Spring 2005 Math 152
8 Techniques of Integration
8.8 Approximate Integration
Wed, 16/Feb ©2005, Art Belmonte

Summary

We use numerical integration to approximate the value of an integral in the following cases.

- when we cannot compute an antiderivative to our integrand;
- when our functional data is given by a table or graph instead of a formula;
- as a check on exact integration when no symbolic constants are involved.

Approximations to a definite integral $\int_a^b f(x)\,dx$

Partition the interval of integration $[a, b]$ into n equal-length subintervals. The **step size** of these subintervals is $h = \dfrac{b-a}{n}$. The various methods depend on the function used to approximate f on $[a, b]$ (given in parentheses).

Left sum rule (piecewise constant) $L_n = h \sum_{k=0}^{n-1} f(a + kh)$

Right sum rule (piecewise constant) $R_n = h \sum_{k=1}^{n} f(a + kh)$

Midpoint Rule (piecewise constant)

$$M_n = h \sum_{k=0}^{n-1} f\left(a + (k + \tfrac{1}{2})h\right)$$

Trapezoid Rule (piecewise linear)

$$T_n = h\left(\frac{f(a) + f(b)}{2} + \sum_{k=1}^{n-1} f(a + kh)\right)$$

Simpson's Rule (piecewise quadratic; *n* MUST be even!)

$$S_n = \frac{h}{3}\left(\left(f(a) + f(b)\right) + 4 \sum_{i=0}^{n/2-1} f(a + (2i+1)h) + 2 \sum_{j=1}^{n/2-1} f(a + 2jh)\right)$$

Comments

- You are familiar with the Left Sum, Right Sum, and Midpoint Rules from your study of Riemann sums in Calc 1.
- Here's an easy way to remember the Trapezoid Rule.

  step size × (average of endpoint func vals + sum of interior func vals)

- Here's an easy way to remember Simpson's Rule.

  $$\tfrac{1}{3}\text{ step size} \times \left(\sum \overset{\text{endpoint}}{\text{func vals}} + 4 \sum \overset{\text{interior odd}}{\text{func vals}} + 2 \sum \overset{\text{interior even}}{\text{func vals}}\right)$$

- The partition points of the interval $[a, b]$ are

  $$a = x_0 < x_1 < x_2 < \cdots < x_{n-1} < x_n = b.$$

  You start counting at ZERO.

Error Bounds

When we approximate $\int_a^b f(x)\,dx$, we have

$$\text{exact value} = \text{approximate value} + \text{error}.$$

That is, the **error** is the **difference** between the exact and approximate values of the integral as given by one of the numerical integration rules.

Suppose $\left|f''(x)\right| \le K$ and $\left|f^{(4)}(x)\right| \le M$ on $[a, b]$. Then the errors E_T, E_M, and E_S in the Trapezoidal, Midpoint, and Simpson's Rules, respectively, satisfy the following inequalities.

$$\begin{aligned} |E_T| &\le \frac{K(b-a)^3}{12n^2} \\ |E_M| &\le \frac{K(b-a)^3}{24n^2} \\ |E_S| &\le \frac{M(b-a)^5}{180n^4}. \end{aligned}$$

Yes, you need to memorize these!

Hand Examples

508/4

With $n = 4$, use (a) the Trapezoidal Rule and (b) Simpson's Rule to approximate $\int_0^1 \cos\left(x^2\right)\,dx$.

Solution

Remember campers: RADIANS mode in math class!

The step size is $h = \dfrac{b-a}{n} = \dfrac{1-0}{4} = \dfrac{1}{4}$. The partition points are $\left\{0, \frac{1}{4}, \frac{1}{2}, \frac{3}{4}, 1\right\}$.

