CHAPTER 20

20.1 The integral can be evaluated analytically as,

$$I = \int_1^2 \left(x + \frac{1}{x}\right)^2 dx = \int_1^2 x^2 + 2 + x^{-2}\, dx = \left[\frac{x^3}{3} + 2x - \frac{1}{x}\right]_1^2 = \frac{2^3}{3} + 2(2) - \frac{1}{2} - \frac{1^3}{3} - 2(1) + \frac{1}{1} = 4.8333$$

The tableau depicting the implementation of Romberg integration to ε_s = 0.5% is

iteration→	**1**	**2**	**3**
ε_t →	6.0345%	0.0958%	0.0028%
ε_a →		1.4833%	0.0058%
1	5.12500000	4.83796296	4.83347014
2	4.90972222	4.83375094	
4	4.85274376		

Thus, the result is 4.83347014.

20.2 **(a)** The integral can be evaluated analytically as,

$$I = \left[-0.011x^5 + 0.215x^4 - 1.4x^3 + 3.15x^2 + 2x\right]_0^8 = 20.992$$

(b) The tableau depicting the implementation of Romberg integration to ε_s = 0.5% is

iteration→	**1**	**2**	**3**	**4**
ε_t →	87.8049%	71.5447%	0.0000%	0.0000%
ε_a →		14.2857%	4.4715%	0.0000000%
1	2.56000000	5.97333333	20.99200000	20.99200000
2	5.12000000	20.05333333	20.99200000	
4	16.32000000	20.93333333		
8	19.78000000			

Thus, the result is exact.

(c) The transformations can be computed as

$$x = \frac{(8+0)+(8-0)x_d}{2} = 4 + 4x_d \qquad dx = \frac{8-0}{2}dx_d = 4dx_d$$

These can be substituted to yield

$$I = \int_{-1}^{1} \left[-0.055(4+4x_d)^4 + 0.86(4+4x_d)^3 - 4.2(4+4x_d)^2 + 6.3(4+4x_d) + 2\right] 4dx_d$$

The transformed function can be evaluated using the values from Table 20.1

$$I = 0.5555556\, f(-0.774596669) + 0.8888889\, f(0) + 0.5555556\, f(0.774596669) = 20.992$$

which is exact.

(d) The following script can be developed and saved as `Prob1802Script.m`:

```
format long g
y = @(x) -0.055*x.^4+0.86*x.^3-4.2*x.^2+6.3*x+2';
I = quad(y,0,8)
```

When it is run, the result is exact:

```
>> Prob1802Script
I =
                         20.992
```

20.3 Although it's not required, the analytical solution can be evaluated simply as

$$I = \int_0^3 xe^{2x}\,dx = \left[0.25e^{2x}(2x-1)\right]_0^3 = 504.53599$$

(a) The tableau depicting the implementation of Romberg integration to ε_s = 0.5% is

iteration →	1	2	3	4
ε_t →	259.8216%	31.8835%	1.8912%	0.0312%
ε_a →		43.2082%	1.8397%	0.0290545%
1	1815.42957072	665.39980101	514.07794398	504.69324146
2	952.90724344	523.53556004	504.83987744	
4	630.87848089	506.00835760		
8	537.22588842			

(b) The transformations can be computed as

$$x = \frac{(3+0)+(3-0)x_d}{2} = 1.5+1.5x_d \qquad dx = \frac{3-0}{2}dx_d = 1.5dx_d$$

These can be substituted to yield

$$I = \int_{-1}^{1} \left[(1.5+1.5x_d)e^{2(1.5+1.5x_d)}\right] 1.5dx_d$$

The transformed function can be evaluated using the values from Table 20.1

$$I = f(-0.577350269) + f(0.577350269)$$
$$f(-0.577350269) = \left[\left(1.5+1.5(-0.577350269)\right)e^{2(1.5+1.5(-0.577350269))}\right] 1.5 = 3.379298$$
$$f(0.577350269) = \left[\left(1.5+1.5(0.577350269)\right)e^{2(1.5+1.5(0.577350269))}\right] 1.5 = 402.9157$$
$$I = 3.379298 + 402.9157 = 406.295$$

which represents a percent relative error of 19.47%.

(c) Using MATLAB

```
>> format long g
>> I = quad(@(x) x.*exp(2*x),0,3)
I =
          504.535991881658
```

```
>> I = quadl(@(x) x.*exp(2*x),0,3)
I =
          504.535991866404
```

20.4 The exact solution can be evaluated simply as

```
>> format long
>> erf(1.5)
ans =
   0.96610514647531
```

(a) The transformations can be computed as

$$x = \frac{(1.5+0)+(1.5-0)x_d}{2} = 0.75 + 0.75x_d \qquad dx = \frac{1.5-0}{2}dx_d = 0.75dx_d$$

These can be substituted to yield

$$I = \frac{2}{\sqrt{\pi}}\int_{-1}^{1}\left[e^{-(0.75+0.75x_d)^2}\right]0.75dx_d$$

The transformed function can be evaluated using the values from Table 20.1

$$I = f(-0.577350269) + f(0.577350269) = 0.974173129$$

which represents a percent relative error of 0.835 %.

