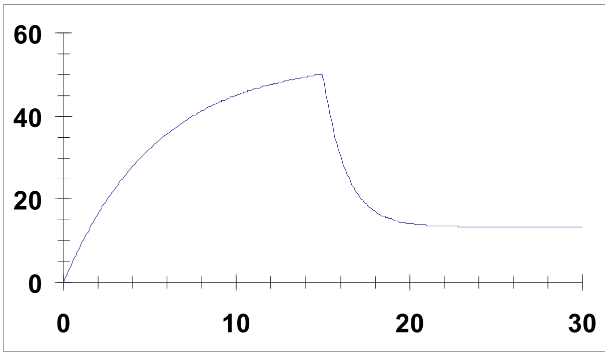



28.33 The differential equations to be solved are

$t < 15$ s: $$\frac{dv}{dt} = 9.8 - \frac{12.5}{68.1}v$$

$t \geq 15$ s: $$\frac{dv}{dt} = 9.8 - \frac{50}{68.1}v$$

The first few steps for the solution (Euler) are shown in the following table, along with the steps when the parachute opens. A plot of the result is also shown below.



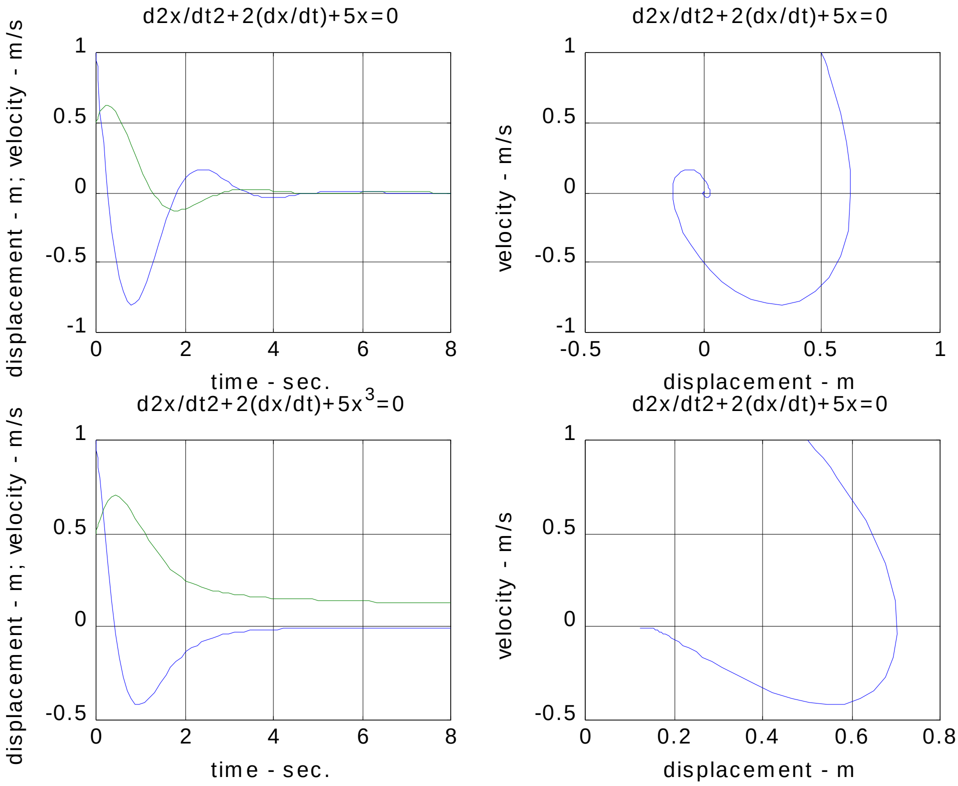

28.34

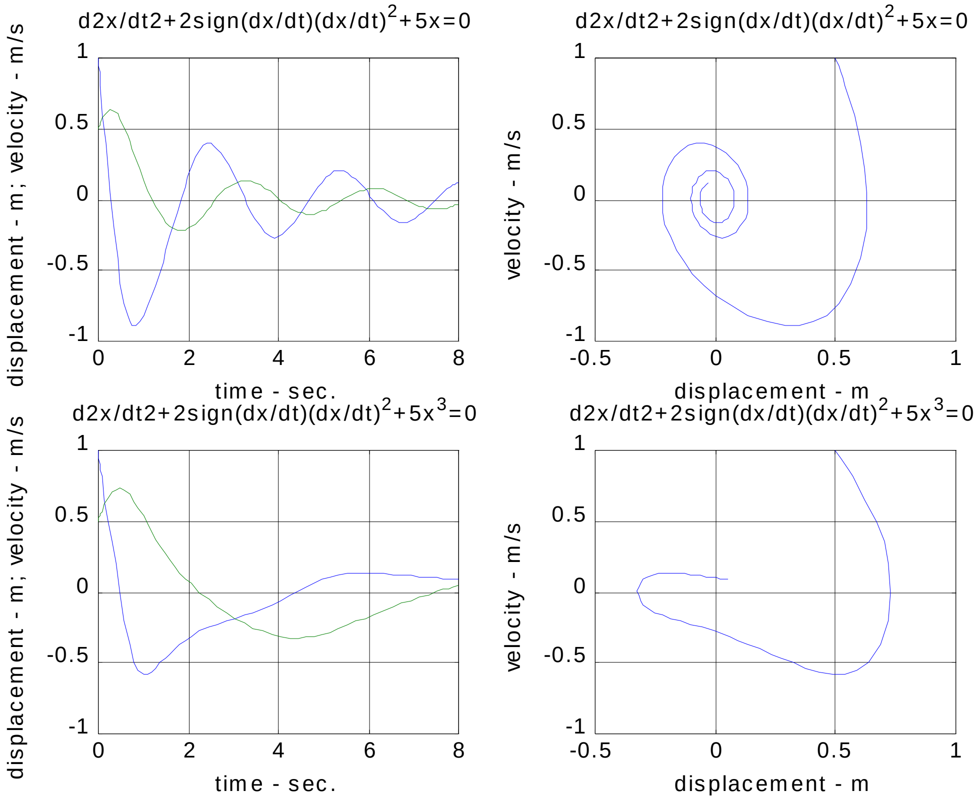
```
%Damped spring mass system
%mass:  m=1 kg
%damping, nonlinear: a sgn(dx/dt) (dx/dt)^2, a=2 N/(m/s)^2
%spring, nonlinear: bx^3, b=5 N/m^3
% MATLAB 5 version

%Independent Variable t, tspan=[tstart tstop]
%initial conditions [x(1)=velocity, x(2)=displacement];

t0=0;
tf=8;
tspan=[0 8]; ic=[1 0.5];

% a) linear solution
[t,x]=ode45('kc',tspan,ic);
subplot(221)
plot(t,x); grid; xlabel('time - sec.');
ylabel('displacement - m; velocity - m/s');
title('d2x/dt2+2(dx/dt)+5x=0')
subplot(222)
%Phase-plane portrait
plot(x(:,2),x(:,1)); grid;
xlabel('displacement - m'); ylabel('velocity - m/s');
title('d2x/dt2+2(dx/dt)+5x=0');
```

```
% b) nonlinear spring
[t,x]=ode45('nlk',tspan,ic);
subplot(223)
plot(t,x); grid;
xlabel('time - sec.');  ylabel('displacement - m; velocity - m/s');
title('d2x/dt2+2(dx/dt)+5x^3=0')
%Phase-plane portrait
subplot(224)
plot(x(:,2),x(:,1)); grid;
xlabel('displacement - m'); ylabel('velocity - m/s');
title('d2x/dt2+2(dx/dt)+5x=0');
pause

% c) nonlinear damping
[t,x]=ode45('nlc',tspan,ic);
subplot(221)
plot(t,x); grid;
xlabel('time - sec.');  ylabel('displacement - m; velocity - m/s');
title('d2x/dt2+2sign(dx/dt)(dx/dt)^2+5x=0')
%Phase-plane portrait
subplot(222)
plot(x(:,2),x(:,1)); grid;
xlabel('displacement - m'); ylabel('velocity - m/s');
title('d2x/dt2+2sign(dx/dt)(dx/dt)^2+5x=0');

% d) nonlinear damping and spring
[t,x]=ode45('nlck',tspan,ic);
subplot(223)
plot(t,x); grid;
xlabel('time - sec.');  ylabel('displacement - m; velocity - m/s');
title('d2x/dt2+2sign(dx/dt)(dx/dt)^2+5x^3=0')
%Phase-plane portrait
subplot(224)
plot(x(:,2),x(:,1)); grid;
xlabel('displacement - m'); ylabel('velocity - m/s');
title('d2x/dt2+2sign(dx/dt)(dx/dt)^2+5x^3=0');
```

Functions:

```
%Damped spring mass system - m d2x/dt2 + c dx/dt + k x =0
%mass:     m=1 kg
%    linear-    c=2 N/(m/s)
%    linear-    k=5 N/m
%x(1)=velocity, x(2)=displacement

function dx=kc(t,x);
dx=[-2*x(1)-5*x(2); x(1)]

%Damped spring mass system - m d2x/dt2 + c dx/dt + k x =0
%mass:     m=1 kg
%damping:  linear-    c=2  N/(m/s)
%spring:   nonlinear- kx=bx^3,   b=5 N/m^3

function dx=nlk(t,x);
dx=[-2*x(1)-5*x(2).*x(2).*x(2); x(1)]

%Damped spring mass system - m d2x/dt2 + c dx/dt + k x =0
%mass:     m=1 kg
%damping:  nonlinear- c dx/dt = a sgn(dx/dt) (dx/dt)^2,  a=2 N/(m/s)^2
%spring:   linear-    kx=5x
%x(1)=velocity, x(2)=dispacement

function dx=nlc(t,x);
dx(1)=-2*sign(x(1))*x(1)*x(1)-5*x(2);
dx(2)= x(1);

%Damped spring mass system - m d2x/dt2 + c dx/dt + k x =0
%mass:     m=1 kg
%damping:  nonlinear- c dx/dt = a sgn(dx/dt) (dx/dt)^2,  a=2 N/(m/s)^2
%spring:   nonlinear- k x = bx^3,   b=5 N/m^3
%x(1)=velocity, x(2)=dispacement

function dx=nlck(t,x);
dx=[-2*sign(x(1)).*x(1).*x(1)-5*x(2).*x(2).*x(2); x(1)]
```



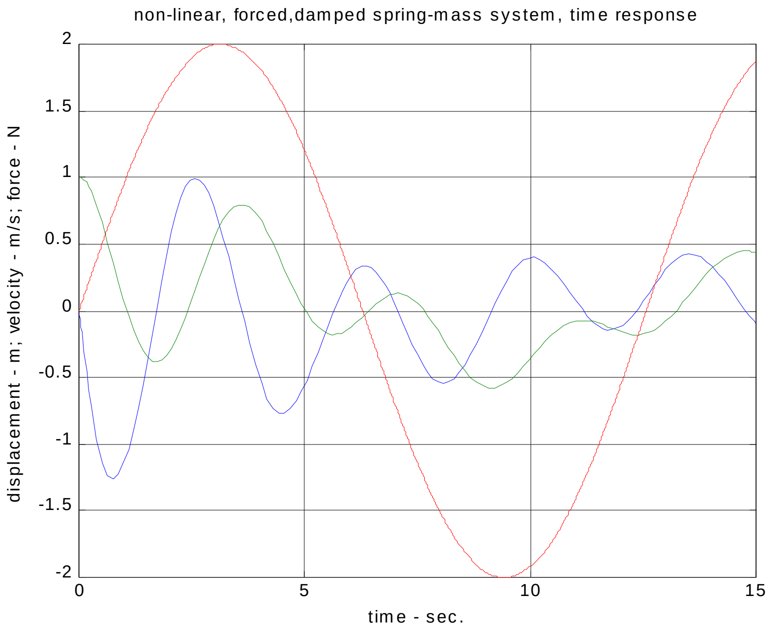

28.35

```
%Forced damped spring-mass system w/ material damping
%mass:  m=2 kg
%damping, nonlinear material: b sgn(dx/dt) abs(x), b=1 N/m
%spring, linear: kx = 6x
%forcing function: F=Fo(sin(wt)),  Fo=2 N, w=0.5 rad/s
```

```
% MATLAB 5 version
%Independent Variable t, tspan=[tstart tstop]
%initial conditions [x(1)=velocity, x(2)=displacement];

tspan=[0 15]; ic=[0 1];
[t,x]=ode45('nlF',tspan,ic);

ts=0:.01:15;
Sin=2*sin(0.5*ts);
plot(t,x,ts,Sin,'--'); grid; xlabel('time - sec.');
    ylabel('displacement - m; velocity - m/s; force - N');
    title('non-linear, forced,damped spring-mass system, time response')
```

Function 'nlF':

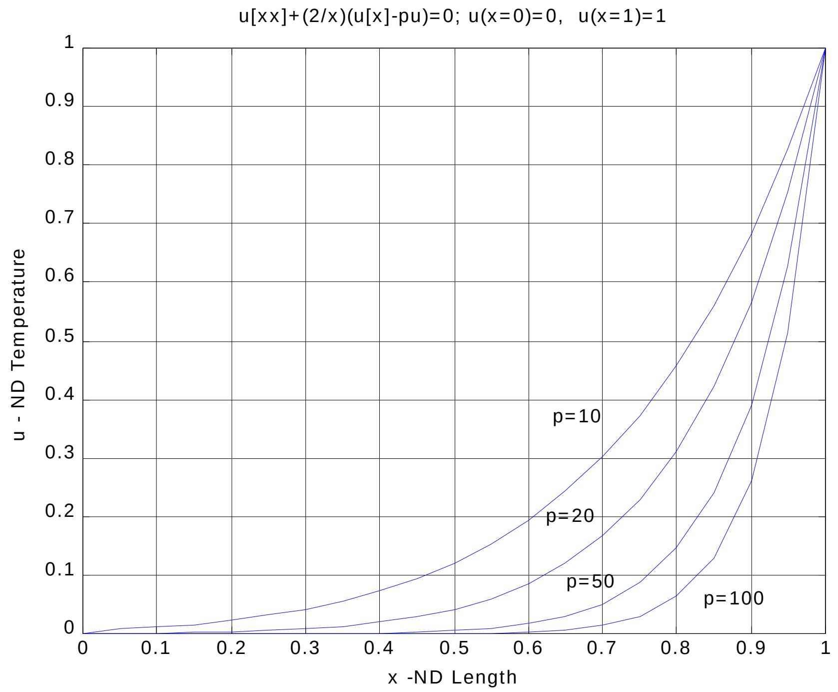
```
%Forced damped spring-mass system w/ material damping
%mass:  m=2 kg
%damping, nonlinear air: b sgn(dx/dt) (dx/dt)^2, b=1 N/m
%spring, linear: kx = 6x
%forcing function: F=Fo(sin(wt),  Fo=2 N, w=0.5 rad/s
% x(1)= velocity, x(2)= displacement

function dx=nlF(t,x);
dx=[-0.5*sign(x(1)).*x(1).*x(1)-3*x(2)+sin(0.5*t); x(1)]
```




28.36

```
%   ODE Boundary Value Problem
%   Tapered conical cooling fin
%      u[xx]+(2/x)(u[x]-pu)=0
%      BC. u(x=0)=0  u(x=1)=1
%   i=spatial index, from 1 to R
%        numbering for points is i=1 to i=R for (R-1) dx spaces
%   u(i=1)=0 and u(i=R)=1

R=21;
%Constants
dx=1/(R-1);
dx2=dx*dx;
%Parameters
p(1)=10; p(2)=20; p(3)=50; p(4)=100;
%sizing matrices
u=zeros(1,R); x=zeros(1,R);
a=zeros(1,R); b=zeros(1,R); c=zeros(1,R); d=zeros(1,R);
ba=zeros(1,R); ga=zeros(1,R);
%Independent Variable
x=0:dx:1;
%Boundary Conditions
u(1)=0; u(R)=1;
```

```
for k=1:4;
  %/Coefficients
  b(2)=-2-2*p(k)*dx2/dx;
  c(2)=2;
  for i=3:R-2,
    a(i)=1-dx/(dx*(i-1));
    b(i)=-2-2*p(k)*dx2/(dx*(i-1));
    c(i)=1+1/(i-1);
  end
  a(R-1)=1-dx/(dx*(R-2));
  b(R-1)=-2-2*p(k)*dx2/(dx*(R-2));
  d(R-1)=-(1+1/(R-2));
  %Solution by Thomas Algorithm
  ba(2)=b(2);
  ga(2)=d(2)/b(2);
  for i=3:R-1,
    ba(i)=b(i)-a(i)*c(i-1)/ba(i-1);
    ga(i)=(d(i)-a(i)*ga(i-1))/ba(i);
  end
  %back substitution step
  u(R-1)=ga(R-1);
  for i=R-2:-1:2,
    u(i)=ga(i)-c(i)*u(i+1)/ba(i);
  end
  %Plot
  plot(x,u)
  title('u[xx]+(2/x)(u[x]-pu)=0; u(x=0)=0,  u(x=1)=1')
  xlabel('x -ND Length')
  ylabel('u - ND Temperature')
  hold on
end
grid
hold off
gtext('p=10');gtext('p=20');gtext('p=50');gtext('p=100');
```




| *t* | *v* | *dvdt* |
|---|---|---|
| 0 | 0 | 9.8 |
| 0.1 | 0.98 | 9.620117 |
| 0.2 | 1.942012 | 9.443537 |
| 0.3 | 2.886365 | 9.270197 |
| 0.4 | 3.813385 | 9.100039 |
| 0.5 | 4.723389 | 8.933005 |

•
•
•

| | | |
|---|---|---|
| 14.9 | 50.01245 | 0.620036 |
| 15 | 50.07445 | -26.9654 |
| 15.1 | 47.37791 | -24.9855 |
| 15.2 | 44.87936 | -23.1511 |
| 15.3 | 42.56425 | -21.4513 |
| 15.4 | 40.41912 | -19.8763 |
| 15.5 | 38.43149 | -18.417 |

| | ***linear*** | | ***nonlinear*** | | ***analytical*** |
|---|---|---|---|---|---|
| ***t*** | ***v*** | ***dvdt*** | ***v*** | ***dvdt*** | |
| 0 | 0 | 9.8 | 0 | 9.8 | 0 |
| 0.1 | 0.98 | 9.620117 | 0.98 | 9.796686 | 0.971061 |
| 0.2 | 1.942012 | 9.443537 | 1.959669 | 9.786748 | 1.92446 |
| 0.3 | 2.886365 | 9.270197 | 2.938343 | 9.770206 | 2.860518 |
| 0.4 | 3.813385 | 9.100039 | 3.915364 | 9.747099 | 3.779552 |
| 0.5 | 4.723389 | 8.933005 | 4.890074 | 9.717481 | 4.681871 |

| ***t*** | ***T*** | ***k11*** | ***T-end*** | ***k21*** | ***phi1*** |
|---|---|---|---|---|---|
| 0 | 0 | 1598 | 159.8 | 1602.005 | 1600.003 |
| 0.1 | 160.0003 | 1593.995 | 319.3997 | 1598 | 1595.997 |
| 0.2 | 319.6 | 1589.97 | 478.597 | 1593.975 | 1591.972 |
| 0.3 | 478.7972 | 1585.924 | 637.3897 | 1589.929 | 1587.927 |
| 0.4 | 637.5899 | 1581.859 | 795.7758 | 1585.863 | 1583.861 |
| 0.5 | 795.976 | 1577.772 | 953.7532 | 1581.777 | 1579.774 |

| ***t*** | ***T*** | ***k11*** | ***T-end*** | ***k21*** | ***phi1*** |
|---|---|---|---|---|---|
| 0 | 90 | -10 | 89 | -9.8 | -9.9 |
| 0.1 | 89.01 | -9.802 | 88.0298 | -9.60596 | -9.70398 |
| 0.2 | 88.0396 | -9.60792 | 87.07881 | -9.41576 | -9.51184 |
| 0.3 | 87.08842 | -9.41768 | 86.14665 | -9.22933 | -9.32351 |
| 0.4 | 86.15607 | -9.23121 | 85.23295 | -9.04659 | -9.1389 |
| 0.5 | 85.24218 | -9.04844 | 84.33733 | -8.86747 | -8.95795 |

| ***x*** | ***v*** | ***k11*** | ***k12*** | ***x*** | ***v*** | ***k21*** | ***k22*** | ***phi1*** | ***phi2*** |
|---|---|---|---|---|---|---|---|---|---|
| 0.3 | 0 | 0 | -312.5 | 0.3 | -0.3125 | -0.3125 | -308.854 | -0.15625 | -310.677 |
| 0.299844 | -0.31068 | -0.31068 | -308.713 | 0.299533 | -0.61939 | -0.61939 | -304.787 | -0.46503 | -306.75 |
| 0.299379 | -0.61743 | -0.61743 | -304.65 | 0.298761 | -0.92208 | -0.92208 | -300.452 | -0.76975 | -302.551 |
| 0.298609 | -0.91998 | -0.91998 | -300.318 | 0.297689 | -1.2203 | -1.2203 | -295.856 | -1.07014 | -298.087 |
| 0.297539 | -1.21806 | -1.21806 | -295.726 | 0.296321 | -1.51379 | -1.51379 | -291.007 | -1.36593 | -293.366 |

| ***t*** | ***thet*** | ***v*** | ***k11*** | ***k12*** | ***thet*** | ***v*** | ***k21*** | ***k22*** | ***thet*** | ***v*** | ***k31*** | ***k32*** | ***thet*** | ***v*** | ***k41*** | ***k42*** | ***phi1*** | ***phi2*** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0.785 | 0.000 | 0.000 | -5.692 | 0.785 | -0.028 | -0.028 | -5.692 | 0.785 | -0.028 | -0.028 | -5.691 | 0.785 | -0.057 | -0.057 | -5.691 | -0.028 | -5.692 |
| 0.01 | 0.785 | -0.057 | -0.057 | -5.691 | 0.785 | -0.085 | -0.085 | -5.689 | 0.785 | -0.085 | -0.085 | -5.688 | 0.784 | -0.114 | -0.114 | -5.686 | -0.085 | -5.688 |
| 0.02 | 0.784 | -0.114 | -0.114 | -5.686 | 0.784 | -0.142 | -0.142 | -5.682 | 0.784 | -0.142 | -0.142 | -5.682 | 0.783 | -0.171 | -0.171 | -5.678 | -0.142 | -5.682 |
| 0.03 | 0.783 | -0.171 | -0.171 | -5.678 | 0.782 | -0.199 | -0.199 | -5.673 | 0.782 | -0.199 | -0.199 | -5.672 | 0.781 | -0.227 | -0.227 | -5.666 | -0.199 | -5.672 |
| 0.04 | 0.781 | -0.227 | -0.227 | -5.666 | 0.780 | -0.256 | -0.256 | -5.660 | 0.780 | -0.256 | -0.256 | -5.659 | 0.778 | -0.284 | -0.284 | -5.652 | -0.256 | -5.659 |

| ***t*** | ***(analytical)*** | ***(Euler)*** | ***didt*** |
|---|---|---|---|
| 0 | 0.600000 | 0.600000 | -0.768000 |
| 0.05 | 0.542902 | 0.561600 | -0.768949 |
| 0.1 | 0.491238 | 0.523153 | -0.759943 |
| 0.15 | 0.444491 | 0.485155 | -0.741923 |
| 0.2 | 0.402192 | 0.448059 | -0.716216 |
| 0.25 | 0.363918 | 0.412248 | -0.684375 |

| t | (analytical) | (Euler) | didt |
|---|---|---|---|
| 0 | 0.600000 | 0.600000 | -1.200000 |
| 0.05 | 0.542902 | 0.540000 | -1.080000 |
| 0.1 | 0.491238 | 0.486000 | -0.972000 |
| 0.15 | 0.444491 | 0.437400 | -0.874800 |
| 0.2 | 0.402192 | 0.393660 | -0.787320 |
| 0.25 | 0.363918 | 0.354294 | -0.708588 |

| t | i | q | k11 | k12 | imid | qmid | k21 | k22 | imid | qmid | k31 | k32 | iend | qend | k41 | k42 | phi1 | phi2 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | -3.282 | 0.100 | -134.37 | -3.282 | -3.617 | 0.092 | -111.24 | -3.617 | -3.560 | 0.091 | -110.7 | -3.560 | -3.835 | 0.082 | -87.70 | -3.835 | -111.0 | -3.578 |
| 0.005 | -3.837 | 0.082 | -87.485 | -3.837 | -4.055 | 0.073 | -63.93 | -4.055 | -3.996 | 0.072 | -64.01 | -3.996 | -4.157 | 0.062 | -41.12 | -4.157 | -64.08 | -4.016 |
| 0.01 | -4.157 | 0.062 | -40.917 | -4.157 | -4.259 | 0.052 | -18.09 | -4.259 | -4.202 | 0.051 | -18.72 | -4.202 | -4.251 | 0.041 | 2.976 | -4.251 | -18.59 | -4.222 |
| 0.015 | -4.250 | 0.041 | 3.159 | -4.250 | -4.242 | 0.030 | 24.25 | -4.242 | -4.189 | 0.030 | 23.16 | -4.189 | -4.134 | 0.020 | 42.736 | -4.134 | 23.452 | -4.208 |
| 0.02 | -4.133 | 0.020 | 42.892 | -4.133 | -4.025 | 0.010 | 61.41 | -4.025 | -3.979 | 0.010 | 59.95 | -3.979 | -3.833 | 0.000 | 76.688 | -3.833 | 60.383 | -3.996 |

| t | i | q | k11 | k12 | imid | qmid | k21 | k22 | imid | qmid | k31 | k32 | iend | qend | k41 | k42 | phi1 | phi2 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0934 | 0.0000 | 0.0047 | 0.0000 | 0.0932 | 0.0047 | 0.0093 | 0.0005 | 0.1837 | 0.0093 | 0.0928 | 0.0031 |
| 0.1 | 0.0093 | 0.0003 | 0.1843 | 0.0093 | 0.0185 | 0.0008 | 0.2729 | 0.0185 | 0.0229 | 0.0012 | 0.2709 | 0.0229 | 0.0364 | 0.0026 | 0.3533 | 0.0364 | 0.2709 | 0.0214 |
| 0.2 | 0.0364 | 0.0025 | 0.3539 | 0.0364 | 0.0541 | 0.0043 | 0.4311 | 0.0541 | 0.0579 | 0.0052 | 0.4273 | 0.0579 | 0.0791 | 0.0082 | 0.4953 | 0.0791 | 0.4277 | 0.0566 |
| 0.3 | 0.0791 | 0.0081 | 0.4958 | 0.0791 | 0.1039 | 0.0121 | 0.5555 | 0.1039 | 0.1069 | 0.0133 | 0.5504 | 0.1069 | 0.1342 | 0.0188 | 0.5985 | 0.1342 | 0.5510 | 0.1058 |
| 0.4 | 0.1342 | 0.0187 | 0.5989 | 0.1342 | 0.1642 | 0.0254 | 0.6361 | 0.1642 | 0.1660 | 0.0269 | 0.6300 | 0.1660 | 0.1972 | 0.0353 | 0.6538 | 0.1972 | 0.6308 | 0.1653 |

| t | p | k1 | pend | k2 | phi |
|---|---|---|---|---|---|
| 0 | 5000 | 750 | 5375 | 786.0938 | 768.0469 |
| 0.5 | 5384.023 | 786.9276 | 5777.487 | 821.7039 | 804.3157 |
| 1 | 5786.181 | 822.4373 | 6197.4 | 855.4023 | 838.9198 |
| 1.5 | 6205.641 | 856.0284 | 6633.655 | 886.6772 | 871.3528 |
| • | | | | | |
| • | | | | | |
| • | | | | | |
| 18 | 18480.96 | 280.733 | 18621.33 | 256.7271 | 268.7301 |
| 18.5 | 18615.33 | 257.7616 | 18744.21 | 235.3885 | 246.575 |
| 19 | 18738.61 | 236.3663 | 18856.8 | 215.5715 | 225.9689 |
| 19.5 | 18851.6 | 216.4921 | 18959.84 | 197.2119 | 206.852 |
| 20 | 18955.02 | 198.0754 | 19054.06 | 180.2397 | 189.1575 |

| t | p | k1 | pend | k2 | phi |
|---|---|---|---|---|---|
| 0 | 5000 | 350 | 5175 | 362.25 | 356.125 |
| 0.5 | 5178.063 | 362.4644 | 5359.295 | 375.1506 | 368.8075 |
| 1 | 5362.466 | 375.3726 | 5550.153 | 388.5107 | 381.9417 |
| 1.5 | 5553.437 | 388.7406 | 5747.807 | 402.3465 | 395.5436 |
| 2 | 5751.209 | 402.5846 | 5952.501 | 416.6751 | 409.6299 |
| • | | | | | |
| • | | | | | |
| • | | | | | |
| 18 | 17622.69 | 1233.588 | 18239.48 | 1276.764 | 1255.176 |
| 18.5 | 18250.28 | 1277.52 | 18889.04 | 1322.233 | 1299.876 |
| 19 | 18900.22 | 1323.015 | 19561.72 | 1369.321 | 1346.168 |
| 19.5 | 19573.3 | 1370.131 | 20258.37 | 1418.086 | 1394.108 |
| 20 | 20270.36 | 1418.925 | 20979.82 | 1468.587 | 1443.756 |

| t | c1 | c2 | c3 | dc1/dt | dc2/dt | dc3/dt |
|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 100 | 21.93515 | 0 | -43.8703 |
| 0.1 | 2.193515 | 0 | 95.61297 | 19.41211 | 0.252723 | -40.9834 |

| | | | | | | |
|---|---|---|---|---|---|---|
| 0.2 | 4.134726 | 0.025272 | 91.51463 | 17.13425 | 0.473465 | -38.3338 |
| 0.3 | 5.848151 | 0.072619 | 87.68125 | 15.07861 | 0.66542 | -35.9004 |
| 0.4 | 7.356012 | 0.139161 | 84.09121 | 13.22439 | 0.83148 | -33.664 |
| 0.5 | 8.678451 | 0.222309 | 80.72481 | 11.55268 | 0.974263 | -31.607 |

| *z* | *y* | *w* | *dydz* | *dwdz* |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0.0036 |
| 1 | 0 | 0.0036 | 0.0036 | 0.003364 |
| 2 | 0.0036 | 0.006964 | 0.006964 | 0.003136 |
| 3 | 0.010564 | 0.0101 | 0.0101 | 0.002916 |
| 4 | 0.020664 | 0.013016 | 0.013016 | 0.002704 |
| 5 | 0.03368 | 0.01572 | 0.01572 | 0.0025 |
| • | | | | |
| • | | | | |
| • | | | | |
| 26 | 0.676 | 0.0377 | 0.0377 | 0.000064 |
| 27 | 0.7137 | 0.037764 | 0.037764 | 0.000036 |
| 28 | 0.751464 | 0.0378 | 0.0378 | 0.000016 |
| 29 | 0.789264 | 0.037816 | 0.037816 | 0.000004 |
| 30 | 0.82708 | 0.03782 | 0.03782 | 0 |

| *z* | *f(z)* | *y* | *w* | *dydz* | *dwdz* |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 31.18357 | 0 | 0 | 0 | 0.002098 |
| 2 | 50.0099 | 0 | 0.002098 | 0.002098 | 0.003137 |
| 3 | 61.40481 | 0.002098 | 0.005235 | 0.005235 | 0.003581 |
| 4 | 68.08252 | 0.007333 | 0.008816 | 0.008816 | 0.003682 |
| 5 | 71.65313 | 0.016148 | 0.012498 | 0.012498 | 0.003583 |
| • | | | | | |
| • | | | | | |
| • | | | | | |
| 26 | 29.63907 | 0.700979 | 0.040713 | 0.040713 | 3.79E-05 |
| 27 | 27.89419 | 0.741693 | 0.040751 | 0.040751 | 2.01E-05 |
| 28 | 26.24164 | 0.782444 | 0.040771 | 0.040771 | 8.4E-06 |
| 29 | 24.67818 | 0.823216 | 0.04078 | 0.04078 | 1.97E-06 |
| 30 | 23.20033 | 0.863995 | 0.040782 | 0.040782 | 0 |

| *z* | *y* | *w* | *dydz* | *dwdz* |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0.0036 |
| 1 | 0 | 0.0036 | 0.0036 | 0.003364 |
| 2 | 0.0036 | 0.006964 | 0.006964 | 0.003136 |
| 3 | 0.010564 | 0.0101 | 0.0101 | 0.002916 |
| 4 | 0.020664 | 0.013016 | 0.013016 | 0.002704 |
| 5 | 0.03368 | 0.01572 | 0.01572 | 0.0025 |
| • | | | | |
| • | | | | |
| • | | | | |
| 26 | 0.676 | 0.0377 | 0.0377 | 0.000064 |
| 27 | 0.7137 | 0.037764 | 0.037764 | 0.000036 |
| 28 | 0.751464 | 0.0378 | 0.0378 | 0.000016 |
| 29 | 0.789264 | 0.037816 | 0.037816 | 0.000004 |
| 30 | 0.82708 | 0.03782 | 0.03782 | 0 |

| t | x | y | Z |
|---|---|---|---|
| 0 | 5 | 5 | 5 |
| 0.1 | 9.78147 | 17.07946 | 10.43947 |
| 0.2 | 17.70297 | 20.8741 | 35.89688 |

| | | | |
|---|---|---|---|
| 0.3 | 10.81088 | -2.52924 | 39.30744 |
| 0.4 | 0.549577 | -5.54419 | 28.07461 |
| 0.5 | -3.16461 | -5.84129 | 22.36888 |
| 0.6 | -5.57588 | -8.42037 | 19.92312 |
| 0.7 | -8.88719 | -12.6789 | 22.14149 |
| 0.8 | -11.9142 | -13.43 | 29.80001 |
| 0.9 | -10.6668 | -7.21784 | 33.39903 |
| 1 | -6.84678 | -3.43018 | 29.30716 |

| | | |
|---|---|---|
| 0 | 2 | 3 |
| 0.1 | 1.887095 | 2.935517 |
| 0.2 | 1.787897 | 2.863301 |
| 0.3 | 1.701588 | 2.785107 |
| 0.4 | 1.627287 | 2.702536 |
| 0.5 | 1.564109 | 2.617016 |

| | | |
|---|---|---|
| 0 | 2 | 3 |
| 0.1 | 1.886984 | 2.935308 |
| 0.2 | 1.787729 | 2.862899 |
| 0.3 | 1.701406 | 2.784535 |
| 0.4 | 1.627125 | 2.701821 |
| 0.5 | 1.56399 | 2.616185 |

| *t* | *x* | *y* |
|---|---|---|
| 0 | 2 | 3 |
| 0.1 | 1.88 | 2.94 |
| 0.2 | 1.773968 | 2.870616 |
| 0.3 | 1.681301 | 2.793738 |
| 0.4 | 1.601231 | 2.711153 |
| 0.5 | 1.532907 | 2.624496 |

| t | C | Te |
|---|---|---|
| 0 | 1 | 25 |
| 0.0625 | 0.941218 | 66.18648 |
| 0.125 | 0.885749 | 85.80247 |
| 0.1875 | 0.833497 | 93.93385 |
| 0.25 | 0.784309 | 96.02265 |
| 0.3125 | 0.738024 | 94.99472 |
| 0.375 | 0.694475 | 92.41801 |
| 0.4375 | 0.653506 | 89.12894 |
| 0.5 | 0.614963 | 85.57041 |
| 0.5625 | 0.578703 | 81.97385 |
| 0.625 | 0.54459 | 78.45733 |
| 0.6875 | 0.512497 | 75.07829 |
| 0.75 | 0.482304 | 71.86194 |
| 0.8125 | 0.453896 | 68.81648 |
| 0.875 | 0.427168 | 65.9413 |
| 0.9375 | 0.40202 | 63.23134 |
| 1 | 0.378358 | 60.67946 |
| •<br>•<br>• | | |

| *x* | *A* | *x* | *A* | *x* | *A* | *x* | *A* |
|---|---|---|---|---|---|---|---|
| 0 | 0.1 | | | | | | |
| 0.2 | 0.067208 | 1.2 | 0.009215 | 2.2 | 0.001263 | 3.2 | 0.000166 |
| 0.4 | 0.045169 | 1.4 | 0.006193 | 2.4 | 0.000848 | 3.4 | 0.000106 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 0.6 | 0.030357 | 1.6 | 0.004162 | 2.6 | 0.000569 | 3.6 | 6.23E-05 |
| 0.8 | 0.020402 | 1.8 | 0.002797 | 2.8 | 0.00038 | 3.8 | 2.88E-05 |
| 1 | 0.013712 | 2 | 0.00188 | 3 | 0.000253 | 4 | 0 |

| *t* | *M* | *m* | *s* | *dmdt* |
|---|---|---|---|---|
| 0 | 1000 | 8000 | 8 | 0 |
| 5 | 997.5 | 8000 | 8.02005 | -0.2005 |
| 10 | 995 | 7998.997 | 8.039193 | -0.39193 |
| 15 | 992.5 | 7997.038 | 8.057469 | -0.57469 |
| 20 | 990 | 7994.164 | 8.074914 | -0.74914 |
| 25 | 987.5 | 7990.419 | 8.091563 | -0.91563 |

| *t* | *cin* | *c* | *k1* | *cend* | *cin-end* | *k2* | *phi* |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 10 | -0.5 | 9 | 9.063462 | 0.003173 | -0.24841 |
| 2 | 9.063462 | 9.503173 | -0.02199 | 9.459202 | 16.484 | 0.35124 | 0.164627 |
| 4 | 16.484 | 9.832427 | 0.332579 | 10.49758 | 22.55942 | 0.603092 | 0.467835 |
| 6 | 22.55942 | 10.7681 | 0.589566 | 11.94723 | 27.53355 | 0.779316 | 0.684441 |
| 8 | 27.53355 | 12.13698 | 0.769829 | 13.67664 | 31.60603 | 0.89647 | 0.833149 |
| 10 | 31.60603 | 13.80328 | 0.890138 | 15.58355 | 34.94029 | 0.967837 | 0.928987 |

| ***t*** | ***c1*** | ***c2*** | ***c3*** | ***c4*** | ***c5*** | ***dc1dt*** | ***dc2dt*** | ***dc3dt*** | ***dc4dt*** | ***dc5dt*** |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 1.0000 | 0.0000 | 5.0000 | 0.0000 | 0.0000 |
| 1 | 1.0000 | 0.0000 | 5.0000 | 0.0000 | 0.0000 | 0.8600 | 0.2000 | 3.6250 | 0.5625 | 0.0300 |
| 2 | 1.8600 | 0.2000 | 8.6250 | 0.5625 | 0.0300 | 0.7396 | 0.3320 | 2.6331 | 0.8726 | 0.0600 |
| 3 | 2.5996 | 0.5320 | 11.2581 | 1.4351 | 0.0900 | 0.6361 | 0.4135 | 1.9173 | 1.0176 | 0.0885 |
| 4 | 3.2357 | 0.9455 | 13.1754 | 2.4528 | 0.1785 | 0.5470 | 0.4580 | 1.4004 | 1.0575 | 0.1147 |
| 5 | 3.7827 | 1.4035 | 14.5758 | 3.5102 | 0.2933 | 0.4704 | 0.4758 | 1.0267 | 1.0328 | 0.1380 |
| • | | | | | | | | | | |
| • | | | | | | | | | | |
| • | | | | | | | | | | |
| 76 | 7.1428 | 7.1426 | 18.8311 | 13.0962 | 7.0053 | 0.0000 | 0.0000 | 0.0000 | 0.0018 | 0.0082 |
| 77 | 7.1428 | 7.1426 | 18.8311 | 13.0980 | 7.0135 | 0.0000 | 0.0000 | 0.0000 | 0.0017 | 0.0078 |
| 78 | 7.1428 | 7.1427 | 18.8311 | 13.0997 | 7.0213 | 0.0000 | 0.0000 | 0.0000 | 0.0016 | 0.0073 |
| 79 | 7.1428 | 7.1427 | 18.8311 | 13.1013 | 7.0286 | 0.0000 | 0.0000 | 0.0000 | 0.0015 | 0.0069 |
| 80 | 7.1428 | 7.1427 | 18.8311 | 13.1028 | 7.0354 | 0.0000 | 0.0000 | 0.0000 | 0.0014 | 0.0064 |

| *t* | *c* | *k1* | *cmid* | *k2* | *cmid* | *k3* | *cend* | *k4* | *phi* |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 10 | 2 | 20 | 1.5 | 17.5 | 1.625 | 26.25 | 1.1875 | 1.572917 |
| 10 | 25.72917 | 1.213542 | 31.79688 | 0.910156 | 30.27995 | 0.986003 | 35.58919 | 0.72054 | 0.9544 |
| 20 | 35.27317 | 0.736342 | 38.95487 | 0.552256 | 38.03445 | 0.598278 | 41.25594 | 0.437203 | 0.579102 |
| 30 | 41.06419 | 0.446791 | 43.29814 | 0.335093 | 42.73965 | 0.363017 | 44.69436 | 0.265282 | 0.351382 |
| 40 | 44.57801 | 0.2711 | 45.93351 | 0.203325 | 45.59463 | 0.220268 | 46.78069 | 0.160965 | 0.213208 |
| 50 | 46.71009 | 0.164495 | 47.53257 | 0.123371 | 47.32695 | 0.133652 | 48.04662 | 0.097669 | 0.129369 |

| *t* | *c* | *k1* | *c* | *k2* | *phi* |
|---|---|---|---|---|---|
| 0 | 10 | 2 | 30 | 1 | 1.5 |
| 10 | 25 | 1.25 | 37.5 | 0.625 | 0.9375 |
| 20 | 34.375 | 0.78125 | 42.1875 | 0.390625 | 0.585938 |
| 30 | 40.23438 | 0.488281 | 45.11719 | 0.244141 | 0.366211 |
| 40 | 43.89648 | 0.305176 | 46.94824 | 0.152588 | 0.228882 |
| 50 | 46.1853 | 0.190735 | 48.09265 | 0.095367 | 0.143051 |

# CHAPTER 32

32.1 First equation

$$6.075c_0 - 3.2c_1 = 262.5$$

Middle equations ($i$ = 1 to 8)

$$-2.1c_{i-1} + 3.45c_i - 1.1c_{i+1} = 0$$

Last equation

$$-3.2c_8 + 3.45c_9 = 0$$

The solution is

32.2 Element equation: (See solution for Prob. 31.4 for derivation of element equation.)

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{Bmatrix} c_1 \\ c_2 \end{Bmatrix} = \begin{Bmatrix} b_1 \\ b_2 \end{Bmatrix}$$

$$a_{11} = \frac{2}{2.5} - \frac{1}{2} + \frac{0.2}{2}(2.5) = 0.55 \qquad a_{12} = \frac{-2}{2.5} + \frac{1}{2} = -0.3$$

$$a_{21} = \frac{-2}{2.5} - \frac{1}{2} = -1.3 \qquad a_{22} = \frac{2}{2.5} + \frac{1}{2} + \frac{0.2}{2}(2.5) = 1.55$$

$$b_1 = -2\frac{dc}{dx}(x_1) \qquad b_2 = 2\frac{dc}{dx}(x_2)$$

Assembly:

$$\begin{bmatrix} 0.55 & -0.3 & & & \\ -1.3 & 2.1 & -0.3 & & \\ & -1.3 & 2.1 & -0.3 & \\ & & -1.3 & 2.1 & -0.3 \\ & & & -1.3 & 1.55 \end{bmatrix} \begin{Bmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \\ c_4 \end{Bmatrix} = \begin{Bmatrix} -\frac{dc}{dx}(x_1) \\ 0 \\ 0 \\ 0 \\ \frac{dc}{dx}(x_2) \end{Bmatrix}$$

Boundary conditions:

Inlet:

$$Uc_{\text{in}} = Uc_0 - D\frac{dc}{dx}(0)$$

$$\frac{dc}{dx}(0) = \frac{Uc_0 - Uc_{\text{in}}}{D}$$

Substitute into first equation

$$0.55c_0 - 0.3c_1 = 100 - c_0$$
$$1.55c_0 - 0.3c_1 = 100$$

Outlet:

$$\frac{dc}{dx}(10) = 0$$

<u>Solution:</u>

$c_0 = 74.4$ $c_1 = 51.08$ $c_2 = 35.15$ $c_3 = 24.72$ $c_2 = 20.74$

32.3 According to Fick's first law, the diffusive flux is

$$J(x) = -D\frac{dc}{dx}(x)$$

where $J(x)$ = flux at position $x$. If $c$ has units of g/m$^3$, $D$ has units of m$^2$/d and $x$ is measured in m, flux has units of g/m$^2$/d. In addition, there will be an advective flux which can be calculated as

$$J(x) = Uc(x)$$

Finite divided differences can be used to approximate the derivatives. For example, for the point at the beginning of the tank, a forward difference can be used to calculate

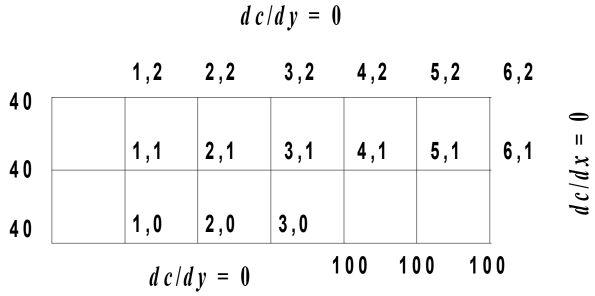
$$\frac{dc}{dx}(0) \cong \frac{52.47 - 76.44}{2.5} = -0.9588 \ \frac{\text{g/m}^3}{\text{m}}$$

Thus, the flux at the head of the tank is

$$J(x) = -2(-0.9588) + 1(76.44) = 19.176 + 76.44 = 95.616 \ \frac{\text{g/m}^3}{\text{m}}$$

The remasinder of the values can be calculated in a similar fashion using centered (middle nodes) and backward differences (the end node):

32.4 Segmentation scheme:




Nodes 1,1 through 5,1

$$0.4\frac{c_{i+1,j} - 2c_{i,j} + c_{i-1,j}}{5^2} + 0.4\frac{c_{i,j+1} - 2c_{i,j} + c_{i,j+1}}{5^2} - 0.2c_{i,j}$$

Collecting terms gives

$$0.264c_{i,j} - 0.016c_{i+1,j} - 0.016c_{i-1,j} - 0.016c_{i,j+1} - 0.016c_{i,j+1} = 0$$

Node 6,1 would be modified to reflect the no flow condition in *x* and the Dirichlet condition at 6,0:

$$0.264c_{6,1} - 0.032c_{5,1} - 0.016c_{6,2} - 0.016(100) = 0$$

The nodes along the upper edge (1,2 through 5,2) would be generally written to reflect the no-flow condition in *y* as

$$0.264c_{i,j} - 0.016c_{i+1,j} - 0.016c_{i-1,j} - 0.032c_{i,j+1} = 0$$

The node at the upper right edge (6,2) would be generally written to reflect the no-flow condition in *x* and *y* as

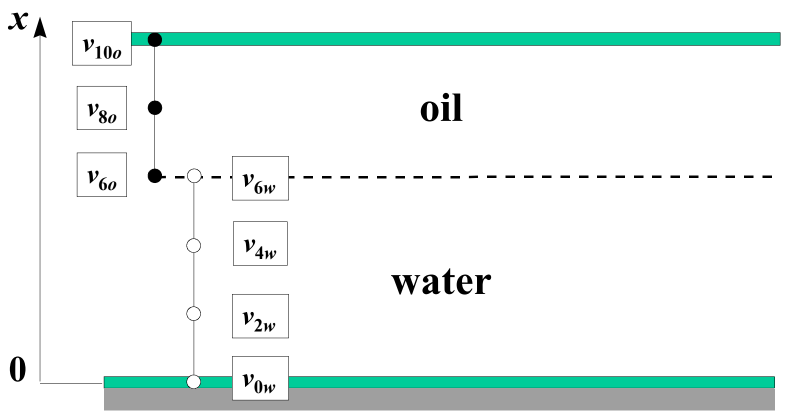
$$0.264c_{6,2} - 0.032c_{5,2} - 0.032c_{6,1} = 0$$

Finally, the nodes along the lower edge (1,0 through 3,0) would be generally written to reflect the no-flow condition in *y* as

$$0.264c_{i,j} - 0.016c_{i+1,j} - 0.016c_{i-1,j} - 0.032c_{i,j+1} = 0$$

These equations can be solved for

32.5 For simplicity, we will use a very coarse grid to solve this problem. Thus, we place nodes as in the following diagram.




A simple explicit solution can be developed by substituting finite-differences for the second derivative terms in the motion equations. This is done for the three non-boundary nodes,

$$\frac{dv_{2w}}{dt} = \mu_w \frac{v_{0w} - 2v_{2w} + v_{4w}}{\Delta x^2}$$

$$\frac{dv_{4w}}{dt} = \mu_w \frac{v_{2w} - 2v_{4w} + v_{6w}}{\Delta x^2}$$

$$\frac{dv_{8o}}{dt} = \mu_o \frac{v_{6o} - 2v_{8o} + v_{10o}}{\Delta x^2}$$

These three equations have 7 unknowns ($v_{0w}$, $v_{2w}$, $v_{4w}$, $v_{6w}$, $v_{6o}$, $v_{8o}$, $v_{10o}$). The boundary conditions at the plates effectively specify $v_{0w} = 0$ and $v_{10o} = 7$. The former is called a "no slip" condition because it specifies that the velocity at the lower plate is zero.