(a) The Trapezoidal Rule gives

step size $\times$ (average of endpoint func vals + sum of interior func vals)

$$\begin{aligned} & \frac{1}{4}\left(\frac{\cos(0^2)+\cos(1^2)}{2} + \cos((\tfrac{1}{4})^2) + \cos((\tfrac{1}{2})^2) + \cos((\tfrac{3}{4})^2)\right) \\ \approx\ & 0.8958 \end{aligned}$$

(b) Simpson's Rule gives

$$\frac{1}{3}\text{ step size} \times \left(\sum \overset{\text{endpoint}}{\text{func vals}} + 4\sum \overset{\text{interior odd}}{\text{func vals}} + 2\sum \overset{\text{interior even}}{\text{func vals}}\right)$$

$$\begin{aligned} & \tfrac{1}{3}\cdot\tfrac{1}{4}\left(\cos(0^2)+\cos(1^2)+4(\cos((\tfrac{1}{4})^2)+\cos((\tfrac{3}{4})^2))+2\cos((\tfrac{1}{2})^2)\right) \\ \approx\ & 0.9045 \end{aligned}$$

508/13

With $n = 4$, use (a) the Trapezoidal Rule, (b) the Midpoint Rule, and (c) Simpson's Rule to approximate $\int_1^2 e^{1/x}\,dx$. Round your answers to six decimal places.

Solution

The step size is $h = \dfrac{b-a}{n} = \dfrac{2-1}{4} = \dfrac{1}{4}$. The partition points are $\{1, 1.25, 1.50, 1.75, 2\}$.

(a) The Trapezoidal Rule gives

step size $\times$ (average of endpoint func vals + sum of interior func vals)

$$\begin{aligned} & \frac{1}{4}\left(\frac{e^{1/1}+e^{1/2}}{2} + \left(e^{1/1.25}+e^{1/1.50}+e^{1/1.75}\right)\right) \\ \approx\ & 2.031893 \end{aligned}$$

(b) The midpoints of the subintervals are $\left\{1\frac{1}{8}, 1\frac{3}{8}, 1\frac{5}{8}, 1\frac{7}{8}\right\}$. The Midpoint Rule gives

$$\begin{aligned} & \frac{1}{4}\times\left(e^{1/1.125}+e^{1/1.375}+e^{1/1.625}+e^{1/1.875}\right) \\ \approx\ & 2.014207 \end{aligned}$$

(c) Simpson's Rule gives

$$\frac{1}{3}\text{ step size} \times \left(\sum \overset{\text{endpoint}}{\text{func vals}} + 4\sum \overset{\text{interior odd}}{\text{func vals}} + 2\sum \overset{\text{interior even}}{\text{func vals}}\right)$$

$$\begin{aligned} & \tfrac{1}{3}\cdot\tfrac{1}{4}\left(e^{1/1}+e^{1/2}+4\left(e^{1/1.25}+e^{1/1.75}\right)+2e^{1/1.50}\right) \\ \approx\ & 2.020651 \end{aligned}$$

508/20

How large do we have to choose n so that the approximations T_n, M_n, and S_n to the integral $\int_0^1 e^x\,dx$ are accurate to withing 10^{-5}?

Solution

- The integrand is $f(x) = e^x$, whose derivatives are $f^{(m)}(x) = e^x$ for all postive integer values of m. Since e^x is increasing for all x, we have $\max\limits_{0\le x\le 1} f^{(m)}(x) = e^1 = e$.
- Let $a = 0$, $b = 1$, and $K = e$. Then
  $$|E_T| \le \frac{K\,(b-a)^3}{12n^2} = \frac{e}{12n^2} < 10^{-5}$$
  provided that $n > \sqrt{\dfrac{10^5 e}{12}} \approx 150.51$. So choose $n_T = 151$.
- Let $a = 0$, $b = 1$, and $K = e$. Then
  $$|E_M| \le \frac{K\,(b-a)^3}{24n^2} = \frac{e}{24n^2} < 10^{-5}$$
  provided that $n > \sqrt{\dfrac{10^5 e}{24}} \approx 106.42$. So choose $n_M = 107$.
- Let $a = 0$, $b = 1$, and $M = e$. Then
  $$|E_S| \le \frac{M\,(b-a)^5}{180n^4} = \frac{e}{180n^4} < 10^{-5}$$
  provided that $n > \sqrt[4]{\dfrac{10^5 e}{180}} \approx 6.23$ and n is even—necessary for Simpson's Rule! So choose $n_S = 8$. This illustrates the fact that Simpson's Rule is *much* more accurate than the Trapezoidal or Midpoint rules.