(b) The transformed function can be evaluated using the values from Table 20.1

$$I = 0.5555556\,f(-0.774596669) + 0.8888889\,f(0) + 0.5555556\,f(0.774596669) = 0.965502083$$

which represents a percent relative error of 0.062 %.

20.5 (a) The tableau depicting the implementation of Romberg integration to ε_s = 0.5% is

iteration $\rightarrow$	**1**	**2**	**3**	**4**
$\varepsilon_a \rightarrow$		17.8666%	0.9589%	0.0382084%
1	348.00501404	1219.63999486	1440.68457469	1476.79729373
2	1001.73124965	1426.86928845	1476.23303250	
4	1320.58477875	1473.14779849		
8	1435.00704356			

Note that if 8 iterations are implemented, the method converges on a value of 1480.56848.

(b) The transformations can be computed as

$$x = \frac{(30+0)+(30-0)x_d}{2} = 15 + 15x_d \qquad dx = \frac{30-0}{2}dx_d = 15dx_d$$

These can be substituted to yield

$$I = 200\int_{-1}^{1}\left[\frac{15+15x_d}{5+15x_d}e^{-2(15+15x_d)/30}\right]15dx_d$$

The transformed function can be evaluated using the values from Table 20.1

$$I = f(-0.577350269) + f(0.577350269) = 1610.572$$

(c) Interestingly, the `quad` function encounters a problem and exceeds the maximum number of iterations

```
>> format long g
>> I = quad(@(z) 200*z/(5+z)*exp(-2*z/30),0,30)
Warning: Maximum function count exceeded; singularity likely.
> In quad at 92
I =
          1520.40857847296
```

The `quadl` function converges rapidly, but does not yield a very accurate result:

```
>> I = quadl(@(z) 200*z/(5+z)*exp(-2*z/30),0,30)
I =
          1483.68924281497
```

20.6 The integral to be evaluated is

$$I = \int_0^{1/2}\left(8e^{-t}\sin 2\pi t\right)^2\,dt$$

Note that although it is not necessary, the integral can be evaluated analytically to yield

$$I = \left[-16e^{-2t}\,\frac{1+4\pi^2-\cos(4\pi t)+2\pi\sin(4\pi t)}{1+4\pi^2}\right]_0^{0.5}$$

which can be evaluated as 9.86406915. Therefore, the I_{RMS} = 3.14071157.

(a) The tableau depicting the implementation of Romberg integration to ε_s = 0.1% is

iteration →	1	2	3	4
ε_t →	100.0000%	31.1763%	1.6064%	0.0156%
ε_a →		25.0000%	2.0824%	0.0253398%
1	0.00000000	12.93932074	9.70561610	9.86561132
2	9.70449056	9.90772264	9.86311139	
4	9.85691462	9.86589959		
8	9.86365335			

Therefore, the I_{RMS} = 3.14095707.

(b) The transformations can be computed as

$$x = \frac{(0.5+0)+(0.5-0)x_d}{2} = 0.25+0.25x_d \qquad dx = \frac{0.5-0}{2}dx_d = 0.25dx_d$$

These can be substituted to yield

$$I = \int_{-1}^{1} \left[8e^{-(0.25+0.25x_d)} \sin 2\pi(0.25+0.25x_d)\right]^2 \; 0.25dx_d$$

For the two-point application, the transformed function can be evaluated using the values from Table 20.1

$$I = f(-0.577350269) + f(0.577350269) \; = 7.678608$$

or an I_{RMS} = 2.77103.

For the three-point application, the transformed function can be evaluated using the values from Table 20.1

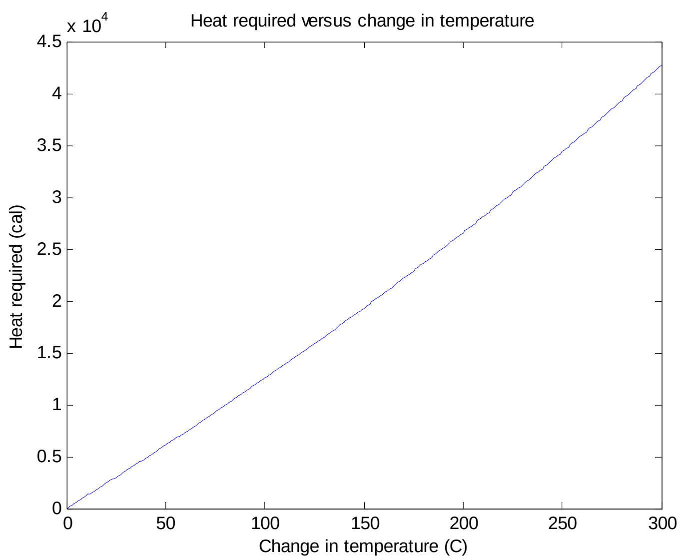
$$I = 0.5555556\, f(-0.774596669) + 0.8888889\, f(0) + 0.5555556\, f(0.774596669) = 10.02083$$

or an I_{RMS} = 3.16557.