The relationships at the oil-water interface can be used to used to eliminate two of the remaining unknowns. The first condition states that

$$v_{6o} = v_{6w} \qquad (i)$$

The second can be rearrange to yield

$$v_{6w} = \frac{\mu_o v_{8o} + \mu_w v_{4w}}{\mu_o + \mu_w} \qquad (ii)$$

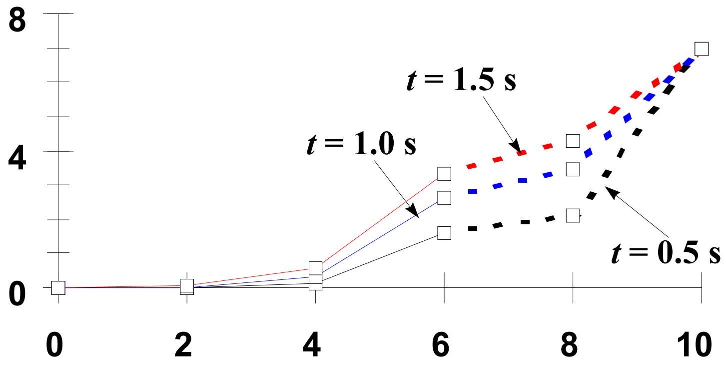
These, along with the wall boundary conditions can be substituted into the differential equations

$$\frac{dv_{2w}}{dt} = \mu_w \frac{-2v_{2w} + v_{4w}}{\Delta x^2}$$

$$\frac{dv_{4w}}{dt} = \mu_w \frac{v_{2w} - 2v_{4w} + \dfrac{\mu_o v_{8o} + \mu_w v_{4w}}{\mu_o + \mu_w}}{\Delta x^2}$$

$$\frac{dv_{8o}}{dt} = \mu_o \frac{\dfrac{\mu_o v_{8o} + \mu_w v_{4w}}{\mu_o + \mu_w} - 2v_{8o} + 7}{\Delta x^2}$$

These equations can now be integrated to determine the velocities as a function of time. Equations ($i$) and ($ii$) can be used to determine $v_{6o}$ and $v_{6w}$. The results are plotted below:

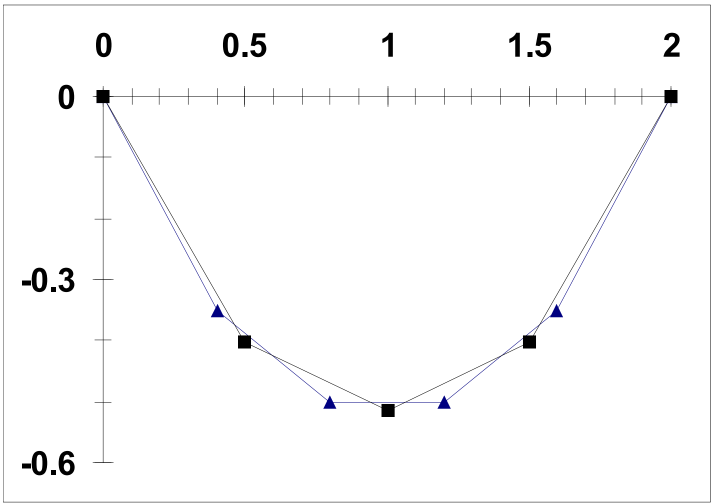



32.6 Using a similar approach to Sec. 32.2, the following nodal equation can be developed for node 11:

$$4u_{11} - 1.21954u_{12} - 1.21954u_{10} - 0.78049u_{12} - 0.78049u_{01} = 0.357866$$

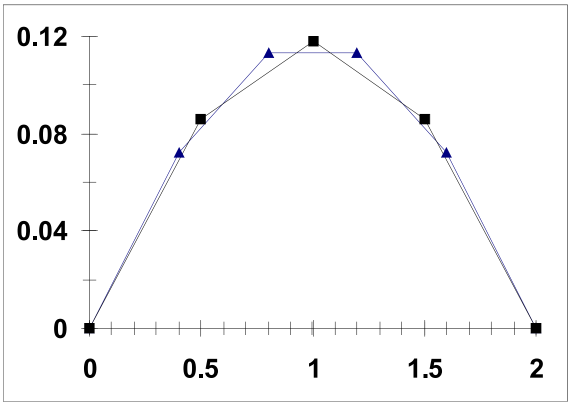
Similar equations can be written for the other nodes and the resulting equations solved for

A graphical comparison of the results from Sec. 32.2 can be made with these results by developing a plot along the $y$ dimension in the middle of the plate:

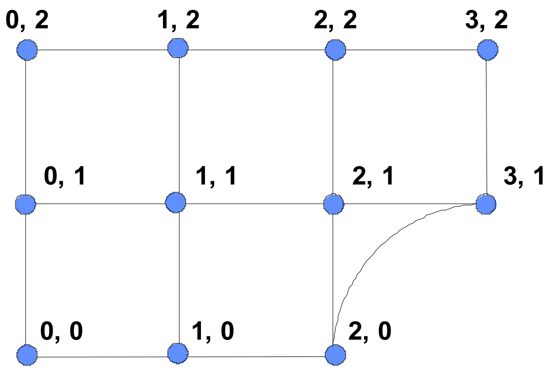



These results can then be used as input to the right-hand side of Eq. 32.14 and the resulting simultaneous equations solved for

Again the comparison is good




32.7 Grid scheme




All nodes in the above scheme can be modeled with the following general difference equation

$$\frac{h_{i+1,j} - 2h_{i,j} + h_{i-1,j}}{\Delta x^2} + \frac{h_{i,j+1} - 2h_{i,j} + h_{i,j-1}}{\Delta y^2} = 0$$

Node 0,0:

$$\frac{h_{1,0} - 2h_{0,0} + h_{-1,0}}{\Delta x^2} + \frac{h_{0,1} - 2h_{0,0} + h_{0,-1}}{\Delta y^2} = 0$$

The external nodes can be approximated with finite differences

$$\frac{dh}{dy} = \frac{h_{0,1} - h_{0,-1}}{2\Delta y}$$

$$h_{0,-1} = h_{0,1} - 2\Delta y \frac{dh}{dy} = h_{0,1}$$

$$\frac{dh}{dx} = \frac{h_{1,0} - h_{-1,0}}{2\Delta x}$$

$$h_{-1,0} = h_{1,0} - 2\Delta x \frac{dh}{dx} = h_{1,0} - 2(1)(1) = h_{1,0} - 2$$

which can be substituted into the difference equation to give

$$\frac{2h_{1,0} - 2h_{0,0} - 2}{\Delta x^2} + \frac{2h_{0,1} - 2h_{0,0}}{\Delta y^2} = 0$$

$$4h_{0,0} - 2h_{1,0} - 2h_{0,1} = -2$$

Node 1,0:

$$4h_{1,0} - 2h_{1,1} - h_{0,0} - h_{2,0} = 0$$

Node 2,0:

$$4h_{2,0} - 2h_{1,0} - 2h_{2,1} = 0$$

Node 0,1:

$$4h_{0,1} - 2h_{1,1} - h_{0,0} - h_{0,2} = -2$$

Node 1,1:

$$4h_{1,1} - h_{1,0} - h_{0,1} - h_{1,2} - h_{2,1} = 0$$

Node 2,1:

$$4h_{2,1} - h_{1,1} - h_{2,2} - h_{3,1} - h_{2,0} = 0$$

Node 0,2:

$$4h_{0,2} - 2h_{0,1} - 2h_{1,2} = -2$$

Node 1,2:

$$4h_{1,2} - h_{0,2} - h_{2,2} - 2h_{1,1} = 0$$

Node 2,2:

$$4h_{2,2} - h_{1,2} - 2h_{2,1} = 20$$

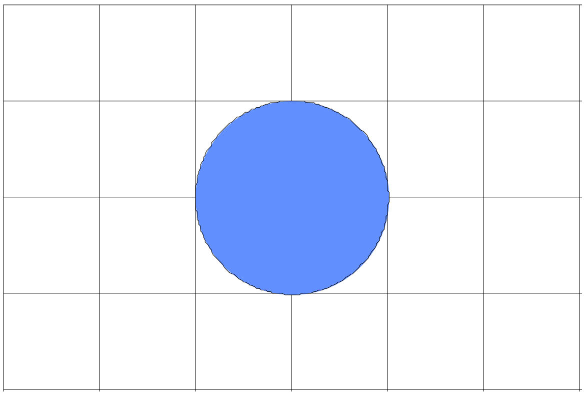
The equations can be solved simultaneously for

More refined results can be obtained by using a finer grid spacing.

32.8 The fluxes can be determined using finite divided differences as

32.9 Because of the equi-spaced grid, the domain can be modeled with simple Laplacians. The resulting solution is

32.10 A convenient segmentation scheme can be developed as



Simple Laplacians reflecting the boundary conditions can be developed and solved for

32.11 The system to be solved is

$$\begin{bmatrix} 2.7 & -2 & & \\ -2 & 2.75 & -0.75 & \\ & -0.75 & 2.25 & -1.5 \\ & & -1.5 & 1.5 \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \\ 0 \\ 2 \end{Bmatrix}$$

which can be solved for $x_1 = 2.857$, $x_2 = 3.857$, $x_3 = 6.5238$, and $x_4 = 7.857$.

32.12 The system to be solved is

$$\begin{bmatrix} 0.6 & -0.4 & & & \\ -0.4 & 1.8 & -1.4 & & \\ & -1.4 & 2.1 & -0.7 & \\ & & -0.7 & 1.6 & -0.9 \\ & & & -0.9 & 0.9 \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{Bmatrix}$$

which can be solved for $x_1 = 5$, $x_2 = 7.5$, $x_3 = 8.214286$, $x_4 = 9.64286$, and $x_5 = 10.75397$.

32.13 Substituting the Crank-Nicolson finite difference analogues to the derivatives

$$\frac{\partial^2 u}{\partial x^2} = \frac{1}{2}\left[\frac{u_{i+1,n+1} - u_{i,n+1} + u_{i-1,n+1}}{\Delta x^2} + \frac{u_{i+1,n} - u_{i,n} + u_{i-1,n}}{\Delta x^2}\right]$$

$$\frac{\partial u}{\partial t} = \frac{u_{i,n+1} - u_{i,n}}{\Delta t}$$

into the governing equations gives the following finite difference equations:

$$[1]u_{i-1,n+1} + \left[-2 - 2\frac{\Delta x^2}{\Delta t}\right]u_{i,n+1} + [1]u_{i+1,n+1} = -u_{i-1,n} + \left[2 - 2\frac{\Delta x^2}{\Delta t}\right]u_{i,n} - u_{i+1,n} \qquad 0 \le x \le \frac{1}{2}$$

$$[r]u_{i-1,n+1} + \left[-2r - 2\frac{\Delta x^2}{\Delta t}\right]u_{i,n+1} + [r]u_{i+1,n+1} = -ru_{i-1,n} + \left[2r - 2\frac{\Delta x^2}{\Delta t}\right]u_{i,n} - ru_{i+1,n} \qquad \frac{1}{2} \le x \le 1$$

Substitute for the end point boundary conditions to get the end point finite difference equations.
Substitute the first order Crank Nicolson analogues to the derivatives

$$\frac{\partial u}{\partial r} = \frac{1}{2}\left[\frac{u_{i+1,n+1} - u_{i-1,n+1}}{2\Delta r} + \frac{u_{i+1,n} - u_{i-1,n}}{2\Delta r}\right]$$ into the midpoint boundary condition and get

$$u^a{}_{L+1,n+1} + u^a{}_{L+1,n} + r(u^b{}_{L+1,n+1} + u^b{}_{L+1,n}) = u_{L-1,n+1} + u_{L=1,n} + r(u_{L+1,n+1} + u_{L+1,n})$$

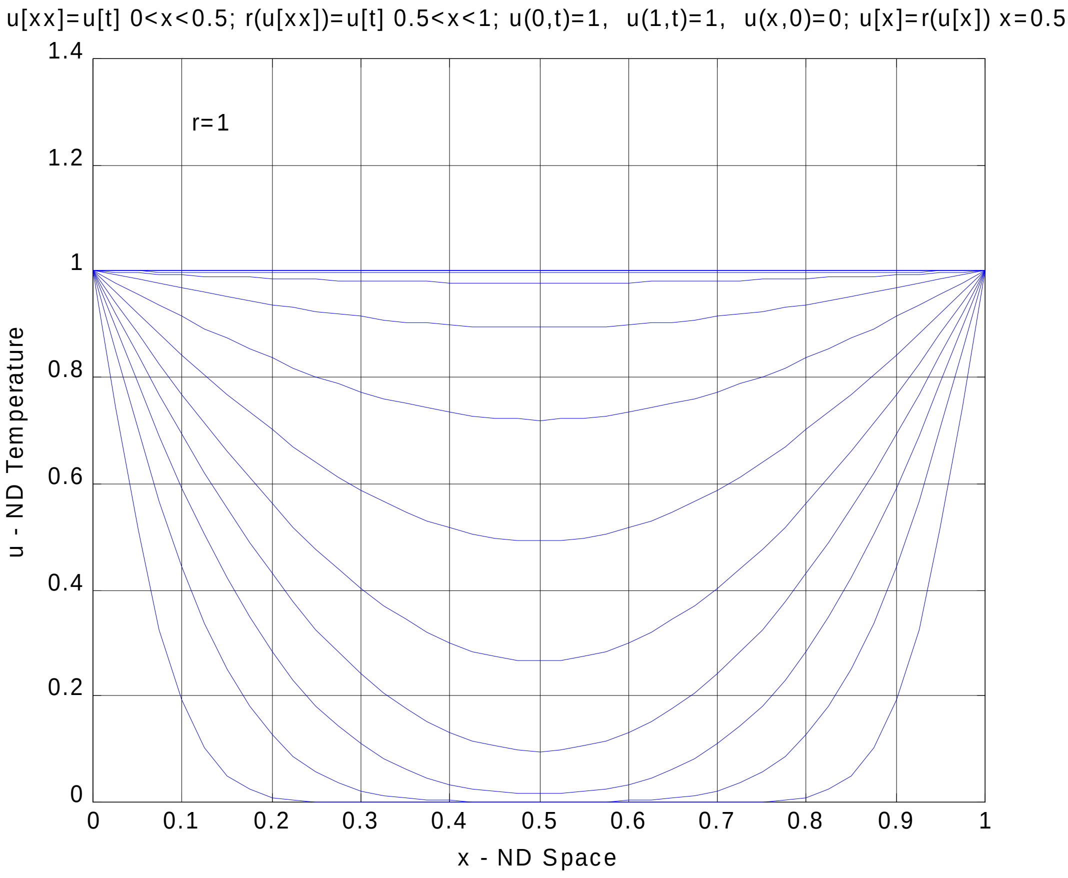
where $u^a$ and $u^b$ are fictitious points located in the opposite side of the midpoint from their half. Write out the two finite difference equations from above for the point i = L (the midpoint) then combine these two equations with the midpoint boundary condition to obtain the midpoint finite difference equation:

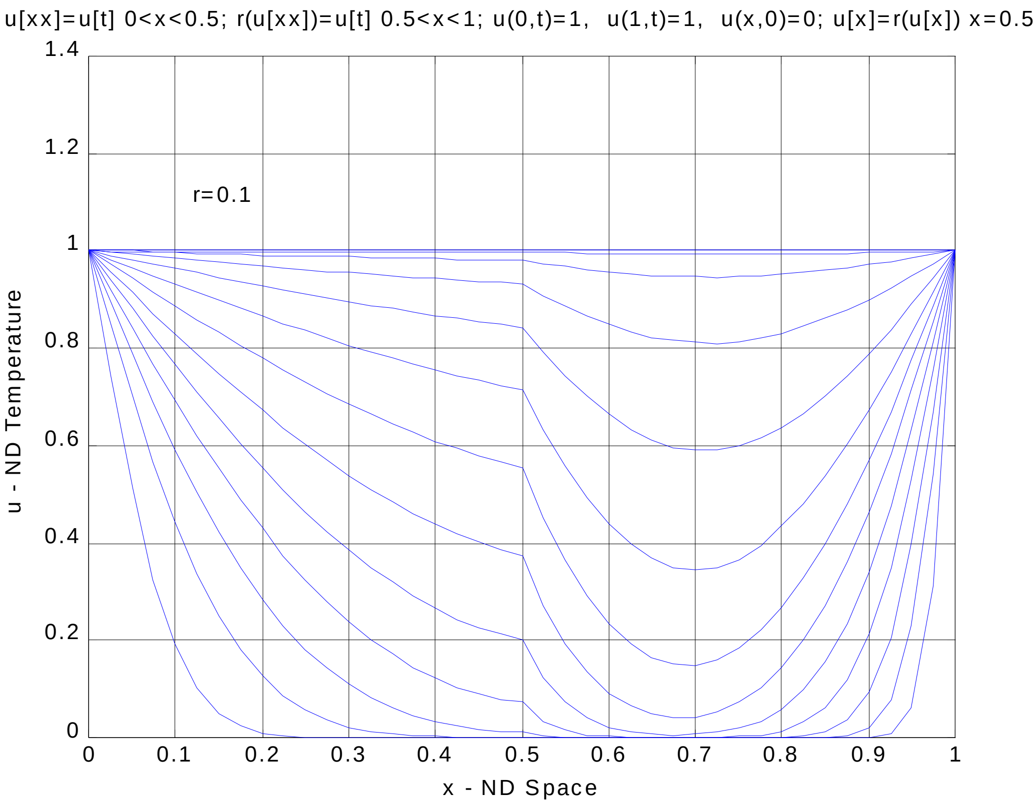
$$[2]u_{L-1,n+1} + \left[-2(1+r) - 4\frac{\Delta x^2}{\Delta t}\right]u_{L,n+1} + [2]u_{L+1,n+1} = -2u_{L-1,n} + \left[2(1+r) - 4\frac{\Delta x^2}{\Delta t}\right]u_{L,n} - (1+r)u_{L+1,n}$$

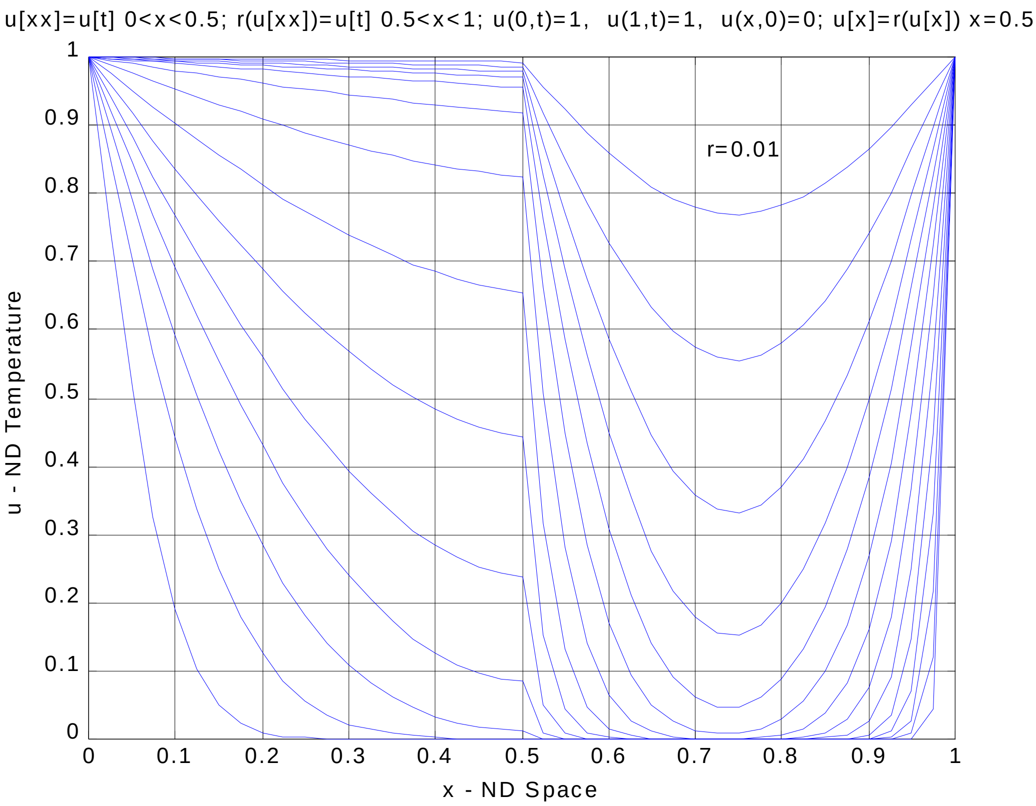
```
%PDE Parabolic Problem - Transient Heat conduction in a composite rod
%   u[xx]=u[t]         0<x<0.5
%   r(u[xx])=u[t]        0.5<x<1
%   BC u(0,t)=1  u(1,t)=1
%   u[x]=r(u[x])        x=0.5
%  IC   u(x,0)=0      0<x<1
%  i=spatial index, from 1 to imax
%  R = no. of x points (R=21 for 20 dx spaces)
%  n=time index  from 1 to N
%  N = no. of time steps,
%   Crank-Nicolson Formulation

R=41;       %(imax must be odd for point L to be correct)
N=69;       % last time step = nmax+1
L=(R-1)/2+1;  %  L = midpoint of point no. (for R=41, L=21)

%  Constants
     r=0.01;
     dx=1/(R-1);
     dx2=dx*dx;
     dt=dx2;     % Setting dt to dx2 for good stabilility and results

%  Independent space variable
     x=0:dx:1;

%  Sizing matrices
     u=zeros(R,N+1); t=zeros(1,N+1);
     a=zeros(1,R);  b=zeros(1,R);
       c=zeros(1,R);  d=zeros(1,R);
     ba=zeros(1,R); ga=zeros(1,R);
     up=zeros(1,R);


%  Boundary Conditions at t=0
         u(1,1)=1;
         u(R,1)=1;

%  Time step loop
%  n=1 represents 0 time, next time = n+1
     t(1)=0;
     for n=1:N
          t(n+1)=t(n)+dt;

% Boundary conditions & Constants
     u(1,n+1)=1;
     u(R,n+1)=1;
     dx2dt=dx2/dt;
```

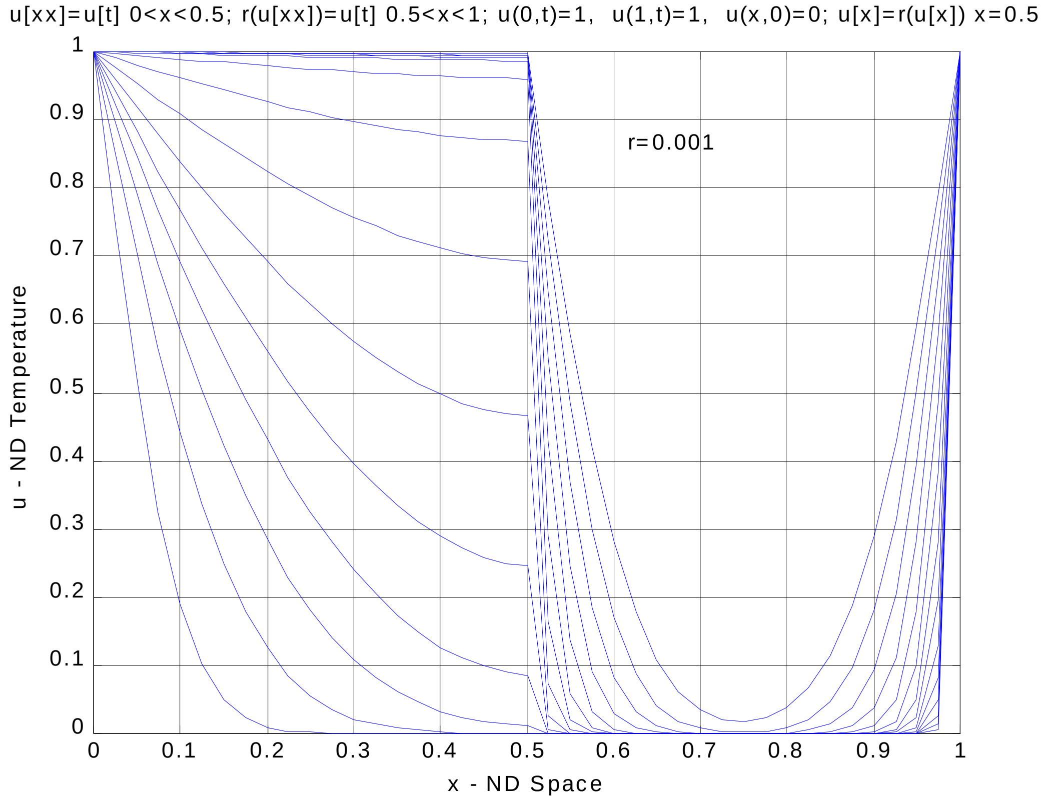
```
% coefficients
        b(2)=-2-2*dx2dt;
        c(2)=1;
        d(2)=(2-2*dx2dt)*u(2,n)-u(3,n)-2;
        for i=3:L-1
         a(i)=1;
         b(i)=-2-2*dx2dt;
         c(i)=1;
         d(i)=-u(i-1,n)+(2-2*dx2dt)*u(i,n)-u(i+1,n);
        end
        a(L)=2;
        b(L)=-2*(1+r)-4*dx2dt;
        c(L)=2*r;
        d(L)=-2*u(L-1,n)+(2*(1+r)-4*dx2dt)*u(L,n)-2*r*u(L+1,n);

      for i=L+1:R-2
         a(i)=r;
         b(i)=-2*r-2*dx2dt;
         c(i)=r;
         d(i)=-r*u(i-1,n)+(2*r-2*dx2dt)*u(i,n)-r*u(i+1,n);
        end
            a(R-1)=r;
            b(R-1)=-2*r-2*dx2dt;
            d(R-1)=-r*u(R-2,n)+(2*r-2*dx2dt)*u(R-1,n)-2*r;

% Solution by Thomas Algorithm
        ba(2)=b(2);
        ga(2)=d(2)/b(2);
        for i=3:R-1
         ba(i)=b(i)-a(i)*c(i-1)/ba(i-1);
         ga(i)=(d(i)-a(i)*ga(i-1))/ba(i);
        end

% Back substitution step
        u(R-1,n+1)=ga(R-1);
        for i=R-2:-1:2
         u(i,n+1)=ga(i)-c(i)*u(i+1,n+1)/ba(i);
        end
        dt=1.1*dt;

end
% end of time step loop


% Plot
% Storing plot value of u as up, at every 5 time steps
% j=time index
% i=space index

for j=5:5:N+1
  for i=1:R
      up(i)=u(i,j);
  end
  plot(x,up)
  hold on
end

grid
title('u[xx]=u[t] 0<x<0.5; r(u[xx])=u[t] 0.5<x<1; u(0,t)=1,  u(1,t)=1,
u(x,0)=0; u[x]=r(u[x]) x=0.5')
xlabel('x - ND Space')
ylabel('u - ND Temperature')
hold off
gtext('r=0.01')
```

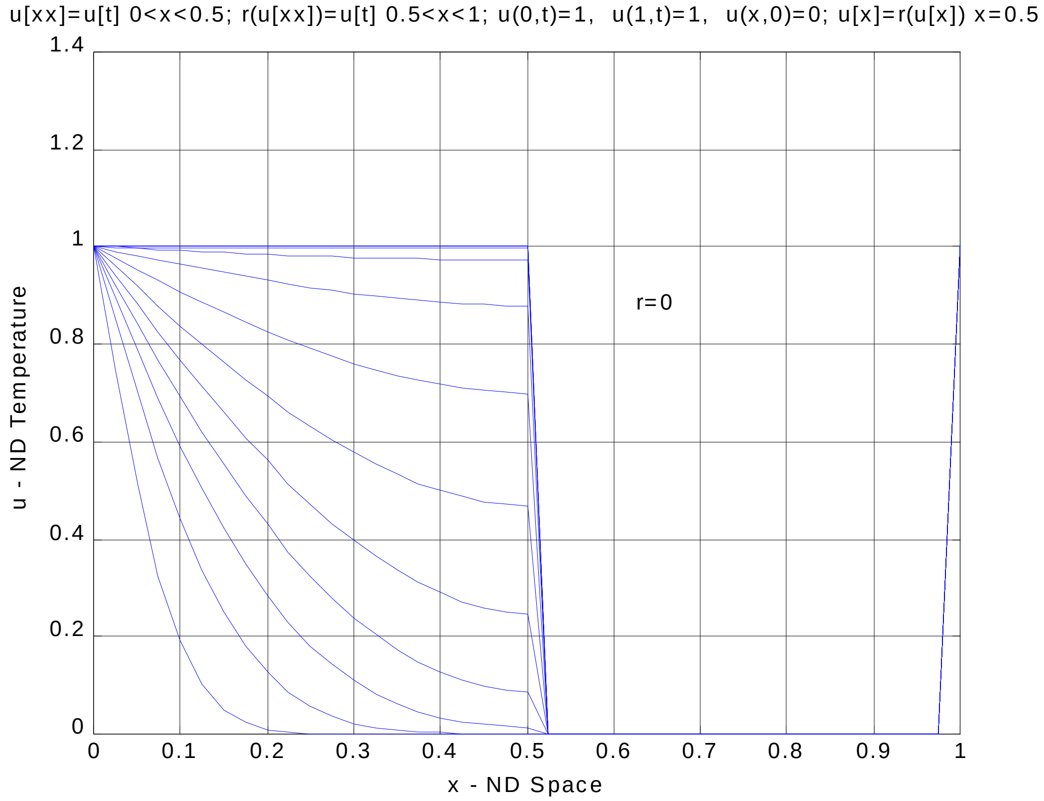
```
% Storing times for temp. profiles
%     These can be saved in a data file or examined in the command file
                    tp=zeros(1,(N-1)/5);
                    i=1;
                    tp(1)=0;
                    for k=5:5:N+1
                             i=i+1;
                             tp(i)=t(k);
                    end
                             tp

tp =

  Columns 1 through 7

         0    0.0029    0.0085    0.0175    0.0320    0.0553    0.0929

  Columns 8 through 14

    0.1534    0.2509    0.4079    0.6607    1.0679    1.7238    2.7799

  Column 15

    4.4809
```




u[xx]=u[t] 0<x<0.5; r(u[xx])=u[t] 0.5<x<1; u(0,t)=1, u(1,t)=1, u(x,0)=0; u[x]=r(u[x]) x=0.5
r=0.1
u - ND Temperature
x - ND Space


u[xx]=u[t] 0<x<0.5; r(u[xx])=u[t] 0.5<x<1; u(0,t)=1, u(1,t)=1, u(x,0)=0; u[x]=r(u[x]) x=0.5
r=0.01
u - ND Temperature
x - ND Space


u[xx]=u[t] 0<x<0.5; r(u[xx])=u[t] 0.5<x<1; u(0,t)=1, u(1,t)=1, u(x,0)=0; u[x]=r(u[x]) x=0.5
r=0.001
u - ND Temperature
x - ND Space
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

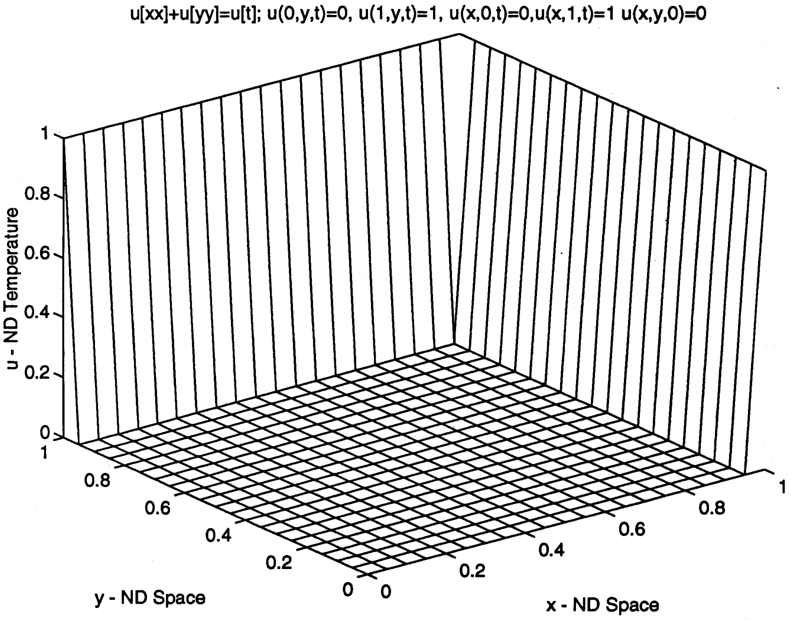

u[xx]=u[t] 0<x<0.5; r(u[xx])=u[t] 0.5<x<1; u(0,t)=1, u(1,t)=1, u(x,0)=0; u[x]=r(u[x]) x=0.5
r=0
u - ND Temperature
x - ND Space
0
0.2
0.4
0.6
0.8
1
1.2
1.4
0.1
0.3
0.5
0.7
0.9

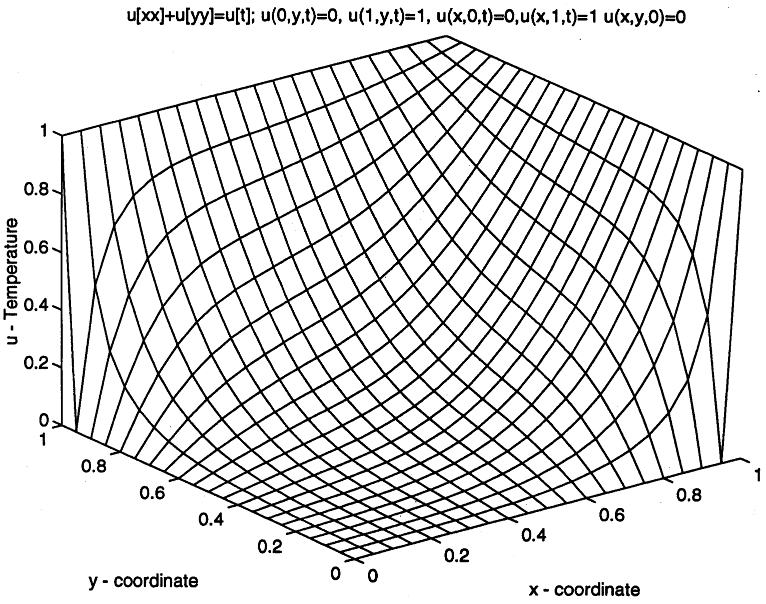
32.14

```
%PDE Parabolic Problem - Heat conduction in a rod
      %     u[xx]+u[yy]=u[t]
      %  BC     u(0,y,t)=0  u(1,y,t)=1
      %         u(x,0,t)=0  u(x,1,t)=1
      %  IC u(x,y,0)=0    0<=x<1  0<=y<1
      %   Crank-Nicolson Formulation
      %   Alternating-Direction-Implicit Solution Method
      %   Intermediate values of u stored as u1
      %   MATLAB 5.x Verson with multidimensional arays
      %      i=spatial index in x-direction, from 1 to R
      %      j=spatial index in y-direction, from 1 to S
      %      n=time index  from 1 to N
R=21;    % last x-point
S=21;    % last y-point
N=20;    % last time step = N+1
      %  Constants
dx=1/(R-1);
dx2=dx*dx;
dy=1/(S-1);
dy2=dy*dy;
dxdy=dx2/dy2;
dydx=dy2/dx2;
dt=dx2;        % Settine dt to dx2 for good stabilility and results
       %  Independent space variables
x=0:dx:1;
y=0:dy:1;
       %  Sizing matrices
u=zeros(R,S,N); u1=zeros(R,S);
       % u(i,j,n) = present time, u1=first pass intermediate results, u(i,j,n+1) = next time step
t=zeros(1,N+1);
a=zeros(1,R);  b=zeros(1,R);  c=zeros(1,R);  d=zeros(1,R);
ba=zeros(1,R); ga=zeros(1,R);
      %  Boundary Conditions
for n=1:N
  for i=1:R
     u(i,S,n)=1
  end
  for j=1:S
     u(R,j,n)=1;
  end
end
      % Intermediate Values
for i=1:R
   u1(i,S)=1;
end
for j=1:S
   u1(R,j)=1;
end
      %Plot Initial Conditions
mesh(x,y,u(:,:,1))
   title('u[xx]+u[yy]=u[t]; u(0,y,t)=0, u(1,y,t)=1, u(x,0,t)=0,u(x,1,t)=1 u(x,y,0)=0')
   xlabel('x-coordinate'); ylabel('y-coordinate'); zlabel('u - Temperature')
pause
% ******************************************************************
       %  Time step loop
       %  n=1 represents 0 time, n+1 = next time step
t(1)=0;
for n=1:N
   t(n+1)=t(n)+2*dt;
```

```
% First pass in x-direction **********************************
% first time step - intermediate valus at u1(i,j) are calculated
      %  Constants
dx2dt=dx2/dt;
      % Coefficients
for j=2:S-1
    b(2)=-2-dx2dt;
    c(2)=1;
    d(2)=-dxdy*u(2,j-1,n)+(2*dxdy-dx2dt)*u(2,j,n)-dxdy*u(2,j+1,n);
    for i=3:R-2
      a(i)=1;
      b(i)=-2-dx2dt;
      c(i)=1;
      d(i)=-dxdy*u(i,j-1,n)+(2*dxdy-dx2dt)*u(i,j,n)-dxdy*u(i,j+1,n);
    end
    a(R-1)=1;
    b(R-1)=-2-dx2dt;
    d(R-1)=-1-dxdy*u(i,j-1,n)+(2*dxdy-dx2dt)*u(i,j,n)-dxdy*u(i,j+1,n);
          % Solution by Thomas Algorithm
    ba(2)=b(2);
    ga(2)=d(2)/b(2);
    for i=3:R-1
       ba(i)=b(i)-a(i)*c(i-1)/ba(i-1);
       ga(i)=(d(i)-a(i)*ga(i-1))/ba(i);
    end
        % Back substitution step
    u1(R-1,j)=ga(R-1);
    for i=R-2:-1:2
       u1(i,j)=ga(i)-c(i)*u1(i+1,j,n)/ba(i);
    end
end
% Second pass in y-direction  *************************************
% Second time step - final valus at u(i,j,n+1) are calculated
dy2dt=dy2/dt;
      % Coefficients
for i=2:R-1
    b(2)=-2-dy2dt;
    c(2)=1;
    d(2)=-dydx*u1(i-1,2)+(2*dydx-dy2dt)*u1(i,2)-dydx*u1(i+1,2);
    for j=3:S-2
      a(j)=1;
      b(j)=-2-dy2dt;
      c(j)=1;
      d(j)=-dydx*u1(i-1,j)+(2*dydx-dy2dt)*u1(i,j)-dydx*u1(i+1,j);
    end
    a(S-1)=1;
    b(S-1)=-2-dy2dt;
    d(S-1)=-1-dydx*u1(i-1,S-1)+(2*dydx-dy2dt)*u1(i,S-1)-dydx*u1(i+1,S-1);
       % Solution by Thomas Algorithm
    ba(2)=b(2);
    ga(2)=d(2)/b(2);
    for j=3:S-1
       ba(j)=b(j)-a(j)*c(j-1)/ba(j-1);
       ga(j)=(d(j)-a(j)*ga(j-1))/ba(j);
    end
       % Back substitution step
    u(i,S-1,n+1)=ga(S-1);
    for j=S-2:-1:2
      u(i,j,n+1)=ga(j)-c(j)*u(i,j+1,n+1)/ba(j);
```

```
        end
    end
        % dt can be incremented at this point if desired as     dt=1.1*dt
end             % End time step loop
%****************************************************************************
        % Plot Results
    mesh(x,y,u(:,:,10)
      title('u[xx]+u[yy]=u[t]; u(0,y,t)=0, u(1,y,t)=1, u(x,0,t)=0,u(x,1,t)=1 u(x,y,0)=0')
      xlabel('x-coordinate'); ylabel('y-coordinate'); zlabel('u - Temperature')
      pause
      t(10)

    mesh(x,y,u(:,:,20))
      title('u[xx]+u[yy]=u[t]; u(0,y,t)=0, u(1,y,t)=1, u(x,0,t)=0,u(x,1,t)=1 u(x,y,0)=0')
      xlabel('x-coordinate'); ylabel('y-coordinate'); zlabel('u - Temperature')
      t(20)
```



| | | | | | | |
|---|---|---|---|---|---|---|
| 99.27296 | 99.15714 | 98.55306 | 96.07795 | 85.75874 | 69.00866 | 50 |
| 99.38879 | 99.40126 | 99.48858 | 100 | 88.97417 | 70.13795 | 50 |
| 99.47967 | 99.57055 | 100 | 100 | 100 | 72.56898 | 50 |
| 99.38879 | 99.40126 | 99.48858 | 100 | 88.97417 | 70.13795 | 50 |
| 99.27296 | 99.15714 | 98.55306 | 96.07795 | 85.75874 | 69.00866 | 50 |

| | | | | | |
|---|---|---|---|---|---|
| 25 | 40 | 40 | 30 | | |
| 10 | 21.87149 | 24.04033 | 20 | 15 | |
| 10 | 13.44564 | 14.28983 | 12.63401 | 10 | 7.5 |
| 10 | 7.62124 | 7.039322 | 6.246222 | 5.311556 | 5 |
| 5 | 0 | 0 | 0 | 0 | 2.5 |

***dh/dx***

| | | | |
|---|---|---|---|
| 1.040287 | 1.106512 | 1.311258 | 1.449779 |
| 1.014349 | 1.057395 | 1.344371 | 1.5883 |
| 0.931015 | 0.778698 | 0.62638 | |

***dh/dy***

| | | | |
|---|---|---|---|
| 0.040287 | 0.066225 | 0.138521 | 0 |
| 0.054636 | 0.109272 | 0.38245 | 0 |
| 0.068985 | 0.152318 | 0.62638 | |

***dh/dn***

| | | | |
|---|---|---|---|
| 1.041067 | 1.108492 | 1.318555 | 1.449779 |
| 1.015819 | 1.063026 | 1.397713 | 1.5883 |
| 0.933568 | 0.793455 | 0.885835 | |

**$\theta$ (radians)**

| | | | |
|---|---|---|---|
| 0.038707 | 0.059779 | 0.105249 | 0 |
| 0.053811 | 0.102975 | 0.277161 | 0 |
| 0.073961 | 0.193167 | 0.785398 | |

**$\theta$ (degrees)**

| | | | |
|---|---|---|---|
| 2.217773 | 3.425088 | 6.030345 | 0 |
| 3.083137 | 5.90002 | 15.88014 | 0 |
| 4.237646 | 11.06765 | 45 | |

| | | | |
|---|---|---|---|
| 16.3372 | 17.37748 | 18.55022 | 20 |
| 16.29691 | 17.31126 | 18.4117 | 20 |
| 16.22792 | 17.15894 | 17.78532 | |

| | | | | |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0.052697 | 0.072156 | 0.052697 | 0 |
| 0 | 0.082316 | 0.113101 | 0.082316 | 0 |
| 0 | 0.082316 | 0.113101 | 0.082316 | 0 |
| 0 | 0.052697 | 0.072156 | 0.052697 | 0 |
| 0 | 0 | 0 | 0 | 0 |

| | | | | |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 0 | -0.27719 | -0.35074 | -0.27719 | 0 |
| 0 | -0.39124 | -0.50218 | -0.39124 | 0 |
| 0 | -0.39124 | -0.50218 | -0.39124 | 0 |
| 0 | -0.27719 | -0.35074 | -0.27719 | 0 |
| 0 | 0 | 0 | 0 | 0 |

| | | | | | | |
|---|---|---|---|---|---|---|
| 20 | 1.387741 | 0.113952 | 0.155496 | 0.864874 | 0.951623 | 0.958962 |
| 20 | 1.391891 | 0.168488 | 0.793428 | 6.581653 | 6.938975 | 6.959813 |
| 20 | 1.409973 | 0.48078 | 6.185917 | 100 | 100 | 100 |

| c | dcdx | J-diff | J-adv | J |
|---|---|---|---|---|
| 76.44 | -9.588 | 19.176 | 76.44 | 95.616 |
| 52.47 | -8.076 | 16.152 | 52.47 | 68.622 |
| 36.06 | -5.484 | 10.968 | 36.06 | 47.028 |
| 25.05 | -3.394 | 6.788 | 25.05 | 31.838 |
| 19.09 | -2.384 | 4.768 | 19.09 | 23.858 |

| | | | |
|---|---|---|---|
| $c_0$ | 76.53 | $c_5$ | 29.61 |
| $c_1$ | 63.25 | $c_6$ | 24.62 |
| $c_2$ | 52.28 | $c_7$ | 20.69 |
| $c_3$ | 43.22 | $c_8$ | 17.88 |
| $c_4$ | 35.75 | $c_9$ | 16.58 |

## CHAPTER 25

25.1 The analytical solution can be derived by separation of variables

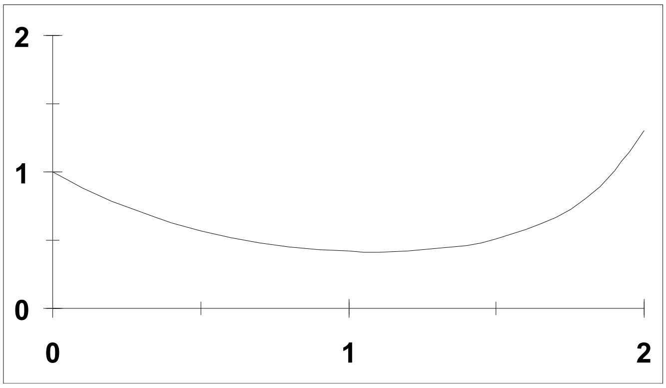
$$\int \frac{dy}{y} = \int x^2 - 1.2\, dx$$

$$\ln y = \frac{x^3}{3} - 1.2x + C$$

Substituting the initial conditions yields $C = 0$. Taking the exponential give the final result

$$y = e^{\frac{x^3}{3} - 1.2x}$$

The result can be plotted as



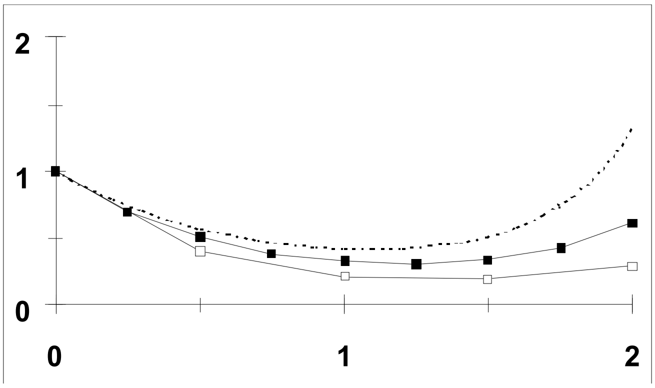

25.2 Euler's method with $h = 0.5$

| ***x*** | ***y*** | ***dy/dx*** |
|---|---|---|
| 0 | 1 | -1.2 |
| 0.5 | 0.4 | -0.38 |
| 1 | 0.21 | -0.042 |
| 1.5 | 0.189 | 0.19845 |
| 2 | 0.288225 | 0.80703 |

Euler's method with $h = 0.25$ gives

| ***x*** | ***y*** | ***dy/dx*** |
|---|---|---|
| 0 | 1 | -1.2 |
| 0.25 | 0.7 | -0.79625 |
| 0.5 | 0.500938 | -0.47589 |
| 0.75 | 0.381965 | -0.2435 |
| 1 | 0.321089 | -0.06422 |
| 1.25 | 0.305035 | 0.110575 |
| 1.5 | 0.332679 | 0.349312 |
| 1.75 | 0.420007 | 0.782262 |
| 2 | 0.615572 | 1.723602 |

The results can be plotted along with the analytical solution as




25.3 For Heun’s method, the value of the slope at $x = 0$ can be computed as $-0.6$ which can be used to compute the value of $y$ at the end of the interval as

$$y(0.5) = 1 + (0 - 1.2(1))0.5 = 0.4$$

The slope at the end of the interval can be computed as

$$y'(0.5) = 0.4(0.5)^2 - 1.2(0.4) = -0.38$$

which can be averaged with the initial slope to predict

$$y(0.5) = 1 + \frac{-0.6 - 0.38}{2} 0.5 = 0.605$$

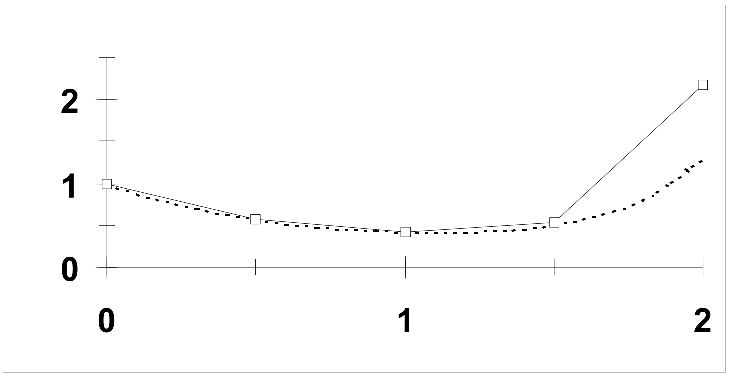
This formula can then be iterated to yield

| $j$ | $y_i^j$ | $\lvert\varepsilon_a\rvert$ |
|---|---|---|
| 0 | 0.4 | |
| 1 | 0.605 | 33.9 |
| 2 | 0.5563124 | 8.75 |
| 3 | 0.5678757 | 2.036 |
| 4 | 0.5651295 | 0.4859 |

The remaining steps can be implemented with the result

| $x_i$ | $y_i$ |
|---|---|
| 0.5 | 0.5651295 |
| 1.0 | 0.4104059 |
| 1.5 | 0.5279021 |
| 2.0 | 2.181574 |

The results along with the analytical solution are displayed below:



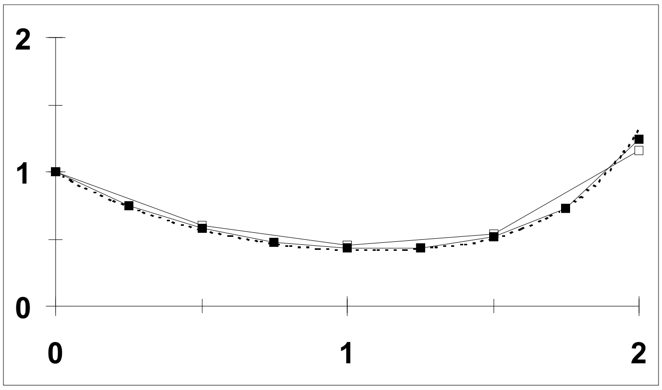

25.4 The midpoint method with $h = 0.5$

| $x$ | $y$ | $dy/dx$ | $ym$ | $dy/dx$-$mid$ |
|---:|---:|---:|---:|---:|
| 0 | 1 | -1.2 | 0.7 | -0.79625 |
| 0.5 | 0.601875 | -0.57178 | 0.45893 | -0.29257 |
| 1 | 0.455591 | -0.09112 | 0.432812 | 0.156894 |
| 1.5 | 0.534038 | 0.56074 | 0.674223 | 1.255741 |
| 2 | 1.161909 | 3.253344 | 1.975245 | 7.629383 |

with $h = 0.25$ gives

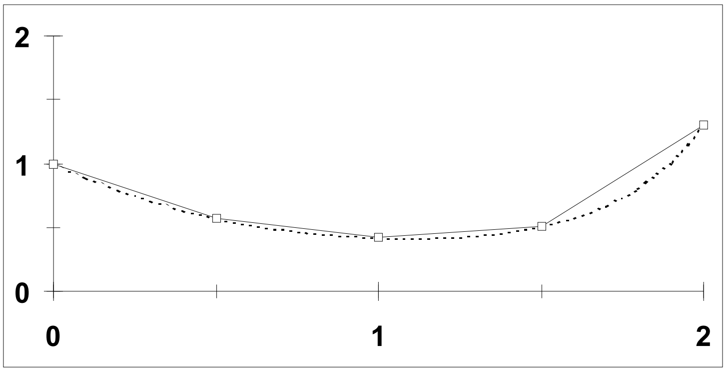
| $x$ | $y$ | $dy/dx$ | $ym$ | $dy/dx$-$mid$ |
|---:|---:|---:|---:|---:|
| 0 | 1 | -1.2 | 0.85 | -1.00672 |
| 0.25 | 0.74832 | -0.85121 | 0.641919 | -0.68003 |
| 0.5 | 0.578312 | -0.5494 | 0.509638 | -0.41249 |
| 0.75 | 0.47519 | -0.30293 | 0.437323 | -0.18996 |
| 1 | 0.4277 | -0.08554 | 0.417007 | 0.027366 |
| 1.25 | 0.434541 | 0.157521 | 0.454231 | 0.313703 |
| 1.5 | 0.512967 | 0.538615 | 0.580294 | 0.835986 |
| 1.75 | 0.721963 | 1.344657 | 0.890046 | 2.061012 |
| 2 | 1.237216 | 3.464206 | 1.670242 | 5.537897 |