508/30

Use Simpson's Rule and the following data to estimate the value of the integral $\int_2^6 y\,dx$.

x	2.0	2.5	3.0	3.5	4.0	4.5	5.0	5.5	6.0
y	9.22	9.01	8.76	8.30	7.52	6.83	7.32	7.69	7.91

Solution

Here $n = 8$ and the step size is $h = \dfrac{b-a}{n} = \dfrac{6-2}{8} = \dfrac{1}{2}$. (Remember, while there are 9 partition points there are 8 subintervals!)

Simpson’s Rule gives

$$\tfrac{1}{3}\text{ step size}\times\left(\sum\overset{\text{endpoint}}{\text{func vals}}+4\sum\overset{\text{interior odd}}{\text{func vals}}+2\sum\overset{\text{interior even}}{\text{func vals}}\right)$$

$$\begin{aligned} & \frac{1}{3}\cdot\frac{1}{2}\Big(9.22+7.91+4\big(9.01+8.30+6.83+7.69\big) \\ & +2\big(8.76+7.52+7.32\big)\Big) \\ \approx\ & 31.94 \end{aligned}$$

MATLAB Examples

I have written five numerical integration routines for you that implement the five rules in the Summary. They are actually just one-line front-ends that call **boxsum**, which is new and improved for 2005.

lsum **rsum** **msum** **trap** **simp**

The routines are “auto-sensing” and “self-correcting.” Or so the marketing boys would have it. . . In other words, the routines can detect whether the function data is continous or discrete. In the latter case, the number of subintervals is determined by the data, overriding whatever you put in for n. Here is the full code for **msum**; others are similar.

```
function s = msum(f,a,b,n,g)
% MSUM:   Midpoint Rule
% Syntax: msum(f,a,b,n,g)
%   f   - integrand:
%           * use function handle for continuous data: @f
%           * use a vector for discrete data: [y1 ... yn]
%   a   - left endpoint of interval
%   b   - right endpoint of interval
%   n   - number of equal-length subintervals
%   g   - 1=illustrate with plot, 0=suppress plot
%
s = boxsum(f,a,b,n,’mid’,g);
```

In the hand examples, you saw how tedious it can be to do numerical integration by hand. With these five routines the procedure is fully automatic as well as illustrative.

I also wrote routines with the same names for the TI-89 calculator as part of the TAMUCALC package. See me if you are interested.

You may also use the built-in MATLAB routine **quad**. It uses an adaptive Simpson’s method that employs *variable* step sizes.

s508x04 [508/4 revisited]

With $n = 4$, use (a) the Trapezoidal Rule and (b) Simpson’s Rule to approximate $\int_0^1 \cos\left(x^2\right)\,dx$.

Solution

The results match those of the hand example. We have displayed six decimal places, which is what Stewart typically does with the answers in the back of the textbook.

```
%
% Stewart 508/4
%
Tn = trap(@f, 0, 1, 4, 0);
fprintf(’%8.6f\n’, Tn)
0.895759
%
Sn = simp(@f, 0, 1, 4, 0);
fprintf(’%8.6f\n’, Sn)
0.904501
%
echo off; diary off
%--------------------
function y = f(x)
y = cos(x.ˆ2);
```

s508x06

With $n = 6$, use (a) the Trapezoidal Rule and (b) Simpson’s Rule to approximate $\int_0^{\pi/4} x\tan x\,dx$.