(c)
```
>> format long g
>> I = quad(@(x) (8*exp(-x).*sin(2*pi*x)).^2,0,0.5)
I =
          9.86406918193917
```

or an I_{RMS} = 3.14071157.

20.7
```
clear,clc,clf
m=1000; DT=[0:300];
H(1)=0;
for i = 2:length(DT)
  H(i)=m*quad(@(T) 0.132+1.56e-4*T+2.64e-7*T.^2,-100,-100+DT(i));
end
plot(DT,H)
title('Heat required versus change in temperature')
xlabel('Change in temperature (C)'),ylabel('Heat required (cal)')
```



20.8 The integral to be evaluated is

$$I = \int_2^8 (9 + 5\cos^2 0.4t)(5e^{-0.5t} + 2e^{0.15t})\ dt$$

(a) The tableau depicting the implementation of Romberg integration to ε_s = 0.1% is

iteration →	1	2	3	4
ε_a →		8.2537%	0.1298%	0.0014429%
1	437.99743327	329.28470773	336.26944122	335.95919795
2	356.46288911	335.83289538	335.96404550	
4	340.99039381	335.95584861		
8	337.21448491			

(b)
```
>> format long g
>> Qc = @(t) (9+5*cos(0.4*t).^2).*(5*exp(-0.5*t)+2*exp(0.15*t));
>> I=quad(Qc,2,8)

I =
          335.962530076433
```

20.9 **(a)** The integral can be evaluated analytically as,

$$\int_{-2}^{2}\left[\frac{x^3}{3} - 3y^2x + y^3\frac{x^2}{2}\right]_0^4 dy$$

$$\int_{-2}^{2}\frac{(4)^3}{3} - 3y^2(4) + y^3\frac{(4)^2}{2}\ dy$$

$$\int_{-2}^{2} 21.33333 - 12y^2 + 8y^3\ dy$$

$$\left[21.33333y - 4y^3 + 2y^4\right]_{-2}^{2}$$

$$21.33333(2) - 4(2)^3 + 2(2)^4 - 21.33333(-2) + 4(-2)^3 - 2(-2)^4 = 21.33333$$

(b) The operation of the `dblquad` function can be understood by invoking `help`,

```
>> help dblquad
```

A session to use the function to perform the double integral can be implemented as,

```
>> dblquad(inline('x.^2-3*y.^2+x*y.^3'),0,4,-2,2)

ans =
   21.3333
```

20.10
```
>> F=@(x) (1.6*x-0.045*x.^2).*cos(-0.00055*x.^3+0.0123*x.^2+0.13*x);
>> W=quad(F,0,30)
W =
 -157.0871
```

20.11 The integral to be determined is

$$I = \int_0^{1/2} (6e^{-1.25t}\sin 2\pi t)^2 \ dt$$

Change of variable:

$$x = \frac{0.5+0}{2} + \frac{0.5-0}{2} x_d = 0.25 + 0.25 x_d \qquad dx = \frac{0.5-0}{2} dx_d = 0.25 dx_d$$

$$I = \int_{-1}^{1} (6e^{-1.25(0.25+0.25x_d)}\sin 2\pi (0.25+0.25x_d))^2 \ 0.25 \ dx_d$$

Therefore, the transformed function is

$$f(x_d) = 0.25\ (6e^{-1.25(0.25+0.25x_d)}\sin 2\pi (0.25+0.25x_d))^2$$

Five-point formula:

$$I = 0.236927f(-0.90618) + 0.478629f(-0.53847) + 0.568889f(0) \\ + 0.478629f(0.53847) + 0.236927f(0.90618) = 5.54854$$

Therefore, the RMS current can be computed as

$$I_{\text{RMS}} = \sqrt{5.54854} = 2.355534$$

20.12 (a)

```
>> F1=@(t) (sin(2*pi*t)).^2;
>> Irms=sqrt(quad(F1,0,1))
Irms =
    0.7071
>> R=5; P=Irms^2*R
P =
    2.5000
```

(b)

```
>> ib=@(t) sin(2*pi*t);
>> Vb=@(t) 5*ib(t)-1.25*ib(t).^2;
>> Pb=@(t) ib(t).*V2(t);
>> P=quad(Pb,0,1)
P =
    2.5000
```