The results can be plotted along with the analytical solution as



25.5 The 4th-order RK method with $h = 0.5$ gives

| *x* | *y* | *k1* | *ym* | *k2* | *ym* | *k3* | *ye* | *k4* | *phi* |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | -1.2 | 0.7 | -0.79625 | 0.800938 | -0.91107 | 0.544467 | -0.51724 | -0.85531 |
| 0.5 | 0.572344 | -0.54373 | 0.436412 | -0.27821 | 0.50279 | -0.32053 | 0.412079 | -0.08242 | -0.30394 |
| 1 | 0.420375 | -0.08407 | 0.399356 | 0.144767 | 0.456567 | 0.165505 | 0.503128 | 0.528284 | 0.177459 |
| 1.5 | 0.509104 | 0.534559 | 0.642744 | 1.197111 | 0.808382 | 1.505611 | 1.26191 | 3.533348 | 1.578892 |
| 2 | 1.29855 | 3.635941 | 2.207535 | 8.526606 | 3.430202 | 13.24915 | 7.923127 | 40.01179 | 14.53321 |




25.6 (*a*) The analytical solution can be derived by separation of variables

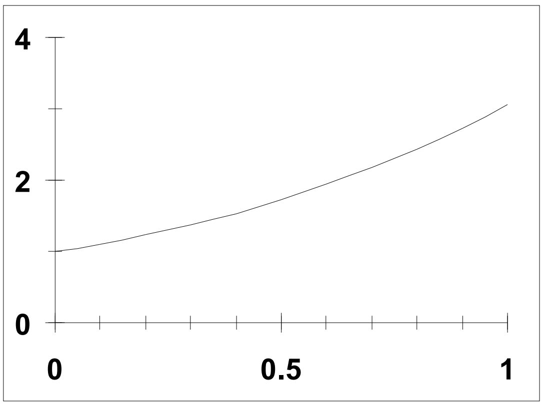
$$\int \frac{dy}{\sqrt{y}} = \int 1 + x \, dx$$

$$2\sqrt{y} = x + \frac{x^2}{2} + C$$

Substituting the initial conditions yields $C = 2$. Substituting this value and solving for $y$ gives the final result

$$y = \frac{(x^2 + 2x + 4)^2}{16}$$

The result can be plotted as




(*b*) Euler's method with $h = 0.5$

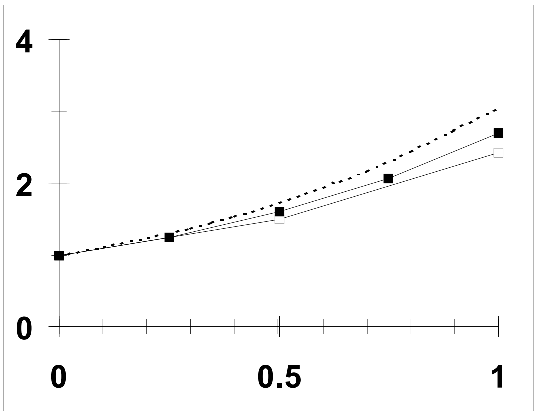
| *x* | *y* | *dy/dx* |
|---|---|---|

| | | |
|---|---|---|
| 0 | 1 | 1 |
| 0.5 | 1.5 | 1.837117 |
| 1 | 2.418559 | 3.110343 |

Euler's method with $h$ = 0.25 gives

| ***x*** | ***y*** | ***dy/dx*** |
|---|---|---|
| 0 | 1 | 1 |
| 0.25 | 1.25 | 1.397542 |
| 0.5 | 1.599386 | 1.897002 |
| 0.75 | 2.073636 | 2.520022 |
| 1 | 2.703642 | 3.288551 |

The results can be plotted along with the analytical solution as



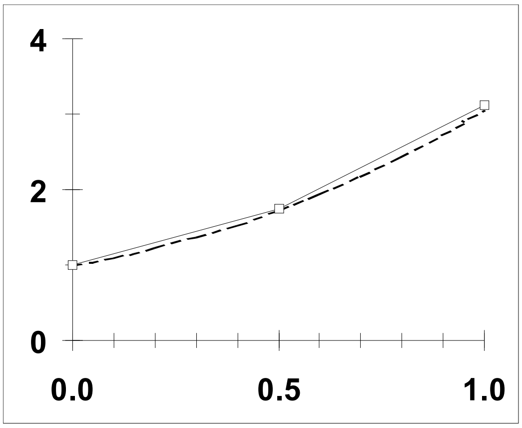

($c$) For Heun's method, the first step along with the associated iterations is

| $j$ | $y_i^j$ | $\|\varepsilon_a\|$ |
|---|---|---|
| 0 | 1.500000 | |
| 1 | 1.709279 | 12.243720 |
| 2 | 1.740273 | 1.780954 |
| 3 | 1.744698 | 2.536284E-01 |

The remaining steps can be implemented with the result

| $x_i$ | $y_i$ |
|---|---|
| 0.00E+00 | 1 |
| 5.00E-01 | 1.744698 |
| 1 | 3.122586 |

The results along with the analytical solution are displayed below:



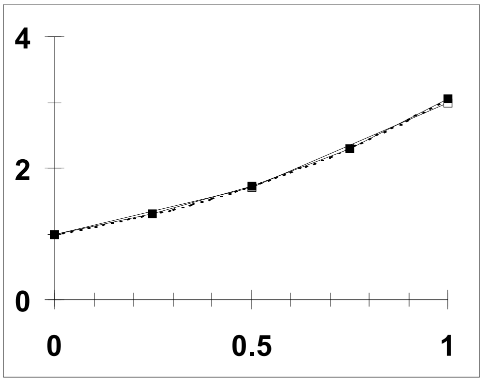

(*d*) The midpoint method with $h = 0.5$

| *x* | *y* | *dy/dx* | *ym* | *dy/dx-mid* |
|---:|---:|---:|---:|---:|
| 0 | 1 | 1 | 1.25 | 1.397542 |
| 0.5 | 1.698771 | 1.955054 | 2.187535 | 2.588305 |
| 1 | 2.992924 | 3.460014 | 3.857927 | 4.419362 |

with $h = 0.25$ gives

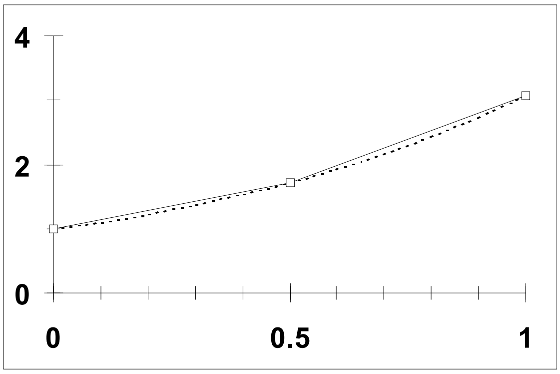
| *x* | *y* | *dy/dx* | *ym* | *dy/dx-mid* |
|---:|---:|---:|---:|---:|
| 0 | 1 | 1 | 1.125 | 1.193243 |
| 0.25 | 1.298311 | 1.424293 | 1.476347 | 1.670694 |
| 0.5 | 1.715984 | 1.964934 | 1.961601 | 2.275929 |
| 0.75 | 2.284966 | 2.645318 | 2.615631 | 3.032421 |
| 1 | 3.043072 | 3.48888 | 3.479182 | 3.96367 |

The results can be plotted along with the analytical solution as




(*e*) The 4th-order RK method with $h = 0.5$ gives

| *x* | *y* | *k1* | *ym* | *k2* | *ym* | *k3* | *ye* | *k4* | *phi* |
|---:|---:|---:|---:|---:|---:|---:|---:|---:|---:|
| 0 | 1 | 1 | 1.25 | 1.397542 | 1.349386 | 1.452038 | 1.726019 | 1.970671 | 1.444972 |
| 0.5 | 1.722486 | 1.968653 | 2.214649 | 2.604297 | 2.37356 | 2.696114 | 3.070543 | 3.504593 | 2.679011 |
| 1 | 3.061992 | 3.499709 | 3.936919 | 4.464376 | 4.178086 | 4.599082 | 5.361533 | 5.788746 | 4.569229 |




25.7 The second-order ODE is transformed into a pair of first-order ODEs as in

$$\frac{dy}{dx} = z \qquad y(0)=2$$

$$\frac{dz}{dx} = x - y \qquad z(0)=0$$

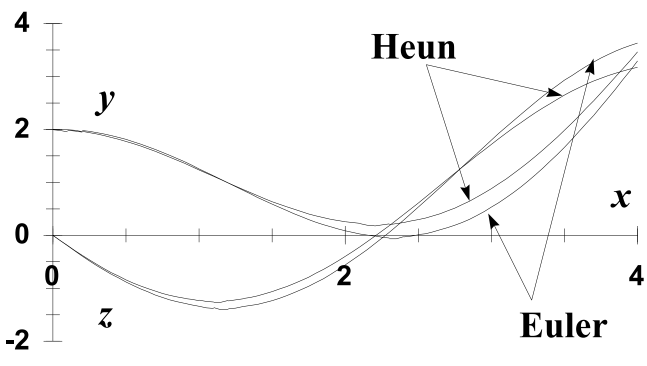
(*a*) The first few steps of Euler's method are

| *x* | *y* | *z* | *dy/dx* | *dz/dx* |
|---|---|---|---|---|
| 0 | 2 | 0 | 0 | -2 |
| 0.1 | 2 | -0.2 | -0.2 | -1.9 |
| 0.2 | 1.98 | -0.39 | -0.39 | -1.78 |
| 0.3 | 1.941 | -0.568 | -0.568 | -1.641 |
| 0.4 | 1.8842 | -0.7321 | -0.7321 | -1.4842 |
| 0.5 | 1.81099 | -0.88052 | -0.88052 | -1.31099 |

(*b*) For Heun (without iterating the corrector) the first few steps are

| x | y | z | dy/dx | dz/dx | yend | zend | dy/dx | dz/dx | ave slope |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 2 | 0 | 0 | -2 | 2 | -0.2 | -0.2 | -1.9 | -0.1 |
| 0.1 | 1.99 | -0.195 | -0.195 | -1.89 | 1.9705 | -0.384 | -0.384 | -1.7705 | -0.2895 |
| 0.2 | 1.96105 | -0.37803 | -0.37803 | -1.76105 | 1.923248 | -0.55413 | -0.55413 | -1.62325 | -0.46608 |
| 0.3 | 1.914442 | -0.54724 | -0.54724 | -1.61444 | 1.859718 | -0.70868 | -0.70868 | -1.45972 | -0.62796 |
| 0.4 | 1.851646 | -0.70095 | -0.70095 | -1.45165 | 1.781551 | -0.84611 | -0.84611 | -1.28155 | -0.77353 |
| 0.5 | 1.774293 | -0.83761 | -0.83761 | -1.27429 | 1.690532 | -0.96504 | -0.96504 | -1.09053 | -0.90132 |

Both results are plotted below:



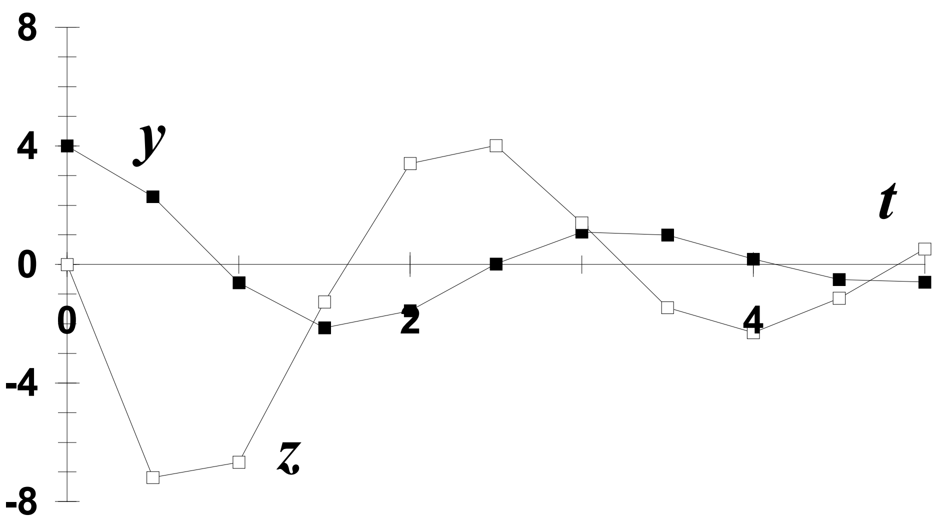

25.8 The second-order ODE is transformed into a pair of first-order ODEs as in

$$\frac{dy}{dt} = z \qquad y(0)=4$$

$$\frac{dz}{dt} = -0.5z - 5y \qquad z(0)=0$$

The results for the 4$^{th}$-order RK method are tabulated and plotted below:

| t | y | z | k11 | k12 | ymid | zmid | k21 | k22 | ymid | zmid | k31 | k32 | yend | zend | k41 | k42 | phi1 | phi2 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 4.0000 | 0.0000 | 0.00 | -20.00 | 4.00 | -5.00 | -5.00 | -17.50 | 2.75 | -4.38 | -4.38 | -11.56 | 1.81 | -1.78 | -1.78 | -8.17 | -3.42 | -14.38 |
| 0.5 | 2.2891 | -7.1914 | -7.19 | -7.85 | 0.49 | -9.15 | -9.15 | 2.12 | 0.00 | -6.66 | -6.66 | 3.33 | -1.04 | 3.95 | 3.95 | 3.23 | -5.81 | 1.05 |
| 1 | -0.6167 | -6.6682 | -6.67 | 6.42 | -2.28 | -5.06 | -5.06 | 13.95 | -1.88 | -3.18 | -3.18 | 11.00 | -2.21 | 4.89 | 4.89 | 8.59 | -3.05 | 10.82 |
| 1.5 | -2.1393 | -1.2584 | -1.26 | 11.33 | -2.45 | 1.57 | 1.57 | 11.48 | -1.75 | 1.61 | 1.61 | 7.92 | -1.33 | 1.82 | 1.82 | 5.75 | 1.16 | 9.32 |
| 2 | -1.5614 | 3.3995 | 3.40 | 6.11 | -0.71 | 4.93 | 4.93 | 1.09 | -0.33 | 3.67 | 3.67 | -0.19 | 0.28 | -1.66 | -1.66 | -0.55 | 3.16 | 1.23 |
| 2.5 | 0.0172 | 4.0139 | 4.01 | -2.09 | 1.02 | 3.49 | 3.49 | -6.85 | 0.89 | 2.30 | 2.30 | -5.60 | 1.17 | -2.78 | -2.78 | -4.45 | 2.14 | -5.24 |
| 3 | 1.0852 | 1.3939 | 1.39 | -6.12 | 1.43 | -0.14 | -0.14 | -7.10 | 1.05 | -0.38 | -0.38 | -5.06 | 0.89 | -1.45 | -1.45 | -3.75 | -0.18 | -5.70 |
| 3.5 | 0.9945 | -1.4562 | -1.46 | -4.24 | 0.63 | -2.52 | -2.52 | -1.89 | 0.37 | -1.93 | -1.93 | -0.86 | 0.03 | 0.56 | 0.56 | -0.43 | -1.63 | -1.70 |
| 4 | 0.1790 | -2.3048 | -2.30 | 0.26 | -0.40 | -2.24 | -2.24 | 3.11 | -0.38 | -1.53 | -1.53 | 2.67 | -0.59 | 1.51 | 1.51 | 2.17 | -1.39 | 2.33 |
| 4.5 | -0.5150 | -1.1399 | -1.14 | 3.15 | -0.80 | -0.35 | -0.35 | 4.18 | -0.60 | -0.10 | -0.10 | 3.07 | -0.56 | 1.02 | 1.02 | 2.31 | -0.17 | 3.32 |
| 5 | -0.6001 | 0.5213 | 0.52 | 2.74 | -0.47 | 1.21 | 1.21 | 1.75 | -0.30 | 0.96 | 0.96 | 1.01 | -0.12 | -0.09 | -0.09 | 0.65 | 0.79 | 1.49 |




25.9 (*a*) The Heun method without iteration can be implemented as in the following table:

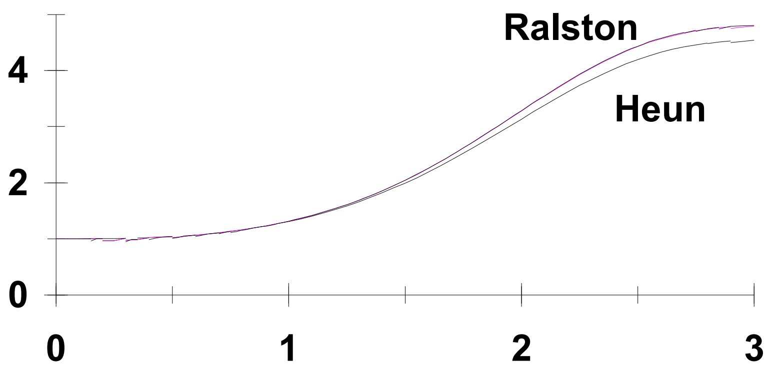
| *t* | *y* | $k_1$ | *yend* | $k_2$ | *phi* |
|---|---|---|---|---|---|
| 0 | 1 | 0 | 1 | 0.009967 | 0.004983 |
| 0.1 | 1.000498 | 0.009972 | 1.001496 | 0.039489 | 0.02473 |
| 0.2 | 1.002971 | 0.039587 | 1.00693 | 0.087592 | 0.063589 |
| 0.3 | 1.00933 | 0.088147 | 1.018145 | 0.153062 | 0.120604 |
| 0.4 | 1.021391 | 0.15489 | 1.03688 | 0.234765 | 0.194828 |
| 0.5 | 1.040874 | 0.239244 | 1.064798 | 0.331852 | 0.285548 |
| • | | | | | • |
| • | | | | | • |
| • | | | | | • |
| 2.9 | 4.527257 | 0.259141 | 4.553171 | 0.09016 | 0.17465 |
| 3 | 4.544722 | 0.090507 | 4.553773 | 0.007858 | 0.049183 |

(*b*) The Ralston 2$^{nd}$ order RK method can be implemented as in the following table:

| *t* | *y* | $k_1$ | *yint* | $k_2$ | *phi* |
|---|---|---|---|---|---|

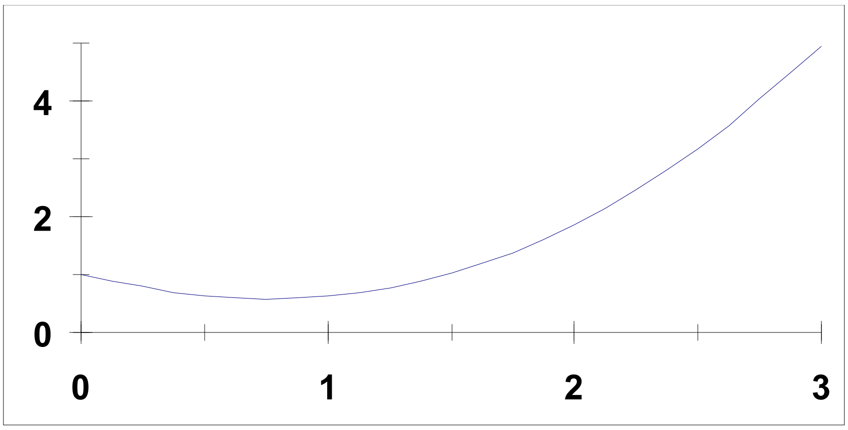
| | | | | | |
|---|---|---|---|---|---|
| 0 | 1 | 0 | 1 | 0.005614 | 0.003743 |
| 0.1 | 1.000374 | 0.00997 | 1.001122 | 0.030348 | 0.023555 |
| 0.2 | 1.00273 | 0.039577 | 1.005698 | 0.074158 | 0.062631 |
| 0.3 | 1.008993 | 0.088118 | 1.015602 | 0.136249 | 0.120205 |
| 0.4 | 1.021013 | 0.154833 | 1.032626 | 0.215982 | 0.195599 |
| 0.5 | 1.040573 | 0.239175 | 1.058511 | 0.313061 | 0.288432 |
| • | | | | | • |
| • | | | | | • |
| • | | | | | • |
| 2.9 | 4.779856 | 0.2736 | 4.800376 | 0.131997 | 0.179198 |
| 3 | 4.797775 | 0.095547 | 4.804941 | 0.021276 | 0.046033 |

Both methods are displayed on the following plot along with the exact solution. The Ralston method performs much better for this case.




25.10 The solution results are as in the following table and plot:

| *t* | *y* | *k1* | *k2* | *k3* | *phi* |
|---|---|---|---|---|---|
| 0 | 1 | -1 | -0.6875 | -0.5625 | -0.71875 |
| 0.5 | 0.640625 | -0.39063 | 0.019531 | 0.144531 | -0.02799 |
| 1 | 0.626628 | 0.373372 | 0.842529 | 0.967529 | 0.78517 |
| 1.5 | 1.019213 | 1.230787 | 1.735591 | 1.860591 | 1.67229 |
| 2 | 1.855358 | 2.144642 | 2.670982 | 2.795982 | 2.604092 |
| 2.5 | 3.157404 | 3.092596 | 3.631947 | 3.756947 | 3.562889 |
| 3 | 4.938848 | 4.061152 | 4.608364 | 4.733364 | 4.537995 |




25.11 (*a*) Euler

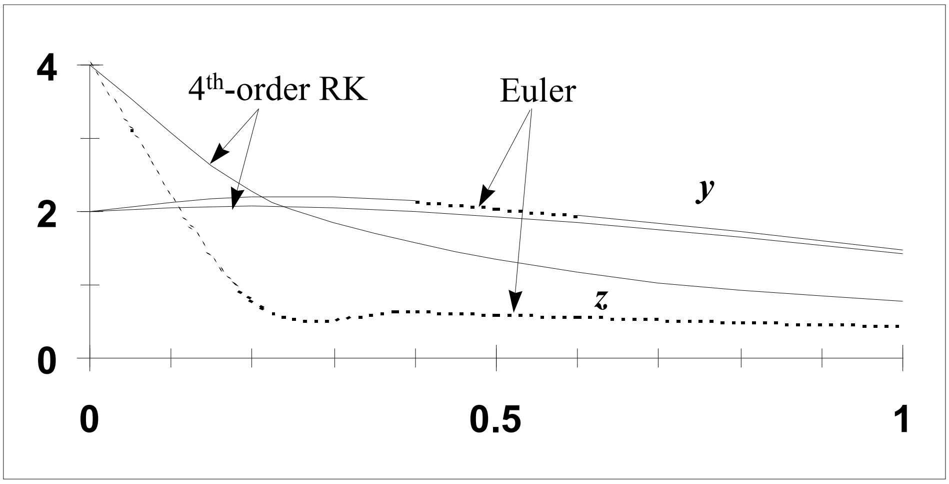
| *x* | *y* | *z* | *dy/dx* | *dz/dx* |
|---|---|---|---|---|
| 0 | 2.0000 | 4.0000 | 1.00 | -16.00 |

| | | | | |
|---|---|---|---|---|
| 0.2 | 2.2000 | 0.8000 | -0.31 | -0.70 |
| 0.4 | 2.1387 | 0.6592 | -0.93 | -0.46 |
| 0.6 | 1.9536 | 0.5663 | -1.16 | -0.31 |
| 0.8 | 1.7209 | 0.5036 | -1.20 | -0.22 |
| 1 | 1.4819 | 0.4600 | -1.12 | -0.16 |

(*b*) 4th-order RK

| *x* | *y* | *z* | *k11* | *k12* | *k21* | *k22* | *k31* | *k32* | *k41* | *k42* | *phi1* | *phi2* |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 2.000 | 4.000 | 1.000 | -16.000 | 0.324 | -6.048 | 0.459 | -11.714 | -0.090 | -0.123 | 0.413 | -8.608 |
| 0.2 | 2.083 | 2.278 | -0.071 | -5.406 | -0.447 | -3.134 | -0.372 | -3.934 | -0.665 | -1.686 | -0.396 | -3.538 |
| 0.4 | 2.003 | 1.571 | -0.655 | -2.472 | -0.843 | -1.698 | -0.806 | -1.884 | -0.941 | -2.438 | -0.816 | -2.012 |
| 0.6 | 1.840 | 1.168 | -0.937 | -1.256 | -1.010 | -0.950 | -0.996 | -1.002 | -1.036 | -2.207 | -0.997 | -1.228 |
| 0.8 | 1.641 | 0.923 | -1.035 | -0.699 | -1.042 | -0.559 | -1.040 | -0.577 | -1.026 | -1.667 | -1.038 | -0.773 |
| 1 | 1.433 | 0.768 | -1.027 | -0.423 | -0.997 | -0.351 | -1.003 | -0.358 | -0.960 | -1.143 | -0.998 | -0.497 |

Both methods are plotted on the same graph below. Notice how Euler's method (particularly for *z*) is very inaccurate for this step size. The 4$^{th}$-order RK is much closer to the exact solution.




25.12 $$\frac{dy}{dx} = 10e^{-\frac{(x-2)^2}{2(0.075)^2}} - 0.6y$$

<u>4$^{th}$-order RK method:</u>

One step ($h$ = 0.5): $y_1 = 0.3704188$
Two steps ($h$ = 0.25): $y_2 = 0.3704096$

$\Delta_{present} = -9.119\times10-6$

$$\text{correction} = \frac{\Delta}{15} = -6.08\times10^{-7}$$

$y_2 = 0.370409$

$$\frac{dy}{dx} = -0.3$$

$$y_{\text{scale}} = 0.5 + |0.5(-0.3)| = 0.65$$

$\Delta_{new} = 0.001(0.65) = 0.00065$

Since $\Delta_{present} < \Delta_{new}$, therefore, increase step.

$$h_{new} = 0.5\left|\frac{0.00065}{9.119\times10^{-6}}\right|^{0.2} = 1.1737$$

25.13 We will look at the first step only

$\Delta_{present} = y_2 - y_1 = -0.24335$

$$\frac{dy}{dx} = 4e^0 - 0.5(2) = 3$$

$y_{scale} = 2 + (2(3)) = 8$

$\Delta_{new} = 0.001(8) = 0.008$

Because $\Delta_{present} > \Delta_{new}$, decrease step.

25.14 The calculation of the $k$'s can be summarized in the following table:

| | ***x*** | ***y*** | ***f(x,y)*** | ***k*** |
|---|---|---|---|---|
| ***k1*** | 0 | 2 | 3 | 3 |
| ***k2*** | 0.25 | 2.75 | 3.510611 | 3.510611 |
| ***k3*** | 0.375 | 3.268609 | 3.765131 | 3.765131 |
| ***k4*** | 0.923077 | 5.636774 | 5.552467 | 5.552467 |
| ***k5*** | 1 | 5.878223 | 5.963052 | 5.963052 |
| ***k6*** | 0.5 | 3.805418 | 4.06459 | 4.06459 |

These can then be used to compute the 4th-order prediction

$$y_1 = 2 + \left(\frac{25}{216}3 + \frac{1408}{2565}3.765131 + \frac{2197}{4104}5.552467 - \frac{1}{5}5.963052\right)1 = 6.193807$$

along with a fifth-order formula:

$$y_1 = 2 + \left(\frac{16}{135}3 + \frac{6656}{12{,}825}3.765131 + \frac{28{,}561}{56{,}430}5.552467 - \frac{9}{50}5.963052 + \frac{2}{55}4.06459\right)1 = 6.194339$$

The error estimate is obtained by subtracting these two equations to give

$$E_a = 6.194339 - 6.193807 = 0.000532$$

25.15

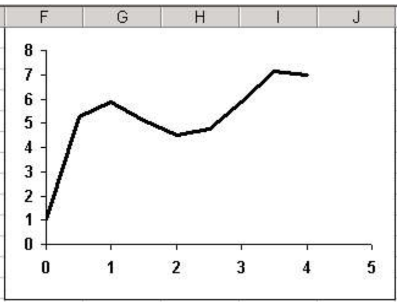
```
Option Explicit

Sub EulerTest()
Dim i As Integer, m As Integer
Dim xi As Single, yi As Single, xf As Single, dx As Single, xout As Single
Dim xp(200) As Single, yp(200) As Single
'Assign values
yi = 1
xi = 0
```

```
xf = 4
dx = 0.5
xout = 0.5
'Perform numerical Integration of ODE
Call ODESolver(xi, yi, xf, dx, xout, xp(), yp(), m)
'Display results
Sheets("Sheet1").Select
Range("a5:b205").ClearContents
Range("a5").Select
For i = 0 To m
  ActiveCell.Value = xp(i)
  ActiveCell.Offset(0, 1).Select
  ActiveCell.Value = yp(i)
  ActiveCell.Offset(1, -1).Select
Next i
Range("a5").Select
End Sub

Sub ODESolver(xi, yi, xf, dx, xout, xp, yp, m)
'Generate an array that holds the solution
Dim x As Single, y As Single, xend As Single
Dim h As Single
m = 0
xp(m) = xi
yp(m) = yi
x = xi
y = yi
Do       'Print loop
  xend = x + xout
  If (xend > xf) Then xend = xf  'Trim step if increment exceeds end
  h = dx
  Call Integrator(x, y, h, xend)
  m = m + 1
  xp(m) = x
  yp(m) = y
  If (x >= xf) Then Exit Do
Loop
End Sub

Sub Integrator(x, y, h, xend)
Dim ynew As Single
Do       'Calculation loop
  If (xend - x < h) Then h = xend - x  'Trim step if increment exceeds end
  Call Euler(x, y, h, ynew)
  y = ynew
  If (x >= xend) Then Exit Do
Loop
End Sub

Sub Euler(x, y, h, ynew)
Dim dydx As Single
'Implement Euler's method
Call Derivs(x, y, dydx)
ynew = y + dydx * h
x = x + h
End Sub

Sub Derivs(x, y, dydx)
'Define ODE
dydx = -2 * x ^ 3 + 12 * x ^ 2 - 20 * x + 8.5
End Sub
```

25.16 Example 25.1:

| x | y |
|---|---|
| 0 | 1 |
| 0.5 | 5.25 |
| 1 | 5.875 |
| 1.5 | 5.125 |
| 2 | 4.5 |
| 2.5 | 4.75 |
| 3 | 5.875 |
| 3.5 | 7.125 |
| 4 | 7 |

RUN



Example 25.4 (nonlinear model). Change time steps and initial conditions to

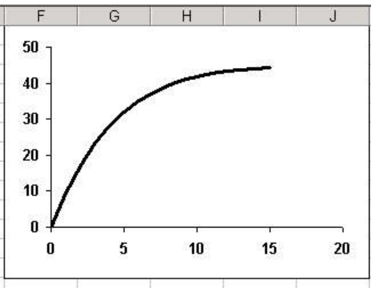
```
'Assign values
yi = 0
xi = 0
xf = 15
dx = 0.5
xout = 1
```

Change Derivs Sub to

```
Sub Derivs(t, v, dvdt)
'Define ODE
dvdt = 9.8 - 12.5 / 68.1 * (v + 8.3 * (v / 46) ^ 2.2)
End Sub
```

| x | y |
|---|---|
| 0 | 0 |
| 1 | 9.34477 |
| 2 | 16.98754 |
| 3 | 23.1603 |
| 4 | 28.08878 |
| 5 | 31.98533 |
| 6 | 35.04111 |
| 7 | 37.42184 |
| 8 | 39.26696 |
| 9 | 40.69109 |
| 10 | 41.78674 |
| 11 | 42.62757 |
| 12 | 43.2716 |
| 13 | 43.76415 |
| 14 | 44.14042 |
| 15 | 44.42762 |

RUN



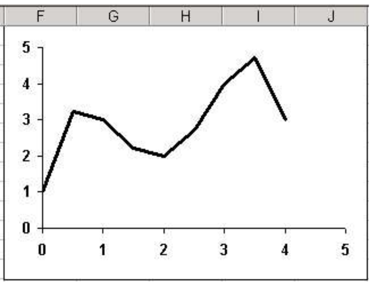
25.17

```
Option Explicit

Sub RK4Test()
Dim i As Integer, m As Integer
Dim xi As Single, yi As Single, xf As Single, dx As Single, xout As Single
Dim xp(200) As Single, yp(200) As Single

'Assign values
yi = 1
xi = 0
xf = 4
dx = 0.5
xout = 0.5

'Perform numerical Integration of ODE
Call ODESolver(xi, yi, xf, dx, xout, xp(), yp(), m)

'Display results
Sheets("Sheet1").Select
Range("a5:b205").ClearContents
Range("a5").Select
For i = 0 To m
  ActiveCell.Value = xp(i)
```

```
    ActiveCell.Offset(0, 1).Select
    ActiveCell.Value = yp(i)
    ActiveCell.Offset(1, -1).Select
  Next i
  Range("a5").Select
  End Sub

  Sub ODESolver(xi, yi, xf, dx, xout, xp, yp, m)
  'Generate an array that holds the solution
  Dim x As Single, y As Single, xend As Single
  Dim h As Single
  m = 0
  xp(m) = xi
  yp(m) = yi
  x = xi
  y = yi
  Do       'Print loop
    xend = x + xout
    If (xend > xf) Then xend = xf  'Trim step if increment exceeds end
    h = dx
    Call Integrator(x, y, h, xend)
    m = m + 1
    xp(m) = x
    yp(m) = y
    If (x >= xf) Then Exit Do
  Loop
  End Sub

  Sub Integrator(x, y, h, xend)
  Dim ynew As Single
  Do       'Calculation loop
    If (xend - x < h) Then h = xend - x  'Trim step if increment exceeds end
    Call RK4(x, y, h, ynew)
    y = ynew
    If (x >= xend) Then Exit Do
  Loop
  End Sub

  Sub RK4(x, y, h, ynew)
  'Implement RK4 method
  Dim k1 As Single, k2 As Single, k3 As Single, k4 As Single
  Dim ym As Single, ye As Single, slope As Single
  Call Derivs(x, y, k1)
  ym = y + k1 * h / 2
  Call Derivs(x + h / 2, ym, k2)
  ym = y + k2 * h / 2
  Call Derivs(x + h / 2, ym, k3)
  ye = y + k3 * h
  Call Derivs(x + h, ye, k4)
  slope = (k1 + 2 * (k2 + k3) + k4) / 6
  ynew = y + slope * h
  x = x + h
  End Sub

  Sub Derivs(x, y, dydx)
  'Define ODE
  dydx = -2 * x ^ 3 + 12 * x ^ 2 - 20 * x + 8.5
  End Sub
```

25.18 Example 25.1:

| | A | B | C | D | E |
|---|---|---|---|---|---|
| 4 | x | y | | | |
| 5 | 0 | 1 | | RUN | |
| 6 | 0.5 | 3.21875 | | | |
| 7 | 1 | 3 | | | |
| 8 | 1.5 | 2.21875 | | | |
| 9 | 2 | 2 | | | |
| 10 | 2.5 | 2.71875 | | | |
| 11 | 3 | 4 | | | |
| 12 | 3.5 | 4.71875 | | | |
| 13 | 4 | 3 | | | |
| 14 | | | | | |
| 15 | | | | | |
| 16 | | | | | |



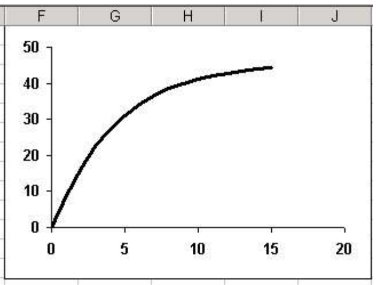
Example 25.5 Change time steps and initial conditions to

```
'Assign values
yi = 2
xi = 0
xf = 4
dx = 1
xout = 1
```

Change Derivs Sub to

```
Sub Derivs(x, y, dydx)
'Define ODE
dydx = 4 * Exp(0.8 * x) - 0.5 * y
End Sub
```

| | A | B | C | D | E |
|---|---|---|---|---|---|
| 4 | x | y | | | |
| 5 | 0 | 0 | | RUN | |
| 6 | 1 | 8.940161 | | | |
| 7 | 2 | 16.30683 | | | |
| 8 | 3 | 22.31247 | | | |
| 9 | 4 | 27.16182 | | | |
| 10 | 5 | 31.04566 | | | |
| 11 | 6 | 34.13524 | | | |
| 12 | 7 | 36.57947 | | | |
| 13 | 8 | 38.50458 | | | |
| 14 | 9 | 40.01546 | | | |
| 15 | 10 | 41.19793 | | | |
| 16 | 11 | 42.12132 | | | |
| 17 | 12 | 42.84114 | | | |
| 18 | 13 | 43.40151 | | | |
| 19 | 14 | 43.83729 | | | |
| 20 | 15 | 44.1759 | | | |




25.19

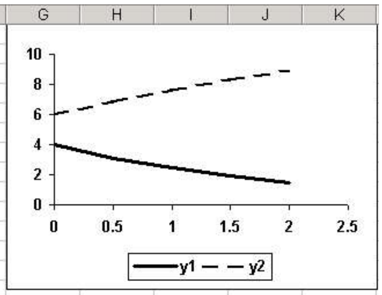
```
Option Explicit

Sub RK4SysTest()
Dim i As Integer, m As Integer, n As Integer, j As Integer
Dim xi As Single, yi(10) As Single, xf As Single, dx As Single, xout As Single
Dim xp(200) As Single, yp(200, 10) As Single

'Assign values
n = 2
xi = 0
xf = 2
yi(1) = 4
yi(2) = 6
dx = 0.5
xout = 0.5

'Perform numerical Integration of ODE
Call ODESolver(xi, yi(), xf, dx, xout, xp(), yp(), m, n)

'Display results
Sheets("Sheet1").Select
Range("a5:n205").ClearContents
Range("a5").Select
For i = 0 To m
  ActiveCell.Value = xp(i)
  For j = 1 To n
    ActiveCell.Offset(0, 1).Select
    ActiveCell.Value = yp(i, j)
  Next j
  ActiveCell.Offset(1, -n).Select
Next i
Range("a5").Select
End Sub

Sub ODESolver(xi, yi, xf, dx, xout, xp, yp, m, n)
'Generate an array that holds the solution
Dim i As Integer
```

```
Dim x As Single, y(10) As Single, xend As Single
Dim h As Single
m = 0
x = xi
'set initial conditions
For i = 1 To n
  y(i) = yi(i)
Next i
'save output values
xp(m) = x
For i = 1 To n
  yp(m, i) = y(i)
Next i
Do       'Print loop
  xend = x + xout
  If (xend > xf) Then xend = xf  'Trim step if increment exceeds end
  h = dx
  Call Integrator(x, y(), h, n, xend)
  m = m + 1
  'save output values
  xp(m) = x
  For i = 1 To n
    yp(m, i) = y(i)
  Next i
  If (x >= xf) Then Exit Do
Loop
End Sub

Sub Integrator(x, y, h, n, xend)
Dim j As Integer
Dim ynew(10) As Single
Do       'Calculation loop
  If (xend - x < h) Then h = xend - x  'Trim step if increment exceeds end
  Call RK4Sys(x, y, h, n, ynew())
  For j = 1 To n
    y(j) = ynew(j)
  Next j
  If (x >= xend) Then Exit Do
Loop
End Sub

Sub RK4Sys(x, y, h, n, ynew)
Dim j As Integer
Dim dydx(10) As Single

Dim ym(10), ye(10)
Dim k1(10), k2(10), k3(10), k4(10)
Dim slope(10)
'Implement RK4 method for systems of ODEs
Call Derivs(x, y, k1())
For j = 1 To n
  ym(j) = y(j) + k1(j) * h / 2
Next j
Call Derivs(x + h / 2, ym, k2())
For j = 1 To n
  ym(j) = y(j) + k2(j) * h / 2
Next j
Call Derivs(x + h / 2, ym, k3())
For j = 1 To n
  ye(j) = y(j) + k3(j) * h
Next j
Call Derivs(x + h, ye, k4())
For j = 1 To n
  slope(j) = (k1(j) + 2 * (k2(j) + k3(j)) + k4(j)) / 6
Next j
For j = 1 To n
  ynew(j) = y(j) + slope(j) * h
Next j
x = x + h

End Sub
```

```
Sub Derivs(x, y, dydx)
'Define ODE
dydx(1) = -0.5 * y(1)
dydx(2) = 4 - 0.3 * y(2) - 0.1 * y(1)
End Sub
```

Application to Example 25.10:

| | A | B | C | D | E | F |
|---|---|---|---|---|---|---|
| 1 | Example 25.10 | | | | | |
| 2 | | | | | | |
| 3 | | | | | | |
| 4 | x | y1 | y2 | | | |
| 5 | 0 | 4 | 6 | | RUN | |
| 6 | 0.5 | 3.115234 | 6.85767 | | | |
| 7 | 1 | 2.426171 | 7.632106 | | | |
| 8 | 1.5 | 1.889523 | 8.326886 | | | |
| 9 | 2 | 1.471577 | 8.946865 | | | |
| 10 | | | | | | |
| 11 | | | | | | |
| 12 | | | | | | |
| 13 | | | | | | |
| 14 | | | | | | |

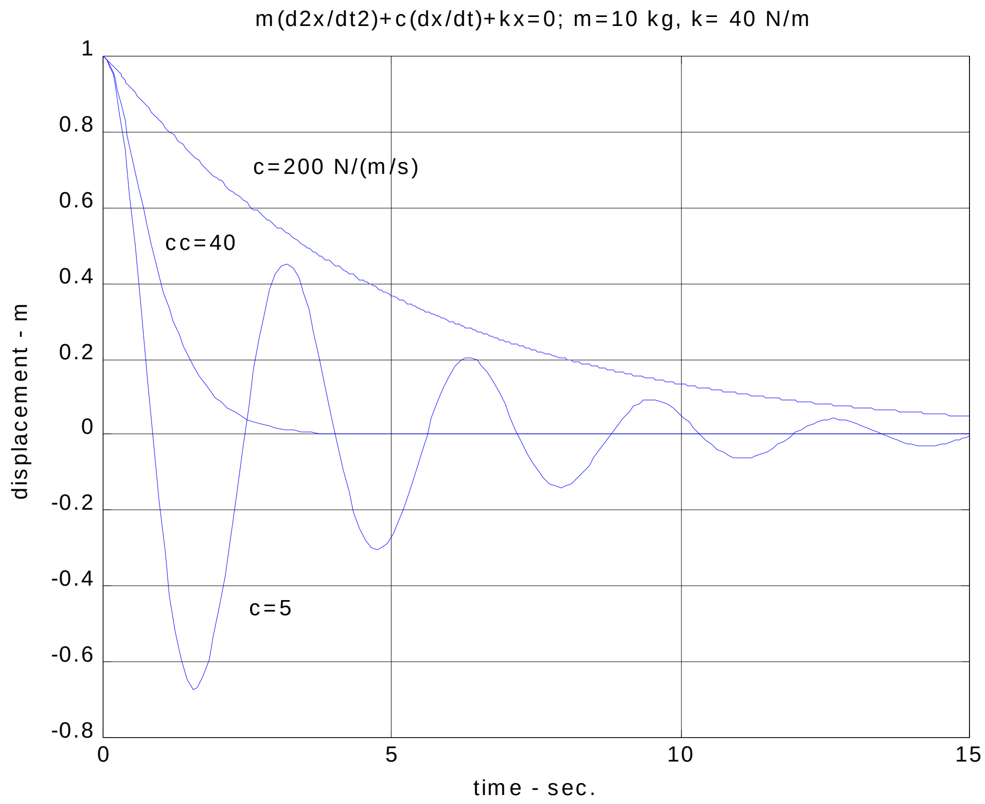



25.20 Main Program:

```
%Damped spring mass system
   %mass:  m=10 kg
   %damping: c=5,40,200 N/(m/s)
   %spring: k=40 N/m
        % MATLAB 5 version
%Independent Variable t, tspan=[tstart tstop]
%initial conditions [x(1)=velocity, x(2)=displacement];

tspan=[0 15];  ic=[0 1];

    global cm km
    m=10; c(1)=5; c(2)=40; c(3)=200; k=40;
    km=k/m;

    for n=1:3
        cm=c(n)/m
    [t,x]=ode45('kc',tspan,ic);
    plot(t,x(:,2)); grid;
    xlabel('time - sec.'); ylabel('displacement - m');
    title('m(d2x/dt2)+c(dx/dt)+kx=0; m=10 kg, k= 40 N/m')
    hold on
end
    gtext('c=5');gtext('cc=40');gtext('c=200 N/(m/s)')
```

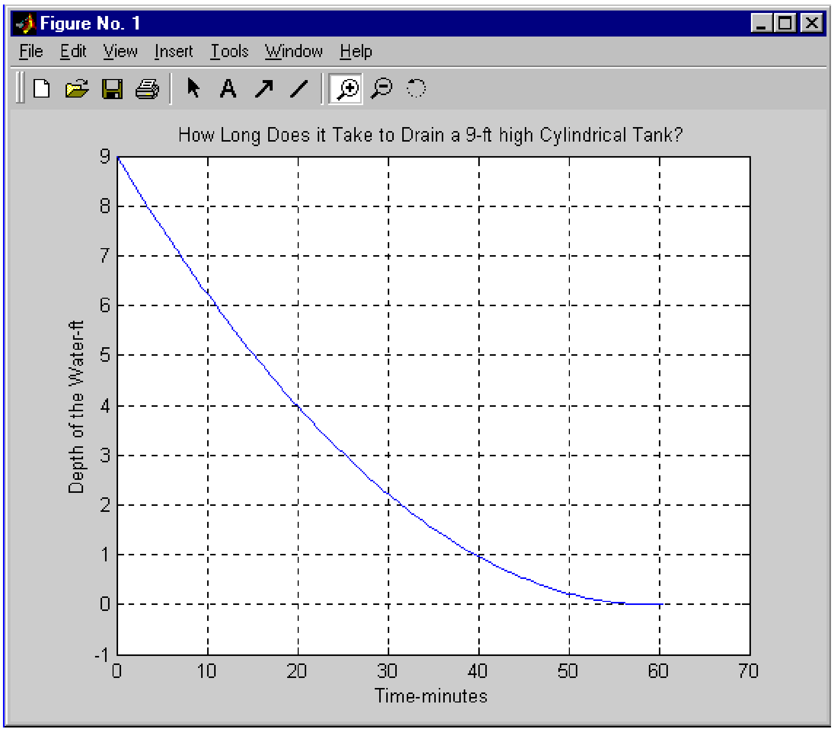
Function ‘kc’:

```
%Damped spring mass system - m d2x/dt2 + c dx/dt + k x =0
  %mass:  m=10 kg
  %damping: c=5,40,200 N/(m/s)
  %spring: k=40 N/m
%x(1)=velocity, x(2)=dispacement

function dx=kc(t,x);
global cm km
dx=[-cm*x(1)-km*x(2); x(1)];
```




25.21 The Matlab program on the following pages performs the Euler Method and the plots shows the depth of the water vs. time. From the plot, we approximate that it takes about 58 minutes to drain the cylindrical tank.

```
%euler.m
dt=0.5;
max=60;
n=max/dt+1;
t=zeros(1,n);
y=zeros(1,n);
t(1)=0;
y(1)=9;
for i=1:n
   y(i+1)=y(i)+dydt(t(i),y(i))*dt;
   t(i+1)=t(i)+dt;
end
plot(t,y)
grid
xlabel('Time-minutes')
ylabel('Depth of the Water-ft')
title('How Long Does it Take to Drain a 9-ft high Cylindrical
Tank?')
zoom

function dy=dydt(t,y);
dy=-0.1*sqrt(y);
```




25.22

$$x = x(1)$$

$$v = \frac{dx}{dt} = x(2)$$

$$\frac{dv}{dt} = \frac{d}{dt}\left(\frac{dx}{dt}\right) = \frac{d^2x}{dt^2}$$

$$\frac{dx(1)}{dt} = x(2)$$

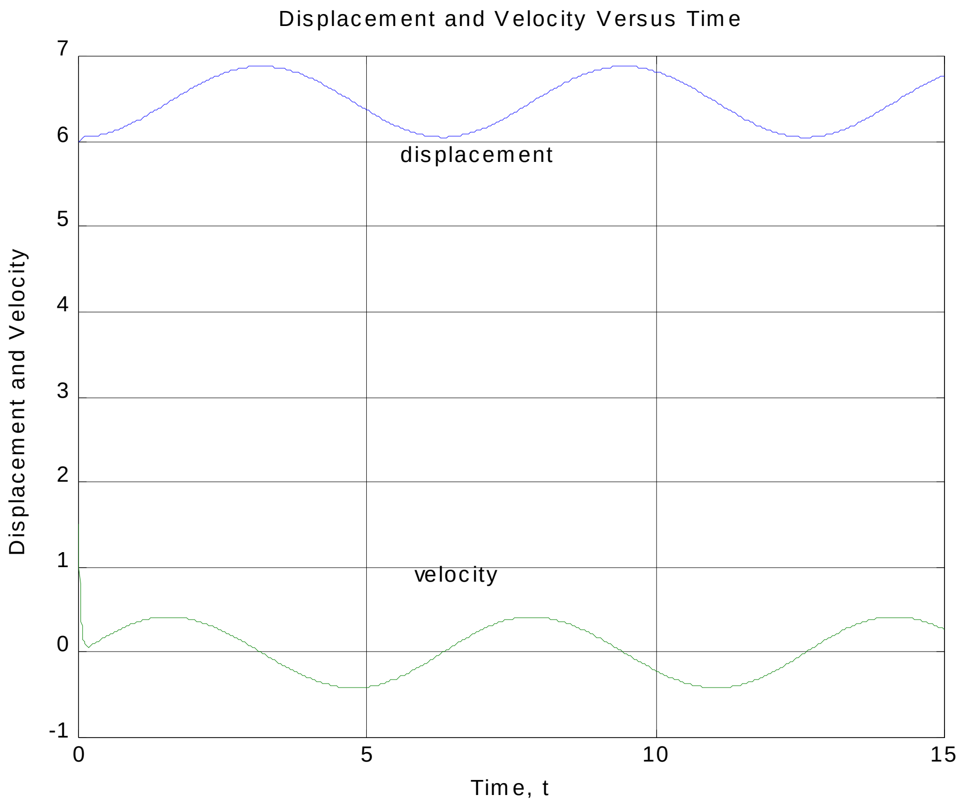
$$\frac{dx(2)}{dt} = -5x(1)x(2) - (x(1)+7)\sin(t)$$

$$x(1)(t=0) = 6$$

$$x(2)(t=0) = 1.5$$

```
tspan=[0,15]';
x0=[6,1.5]';
[t,x]=ode45('dxdt',tspan,x0);
plot(t,x(:,1),t,x(:,2),'--')
grid
title('Displacement and Velocity Versus Time')
xlabel('Time, t')
ylabel('Displacement and Velocity')
gtext('displacement')
gtext('velocity')
```

```
function dx=dxdt(t,x)
dx=[x(2);-5*x(1)*x(2)+(x(1)+7)*sin(1*t)];
```



Displacement and Velocity Versus Time
displacement
velocity
Displacement and Velocity
Time, t
7
6
5
4
3
2
1
0
-1
0
5
10
15

## CHAPTER 26

26.1 (*a*) $h < 2/100{,}000 = 2\times10^{-5}$.