Solution

```
%
% Stewart 508/6
%
Tn = trap(@f, 0, pi/4, 6, 0);
fprintf(’%8.6f\n’, Tn)
0.189445
%
Sn = simp(@f, 0, pi/4, 6, 0);
fprintf(’%8.6f\n’, Sn)
0.185822
%
echo off; diary off
%--------------------
function y = f(x)
y = x .* tan(x);
```

s508x08

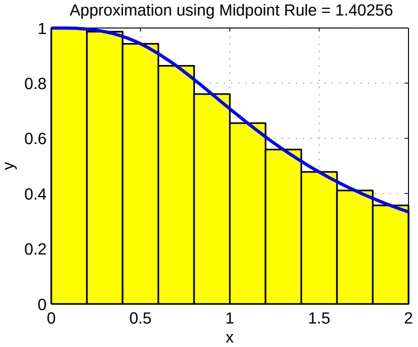
With $n = 10$, use (a) the Trapezoidal Rule, (b) the Midpoint Rule, and (c) Simpson’s Rule to approximate $\displaystyle\int_0^2 \frac{1}{\sqrt{1+x^3}}\,dx$.

Solution

Here we illustrate the Midpoint Rule with a plot. Beauty, eh?

```
% Stewart 508/8
%
Tn = trap(@f, 0, 2, 10, 0);
fprintf(’%8.6f\n’, Tn)
1.401435
%
Mn = msum(@f, 0, 2, 10, 1);
fprintf(’%8.6f\n’, Mn)
```

```
1.402558
%
Sn = simp(@f, 0, 2, 10, 0);
fprintf('%8.6f\n', Sn)
1.402206
echo off; diary off
%--------------------
function y = f(x)
y = 1 ./ sqrt(1 + x.^3);
```



s508x13 [508/13 revisited]

With $n = 4$, use (a) the Trapezoidal Rule, (b) the Midpoint Rule, and (c) Simpson's Rule to approximate $\int_1^2 e^{1/x}\,dx$. Round your answers to six decimal places.

Solution

```
% Stewart 508/13
%
Tn = trap(@f, 1, 2, 4, 0);
fprintf('%8.6f\n', Tn)
2.031893
%
Mn = msum(@f, 1, 2, 4, 0);
fprintf('%8.6f\n', Mn)
2.014207
%
Sn = simp(@f, 1, 2, 4, 0);
fprintf('%8.6f\n', Sn)
2.020651
echo off; diary off
%--------------------
function y = f(x)
y = exp(1./x);
```

s508x17

(a) Find the approximations T_{10} and M_{10} for the integral $\int_0^2 e^{-x^2}\,dx$.

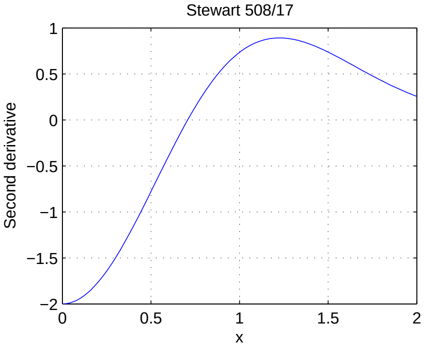
(b) Estimate the errors in the approximations of part (a).

Solution

(a) Here are the requested approximations.

```
% Stewart 508/17
%
Tn = trap(@f, 0, 2, 10, 0);
fprintf('%8.6f\n', Tn)
0.881839
%
Mn = msum(@f, 0, 2, 10, 0);
fprintf('%8.6f\n', Mn)
0.882202
echo off; diary off
%--------------------
function y = f(x)
y = exp(-x.^2);
```

(b) Let's plot the second derivative of $f(x) = e^{-x^2}$. We see that $\left|f''(x)\right| \le K = 2$ on $[0, 2]$.