Interestingly, the power is identical. The reason for this can be seen by inspecting each of the power functions. For **(a)**, the power function is

$$P = I^2 R = 5(\sin 2\pi t)^2$$

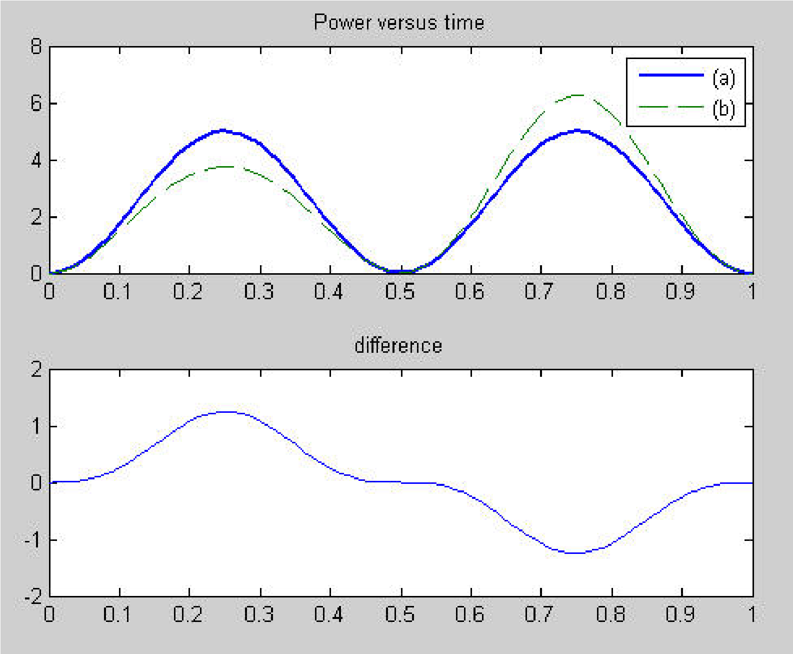
For **(b)**, it is

$$P = IV = I(5I - 1.25I^2) = 5I^2 - 6.25I^3$$
$$P = 5(\sin 2\pi t)^2 - 6.25(\sin 2\pi t)^3$$

A plot can be developed of both functions along with their difference.

```
t=linspace(0,1);
```

```
Pa=@(t) 5*(sin(2*pi*t)).^2;
P1=Pa(t); P2=Pb(t);
subplot(2,1,1),plot(t,P1,t,P2,'--')
legend('(a)','(b)'),title('Power versus time')
subplot(2,1,2),plot(t,delta)
title('difference')
```



As can be seen, the difference is symmetrical across the period. Therefore, the positive and negative discrepancies cancel.

20.13 The average voltage can be computed as

$$\bar{V} = \frac{\int_0^{60} i(t)R(i)\,dt}{60}$$

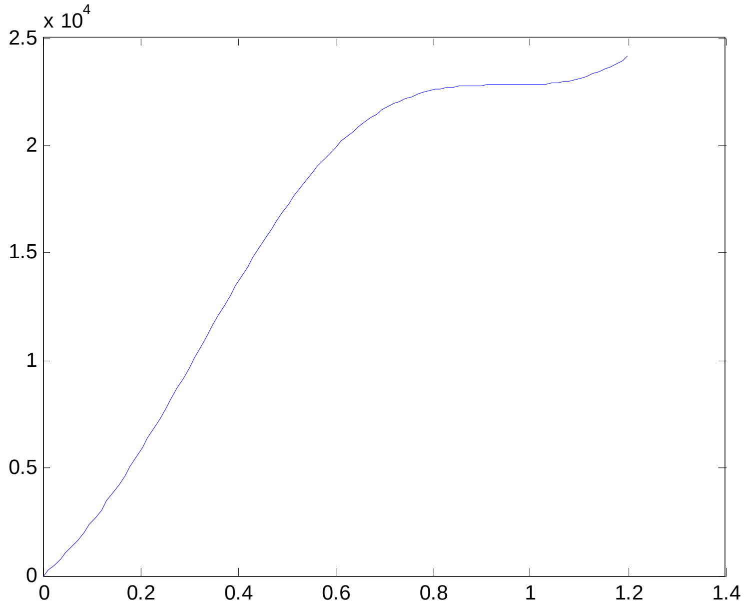
We can use the formulas to generate values of $i(t)$ and $R(i)$ and their product for various equally-spaced times over the integration interval as summarized in the table below. The last column shows the integral of the product as calculated with Simpson's 1/3 rule.

t	*i(t)*	*R(i)*	*i(t)* × *R(i)*	Simpson's 1/3
0	3600.000	36469.784	131291223	
6	2950.461	29916.029	88266063	1075071847
12	2288.787	23235.215	53180447	
18	1726.549	17553.332	30306694	381186392
24	1260.625	12839.641	16185971	
30	878.355	8966.984	7876196	102019847
36	569.294	5830.320	3319166	
42	327.533	3370.360	1103904	15944048
48	151.215	1568.912	237242	
54	41.250	436.372	18000	618486
60	0.000	0.000	0	
			Sum →	**1574840619**

The average voltage can therefore be computed as

$$\bar{V} = \frac{1{,}574{,}840{,}619}{60} = 2.6247344 \times 10^7$$

20.14
```
clear,clc,clf
t = [0 0.2 0.4 0.6 0.8 1 1.2];
icurr = [0.2 0.3683 0.3819 0.2282 0.0486 0.0082 0.1441];
C=1e-5;
p=polyfit(t,icurr,5);
f= @(t) p(1)*t.^5+p(2)*t.^4+p(3)*t.^3+p(4)*t.^2+p(5)*t+p(6);
t=[0:1.2/100:1.2]; V(1)=0;
for i = 2:length(t)
  V(i)=quad(f,0,t(i))/C;
end
plot(t,V)
```