(*b*) The implicit Euler can be written for this problem as

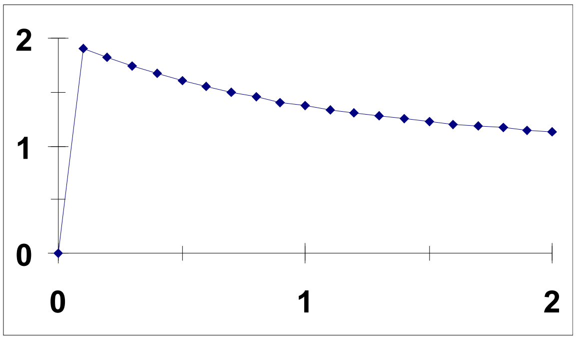
$$y_{i+1} = y_i + \left(-100{,}000\,y_{i+1} + 100{,}000e^{-x_{i+1}} - e^{-x_{i+1}}\right)h$$

which can be solved for

$$y_{i+1} = \frac{y_i + 100{,}000e^{-x_{i+1}}h - e^{-x_{i+1}}h}{1 + 100{,}000h}$$

The results of applying this formula for the first few steps are shown below. A plot of the entire solution is also displayed

| x | y |
|---|---|
| 0 | 0 |
| 0.1 | 1.904638 |
| 0.2 | 1.818731 |
| 0.3 | 1.740819 |
| 0.4 | 1.67032 |
| 0.5 | 1.606531 |




26.2 The implicit Euler can be written for this problem as

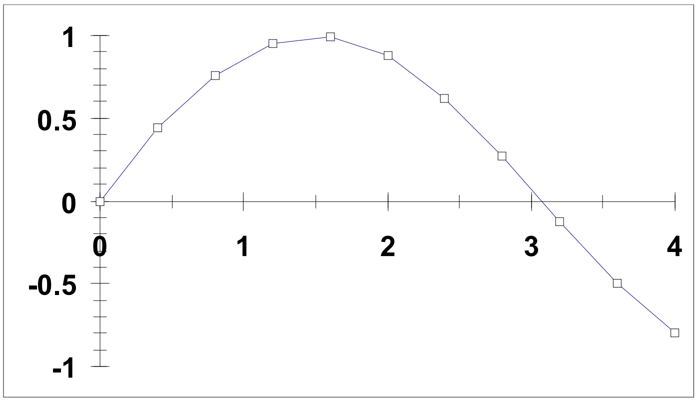
$$y_{i+1} = y_i + \left(30(\sin t_{i+1} - y_{i+1}) + 3\cos t_{i+1}\right)h$$

which can be solved for

$$y_{i+1} = \frac{y_i + 30\sin t_{i+1}h + 3\cos t_{i+1}h}{1 + 30h}$$

The results of applying this formula are tabulated and graphed below.

| x | y | x | y | x | y | x | y |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 1.2 | 0.952306 | 2.4 | 0.622925 | 3.6 | -0.50089 |
| 0.4 | 0.444484 | 1.6 | 0.993242 | 2.8 | 0.270163 | 4 | -0.79745 |
| 0.8 | 0.760677 | 2 | 0.877341 | 3.2 | -0.12525 | | |




26.3 (*a*) The explicit Euler can be written for this problem as

$$x_{1,i+1} = x_{1,i} + \left(999x_{1,i} + 1999x_{2,i}\right)h$$
$$x_{2,i+1} = x_{2,i} + \left(-1000x_{1,i} - 2000x_{2,i}\right)h$$

Because the step-size is much too large for the stability requirements, the solution is unstable,

| t | x1 | x2 | dx1 | dx2 |
|---|---|---|---|---|
| 0 | 1 | 1 | 2998 | -3000 |
| 0.05 | 150.9 | -149 | -147102 | 147100 |
| 0.1 | -7204.2 | 7206 | 7207803 | -7207805 |
| 0.15 | 353186 | -353184 | -3.5E+08 | 3.53E+08 |
| 0.2 | -1.7E+07 | 17305943 | 1.73E+10 | -1.7E+10 |

(*b*) The implicit Euler can be written for this problem as

$$x_{1,i+1} = x_{1,i} + \left(999x_{1,i+1} + 1999x_{2,i+1}\right)h$$
$$x_{2,i+1} = x_{2,i} + \left(-1000x_{1,i+1} - 2000x_{2,i+1}\right)h$$

or collecting terms

$$(1-999h)x_{1,i+1} - 1999hx_{2,i+1} = x_{1,i}$$
$$1000hx_{1,i+1} + (1+2000h)x_{2,i+1} = x_{2,i}$$

or substituting $h$ = 0.05 and expressing in matrix format

$$\begin{bmatrix} -48.95 & -99.95 \\ 50 & 101 \end{bmatrix} \begin{Bmatrix} x_{1,i+1} \\ x_{2,i+1} \end{Bmatrix} = \begin{Bmatrix} x_{1,i} \\ x_{2,i} \end{Bmatrix}$$

Thus, to solve for the first time step, we substitute the initial conditions for the right-hand side and solve the 2x2 system of equations. The best way to do this is with LU decomposition since we will have to solve the system repeatedly. For the present case, because its easier to display, we will use the matrix inverse to obtain the solution. Thus, if the matrix is inverted, the solution for the first step amounts to the matrix multiplication,

$$\begin{Bmatrix} x_{1,i+1} \\ x_{2,i+1} \end{Bmatrix} = \begin{bmatrix} 1.886088 & 1.86648 \\ -0.93371 & -0.9141 \end{bmatrix} \begin{Bmatrix} 1 \\ 1 \end{Bmatrix} = \begin{Bmatrix} 3.752568 \\ -1.84781 \end{Bmatrix}$$

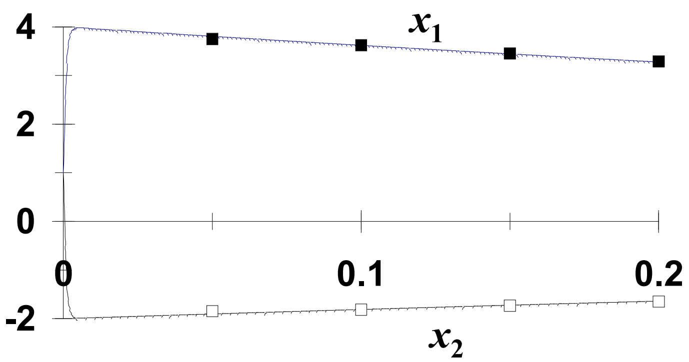
For the second step (from $x$ = 0.05 to 0.1),

$$\begin{Bmatrix} x_{1,i+1} \\ x_{2,i+1} \end{Bmatrix} = \begin{bmatrix} 1.886088 & 1.86648 \\ -0.93371 & -0.9141 \end{bmatrix} \begin{Bmatrix} 3.752568 \\ -1.84781 \end{Bmatrix} = \begin{Bmatrix} 3.62878 \\ -1.81472 \end{Bmatrix}$$

The remaining steps can be implemented in a similar fashion to give

| t | x1 | x2 |
|---|---|---|
| 0 | 1 | 1 |
| 0.05 | 3.752568 | -1.84781 |
| 0.1 | 3.62878 | -1.81472 |
| 0.15 | 3.457057 | -1.72938 |
| 0.2 | 3.292457 | -1.64705 |

The results are plotted below, along with a solution with the explicit Euler using a step of 0.0005.




26.4 <u>First step:</u>

Predictor:

$y_1^0$=5.222138+[−0.5(4.143883)+$e^{-2}$]1 = 3.285532

Corrector:

$$y_1^{\,1} = 4.143883 + \frac{-0.5(4.143883) + e^{-2} - 0.5(3.285532) + e^{-2.5}}{2} 0.5 = 3.269562$$

The corrector can be iterated to yield

| $j$ | $y_{i+1}^j$ | $\lvert\varepsilon_a\rvert$,% |
|---|---|---|
| 1 | 3.269562 | |
| 2 | 3.271558 | 0.061 |

<u>Second step:</u>

Predictor:

$y_2^0$=4.143883+[−0.5(3.271558)+$e^{-2.5}$]1 = 2.590189

Predictor Modifier:

$y_2^0$ = 2.590189+4/5(3.271558-3.285532) = 2.579010

Corrector:

$$y_2^{\,1} = 3.271558 + \frac{-0.5(3.271558) + e^{-2.5} - 0.5(2.579010) + e^{-3}}{2} 0.5 = 2.573205$$

The corrector can be iterated to yield

| $j$ | $y_{i+1}^{j}$ | $\lvert \varepsilon_a \rvert$,% |
|---|---|---|
| 1 | 2.573205 | |
| 2 | 2.573931 | 0.0282 |

26.5

```
predictor =        3.270674927
Corrector Iteration
x           y                 ea
2.5    3.274330476       1.12E-01
2.5    3.273987768       1.05E-02
2.5    3.274019897       9.81E-04

predictor =        2.576436209
Corrector Iteration
x           y                 ea
3           2.57830404        7.24E-02
3           2.578128931       6.79E-03
3           2.377128276       3.32E-02
3           2.377202366       3.12E-03
```

26.6

(a) non self-starting Huen

predictor

$$y^{0}_{4.5} = 0.8571429 + f(4, 0.75)(1) = 0.6696429$$

corrector

$$y^{1}_{4.5} = 0.75 + \frac{\left[f(4, 0.75) + f(4.5, 0.6696429)\right]}{2}(.5) = 0.6659226$$

$$\epsilon_a = \left| \frac{0.6659226 - 0.6696429}{0.6659226} \right| \times 100 = 0.56\ \% < 1\%$$

Next step

predictor

$$y^{0}_{5} = 0.75 + f(4.5, 0.6659226)(1) = 0.6020172 \quad \text{unmodified}$$

$$y^{0}_{5} = 0.6020172 + \frac{4}{5}(0.6659226 - 0.6696429)$$

$$= 0.599041 \quad \text{modified}$$

corrector

$$y^{1}_{5} = 0.6659226 + \frac{f(4.5, 0.6659226) + f(5, 0.599041)}{2}(.5)$$

$$= 0.5989749$$

$\epsilon_a = 0.51\%$

$< 1\%$

(b)

```
predictor =        0.669232229
Corrector Iteration
```

| x | y | ea | et |
|---|---|---|---|
| 4.5 | 0.666462335 | 4.16E-01 | 0.030654791 |
| 4.5 | 0.666577747 | 1.73E-02 | 0.013342954 |
| 4.5 | 0.666572938 | 7.21E-04 | 0.014064281 |

predictor = 0.601036948

Corrector Iteration

| x | y | ea | et |
|---|---|---|---|
| 5 | 0.599829531 | 2.01E-01 | 0.028411529 |
| 5 | 0.599874809 | 7.55E-03 | 0.020865171 |

26.7 use 4th order RK to generate

| x | y |
|---|---|
| −0.25 | 1.277355 |
| 0 | 1.000006 |
| 0.25 | 0.7828768 |
| 0.50 | 0.6323447 |

Now use $y(0) = 1.000006$

and $y(-0.25) = 1.277355$

to implement non-self starting Huen

predictor

$$y^0_{0.25} = 1.277355 + f(0, 1.000006)(0.5) = 0.777352$$

corrector

$$y^1_{0.25} = 1.000006 + \left[\frac{f(0, 1.000006) + f(0.25, 0.777352)}{2}\right](0.25)$$

$$= 0.7839093$$

now iterate

$$y^2_{0.25} = 0.7831408$$

$$y^3_{0.25} = 0.7832309$$

$$y^6_{0.25} = 0.7832214$$

Next step

predictor

un modified

$$y^0_{0.5} = 1.000006 + f(0.25, 0.7832214)(0.5) = 0.6328709$$

$$y^0_{0.5} = 0.6328709 + \frac{4}{5}(0.7832214 - 0.777352) = 0.6375664$$

corrector

$$y^1_{0.5} = 0.7832214 + \left[\frac{f(0.25, 0.7832214) + f(0.5, 0.6375664)}{2}\right] = 0.6316658$$

now iterate $y^5_{0.5} = 0.6321715$

26.8

predictor = 0.737731653

Corrector Iteration

| x | y | ea |
|---|---|---|
| 2 | 0.660789924 | 1.16E+01 |
| 2 | 0.665598782 | 7.22E-01 |
| 2 | 0.665298229 | 4.52E-02 |
| 2 | 0.665317013 | 0.002823406 |

predictor = 0.585786168

Corrector Iteration

| x | y | ea |
|---|---|---|

```
2.5         0.569067395       2.94E+00
2.5         0.569963043       1.57E-01
2.5         0.569915062       8.42E-03
```

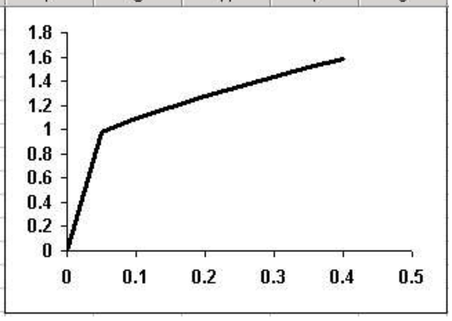
26.9

```
Option Explicit

Sub SimpImplTest()
Dim i As Integer, m As Integer
Dim xi As Single, yi As Single, xf As Single, dx As Single, xout As Single
Dim xp(200) As Single, yp(200) As Single

'Assign values
yi = 0
xi = 0
xf = 0.4
dx = 0.05
xout = 0.05

'Perform numerical Integration of ODE
Call ODESolver(xi, yi, xf, dx, xout, xp(), yp(), m)

'Display results
Sheets("Sheet1").Select
Range("a5:b205").ClearContents
Range("a5").Select
For i = 0 To m
  ActiveCell.Value = xp(i)
  ActiveCell.Offset(0, 1).Select
  ActiveCell.Value = yp(i)
  ActiveCell.Offset(1, -1).Select
Next i
Range("a5").Select
End Sub

Sub ODESolver(xi, yi, xf, dx, xout, xp, yp, m)
'Generate an array that holds the solution
Dim x As Single, y As Single, xend As Single
Dim h As Single
m = 0
xp(m) = xi
yp(m) = yi
x = xi
y = yi
Do       'Print loop
  xend = x + xout
  If (xend > xf) Then xend = xf  'Trim step if increment exceeds end
  h = dx
  Call Integrator(x, y, h, xend)
  m = m + 1
  xp(m) = x
  yp(m) = y
  If (x >= xf) Then Exit Do
Loop
End Sub

Sub Integrator(x, y, h, xend)
Dim ynew As Single
Do       'Calculation loop
  If (xend - x < h) Then h = xend - x  'Trim step if increment exceeds end
  Call SimpImpl(x, y, h, ynew)
  y = ynew
  If (x >= xend) Then Exit Do
Loop
End Sub

Sub SimpImpl(x, y, h, ynew)
'Implement Simple Implicit method
ynew = (y + h * FF(x)) / (1 + 1000 * h)
x = x + h
End Sub
```

```
Function FF(t)
'Define Forcing Function
FF = 3000 - 2000 * Exp(-t)
End Function
```

| x | y |
|---|---|
| 0 | 0 |
| 0.05 | 0.980392 |
| 0.1 | 1.095244 |
| 0.15 | 1.188461 |
| 0.2 | 1.276817 |
| 0.25 | 1.360858 |
| 0.3 | 1.4408 |
| 0.35 | 1.516843 |
| 0.4 | 1.589177 |

RUN




26.10 All linear systems are of the form

$$\frac{dy_1}{dt} = a_{11}y_1 + a_{12}y_2 + F_1$$

$$\frac{dy_2}{dt} = a_{21}y_1 + a_{22}y_2 + F_2$$

As shown in the book (p. 730), the implicit approach amounts to solving

$$\begin{bmatrix} 1 - a_{11}h & -a_{12} \\ -a_{21} & 1 - a_{22}h \end{bmatrix} \begin{Bmatrix} y_{1,i+1} \\ y_{2,i+1} \end{Bmatrix} = \begin{Bmatrix} y_{1,i} + F_1 h \\ y_{2,i} + F_2 h \end{Bmatrix}$$

Therefore, for Eq. 26.6: $a_{11} = -5$, $a_{12} = 3$, $a_{21} = 100$, $a_{22} = -301$, $F_1 =$, and $F_2 = 0$,

$$\begin{bmatrix} 1 + 5h & -3 \\ -100 & 1 + 301h \end{bmatrix} \begin{Bmatrix} y_{1,i+1} \\ y_{2,i+1} \end{Bmatrix} = \begin{Bmatrix} y_{1,i} \\ y_{2,i} \end{Bmatrix}$$

A VBA program written in these terms is

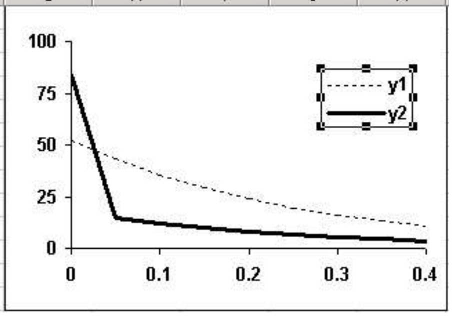
```
Option Explicit

Sub StiffSysTest()
Dim i As Integer, m As Integer, n As Integer, j As Integer
Dim xi As Single, yi(10) As Single, xf As Single, dx As Single, xout As Single
Dim xp(200) As Single, yp(200, 10) As Single

'Assign values
n = 2
xi = 0
xf = 0.4
yi(1) = 52.29
yi(2) = 83.82
dx = 0.05
xout = 0.05

'Perform numerical Integration of ODE
Call ODESolver(xi, yi(), xf, dx, xout, xp(), yp(), m, n)

'Display results
Sheets("Sheet1").Select
Range("a5:n205").ClearContents
Range("a5").Select
For i = 0 To m
```

```
  ActiveCell.Value = xp(i)
  For j = 1 To n
    ActiveCell.Offset(0, 1).Select
    ActiveCell.Value = yp(i, j)
  Next j
  ActiveCell.Offset(1, -n).Select
Next i
Range("a5").Select
End Sub

Sub ODESolver(xi, yi, xf, dx, xout, xp, yp, m, n)
'Generate an array that holds the solution
Dim i As Integer
Dim x As Single, y(10) As Single, xend As Single
Dim h As Single
m = 0
x = xi
'set initial conditions
For i = 1 To n
  y(i) = yi(i)
Next i
'save output values
xp(m) = x
For i = 1 To n
  yp(m, i) = y(i)
Next i
Do        'Print loop
  xend = x + xout
  If (xend > xf) Then xend = xf  'Trim step if increment exceeds end
  h = dx
  Call Integrator(x, y(), h, n, xend)
  m = m + 1
  'save output values
  xp(m) = x
  For i = 1 To n
    yp(m, i) = y(i)
  Next i
  If (x >= xf) Then Exit Do
Loop
End Sub

Sub Integrator(x, y, h, n, xend)
Dim j As Integer
Dim ynew(10) As Single
Do        'Calculation loop
  If (xend - x < h) Then h = xend - x  'Trim step if increment exceeds end
  Call StiffSys(x, y, h, n, ynew())
  For j = 1 To n
    y(j) = ynew(j)
  Next j
  If (x >= xend) Then Exit Do
Loop
End Sub

Sub StiffSys(x, y, h, n, ynew)
Dim j As Integer
Dim FF(2) As Single, b(2, 2) As Single, c(2) As Single, den As Single
Call Force(x, FF())
'MsgBox "pause"

b(1, 1) = 1 + 5 * h
b(1, 2) = -3 * h
b(2, 1) = -100 * h
b(2, 2) = 1 + 301 * h
For j = 1 To n
  c(j) = y(j) + FF(j) * h
Next j
den = b(1, 1) * b(2, 2) - b(1, 2) * b(2, 1)
ynew(1) = (c(1) * b(2, 2) - c(2) * b(1, 2)) / den
ynew(2) = (c(2) * b(1, 1) - c(1) * b(2, 1)) / den
x = x + h
End Sub
```

```
Sub Force(t, FF)
'Define Forcing Function
FF(0) = 0
FF(1) = 0
End Sub
```

The result compares well with the analytical solution. If a smaller step size were used, the solution would improve

| | A | B | C | D | E | F |
|---|---|---|---|---|---|---|
| 1 | Example 25.10 | | | | | |
| 2 | | | | | | |
| 3 | | | | | | |
| 4 | x | y1 | y2 | | | |
| 5 | 0 | 52.29 | 83.82 | | RUN | |
| 6 | 0.05 | 43.39626 | 14.61116 | | | |
| 7 | 0.1 | 35.5619 | 11.9733 | | | |
| 8 | 0.15 | 29.1419 | 9.811755 | | | |
| 9 | 0.2 | 23.88091 | 8.040438 | | | |
| 10 | 0.25 | 19.56968 | 6.588894 | | | |
| 11 | 0.3 | 16.03671 | 5.399384 | | | |
| 12 | 0.35 | 13.14156 | 4.424619 | | | |
| 13 | 0.4 | 10.76908 | 3.625831 | | | |



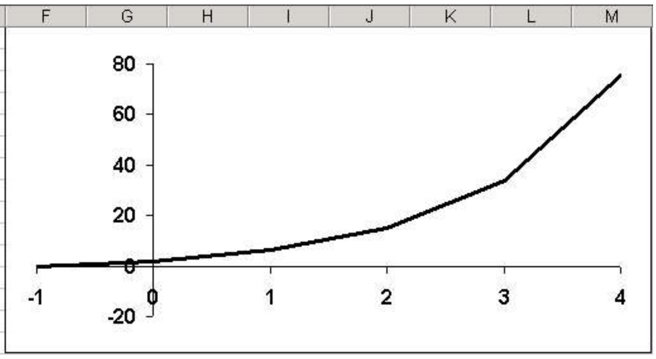

26.11 (**Errata for first printing**) Last sentence of problem statement should read: Test the program by duplicating **Example 26.4**. Later printings should have this correction.

```
Option Explicit

Sub NonSelfStartHeun()
Dim n As Integer, m As Integer, i As Integer, iter As Integer
Dim iterp(1000) As Integer
Dim xi As Single, xf As Single, yi As Single, h As Single
Dim x As Single, y As Single
Dim xp(1000) As Single, yp(1000) As Single

xi = -1
xf = 4
yi = -0.3929953
n = 5
h = (xf - xi) / n
x = xi
y = yi
m = 0
xp(m) = x
yp(m) = y

Call RK4(x, y, h)

m = m + 1
xp(m) = x
yp(m) = y
For i = 2 To n
  Call NSSHeun(xp(i - 2), yp(i - 2), xp(i - 1), yp(i - 1), x, y, h, iter)
  m = m + 1
  xp(m) = x
  yp(m) = y
  iterp(m) = iter
Next i

Sheets("NSS Heun").Select
Range("a5").Select
For i = 0 To m
  ActiveCell.Value = xp(i)
  ActiveCell.Offset(0, 1).Select
  ActiveCell.Value = yp(i)
  ActiveCell.Offset(0, 1).Select
  ActiveCell.Value = iterp(i)
  ActiveCell.Offset(1, -2).Select
Next i
```

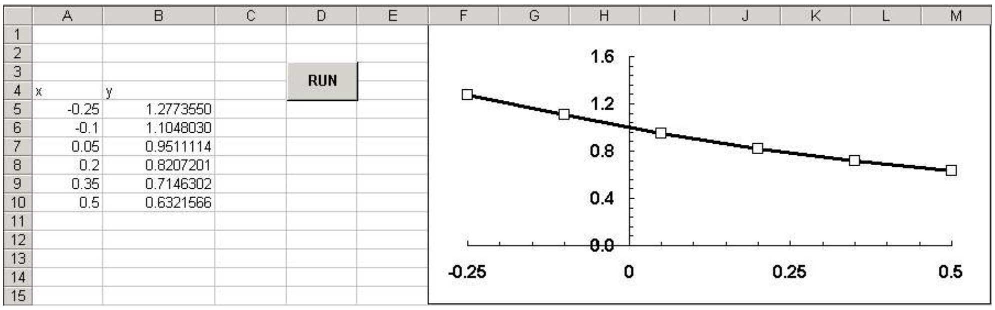
```
Range("a5").Select
End Sub

Sub RK4(x, y, h)
'Implement RK4 method
Dim k1 As Single, k2 As Single, k3 As Single, k4 As Single
Dim ym As Single, ye As Single, slope As Single
Call Derivs(x, y, k1)
ym = y + k1 * h / 2
Call Derivs(x + h / 2, ym, k2)
ym = y + k2 * h / 2
Call Derivs(x + h / 2, ym, k3)
ye = y + k3 * h
Call Derivs(x + h, ye, k4)
slope = (k1 + 2 * (k2 + k3) + k4) / 6
y = y + slope * h
x = x + h
End Sub

Sub NSSHeun(x0, y0, x1, y1, x, y, h, iter)
'Implement Non Self-Starting Heun
Dim i As Integer
Dim y2 As Single
Dim slope As Single, k1 As Single, k2 As Single
Dim ea As Single
Dim y2p As Single
Static y2old As Single, y2pold As Single
Call Derivs(x1, y1, k1)
y2 = y0 + k1 * 2 * h
y2p = y2
If iter > 0 Then
  y2 = y2 + 4 * (y2old - y2pold) / 5
End If
x = x + h
iter = 0
Do
  y2old = y2
  Call Derivs(x, y2, k2)
  slope = (k1 + k2) / 2
  y2 = y1 + slope * h
  iter = iter + 1
  ea = Abs((y2 - y2old) / y2) * 100
  If ea < 0.01 Then Exit Do
Loop
y = y2 - (y2 - y2p) / 5
y2old = y2
y2pold = y2p
End Sub

Sub Derivs(x, y, dydx)
'Define ODE
dydx = 4 * Exp(0.8 * x) - 0.5 * y
End Sub
```

| x | y |
|---|---|
| -1 | -0.3929953 |
| 0 | 2.0025680 |
| 1 | 6.2108493 |
| 2 | 14.8887939 |
| 3 | 33.7849922 |
| 4 | 75.5849609 |

RUN

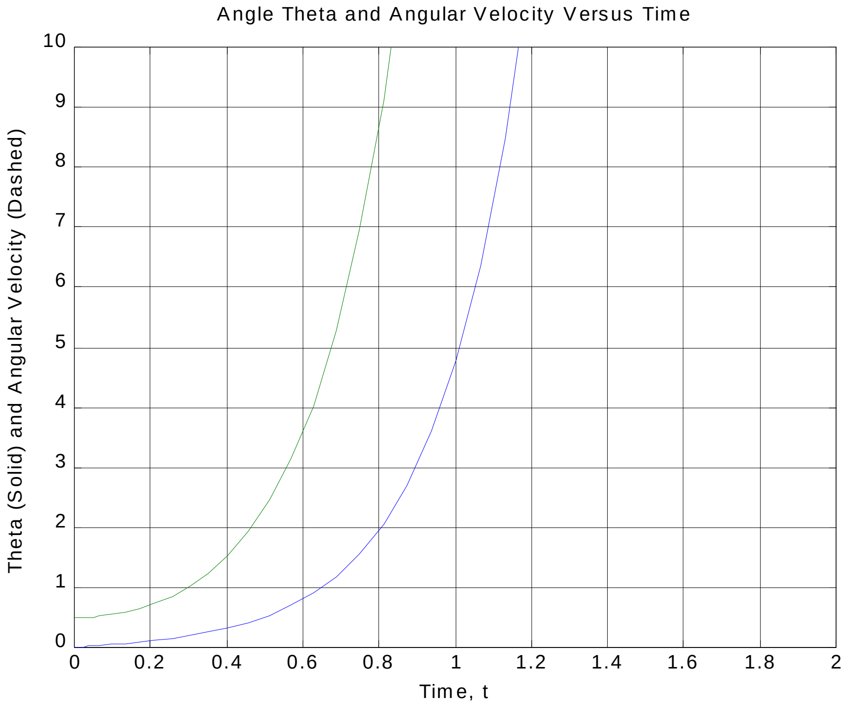


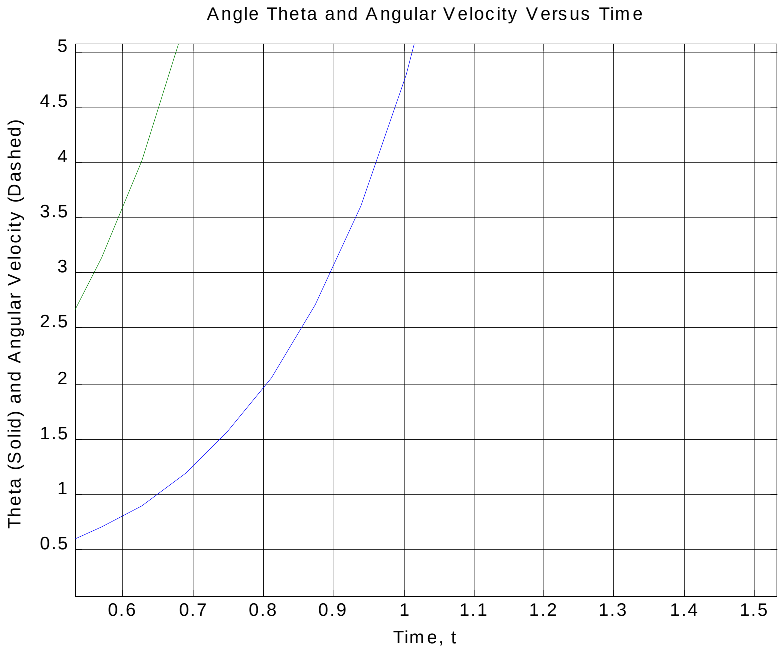

26.12




26.13 Use Matlab to solve

```
tspan=[0,5]';
x0=[0,0.5]';
[t,x]=ode45('dxdt',tspan,x0);
plot(t,x(:,1),t,x(:,2),'--')
grid
title('Angle Theta and Angular Velocity Versus Time')
xlabel('Time, t')
ylabel('Theta (Solid) and Angular Velocity (Dashed)')
axis([0 2 0 10])
zoom

function dx=dxdt(t,x)
dx=[x(2);(9.8/0.5)*x(1)];
```

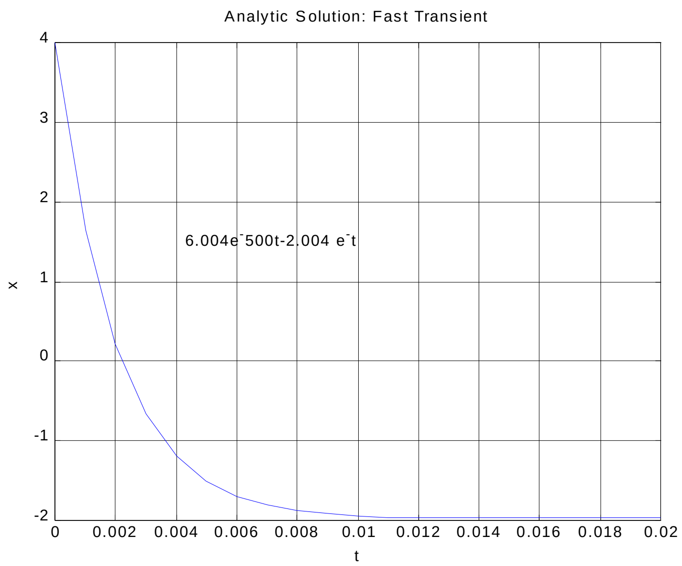


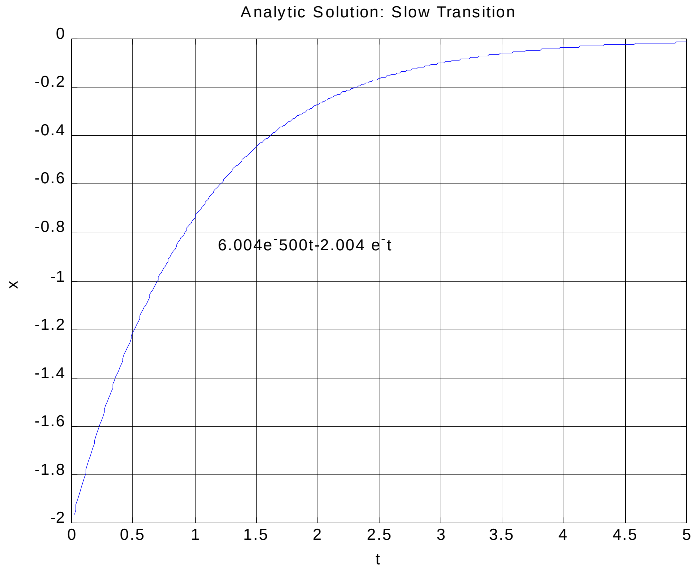



26.14 **Analytic solution:** $x = 6.004e^{-500t} - 2.004e^{-t}$

```
t=[0:.01:.02];
x=6.004*exp(-500*t)-2.004*exp(-t);
plot(t,x)
grid
xlabel('t')
ylabel('x')
title('Analytic Solution:Fast Transient')
gtext('6.004e^-500t-2.004 e^-t')
```

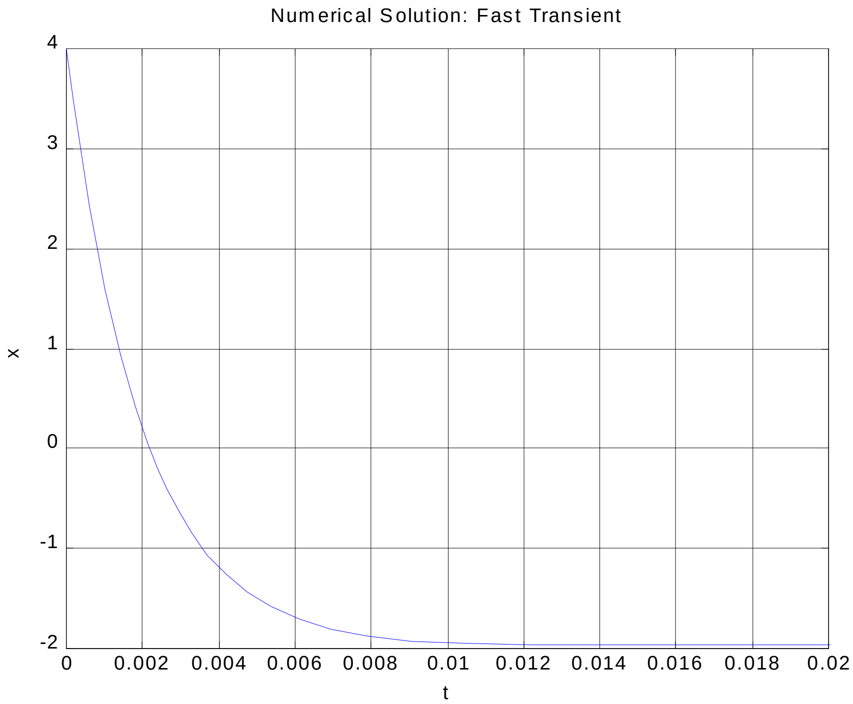
```
t=[0.02:.01:5];
x=6.004*exp(-500*t)-2.004*exp(-t);
plot(t,x)
grid
xlabel('t')
ylabel('x')
title('Analytic Solution: Slow Transition')
gtext('6.004e^-500t-2.004 e^-t')
```

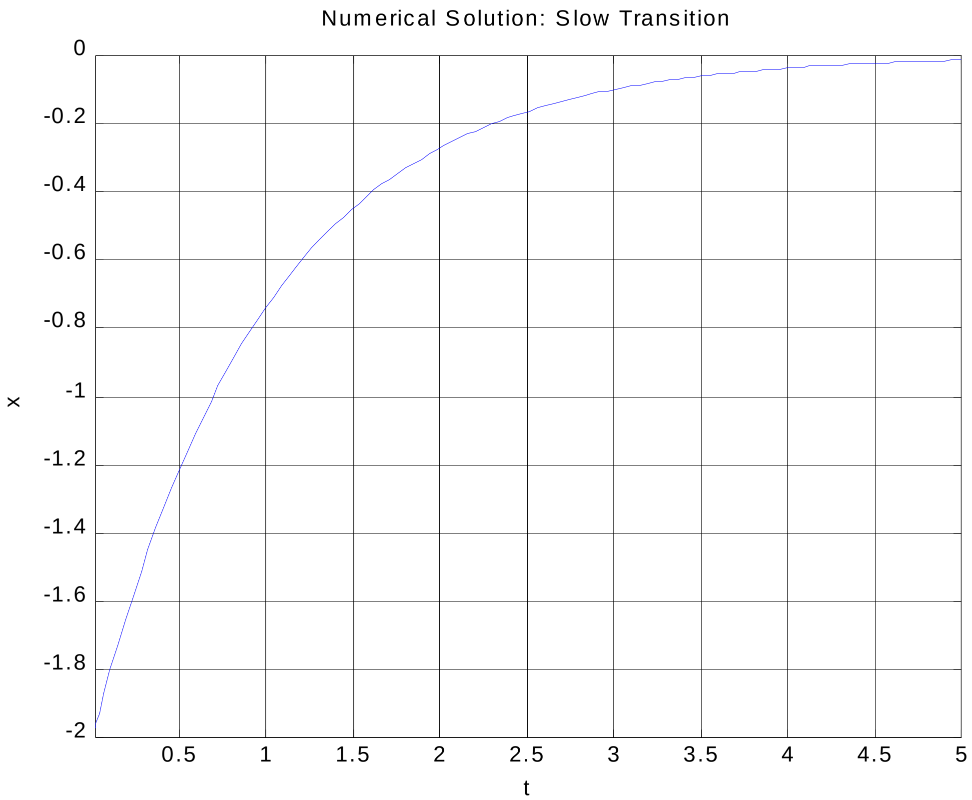



## Numerical solution:

```
tspan=[0 5];
xo=[4];
[t,x]=ode23s('eqn',tspan,xo);
plot(t,x)
grid
xlabel('t')
ylabel('x')
title('Numerical Solution: Fast Transient')
axis([0 .02 -2 4])

tspan=[0 5];
xo=[4];
[t,x]=ode23s('eqn',tspan,xo);
plot(t,x)
grid
xlabel('t')
ylabel('x')
title('Numerical Solution: Slow Transition')
axis([0.02 5 -2 0])
```


Numerical Solution: Fast Transient
x
4
3
2
1
0
-1
-2
0
0.002
0.004
0.006
0.008
0.01
0.012
0.014
0.016
0.018
0.02
t


Numerical Solution: Slow Transition
x
0
-0.2
-0.4
-0.6
-0.8
-1
-1.2
-1.4
-1.6
-1.8
-2
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5
t

## CHAPTER 27

27.1 The solution can be assumed to be $T = e^{\lambda x}$. This, along with the second derivative $T'' = \lambda^2 e^{\lambda x}$, can be substituted into the differential equation to give

$$\lambda^2 e^{\lambda x} - 0.1 e^{\lambda x} = 0$$

which can be used to solve for

$$\lambda^2 - 0.1 = 0$$
$$\lambda = \pm\sqrt{0.1}$$

Therefore, the general solution is

$$T = Ae^{\sqrt{0.1}x} + Be^{-\sqrt{0.1}x}$$

The constants can be evaluated by substituting each of the boundary conditions to generate two equations with two unknowns,

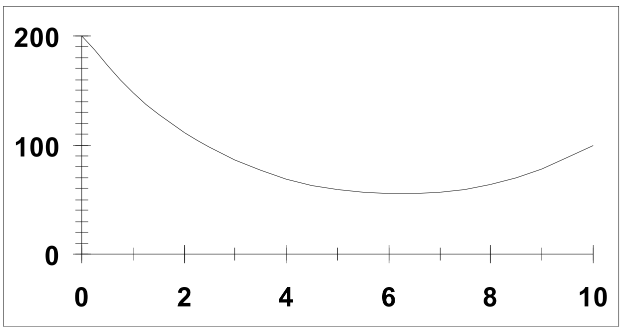
$$200 = A + B$$
$$100 = 23.62434A + 0.042329B$$

which can be solved for $A$ = 3.881524 and $B$ = 196.1185. The final solution is, therefore,

$$T = 3.881524e^{\sqrt{0.1}x} + 196.1185e^{-\sqrt{0.1}x}$$

which can be used to generate the values below:

| *x* | *T* |
|---|---|
| 0 | 200 |
| 1 | 148.2747 |
| 2 | 111.5008 |
| 3 | 85.97028 |
| 4 | 69.10864 |
| 5 | 59.21565 |
| 6 | 55.29373 |
| 7 | 56.94741 |
| 8 | 64.34346 |
| 9 | 78.22764 |
| 10 | 100 |




27.2 Reexpress the second-order equation as a pair of ODEs:

$$\frac{dT}{dx} = z$$

$$\frac{dz}{dx} = 0.1T$$

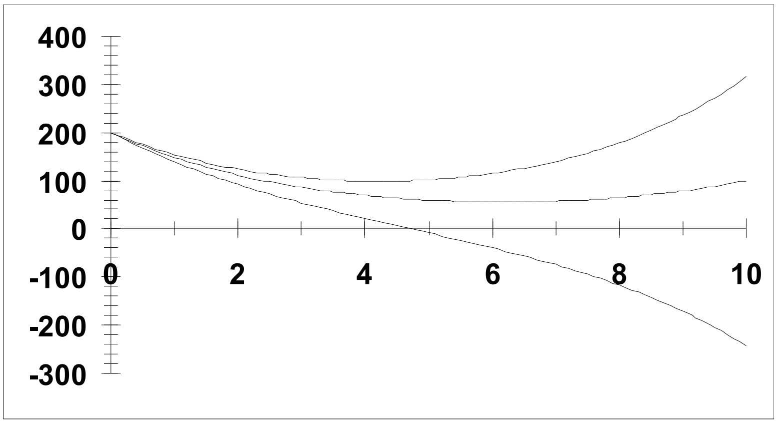
The solution was then generated on the Excel spreadsheet using the Heun method (without iteration) with a step-size of 0.01. An initial condition of $z = -55$ was chosen for the first shot. The first few calculation results are shown below.

| x | T | z | k11 | k12 | Tend | zend | k21 | k22 | phi1 | phi2 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 200.000 | -55.000 | -55.000 | 20.000 | 194.500 | -53.000 | -53.000 | 19.450 | -54.000 | 19.725 |
| 0.1 | 194.600 | -53.028 | -53.028 | 19.460 | 189.297 | -51.082 | -51.082 | 18.930 | -52.055 | 19.195 |
| 0.2 | 189.395 | -51.108 | -51.108 | 18.939 | 184.284 | -49.214 | -49.214 | 18.428 | -50.161 | 18.684 |
| 0.3 | 184.378 | -49.240 | -49.240 | 18.438 | 179.454 | -47.396 | -47.396 | 17.945 | -48.318 | 18.192 |
| 0.4 | 179.547 | -47.420 | -47.420 | 17.955 | 174.805 | -45.625 | -45.625 | 17.480 | -46.523 | 17.718 |
| 0.5 | 174.894 | -45.649 | -45.649 | 17.489 | 170.330 | -43.900 | -43.900 | 17.033 | -44.774 | 17.261 |

The resulting value at $x = 10$ was $T(10) = 315.759$. A second shot using an initial condition of z (0) = $-70$ was attempted with the result at $x = 10$ of $T(10) = -243.249$. These values can then be used to derive the correct initial condition,

$$z(0) = -55 + \frac{-70+55}{-243.249-315.759}(100-315.759) = -60.79$$

The resulting fit, along with the two "shots" are displayed below:




27.3 A centered finite difference can be substituted for the second derivative to give,

$$\frac{T_{i-1} - 2T_i + T_{i+1}}{h^2} - 0.1T_i = 0$$

or for $h = 1$,

$$-T_{i-1} + 2.1T_i - T_{i+1} = 0$$

The first node would be

$$2.1T_1 - T_2 = 200$$

and the last node would be

$$-T_9 + 2.1T_{10} = 100$$

The tridiagonal system can be solved with the Thomas algorithm or Gauss-Seidel for (the analytical solution is also included)

| *x* | *T* | Analytical |
|---|---|---|
| 0 | 200 | 200 |
| 1 | 148.4838 | 148.2747 |
| 2 | 111.816 | 111.5008 |
| 3 | 86.32978 | 85.97028 |
| 4 | 69.47655 | 69.10864 |
| 5 | 59.57097 | 59.21565 |
| 6 | 55.62249 | 55.29373 |
| 7 | 57.23625 | 56.94741 |
| 8 | 64.57365 | 64.34346 |
| 9 | 78.3684 | 78.22764 |
| 10 | 100 | 100 |

27.4 The second-order ODE can be expressed as the following pair of first-order ODEs,

$$\frac{dy}{dx} = z$$

$$\frac{dz}{dx} = \frac{2z + y - x}{8}$$

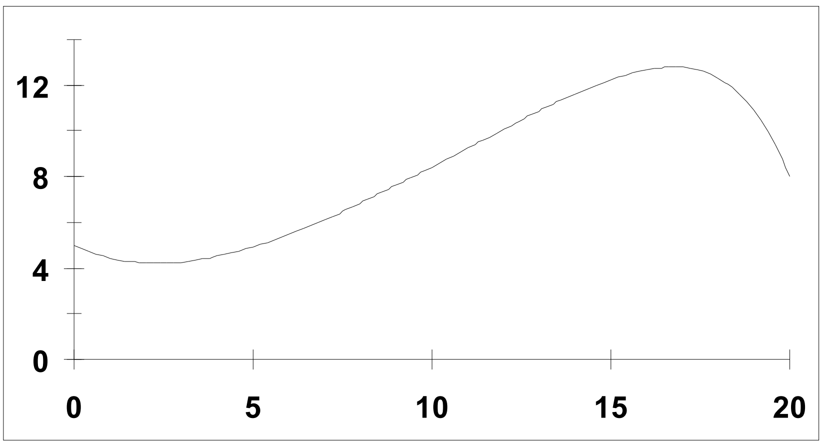
These can be solved for two guesses for the initial condition of $z$. For our cases we used

| | | |
|---|---|---|
| $z(0)$ | −1 | −0.5 |
| $y(20)$ | −6523.000507 | 7935.937904 |

Clearly, the solution is quite sensitive to the initial conditions. These values can then be used to derive the correct initial condition,

$$z(0) = -1 + \frac{-0.5 + 1}{7935.937904 + 6523.000507}(8 + 6523.000507) = -0.774154$$

The resulting fit is displayed below:




27.5 Centered finite differences can be substituted for the second and first derivatives to give,

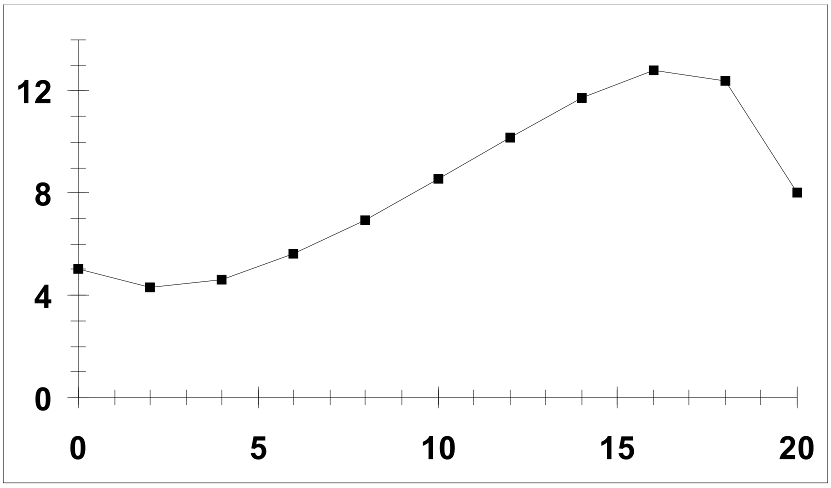
$$8\frac{y_{i+1} - 2y_i + y_{i-1}}{\Delta x^2} - 2\frac{y_{i+1} - y_{i-1}}{\Delta x} - y_i + x_i = 0$$

or substituting $\Delta x = 2$ and collecting terms yields

$$2.5y_{i+1} - 5y_i + 1.5y_{i-1} + x_i = 0$$

This equation can be written for each node and solved with either the Gauss-Seidel method or a tridiagonal solver to give

| x | T |
|---|---|
| 0 | 5 |
| 2 | 4.287065 |
| 4 | 4.623551 |
| 6 | 5.600062 |
| 8 | 6.960955 |
| 10 | 8.536414 |
| 12 | 10.18645 |
| 14 | 11.72749 |
| 16 | 12.78088 |
| 18 | 12.39044 |
| 20 | 8 |




27.6 The second-order ODE can be expressed as the following pair of first-order ODEs,

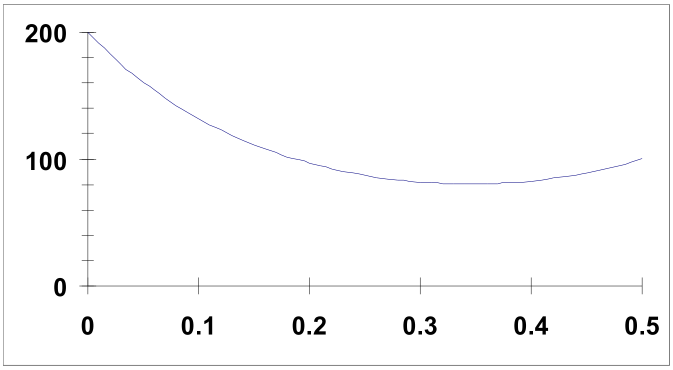
$$\frac{dT}{dx} = z$$

$$\frac{dz}{dx} = 1.2 \times 10^7 (T + 273)^4 - 5(150 - T)$$

The solution was then generated on the Excel spreadsheet using the Heun method (without iteration) with a step-size of 0.01. The Excel Solver was used to adjust the initial condition of $z$ until the value of $T(0.5) = 100$. Part of the resulting spreadsheet is shown below along with a graph of the final solution.

| $x$ | $T$ | $z$ | $k11$ | $k12$ | $Tend$ | $zend$ | $k21$ | $k22$ | $\phi 1$ | $\phi 2$ |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 200.000 | -927.673 | -927.673 | 6256.560 | 190.723 | -865.107 | -865.107 | 5752.643 | -896.390 | 6004.601 |
| 0.01 | 191.036 | -867.627 | -867.627 | 5769.196 | 182.360 | -809.935 | -809.935 | 5321.210 | -838.781 | 5545.203 |

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 0.02 | 182.648 | -812.175 | -812.175 | 5335.738 | 174.527 | -758.817 | -758.817 | 4936.083 | -785.496 | 5135.910 |
| 0.03 | 174.793 | -760.816 | -760.816 | 4948.905 | 167.185 | -711.327 | -711.327 | 4591.217 | -736.071 | 4770.061 |
| 0.04 | 167.433 | -713.115 | -713.115 | 4602.594 | 160.301 | -667.089 | -667.089 | 4281.522 | -690.102 | 4442.058 |
| 0.05 | 160.532 | -668.694 | -668.694 | 4291.667 | 153.845 | -625.778 | -625.778 | 4002.685 | -647.236 | 4147.176 |




27.7 The second-order ODE can be linearized as in

$$\frac{d^2T}{dx^2} - 1.2\times10^7(T_b+273)^4 - 4.8\times10^7(T_b+273)^3(T-T_b) + 5(150-T) = 0$$

Substituting $T_b$ = 150 and collecting terms gives

$$\frac{d^2T}{dx^2} - 41.32974T + 2357.591 = 0$$

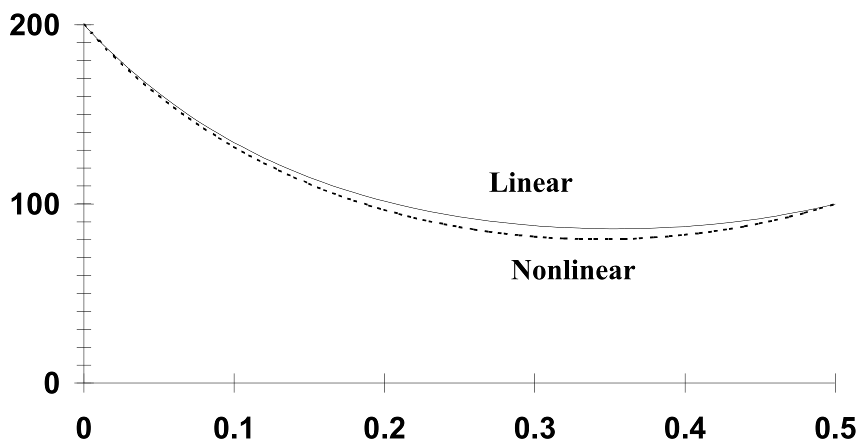
Substituting a centered-difference approximation of the second derivative gives

$$-T_{i-1} + (2 + 41.32974\Delta x^2)T_i - T_{i+1} = 2357.591\Delta x^2$$

We used the Gauss-Seidel method to solve these equations. The results for a few selected points are:

| $x$ | 0 | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 |
|---|---|---|---|---|---|---|
| $T$ | 200 | 134.2765 | 101.5758 | 87.91595 | 87.45616 | 100 |

A graph of the entire solution along with the nonlinear result from Prob. 27.7 is shown below:



27.8 For three springs

$$
\begin{aligned}
\left(\frac{2k_1}{m_1}-\omega^2\right)A_1 \quad & -\frac{k_1}{m_1}A_2 & & = 0 \\
-\frac{k_1}{m_1}A_1 \quad & +\left(\frac{2k_1}{m_1}-\omega^2\right)A_2 & -\frac{k_1}{m_1}A_3 & = 0 \\
& -\frac{k_1}{m_1}A_2 & +\left(\frac{2k_1}{m_1}-\omega^2\right)A_3 & = 0
\end{aligned}
$$

Substituting $m = 40$ kg and $k = 240$ gives

$$
\begin{aligned}
\left(12-\omega^2\right)A_1 \quad & -6A_2 & & = 0 \\
-6A_1 \quad & +\left(12-\omega^2\right)A_2 & -6A_3 & = 0 \\
& -6A_2 & +\left(12-\omega^2\right)A_3 & = 0
\end{aligned}
$$

The determinant is

$$-\omega^6 + 36\omega^4 - 360\omega^2 + 864 = 0$$

which can be solved for $\omega^2$ = 20.4853, 12, and 3.5147 $s^{-2}$. Therefore the frequencies are $\omega$ = 4.526, 3.464, and 1.875 $s^{-1}$. Substituting these values into the original equations yields for $\omega^2$ = 20.4853,

$A_1 = -0.707A_2 = A_3$

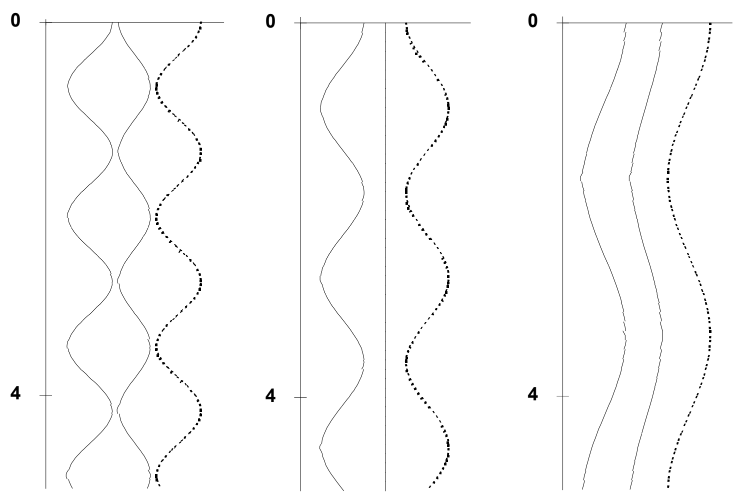
for $\omega^2 = 12$

$A_1 = -A_3$, and $= A_2 = 0$

for $\omega^2 = 3.5147$

$A_1 = 0.707A_2 = A_3$

Plots:



27.9 For 5 interior points ($h = 3/6 = 0.5$), the result is Eq. (27.19) with $2 - 0.25p^2$ on the diagonal. Dividing by 0.25 gives,

$$\begin{bmatrix} 8-p^2 & -4 & & & \\ -4 & 8-p^2 & -4 & & \\ & -4 & 8-p^2 & -4 & \\ & & -4 & 8-p^2 & -4 \\ & & & -4 & 8-p^2 \end{bmatrix} = 0$$

The determinant can be expanded (e.g., with Fadeev-Leverrier or the MATLAB **poly** function) to give

$$0 = -p^{10} + 40p^8 - 576p^6 + 3{,}584p^4 - 8960p^2 + 6{,}144$$

The roots of this polynomial can be determined as (e.g., with Bairstow's methods or the MATLAB **roots** function) $p^2 = 1.072$, 4, 8, 12, 14.94. The square root of these roots yields $p =$ 1.035, 2, 2.828, 3.464, and 3.864.