Therefore,

$$\begin{aligned} |E_T| &\le \frac{K\,(b-a)^3}{12n^2} \\ &= \frac{2\,(2-0)^3}{12\,(10)^2} = \frac{1}{75} \approx 0.01\bar{3} \end{aligned}$$

Similarly,

$$\begin{aligned} |E_M| &\le \frac{K\,(b-a)^3}{24n^2} \\ &= \frac{2\,(2-0)^3}{24\,(10)^2} = \frac{1}{150} \approx 0.00\bar{6} \end{aligned}$$

Here is a MATLAB diary file.

```
%
% Stewart 508/17s
%
syms x
f = exp(-x^2); pretty(f)

                                          2
                                  exp(-x )
D2f = factor(diff(f,x,2)); pretty(D2f)

                                    2            2
                          2 exp(-x ) (-1 + 2 x )
x = linspace(0, 2);
D2f = eval(vectorize(D2f));
plot(x, D2f); grid on
xlabel('x'); ylabel('Second derivative')
title('Stewart 508/17')
%
a = 0; b = 2; K = 2; n = 10;
format long
ET_UB = K * (b-a)^3 / (12*n^2)
ET_UB =
   0.01333333333333
EM_UB = K * (b-a)^3 / (24*n^2)
%
```

```
EM_UB =
   0.00666666666667
format rat
ET_UB, EM_UB
ET_UB =
       1/75
EM_UB =
       1/150
format short
%
echo off; diary off
```

s508x25

Consider the integral $\int_0^1 x^3\,dx$. Find the approximations L_n, R_n, T_n, and M_n for $n = 4$, 8, 16. Then compute the corresponding errors E_L, E_R, E_T, and E_M. (Round your answers to six decimal places.) What observations can you make?

Solution

The exact value of the integral is $\int_0^1 x^3\,dx = \frac{1}{4}x^4\Big|_0^1 = \frac{1}{4} - 0 = \frac{1}{4}$. Here are a script M-file and a diary file. The observations mentioned in the center of page 502 of your Stewart textbook (*q.v.*) are borne out in this problem as well.

```
%
% Stewart 508/25
%
% Preallocate vectors.
I = 1/4; z = zeros(3,1); nn = z;
Ln = z; Rn = z; Tn = z; Mn = z;
EL = z; ER = z; ET = z; EM = z;
% Render the needful.
for k = 1:3
    n = 2^(k+1); nn(k) = n;
    Ln(k) = lsum(@f,0,1,n,0);
    Rn(k) = rsum(@f,0,1,n,0);
    Tn(k) = trap(@f,0,1,n,0);
    Mn(k) = msum(@f,0,1,n,0);
    EL(k) = I - Ln(k);
    ER(k) = I - Rn(k);
    ET(k) = I - Tn(k);
    EM(k) = I - Mn(k);
end
% Make a pretty table.
T1 = [nn Ln Rn Tn Mn]';
T2 = [nn EL ER ET EM]';
disp(' n      Ln         Rn         Tn         Mn ')
disp('---------------------------------------------')
fprintf('%2i %10.6f %10.6f %10.6f %10.6f\n', T1)
fprintf('\n\n')
disp(' n      EL         ER         ET         EM ')
disp('---------------------------------------------')
fprintf('%2i %10.6f %10.6f %10.6f %10.6f\n', T2)
%
echo off; diary off
%
 n      Ln         Rn         Tn         Mn
---------------------------------------------
 4   0.140625   0.390625   0.265625   0.242188
 8   0.191406   0.316406   0.253906   0.248047
16   0.219727   0.282227   0.250977   0.249512
 n      EL         ER         ET         EM
---------------------------------------------
 4   0.109375  -0.140625  -0.015625   0.007812
 8   0.058594  -0.066406  -0.003906   0.001953

16   0.030273  -0.032227  -0.000977   0.000488
```

s508x30 [508/30 revisited]

Use Simpson's Rule and the following data to estimate the value of the integral $\int_2^6 y\,dx$.