20.15 The work is computed as the product of the force times the distance, where the latter can be determined by integrating the velocity data,

$$W = F\int_0^t v(t)\,dt$$

Before solving this problem numerically, it can be solved analytically,

$$W = F\left[\int_0^5 4t\ dt + \int_5^{15} 20 + (5-t)^2\ dt\right] = 200\left[\left[2t^2\right]_0^5 + \left[\frac{t^3}{3} - 5t^2 + 45t\right]_5^{15}\right]$$

$$= 200\left[50 + 533.333\right] = 200(583.333) = 116{,}666.7\ \text{N}\cdot\text{m}$$

Romberg integration gives

iteration→	1	2	3	4	5
ea→		15.0000%	0.2660%	0.0127681%	0.0004501%
1	900.00000000	562.50000000	587.50000000	582.73809524	583.41036415
2	646.87500000	585.93750000	582.81250000	583.40773810	
4	601.17187500	583.00781250	583.39843750		
8	587.54882813	583.37402344			
16	584.41772461				

Thus, we are converging on the exact result.

Interestingly, the MATLAB quad function does not perform as well. To illustrate this, a function can be developed to hold the integrand

```
function v=velocity(t)
if t <= 5
  v=4*t;
else
  v=20+(5-t).^2;
end
```

A script can then be written to evaluate the integral and compute the work:

```
clear,clc,clf
format long g
Work=200*quad(@velocity,0,15)

Work =
          117062.875572907
```

Thus, the work is computed as 117,062.9 N· m.

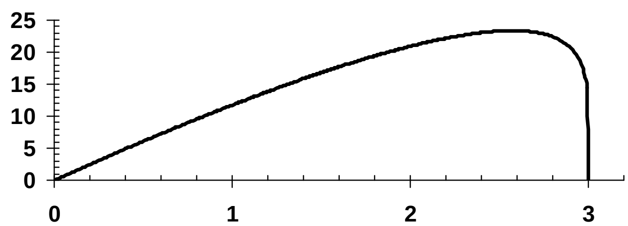
20.16 As in the plot, the initial point is assumed to be e = 0, s = 40. We can then use a combination of the trapezoidal and Simpsons rules to integrate the data as

$$I = (0.02-0)\frac{40+40}{2} + (0.05-0.02)\frac{40+37.5}{2} + (0.25-0.05)\frac{37.5+4(43+60)+2(52)+60}{12} =$$
$$0.8 + 1.1625 + 4.358333 + 5.783333 = 12.10417$$

20.17 The function to be integrated is

$$Q = \int_0^3 2\left(1-\frac{r}{r_0}\right)^{1/6} (2\pi r)\, dr$$

A plot of the integrand can be developed as



As can be seen, the shape of the function indicates that we must use fine segmentation to attain good accuracy. Here are the results of using a variety of segments.

n	*Q*	*n*	*Q*
2	25.1896	1024	44.7289
4	36.1635	2048	44.7361
8	40.9621	4096	44.7392
16	43.0705	8192	44.7407
32	44.0009	16384	44.7413
64	44.4127	32768	44.7416
128	44.5955	65536	44.7417
256	44.6767	131072	44.7418
512	44.7128	262144	44.7418

Therefore, the result to 4 significant figures appears to be 44.7418. The same evaluation can be performed simply with MATLAB

```
>> vA=@(r) 2*(1-r/3).^(1/6)*2*pi.*r;
>> Q=quad(vA,0,3)
Q =
   44.7418
```

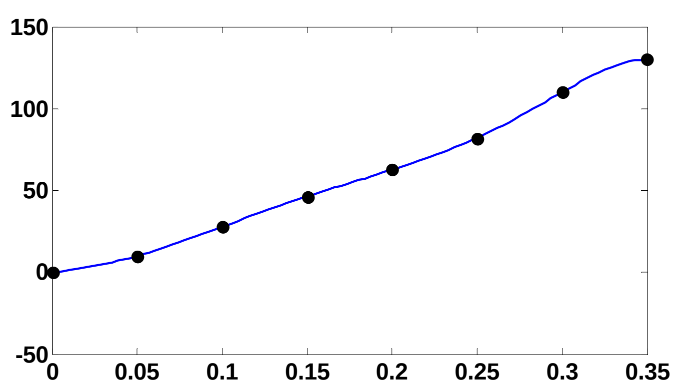
20.18 The work is computed as

$$W = k\int_0^x F\ dx$$

The following script fits a 6th-order polynomial to the data and then evaluates the integral of this polynomial with the `quad` function. A plot of the polynomial fit is also displayed.

```
clear,clc,clf
x = [0 0.05 0.1 0.15 0.2 0.25 0.3 0.35];
F = [0 10 28 46 63 82 110 130];
k=300;
p=polyfit(x,F,6);
f= @(x) p(1)*x.^6+p(2)*x.^5+p(3)*x.^4+p(4)*x.^3+p(5)*x.^2+p(6)*x+p(7);
xp=linspace(0,max(x),100); Fp=f(xp);
plot(xp,Fp,x,F,'o')
I=quad(f,0,max(x))
Work=k*I

I =
          20.2181915377859
Work =
          6065.45746133578
```