27.10 Minors:

$$(2-\lambda)\begin{vmatrix} 3-\lambda & 4 \\ 4 & 7-\lambda \end{vmatrix} - 2\begin{vmatrix} 8 & 4 \\ 10 & 7-\lambda \end{vmatrix} + 10\begin{vmatrix} 8 & 3-\lambda \\ 10 & 4 \end{vmatrix} = -\lambda^3 + 10\lambda^2 + 101\lambda + 18$$

27.11 Although the following computation can be implemented on a pocket calculator, a spreadsheet or with a program, we've used MATLAB.

```
>> a=[2 2 10;8 3 4;10 4 5]
a =
     2     2    10
     8     3     4
    10     4     5

>> x=[1 1 1]'
x =
     1
     1
     1
```

First iteration:

```
>> x=a*x
x =
    14
    15
    19

>> e=max(x)
e =
    19

>> x=x/e
x =
    0.7368
    0.7895
    1.0000
```

Second iteration:

```
>> x=a*x
x =
   13.0526
```

```
   12.2632
   15.5263

>> e=max(x)
e =
   15.5263

>> x=x/e
x =
    0.8407
    0.7898
    1.0000
```

Third iteration:

```
>> x=a*x
x =
   13.2610
   13.0949
   16.5661

>> e=max(x)
e =
   16.5661

>> x=x/e
x =
    0.8005
    0.7905
    1.0000
```

Fourth iteration:

```
>> x=a*x
x =
   13.1819
   12.7753
   16.1668

>> e=max(x)
e =
   16.1668

>> x=x/e
x =
    0.8154
    0.7902
    1.0000
```

Thus, after four iterations, the result is converging on a highest eigenvalue of 16.2741 with a corresponding eigenvector of [0.811 0.790 1].

27.12 As in Example 27.10, the computation can be laid out as

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| | 2 | 2 | 10 | | | | |
| [A] = | 8 | 3 | 4 | | | | |
| | 10 | 4 | 5 | | | | |
| | | | | | | | |
| First iteration: | | | | | | eigenvalue | eigenvector |
| -0.05556 | 1.666667 | -1.22222 | 1 | | 0.388889 | | -0.3888889 |
| 0 | -5 | 4 | 1 | = | -1 | -1 | 1 |
| 0.111111 | 0.666667 | -0.55556 | 1 | | 0.222222 | | -0.2222222 |
| | | | | | | | |
| Second iteration: | | | | | | | |
| -0.05556 | 1.666667 | -1.22222 | -0.38889 | | 1.959877 | | -0.3328092 |
| 0 | -5 | 4 | 1 | = | -5.88889 | -5.88889 | 1 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 0.111111 | 0.666667 | -0.55556 | -0.22222 | | 0.746914 | | -0.1268344 |
| | | | | | | | |
| Third iteration: | | | | | | | |
| -0.05556 | 1.666667 | -1.22222 | -0.33281 | | 1.840176 | | -0.3341317 |
| 0 | -5 | 4 | 1 | = | -5.50734 | -5.50734 | 1 |
| 0.111111 | 0.666667 | -0.55556 | -0.12683 | | 0.700151 | | -0.1271307 |
| | | | | | | | |
| Fourth iteration: | | | | | | | |
| -0.05556 | 1.666667 | -1.22222 | -0.33413 | | 1.840611 | | -0.3341389 |
| 0 | -5 | 4 | 1 | = | -5.50852 | -5.50852 | 1 |
| 0.111111 | 0.666667 | -0.55556 | -0.12713 | | 0.700169 | | -0.1271065 |

Thus, after four iterations, the estimate of the lowest eigenvalue is $1/(-5.5085) = -0.1815$ with an eigenvector of [–0.3341 1 -0.1271].

27.13 Here is VBA Code to implement the shooting method:

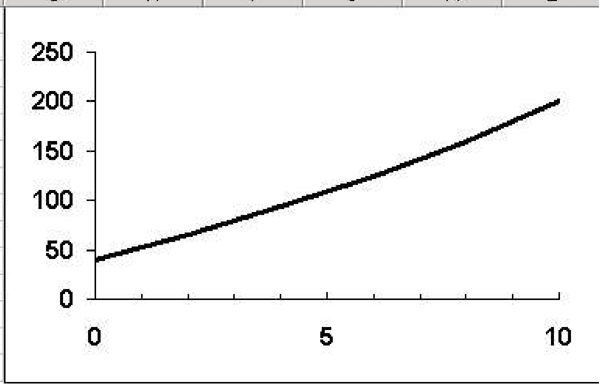
```
Public hp As Single, Ta As Single

Option Explicit

Sub Shoot()

Dim n As Integer, m As Integer, i As Integer, j As Integer
Dim x0 As Single, xf As Single
Dim x As Single, y(2) As Single, h As Single, dx As Single, xend As Single
Dim xp(200) As Single, yp(2, 200) As Single, xout As Single
Dim z01 As Single, z02 As Single, T01 As Single, T02 As Single
Dim T0 As Single, Tf As Single
Dim Tf1 As Single, Tf2 As Single

'set parameters
n = 2
hp = 0.01
Ta = 20
x0 = 0
T0 = 40
xf = 10
Tf = 200
dx = 2
xend = xf
xout = 2
'first shot
x = x0
y(1) = T0
y(2) = 10
Call RKsystems(x, y, n, dx, xf, xout, xp(), yp(), m)
z01 = yp(2, 0)
Tf1 = yp(1, m)
'second shot
x = x0
y(1) = T0
y(2) = 20
Call RKsystems(x, y, n, dx, xf, xout, xp(), yp(), m)
z02 = yp(2, 0)
Tf2 = yp(1, m)
'last shot
x = x0
y(1) = T0
'linear interpolation
y(2) = z01 + (z02 - z01) / (Tf2 - Tf1) * (Tf - Tf1)
Call RKsystems(x, y, n, dx, xf, xout, xp(), yp(), m)
'output results
Range("a4:C1004").ClearContents
Range("A4").Select
For j = 0 To m
  ActiveCell.Value = xp(j)
```

```
  For i = 1 To n
    ActiveCell.Offset(0, 1).Select
    ActiveCell.Value = yp(i, j)
  Next i
  ActiveCell.Offset(1, -n).Select
Next j
Range("A4").Select
End Sub

Sub RKsystems(x, y, n, dx, xf, xout, xp, yp, m)
Dim i As Integer
Dim xend As Single, h As Single

m = 0
For i = 1 To n
  yp(i, m) = y(i)
Next i
Do
  xend = x + xout
  If xend > xf Then xend = xf
  h = dx
  Do
    If xend - x < h Then h = xend - x
    Call RK4(x, y, n, h)
    If x >= xend Then Exit Do
  Loop
  m = m + 1
  xp(m) = x
  For i = 1 To n
    yp(i, m) = y(i)
  Next i
  If x >= xf Then Exit Do
Loop
End Sub

Sub RK4(x, y, n, h)

Dim i
Dim ynew, dydx(10), ym(10), ye(10)
Dim k1(10), k2(10), k3(10), k4(10)
Dim slope(10)
Call Derivs(x, y, k1)
For i = 1 To n
  ym(i) = y(i) + k1(i) * h / 2
Next i
Call Derivs(x + h / 2, ym, k2)
For i = 1 To n
  ym(i) = y(i) + k2(i) * h / 2
Next i
Call Derivs(x + h / 2, ym, k3)
For i = 1 To n
  ye(i) = y(i) + k3(i) * h
Next i
Call Derivs(x + h, ye, k4)
For i = 1 To n
  slope(i) = (k1(i) + 2 * (k2(i) + k3(i)) + k4(i)) / 6
Next i
For i = 1 To n
  y(i) = y(i) + slope(i) * h
Next i
x = x + h
End Sub

Sub Derivs(x, y, dydx)

dydx(1) = y(2)
dydx(2) = hp * (y(1) - Ta)
End Sub
```

Prob. 27.13

| t | T | z |
|---|---|---|
| 0 | 40 | 12.69067 |
| 2 | 65.95189 | 13.348 |
| 4 | 93.74796 | 14.54101 |
| 6 | 124.5037 | 16.3176 |
| 8 | 159.4536 | 18.74904 |
| 10 | 200 | 21.93294 |

RUN

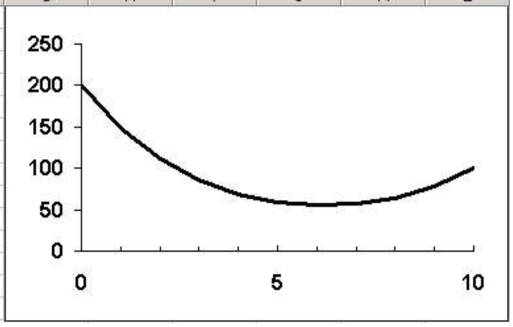



27.14

Prob. 27.2

| t | T | z |
|---|---|---|
| 0 | 200 | -60.7906 |
| 1 | 148.275 | -43.5206 |
| 2 | 111.5014 | -30.639 |
| 3 | 85.97091 | -20.847 |
| 4 | 69.10934 | -13.157 |
| 5 | 59.2164 | -6.79365 |
| 6 | 55.29456 | -1.11537 |
| 7 | 56.94834 | 4.450443 |
| 8 | 64.34451 | 10.46502 |
| 9 | 78.22887 | 17.53484 |
| 10 | 100.0015 | 26.37279 |

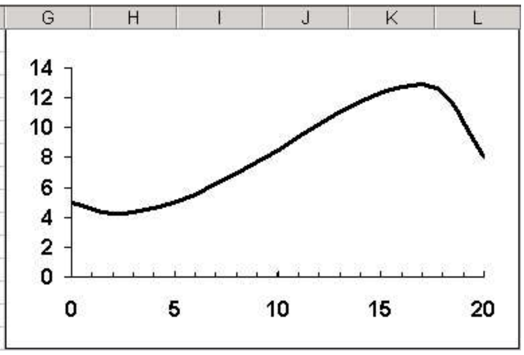
RUN




Prob. 27.4

| t | T | z |
|---|---|---|
| 0 | 5 | -0.75034 |
| 2 | 4.244752 | -0.06212 |
| 4 | 4.571954 | 0.354484 |
| 6 | 5.552851 | 0.604916 |
| 8 | 6.922503 | 0.750695 |
| 10 | 8.506933 | 0.822492 |
| 12 | 10.16451 | 0.820854 |
| 14 | 11.71116 | 0.697035 |
| 16 | 12.76845 | 0.288061 |
| 18 | 12.38154 | -0.86756 |
| 20 | 7.999797 | -4.03548 |

RUN




27.15 A general formulation that describes Example 27.3 as well as Probs. 27.3 and 27.5 is

$$a\frac{d^2y}{dx^2} + b\frac{dy}{dx} + cy + f(x) = 0$$

Finite difference approximations can be substituted for the derivatives:

$$a\frac{y_{i+1} - 2y_i + y_{i-1}}{\Delta x^2} + b\frac{y_{i+1} - y_{i-1}}{2\Delta x} + cy_i + f(x_i) = 0$$

Collecting terms

$$-(a - 0.5b\Delta x)y_{i-1} + (2a + c\Delta x^2)y_i - (a + 0.5b\Delta x)y_{i+1} = f(x_i)\Delta x^2$$

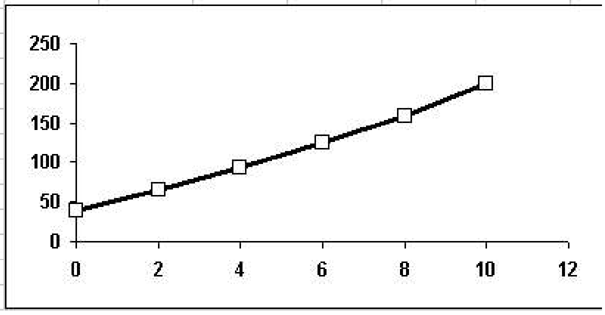
The following VBA code implants this equation as applied to Example 27.3.

```
Public hp As Single, dx As Single
Option Explicit
Sub FDBoundaryValue()
Dim ns As Integer, i As Integer
Dim a As Single, b As Single, c As Single
Dim e(100) As Single, f(100) As Single, g(100) As Single
Dim r(100) As Single, y(100) As Single
Dim Lx As Single, xx As Single, x(100) As Single
Lx = 10
dx = 2
ns = Lx / dx
xx = 0
hp = 0.01
a = 1
b = 0
c = hp
y(0) = 40
y(ns) = 200
For i = 0 To ns
  x(i) = xx
  xx = xx + dx
Next i
f(1) = 2 * a / dx ^ 2 + c
g(1) = -(a / dx ^ 2 - b / (2 * dx))
r(1) = ff(x(1)) + (a / dx ^ 2 + b / (2 * dx)) * y(0)
For i = 2 To ns - 2
  e(i) = -(a / dx ^ 2 + b / (2 * dx))
  f(i) = 2 * a / dx ^ 2 + c
  g(i) = -(a / dx ^ 2 - b / (2 * dx))
  r(i) = ff(x(i))
Next i
e(ns - 1) = -(a / dx ^ 2 + b / (2 * dx))
f(ns - 1) = 2 * a / dx ^ 2 + c
r(ns - 1) = ff(x(ns - 1)) + (a / dx ^ 2 - b / (2 * dx)) * y(ns)
Sheets("Sheet2").Select
Range("a5:d105").ClearContents
Range("a5").Select
For i = 1 To ns - 1
  ActiveCell.Value = e(i)
  ActiveCell.Offset(0, 1).Select
  ActiveCell.Value = f(i)
  ActiveCell.Offset(0, 1).Select
  ActiveCell.Value = g(i)
  ActiveCell.Offset(0, 1).Select
  ActiveCell.Value = r(i)
  ActiveCell.Offset(1, -3).Select
Next i
Range("a5").Select
Call Tridiag(e, f, g, r, ns - 1, y)
Sheets("Sheet1").Select
Range("a5:b105").ClearContents
Range("a5").Select
For i = 0 To ns
  ActiveCell.Value = x(i)
  ActiveCell.Offset(0, 1).Select
  ActiveCell.Value = y(i)
  ActiveCell.Offset(1, -1).Select
Next i
Range("a5").Select
End Sub

Sub Tridiag(e, f, g, r, n, x)
Dim k As Integer
For k = 2 To n
  e(k) = e(k) / f(k - 1)
  f(k) = f(k) - e(k) * g(k - 1)
Next k
For k = 2 To n
  r(k) = r(k) - e(k) * r(k - 1)
Next k
x(n) = r(n) / f(n)
```

```
For k = n - 1 To 1 Step -1
  x(k) = (r(k) - g(k) * x(k + 1)) / f(k)
Next k
End Sub

Function ff(x)
ff = hp * 20
End Function
```

| | A | B |
|---|---|---|
| 1 | Example 27.3 | |
| 2 | | |
| 3 | | |
| 4 | x | y |
| 5 | 0 | 40 |
| 6 | 2 | 65.96983 |
| 7 | 4 | 93.77847 |
| 8 | 6 | 124.5382 |
| 9 | 8 | 159.4795 |
| 10 | 10 | 200 |
| 11 | | |
| 12 | | |
| 13 | | |
| 14 | | |
| 15 | | |

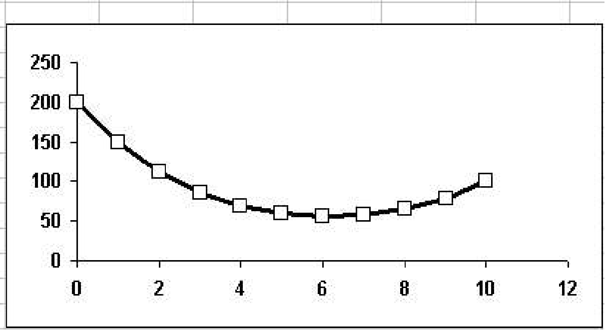
RUN




27.16

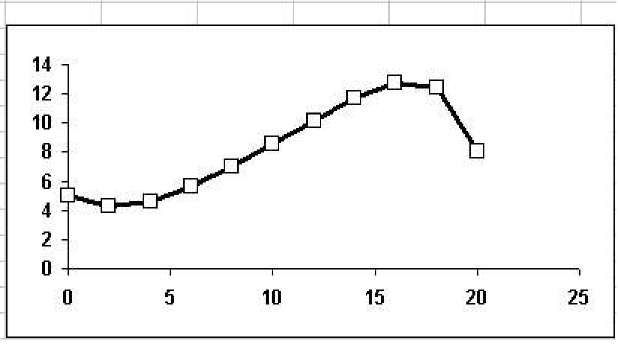
| | A | B |
|---|---|---|
| 1 | Prob. 27.3 | |
| 2 | | |
| 3 | | |
| 4 | x | y |
| 5 | 0 | 200 |
| 6 | 1 | 148.4838 |
| 7 | 2 | 111.816 |
| 8 | 3 | 86.32983 |
| 9 | 4 | 69.47659 |
| 10 | 5 | 59.57101 |
| 11 | 6 | 55.62252 |
| 12 | 7 | 57.23628 |
| 13 | 8 | 64.57366 |
| 14 | 9 | 78.36842 |
| 15 | 10 | 100 |

RUN




| | A | B |
|---|---|---|
| 1 | Prob. 27.5 | |
| 2 | | |
| 3 | | |
| 4 | x | y |
| 5 | 0 | 5 |
| 6 | 2 | 4.287066 |
| 7 | 4 | 4.623551 |
| 8 | 6 | 5.600062 |
| 9 | 8 | 6.960955 |
| 10 | 10 | 8.536414 |
| 11 | 12 | 10.18645 |
| 12 | 14 | 11.72749 |
| 13 | 16 | 12.78088 |
| 14 | 18 | 12.39044 |
| 15 | 20 | 8 |

RUN




27.17

```
Option Explicit

Sub Power()
Dim n As Integer, i As Integer, iter As Integer
Dim aa As Single, bb As Single
Dim a(10, 10) As Single, c(10) As Single
Dim lam As Single, lamold As Single, v(10) As Single
Dim es As Single, ea As Single
```

```
es = 0.001
n = 3
aa = 2 / 0.5625
bb = -1 / 0.5625
a(1, 1) = aa
a(1, 2) = bb
For i = 2 To n - 1
  a(i, i - 1) = bb
  a(i, i) = aa
  a(i, i + 1) = bb
Next i
a(i, i - 1) = bb
a(i, i) = aa

lam = 1
For i = 1 To n
  v(i) = lam
Next i

Sheets("sheet1").Select
Range("a3:b1000").ClearContents
Range("a3").Select
Do
  iter = iter + 1
  Call Mmult(a(), (v()), v(), n, n, 1)
  lam = Abs(v(1))
  For i = 2 To n
    If Abs(v(i)) > lam Then lam = Abs(v(i))
  Next i
  ActiveCell.Value = "iteration: "
  ActiveCell.Offset(0, 1).Select
  ActiveCell.Value = iter
  ActiveCell.Offset(1, -1).Select
  ActiveCell.Value = "eigenvalue: "
  ActiveCell.Offset(0, 1).Select
  ActiveCell.Value = lam
  ActiveCell.Offset(1, -1).Select
  For i = 1 To n
    v(i) = v(i) / lam
  Next i
  ActiveCell.Value = "eigenvector:"
  ActiveCell.Offset(0, 1).Select
  For i = 1 To n
    ActiveCell.Value = v(i)
    ActiveCell.Offset(1, 0).Select
  Next i
  ActiveCell.Offset(1, -1).Select
  ea = Abs((lam - lamold) / lam) * 100
  lamold = lam
  If ea <= es Then Exit Do
Loop

End Sub

Sub Mmult(a, b, c, m, n, l)

Dim i As Integer, j As Integer, k As Integer
Dim sum As Single

For i = 1 To n
  sum = 0
  For k = 1 To m
    sum = sum + a(i, k) * b(k)
  Next k
  c(i) = sum
Next i

End Sub
```

| | A | B | C | D | E |
|---|---|---|---|---|---|
| 1 | Example 27.7 | | | | |
| 2 | | | | | |
| 3 | iteration: | 1 | | RUN | |
| 4 | eigenvalue: | 1.777778 | | | |
| 5 | eigenvector: | 1 | | | |
| 6 | | 0 | | | |
| 7 | | 1 | | | |
| 8 | | | | | |
| 9 | iteration: | 2 | | | |
| 10 | eigenvalue: | 3.555556 | | | |
| 11 | eigenvector: | 1 | | | |
| 12 | | -1 | | | |
| 13 | | 1 | | | |
| 14 | | | | | |
| 15 | iteration: | 3 | | | |
| 16 | eigenvalue: | 7.111111 | | | |
| 17 | eigenvector: | 0.75 | | | |
| 18 | | -1 | | | |
| 19 | | 0.75 | | | |

•
•
•

| | | | | | |
|---|---|---|---|---|---|
| 57 | iteration: | 10 | | | |
| 58 | eigenvalue: | 6.069717 | | | |
| 59 | eigenvector: | 0.707107 | | | |
| 60 | | -1 | | | |
| 61 | | 0.707107 | | | |

27.18

```
Option Explicit

Sub Power()
Dim n As Integer, i As Integer, iter As Integer, j As Integer
Dim aa As Single, bb As Single
Dim a(10, 10) As Single, c(10) As Single
Dim lam As Single, lamold As Single, v(10) As Single
Dim es As Single, ea As Single
Dim x(10) As Single, ai(10, 10) As Single

es = 0.0000011
n = 3
aa = 2 / 0.5625
bb = -1 / 0.5625
a(1, 1) = aa
a(1, 2) = bb
For i = 2 To n - 1
  a(i, i - 1) = bb
  a(i, i) = aa
  a(i, i + 1) = bb
Next i
a(i, i - 1) = bb
a(i, i) = aa

Call LUDminv(a(), n, x())

lam = 1
For i = 1 To n
  v(i) = lam
Next i
```

```
Sheets("sheet1").Select
Range("a3:j1000").ClearContents
Range("a3").Select
ActiveCell.Value = "Matrix inverse:"
ActiveCell.Offset(1, 0).Select
For i = 1 To n
  For j = 1 To n
    ActiveCell.Value = a(i, j)
    ActiveCell.Offset(0, 1).Select
  Next j
  ActiveCell.Offset(1, -n).Select
Next i
ActiveCell.Offset(1, 0).Select

Do
  iter = iter + 1
  Call Mmult(a(), (v()), v(), n, n, 1)
  lam = Abs(v(1))
  For i = 2 To n
    If Abs(v(i)) > lam Then lam = Abs(v(i))
  Next i
  ActiveCell.Value = "iteration: "
  ActiveCell.Offset(0, 1).Select
  ActiveCell.Value = iter
  ActiveCell.Offset(1, -1).Select
  ActiveCell.Value = "eigenvalue: "
  ActiveCell.Offset(0, 1).Select
  ActiveCell.Value = lam
  ActiveCell.Offset(1, -1).Select
  For i = 1 To n
    v(i) = v(i) / lam
  Next i
  ActiveCell.Value = "eigenvector:"
  ActiveCell.Offset(0, 1).Select
  For i = 1 To n
    ActiveCell.Value = v(i)
    ActiveCell.Offset(1, 0).Select
  Next i
  ActiveCell.Offset(1, -1).Select
  ea = Abs((lam - lamold) / lam) * 100
  lamold = lam
  If ea <= es Then Exit Do
Loop

End Sub

Sub Mmult(a, b, c, m, n, l)

Dim i As Integer, j As Integer, k As Integer
Dim sum As Single

For i = 1 To n
  sum = 0
  For k = 1 To m
    sum = sum + a(i, k) * b(k)
  Next k
  c(i) = sum
Next i

End Sub

Sub LUDminv(a, n, x)
Dim i As Integer, j As Integer, er As Integer
Dim o(3) As Single, s(3) As Single, b(3) As Single
Dim ai(10, 10) As Single, tol As Single

tol = 0.00001
Call Decompose(a, n, tol, o(), s(), er)
If er = 0 Then
  For i = 1 To n
    For j = 1 To n
      If i = j Then
```

```
        b(j) = 1
      Else
        b(j) = 0
      End If
    Next j
    Call Substitute(a, o, n, b, x)
    For j = 1 To n
      ai(j, i) = x(j)
    Next j
  Next i
End If
For i = 1 To n
  For j = 1 To n
    a(i, j) = ai(i, j)
  Next j
Next i
End Sub

Sub Decompose(a, n, tol, o, s, er)
Dim i As Integer, j As Integer, k As Integer
Dim factor As Single
For i = 1 To n
  o(i) = i
  s(i) = Abs(a(i, 1))
  For j = 2 To n
    If Abs(a(i, j)) > s(i) Then s(i) = Abs(a(i, j))
  Next j
Next i
For k = 1 To n - 1
  Call Pivot(a, o, s, n, k)
  If Abs(a(o(k), k) / s(o(k))) < tol Then
    er = -1
    Exit For
  End If
  For i = k + 1 To n
    factor = a(o(i), k) / a(o(k), k)
    a(o(i), k) = factor
    For j = k + 1 To n
      a(o(i), j) = a(o(i), j) - factor * a(o(k), j)
    Next j
  Next i
Next k
If (Abs(a(o(k), k) / s(o(k))) < tol) Then er = -1
End Sub

Sub Pivot(a, o, s, n, k)
Dim ii As Integer, p As Integer
Dim big As Single, dummy As Single
p = k
big = Abs(a(o(k), k) / s(o(k)))
For ii = k + 1 To n
  dummy = Abs(a(o(ii), k) / s(o(ii)))
  If dummy > big Then
    big = dummy
    p = ii
  End If
Next ii
dummy = o(p)
o(p) = o(k)
o(k) = dummy
End Sub

Sub Substitute(a, o, n, b, x)
Dim k As Integer, i As Integer, j As Integer
Dim sum As Single, factor As Single
For k = 1 To n - 1
  For i = k + 1 To n
    factor = a(o(i), k)
    b(o(i)) = b(o(i)) - factor * b(o(k))
  Next i
Next k
x(n) = b(o(n)) / a(o(n), n)
```

```
For i = n - 1 To 1 Step -1
  sum = 0
  For j = i + 1 To n
    sum = sum + a(o(i), j) * x(j)
  Next j
  x(i) = (b(o(i)) - sum) / a(o(i), i)
Next i
End Sub
```

| | A | B | C | D | E | F |
|---|---|---|---|---|---|---|
| 1 | Example 27.8 | | | | | |
| 2 | | | | | | |
| 3 | Matrix inverse: | | | | RUN | |
| 4 | 0.421875 | 0.28125 | 0.140625 | | | |
| 5 | 0.28125 | 0.5625 | 0.28125 | | | |
| 6 | 0.140624985 | 0.28125 | 0.421875 | | | |
| 7 | | | | | | |
| 8 | iteration: | 1.0000 | | | | |
| 9 | eigenvalue: | 1.125 | | | | |
| 10 | eigenvector: | 0.75 | | | | |
| 11 | | 1 | | | | |
| 12 | | 0.75 | | | | |
| 13 | | | | | | |
| 14 | iteration: | 2 | | | | |
| 15 | eigenvalue: | 0.984375 | | | | |
| 16 | eigenvector: | 0.714286 | | | | |
| 17 | | 1 | | | | |
| 18 | | 0.714286 | | | | |

•

•

•

| | | | | | | |
|---|---|---|---|---|---|---|
| 50 | iteration: | 8 | | | | |
| 51 | eigenvalue: | 0.960248 | | | | |
| 52 | eigenvector: | 0.707107 | | | | |
| 53 | | 1 | | | | |
| 54 | | 0.707107 | | | | |

27.19 This problem can be solved by recognizing that the solution corresponds to driving the differential equation to zero. To do this, a finite difference approximation can be substituted for the second derivative to give

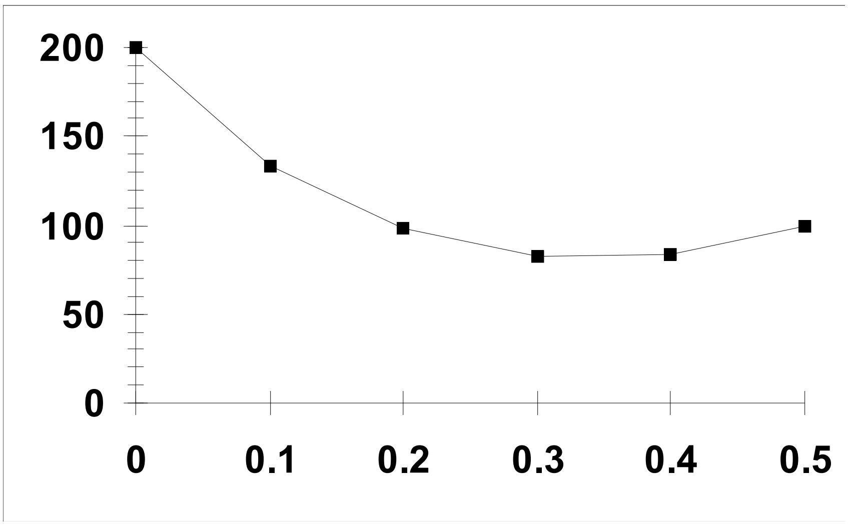
$$R = \frac{T_{i-1} - 2T_i + T_{i+1}}{(\Delta x)^2} - 1.2 \times 10^{-7}(T_i + 273)^4 + 5(150 - T_i)$$

where $R$ = the residual, which is equal to zero when the equation is satisfied. Next, a spreadsheet can be set up as below. Guesses for $T$ can be entered in cells B11:B14. Then, the residual equation can be written in cells C11:C14 and referenced to the temperatures in column B. The square of the $R$'s can then be entered in column D and summed (D17). The Solver can then be invoked to drive cell D17 to zero by varying B11:B14. The result is as shown in the spreadsheet. A plot is also displayed below.

| | A | B | C | D |
|---|---|---|---|---|
| 1 | E | 1 | | |
| 2 | sigma | 1.20E-07 | | |
| 3 | k | 5 | | |
| 4 | Ta | 150 | | |
| 5 | T0 | 200 | | |
| 6 | Tn | 100 | | |
| 7 | dx | 0.1 | | |
| 8 | | | | |
| 9 | x | T | R | R^2 |
| 10 | 0 | 200 | | |
| 11 | 0.1 | 133.015 | **4.32E-05** | 1.87E-09 |
| 12 | 0.2 | 97.79076 | 0.000185 | 3.42E-08 |
| 13 | 0.3 | 82.63883 | -0.00076 | 5.8E-07 |
| 14 | 0.4 | 83.31515 | 0.001114 | 1.24E-06 |
| 15 | 0.5 | 100 | | |
| 16 | | | | |
| 17 | | | SSR | **1.86E-06** |

**=sum(D11:D14)**

**=(B10-2*B11+B12)/$B$7^2-$B$2*(B11+273)^4+$B$3*($B$4-B11)**



27.20 First, an m-file containing the system of ODEs can be created and saved (in this case as odesys.m),

```
function dy = predprey(t,y)
dy=[0.3*y(1)-1.5*y(1)*y(2);-0.1*y(2)+0.036*y(1)*y(2)];
```

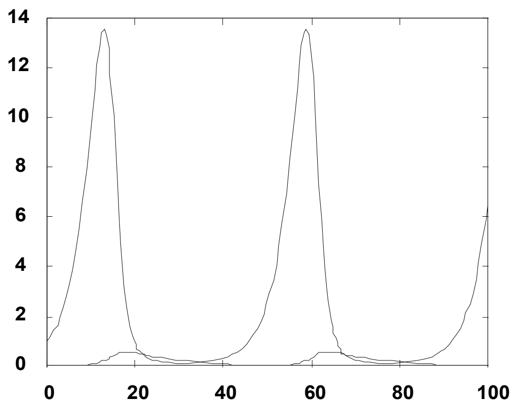
Then, the following MATLAB session is used to generate the solution:

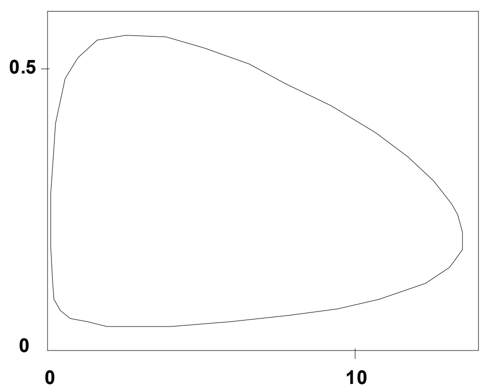
```
>> [t,y]=ode45('odesys',[0 100],[1;.05]);
```

A plot of the solution along with the state-space plot are generated with

```
>> plot(t,y)
>> plot(y(:,1),y(:,2))
```

These plots are displayed below




27.21 First, the 2nd-order ODE can be reexpressed as the following system of 1st-order ODE's

$$\frac{dx}{dt} = z$$
$$\frac{dz}{dt} = -8.333333z - 1166.667x$$

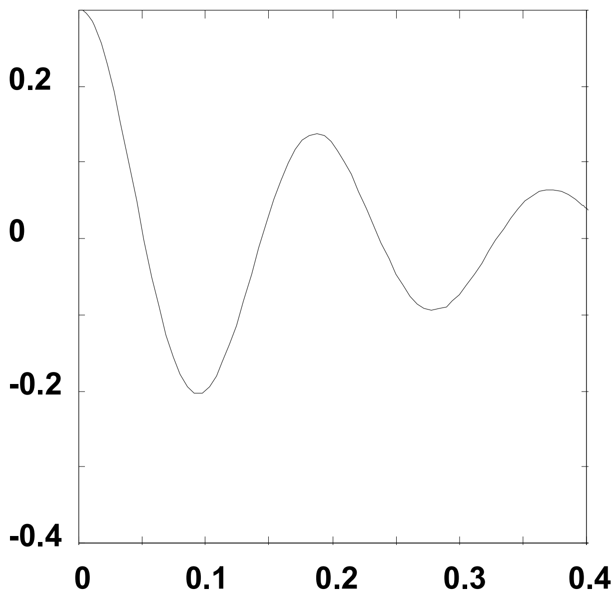
Next, we create an m-file to hold the ODEs:

```
function dx=spring(t,y)
dx=[y(2);-8.333333*y(2)-1166.667*y(1)]
```

Then we enter the following commands into MATLAB

```
[t,y]=ode45('spring',[0 .4],[0.3;0])
plot(t,y(:,1));
```

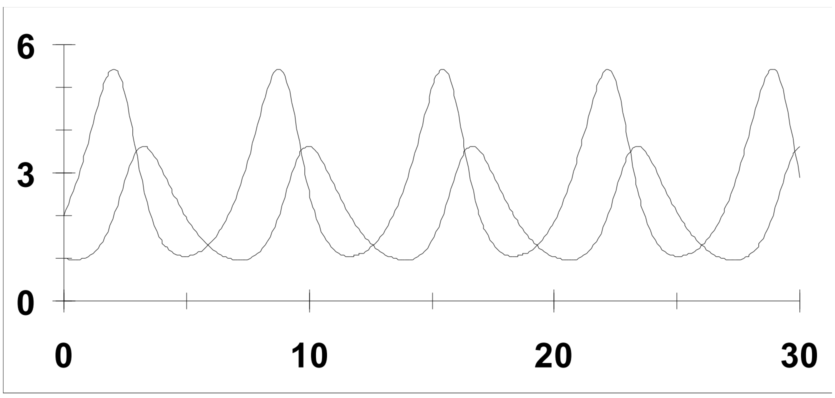
The following plot results:




(*b*) The eigenvalues and eigenvectors can be determined with the following commands:

```
>> a=[0 -1;8.333333 1166.667];
>> format short e
>> [v,d]=eig(a)

v =

 -9.9997e-001  8.5715e-004
  7.1427e-003 -1.0000e+000


d =

  7.1429e-003            0
            0  1.1667e+003
```

27.22 This problem is solved in an identical fashion to that employed in Example 27.12. For part (*a*), the solution is as displayed in the following plot:




(*b*) The solution for this set of equations is laid out in Sec. 28.2 (Fig. 28.9).

27.23

Boundary Value Problem

1. x-spacing

at x=0, i=1 ; and at x=2, i=n

$$\Delta x = \frac{2-0}{n-1}$$

2. Finite Difference Equation

$$\frac{d^2u}{dx^2} + 6\frac{du}{dx} - u = 2$$

Substitute in finite difference approximations

$$\frac{u_{i+1} - u_i + u_{i-1}}{\Delta x^2} + 6\frac{u_{i+1} - u_{i-1}}{2\Delta x} - u_i = 2$$

$$[1 - 3(\Delta x)]u_{i-1} + [-2 - \Delta x^2]u_i + [1 + 3\Delta x]u_{i+1} = 2\Delta x^2$$

Coefficients

$$a_i = 1 - 3(\Delta x) \qquad b_i = -2 - \Delta x^2 \qquad c_i = 1 + 3\Delta x \qquad d_i = 2\Delta x^2$$

3. End point equations

i=2 $\quad [1 - 3(\Delta x)]10 + [-2 - \Delta x^2]u_2 + [1 + 3\Delta x]u_3 = 2\Delta x^2$

Coefficients

$$a_2 = 0 \qquad b_2 = -2 - \Delta x^2 \qquad c_2 = 1 + 3\Delta x \qquad d_2 = 2\Delta x^2 - 10(1 - 3(\Delta x))$$

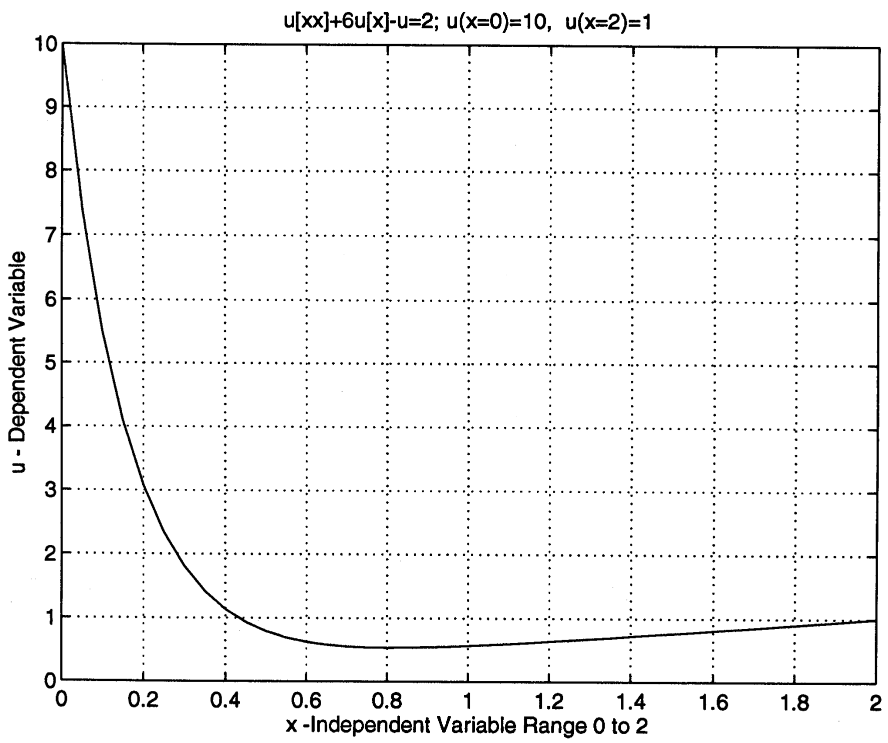
i=n-1 $\quad [1 - 3(\Delta x)]u_{n-2} + [-2 - \Delta x^2]u_{n-1} + [1 + 3\Delta x]1 = 2\Delta x^2$

Coefficients

$$a_{n-1} = 1 - 3(\Delta x) \qquad b_{n-1} = -2 - \Delta x^2 \qquad c_{n-1} = 0 \qquad d_{n-1} = 2\Delta x^2 - (1 - 3(\Delta x))$$

```
%Boundary Value Problem
%       u[xx]+6u[x]-u=2
%       BC. u(x=0)=10  u(x=2)=1
%  i=spatial index, from 1 to n
%   numbering for points is i=1 to i=21 for 20 dx spaces
%   u(1)=10 and u(n)=1


n=41; xspan=2.0;
   % Constants
dx=xspan/(n-1);
dx2=dx*dx;

   % sizing matrices
u=zeros(1,n);  x=zeros(1,n);
a=zeros(1,n);  b=zeros(1,n);  c=zeros(1,n);  d=zeros(1,n);
ba=zeros(1,n);  ga=zeros(1,n);
```

```
%Coefficients and Boundary Conditions
   x=0:dx:2;
   u(1)=10;   u(n)=1;
   b(2)=-2-dx2;
   c(2)=1+3*dx;
   d(2)=2*dx2-(1-3*dx)*10;
 for i=3:n-2,
   a(i)=1-3*dx;
   b(i)=-2-dx2;
   c(i)=1+3*dx;
   d(i)=2*dx2;
 end
   a(n-1)=1-3*dx;
   b(n-1)=-2-dx2;
   d(n-1)=2*dx2-(1+3*dx);

   %Solution by Thomas Algorithm
ba(2)=b(2);
ga(2)=d(2)/b(2);
for i=3:n-1,
   ba(i)=b(i)-a(i)*c(i-1)/ba(i-1);
   ga(i)=(d(i)-a(i)*ga(i-1))/ba(i);
end
   %back substitution step
u(n-1)=ga(n-1);
for i=n-2:-1:2,
   u(i)=ga(i)-c(i)*u(i+1)/ba(i);
end

   %Plot
plot(x,u)
  title('u[xx]+6u[x]-u=2; u(x=0)=10,  u(x=2)=1')
  xlabel('x -Independent Variable Range 0 to 2');  ylabel('u - Dependent Variable')
  grid
```



27.24

1. Divide the radial coordinate into $n$ finite points,

$$\Delta r = \frac{1}{n-1}$$

2. The finite difference approximations for the general point $i$

$$\frac{d^2T}{dr^2} = \frac{T_{i+1} - 2T_i + T_{i-1}}{\Delta r^2}$$

$$\frac{dT}{dr} = \frac{T_{i+1} - T_{i-1}}{2\Delta r}$$

$$r = \Delta r(i-1)$$

3. Substituting in the finite difference approximations for the derivatives

$$\frac{d^2T}{dt^2} + \frac{1}{r}\frac{dT}{dr} + S = 0$$

$$\frac{T_{i+1} - 2T_i + T_{i-1}}{\Delta r^2} + \left(\frac{1}{\Delta r(i-1)}\right)\left(\frac{T_{i+1} - T_{i-1}}{2\Delta r}\right) + S = 0$$

4. Collecting like terms results in the general finite difference equation at pint $i$

$$\left[1 - \frac{1}{2(i-1)}\right]T_{i-1} + [-2]T_i + \left[1 + \frac{1}{2(i-1)}\right]T_{i+1} = -\Delta r^2 S$$

5. End point equation at $i = 1$

$$\frac{dT}{dr}(r = 0) = 0$$

Substituting in the FD approximation gives

$$\frac{T_2 - T_0}{2\Delta r} = 0$$

where $T_o$ is a fictitious point, however we see that $T_0 = T_2$ for zero slope at $r = 0$. Writing out the general equation at point $i = 1$ gives:

$$\left[1 - \frac{1}{2(i-1)}\right]T_2 + [-2]T_1 + \left[1 + \frac{1}{2(i-1)}\right]T_2 = -\Delta r^2 S$$

and noting the two undefined terms $1/(2(i-1))$ add out of the equation gives (see not at end)

$$[-2]T_1 + [2]T_2 = -\Delta r^2 S$$

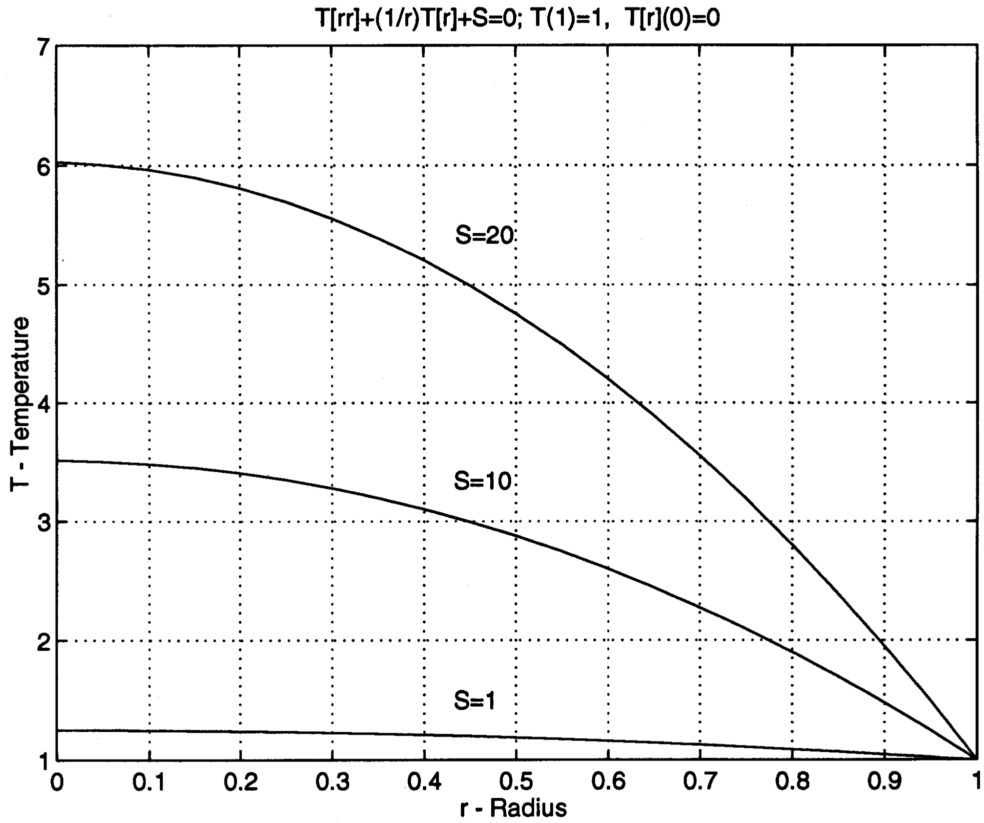
6. End point equation at $i = n - 1$

$$\left[1 - \frac{1}{2(i-1)}\right]T_{n-2} + [-2]T_{n-1} = -\Delta r^2 S - \left[1 + \frac{1}{2(i-1)}\right]$$

7. Solve the resulting tridiagonal system of algebraic equations using the Thomas Algorithm.
8. Following program in MATLAB.

