**19.8**

(a)

Using eq 19.30, I=(2-1)\*(6+5)/2 + (3.25-2)\*(5.5+6)/2 + (4.5-3.25)\*(7+5.5)/2 + (6-4.5)\*(8.5+7)/2 + (7-6)\*(8+8.5)/2 + (8-7)\*(6+8)/2 + (8.5-8)\*(7+6)/2 + (9-8.5)\*(7+7)/2 + (10-9)\*(7+5)/2 = **60.125**

Avg velocity = 60.125/10 = **6.0125**

(b)

Cubic equation with polynomial regression => m=3

N=10, since we have 10 data points

Sum(t\_i) = 1+2+3.25+4.5+6+7+8+8.5+9+10 = 59.25

Sum(v\_i) = 5+6+5.5+7+8.5+8+6+7+7+5 = 65

Sum(t\_i^2) = 1^2 + 2^2 + … + 9^2 + 10^2 = 438.063

Sum(v\_i^2) = 5^2 + 6^2 + .. +7^2 + 5^2 = 435.5

Sum(t\_i^3) = 1^3 + 2^3 + … + 9^3 + 10^3 = 3548.6

Sum(t\_i^4) = 1^4 + 2^4 + … + 9^4 + 10^4 = 30113

Sum(t\_i^5) = 1^5 + 2^5 + … + 9^5 + 10^5= 263010

Sum(t\_i^6) = 1^6 + 2^6 + … + 9^6 + 10^6= 2344600

Sum(t\_i\*v\_i) = 1\*5 + 2\*6 + …+ 9\*7 + 10\*5 = 393.875

Sum((t\_i^2)\*v\_i) = (1^2)\*5 + (2^2)\*6 + …+ (9^2)\*7 + (10^2)\*5 = 2883.6

Sum((t\_i^3)\*v\_i) = (1^3)\*5 + (2^3)\*6 + …+ (9^3)\*7 + (10^3)\*5 = 22934

Substituting these into the normal equations gives the following system of equations

**60.12**

**19.9**

Let

Then

42875000

66150000

89670000

Using Simpson’s 1/3 Rule with n=6

=**3.98 X 10^9**

Let

Then

2.14e9

1.65e9

* **d = 1.1e11 / 3.98e9 = 27.63**

**20.2**

**(a)**

= **20.99**

**(b)**

>> f=@(x) -.055\*x^4+ .86\*x^3 -4.2\*x^2 + 6.3\*x + 2;

>>[q,ea,iter]=romberg(f,0,8,.5)

**q =**

**20.9920**

**So the answer is 20.99** when using Romberg integration with eps=.5%

See the Matlab m-script below:

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);

function I = trap(func,a,b,n,varargin)

% trap: composite trapezoidal rule quadrature

% I = trap(func,a,b,n,pl,p2,...):

% composite trapezoidal rule

% input:

% func = name of function to be integrated

% a, b = integration limits

% n = number of segments (default = 100)

% pl,p2,... = additional parameters used by func

% output:

% I = integral estimate

if nargin<3,error('at least 3 input arguments required'),end

if ~(b>a),error('upper bound must be greater than lower'),end

if nargin<4|isempty(n),n=100;end

x = a; h = (b - a)/n;

s=func(a,varargin{:});

for i = 1 : n-1

x = x + h;

s = s + 2\*func(x,varargin{:});

end

s = s + func(b,varargin{:});

I = (b - a) \* s/(2\*n);

(c)

Must perform change of variable so that the limits are from -1 to 1

Substitute a=0 and b=8 into Eqs 20.22 and 20.23 to yield

and

Substitute these into the original equation to yield

Therefore the right-hand sie is in the form suitable for Gaussian quadrature

Let

Then

19.46

12.17

According to Table 20.1 the three-point formula is

**= 21.00**

**(d)**

>> f=@(x) -.055\*x.^4+ .86\*x.^3 -4.2\*x.^2 + 6.3\*x + 2;

**>> integral=quad(f,0,8)**

**integral =**

**20.9920**

**(Exercise 20.13)**

Use 5 points, so N=5 and h=(60-0)/(5-1) = 15

Let

Then

v(0) =

Similarly, v(15) = 342.3, v(30) = 152.7, v(45) = 38.9, v(60) = 0

* = **9768**

**(Exercise 20.14)**

>> t=[0 .2 .4 .6 .8 1 1.2]; i=[.2e-3 .3683e-3 .3819e-3 .2282e-3 .0486e-3 .0082e-3 .1441e-3];

>> p=polyfit(t,i,5)

p =

-0.0037 0.0115 -0.0104 0.0015 0.0009 0.0002

* Using Matlab to fit these data with a 5th order polynomial gives

Using Simpson’s 1/3 Rule for V(.4) gives

Us Simpson’s Composite 1/3 Rule for V(1.2) gives (7 data points => N=6)

Using Trapezoidal computation with 16 segments, V(.4) = -77

V(1.2) = -83, V(2) = 178.6, V(2.2) = 467.7, V(2.5) = 1303, V(3) = 4592.3

Plot is attached

>> T=[0 .4 1.2 2 2.2 2.5 3]; V=[0 -77 -83 178.6 467.7 1303 4592.3];

>> plot(T,V)

M-script for Trapezoidal computation is below

function I = trap(func,a,b,n,varargin)

% trap: composite trapezoidal rule quadrature

% I = trap(func,a,b,n,pl,p2,...):

% composite trapezoidal rule

% input:

% func = name of function to be integrated

% a, b = integration limits

% n = number of segments (default = 100)

% pl,p2,... = additional parameters used by func

% output:

% I = integral estimate

if nargin<3,error('at least 3 input arguments required'),end

if ~(b>a),error('upper bound must be greater than lower'),end

if nargin<4|isempty(n),n=100;end

x = a; h = (b - a)/n;

s=func(a,varargin{:});

for i = 1 : n-1

x = x + h;

s = s + 2\*func(x,varargin{:});

end

s = s + func(b,varargin{:});

I = (b - a) \* s/(2\*n);