20.19 The distance traveled is equal to the integral of velocity

$$y = \int_{t_1}^{t_2} v(t)\, dt$$

A table can be set up holding the velocities at evenly spaced times (h = 1) over the integration interval. The Simpson's 1/3 rule can then be used to integrate this data as shown in the last column of the table

t	*v*	**Simp 1/3 rule**
0	0	
1	6	19.33333
2	34	
3	84	175.3333
4	156	
5	250	507.3333
6	366	
7	504	1015.333
8	664	
9	846	1699.333
10	1050	
11	1045	2090
12	1040	
13	1035	2070
14	1030	
15	1025	2050
16	1020	
17	1015	2030
18	1010	
19	1005	2010
20	1000	
21	1052	2105.333
22	1108	
23	1168	2337.333
24	1232	
25	1300	2601.333
26	1372	
27	1448	2897.333
28	1528	
29	1612	3225.333
30	1700	
	Sum →	**26833.33**

Since the underlying functions are second order or less, this result should be exact. We can verify this by evaluating the integrals analytically,

$$y = \int_0^{10} 11t^2 - 5t \; dt = \left[3.66667t^3 - 2.5t^2\right]_0^{10} = 3416.667$$

$$y = \int_{10}^{20} 1100 - 5t \; dt = \left[1100t - 2.5t^2\right]_{10}^{20} = 10{,}250$$

$$y = \int_{20}^{30} 50t + 2(t-20)^2 \; dt = \left[\frac{2}{3}t^3 - 15t^2 + 800t\right]_{20}^{30} = 13{,}166.67$$

The total distance traveled is therefore 3416.667 + 10,250 + 13,166.67 = 26,833.33.

Interestingly, the MATLAB `quad` function does not perform as well. To illustrate this, a function can be developed to hold the integrand

```
function v=vel(t)
if t <= 10
  v=11*t.^2-5*t;
elseif t <= 20
  v=1100-5*t;
else
  v=50*t+2*(t-20).^2;
end
```

A script can then be written to evaluate the integral and compute the work:

```
clear,clc
format long g
Distance=quad(@vel,0,30)

Distance =
          26835.4501169434
```

20.20 6-segment trapezoidal rule:

$$y=(30-0)\frac{0+2(101.439+216.213+346.916+496.983+671.095)+875.867}{2(6)}=11{,}352.9$$

<u>6-segment Simpson's 1/3 rule</u>:

$$y=(30-0)\frac{0+4(101.439+346.916+671.095)+2(216.213+496.983)+875.867}{3(6)}=11{,}300.1$$

<u>6-point Gauss quadrature</u>: y = 11,299.831051

<u>Romberg integration</u>:

	1	**2**	**3**	**4**
n	ε_a →	4.0210%	0.0097%	0.0000288%
1	13138.00101	11317.65672	11300.04046	**11299.83245**
2	11772.74279	11301.14147	11299.83570	
4	11419.04180	11299.91731		
8	11329.69844			

MATLAB script:

```
clear,clc
format long g
v=@(t) 1850*log(160000./(160000-2500*t))-9.81*t;
y=quad(v,0,30)

y =
          11299.8310550376
```

20.21 (a) Create the following M function:

```
>> y=@(x) 1/sqrt(2*pi)*exp(-(x.^2)/2);
```

```
>> Q=quad(y,-1,1)
Q =
    0.6827
>> Q=quad(y,-2,2)
Q =
    0.9545
```

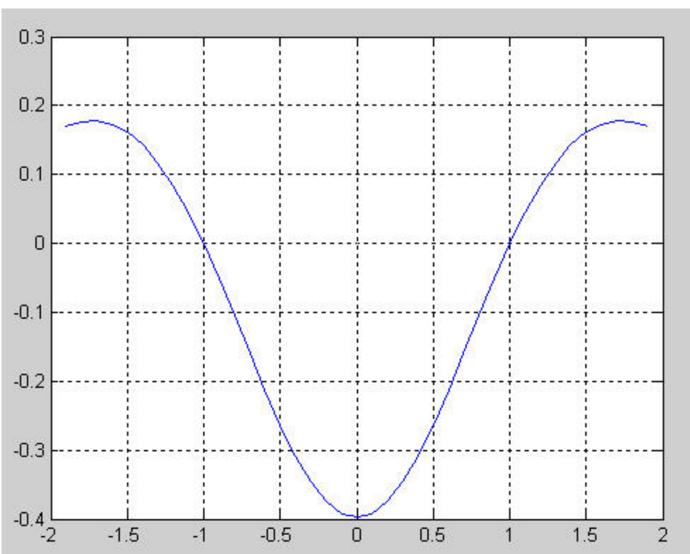
Thus, about 68.3% of the area under the curve falls between –1 and 1 and about 95.45% falls between –2 and 2.