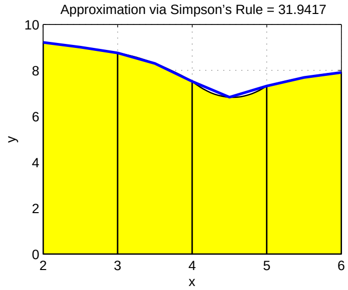
x	2.0	2.5	3.0	3.5	4.0	4.5	5.0	5.5	6.0
y	9.22	9.01	8.76	8.30	7.52	6.83	7.32	7.69	7.91

Solution

In this exercise we have discrete function values from a table instead of a formula like the other exercises. (Or we might have a graph from which we read off function values.) The answer agrees with the one from the hand example.

```
%
% Stewart 508/30
%
fv = [9.22 9.01 8.76 8.30 7.52 6.83 7.32 7.69 7.91];
Sn = simp(fv, 2, 6, 1776, 1)
Sn =
   31.9417
%
% Actually, n = 8, NOT 1776. The routine automatically
% counts the number of function values and corrects this
% so that you don't have to do so. This avoids mistakes,
% especially when you have a large number of data.
%
echo off; diary off
```

Here's a plot of the area under the curve, namely $\int_2^6 y\,dx$, as estimated by S_8.



s508x34

A wooden log 10 meters long is cut at 1-meter intervals and its cross-sectional areas A (at a distance x from the end of the log) are listed in the following table. Use Simpson's Rule to estimate the volume $V = \int_0^{10} A\,dx$ of the log.

x (m)	0	1	2	3	4	5	6	7	8	9	10
A (m^2)	0.68	0.65	0.64	0.61	0.58	0.59	0.53	0.55	0.52	0.50	0.48

Solution

The volume is approximately 5.77 m^3. (A diary file appears on the next page.)

```
%
% Stewart 509/34
%
fv = [0.68 0.65 0.64 0.61 0.58 0.59 ...
    0.53 0.55 0.52 0.50 0.48];
Sn = simp(fv, 0, 10, 10, 0)
Sn =
    5.7667
%
echo off; diary off
```

s508x23

Consider $I = \int_0^{2\pi} f(x)\,dx$, where $f(x) = e^{\cos x}$.

(a) Use a graph to get a good upper bound K for $\max\limits_{0 \le x \le 2\pi} \left| f''(x) \right|$.

(b) Use M_{10} (the Midpoint Rule approximation with $n = 10$) to approximate I.

(c) Use part (a) to estimate the error in part (b).

(d) Use MATLAB’s **quad** command to approximate I. Code $g(x) = e^{\cos x}$ as a function M-file *g.m*.

(e) How does the actual error $I - M_{10}$ compare with the estimate in part (c)?

(f) Use a graph to get a good upper bound M for $\max\limits_{0 \le x \le 2\pi} \left| f^{(4)}(x) \right|$.

(g) Use S_{10} (the Simpson’s Rule approximation with $n = 10$) to approximate I.

(h) Use part (f) to estimate the error in part (g).

(i) How does the actual error $I - S_{10}$ compare with the estimate in part (h)?

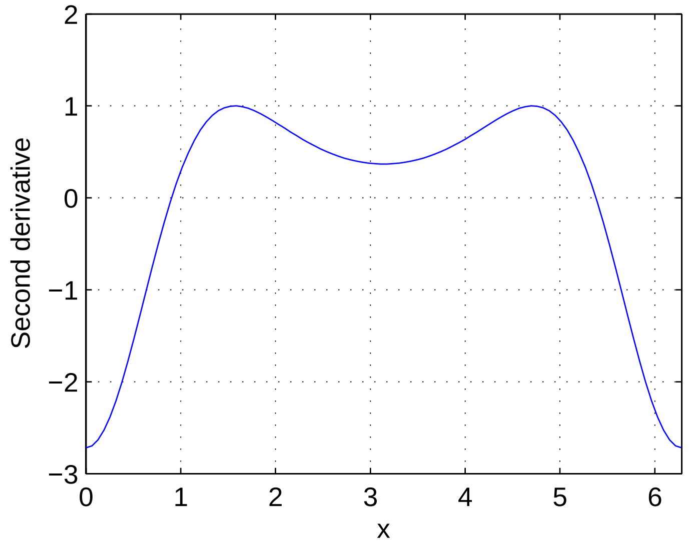
(j) How large should n be to guarantee that the size of the error in using S_n is less than 10^{-4}?

