Leoza Kabir
V00840048
Instructor: Rich Little

CSC 349A: Numerical Analysis
Assignment 6

2018-11-27

---

1. (a) (2 points) Determine the Lagrange form of the interpolating polynomial P (x) that interpolates a function f(x) at x = 0, 2h and 3h, where h > 0. (Multiply the linear factors together, but leave P (x) as a sum of 3 quadratics in the variable x.)

DELIVERABLES: All your work in constructing the polynomial. This is to be done by hand not MATLAB.

1) a) $P(x) = \sum_{i=0}^{2} L_i(f(x)) f(x_i) = L_0(x) f(x_0) + L_1 f(x_1) + L_2(x_2) f(x_2)$.

$$L_0(x) = \frac{(x-x_1)(x-x_2)}{(x_0-x_1)(x_0-x_2)} = \frac{(x-2h)(x-3h)}{(0-2h)(0-3h)} = \frac{x^2-5xh+6h^2}{6h^2}$$

$$L_1(x) = \frac{(x-x_0)(x-x_2)}{(x_1-x_0)(x_1-x_2)} = \frac{(x-0)(x-3h)}{(2h-0)(2h-3h)} = \frac{x^2-3xh}{-2h^2}$$

$$L_2(x) = \frac{(x-x_0)(x-x_1)}{(x_2-x_0)(x_2-x_1)} = \frac{(x-0)(x-2h)}{(3h-0)(3h-2h)} = \frac{x^2-2xh}{3h^2}$$

$$P(x) = \frac{x^2-5xh+6h^2}{6h^2} f(0) + \frac{x^2-3xh}{-2h^2} f(2h) + \frac{x^2-2xh}{3h^2} f(3h).$$

(b) (2 points) Derive the quadrature formula of the form

$$a_0 f(0) + a_1 f(2h) + a_2 f(3h)$$

for approximating $I = \int_0^{3h} f(x)dx$ that results from approximating the integral I by $I \approx \int_0^{3h} P(x)dx$.

Note: if you know only 3 function values of f(x) and they are at 3 unequally- spaced points 0, 2h and 3h, then this kind of quadrature formula can be used to approximate I.

DELIVERABLES: All your work in deriving the quadrature formula.

b) $I \approx \int_0^{3h} P(x)dx = \int_0^{3h} \frac{x^2-5xh+6h^2}{6h^2} f(0) + \frac{x^2-3xh}{-2h^2} f(2h) + \frac{x^2-2xh}{3h^2} (3h)$

$$I = \int_0^{2h} f(x)dx + \int_{2h}^{3h} f(x)dx = \frac{h}{2}[f(0)+f(2h)] + \frac{h}{2}[f(2h)+f(3h)]$$

$$= \frac{\cancel{2}h-0}{\cancel{2}}[f(0)+f(2h)] + \frac{3h-2h}{2}[f(2h)+f(3h)]$$

$$= h[f(0)+f(2h)] + \frac{h}{2}[f(2h)+f(3h)]$$

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

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

(c) (2 points) Suppose that you know only the following function values of f(x):

x	f(x)
0.0	0.5
0.24	0.50727
0.36	0.51656

Use the quadrature formula from (b) to approximate $\int_0^{0.36} f(x)dx$

Note: the above data corresponds to the function f(x) = 1/cos(x), and the exact value is about $\int_0^{0.36} f(x)dx$ = 0.1819695.... Use this information only to assessthe accuracy of your computed approximation. If you do not obtain a fairly good approximation in (c), then your answer in (b) is incorrect. That is, the relative error between your answer and the true answer should be way less than 1%.

c) $h = 0.36 - 0.24 = 0.12$

$2h = 0.24$
$h = \frac{0.24}{2} = 0.12$

$3h = 0.36$
$h = 0.12$

$$\int_0^{0.36} f(x)dx = hf(0) + \frac{3h}{2}f(2h) + \frac{h}{2}f(3h)$$

$$= 0.12(0.5) + \frac{3(0.12)}{2}(0.50727) + \frac{0.12}{2}(0.51656)$$

$$= 0.06 + 0.0913086 + 0.0309936 = 0.1823022$$

$$|E_t| = \left|\frac{0.1819695 - 0.1823022}{0.1819695}\right| = 0.0018283 \approx 0.18\%$$