Note: An alternate solution method is to move the first point $i = 1$ over half a $\Delta r$ step. This avoids the undefined quantities at the point $r = 0$.

```
%Solutions of the ODE Boundary Value Problem
%       T[rr]+(1/r)T[r]+S=0
%         BC. T(r=1)=1  T[r](r=0)=0
%          i=spatial index, from 1 to n
%          numbering for points : i=1 to i=21 for 20 dr spaces
%          i=1 (r=0), and i=n (r=1)
%          T(n)=1 and T'(0)=0

     %Constants
n=21;
dr=1/(n-1);
dr2=dr*dr;
S=1;

    %sizing matrices
r=0:dr:1;
T=zeros(1,n);
a=zeros(1,n);  b=zeros(1,n);  c=zeros(1,n);  d=zeros(1,n);
ba=zeros(1,n);  ga=zeros(1,n);

    %Coefficients and Boundary Conditions
 b(1)=-2;
 c(1)=2;
 d(1)=-dr2*S
 for i=2:n-2,
    a(i)=1-1/(2*(i-1));
    b(i)=-2;
    c(i)=1+1/(2*(i-1));
    d(i)=-dr2*S;
 end
 a(n-1)=1-1/(2*(n-2));
 b(n-1)=-2;
 d(n-1)=-dr2*S-(1+1/(2*(n-2)));
 T(n)=1;
     %Solution by Thomas Algorithm
 ba(1)=b(1);
 ga(1)=d(1)/b(1);
 for i=2:n-1,
    ba(i)=b(i)-a(i)*c(i-1)/ba(i-1);
    ga(i)=(d(i)-a(i)*ga(i-1))/ba(i);
 end
    %back substitution step
 T(n-1)=ga(n-1);
 for i=n-2:-1:1,
    T(i)=ga(i)-c(i)*T(i+1)/ba(i);
 end
    %Plot
 plot(r,T)
 title('T[rr]+(1/r)T[r]+S=0; T(1)=1,  T[r](0)=0')
 xlabel('r - Radius');  ylabel('T - Temperature')
 grid
 hold on
 gtext('S=20'); gtext('S=10'); gtext('S=1')
```




27.25

By summing forces on each mass and equating that to the mass times acceleration, the resulting differential equations are

$$\ddot{x}_1 + \left(\frac{k_1 + k_2}{m_1}\right)x_1 - \left(\frac{k_2}{m_1}\right)x_2 = 0$$

$$\ddot{x}_2 - \left(\frac{k_2}{m_2}\right)x_1 + \left(\frac{k_2 + k_3}{m_2}\right)x_2 - \left(\frac{k_3}{m_2}\right)x_3 = 0$$

$$\ddot{x}_3 - \left(\frac{k_3}{m_3}\right)x_2 + \left(\frac{k_3 + k_4}{m_3}\right)x_3 = 0$$

In matrix form

$$\begin{bmatrix} \ddot{x}_1 \\ \ddot{x}_2 \\ \ddot{x}_3 \end{bmatrix} + \begin{bmatrix} \left(\frac{k_1 + k_2}{m_1}\right) & -\left(\frac{k_2}{m_1}\right) & 0 \\ -\left(\frac{k_2}{m_2}\right) & \left(\frac{k_2 + k_3}{m_2}\right) & -\left(\frac{k_3}{m_2}\right) \\ 0 & -\left(\frac{k_3}{m_3}\right) & \left(\frac{k_3 + k_4}{m_3}\right) \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

The k/m matrix becomes with: $k_1 = k_4 = 10$ N/m, $k_2 = k_3 = 40$ N/m, and $m_1 = m_2 = m_3 = 1$ kg

$$\left[\frac{k}{m}\right] = \begin{bmatrix} 50 & -40 & 0 \\ -40 & 80 & -40 \\ 0 & -40 & 50 \end{bmatrix}$$

Solve for the eigenvalues/natural frequencies using MATLAB.

```
% 3 mass - 4 spring system
% natural frequencies
     k1=k4= 10 N/m     k2=k3= 40 N/m
%    m1=m2=m3=m4 = 1 kg

km=[50 -40 0; -40 80 -40; 0 -40 50]
w2=eig(km)

w=sqrt(w2)
km =

    50   -40     0
   -40    80   -40
     0   -40    50


w2 =

    6.4765
   50.0000
  123.5235


w =

    2.5449
    7.0711
   11.1141
```

27.26

```
k=1;
kmw2=[2*k,-k,-k;-k,2*k,-k;-k,-k,2*k];
[v,d]=eig(kmw2)

v =
   0.8034    0.1456    0.5774
  -0.2757   -0.7686    0.5774
  -0.5278    0.6230    0.5774

d =
   3.0000         0         0
        0    3.0000         0
        0         0    0.0000
```

Therefore, the eigenvalues are 0, 3, and 3. Setting these eigenvalues equal to $m\omega^2$, the three frequencies can be obtained.

$m\omega_1^2 = 0 \Rightarrow \omega_1 = 0$ (Hz) 1st mode of oscillation
$m\omega_2^2 = 0 \Rightarrow \omega_2 = \sqrt{3}$ (Hz) 2nd mode
$m\omega_3^2 = 0 \Rightarrow \omega_3 = \sqrt{3}$ (Hz) 3rd mode

27.7 (*a*) The exact solution is

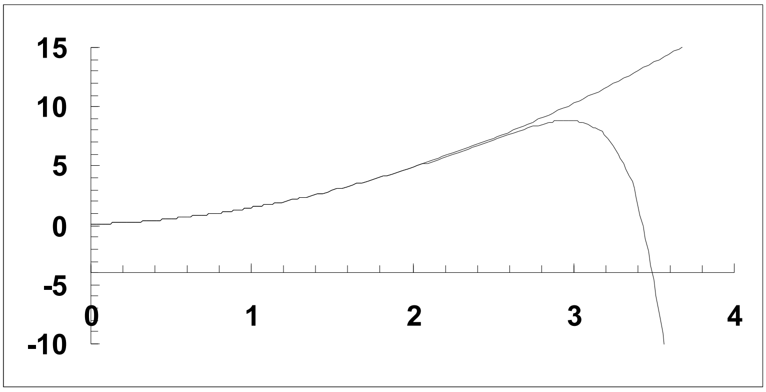
$$y = Ae^{5t} + t^2 + 0.4t + 0.08$$

If the initial condition at $t = 0$ is 0.8, A = 0,

$$y = t^2 + 0.4t + 0.08$$

Note that even though the choice of the initial condition removes the positive exponential terms, it still lurks in the background. Very tiny round off errors in the numerical solutions bring it to the fore. Hence all of the following solutions eventually diverge from the analytical solution.

(*b*) 4th order RK. The plot shows the numerical solution (bold line) along with the exact solution (fine line).




(*c*)

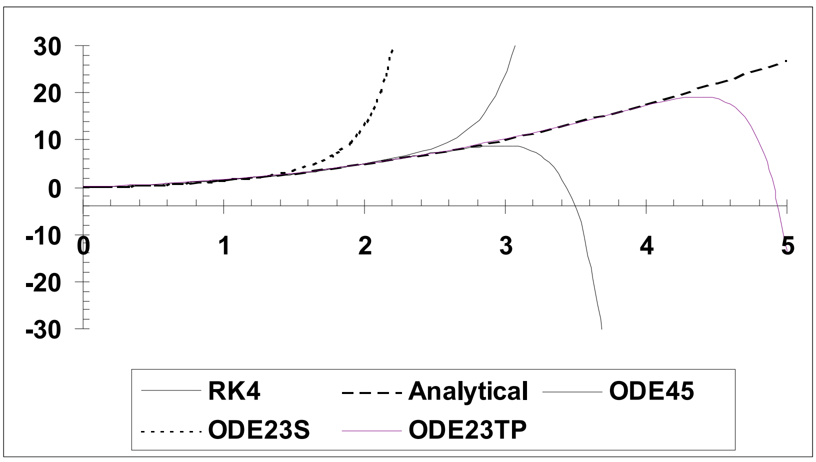
```
function yp=dy(t,y)
yp=5*(y-t^2);

>> tspan=[0,5];
>> y0=0.08;
>> [t,y]=ode45('dy1',tspan,y0);
```

(*d*)

```
>> [t,y]=ode23S('dy1',tspan,y0);
```

(*e*)

```
>> [t,y]=ode23TB('dy1',tspan,y0);
```



# CHAPTER 29

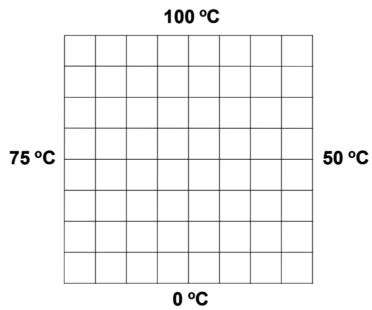
29.1

```
First iteration:
       7.500000        2.250000       15.675000
       9.750000        3.600000       20.782500
      55.425000       62.707500       85.047000
Error:
     100.000000      100.000000      100.000000
     100.000000      100.000000      100.000000
     100.000000      100.000000      100.000000
Second iteration:
       9.600000        8.212501       20.563500
      26.137500       34.632000       52.916250
      68.068500       88.782750       85.500310
Error:
      21.875000       72.602740       23.772710
      62.697270       89.604990       60.725670
      18.574660       29.369720    5.301830E-01
•
•
•
Seventh iteration:
      25.013610       28.806340       33.932440
      46.216590       56.257030       56.921290
      78.575310       93.082440       87.501180
Error:
   2.954020E-01    2.531316E-02    1.679560E-02
   2.267362E-02    2.082395E-02    1.041445E-02
   2.165254E-03    3.590016E-03    1.743838E-03
```

29.2 The fluxes for Prob. 29.1 can be calculated as

```
qx=
  -9.325527E-02   -2.185114E-01   -5.192447E-01
  -7.657973E-01   -2.622653E-01    1.532972E-01
      -1.668020   -2.186839E-01        1.055520
qy=
      -1.132306       -1.378297       -1.394572
      -1.312262       -1.574765       -1.312434
      -2.542694       -2.296703       -2.280428
qn=
       1.136140        1.395511        1.488101
       1.519367        1.596454        1.321357
       3.040984        2.307091        2.512862
theta=
     -94.708180      -99.008540     -110.421900
    -120.266600      -99.455400      -83.337820
    -123.265100      -95.439100      -65.162450
```

29.3 The plate is redrawn below




After 15 iterations of the Liebmann method, the result is

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 0 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 0 |
| 50 | 73.6954 | 82.3973 | 86.06219 | 87.7991 | 88.54443 | 88.19118 | 85.32617 | 75 |
| 50 | 62.3814 | 69.8296 | 74.0507 | 76.58772 | 78.18341 | 78.8869 | 78.10995 | 75 |
| 50 | 55.9987 | 60.4898 | 63.72554 | 66.32058 | 68.71677 | 71.06672 | 73.23512 | 75 |
| 50 | 51.1222 | 52.4078 | 54.04625 | 56.25934 | 59.3027 | 63.42793 | 68.75568 | 75 |
| 50 | 46.0804 | 43.9764 | 43.79945 | 45.37425 | 48.80563 | 54.57569 | 63.33804 | 75 |
| 50 | 39.2206 | 33.6217 | 31.80514 | 32.62971 | 35.95756 | 42.71618 | 54.995 | 75 |
| 50 | 27.1773 | 19.4897 | 17.16646 | 17.3681 | 19.66293 | 25.31308 | 38.86852 | 75 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

with percent approximate errors of

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0.0030 | 0.0040 | 0.0043 | 0.0049 | 0.0070 | 0.0114 | 0.0120 | 0 |
| 0 | 0.0050 | 0.0062 | 0.0057 | 0.0055 | 0.0079 | 0.0120 | 0.0109 | 0 |
| 0 | 0.0062 | 0.0067 | 0.0036 | 0.0007 | 0.0007 | 0.0097 | 0.0241 | 0 |
| 0 | 0.0076 | 0.0066 | 0.0020 | 0.0106 | 0.0067 | 0.0164 | 0.0542 | 0 |
| 0 | 0.0106 | 0.0079 | 0.0033 | 0.0074 | 0.0077 | 0.0400 | 0.1005 | 0 |
| 0 | 0.0149 | 0.0099 | 0.0119 | 0.0327 | 0.0630 | 0.1192 | 0.2343 | 0 |
| 0 | 0.0136 | 0.0013 | 0.0302 | 0.1259 | 0.2194 | 0.2925 | 0.7119 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

29.4 The solution is identical to Prob. 29.3, except that now the top edge must be modeled. This means that the nodes along the top edge are simulated with equations of the form

$$4T_{i,j} - T_{i-1,j} - T_{i+1,j} - 2T_{i,j-1} = 0$$

The resulting simulation (after 14 iterations) yields

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 50 | 50.38683 | 51.16385 | 52.6796 | 55.17802 | 58.7692 | 63.41846 | 68.9398 | 75 |
| 50 | 50.17211 | 50.76425 | 52.15054 | 54.58934 | 58.20129 | 62.96008 | 68.67918 | 75 |
| 50 | 49.51849 | 49.56564 | 50.58556 | 52.86931 | 56.56024 | 61.64839 | 67.93951 | 75 |
| 50 | 48.31607 | 47.39348 | 47.78093 | 49.79691 | 53.61405 | 59.2695 | 66.58047 | 75 |
| 50 | 46.33449 | 43.91569 | 43.37764 | 44.99165 | 48.94264 | 55.38806 | 64.29121 | 75 |
| 50 | 43.09381 | 38.56608 | 36.8614 | 37.93565 | 41.91332 | 49.21507 | 60.37012 | 75 |
| 50 | 37.46764 | 30.4051 | 27.61994 | 28.08718 | 31.71478 | 39.39338 | 53.1291 | 75 |
| 50 | 26.36368 | 17.98153 | 15.18654 | 15.20479 | 17.63115 | 23.73251 | 38.00928 | 75 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

with percent approximate errors of

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 0 | 0.0584 | 0.1318 | 0.2034 | 0.2606 | 0.2828 | 0.2493 | 0.1529 | 0 |
| 0 | 0.0722 | 0.1603 | 0.2473 | 0.3173 | 0.3424 | 0.2983 | 0.1862 | 0 |
| 0 | 0.0854 | 0.1883 | 0.2937 | 0.3788 | 0.4077 | 0.3438 | 0.2096 | 0 |
| 0 | 0.0933 | 0.2121 | 0.3441 | 0.4464 | 0.4754 | 0.3972 | 0.2247 | 0 |
| 0 | 0.0930 | 0.2300 | 0.3913 | 0.5097 | 0.5328 | 0.4468 | 0.2605 | 0 |
| 0 | 0.0873 | 0.2469 | 0.4299 | 0.5474 | 0.5611 | 0.4237 | 0.2747 | 0 |
| 0 | 0.0913 | 0.2827 | 0.4995 | 0.5852 | 0.5525 | 0.3157 | 0.0477 | 0 |
| 0 | 0.1131 | 0.3612 | 0.7054 | 0.9164 | 0.7958 | 0.5085 | 0.6345 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

29.5 The solution is identical to Examples 29.1 and 29.3, except that now heat balances must be developed for the three interior nodes on the bottom edge. For example, using the control-volume approach, node 1,0 can be modeled as

$$-0.49(5)\frac{T_{10}-T_{00}}{10}+0.49(5)\frac{T_{20}-T_{10}}{10}+0.49(10)\frac{T_{11}-T_{10}}{10}-2(10)=0$$

$$4T_{10}-T_{00}-T_{20}-2T_{11}=-81.63265$$

The resulting simulation yields (with a stopping criterion of 1% and a relaxation coefficient of 1.5)

| | | | | |
|---|---|---|---|---|
| 87.5 | 100 | 100 | 100 | 75 |
| 75 | 79.91669 | 77.76103 | 70.67812 | 50 |
| 75 | 66.88654 | 60.34068 | 55.39378 | 50 |
| 75 | 52.26597 | 40.84576 | 40.26148 | 50 |
| 75 | 27.12079 | 10.54741 | 14.83802 | 50 |

The fluxes for the computed nodes can be computed as

***qx***

| | | |
|---|---|---|
| -0.06765 | 0.226345 | 0.680145 |
| 0.359153 | 0.281573 | 0.253347 |
| 0.836779 | 0.29411 | -0.22428 |
| 1.579088 | 0.300928 | -0.96659 |

| ***qy*** | | |
|---|---|---|
| -0.81128 | -0.97165 | -1.09285 |
| -0.67744 | -0.90442 | -0.74521 |
| -0.97426 | -1.21994 | -0.99362 |
| -1.23211 | -1.48462 | -1.24575 |

***qn***

| | | |
|---|---|---|
| 0.814095 | 0.997668 | 1.287216 |
| 0.766759 | 0.947241 | 0.787095 |
| 1.284283 | 1.254887 | 1.018614 |
| 2.002904 | 1.514811 | 1.576764 |

**θ (radians)**

| | | |
|---|---|---|
| -1.65398 | -1.34193 | -1.0141 |
| -1.08331 | -1.26898 | -1.24309 |
| -0.86117 | -1.33422 | -1.7928 |

| | | |
|---|---|---|
| -0.66259 | -1.37081 | -2.23067 |

**θ (degrees)**

| | | |
|---|---|---|
| -94.7663 | -76.8869 | -58.1036 |
| -62.0692 | -72.7072 | -71.2236 |
| -49.3412 | -76.4454 | -102.72 |
| -37.9638 | -78.5416 | -127.808 |

29.6 The solution is identical to Example 29.5 and 29.3, except that now heat balances must be developed for the interior nodes at the lower left and the upper right edges. The balances for nodes 1,1 and 3,3 can be written as

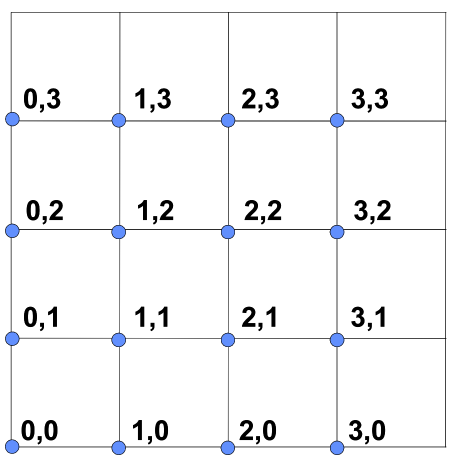
$$-4T_{11} + 0.8453T_{21} + 0.8453T_{12} = -1.154701(T_{01} + T_{10})$$

$$-4T_{33} + 0.8453T_{32} + 0.8453T_{23} = -1.154701(T_{34} + T_{43})$$

Using the appropriate boundary conditions, simple Laplacians can be used for the remaining interior nodes. The resulting simulation yields

| | | | | |
|---|---|---|---|---|
| 75 | 50 | 50 | 50 | |
| 100 | 75 | 63.97683 | 55.90731 | 50 |
| 100 | 86.02317 | 75 | 63.97683 | 50 |
| 100 | 94.09269 | 86.02317 | 75 | 50 |
| | 100 | 100 | 100 | 75 |

29.7 The nodes to be simulated are




Simple Laplacians are used for all interior nodes. Balances for the edges must take insulation into account. Fo example, node 1,0 is modeled as

$$4T_{1,0} - T_{0,0} - T_{2,0} - 2T_{1,1} = 0$$

The corner node, 0,0 would be modeled as

$$4T_{0,0} - 2T_{1,0} - 2T_{0,1} = 0$$

The resulting set of equations can be solved for

| | | | | |
|---|---|---|---|---|
| 0 | 12.5 | 25 | 37.5 | 50 |
| 11.94853 | 16.08456 | 22.79412 | 30.14706 | 37.5 |
| 15.625 | 17.09559 | 19.94485 | 22.79412 | 25 |
| 16.36029 | 16.72794 | 17.09559 | 16.08456 | 12.5 |
| 16.36029 | 16.36029 | 15.625 | 11.94853 | 0 |

The fluxes can be computed as

| $J_x$ | | | | |
|---|---|---|---|---|
| -0.6125 | -0.6125 | -0.6125 | -0.6125 | -0.6125 |
| -0.20267 | -0.26572 | -0.34453 | -0.36029 | -0.36029 |
| -0.07206 | -0.10584 | -0.13961 | -0.12385 | -0.10809 |
| -0.01801 | -0.01801 | 0.015763 | 0.112592 | 0.175643 |
| -5.6E-13 | 0.018015 | 0.108088 | 0.382812 | 0.585478 |

| $J_y$ | | | | |
|---|---|---|---|---|
| 0.585478 | 0.175643 | -0.10809 | -0.36029 | -0.6125 |
| 0.382812 | 0.112592 | -0.12385 | -0.36029 | -0.6125 |
| 0.108088 | 0.015763 | -0.13961 | -0.34453 | -0.6125 |
| 0.018015 | -0.01801 | -0.10584 | -0.26572 | -0.6125 |
| 0 | -0.01801 | -0.07206 | -0.20267 | -0.6125 |

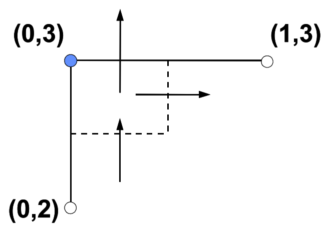
| $J_n$ | | | | |
|---|---|---|---|---|
| 0.847314 | 0.637187 | 0.621964 | 0.710611 | 0.866206 |
| 0.43315 | 0.288587 | 0.366116 | 0.509533 | 0.710611 |
| 0.129906 | 0.107004 | 0.197444 | 0.366116 | 0.621964 |
| 0.025477 | 0.025477 | 0.107004 | 0.288587 | 0.637187 |
| 5.63E-13 | 0.025477 | 0.129906 | 0.43315 | 0.847314 |

| **$\theta$ (radians)** | | | | |
|---|---|---|---|---|
| 2.378747 | 2.862322 | -2.96692 | -2.60987 | -2.35619 |
| 2.057696 | 2.740799 | -2.7965 | -2.35619 | -2.10252 |
| 2.158799 | 2.993743 | -2.35619 | -1.91589 | -1.74547 |
| 2.356194 | -2.35619 | -1.42295 | -1.17 | -1.29153 |
| 3.141593 | -0.7854 | -0.588 | -0.4869 | -0.80795 |

| **$\theta$ (degrees)** | | | | |
|---|---|---|---|---|
| 136.2922 | 163.999 | -169.992 | -149.534 | -135 |
| 117.8973 | 157.0362 | -160.228 | -135 | -120.466 |
| 123.6901 | 171.5289 | -135 | -109.772 | -100.008 |
| 135 | -135 | -81.5289 | -67.0362 | -73.999 |
| 180 | -45 | -33.6901 | -27.8973 | -46.2922 |

29.8 Node 0,3:

There are two approaches for modeling this node. One would be to consider it a Dirichlet node and not model it at all (i.e., set it's temperature at 50°C). The second alternative is to use a heat balance to model it as shown here



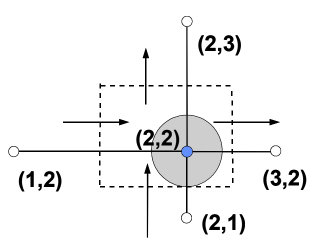

$$0 = 0.5(15)(1)\frac{T_{1,3} - T_{0,3}}{40} - 0.5(20)(1)\frac{T_{0,3} - T_{0,2}}{30} + 0.01(20)(1)(10 - T_{0,3})$$

$$-0.29752 T_{1,3} + 4T_{0,3} - 0.52893 T_{0,2} = 3.17355$$

Node 2,3:

$$0 = -0.5(15)(1)\frac{T_{2,3} - T_{1,3}}{40} + 0.5(15)(1)\frac{T_{3,3} - T_{2,3}}{20} - 0.5(30)(1)\frac{T_{2,3} - T_{2,2}}{30} + 0.01(30)(1)(10 - T_{2,3})$$
$$4T_{2,3} - 0.70588 T_{1,3} - 1.41176 T_{3,3} - 1.88235 T_{2,2}$$

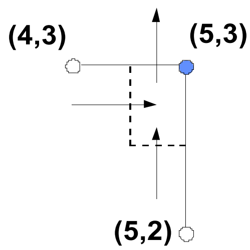
Node 2,2:




$$0 = -0.5(22.5)(1)\frac{T_{2,2} - T_{1,2}}{40} + 0.5(22.5)(1)\frac{T_{3,2} - T_{2,2}}{20} - 0.5(30)(1)\frac{T_{2,2} - T_{2,1}}{15}$$
$$+ 0.5(30)(1)\frac{T_{2,3} - T_{2,2}}{30} + 10\pi(7.5)^2$$

$$4T_{2,2} - 0.48T_{1,2} - 0.96T_{3,2} - 1.70667 T_{2,1} - 0.85333 T_{2,3} = 3015.93$$

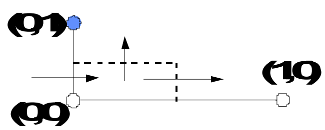
Node 5,3:




$$0 = -0.5(15)(1)\frac{T_{5,3} - T_{4,3}}{20} - 0.5(10)(1)\frac{T_{5,3} - T_{5,2}}{30} + 0.01(10)(1)(10 - T_{5,3})$$

$$4T_{5,3} - 2.33766T_{4,3} - 1.03896T_{5,2} = 6.23377$$

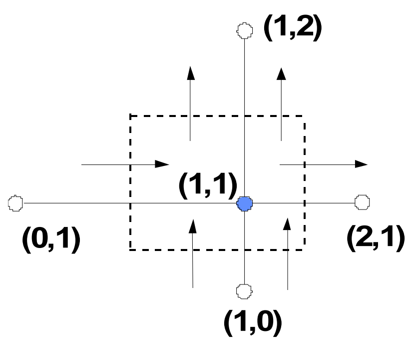
29.9 Node 0,0:



$$0 = 0.01(7.5)(2)(20 - T_{0,0}) - 0.7(7.5)(2)\frac{T_{1,0} - T_{0,0}}{40} + 0.7(20)(2)\frac{T_{0,1} - T_{0,0}}{15}$$

$$4T_{0,0} - 0.46069\,T_{1,0} - 3.27605T_{0,1} = 5.26508$$

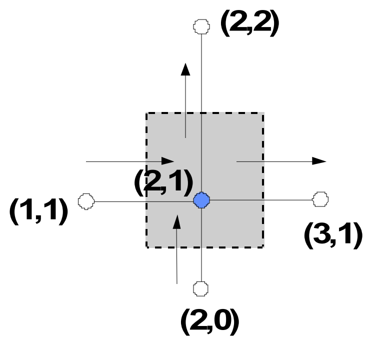
Node 1,1:




$$0 = -0.7(22.5)(2)\frac{T_{1,1} - T_{0,1}}{40} + 0.5(22.5)(2)\frac{T_{2,1} - T_{1,1}}{20} - 0.7(20)(2)\frac{T_{1,1} - T_{1,0}}{15}$$
$$- 0.5(10)(2)\frac{T_{1,1} - T_{1,0}}{15} + 0.7(20)(2)\frac{T_{1,2} - T_{1,1}}{30} + 0.5(10)(2)\frac{T_{1,2} - T_{1,1}}{30}$$

$$4T_{1,1} - 0.78755T_{2,1} - 1.77389\,T_{1,0} - 0.88694\,T_{1,2} - 0.55142\,T_{0,1} = 0$$

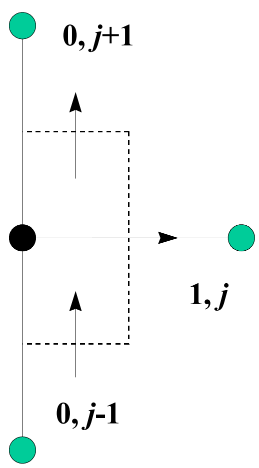
Node 2,1:




$$0 = -0.5(22.5)(2)\frac{T_{2,1} - T_{1,1}}{20} + 0.5(22.5)(2)\frac{T_{3,1} - T_{2,1}}{20} - 0.5(20)(2)\frac{T_{2,1} - T_{2,0}}{15}$$
$$- 0.5(20)(2)\frac{T_{2,2} - T_{2,1}}{30} + 10(22.5)(20)$$

$$4T_{2,1} - 1.05882T_{1,1} - 1.05882T_{3,1} - 1.2549T_{2,0} - 0.62745T_{2,2} = 4235.29$$

29.10 The control volume is drawn as in



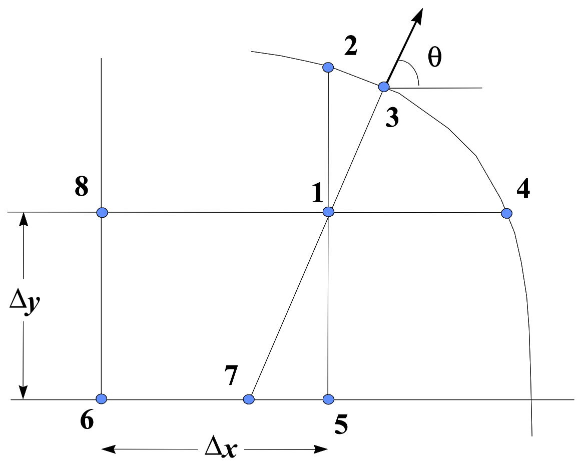

A flux balance around the node can be written as (note $\Delta x = \Delta y = h$)

$$- kh\Delta z\frac{T_{1,j} - T_{0,j}}{h} + k(h/2)\Delta z\frac{T_{0,j} - T_{0,j-1}}{h} - k(h/2)\Delta z\frac{T_{1,j} - T_{0,j}}{h} = 0$$

Collecting and cancelling terms gices

$$T_{0,j} - T_{0,j-1} - T_{0,j+1} - 2T_{1,j} = 0$$

29.11 A setup similar to Fig. 29.11, but with θ > 45° can be drawn as in




The normal derivative at node 3 can be approximated by the gradient between nodes 1 and 7,

$$\left.\frac{\partial T}{\partial \eta}\right|_3 = \frac{T_1 - T_7}{L_{17}}$$

When θ is greater than 45° as shown, the distance from node 5 to 7 is $\Delta y$ cotθ, and linear interpolation can be used to estimate

$$T_7 = T_5 + (T_6 - T_5)\frac{\Delta y \cot\theta}{\Delta x}$$

The length $L_{17}$ is equal to $\Delta y/\sin\theta$. This length, along with the approximation for $T_7$ can be substituted into the gradient equation to give

$$T_1 = \left(\frac{\Delta y}{\sin\theta}\right)\left.\frac{\partial T}{\partial \eta}\right|_3 - T_6\frac{\Delta y \cot\theta}{\Delta x} - T_5\left(1 - \frac{\Delta y \cot\theta}{\Delta x}\right)$$

29.12 The following Fortran-90 program implements Liebmann's method with relaxation.

```
PROGRAM liebmann
IMPLICIT NONE
INTEGER :: nx,ny,l,i,j
REAL :: T(0:5,0:5),ea(0:5,0:5),Told(0:5,0:5)
REAL :: qy(0:5,0:5),qx(0:5,0:5),qn(0:5,0:5),th(0:5,0:5)
REAL :: Trit,Tlef,Ttop,Tbot,lam,emax,es,pi
REAL :: k,x,y,dx,dy
nx=4
ny=4
pi=4.*atan(1.)
x=40.
y=40.
k=0.49
lam=1.2
es=1.
dx=x/nx
```

```
dy=y/ny
Tbot=0.
Tlef=25.
Trit=50.
Ttop=150.
DO i=1,nx-1
  T(i,0)=Tbot
END DO
DO i=1,nx-1
  T(i,ny)=Ttop
END DO
DO j=1,ny-1
  T(0,j)=Tlef
END DO
DO j=1,ny-1
  T(nx,j)=Trit
END DO
l=0
DO
  l=l+1
  emax=0.
  DO j = 1,ny-1
    DO i = 1,nx-1
      Told(i,j)=T(i,j)
      T(i,j)=(T(i+1,j)+T(i-1,j)+T(i,j+1)+T(i,j-1))/4
      T(i,j)=lam*T(i,j)+(1-lam)*Told(i,j)
      ea(i,j)=abs((T(i,j)-Told(i,j))/T(i,j))*100.
      if(ea(i,j).GT.emax) emax=ea(i,j)
    END DO
  END DO
  PRINT *, 'iteration = ',l
  DO j = 1,ny-1
    PRINT *, (T(i,j),i=1,nx-1)
  END DO
  PRINT *
  DO j = 1,ny-1
    PRINT *, (ea(i,j),i=1,nx-1)
  END DO
  IF (emax.LE.es) EXIT
END DO
DO j = 1,ny-1
  DO i = 1,nx-1
    qy(i,j)=-k*(T(i,j+1)-T(i,j-1))/2/dy
    qx(i,j)=-k*(T(i+1,j)-T(i-1,j))/2/dx
    qn(i,j)=sqrt(qy(i,j)**2+qx(i,j)**2)
    th(i,j)=atan2(qy(i,j),qx(i,j))*180./pi
  END DO
END DO
PRINT *,'qx='
DO j = 1,ny-1
  PRINT *, (qx(i,j),i=1,nx-1)
END DO
PRINT *,'qy='
DO j = 1,ny-1
  PRINT *, (qy(i,j),i=1,nx-1)
END DO
PRINT *,'qn='
DO j = 1,ny-1
  PRINT *, (qn(i,j),i=1,nx-1)
END DO
PRINT *,'theta='
DO j = 1,ny-1
  PRINT *, (th(i,j),i=1,nx-1)
END DO
END
```

When the program is run, the result of the last iteration is:

```
iteration =            6
   42.81303       33.26489       33.93646
   63.17175       56.26600       52.46138
   78.57594       76.12081       69.64268

  0.5462000      0.1074174      2.4864437E-02
  1.1274090E-02  2.0983342E-02  4.8064217E-02
  3.1769749E-02  3.6572997E-02  2.4659829E-02
 qx=
   1.022510      0.2174759     -0.4100102
  0.4589829      0.2624041      0.1535171
 -2.7459882E-02  0.2188648      0.6399599
 qy=
  -1.547708      -1.378517      -1.285304
 -0.8761914      -1.049970     -0.8748025
 -0.9022922      -1.071483      -1.164696
 qn=
   1.854974       1.395566       1.349116
  0.9891292       1.082263      0.8881705
  0.9027100       1.093608       1.328934
 theta=
  -56.54881      -81.03486      -107.6926
  -62.35271      -75.96829      -80.04664
  -91.74317      -78.45538      -61.21275
Press any key to continue
```

29.13 When the program is run, the result of the last iteration is:

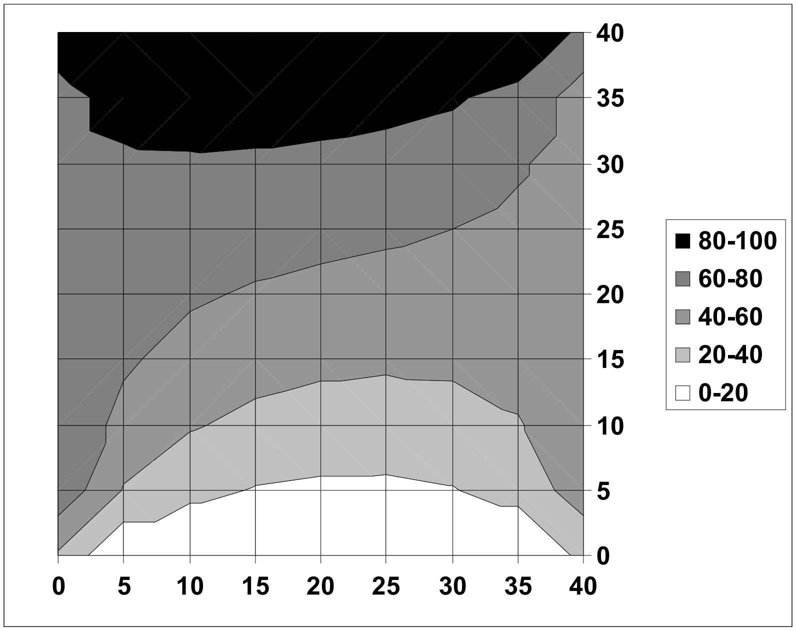
```
iteration =            7
   25.01361       28.80634       33.93244
   46.21659       56.25703       56.92129
   78.57531       93.08244       87.50118

  0.2954020      2.5313158E-02  1.6795604E-02
  2.2673620E-02  2.0823948E-02  1.0414450E-02
  2.1652540E-03  3.5900162E-03  1.7438381E-03
 qx=
 -9.3255267E-02 -0.2185114     -0.5192447
 -0.7657973     -0.2622653      0.1532972
  -1.668020     -0.2186839       1.055520
 qy=
  -1.132306      -1.378297      -1.394572
  -1.312262      -1.574765      -1.312434
  -2.542694      -2.296703      -2.280428
 qn=
   1.136140       1.395511       1.488101
   1.519367       1.596454       1.321357
   3.040984       2.307091       2.512862
 theta=
  -94.70818      -99.00854      -110.4219
  -120.2666      -99.45540      -83.33782
  -123.2651      -95.43910      -65.16245
Press any key to continue
```

29.14 When the program is run, the result of the last iteration is:

```
iteration =   19
38.490 24.764 19.044 16.783 16.696      19.176 27.028
54.430 41.832 34.955 31.682 31.041      33.110 38.976
62.710 53.570 47.660 44.291 42.924      43.388 45.799
68.165 62.478 58.219 55.234 53.215      51.848 50.854
72.761 70.301 67.841 65.489 63.051      60.034 55.780
```

```
77.795 78.373 77.594 76.027 73.595       69.522 62.233
85.175 87.944 88.261 87.530 85.843       82.249 73.624
```

This data can be imported into Excel and the following contour plot created:



29.15

$$\Sigma q = 0$$

$$q_{LEFT} - q_{RIGHT} + q_{LOWER-A} + q_{LOWER-B} - q_{UPPER-A} - q_{UPPER-B} + q_{SOURCE} = 0$$

$$k_A\,(h\,\Delta z)\,\frac{T_{21}-T_{22}}{h} - k_B\,(h\,\Delta z)\,\frac{T_{22}-T_{23}}{h/2} + k_A\left(\frac{h}{2}\,\Delta z\right)\frac{T_{12}-T_{22}}{h} + k_B\left(\frac{h}{4}\,\Delta z\right)\frac{T_{12}-T_{22}}{h} -$$

$$- k_A\left(\frac{h}{2}\,\Delta z\right)\frac{T_{22}-T_{32}}{h} - k_B\left(\frac{h}{4}\,\Delta z\right)\frac{T_{22}-T_{32}}{h} + S\,(h)\left(\frac{3}{4}h\right)\Delta z = 0$$

$$\left(.3\,\frac{W}{cm\,^\circ C}\right)(10\text{ cm})(.5\text{ cm})\left(\frac{T_{21}-T_{22}}{10\text{ cm}}\right) - (.5)(10)(.5)\left(\frac{T_{22}-T_{23}}{5}\right) + (.3)(5)(.5)\left(\frac{T_{12}-T_{22}}{10}\right) +$$

$$+ (.5)(2.5)(.5)\left(\frac{T_{12}-T_{22}}{10}\right) - (.3)(5)(.5)\left(\frac{T_{22}-T_{32}}{10}\right) - (.5)(2.5)(.5)\left(\frac{T_{22}-T_{32}}{10}\right) + \left(5\,\frac{W}{cm^3}\right)(10)\left(\frac{3}{4}10\right)(.5) =$$

$$0.15(T_{21}-T_{22}) - 0.5(T_{22}-T_{23}) + 0.075(T_{12}-T_{22}) + 0.0625(T_{12}-T_{22})$$

$$- 0.075(T_{22}-T_{32}) - 0.0625(T_{22}-T_{32}) + 187.5 = 0 \qquad \text{WATTS}$$

$$.15\,T_{21} + (-.15 - .5 - .075 - .0625 - .075 - .0625)\,T_{22} + .5\,T_{23} + (.075 + .0625)\,T_{12} +$$

$$(.075 + .0625)\,T_{32} = -187.5$$

$$-0.93\,T_{22} + .15\,T_{21} + .5\,T_{23} + 0.1375\,T_{12} + 0.1375\,T_{32} = -187.5$$

$$4\,T_{22} - 0.645\,T_{21} - 2.15\,T_{23} - 0.591\,T_{12} - 0.591\,T_{32} = 806.4$$

29.16

$$q = -k\frac{dT}{dx}$$

HORIZONTAL FLUX

$$q_{xA} = -\left(.3\frac{W}{cm\,°C}\right)\frac{(51.6 - 74.3)\,°C}{10\ cm} = 0.681\ \frac{W}{cm^2}$$ FROM A INTO B

$$q_{xB} = -(.5)\left(\frac{44.8 - 51.6}{5\ cm}\right) = 0.680\ \frac{W}{cm^2}$$ FROM A INTO B

THE FLUX AT THE BOUNDARY SHOULD BE EQUAL.

VERTICAL FLUX

$$q_{yA} = -.3\left(\frac{38.6 - 87.4}{2\ (10\ cm)}\right) = 0.732\ \frac{W}{cm^2}$$ UPWARD

$$q_{yB} = -.5\left(\frac{38.6 - 87.4}{2(10)}\right) = 1.22\ \frac{W}{cm^2}$$ UPWARD

THE 2 FLUXES MUST BE UNEQUAL.

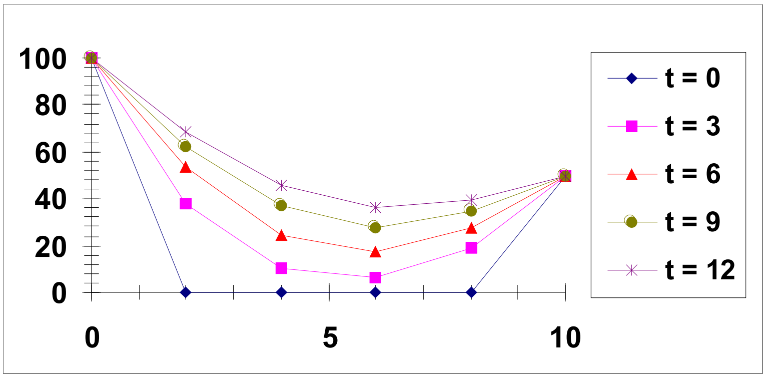
## CHAPTER 30

30.1 The key to approaching this problem is to recast the PDE as a system of ODEs. Thus, by substituting the finite-difference approximation for the spatial derivative, we arrive at the following general equation for each node

$$\frac{dT_i}{dt} = k\frac{T_{i-1} - 2T_i + T_{i+1}}{\Delta x^2}$$

By writing this equation for each node, the solution reduces to solving 4 simultaneous ODEs with Heun's method. The results for the first two steps along with some later selected values are tabulated below. In addition, a plot similar to Fig. 30.4, is also shown

| ***t*** | ***x* = 0** | ***x* = 2** | ***x* = 4** | ***x* = 6** | ***x* = 8** | ***x* = 10** |
|---|---|---|---|---|---|---|
| 0 | 100 | 0 | 0 | 0 | 0 | 50 |
| 0.1 | 100 | 2.04392 | 0.02179 | 0.01089 | 1.02196 | 50 |
| 0.2 | 100 | 4.00518 | 0.08402 | 0.04267 | 2.00259 | 50 |
| • | | | | | | |
| • | | | | | | |
| • | | | | | | |
| 3 | 100 | 37.54054 | 10.2745 | 6.442321 | 18.95732 | 50 |
| 6 | 100 | 53.24295 | 24.66054 | 17.46032 | 27.92252 | 50 |
| 9 | 100 | 62.39033 | 36.64937 | 27.84901 | 34.34692 | 50 |
| 12 | 100 | 68.71329 | 46.03496 | 36.5421 | 39.53549 | 50 |




30.2 Because we now have derivative boundary conditions, the boundary nodes must be simulated. For node 0,

$$T_0^{l+1} = T_0^l + \lambda(T_1^l - 2T_0^l + T_{-1}^l) \qquad (i)$$

This introduces an exterior node into the solution at $i = -1$. The derivative boundary condition can be used to eliminate this node,

$$\left.\frac{dT}{dx}\right|_0 = \frac{T_1 - T_{-1}}{2\Delta x}$$

which can be solved for

$$T_{-1} = T_1 - 2\Delta x\frac{dT_0}{dx}$$

which can be substituted into Eq. (i) to give

$$T_0^{l+1} = T_0^l + \lambda\left(2T_1^l - 2T_0^l - 2\Delta x \frac{dT_0^l}{dx}\right)$$

For our case, $dT_0/dx = 1$ and $\Delta x = 2$, and therefore $T_{-1} = T_1 + 4$. This can be substituted into Eq. ($i$) to give,

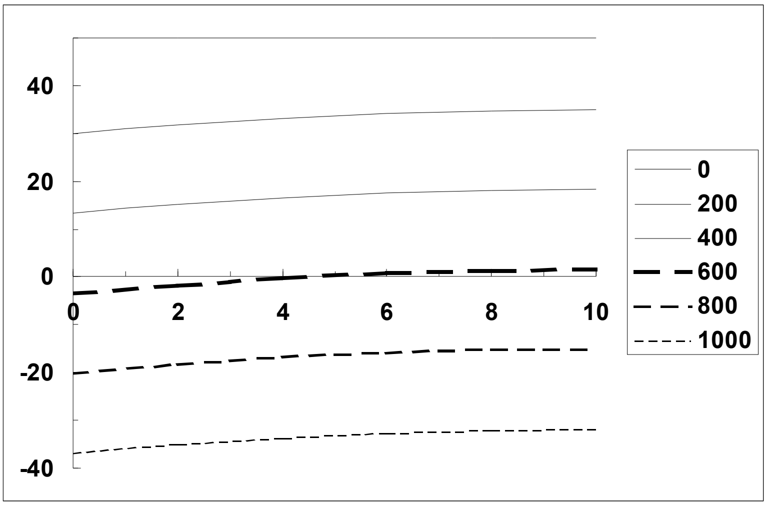
$$T_0^{l+1} = T_0^l + \lambda(2T_1^l - 2T_0^l + 4)$$

A similar analysis can be used to embed the zero derivative in the equation for the fifth node. The result is

$$T_5^{l+1} = T_5^l + \lambda(2T_4^l - 2T_5^l)$$

Together with the equations for the interior nodes, the entire system can be iterated with a step of 0.1 s. The results for some of the early steps along with some later selected values are tabulated below. In addition, a plot of the later results is also shown

| ***t*** | ***x* = 0** | ***x* = 2** | ***x* = 4** | ***x* = 6** | ***x* = 8** | ***x* = 10** |
|---|---|---|---|---|---|---|
| 0 | 50.0000 | 50.0000 | 50.0000 | 50.0000 | 50.0000 | 50.0000 |
| 0.1 | 49.9165 | 50.0000 | 50.0000 | 50.0000 | 50.0000 | 49.9165 |
| 0.2 | 49.8365 | 49.9983 | 50.0000 | 50.0000 | 49.9983 | 49.8365 |
| 0.3 | 49.7597 | 49.9949 | 50.0000 | 50.0000 | 49.9949 | 49.7597 |
| 0.4 | 49.6861 | 49.9901 | 49.9999 | 49.9999 | 49.9901 | 49.6861 |
| 0.5 | 49.6153 | 49.9840 | 49.9997 | 49.9997 | 49.9840 | 49.6153 |
| • | | | | | | |
| • | | | | | | |
| • | | | | | | |
| 200 | 30.00022 | 31.80019 | 33.20009 | 34.19992 | 34.79981 | 34.99978 |
| 400 | 13.30043 | 15.10041 | 16.50035 | 17.50028 | 18.10024 | 18.30023 |
| 600 | -3.40115 | -1.60115 | -0.20115 | 0.798846 | 1.398847 | 1.598847 |
| 800 | -20.1055 | -18.3055 | -16.9055 | -15.9055 | -15.3055 | -15.1055 |
| 1000 | -36.8103 | -35.0103 | -33.6103 | -32.6103 | -32.0103 | -31.8103 |




Notice what's happening. The rod never reaches a steady state, because of the heat loss at the left end (unit gradient) and the insulated condition (zero gradient) at the right.