(b) The inflection point is indicated by a zero second derivative. Recall from Chap. 4 (p. 103), that the second derivative can be approximated by

$$f''(x_i) \cong \frac{f(x_{i+1}) - 2f(x_{i+1}) + f(x_{i-1})}{h^2}$$

The following script uses this formula to compute the second derivative and generate a plot of the results,

```
x=-2:.1:2;
y=1/sqrt(2*pi)*exp(-(x.^2)/2);
xx=zeros(length(x)-2);
d2ydx2=xx;
for i=2:length(x)-1
  xx(i-1)=x(i);
  d2ydx2(i-1)=(y(i-1)-2*y(i)+y(i+1))/(x(i)-x(i-1))^2;
end
plot(xx,d2ydx2);grid
```



Thus, inflection points ($d^2y/dx^2 = 0$) occur at –1 and 1.

Note that in the next chapter we will introduce the `diff` function which provides an alternative way to make the same assessment. Here is a script that illustrates how this might be done:

```
x=-2:.1:2;
f=y(x);
d=diff(f)./diff(x);
xx=-1.95:.1:1.95;
```

```
d2=diff(d)./diff(xx);
xxx=-1.9:.1:1.9;
plot(xxx,d2,'o')
```

20.22

	1	2	3
n	$\varepsilon_a \rightarrow$	**7.9715%**	**0.0997%**
1	1.34376994	1.97282684	1.94183605
2	1.81556261	1.94377297	
4	1.91172038		

20.23 (a) Romberg:

	1	2	3	4	5
n	$\varepsilon_a \rightarrow$	4.2665%	0.0316%	0.0001124%	0.0000000%
1	212.75103	256.53098	255.24166	255.26002	**255.26003094**
2	245.58600	255.32225	255.25974	255.26003	
4	252.88818	255.26364	255.26003		
8	254.66978	255.26025			
16	255.11263				

(b) MATLAB script:

```
clear,clc
format long g
g=9.81;m=80;cd=0.2;
v=@(t) sqrt(g*m/cd)*tanh(sqrt(g*cd/m)*t);
y=quad(v,0,8)

y =
          255.260030099128
```

20.24 Equation 20.30 is

$$I = I(h_2) + \frac{1}{15}\left[I(h_2) - I(h_1)\right]$$

The integrals can be represented by

$$I(h_1) = (x_4 - x_0)\frac{f(x_0) + 4f(x_2) + f(x_4)}{2}$$
$$I(h_2) = (x_2 - x_0)\frac{f(x_0) + 4f(x_1) + f(x_2)}{6} + (x_4 - x_2)\frac{f(x_2) + 4f(x_3) + f(x_4)}{6}$$

Substituting these into Eq. (20.30) gives

$$I = (x_2 - x_0)\frac{f(x_0) + 4f(x_1) + f(x_2)}{6} + (x_4 - x_2)\frac{f(x_2) + 4f(x_3) + f(x_4)}{6}$$
$$+\frac{1}{15}\left[(x_2 - x_0)\frac{f(x_0) + 4f(x_1) + f(x_2)}{6} + (x_4 - x_2)\frac{f(x_2) + 4f(x_3) + f(x_4)}{6} - (x_4 - x_0)\frac{f(x_0) + 4f(x_2) + f(x_4)}{6}\right]$$

Note that if $h = x_4 - x_0$, $x_2 - x_0 = x_4 - x_2 = h/2$. Therefore,

$$I = \frac{h}{2}\frac{f(x_0)+4f(x_1)+f(x_2)}{6} + \frac{h}{2}\frac{f(x_2)+4f(x_3)+f(x_4)}{6}$$
$$+\frac{1}{15}\left[\frac{h}{2}\frac{f(x_0)+4f(x_1)+f(x_2)}{6} + \frac{h}{2}\frac{f(x_2)+4f(x_3)+f(x_4)}{6} - h\frac{f(x_0)+4f(x_2)+f(x_4)}{6}\right]$$

Collecting terms

$$\frac{I}{h} = \frac{1}{12}f(x_0) + \frac{1}{15}\frac{1}{12}f(x_0) - \frac{1}{15}\frac{1}{6}f(x_0) + \frac{4}{12}f(x_1) + 4\frac{1}{15}\frac{1}{12}f(x_1) + \frac{1}{12}f(x_2) + \frac{1}{12}f(x_2) + \frac{1}{15}\frac{1}{12}f(x_2)$$
$$+\frac{1}{15}\frac{1}{12}f(x_2) - 4\frac{1}{15}\frac{1}{6}f(x_2) + 4\frac{1}{15}\frac{1}{12}f(x_3) + \frac{4}{12}f(x_3) + \frac{1}{12}f(x_4) + \frac{1}{15}\frac{1}{12}f(x_4) - \frac{1}{15}\frac{1}{6}f(x_4)$$

or

$$\frac{I}{h} = 0.0777778\,f(x_0) + 0.35555556\,f(x_1) + 0.1333333\,f(x_2) + 0.35555556\,f(x_3) + 0.0777778\,f(x_4)$$