2. (a) (4 points) Determine the degree of precision of the quadrature formula

2)a) $\frac{5}{9}f\left(-\sqrt{3/5}\right)+\frac{8}{9}f(0)+\frac{5}{9}f\left(\sqrt{3/5}\right)$

$\int_{-1}^{1} f(x)\,dx$

$f(x)$	$\int_{-1}^{1} f(x)dx$	$5/9 f(-\sqrt{3/5})+8/9 f(0)+5/9 f(\sqrt{3/5})$
1	$1-(-1)=2$	$5/9(1)+8/9(1)+5/9(1)=2$
x	$\frac{1^2-(-1)^2}{2}=0$	$5/9(-\sqrt{3/5})+8/9(0)+5/9\sqrt{3/5}=0$
x^2	$\frac{1^3-(-1)^3}{3}=\frac{2}{3}$	$5/9(-\sqrt{3/5})^2+8/9(0)^2+5/9(\sqrt{3/5})^2=2/3$
x^3	$\frac{1^4-(-1)^4}{4}=0$	$5/9(-\sqrt{3/5})^3+8/9(0)^3+5/9(\sqrt{3/5})^3=0$
x^4	$\frac{1^5-(-1)^5}{5}=2/5$	$5/9(-\sqrt{3/5})^4+8/9(0)^4+5/9(\sqrt{3/5})^4=2/5$
x^5	$\frac{1^6-(-1)^6}{6}=0$	$5/9(-\sqrt{3/5})^5+8/9(0)^5+5/9(\sqrt{3/5})^5=0$
x^6	$\frac{1^7-(-1)^7}{7}=2/7$	$5/9(-\sqrt{3/5})^6+8/9(0)^6+5/9(\sqrt{3/5})^6=6/25\neq 2/7$

The degree of precision is 5

(b) (2 points) Use the quadrature formula in (a) to approximate

$$\int^{1} e^{-x}\sqrt{x+2}\,dx.$$

b) $\frac{5}{9} f(-\sqrt{3/5}) + \frac{8}{9} f(0) + \frac{5}{9} f(\sqrt{3/5})$

$\frac{5}{9}\left(e^{-(-\sqrt{3/5})}\sqrt{(-\sqrt{3/5})+2}\right) + 8/9\left(e^{0}\sqrt{0+2}\right) + 5/9\left(e^{-\sqrt{3/5}}\sqrt{\sqrt{3/5}+2}\right)$

$= 1.334135099 + 1.257078722 + 0,426505234$

$= 3.017934946$

3. (a) (4 points) Fill in the blanks in the following MATLAB function M-file trap so that it implements the composite trapezoidal rule, using a sequence of m = 1, 2, 4, 8, 16, . . . trapezoids, to approximate $\int_a^b f(x)dx$.
This M-file uses the MATLAB built-in function trapz. If

$$x = (x_1, x_2, \ldots, x_{m+1}) \text{ and } y = (f(x_1), f(x_2), \ldots, f(x_{m+1})),$$

then the execution of

z = trapz(x, y)

approximates

$$\int_{x_1}^{x_{m+1}} f(x)dx$$

using the composite trapezoidal rule (with m trapezoids). (Note that the entries of x are labeled starting at x_1 because MATLAB does not allow an index of 0.) The function M-file trap has 4 input parameters:

- **a and b (the lower and upper limits of the integral);**
- **maxiter (the maximum number of iterations that are allowed; the above program will implement the composite trapezoidal rule with a sequence of 1, 2, 4, 8, 16, . . . trapezoids, up to a maximum of $2^{maxiter}$ trapezoids);**
- **tol(a tolerance for testing the relative error of the computed approximation). If the difference between 2 successive approximations (with respect to relative error) is < tol, then the algorithm terminates.**

