CS / MATH 4334 : Numerical Analysis
Homework Assignment 4

Matthew McMillian
mgm160130@utdallas.edu

November 28, 2018

MatLab Problems

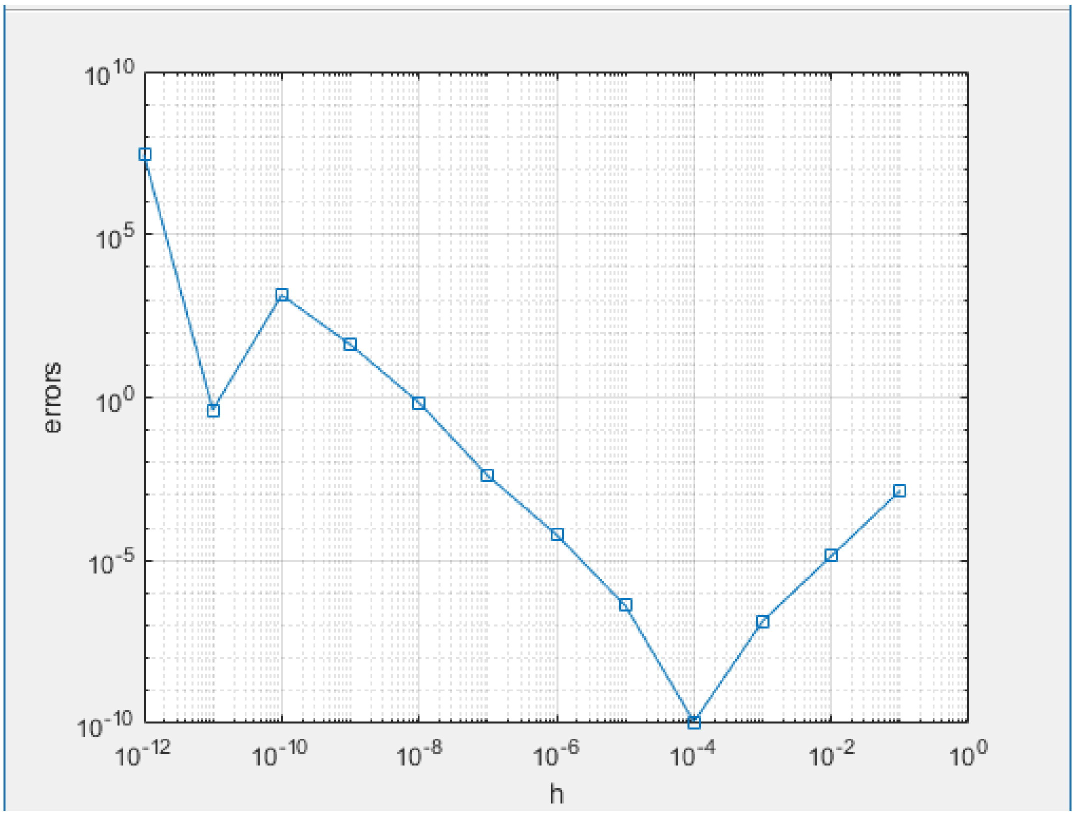
Problem 1 : q1.m

```
format long e

% How many h's we have
range = 12;

% Creates evenely spaced points for H
h = [10^-1, 10^-2, 10^-3, 10^-4, 10^-5, 10^-6, 10^-7, 10^-8,
    10^-9, 10^-10, 10^-11, 10^-12];
approx = zeros(range,1);
errors = zeros(range,1);

% Calculates the approx values of h
for i=1:range
    approx(i) = ((cos(2 - 2*h(i)) - 2*cos(2) + cos(2 + 2*h(i))
        ) / (4*h(i)^2));
end

% Calculates the errors for the approx values
for i=1:range
    errors(i) = abs(-cos(2) - approx(i));
end



% Testing our h values locally (they are close)
[m, l] = min(errors);
h(l);
(6*(10^-16)/1)^.25;

sprintf('We notice that as h gets smaller, the error decreases
    . However, due to a combonation of truncation, roundoff
    errors and loss of signifigance we see weird behavious from
     h as it gets smaller and smaller. Our h value that we got
    was close our expected h value form problem 2.')

loglog(h, errors, '-s')
xlabel('h')
ylabel('errors')
grid on
```

>> q1.m

ans =

'We notice that as h gets smaller, the error decreases. However, due to a combonation of truncation, roundoff errors and loss of signifigance we see weird behavious from h as it gets smaller and smaller. Our h value that we got was close our expected h value form problem 2.'