Multiplying by 90*h* gives Boole’s rule

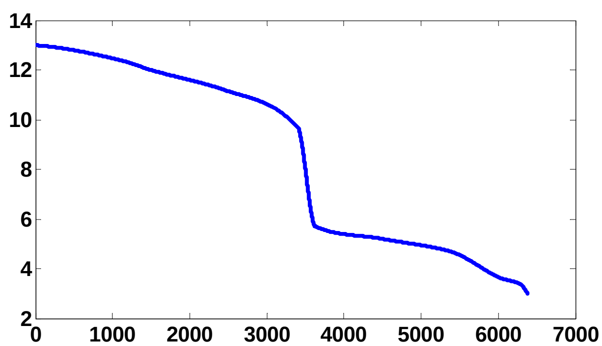
$$I = h\frac{7f(x_0)+32f(x_1)+12f(x_2)+32f(x_3)+7f(x_4)}{90}$$

20.25 Here is a script to solve the problem:

```
clear,clc,clf
format short g
r =[0 1100 1500 2450 3400 3630 4500 5380 6060 6280 6380];
rho = [13 12.4 12 11.2 9.7 5.7 5.2 4.7 3.6 3.4 3];
rp=[min(r):max(r)];
rhop=interp1(r,rho,rp,'pchip');
plot(rp,rhop)
%units
rp=rp*1e3; %convert km to meters
rhop=rhop*1e6/1e3; %convert g/cm3 to kg/m3
Area=4*pi*rp.^2;
rpp=Area.*rhop;
Mass=trapz(rp,rpp)
```

When it is run, the result is

```
Mass =
  6.0905e+024
```



20.26

```
function [q,ea,iter]=romberg(func,a,b,es,maxit,varargin)
% romberg: Romberg integration quadrature
% q = romberg(func,a,b,es,maxit,p1,p2,...):
% Romberg integration.
% input:
% func = name of function to be integrated
% a, b = integration limits
% es = desired relative error (default = 0.000001%)
% maxit = maximum allowable iterations (default = 30)
% pl,p2,... = additional parameters used by func
% output:
% q = integral estimate
% ea = approximate relative error (%)
% iter = number of iterations
if nargin<3,error('at least 3 input arguments required'),end
if nargin<4|isempty(es), es=0.000001;end
if nargin<5|isempty(maxit), maxit=50;end
n = 1;
I(1,1) = trap(func,a,b,n,varargin{:});
iter = 0;
while iter<maxit
  iter = iter+1;
  n = 2^iter;
  I(iter+1,1) = trap(func,a,b,n,varargin{:});
  for k = 2: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);
  end
  ea = abs((I(1,iter+1)-I(2,iter))/I(1,iter+1))*100;
  if ea<=es, break; end
end
q = I(1,iter+1);
```

Script to solve Example 20.1:

```
clear,clc
format long g
fx=@(x) 0.2+25*x-200*x^2+675*x^3-900*x^4+400*x^5;
[q,ea,iter]=romberg(fx,0,0.8)
```

Output:

```
q =
          1.64053333333334
ea =
     0
iter =
     3
```

Script to solve Prob 20.1:

```
clear,clc
format long g
fx=@(x) (x+1/x)^2;
[q,ea,iter]=romberg(fx,1,2,0.5)
```

Output:

```
q =
            4.833470143613
ea =
       0.00580951575223314
iter =
     2
```

20.27

```
function q = quadadapt(f,a,b,tol,varargin)
% Evaluates definite integral of f(x) from a to b
if nargin < 4 | isempty(tol),tol = 1.e-6;end
c = (a + b)/2;
fa = feval(f,a,varargin{:});
fc = feval(f,c,varargin{:});
fb = feval(f,b,varargin{:});
q = quadstep(f, a, b, tol, fa, fc, fb, varargin{:});
end

function q = quadstep(f,a,b,tol,fa,fc,fb,varargin)
% Recursive subfunction used by quadadapt.
h = b - a; c = (a + b)/2;
fd = feval(f,(a+c)/2,varargin{:});
fe = feval(f,(c+b)/2,varargin{:});
q1 = h/6 * (fa + 4*fc + fb);
q2 = h/12 * (fa + 4*fd + 2*fc + 4*fe + fb);
if abs(q2 - q1) <= tol
  q = q2 + (q2 - q1)/15;
else
  qa = quadstep(f, a, c, tol, fa, fd, fc, varargin{:});
  qb = quadstep(f, c, b, tol, fc, fe, fb, varargin{:});
  q = qa + qb;
end
end
```

Script to solve Example 20.1:

```
clear,clc
format long g
fx=@(x) 0.2+25*x-200*x^2+675*x^3-900*x^4+400*x^5;
q = quadadapt(fx,0,0.8)
```

<u>Output:</u>

```
q =
          1.64053333333334
```

Script to solve Prob 20.1:

```
clear,clc
format long g
fx=@(x) (x+1/x)^2;
[q,ea,iter]=romberg(fx,1,2,0.5)
```

<u>Output:</u>

```
q =
          11299.8310550333
```