30.3 The solution for $\Delta t = 0.1$ is (as computed in Example 30.1),

| $t$ | $x = 0$ | $x = 2$ | $x = 4$ | $x = 6$ | $x = 8$ | $x = 10$ |
|---|---|---|---|---|---|---|
| 0 | 100 | 0 | 0 | 0 | 0 | 50 |
| 0.1 | 100 | 2.0875 | 0 | 0 | 1.04375 | 50 |
| 0.2 | 100 | 4.087847 | 0.043577 | 0.021788 | 2.043923 | 50 |

For $\Delta t = 0.05$, it is

| $t$ | $x = 0$ | $x = 2$ | $x = 4$ | $x = 6$ | $x = 8$ | $x = 10$ |
|---|---|---|---|---|---|---|
| 0 | 100 | 0 | 0 | 0 | 0 | 50 |
| 0.05 | 100 | 1.04375 | 0 | 0 | 0.521875 | 50 |
| 0.1 | 100 | 2.065712 | 1.09E-02 | 5.45E-03 | 1.032856 | 50 |
| 0.15 | 100 | 3.066454 | 3.23E-02 | 0.016228 | 1.533227 | 50 |
| 0.2 | 100 | 4.046528 | 6.38E-02 | 3.22E-02 | 2.023265 | 50 |

To assess the differences between the results, we performed the simulation a third time using a more accurate approach (the Heun method) with a much smaller step size ($\Delta t$ = 0.001). It was assumed that this more refined approach would yield a prediction close to true solution. These values could then be used to assess the relative errors of the two Euler solutions. The results are summarized as

| | x = 0 | x = 2 | x = 4 | x = 6 | x = 8 | x = 10 |
|---|---|---|---|---|---|---|
| **Heun (h = 0.001)** | 100 | 4.006588 | 0.083044 | 0.042377 | 2.003302 | 50 |
| **Euler (h = 0.1)** | 100 | 4.087847 | 0.043577 | 0.021788 | 2.043923 | 50 |
| **Error relative to Heun** | | 2.0% | 47.5% | 48.6% | 2.0% | |
| **Euler (h = 0.05)** | 100 | 4.046528 | 0.063786 | 0.032229 | 2.023265 | 50 |
| **Error relative to Heun** | | 1.0% | 23.2% | 23.9% | 1.0% | |

Notice, that as would be expected for Euler's method, halving the step size approximately halves the global relative error.

30.4 The approach described in Example 30.2 must be modified to account for the zero derivative at the right hand node ($i$ = 5). To do this, Eq. (30.8) is first written for that node as

$$-\lambda T_4^{l+1} + (1+2\lambda)T_5^{l+1} - \lambda T_6^{l+1} = T_5^l \qquad (i)$$

The value outside the system ($i$ = 6) can be eliminated by writing the finite difference relationship for the derivative at node 5 as

$$\left.\frac{dT}{dx}\right|_5 = \frac{T_6 - T_4}{2\Delta x}$$

which can be solved for

$$T_6 = T_4 - 2\Delta x \left.\frac{dT}{dx}\right|_5$$

For our case, $dT/dx = 0$, so $T_6 = T_4$ and Eq. ($i$) becomes

$$-2\lambda T_4^{l+1} + (1+2\lambda) T_5^{l+1} = T_5^l$$

Thus, the simultaneous equations to be solved at the first step are

$$\begin{bmatrix} 1.04175 & -0.020875 & & & \\ -0.020875 & 1.04175 & -0.020875 & & \\ & -0.020875 & 1.04175 & -0.020875 & \\ & & -0.020875 & 1.04175 & -0.020875 \\ & & & -0.04175 & 1.04175 \end{bmatrix} \begin{Bmatrix} T_1^1 \\ T_2^1 \\ T_3^1 \\ T_4^1 \\ T_5^1 \end{Bmatrix} = \begin{Bmatrix} 2.0875 \\ 0 \\ 0 \\ 0 \\ 0 \end{Bmatrix}$$

which can be solved for

$$\begin{Bmatrix} 2.004645 \\ 0.040186 \\ 0.000806 \\ 1.62 \times 10^{-5} \\ 6.47 \times 10^{-7} \end{Bmatrix}$$

For the second step, the right-hand side is modified to reflect these computed values of $T$ at $t = 0.1$,

$$\begin{bmatrix} 1.04175 & -0.020875 & & & \\ -0.020875 & 1.04175 & -0.020875 & & \\ & -0.020875 & 1.04175 & -0.020875 & \\ & & -0.020875 & 1.04175 & -0.020875 \\ & & & -0.04175 & 1.04175 \end{bmatrix} \begin{Bmatrix} T_1^1 \\ T_2^1 \\ T_3^1 \\ T_4^1 \\ T_5^1 \end{Bmatrix} = \begin{Bmatrix} 4.092145 \\ 0.040186 \\ 0.000806 \\ 1.62 \times 10^{-5} \\ 6.47 \times 10^{-7} \end{Bmatrix}$$

which can be solved for

$$\begin{Bmatrix} 3.930497 \\ 0.117399 \\ 0.003127 \\ 7.83 \times 10^{-5} \\ 3.76 \times 10^{-6} \end{Bmatrix}$$

30.5 The solution is identical to Example 30.3, but with 6 segments. Thus, the simultaneous equations to be solved at the first step are

$$\begin{bmatrix} 2.06012 & -0.03006 & & & \\ -0.020875 & 2.06012 & -0.03006 & & \\ & -0.03006 & 2.06012 & -0.03006 & \\ & & -0.03006 & 2.06012 & -0.03006 \\ & & & -0.03006 & 2.06012 \end{bmatrix} \begin{Bmatrix} T_1^1 \\ T_2^1 \\ T_3^1 \\ T_4^1 \\ T_5^1 \end{Bmatrix} = \begin{Bmatrix} 6.012 \\ 0 \\ 0 \\ 0 \\ 3.006 \end{Bmatrix}$$

which can be solved for

$$\begin{Bmatrix} 2.91890 \\ 0.04260 \\ 0.00093 \\ 0.02131 \\ 1.45945 \end{Bmatrix}$$

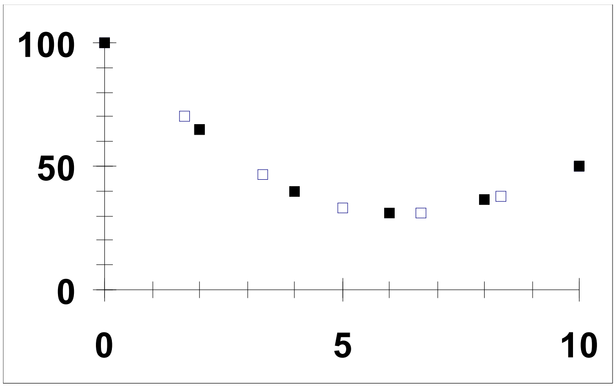
For the second step, the right-hand side is modified to reflect these computed values of $T$ at $t = 0.1$,

$$\begin{bmatrix} 2.06012 & -0.03006 & & & \\ -0.020875 & 2.06012 & -0.03006 & & \\ & -0.03006 & 2.06012 & -0.03006 & \\ & & -0.03006 & 2.06012 & -0.03006 \\ & & & -0.03006 & 2.06012 \end{bmatrix} \begin{Bmatrix} T_1^1 \\ T_2^1 \\ T_3^1 \\ T_4^1 \\ T_5^1 \end{Bmatrix} = \begin{Bmatrix} 11.67559 \\ 0.17042 \\ 0.00373 \\ 0.08524 \\ 5.83780 \end{Bmatrix}$$

which can be solved for

$$\begin{Bmatrix} 5.66986 \\ 0.16553 \\ 0.00543 \\ 0.08282 \\ 2.83493 \end{Bmatrix}$$

The solution at $t = 10$ for this problem ($n = 6$) along with the results determined for $n = 5$, as in Example 30.3, are displayed in the following plot:




30.6 Using the approach followed in Example 30.5, Eq. (30.20) is applied to nodes (1,1), (1,2), and (1,3) to yield the following tridiagonal equations

$$\begin{bmatrix} 2.167 & -0.0835 & \\ -0.0835 & 2.167 & -0.0835 \\ & -0.0835 & 2.167 \end{bmatrix} \begin{Bmatrix} T_{1,1} \\ T_{1,2} \\ T_{1,3} \end{Bmatrix} = \begin{Bmatrix} 6.2625 \\ 6.2625 \\ 18.7875 \end{Bmatrix}$$

which can be solved for

$$T_{1,1} = 3.018843 \quad T_{1,2} = 3.345301 \quad T_{1,3} = 8.798722$$

In a similar fashion, tridiagonal equations can be developed and solved for

$$T_{2,1} = 0.130591 \quad T_{2,2} = 0.370262 \quad T_{2,3} = 6.133184$$

and

$$T_{3,1} = 1.1017962 \quad T_{3,2} = 1.287655 \quad T_{3,3} = 7.029137$$

For the second step to $t$ = 10, Eq. (30.22) is applied to nodes (1,1), (2,1), and (3,1) to yield

$$\begin{bmatrix} 2.167 & -0.0835 & \\ -0.0835 & 2.167 & -0.0835 \\ & -0.0835 & 2.167 \end{bmatrix} \begin{Bmatrix} T_{1,1} \\ T_{1,2} \\ T_{1,3} \end{Bmatrix} = \begin{Bmatrix} 12.07537 \\ 0.27029 \\ 4.060943 \end{Bmatrix}$$

which can be solved for

$$T_{1,1} = 5.5883 \quad T_{1,2} = 0.412884 \quad T_{1,3} = 1.889903$$

Tridiagonal equations for the other rows can be developed and solved for

$$T_{2,1} = 6.308761 \quad T_{2,2} = 0.902193 \quad T_{2,3} = 2.430939$$

and

$$T_{3,1} = 16.8241 \quad T_{3,2} = 12.1614 \quad T_{3,3} = 13.25121$$

Thus, the result at the end of the first step can be summarized as

| | $i = 0$ | $i = 1$ | $i = 2$ | $i = 3$ | $i = 4$ |
|---|---|---|---|---|---|
| $j = 4$ | | 150 | 150 | 150 | |
| $j = 3$ | 75 | 16.824 | 12.161 | 13.251 | 25 |
| $j = 2$ | 75 | 6.309 | 0.902 | 2.431 | 25 |
| $j = 1$ | 75 | 5.588 | 0.413 | 1.89 | 25 |
| $j = 0$ | | 0 | 0 | 0 | |

The computation can be repeated, and the results for $t$ = 2000 s are below:

| | $i = 0$ | $i = 1$ | $i = 2$ | $i = 3$ | $i = 4$ |
|---|---|---|---|---|---|
| $j = 4$ | | 150 | 150 | 150 | |
| $j = 3$ | 75 | 98.214 | 97.768 | 80.357 | 25 |
| $j = 2$ | 75 | 70.089 | 62.5 | 48.661 | 25 |
| $j = 1$ | 75 | 44.643 | 33.482 | 26.786 | 25 |
| $j = 0$ | | 0 | 0 | 0 | |

30.7 Although this problem can be modeled with the finite-difference approach (see Sec. 32.1), the control-volume method provides a more straightforward way to handle the boundary conditions.

The boundary fluxes and the reaction term can be used to develop the discrete form of the advection-diffusion equation for the interior volumes as

$$\Delta x \frac{dc_i^l}{dt} = -D \frac{c_i^l - c_{i-1}^l}{\Delta x} + D \frac{c_{i+1}^l - c_i^l}{\Delta x} + U \frac{c_i^l + c_{i-1}^l}{2} - U \frac{c_{i+1}^l + c_i^l}{2} - k \Delta x c_i^l$$

or dividing both sides by $\Delta x$,

$$\frac{dc_i^l}{dt} = D \frac{c_{i+1}^l - 2c_i^l + c_{i-1}^l}{\Delta x^2} - U \frac{c_{i+1}^l + c_{i-1}^l}{2\Delta x} - kc_i^l$$

which is precisely the form that would have resulted by substituting centered finite difference approximations into the advection-diffusion equation.

For the first boundary node, no diffusion is allowed up the entrance pipe and advection is handled with a backward difference,

$$\Delta x \frac{dc_1^l}{dt} = D \frac{c_2^l - c_1^l}{\Delta x} + Uc_0^l - U \frac{c_2^l + c_1^l}{2} - k \Delta x c_1^l$$

or dividing both sides by $\Delta x$,

$$\frac{dc_1^l}{dt} = D \frac{c_2^l - c_1^l}{\Delta x^2} + \frac{2c_0^l - c_2^l - c_1^l}{2\Delta x} - kc_1^l$$

For the last boundary node, no diffusion is allowed through the exit pipe and advection out of the tank is again handled with a backward difference,

$$\Delta x \frac{dc_n^l}{dt} = -D \frac{c_n^l - c_{n-1}^l}{\Delta x} + U \frac{c_n^l + c_{n-1}^l}{2} - Uc_n^l - k \Delta x c_n^l$$

or dividing both sides by $\Delta x$,

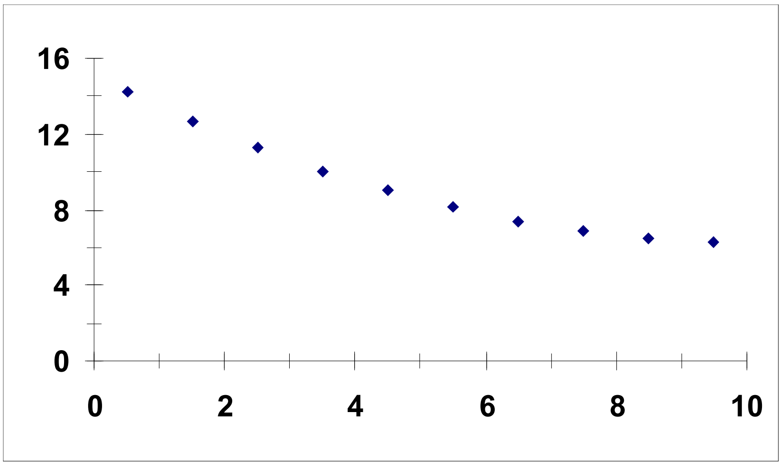
$$\frac{dc_n^l}{dt} = -D \frac{c_n^l - c_{n-1}^l}{\Delta x^2} + U \frac{c_{n-1}^l - c_n^l}{2\Delta x} - kc_n^l$$

By writing these equations for each equally-spaced volume, the PDE is transformed into a system of ODEs. Explicit methods like Euler's method or other higher-order RK methods can then be used to solve the system.

The results with and initial condition that the reactor has zero concentration with an inflow concentration of 100 (using Euler with a step size of 0.005) for t = 100 are

| x | 0.5 | 1.5 | 2.5 | 3.5 | 4.5 | 5.5 | 6.5 | 7.5 | 8.5 | 9.5 |
|---|---|---|---|---|---|---|---|---|---|---|
| c | 14.2042 | 12.6506 | 11.2610 | 10.0385 | 8.9847 | 8.1025 | 7.3928 | 6.8583 | 6.5000 | 6.3201 |

The results are also plotted below:




30.8

```
Option Explicit

Sub EulerPDE()

Dim i As Integer, j As Integer, np As Integer, ns As Integer
Dim Te(20) As Single, dTe(20) As Single, tpr(20) As Single, Tepr(20, 20)
As Single
Dim k As Single, dx As Single, L As Single, tc As Single, tf As Single
Dim tp As Single, t As Single, tend As Single, h As Single

L = 10
ns = 5
dx = 2
k = 0.835
Te(0) = 100
Te(5) = 50
tc = 0.1
tf = 1
tp = 0.1
np = 0
tpr(np) = t
For i = 0 To ns
  Tepr(i, np) = Te(i)
Next i
Do
  tend = t + tp
  If tend > tf Then tend = tf
  h = tc
  Do
    If t + h > tend Then h = tend - t
    Call Derivs(Te, dTe, ns, dx, k)
    For j = 1 To ns - 1
      Te(j) = Te(j) + dTe(j) * h
    Next j
    t = t + h
    If t >= tend Then Exit Do
  Loop
  np = np + 1
  tpr(np) = t
  For j = 0 To ns
    Tepr(j, np) = Te(j)
  Next j
  If t >= tf Then Exit Do
Loop
Sheets("sheet1").Select
Range("a4").Select
For i = 0 To np
  ActiveCell.Value = tpr(i)
  For j = 0 To ns
    ActiveCell.Offset(0, 1).Select
    ActiveCell.Value = Tepr(j, i)
  Next j
```

```
  ActiveCell.Offset(1, -ns - 1).Select
Next i

End Sub

Sub Derivs(Te, dTe, ns, dx, k)

Dim j As Integer
For j = 1 To ns - 1
  dTe(j) = k * (Te(j - 1) - 2 * Te(j) + Te(j + 1)) / dx ^ 2
Next j
End Sub
```

30.9 This program is set up to either use Dirichlet or gradient boundary conditions depending on the values of the parameters *istrt* and *iend*.

```
Option Explicit

Sub EulerPDE()

Dim i As Integer, j As Integer, np As Integer, ns As Integer
Dim istrt As Integer, iend As Integer
Dim Te(20) As Single, dTe(20) As Single, tpr(200) As Single, Tepr(20,
200) As Single
Dim k As Single, dx As Single, L As Single, tc As Single, tf As Single
Dim tp As Single, t As Single, tend As Single, h As Single
Dim dTedx(20) As Single

L = 10
ns = 5
dx = 2
k = 0.835
dTedx(0) = 1
istrt = 0
dTedx(ns) = 0
iend = ns
Te(0) = 50
Te(1) = 50
Te(2) = 50
Te(3) = 50
Te(4) = 50
Te(5) = 50
tc = 0.1
tf = 1000
tp = 200
np = 0
tpr(np) = t
For i = 0 To ns
  Tepr(i, np) = Te(i)
Next i

Do
  tend = t + tp
  If tend > tf Then tend = tf
  h = tc
  Do
    If t + h > tend Then h = tend - t
    Call Derivs(Te(), dTe(), istrt, iend, ns, dx, k, dTedx())
    For j = istrt To iend
      Te(j) = Te(j) + dTe(j) * h
  Next j
    t = t + h
    If t >= tend Then Exit Do
  Loop
  np = np + 1
  tpr(np) = t
  For j = 0 To ns
    Tepr(j, np) = Te(j)
  Next j
  If t >= tf Then Exit Do
Loop
```

```
Sheets("sheet1").Select
Range("a4").Select
For i = 0 To np
  ActiveCell.Value = tpr(i)
  For j = 0 To ns
    ActiveCell.Offset(0, 1).Select
    ActiveCell.Value = Tepr(j, i)
  Next j
  ActiveCell.Offset(1, -ns - 1).Select
Next i

End Sub

Sub Derivs(Te, dTe, istrt, iend, ns, dx, k, dTedx)

Dim j As Integer
If istrt = 0 Then
  dTe(0) = k * (2 * Te(1) - 2 * Te(0) - 2 * dx * dTedx(0)) / dx ^ 2
End If
For j = 1 To ns - 1
  dTe(j) = k * (Te(j - 1) - 2 * Te(j) + Te(j + 1)) / dx ^ 2
Next j
If iend = ns Then
  dTe(ns) = k * (2 * Te(ns - 1) - 2 * Te(ns) + 2 * dx * dTedx(ns)) / dx
^ 2
End If

End Sub
```

30.10

```
Option Explicit

Sub SimpImplicit()

Dim np, ns, i, j, n
Dim Te(10), dTe(10), tpr(100), Tepr(10, 100), Tei As Single
Dim k, dx, L, tc, tf, tp, t, tend, h, lambda
Dim e(10), f(10), g(10), r(10), x(10), xrod

L = 10#
ns = 5
dx = L / ns
k = 0.835
Te(0) = 100#
Te(ns) = 50#
Tei = 0
For i = 1 To ns - 1
  Te(i) = Tei
Next i
t = 0
np = 0
tpr(np) = t
For i = 0 To ns
  Tepr(i, np) = Te(i)
Next i
tc = 0.1
tp = 0.1
tf = 1

Do
  tend = t + tp
  If tend > tf Then tend = tf
  h = tc
  Do
    If t + h > tend Then h = tend - t
    lambda = k * h / dx ^ 2
    f(1) = 1 + 2 * lambda
    g(1) = -lambda
    r(1) = Te(1) + lambda * Te(0)
    For j = 2 To ns - 2
      e(j) = -lambda
```

```
      f(j) = 1 + 2 * lambda
      g(j) = -lambda
      r(j) = Te(j)
    Next j
    e(ns - 1) = -lambda
    f(ns - 1) = 1 + 2 * lambda
    r(ns - 1) = Te(ns - 1) + lambda * Te(ns)
    Call Tridiag(e(), f(), g(), r(), Te(), ns - 1)
    t = t + h
    If t >= tend Then Exit Do
  Loop
  np = np + 1
  tpr(np) = t
  For j = 0 To ns
    Tepr(j, np) = Te(j)
  Next j
  If t >= tf Then Exit Do
Loop

Range("b5").Select
xrod = 0
For j = 0 To ns
  ActiveCell.Value = xrod
  ActiveCell.Offset(0, 1).Select
  xrod = xrod + dx
Next j
Range("a6").Select
For i = 0 To np
  ActiveCell.Value = "t = " & tpr(i)
  For j = 0 To ns
    ActiveCell.Offset(0, 1).Select
    ActiveCell.Value = Tepr(j, i)
  Next j
  ActiveCell.Offset(1, -ns - 1).Select
Next i
Range("a6").Select

End Sub


Sub Tridiag(e, f, g, r, x, n)

Call Decomp(e, f, g, n)
Call Substit(e, f, g, r, n, x)

End Sub

Sub Decomp(e, f, g, n)

Dim k As Integer
For k = 2 To n
  e(k) = e(k) / f(k - 1)
  f(k) = f(k) - e(k) * g(k - 1)
Next k
End Sub

Sub Substit(e, f, g, r, n, x)

Dim k As Integer
For k = 2 To n
  r(k) = r(k) - e(k) * r(k - 1)
Next k
x(n) = r(n) / f(n)
For k = n - 1 To 1 Step -1
  x(k) = (r(k) - g(k) * x(k + 1)) / f(k)
Next k
End Sub
```

30.11

```
Option Explicit
```

```
Sub CrankNic()

Dim np, ns, i, j, n
Dim Te(10), dTe(10), tpr(100), Tepr(10, 100), Tei As Single
Dim k, dx, L, tc, tf, tp, t, tend, h, lambda
Dim e(10), f(10), g(10), r(10), x(10), xrod

L = 10#
ns = 5
dx = L / ns
k = 0.835
Te(0) = 100#
Te(5) = 50#
Tei = 0
t = 0
np = 0
tpr(np) = t
For i = 0 To ns
  Tepr(i, np) = Tei
Next i
tc = 0.1
tf = 10#
tp = 1#

Do
  tend = t + tp
  If tend > tf Then tend = tf
  h = tc
  Do
    If t + h > tend Then h = tend - t
    lambda = k * h / dx ^ 2
    f(1) = 2 * (1 + lambda)
    g(1) = -lambda
    r(1) = lambda * Te(0) + 2 * (1 - lambda) * Te(1) + lambda * Te(2)
    r(1) = r(1) + lambda * Te(0)
    For j = 2 To ns - 2
      e(j) = -lambda
      f(j) = 2 * (1 + lambda)
      g(j) = -lambda
      r(j) = lambda * Te(j - 1) + 2 * (1 - lambda) * Te(j) + lambda * Te
(j + 1)
    Next j
    e(ns - 1) = -lambda
    f(ns - 1) = 2 * (1 + lambda)
     r(ns - 1) = lambda * Te(ns - 2) + 2 * (1 - lambda) * Te(ns - 1) +
lambda * Te(ns)
    r(ns - 1) = r(ns - 1) + lambda * Te(ns)
    Call Tridiag(e(), f(), g(), r(), Te(), ns - 1)
    t = t + h
    If t >= tend Then Exit Do
  Loop
  np = np + 1
  tpr(np) = t
  For j = 0 To ns
    Tepr(j, np) = Te(j)
  Next j
  If t >= tf Then Exit Do
Loop

Range("b5").Select
xrod = 0
For j = 0 To ns
  ActiveCell.Value = xrod
  ActiveCell.Offset(0, 1).Select
  xrod = xrod + dx
Next j
Range("a6").Select
For i = 0 To np
  ActiveCell.Value = "t = " & tpr(i)
  For j = 0 To ns
    ActiveCell.Offset(0, 1).Select
```

```
        ActiveCell.Value = Tepr(j, i)
      Next j
      ActiveCell.Offset(1, -ns - 1).Select
    Next i
    Range("a6").Select

    End Sub

    Sub Tridiag(e, f, g, r, x, n)

    Call Decomp(e, f, g, n)
    Call Substit(e, f, g, r, n, x)

    End Sub

    Sub Decomp(e, f, g, n)

    Dim k As Integer
    For k = 2 To n
      e(k) = e(k) / f(k - 1)
      f(k) = f(k) - e(k) * g(k - 1)
    Next k
    End Sub

    Sub Substit(e, f, g, r, n, x)

    Dim k As Integer
    For k = 2 To n
      r(k) = r(k) - e(k) * r(k - 1)
    Next k
    x(n) = r(n) / f(n)
    For k = n - 1 To 1 Step -1
      x(k) = (r(k) - g(k) * x(k + 1)) / f(k)
    Next k
    End Sub
```

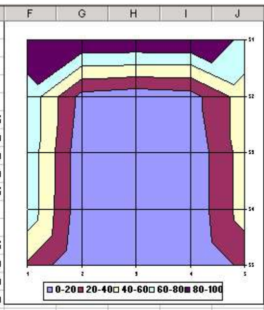
30.12 Here is VBA code to solve this problem. The Excel output is also attached showing values for the first two steps along with selected snapshots of the solution as it evolves in time.

```
Option Explicit

Sub ADI()
Dim np As Integer, i As Integer, j As Integer
Dim nx As Integer, ny As Integer
Dim Lx As Single, dx As Single
Dim Ly As Single, dy As Single
Dim Te(10, 10) As Single, dTe(10, 10) As Single
Dim tpr(100) As Single, Tepr(10, 10, 100) As Single, Tei As Single
Dim k As Single
Dim dt As Single, ti As Single, tf As Single, tp As Single
Dim t As Single, tend As Single, h As Single
Dim lamx As Single, lamy As Single
Dim e(10) As Single, f(10) As Single, g(10) As Single, r(10) As Single, Ted(10) As
    Single
'set computation parameters
Lx = 40
nx = 4
dx = Lx / nx
Ly = 40
ny = 4
dy = Ly / ny
k = 0.835
dt = 10
tf = 500
ti = 0
tp = 10
Tei = 0
'set top boundary
For i = 1 To nx - 1
  Te(i, ny) = 100
Next i
'set bottom boundary
For i = 1 To nx - 1
  Te(i, 0) = 0
```

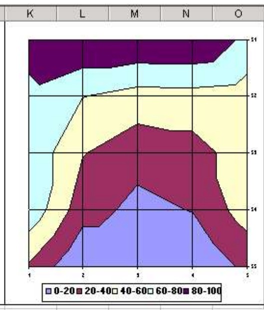
```
Next i
'set left boundary
For j = 1 To ny - 1
  Te(0, j) = 75
Next j
'set right boundary
For j = 1 To ny - 1
  Te(nx, j) = 50
Next j
'set corners for plot
Te(0, 0) = (dy * Te(1, 0) + dx * Te(0, 1)) / (dy + dx)
Te(nx, 0) = (dy * Te(nx - 1, 0) + dx * Te(nx, 1)) / (dy + dx)
Te(0, ny) = (dy * Te(1, ny) + dx * Te(0, ny - 1)) / (dy + dx)
Te(nx, ny) = (dy * Te(nx - 1, ny) + dx * Te(nx, ny - 1)) / (dy + dx)
'set interior
For i = 1 To nx - 1
  For j = 1 To ny - 1
    Te(i, j) = Tei
  Next j
Next i
'save initial values for output
np = 0
t = ti
tpr(np) = t
For i = 0 To nx
  For j = 0 To ny
    Tepr(i, j, np) = Te(i, j)
  Next j
Next i
Do
  tend = t + tp
  If tend > tf Then tend = tf
  h = dt
  Do
    If t + h > tend Then h = tend - t
    'Sweep y
    lamx = k * h / dx ^ 2
    lamy = k * h / dy ^ 2
    For i = 1 To nx - 1
      f(1) = 2 * (1 + lamy)
      g(1) = -lamy
      r(1) = lamx * Te(i - 1, 1) + 2 * (1 - lamx) * Te(i, 1) + lamx * Te(i + 1, 1) _
            + lamy * Te(i, 0)
      For j = 2 To ny - 2
        e(j) = -lamy
        f(j) = 2 * (1 + lamy)
        g(j) = -lamy
        r(j) = lamx * Te(i - 1, j) + 2 * (1 - lamx) * Te(i, j) + lamx * Te(i + 1, j)
      Next j
      e(ny - 1) = -lamy
      f(ny - 1) = 2 * (1 + lamy)
      r(ny - 1) = lamx * Te(i - 1, ny - 1) + 2 * (1 - lamx) * Te(i, ny - 1) _
                                + lamx * Te(i + 1, ny - 1) + lamy * Te(i, nx)
      Call Tridiag(e(), f(), g(), r(), Ted(), nx - 1)
      For j = 1 To ny - 1
        Te(i, j) = Ted(j)
      Next j
    Next i
    t = t + h / 2
    'Sweep x
    For j = 1 To ny - 1
      f(1) = 2 * (1 + lamx)
      g(1) = -lamx
      r(1) = lamy * Te(1, j - 1) + 2 * (1 - lamy) * Te(1, j) + lamy * Te(1, j + 1) _
                                                              + lamx * Te(0, j)
      For i = 2 To nx - 2
        e(i) = -lamx
        f(i) = 2 * (1 + lamx)
        g(i) = -lamx
        r(i) = lamy * Te(i, j - 1) + 2 * (1 - lamy) * Te(i, j) + lamy * Te(i, j + 1)
      Next i
      e(nx - 1) = -lamx
      f(nx - 1) = 2 * (1 + lamx)
      r(nx - 1) = lamy * Te(nx - 1, j - 1) + 2 * (1 - lamy) * Te(nx - 1, j) _
                                + lamy * Te(nx - 1, j + 1) + lamx * Te(ny, j)
      Call Tridiag(e(), f(), g(), r(), Ted(), nx - 1)
      For i = 1 To nx - 1
        Te(i, j) = Ted(i)
      Next i
    Next j
    t = t + h / 2
    If t >= tend Then Exit Do
```

```
  Loop
  'save values for output
  np = np + 1
  tpr(np) = t
  For i = 0 To nx
    For j = 0 To ny
      Tepr(i, j, np) = Te(i, j)
    Next j
  Next i
  If t >= tf Then Exit Do
Loop
'output results back to sheet
Range("a5").Select
Range("a5:e2005").ClearContents
For k = 0 To np
  ActiveCell.Value = "t = " & tpr(k)
  ActiveCell.Offset(1, 0).Select
  For j = ny To 0 Step -1
    For i = 0 To nx
      ActiveCell.Value = Tepr(i, j, k)
      ActiveCell.Offset(0, 1).Select
    Next i
    ActiveCell.Offset(1, -nx - 1).Select
  Next j
  ActiveCell.Offset(1, 0).Select
Next k
Range("a5").Select
End Sub

Sub Tridiag(e, f, g, r, x, n)
Call Decomp(e, f, g, n)
Call Substit(e, f, g, r, n, x)
End Sub

Sub Decomp(e, f, g, n)
Dim k As Integer
For k = 2 To n
  e(k) = e(k) / f(k - 1)
  f(k) = f(k) - e(k) * g(k - 1)
Next k
End Sub

Sub Substit(e, f, g, r, n, x)
Dim k As Integer
For k = 2 To n
  r(k) = r(k) - e(k) * r(k - 1)
Next k
x(n) = r(n) / f(n)
For k = n - 1 To 1 Step -1
  x(k) = (r(k) - g(k) * x(k + 1)) / f(k)
Next k
End Sub
```

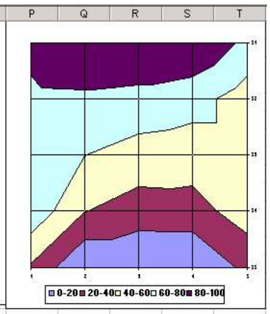
| | A | B | C | D | E |
|---|---|---|---|---|---|
| 1 | Prob. 30.12 | | | | |
| 2 | | | | RUN | |
| 3 | | | | | |
| 4 | | | | | |
| 5 | t = 0 | | | | |
| 6 | 87.5 | 100 | 100 | 100 | 75 |
| 7 | 75 | 0 | 0 | 0 | 50 |
| 8 | 75 | 0 | 0 | 0 | 50 |
| 9 | 75 | 0 | 0 | 0 | 50 |
| 10 | 37.5 | 0 | 0 | 0 | 25 |
| 11 | | | | | |
| 12 | t = 10 | | | | |
| 13 | 87.5 | 100 | 100 | 100 | 75 |
| 14 | 75 | 13.112 | 8.3207 | 11.361 | 50 |
| 15 | 75 | 6.1683 | 0.8238 | 4.2359 | 50 |
| 16 | 75 | 5.58552 | 0.4782 | 3.7388 | 50 |
| 17 | 37.5 | 0 | 0 | 0 | 25 |
| 18 | | | | | |
| 19 | t = 20 | | | | |
| 20 | 87.5 | 100 | 100 | 100 | 75 |
| 21 | 75 | 23.467 | 15.889 | 20.334 | 50 |
| 22 | 75 | 11.8449 | 2.7523 | 8.3305 | 50 |
| 23 | 75 | 10.0556 | 1.4948 | 6.7745 | 50 |
| 24 | 37.5 | 0 | 0 | 0 | 25 |



t = 10 s



t = 100 s



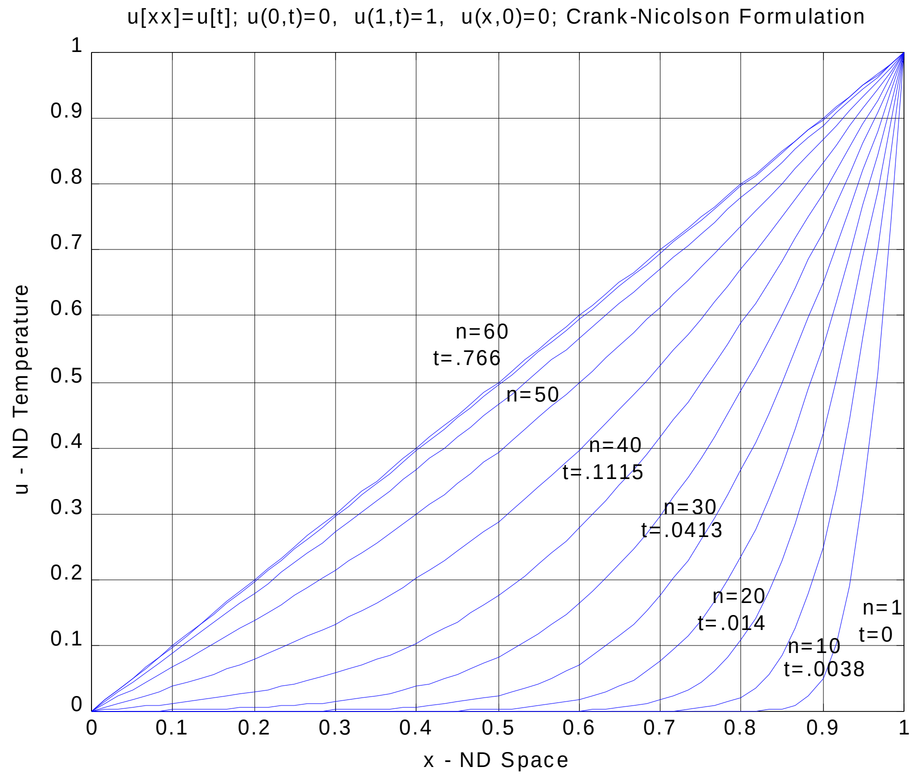
t = 300 s

## 30.13 MATLAB solution:

```
%PDE Parabolic Problem - Heat conduction in a rod
%      u[xx]=u[t]
%     BC u(0,t)=0   u(1,t)=1
%     IC  u(x,0)=0    x<1
```

```
%           i=spatial index, from 1 to imax
%                  imax = no. of x points
%           n=time index  from 1 to nmax
%                  nmax = no. of time steps,
%    Crank-Nicolson Formulation
     imax=61;
     nmax=60;      % last time step = nmax+1
%   Constants
     dx=1/(imax-1);
     dx2=dx*dx;
     dt=dx2;    % Setting dt to dx2 for good stability and results

%   Independent space variable
     x=0:dx:1;
%   Sizing matrices
     u=zeros(imax,nmax+1); t=zeros(1,nmax+1);
     a=zeros(1,imax);  b=zeros(1,imax);
      c=zeros(1,imax);   d=zeros(1,imax);
     ba=zeros(1,imax); ga=zeros(1,imax);
     up=zeros(1,imax);

%   Boundary Conditions
        u(1,1)=0;
        u(imax,1)=1;

%   Time step loop
%   n=1 represents 0 time, n+1 = next time step
     t(1)=0;
     for n=1:nmax
        t(n+1)=t(n)+dt;

% Boundary conditions & Constants
     u(1,n+1)=0;
     u(imax,n+1)=1;
     dx2dt=dx2/dt;

% coefficients
        b(2)=-2-2*dx2dt;
        c(2)=1;
        d(2)=(2-2*dx2dt)*u(2,n)-u(3,n);
           for i=3:imax-2
            a(i)=1;
            b(i)=-2-2*dx2dt;
            c(i)=1;
            d(i)=-u(i-1,n)+(2-2*dx2dt)*u(i,n)-u(i+1,n);
           end
               a(imax-1)=1;
               b(imax-1)=-2-2*dx2dt;
               d(imax-1)=-u(imax-2,n)+(2-2*dx2dt)*u(imax-1,n)-2;
% Solution by Thomas Algorithm

ba(2)=b(2);
ga(2)=d(2)/b(2);
for i=3:imax-1
    ba(i)=b(i)-a(i)*c(i-1)/ba(i-1);
    ga(i)=(d(i)-a(i)*ga(i-1))/ba(i);
end


% Back substitution step
u(imax-1,n+1)=ga(imax-1);
for i=imax-2:-1:2
    u(i,n+1)=ga(i)-c(i)*u(i+1,n+1)/ba(i);
end
dt=1.1*dt;
end
% end of time step loop


% Plot
% Storing plot value of u as up, at every 5 time steps, np=5
%j=time index
```

```
%i=space index

np=5;
for j=np:np:nmax
for i=1:imax
  up(i)=u(i,j);
end
plot(x,up)
hold on
end
grid
title('u[xx]=u[t]; u(0,t)=0,  u(1,t)=1,  u(x,0)=0; Crank-Nicolson
Formulation')
xlabel('x - ND Space')
ylabel('u - ND Temperature')
hold off

% Storing times for temp. profiles
%These can be saved in a data file or examined in the command file
tp=zeros(1,(nmax-1)/np);
i=1;
tp(1)=0;
for k=np:np:nmax
i=i+1;
tp(i)=t(k);
end
tp

                     gtext('n=60');gtext('n=50');gtext('n=40');gtext
('n=30');                    gtext('n=20');gtext('n=10');gtext('n=1');
gtext('t=.766');
gtext('t=.1115');gtext('t=.0413');gtext('t=.014');
gtext('t=.0038');gtext('t=0')
```




```
tp =

  Columns 1 through 7

         0    0.0013    0.0038    0.0078    0.0142    0.0246    0.0413
```

```
   Columns 8 through 13

     0.0682    0.1115    0.1813    0.2937    0.4746    0.7661
```

30.14

$$\frac{\partial^2 u}{\partial r^2}+\frac{1}{r}\frac{\partial u}{\partial r}=\frac{\partial u}{\partial t}$$

Substituting of second order correct Crank-Nicolson analogues

$$\frac{\partial^2 u}{\partial r^2}=\frac{1}{2}\left[\frac{u_{i+1,n+1}-u_{i,n+1}+u_{i-1,n+1}}{\Delta r^2}+\frac{u_{i+1,n}-u_{i,n}+u_{i-1,n}}{\Delta r^2}\right]$$
$$\frac{\partial u}{\partial r}=\frac{1}{2}\left[\frac{u_{i+1,n+1}-u_{i-1,n+1}}{2\Delta r}+\frac{u_{i+1,n}-u_{i-1,n}}{2\Delta r}\right]$$
$$r=(i-1)\Delta r$$
$$\frac{\partial u}{\partial t}=\frac{u_{i,n+1}-u_{i,n}}{\Delta t}$$

into the governing equation give the following finite difference equations:

$$\left[1-\frac{1}{2(i-1)}\right]u_{i-1,n+1}+\left[-2-2\frac{\Delta r^2}{\Delta t}\right]u_{i,n+1}+\left[1+\frac{1}{2(i-1)}\right]u_{i+1,n+1}=\left[-1+\frac{1}{2(i-1)}\right]u_{i-1,n}$$
$$-\left[2-2\frac{\Delta r^2}{\Delta t}\right]u_{i,n}+\left[-1+\frac{1}{2(i-1)}\right]u_{i+1,n}$$

For the end points:

$x = 1$ ($i$ = R), substitute the value of $u_R = 1$ into the above FD equation
$x = 0$ ($i = 1$), set the FD analog to the first derivative = 0

$$\left[\frac{\partial u}{\partial r}\right]_{i=1}=\frac{1}{2}\left[\frac{u_{2,n+1}-u_{0,n+1}}{2\Delta r}+\frac{u_{2,n}-u_{0,n}}{2\Delta r}\right]=0$$

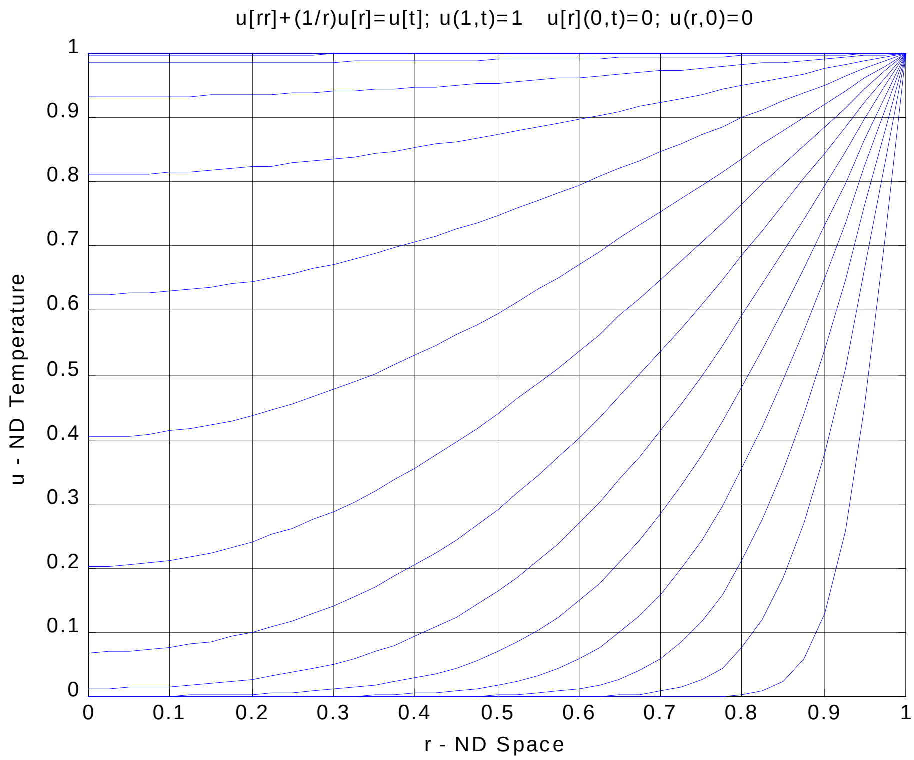
Also substitute in $i = 1$ into the finite difference equation and algebraically eliminate $u_{0,n+1}+u_{0,n}$ from the two equations and get the FD equation at $i = 1$:

$$\left[-2-2\frac{\Delta r^2}{\Delta t}\right]u_{1,n+1}+[2]u_{2,n+1}=-\left[2-2\frac{\Delta r^2}{\Delta t}\right]u_{1,n}+[-2]u_{2,n}$$

```
%PDE Parabolic Problem - Heat conduction in the radial direction in a
circular rod
%     u[rr]+(1/r)u[r]=u[t]        0<r<1
%      BC   u(1,t)=1   u[r](0,t)=0
%      IC  u(r,0)=0           0<r<1
%           i=spatial index, from 1 to imax
%            imax = no. of r points (imax=21 for 20 dr spaces)
%           n=time index  from 1 to nmax
%            nmax = no. of time steps,
%      Crank-Nicolson Formulation
            imax=41;
            nmax=60;               % last time step = nmax+1
```

```
%  Constants
            dr=1/(imax-1);
            dr2=dr*dr;
            dt=dr2;   % Setting dt to dr2 for good stability and results

%  Independent space variable
            r=0:dr:1;

%  Sizing matrices
            u=zeros(imax,nmax+1); t=zeros(1,nmax+1);
            a=zeros(1,imax);  b=zeros(1,imax);
           c=zeros(1,imax);  d=zeros(1,imax);
            ba=zeros(1,imax); ga=zeros(1,imax);
            up=zeros(1,imax);

%  Boundary Conditions
            u(imax,1)=1;

%  Time step loop
%  n=1 represents 0 time, new time = n+1
            t(1)=0;
            for n=1:nmax
                t(n+1)=t(n)+dt;

%   Boundary conditions & Constants
                    u(imax,n+1)=1;
                    dr2dt=dr2/dt;

% coefficients
                    b(1)=-2-2*dr2dt;
                    c(1)=2;
                    d(1)=(2-2*dr2dt)*u(1,n)-2*u(2,n);
                    for i=2:imax-2
                              a(i)=1-1/(2*(i-1));
                              b(i)=-2-2*dr2dt;
                              c(i)=1+1/(2*(i-1));
                              d(i)=(-1+1/(2*(i-1)))*u(i-1,n)+(2-
2*dr2dt)*u(i,n)+(-1-1/(2*(i-1)))*u(i+1,n);
                    end
                             a(imax-1)=1-1/(2*(imax-2));
                             b(imax-1)=-2-2*dr2dt;
                             d(imax-1)=(-1+1/(2*(imax-2)))*u(imax-2,n)+
(2-2*dr2dt)*u(imax-1,n)-2*(1+1/(2*(imax-2)))
% Solution by Thomas Algorithm
                          ba(1)=b(1);
                          ga(1)=d(1)/b(1);
                          for i=2:imax-1
                              ba(i)=b(i)-a(i)*c(i-1)/ba(i-1);
                              ga(i)=(d(i)-a(i)*ga(i-1))/ba(i);
                          end

% Back substitution step
                          u(imax-1,n+1)=ga(imax-1);
                          for i=imax-2:-1:1
                                u(i,n+1)=ga(i)-c(i)*u(i+1,n+1)/ba(i);
                          end
                          dt=1.1*dt;
                       end
% end of time step loop


% Plot
% Storing plot value of u as up, at every 5 time steps
%j=time index
%i=space index
                       istart=4;
                       for j=istart:istart:nmax+1
                                 for i=1:imax
                                     up(i)=u(i,j);
                                 end
                                 plot(r,up)
```

```
                    hold on
          end
          grid

title('u[rr]+(1/r)u[r]=u[t]; u(1,t)=1   u[r](0,t)=0; u(r,0)=0')
xlabel('r - ND Space')
ylabel('u - ND Temperature')
hold off

% Storing times for temp. profiles
%    These can be saved in a data file or examined in the command file
          tp=zeros(1,(nmax-1)/istart);
          i=1;
          tp(1)=0;
          for k=istart:istart:nmax+1
                    i=i+1;
                    tp(i)=t(k);
          end
                    tp
```




```
tp =

  Columns 1 through 7

         0    0.0021    0.0059    0.0116    0.0199    0.0320    0.0497

  Columns 8 through 14

    0.0757    0.1137    0.1694    0.2509    0.3703    0.5450    0.8008

  Columns 15 through 16

    1.1754    1.7238
```

30.15

$$\frac{\partial^2 u}{dx^2}+b\frac{\partial u}{\partial x}=\frac{\partial u}{\partial t}$$

Substituting of second order correct Crank-Nicolson analogues

$$\frac{\partial^2 u}{\partial x^2}=\frac{1}{2}\left[\frac{u_{i+1,n+1}-u_{i,n+1}+u_{i-1,n+1}}{\Delta x^2}+\frac{u_{i+1,n}-u_{i,n}+u_{i-1,n}}{\Delta x^2}\right]$$

$$\frac{\partial u}{\partial x}=\frac{1}{2}\left[\frac{u_{i+1,n+1}-u_{i-1,n+1}}{2\Delta x}+\frac{u_{i+1,n}-u_{i-1,n}}{2\Delta x}\right]$$

$$\frac{\partial u}{\partial t}=\frac{u_{i,n+1}-u_{i,n}}{\Delta t}$$

into the governing equation give the following finite difference equations

$$\left[1-\frac{1}{2}b\Delta x\right]u_{i-1,n+1}+\left[-2-2\frac{\Delta x^2}{\Delta t}\right]u_{i,n+1}+\left[1+\frac{1}{2}b\Delta x\right]u_{i+1,n+1}=\left[-1+\frac{1}{2}b\Delta x\right]u_{i-1,n}$$
$$+\left[2-2\frac{\Delta x^2}{\Delta t}\right]u_{i,n}+\left[-1-\frac{1}{2}b\Delta x\right]u_{i+1,n}$$

```
%PDE Parabolic Problem with a dispersion term
%      u[xx]+bu[x]=u[t]
%       BC u(0,t)=0  u(1,t)=1
%       IC  u(x,0)=0   x<1
%           i=spatial index, from 1 to imax
%              imax = no. of spatial points (imax=21 for 20 dx spaces)
%           n=time index, from 1 to nmax
%              nmax = no. of time steps
%       Crank-Nicholson formulation for the spatial derivatives
       imax=61;
       nmax=60;     % last time step = nmax+1

% constants
       dx=1/(imax-1);
       dx2=dx*dx;
       dt=dx2;

% Parameters
       B=-4;

% Independent spatial variable
       x=0:dx:1;


% Sizing matrices
       u=zeros(imax,nmax); t=zeros(1,nmax);
       a=zeros(1,imax);  b=zeros(1,imax);
      c=zeros(1,imax);  d=zeros(1,imax);
       ba=zeros(1,imax);  ga=zeros(1,imax);
       up=zeros(1,imax);

% Boundary Conditions
       u(1,1)=0;
       u(imax,1)=1;


%   Time step loop
%  n=1 represents 0 time, new time = n+1
       t(1)=0;
       for n=1:nmax
```