Problem 2 : q2.m

```
func = @(x) 20*exp((-(x-60)^2)/50)/power((2*pi),.5);

trapfun(10000, 0, 69, func)

simpfun(100, 0, 69, func)

sprintf('a) 96.41 students would fail this class based on
    trapezoid rule\nb) 96.41 students would fail based on
    simpsons rule\nNo, we definitely would not want to take
    this class')
```

Problem 2 : trapfun.m

```
function int = trapfun(m,a,b,func)
    int = func(a) + func(b);
    x = linspace(a,b,m);
    h = (b-a)/m;

    for i=1:m-1
        int = int + 2*(func(x(i+1)));
    end

    int = (h/2)*int;

end
```

Problem 2 : simpfun.m

```
function int = simpfun(m,a,b,func)
    int = func(a) + func(b);
    x = linspace(a,b,2*m+1);
    h = (b-a)/(2*m);

    for i = 1:m %1 3 5 7 9 -> 2 4 6 8 10
        int = int + 4*func(x(2*(i)));
    end

    for i = 1:m-1 % 2 4 6 8 -> 3 5 7 9
        int = int + 2*func(x(2*(i)+1));
    end

```

```
    int = (h/3)*int;
end
```

>> q2.m

```
ans =
9.640822025822844e+01

ans =
9.640696765218401e+01

ans =
'a) 96.41 students would fail this class based on trapezoid rule.
b) 96.41 students would fail based on simpsons rule.
No, we definitely would not want to take this class'
```

Problem 3 : q3.m

```
format long e

func = @(x) exp((x^2));
rombergmod(func, 0, 1, (0.5*10^-12))

% order 16, since order is 2*colnum, 127 fevals?no
```

Problem 3 : rombergmod.m

```
% Program 5.1 Romberg integration
% Computes approximation to definite integral
% Inputs: Matlab inline function specifying integrand f,
%     a,b integration interval, n=number of rows
% Output: Romberg tableau r
function r=rombergmod(f,a,b,error)
n = 100;
h=(b-a)./(2.^(0:n-1));
fa = feval(f,a);
fb = feval(f,b);
r(1,1)=(b-a)*(fa+fb)/2;
fevals = 2;
for j=2:n
  subtotal = 0;
  for i=1:2^(j-2)
    fmid = feval(f,a+(2*i-1)*h(j));
    fevals = fevals + 1;
    subtotal = subtotal + fmid;
  end
  r(j,1) = r(j-1,1)/2+h(j)*subtotal;
  for k=2:j
    r(j,k) = (4^(k-1)*r(j,k-1)-r(j-1,k-1))/(4^(k-1)-1);
  end
  if(abs(r(j,j) - r(j-1,j-1)) < error)
    break
  end
end
fevals
```

>> q3.m

There are 129 function evals, and the approximation is of order 16.

```
fevals =

   127


ans =

  Columns 1 through 6

    1.859140914229523e+00                       0                       0                       0                       0                       0
    1.571583165458632e+00   1.475730582535002e+00                       0                       0                       0                       0
    1.490678861698856e+00   1.463710760445597e+00   1.462909438972970e+00                       0                       0                       0
    1.469712276429665e+00   1.462723414673269e+00   1.462657591621780e+00   1.462653594044777e+00                       0                       0
    1.464420310149482e+00   1.462656321389421e+00   1.462651848503831e+00   1.462651757343228e+00   1.462651750140477e+00                       0
    1.463094102606428e+00   1.462652033425411e+00   1.462651747561143e+00   1.462651745958878e+00   1.462651745914234e+00   1.462651745910103e+00
    1.462762348577727e+00   1.462651763901493e+00   1.462651745933232e+00   1.462651745907392e+00   1.462651745907190e+00   1.462651745907183e+00
    1.462679397418588e+00   1.462651747032208e+00   1.462651745907589e+00   1.462651745907182e+00   1.462651745907181e+00   1.462651745907181e+00

  Columns 7 through 8

                        0                       0
                        0                       0
                        0                       0
                        0                       0
                        0                       0
                        0                       0
    1.462651745907183e+00                       0
    1.462651745907181e+00   1.462651745907181e+00
```