Numerical Calculus, 8th Laboratory

1. a) Approximate the integral

$$I = \int_0^1 f(x)dx, \qquad \text{for } f(x) = \frac{2}{1+x^2},$$

using trapezium formula.

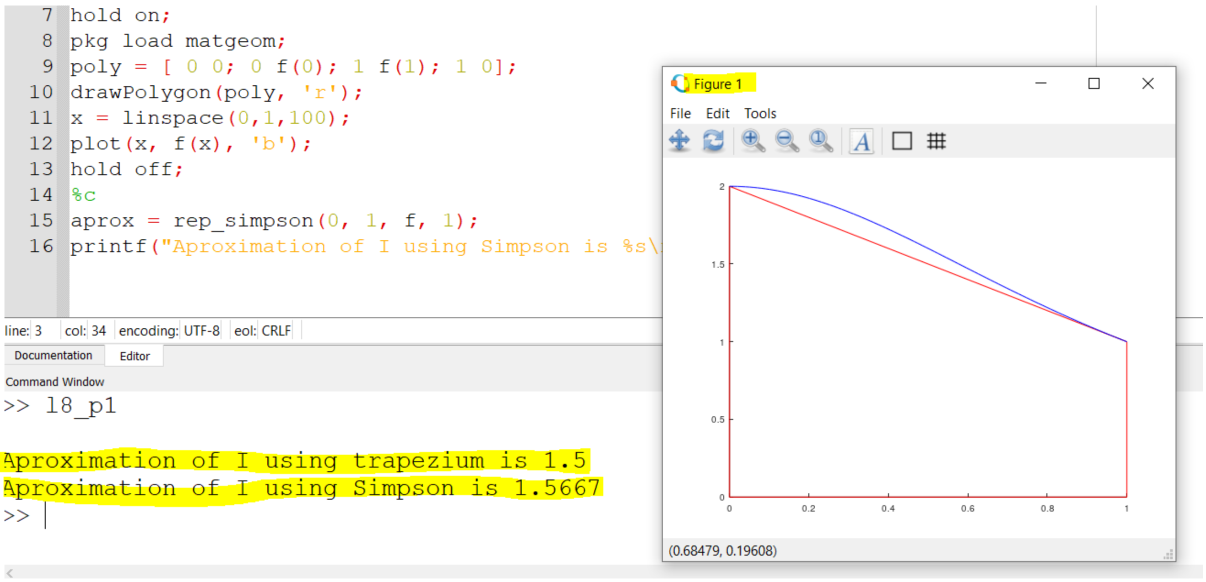
b) Plot the graph of the function f and the graph of the trapezium with vertices $(0,0), (0, f(0)), (1, f(1))$ and $(1, 0)$.

c) Approximate the integral I using Simpson's formula.

Code

```
%a
f=@(x) 2./(1+x.^2);
aprox = rep_trapezium(0, 1, f,1);
printf("Aproximation of I using trapezium is %f\n", num2str(aprox));
%b
clf;
hold on;
pkg load matgeom;
poly = [ 0 0; 0 f(0); 1 f(1); 1 0];
drawPolygon(poly, 'r');
x = linspace(0,1,100);
plot(x, f(x), 'b');
hold off;
%c
aprox = rep_simpson(0, 1, f, 1);
printf("Aproximation of I using Simpson is %s\n", num2str(aprox));
```

Results and graph



Alb Dan-Cristian, 931

2. Approximate the following double integral

$$\int_{1.4}^{2} \int_{1}^{1.5} \ln(x + 2y)dydx$$

using trapezium formula for double integrals, given in (1). (*Result*: 0.4295545)

Code and result:

```
f = @(x, y) log(x+2*y);
aprox = doub_trap(1.4, 2, 1, 1.5, f);
printf("Aproximation is: \n");
disp(aprox);
```

line: 1 col: 20 encoding: UTF-8 eol: CRLF

Documentation Editor

Command Window

```
Aproximation is:
0.429062791946508
```

3. Evaluate the integral that arises in electrical field theory:

$$H(p, r) = \frac{60r}{r^2 - p^2} \int_0^{2\pi} \left[1 - \left(\frac{p}{r}\right)^2 \sin x\right]^{1/2} dx,$$

for $r = 110$, $p = 75$, using the repeated trapezium formula for two given values of n. (*Result:* 6.3131)