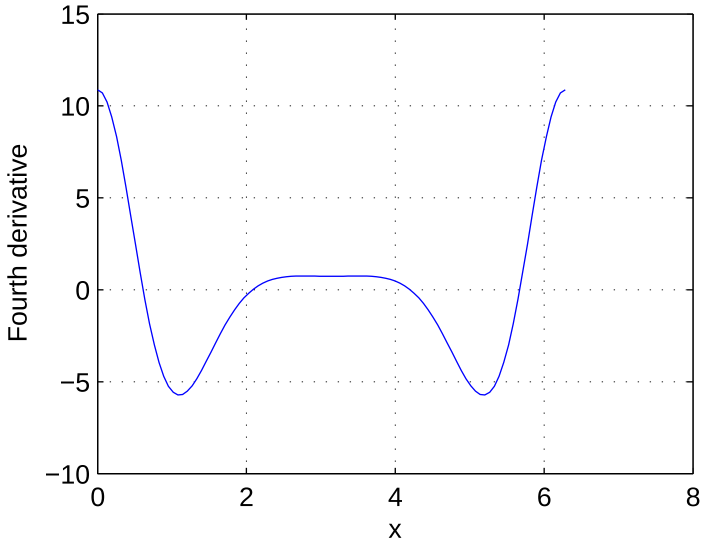
Solution

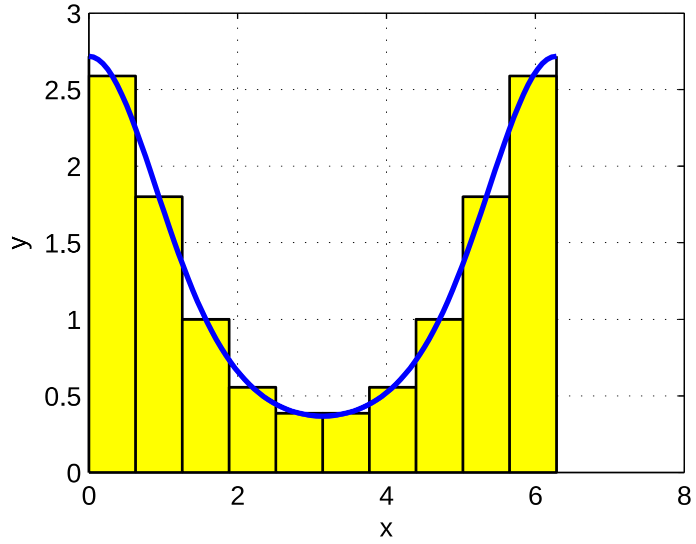
Here is a script M-file followed by formated output in a diary file and graphs.