You also must define and store a MATLAB function M-file

y = f(x)

which must be written to accept a vector xas the argument. Note that this means that you must use arithmetic operators like .* , .^ , ./ and so on, rather than just *, ^, /.

```
function trap(a, b, maxiter, tol)
m = 1;
x = linspace(a, b, m+1);
y = f(x);
approx = trapz(x, y);
disp('    m     integral approximation');
fprintf(' %5.0f %16.10f \n ', m, approx);
```

for i = 1 : maxiter

m = ______________ ;

oldapprox = ____________;____________________________

x = linspace (___ , ___ , ____) ;
y = f(x);
approx = trapz(x, y);
fprintf(' %5.0f %16.10f \n ', m, approx);
if abs(____) < tol
return

end
end

fprintf('Did not converge in %g iterations', maxiter)

DELIVERABLES: A copy of your function trap. Feel free to add an extra parameter f at the end of parameter list to make your function calls simpler.

```
function trap(a,b,maxiter,tol,f)
m=1;
x=linspace(a,b,m+1);
y=f(x);
approx=trapz(x,y);
disp('          m       integral approximation');
fprintf('  %5.0f  %16.10f \n  ' , m, approx);
for i=1: maxiter
    m=m*2;
    oldapprox=approx;
    x=linspace(a,b,m+1);
    y=f(x);
    approx=trapz(x,y);
    fprintf('  %f.0f  %16.10f  \n  ',m,approx);
    if abs((approx-oldapprox)/approx) < tol
        return
    end
end
fprintf('Did not converge in %g iterations',maxiter)
```

(b) (4 points)

Use the above MATLAB script to approximate the following two integrals:

$$\int_{0.1}^{3} \sin(1/x)dx \text{ using maxiter = 20 and tol} = 10^{-6},$$

$$\int_{0}^{1} \frac{e^{3x}}{\sqrt{x^3+1}}dx \text{ using maxiter = 20 and tol} = 10^{-10}.$$

DELIVERABLES: The MATLAB calls and output for each. If you use function files to evaluate each of the functions I also want the .m files for those.

i.
```
function[fh]=func(x)
        fh=sin(1./x);
    end
>> a=0.1;
>> b=3;
>> maxiter=20;
>> tol=10.^-6;
>> trap(a,b,maxiter,tol,@func)
         m       integral approximation
      1      -0.3143983004
    2.000000.0f      0.7147254605
    4.000000.0f      1.3447434609
    8.000000.0f      1.5589483255
    16.000000.0f      1.4776583126
    32.000000.0f      1.4679626280
    64.000000.0f      1.5197926883
    128.000000.0f      1.5355585774
    256.000000.0f      1.5386514853
    512.000000.0f      1.5393496800
    1024.000000.0f      1.5395196356
    2048.000000.0f      1.5395618423
    4096.000000.0f      1.5395723764
    8192.000000.0f      1.5395750089
    16384.000000.0f      1.5395756669
```

ii.
```
function[fh]=func1(x)
b=sqrt((x.^3)+1);
fh= exp(3*x)./b;
end

  >> a=0;
>> b=1;
>> maxiter=20;
>> tol=10.^-10;
>> trap(a,b,maxiter,tol,@func1)
         m       integral approximation
      1       7.6013096811
    2.000000.0f      5.9133433291
    4.000000.0f      5.4710046573
    8.000000.0f      5.3585418274
    16.000000.0f      5.3303053079
    32.000000.0f      5.3232385483
    64.000000.0f      5.3214713803
    128.000000.0f      5.3210295584
    256.000000.0f      5.3209191010
    512.000000.0f      5.3208914866
    1024.000000.0f      5.3208845830
    2048.000000.0f      5.3208828571
    4096.000000.0f      5.3208824256
    8192.000000.0f      5.3208823177
    16384.000000.0f      5.3208822908
    32768.000000.0f      5.3208822840
    65536.000000.0f      5.3208822823
    131072.000000.0f      5.3208822819
```