Code:

```
n = [1 2 3 4 5];
f = @(x) (1-(75/110).^2*sin(x)).^(1/2);
for ind=1:length(n)
  aprox = rep_trapezium(0, 2*pi, f, n(ind));
  aprox = aprox*60*110./(110.^2-75.^2);
  printf("Aprximation for n=%d is:\n", n(ind));
  disp(aprox);
endfor
```

Results for n in {1,2,3,4,5}

```
Aprximation for n=1 is:
6.40448232083170l
Aprximation for n=2 is:
6.40448232083170l
Aprximation for n=3 is:
6.31317991001801
Aprximation for n=4 is:
6.31136484871835
Aprximation for n=5 is:
6.31312408082650
>>
```

4. Find the smallest value of n that gives an approximation of the integral $\int_1^2 x\ln(x)dx$ which is correct to three decimals, using the repeated trapezium formula. Apply the repeated trapezium formula for the obtained value of n to approximate the integral. (*Result: 0.636294368858383*)

Solution idea: start from n=1 and stop when finding an aproximation that satisfies the condition. In our case: first 3 decimals to be equal.

Code and result:

```
% find smallest n sucht that the aproximation is correct to three decimals
% aproximation using rep. trapezium formula
f = @(x) x.*log(x);
n=1;
ok = false;
precission = 3;
result = 0.6362943688583;
real_aprox = floor((result-floor(result))*10.^precission);
while !ok
  aproximation = rep_trapezium(1, 2, f, n);
  prec = aproximation - floor(aproximation);
  prec = floor(aproximation*10.^precission);
  if prec == real_aprox && floor(result) == floor(aproximation)
    ok = true;
    printf("The value of n is: %d and aproximation is:\n", n);
    disp(aproximation);
  endif
  n+=1;
endwhile
```

line: 15 | col: 34 | encoding: UTF-8 | eol: CRLF

Documentation | Editor

Command Window

```
The value of n is: 10 and aproximation is:
0.636871879788736
```

5. Evaluate the integral

$$\int_0^{\pi} \frac{dx}{4 + \sin 20x}$$

using the repeated Simpson's formula for $n = 10$ and 30. (*Result: 0.8111579*)

Code and results:

```
f = @(x) 1./(4+sin(20*x));
n = [10 30];
for ind=1:length(n)
  aprox = rep_simpson(0, pi, f, n(ind));
  printf("Aproximation for n=%d is\n", n(ind));
  disp(aprox);
endfor
```

line: 7 | col: 7 | encoding: UTF-8 | eol: CRLF

Documentation | Editor

Command Window

```
Aproximation for n=10 is
0.785398163397448
Aproximation for n=30 is
0.811148922853102
```

6. The error function $erf(x)$ is defined by

$$erf(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt.$$

Use the repeated Simpson's formula to evaluate $erf(0.5)$ with $n = 4$ and $n = 10$. Estimate the accuracy of your result and compare with the correct value $erf(0.5) = 0.520499876$.

Code:

```

x = [0.5];
n = [4 10];
f = @(t) exp(-t.^2);
real_value = [0.5204998];
for i=1:length(x)
  printf("------------------\n");
  printf("Aproximations for x=%d\n", x(i));
  for j=1:length(n)
    aproximation = rep_simpson(0, x(i), f, n(j));
    aproximation = aproximation*2./sqrt(pi);
    m1 = max(real_value(i), aproximation);
    m2 = min(real_value(i), aproximation);
    acc = m2./m1;
    acc = acc * 100;
    printf("Aproximation for n=%d is:\n", n(j));
    disp(aproximation);
    printf("Accuracy is: %f%%\n", acc);
  endfor
   printf("------------------\n");
endfor
```

Results:

```
------------------
Aproximations for x=0.5
Aproximation for n=4 is:
0.520500251716589
Accuracy is: 99.999913%
Aproximation for n=10 is:
0.520499887354270
Accuracy is: 99.999983%
------------------
```