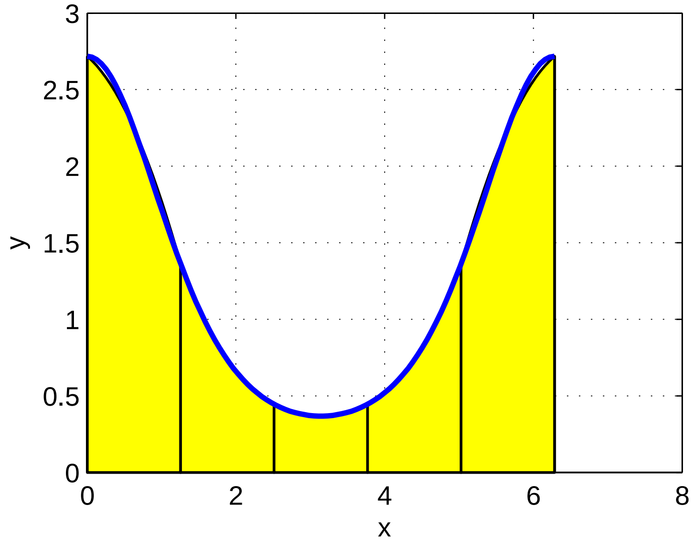
```
%-----------------------------------------------
delete s508x23.txt; diary s508x23.txt
clear; clc; close all; echo off
%
% Stewart 508/23
%
a = 0; b = 2*pi; n = 10;
syms x
f = exp(cos(x)); disp(’function’); pretty(f)
D2f = diff(f,x,2); disp(’2nd derivative’); pretty(D2f)
D4f = diff(f,x,4); disp(’4th derivative’); pretty(D4f)
% (a)
x = linspace(a, b);
D2f = eval(vectorize(D2f));
plot(x, D2f); grid on
xlabel(’x’); ylabel(’Second derivative’)
title(’Stewart 508/23: Plot of second derivative’)
axis([a b -3 2])
K = max(abs(D2f)) + eps;
fprintf(’K = %f\n’, K)
% (b)
figure
M_10 = msum(@g, a, b, n, 1);
fprintf(’M_10 = %11.9f\n’, M_10)
% (c)
M_10_AEB = K*(b-a)ˆ3 / (23*nˆ2);
fprintf(’M_10 error estimate = %7.5f\n’, M_10_AEB)
% (d)
I = quad(@g, a, b, 1e-9);
fprintf(’I via quad = %11.9f\n’, I)
% (e)
err_M_10 = I - M_10;
fprintf(’Actual error in M_10 = %10.7g\n’, err_M_10)
% (f)
D4f = eval(vectorize(D4f));
figure
plot(x, D4f); grid on
xlabel(’x’); ylabel(’Fourth derivative’)
title(’Stewart 508/23: Plot of fourth derivative’)
M = max(abs(D4f)) + eps;
fprintf(’M = %f\n’, M)
% (g)
figure
S_10 = simp(@g, a, b, n, 1);
fprintf(’S_10 = %11.9f\n’, S_10)
% (h)
S_10_AEB = M*(b-a)ˆ5 / (180*nˆ4);
fprintf(’S_10 error estimate = %8.6f\n’, S_10_AEB)
% (i)
err_S_10 = I - S_10;
fprintf(’Actual error in S_10 = %10.7g\n’, err_S_10)
% (j)
N = ceil( (M*(b-a)ˆ5 * 1e4 / 180)ˆ(1/4) )
%
echo off; diary off
%-----------------------------------------------
our function

                                  exp(cos(x))
2nd derivative

                                               2
                   -cos(x) exp(cos(x)) + sin(x)  exp(cos(x))
4th derivative

                             2                        2
cos(x) exp(cos(x)) - 4 sin(x)  exp(cos(x)) + 3 cos(x)  exp(cos(x))
                          2                     4
         - 6 cos(x) sin(x)  exp(cos(x)) + sin(x)  exp(cos(x))
K = 2.718282
M_10 = 7.954926518
M_10 error estimate = 0.29316
I via quad = 7.954926521
Actual error in M_10 = 3.412742e-09
M = 10.873127
S_10 = 7.953789422
S_10 error estimate = 0.059154
Actual error in S_10 = 0.001137099
N =

    50
```

(The plots are on the following page.)


Stewart 508/23: Plot of second derivative


Stewart 508/23: Plot of fourth derivative


Approximation using Midpoint Rule = 7.95493


Approximation via Simpson's Rule = 7.95379