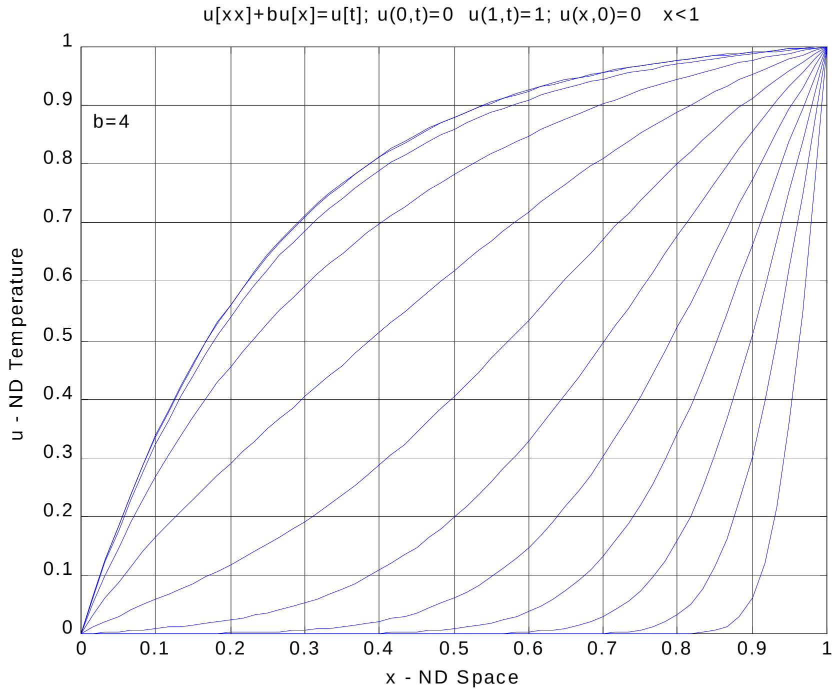
```
          t(n+1)=t(n)+dt;

%  Boundary conditions & constants
       u(1,n+1)=0;
       u(imax,n+1)=1;
       dx2dt=dx2/dt;

% Coefficients
       b(2)=-2-2*dx2dt;
       c(2)=1+0.5*B*dx;
       d(2)=(-1-0.5*B*dx)*u(3,n)+(2-2*dx2dt)*u(2,n);
       for i=3:imax-2
           a(i)=1-0.5*B*dx;
           b(i)=-2-2*dx2dt;
           c(i)=1+0.5*B*dx;
           d(i)=(-1-0.5*B*dx)*u(i+1,n)+(2-2*dx2dt)*u(i,n)+(-1+0.5*B*dx)*u
(i-1,n);
      end

a(imax-1)=1-0.5*B*dx;
b(imax-1)=-2-2*dx2dt;
d(imax-1)=2*(-1-0.5*B*dx)+(2-2*dx2dt)*u(imax-1,n)+(-1+0.5*B*dx)*u(imax-
2,n);

% Solution by Thomas Algorithm
       ba(2)=b(2);
       ga(2)=d(2)/b(2);
       for i=3:imax-1
           ba(i)=b(i)-a(i)*c(i-1)/ba(i-1);
           ga(i)=(d(i)-a(i)*ga(i-1))/ba(i);
       end

% Back substitution step
       u(imax-1,n+1)=ga(imax-1);
       for i=imax-2:-1:2
           u(i,n+1)=ga(i)-c(i)*u(i+1,n+1)/ba(i);
       end
       dt=1.1*dt;
       end
% End of time step loop


%Plot
%Storing plot value of u as up, at ever 5 time steps
%       j=time index
%       i=speace index
                for j=5:5:nmax
                   for i=1:imax
                       up(i)=u(i,j);
                   end
                      plot(x,up)
                      hold on
                end
                grid
title('u[xx]+bu[x]=u[t]; u(0,t)=0  u(1,t)=1; u(x,0)=0   x<1')
xlabel('x - ND Space')
ylabel('u - ND Temperature')
hold off
gtext('b=-4')
% Storing times for temp. profiles
%   These can be used in a data file or examined in the command file
                tp=zeros(1,(nmax-1)/5);
                i=1;
                tp(1)=0;
                for k=5:5:nmax
                          i=i+1;
                          tp(i)=t(k);
                end
                tp


tp =
```

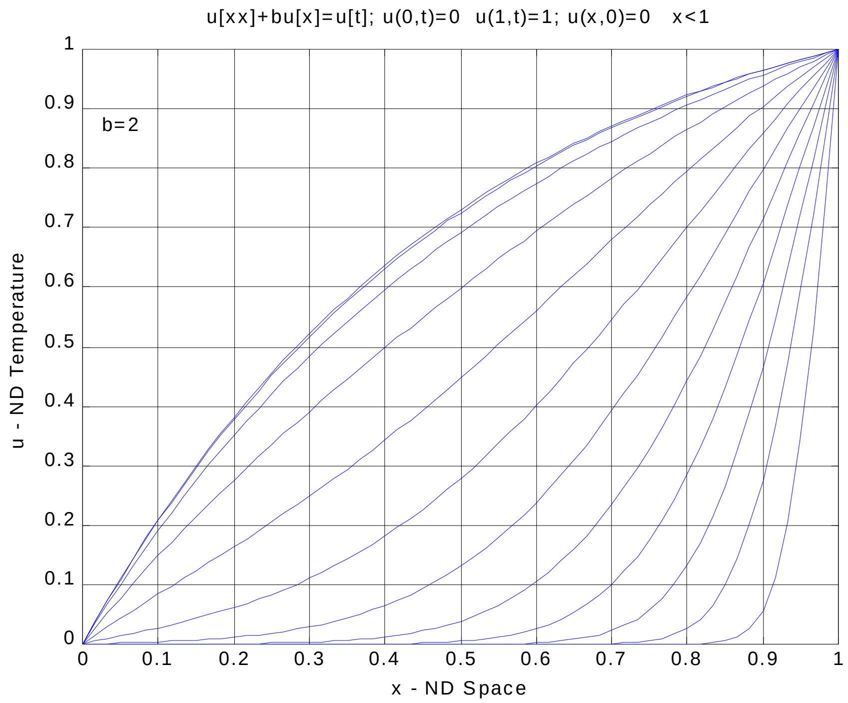
Columns 1 through 7

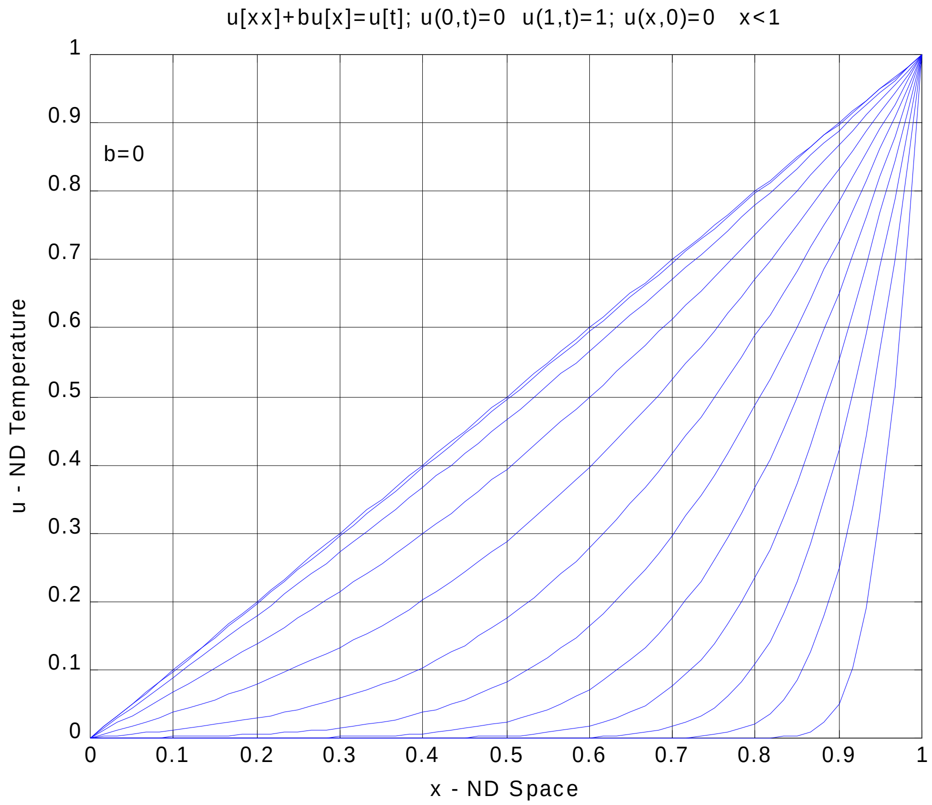
| | | | | | | |
|---|---|---|---|---|---|---|
| 0 | 0.0013 | 0.0038 | 0.0078 | 0.0142 | 0.0246 | 0.0413 |

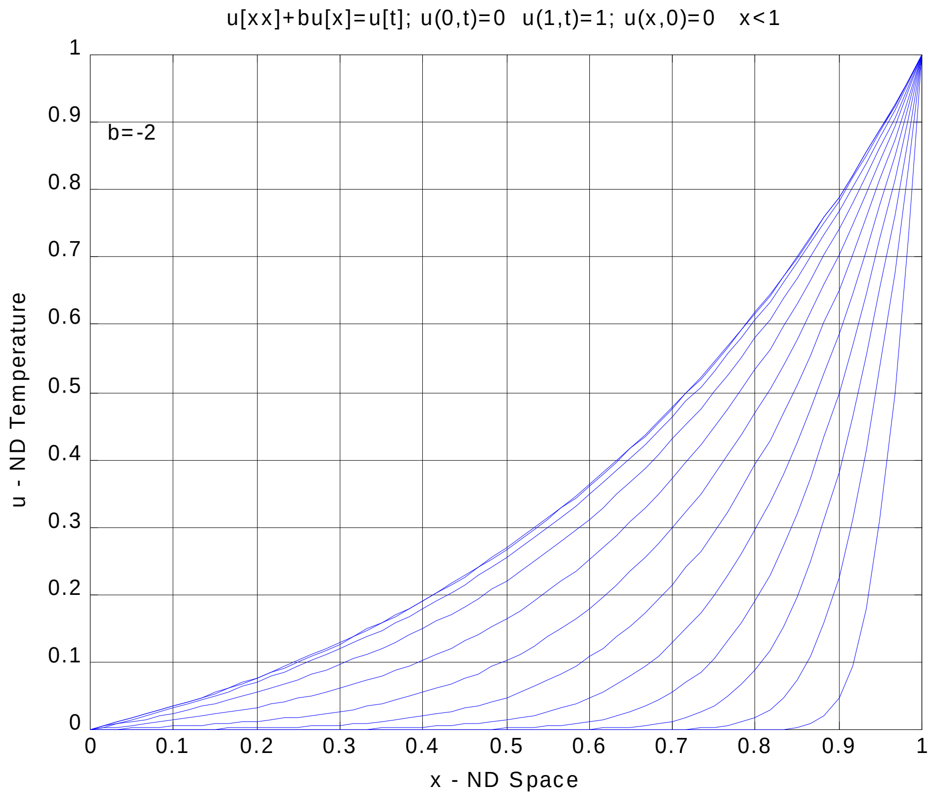
Columns 8 through 13

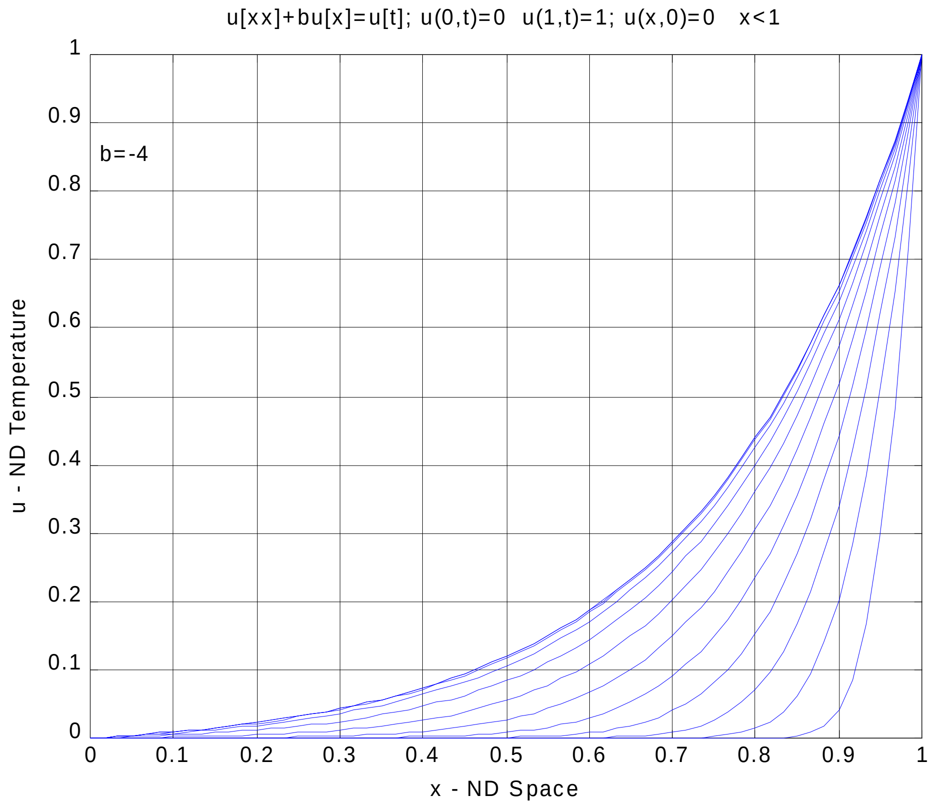
| | | | | | |
|---|---|---|---|---|---|
| 0.0682 | 0.1115 | 0.1813 | 0.2937 | 0.4746 | 0.7661 |




u[xx]+bu[x]=u[t]; u(0,t)=0 u(1,t)=1; u(x,0)=0 x<1
b=0
u - ND Temperature
x - ND Space
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1


u[xx]+bu[x]=u[t]; u(0,t)=0 u(1,t)=1; u(x,0)=0 x<1
b=-2
u - ND Temperature
x - ND Space
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1


u[xx]+bu[x]=u[t]; u(0,t)=0 u(1,t)=1; u(x,0)=0 x<1
b=-4
u - ND Temperature
x - ND Space
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

# CHAPTER 31

31.1 The equation to be solved is

$$\frac{d^2T}{dx^2} = -20$$

Assume a solution of the form $T = ax^2 + bx + c$ which can be differentiated twice to give $T'' = 2a$. Substituting this result into the differential equation gives $a = -10$. The boundary conditions can be used to evaluate the remaining coefficients. For the first condition at $x = 0$,

$$50 = -10(0)^2 + b(0) + c$$

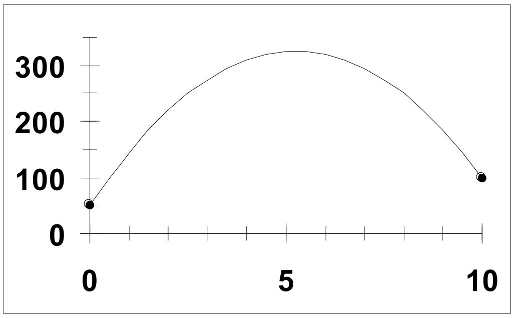
or $c = 50$. Similarly, for the second condition.

$$100 = -10(10)^2 + b(10) + 50$$

which can be solved for $b = 105$. Therefore, the final solution is

$$T = -10x^2 + 105x + 50$$

The results are plotted in Fig. 31.5.




31.2 The heat source term in the first row of Eq. (31.26) can be evaluated by substituting Eq. (31.3) and integrating to give

$$\int_0^{2.5} 20 \frac{2.5 - x}{2.5} dx = 25$$

Similarly, Eq. (31.4) can be substituted into the heat source term of the second row of Eq. (31.26), which can also be integrated to yield

$$\int_0^{2.5} 20 \frac{x - 0}{2.5} dx = 25$$

These results along with the other parameter values can be substituted into Eq. (31.26) to give

$$0.4T_1 - 0.4T_2 = -\frac{dT}{dx}(x_1) + 25$$

and

$$-0.4T_1 + 0.4T_2 = \frac{dT}{dx}(x_2) + 25$$

31.3 In a manner similar to Fig. 31.7, the equations can be assembled for the total system,

$$\begin{bmatrix} 0.4 & -0.4 & & & \\ -0.4 & 0.8 & -0.4 & & \\ & -0.4 & 0.8 & -0.4 & \\ & & -0.4 & 0.8 & -0.4 \\ & & & -0.4 & 0.4 \end{bmatrix} \begin{Bmatrix} T_1 \\ T_2 \\ T_3 \\ T_4 \\ T_5 \end{Bmatrix} = \begin{Bmatrix} -dT(x_1)/dx + 25 \\ 50 \\ 50 \\ 50 \\ dT(x_1)/dx + 25 \end{Bmatrix}$$

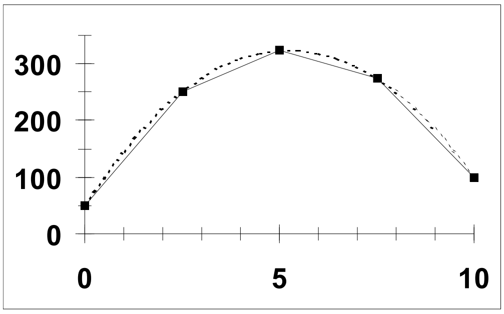
The unknown end temperatures can be substituted to give

$$\begin{bmatrix} 1 & -0.4 & & & \\ & 0.8 & -0.4 & & \\ & -0.4 & 0.8 & -0.4 & \\ & & -0.4 & 0.8 & \\ & & & -0.4 & -1 \end{bmatrix} \begin{Bmatrix} dT(x_1)/dx \\ T_2 \\ T_3 \\ T_4 \\ -dT(x_5)/dx \end{Bmatrix} = \begin{Bmatrix} 5 \\ 70 \\ 50 \\ 90 \\ -15 \end{Bmatrix}$$

These equations can be solved for

$$\begin{Bmatrix} dT(x_1)/dx \\ T_2 \\ T_3 \\ T_4 \\ -dT(x_5)/dx \end{Bmatrix} = \begin{Bmatrix} 105 \\ 250 \\ 325 \\ 275 \\ -95 \end{Bmatrix}$$

The solution, along with the analytical solution (dashed line) is shown below:




31.4

$$0 = D\frac{d^2c}{dx^2} - U\frac{dc}{dx} - kc$$

$$R = D\frac{d^2\tilde{c}}{dx^2} - U\frac{d\tilde{c}}{dx} - k\tilde{c}$$

$$\int_{x_1}^{x_2} \left[ D\frac{d^2\tilde{c}}{dx^2} - U\frac{d\tilde{c}}{dx} - k\tilde{c} \right] N_i \, dx$$

$$D\int_{x_1}^{x_2} \frac{d^2\widetilde{c}}{dx^2} N_i(x)dx \qquad \textbf{(1)}$$

$$-U\int_{x_1}^{x_2} \frac{d\widetilde{c}}{dx} N_i(x)dx \qquad \textbf{(2)}$$

$$-k\int_{x_1}^{x_2} \widetilde{c} N_i(x)dx \qquad \textbf{(3)}$$

Term (1):

$$D\int_{x_1}^{x_2} \frac{d^2\widetilde{c}}{dx^2} N_i(x)dx = D\begin{Bmatrix} -\dfrac{dc}{dx}(x_1) - \dfrac{c_1 - c_2}{x_2 - x_1} \\ \dfrac{dc}{dx}(x_2) - \dfrac{c_2 - c_1}{x_2 - x_1} \end{Bmatrix}$$

Term (2):

$$\int_{x_1}^{x_2} \frac{d\widetilde{c}}{dx} N_i(x)dx = \int_{x_1}^{x_2} \frac{c_2 - c_1}{x_2 - x_1} N_i(x)dx$$

$$\int_{x_1}^{x_2} N_i(x)dx = \frac{x_2 - x_1}{2}$$

$$\therefore \int_{x_1}^{x_2} \frac{d\widetilde{c}}{dx} N_i(x)dx = \frac{c_2 - c_1}{2}$$

$$-U\int_{x_1}^{x_2} \frac{d\widetilde{c}}{dx} N_i(x)dx = -U\begin{Bmatrix} \dfrac{c_2 - c_1}{2} \\ \dfrac{c_2 - c_1}{2} \end{Bmatrix}$$

Term (3):

$$-k\int_{x_1}^{x_2} \widetilde{c} N_i(x)dx = -\frac{k(x_2 - x_1)}{2}\begin{Bmatrix} c_1 \\ c_2 \end{Bmatrix}$$

Total element equation [(1) + (2) + (3)]

$$\begin{bmatrix} a_{11} & a_{11} \\ a_{11} & a_{11} \end{bmatrix}\begin{Bmatrix} c_1 \\ c_2 \end{Bmatrix} = \begin{Bmatrix} b_1 \\ b_2 \end{Bmatrix}$$

where

$$a_{11} = \frac{D}{x_2 - x_1} - \frac{U}{2} + \frac{k}{2}(x_2 - x_1) \qquad a_{12} = \frac{-D}{x_2 - x_1} + \frac{U}{2} \qquad a_{21} = \frac{-D}{x_2 - x_1} - \frac{U}{2}$$

$$a_{22} = -\frac{D}{x_2 - x_1} + \frac{U}{2} + \frac{k}{2}(x_2 - x_1)$$

$$b_1 = -D\frac{dc}{dx}(x_1) \qquad\qquad b_2 = D\frac{dc}{dx}(x_2)$$

31.5 First we can develop an analytical function for comparison. Substituting parameters gives

$$1.5\times 10^8 \frac{d^2u}{dx^2} = 50$$

Assume a solution of the form

$$u = ax^2 + bx + c$$

This can be differentiated twice to yield $d^2u/dx^2 = 2a$. This can be substituted into the ODE, which can then be solved for $a = 1.6667\times10^{-7}$. The boundary conditions can then be used to evaluate the remaining coefficients. At the left side, $u(0) = 0$ and

$$0 = 1.6667\times10^{-7}(0)^2 + b(0) + c$$

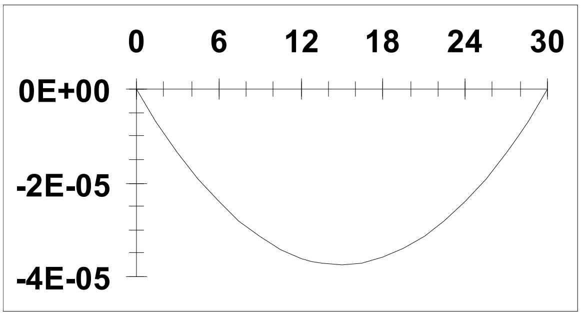
and therefore, $c = 0$. At the right side of the bar, $u(30) = 0$ and

$$0 = 1.6667\times10^{-7}(30)^2 + b(30)$$

and therefore, $b = -5\times10^{-6}$, and the solution is

$$u = 1.6667\times10^{-7}x^2 - 5\times10^{-6}x$$

which can be displayed as




The element equation can be written as

$$\frac{A_cE}{x_2 - x_1}\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}\begin{Bmatrix} u_1 \\ u_2 \end{Bmatrix} = A_cE\begin{Bmatrix} -\dfrac{du}{dx}(x_1) \\ \dfrac{du}{dx}(x_2) \end{Bmatrix} + \begin{Bmatrix} \int_{x_1}^{x_2} P(x)N_1(x)dx \\ \int_{x_1}^{x_2} P(x)N_2(x)dx \end{Bmatrix}$$

The distributed load can be evaluated as

$$\int_0^6 -50\frac{6-x}{6}dx = -150 \qquad\qquad \int_0^6 -50\frac{x-0}{6}dx = -150$$

Thus, the final element equation is

$$\begin{bmatrix} 2.5\times10^7 & -2.5\times10^7 \\ -2.5\times10^7 & 2.5\times10^7 \end{bmatrix} \begin{Bmatrix} u_1 \\ u_2 \end{Bmatrix} = A_c E \begin{Bmatrix} -\dfrac{du}{dx}(x_1) \\ \dfrac{du}{dx}(x_2) \end{Bmatrix} + \begin{Bmatrix} -150 \\ -150 \end{Bmatrix}$$

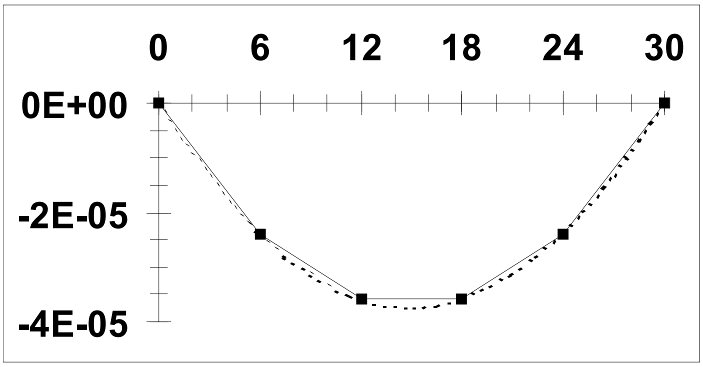
Assembly yields

$$\begin{bmatrix} 1.5\times10^8 & -2.5\times10^7 & & & & \\ & 5\times10^7 & -2.5\times10^7 & & & \\ & -2.5\times10^7 & 5\times10^7 & -2.5\times10^7 & & \\ & & -2.5\times10^7 & 5\times10^7 & -2.5\times10^7 & \\ & & & -2.5\times10^7 & 5\times10^7 & \\ & & & & -2.5\times10^7 & -1.5\times10^7 \end{bmatrix} \begin{Bmatrix} \dfrac{du}{dx}(x_1) \\ u_1 \\ u_1 \\ u_1 \\ u_1 \\ \dfrac{du}{dx}(x_2) \end{Bmatrix} = \begin{Bmatrix} -150 \\ -300 \\ -300 \\ -300 \\ -300 \\ -150 \end{Bmatrix}$$

which can be solved for

$$\begin{Bmatrix} \dfrac{du}{dx}(x_1) \\ u_1 \\ u_1 \\ u_1 \\ u_1 \\ \dfrac{du}{dx}(x_2) \end{Bmatrix} = \begin{Bmatrix} -5\times10^{-6} \\ -2.4\times10^{-5} \\ -3.6\times10^{-5} \\ -3.6\times10^{-5} \\ -2.4\times10^{-5} \\ 5\times10^{-6} \end{Bmatrix}$$

These results, along with the analytical solution (dashed line) are displayed below:




31.6

```
Option Explicit

Sub FErod()

Dim ns As Integer, ii As Integer, i As Integer, j As Integer
Dim k As Integer, m As Integer
Dim x(5) As Single, st(2, 2) As Single, c As Single
Dim s(2, 2) As Single, a(5, 5) As Single, b(5) As Single, d(5) As Single
Dim Te(5) As Single, ff As Single
Dim e(5) As Single, f(5) As Single, g(5) As Single, r(5) As Single
Dim dum1 As Single, dum2 As Single
Dim dTeLeft As Single, dTeRight As Single

'set parameters
```

```
ns = 4
x(1) = 0
x(2) = 2.5
x(3) = 5
x(4) = 7.5
x(5) = 10
Te(1) = 40
Te(5) = 200
ff = 10

'construct system matrix
st(1, 1) = 1: st(1, 2) = -1: st(2, 1) = -1: st(2, 2) = 1
For ii = 1 To ns
  c = 1 / (x(ii + 1) - x(ii))
  For i = 1 To 2
    For j = 1 To 2
      s(i, j) = c * st(i, j)
    Next j
  Next i
  For i = 1 To 2
    k = ii - 1 + i
    For j = 1 To 2
      m = ii - 1 + j
      a(k, m) = a(k, m) + s(i, j)
    Next j
    b(k) = b(k) + ff * ((x(ii + 1) - x(ii)) - (x(ii + 1) - x(ii)) / 2)
  Next i
Next ii

'determine impact of uniform source and boundary conditions
Call Mmult(a(), Te(), d(), ns + 1, ns + 1, 1)
For i = 1 To ns + 1
  b(i) = b(i) - d(i)
Next i
a(1, 1) = 1
a(2, 1) = 0
a(ns + 1, ns + 1) = -1
a(ns, ns + 1) = 0

'Transform square matrix into tridiagonal form
f(1) = a(1, 1)
g(1) = a(1, 2)
r(1) = b(1)
For i = 2 To ns
  e(i) = a(i, i - 1)
  f(i) = a(i, i)
  g(i) = a(i, i + 1)
  r(i) = b(i)
Next i
e(ns + 1) = a(ns + 1, ns)
f(ns + 1) = a(ns + 1, ns + 1)
r(ns + 1) = b(ns + 1)

'Tridiagonal solver
dum1 = Te(1)
dum2 = Te(ns + 1)
Call Tridiag(e, f, g, r, ns + 1, Te())
dTeLeft = Te(1)
dTeRight = Te(ns + 1)
Te(1) = dum1
Te(ns + 1) = dum2

'output results
Range("a3").Select
ActiveCell.Value = "dTe(x = " & x(0) & ")/dx = "
ActiveCell.Offset(0, 1).Select
ActiveCell.Value = dTeLeft
ActiveCell.Offset(1, -1).Select
ActiveCell.Value = "dTe(x = " & x(ns + 1) & ")/dx = "
ActiveCell.Offset(0, 1).Select
ActiveCell.Value = dTeRight
ActiveCell.Offset(3, -1).Select
```

```
For i = 1 To ns + 1
  ActiveCell.Value = x(i)
  ActiveCell.Offset(0, 1).Select
  ActiveCell.Value = Te(i)
  ActiveCell.Offset(1, -1).Select
Next i
Range("b3").Select

End Sub

Sub Mmult(a, b, c, m, n, l)

Dim i As Integer, j As Integer, k As Integer
Dim sum As Single

For i = 1 To n
  sum = 0
  For k = 1 To m
    sum = sum + a(i, k) * b(k)
  Next k
  c(i) = sum
Next i

End Sub

Sub Tridiag(e, f, g, r, n, x)
Dim k As Integer

'decompose
For k = 2 To n
  e(k) = e(k) / f(k - 1)
  f(k) = f(k) - e(k) * g(k - 1)
Next k
'substitute
For k = 2 To n
  r(k) = r(k) - e(k) * r(k - 1)
Next k
x(n) = r(n) / f(n)
For k = n - 1 To 1 Step -1
  x(k) = (r(k) - g(k) * x(k + 1)) / f(k)
Next k
End Sub
```

The output is

| | A | B | C | D | E |
|---|---|---|---|---|---|
| 1 | **Prob31.6** | | | | |
| 2 | | | | | |
| 3 | **dTe(x = 0)/dx =** | **66** | | RUN | |
| 4 | **dTe(x = 10)/dx =** | **-34** | | | |
| 5 | | | | | |
| 6 | **x** | **Te** | | | |
| 7 | **0** | **40** | | | |
| 8 | **2.5** | **173.75** | | | |
| 9 | **5** | **245** | | | |
| 10 | **7.5** | **253.75** | | | |
| 11 | **10** | **200** | | | |

31.7 After setting up the original spreadsheet, the following modifications would be made to insulate the right edge and add the sink:

Cell I1: Set to 100

Cell I2: `=(I1+2*H2+I3)/4`; This formula would then be copied to cells I3:I8.

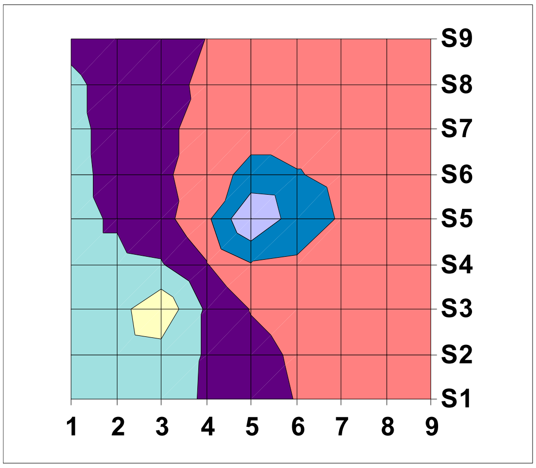
Cell I9: `=(I8+H9)/2`

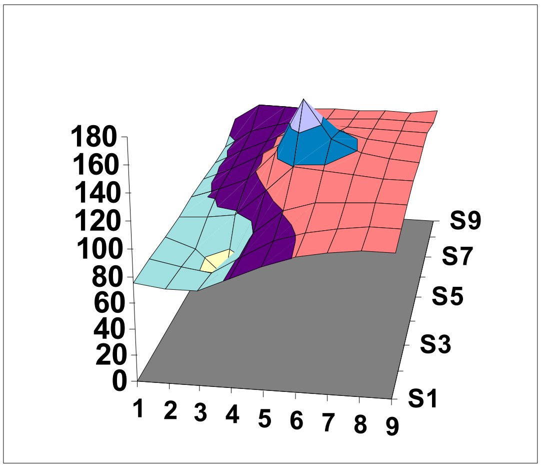
Cell C7: `=(C6+D7+C8+B7-110)/4`

The resulting spreadsheet is displayed below:

| | A | B | C | D | E | F | G | H | I |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 87.5 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 102.8 |
| 2 | 75 | 89.6 | 96.9 | 101.7 | 104.9 | 105.7 | 105.4 | 105.1 | 105.6 |
| 3 | 75 | 86.4 | 96.2 | 105.2 | 112.1 | 112.4 | 110.8 | 109.5 | 109.2 |
| 4 | 75 | 85.0 | 96.3 | 110.8 | 126.0 | 120.9 | 115.9 | 113.1 | 112.2 |
| 5 | 75 | 82.2 | 93.2 | 115.7 | 160.1 | 129.4 | 118.7 | 114.6 | 113.5 |
| 6 | 75 | 75.6 | 78.4 | 98.8 | 119.4 | 117.9 | 114.9 | 113.2 | 112.6 |
| 7 | 75 | 66.8 | 46.1 | 81.8 | 100.8 | 107.7 | 109.9 | 110.5 | 110.6 |
| 8 | 75 | 70.3 | 67.5 | 81.4 | 94.3 | 102.3 | 106.5 | 108.4 | 108.9 |
| 9 | 75 | 72.0 | 72.2 | 82.0 | 92.8 | 100.7 | 105.3 | 107.6 | 108.2 |

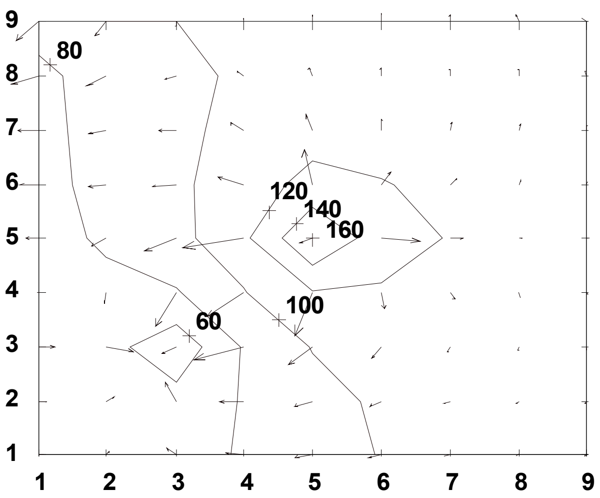
Corresponding contour plots can be generated as




31.8 The results of the preceding problem (31.8) can be saved as a tab-delimited text file (in our case, we called the file prob3108.txt). The following commands can then be used to load this file into MATLAB, as well as to generate the contour plot along with heat flow vectors.

```
>> load prob3108.txt
>> [px,py]=gradient(prob3108);
>> cs=contour(prob3108);clabel(cs);hold on
>> quiver(-px,-py);hold off
```




31.9 The scheme for implementing this calculation on Excel is shown below:

| | A | B | C | D | E | F | G | H | I | J | K |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **1** | 87.5 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 62.5 |
| **2** | 75 | | | | | | | | | | 25 |
| **3** | 75 | | | | | | | | | | 25 |
| **4** | 75 | | | | | | | | | | 25 |
| **5** | 75 | | | | | | | | | | 25 |
| **6** | 62.5 | 50 | 50 | 50 | 50 | 50 | 50 | 50 | 50 | 50 | 37.5 |

The simple Laplace equation applies directly to the blank white cells (recall Fig. 31.14). However, for the shaded cells impacted by the heat sink, a heat balance equation must be written. For example, for cell E3, the control volume approach can be developed as

$$0 = -k'\frac{E3-D3}{\Delta x}\Delta y\Delta z + k'\frac{F3-D3}{\Delta x}\Delta y\Delta z - k'\frac{E3-E4}{\Delta y}\Delta x\Delta z + k'\frac{E2-E3}{\Delta y}\Delta x\Delta z - 100\Delta x\Delta y$$

Collecting and canceling terms yields

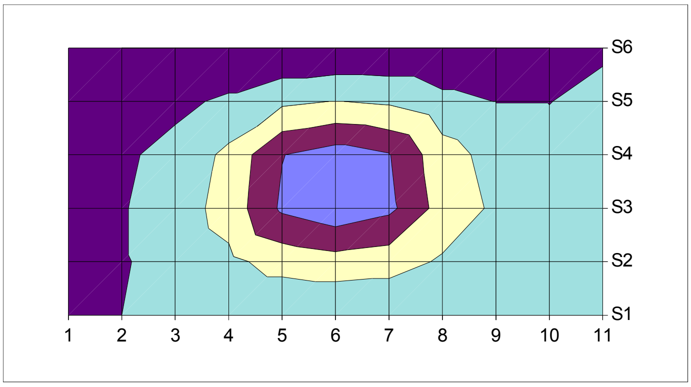
$$0 = -4E3 + D3 + F3 + E4 + D3 - 100\frac{\Delta x\Delta y}{\Delta z k'}$$

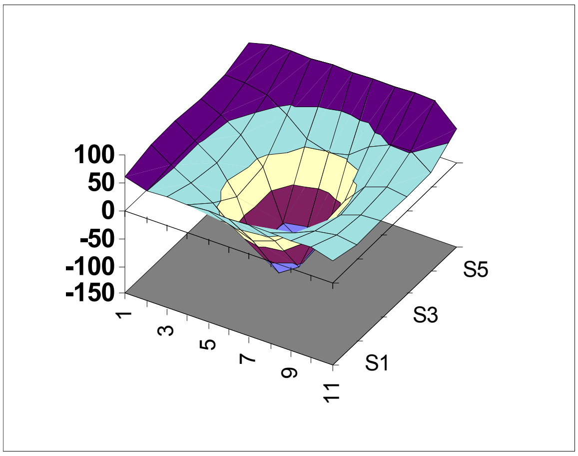
Substituting the length dimensions and the coefficient of thermal conductivity gives,

$$E3 = \frac{D3 + F3 + E4 + D3 - 160}{4}$$

The result is depicted below, along with the corresponding contour plots.

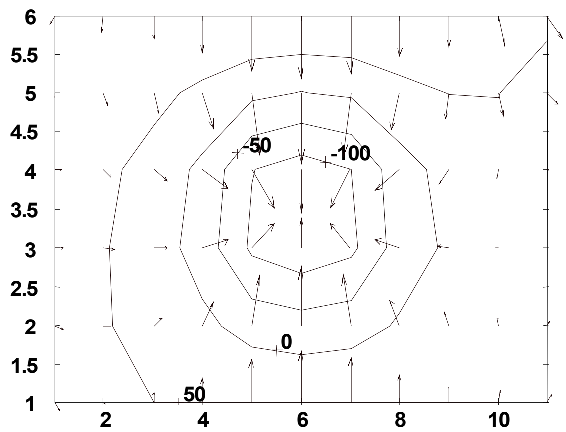
| | A | B | C | D | E | F | G | H | I | J | K |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **1** | 87.5 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 62.5 |
| **2** | 75 | 74.0 | 62.0 | 40.2 | 10.1 | -1.5 | 7.9 | 34.7 | 50.8 | 51.3 | 25 |
| **3** | 75 | 59.0 | 33.7 | -11.2 | -98.2 | -123.8 | -101.8 | -19.8 | 17.3 | 29.2 | 25 |
| **4** | 75 | 53.1 | 25.3 | -20.7 | -108.0 | -133.7 | -111.6 | -29.2 | 8.8 | 23.4 | 25 |
| **5** | 75 | 53.3 | 34.9 | 11.2 | -19.5 | -31.3 | -21.8 | 5.7 | 23.8 | 30.5 | 25 |
| **6** | 62.5 | 50 | 50 | 50 | 50 | 50 | 50 | 50 | 50 | 50 | 37.5 |




31.10 The results of the preceding problem (31.10) can be saved as a tab-delimited text file (in our case, we called the file prob3110.txt). The following commands can then be used to load this file into MATLAB, as well as to generate the contour plot along with heat flow vectors.

```
>> load prob3110.txt
>> [px,py]=gradient(prob3110);
>> cs=contour(prob3110);clabel(cs);hold on
>> quiver(-px,-py);hold off
```




31.11

```
Program Plate
Use IMSL
Implicit None
Integer::ncval, nx, nxtabl, ny, nytabl
Parameter (ncval=11, nx=33, nxtabl=5, ny=33, nytabl=5)
Integer::i, ibcty(4), iorder, j, nout
Real::ax,ay,brhs,bx,by,coefu,prhs,u(nx,ny),utabl,x,xdata(nx),y,ydata(ny)
External brhs, prhs
ax = 0
bx = 40
ay = 0
by = 40
ibcty(1) = 1
ibcty(2) = 2
ibcty(3) = 1
ibcty(4) = 1
coefu = 0
iorder = 4
Call FPS2H(prhs, brhs, coefu, nx, ny, ax, bx, ay, by, ibcty, iorder, u, nx)
Do i=1, nx
  xdata(i) = ax + (bx - ax) * Float(i - 1) / Float(nx - 1)
End Do
Do j=1, ny
  ydata(j) = ay + (by - ay) * Float(j - 1) / Float(ny - 1)
```

```
      End Do
      Call UMACH(2, nout)
      Write (nout,'(8X,A,11X,A,11X,A)') 'X', 'Y', 'U'
      Do j=1, nytabl
        Do i=1, nxtabl
          x = ax + (bx - ax) * Float(i - 1) / Float(nxtabl - 1)
          y = ay + (by - ay) * Float(j - 1) / Float(nytabl - 1)
          utabl = QD2VL(x,y,nx,xdata,ny,ydata,u,nx,.FALSE.)
          Write (nout,'(4F12.4)') x, y, utabl
        End Do
      End Do
      End Program

      Function prhs(x, y)
      Implicit None
      Real::prhs, x, y
      prhs = 0
      End Function

      Real Function brhs(iside, x, y)
      Implicit None
      Integer::iside
      Real::x , y
      If (iside == 1) Then
        brhs = 50
      ElseIf (iside == 2) Then
        brhs = 0
      ElseIf (iside == 3) Then
        brhs = 75
      Else
        brhs = 100
      End If
      End Function
```

Output:

```
       0.0000      0.0000     75.0000
      10.0000      0.0000     71.6339
      20.0000      0.0000     66.6152
      30.0000      0.0000     59.1933
      40.0000      0.0000     50.0000
       0.0000     10.0000     75.0000
      10.0000     10.0000     72.5423
      20.0000     10.0000     67.9412
      30.0000     10.0000     60.1914
      40.0000     10.0000     50.0000
       0.0000     20.0000     75.0000
      10.0000     20.0000     75.8115
      20.0000     20.0000     72.6947
      30.0000     20.0000     64.0001
      40.0000     20.0000     50.0000
       0.0000     30.0000     75.0000
      10.0000     30.0000     83.5385
      20.0000     30.0000     83.0789
      30.0000     30.0000     74.3008
      40.0000     30.0000     50.0000
       0.0000     40.0000     87.5000
      10.0000     40.0000    100.0000
      20.0000     40.0000    100.0000
      30.0000     40.0000    100.0000
      40.0000     40.0000     75.0000
Press any key to continue
```

31.12

Element No. 1

Node No. 1

$$\sum q_k + f(x) = 0$$

$$\left(-kA\frac{dT}{dx}\bigg|_1 + kA\left(\frac{1}{x_2 - x_1}\right)(T_2 - T_1)\right) + \int_{x_1}^{x_2} N_1 f(x)dx = 0$$

$$\left(-100\frac{dT}{dx}\bigg|_1 + \frac{100}{10}(T_2 - T_1)\right) + \int_0^{10}\left(\frac{10-x}{10}\right)30dx = 0$$

$$10(T_1 - T_2) = -100\frac{dT}{dx}\bigg|_1 + 150$$

Node No. 2

$$\left(kA\frac{dT}{dx}\bigg|_2 - kA\left(\frac{1}{x_2 - x_1}\right)(T_2 - T_1)\right) + \int_{x_1}^{x_2} N_2 f(x)dx = 0$$

$$\left(100\left(\frac{dT}{dx}\right)\bigg|_1 - \frac{100}{10}(T_2 - T_1)\right) + \int_0^{10}\left(\frac{x-0}{10}\right)30dx = 0$$

$$-10(T_1 - T_2) = 100\frac{dT}{dx}\bigg|_2 + 150$$

**Other node equations are derived similarly**

Element No. 2

Node No. 2

$$10(T_2 - T_3) = -100\frac{dT}{dx}\bigg|_2 + 150$$

Node No. 3

$$-10(T_2 - T_3) = \frac{dT}{dx}\bigg|_3 + 150$$

Other element equations are similar.

Equation Assembly

$$\begin{bmatrix} 10 & -10 & 0 & 0 & 0 & 0 \\ -10 & 20 & -10 & 0 & 0 & 0 \\ 0 & -10 & 20 & -10 & 0 & 0 \\ 0 & 0 & -10 & 20 & -10 & 0 \\ 0 & 0 & 0 & -10 & 20 & -10 \\ 0 & 0 & 0 & 0 & -10 & 10 \end{bmatrix} + \begin{bmatrix} T_1 \\ T_2 \\ T_3 \\ T_4 \\ T_5 \\ T_6 \end{bmatrix} = \begin{bmatrix} -100\frac{dT}{dx}\big|_1 + 150 \\ 300 \\ 300 \\ 300 \\ 300 \\ 100\frac{dT}{dx}\big|_6 + 150 \end{bmatrix}$$

Inserting Boundary Conditions

$$\begin{bmatrix} 10 & -10 & 0 & 0 & 0 & 0 \\ -10 & 20 & -10 & 0 & 0 & 0 \\ 0 & -10 & 20 & -10 & 0 & 0 \\ 0 & 0 & -10 & 20 & -10 & 0 \\ 0 & 0 & 0 & -10 & 20 & 0 \\ 0 & 0 & 0 & 0 & -10 & -100 \end{bmatrix} + \begin{bmatrix} T_1 \\ T_2 \\ T_3 \\ T_4 \\ T_5 \\ \left.\frac{dT}{dx}\right|_6 \end{bmatrix} = \begin{bmatrix} 125 \\ 300 \\ 300 \\ 300 \\ 1300 \\ -850 \end{bmatrix}$$

```
% Solution of Linear Algebraic Equation
% Problem 31.1
% Equation form Cx=b


C=[ 10   -10    0      0       0      0;
   -10    20   -10     0       0      0;
    0    -10    20    -10      0      0;
    0     0    -10     20     -10     0;
    0     0     0     -10      20     0;
    0     0     0      0      -10   -100];

b=[125  300  300  300  1300  -850]';

     %Solution by inverse of A
     %Matrix Eqn. Form A*x=b
x=inv(C)*b;
fprintf('%5.1f \n' ,x)

 »
 462.5
 450.0
 407.5
 335.0
 232.5
 -14.8
 »
```

31.13

Element No. 1

Node No. 1

$$\sum q_k + f(x) = 0$$

$$\left(-kA\frac{dT}{dx}\bigg|_1 + kA\left(\frac{1}{x_2 - x_1}\right)(T_2 - T_1)\right) + \int_{x_1}^{x_2} N_1 f(x)dx = 0$$

$$\left(-100\frac{dT}{dx}\bigg|_1 + \frac{95}{10}(T_2 - T_1)\right) + \int_0^{10}\left(\frac{10 - x}{10}\right)30dx = 0$$

$$9.5(T_1 - T_2) = -100\frac{dT}{dx}\bigg|_1 + 150$$

Node No. 2

$$\left(kA\frac{dT}{dx}\bigg|_2 - kA\left(\frac{1}{x_2 - x_1}\right)(T_2 - T_1)\right) + \int_{x_1}^{x_2} N_2 f(x)dx = 0$$

$$\left(90\left(\frac{dT}{dx}\right)\bigg|_1 - \frac{95}{10}(T_2 - T_1)\right) + \int_0^{10}\left(\frac{x - 0}{10}\right)30dx = 0$$

$$-9.5(T_1 - T_2) = 90\frac{dT}{dx}\bigg|_2 + 150$$

**Other node equations are derived similarly**

Element No. 2

Node No. 2

$$8.5(T_2 - T_3) = -90\frac{dT}{dx}\bigg|_2 + 150$$

Node No. 3

$$-8.5(T_2 - T_3) = 80\frac{dT}{dx}\bigg|_3 + 150$$

Element No. 3

Node No. 3

$$7.5(T_3 - T_4) = -80\frac{dT}{dx}\bigg|_3 + 150$$

Node No. 4

$$-7.5(T_3 - T_4) = 70\frac{dT}{dx}\bigg|_4 + 150$$

Element No. 4

Node No. 4

$$6.5(T_4 - T_5) = -70\frac{dT}{dx}\bigg|_4 + 150$$

Node No. 5

$$-6.5(T_4 - T_5) = 60\frac{dT}{dx}\bigg|_5 + 150$$

Element No. 5

Node No. 5

$$5.5(T_5 - T_6) = -60\frac{dT}{dx}\bigg|_5 + 150$$

Node No. 6

$$-5.5(T_5 - T_6) = 50\frac{dT}{dx}\bigg|_6 + 150$$

Equation Assembly

$$\begin{bmatrix} 9.5 & -9.5 & 0 & 0 & 0 & 0 \\ -9.5 & 18 & -8.5 & 0 & 0 & 0 \\ 0 & -8.5 & 16 & -7.5 & 0 & 0 \\ 0 & 0 & -7.5 & 14 & -6.5 & 0 \\ 0 & 0 & 0 & -6.5 & 12 & -5.5 \\ 0 & 0 & 0 & 0 & -5.5 & 5.5 \end{bmatrix} + \begin{bmatrix} T_1 \\ T_2 \\ T_3 \\ T_4 \\ T_5 \\ T_6 \end{bmatrix} = \begin{bmatrix} -100\frac{dT}{dx}\Big|_1 + 150 \\ 300 \\ 300 \\ 300 \\ 300 \\ 50\frac{dT}{dx}\Big|_6 + 150 \end{bmatrix}$$

Inserting Boundary Conditions

$$\begin{bmatrix} 100 & -9.5 & 0 & 0 & 0 & 0 \\ 0 & 18 & -8.5 & 0 & 0 & 0 \\ 0 & -8.5 & 16 & -7.5 & 0 & 0 \\ 0 & 0 & -7.5 & 14 & -6.5 & 0 \\ 0 & 0 & 0 & -6.5 & 12 & 0 \\ 0 & 0 & 0 & 0 & -5.5 & -50 \end{bmatrix} + \begin{bmatrix} \frac{dT}{dx}\Big|_1 \\ T_2 \\ T_3 \\ T_4 \\ T_5 \\ \frac{dT}{dx}\Big|_6 \end{bmatrix} = \begin{bmatrix} -800 \\ 1200 \\ 300 \\ 300 \\ 575 \\ 125 \end{bmatrix}$$

```
% Solution of Linear Algebraic Equation
% Problem 31.1
% Equation form Cx=b


 C=[ 10   -10     0      0        0      0;
    -10    20   -10      0        0      0;
      0   -10    20    -10        0      0;
      0     0   -10     20      -10      0;
      0     0     0    -10       20      0;
      0     0     0      0      -10   -100];

 b=[125  300  300  300  1300  -850]';

      %Solution by inverse of A
      %Matrix Eqn. Form A*x=b
 x=inv(C)*b;
 fprintf('%5.1f \n' ,x)

»
  7.2
159.7
197.1
199.4
155.9
-19.7
»
```