ISTANBUL TECHNICAL UNIVERSITY, MATHEMATICS DEPARTMENT
NUMERICAL METHODS FOR PARTIAL DIFFERENTIAL EQUATIONS I

Homework I

Haydar Altuğ Yıldırım

509161108

October 10, 2016

1 Numerical and Analytical Derivation of Functions and Relative Errors

Find the numerical(forward, backward and central) and analytical solutions of the functions $u_1 = sin(x)$ and $u_2 = e^{x^2}$ at the point $x = 1$ and calculate the percentage error.

u_1	forward differences				backward differences				central differences			
Δ x	analytic	numeric	error(%)	p(ε)	analytic	numeric	error(%)	p(ε)	analytic	numeric	error(%)	p(ε)
0.1	0.5403	0.4973	7.9585	-0.9008	0.5403	0.5814	7.6068	-0.8812	0.5403	0.5393	0.1665	0.7783
0.05	0.5403	0.519	3.9422	-0.4579	0.5403	0.5611	3.8497	-0.4499	0.5403	0.54	5.5524E-2	0.9650
0.01	0.5403	0.536	0.7958	4.9583E-2	0.5403	0.5444	0.7773	5.4692E-2	0.5403	0.5402	1.8508E-2	0.8663
5E-3	0.5403	0.5381	0.4071	0.1695	0.5403	0.5423	0.3886	0.1783	0.5403	0.5403	0	
1E-3	0.5403	0.5397	9.2541E-2	0.3445	0.5403	0.5406	7.4032E-2	0.3768	0.5403	0.5403	0	

Table 1.1: $u_1 = sin(x)$ calculations

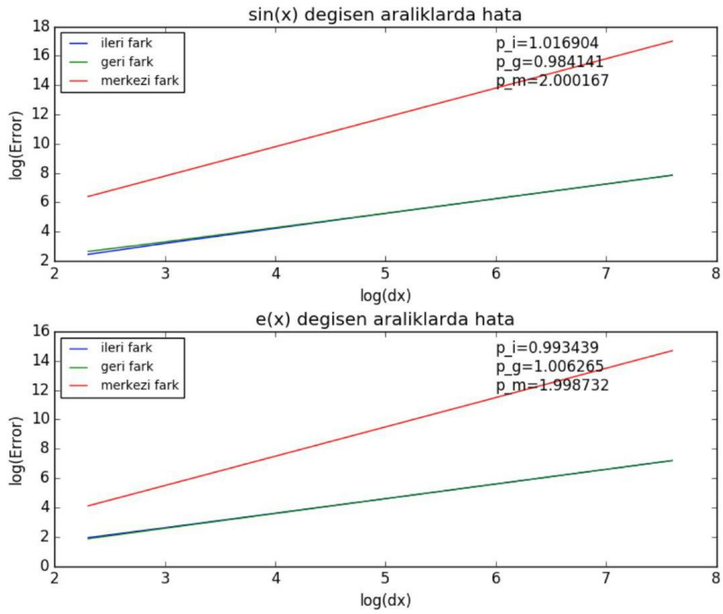
u_2	forward differences				backward differences				central differences			
Δ x	analytic	numeric	error(%)	p(ε)	analytic	numeric	error(%)	p(ε)	analytic	numeric	error(%)	p(ε)
0.1	5.4364	6.352	16.8398	-1.2263	5.4364	4.7037	13.4792	-1.1296	5.4364	5.5278	1.6793	-0.2251
0.05	5.4364	5.868	7.937	-0.6914	5.4364	5.0503	7.1019	-0.6543	5.4364	5.4592	0.4175	0.2915
0.01	5.4364	5.5191	1.5175	-9.0567E-2	5.4364	5.3559	1.4825	-8.5507E-2	5.4364	5.4374	1.6554E-2	0.8905
5E-3	5.4364	5.4775	0.7541	5.3252E-2	5.4364	5.395	0.7449	5.5568E-2	5.4364	5.4367	3.6788E-3	1.0579
1E-3	5.4364	5.4447	0.1508	0.2738	5.4364	5.4283	0.1489	0.2756	5.4364	5.4364	0	

Table 1.2: $u_2 = e^{x^2}$ calculations

p(ε) calculated from the following arguments;

$$Error \approx \Delta x \tag{1.1}$$
$$E = C(\Delta x)^p \tag{1.2}$$
$$log(E) \approx log(C(\Delta x)^p) \tag{1.3}$$
$$\approx log(C) + log((\Delta x)^p) \tag{1.4}$$
$$log(E) \approx log(C) + p * log(\Delta x) \tag{1.5}$$
$$p = log(E)/log(\Delta x) \tag{1.6}$$

If we plot the $log(E)$ and $log(\Delta x)$ and calculate the slopes ;



Code that written for the values in the table and to plot the graph;

```
#!/usr/bin/python

from math import sin, exp
from numpy import *
import matplotlib.pyplot as plt

#Fonsksiyonlar

def fsin(x):
 return sin(x)

def fexp(x):
 return exp(x**2)
```

```
#Initialization

exact_sin = cos(1)
exact_e = 2*1*exp(1**2)

d = [1e-1,5e-2,1e-2,5e-3,1e-3,5e-4]

r_sin=ones((3,size(d)))
r_e=ones((3,size(d)))

error_sin=ones((3,size(d)))
error_e=ones((3,size(d)))

perror_sin=ones((3,size(d)))
perror_e=ones((3,size(d)))


##Sayisal turevler

def ileri (f, x, dx):
 return (f(x+dx)-f(x))/dx

def geri (f,x,dx):
 return (f(x) - f(x-dx))/dx

def central (f,x,dx):
 return (f(x+dx)-f(x-dx))/(2*dx)

#####
def op(f, d, r, error,perc_error, exact, x):

 for i in range(size(d)):
  r[0][i]=ileri(f,x,d[i])
  r[1][i]=geri(f,x,d[i])
  r[2][i]=central(f,x,d[i])

  error = abs((exact-r)/r)
  perc_error = error*100
 return r , error, perc_error

###Uygulama

r_sin, error_sin, perror_sin = op(fsin, d, r_sin, error_sin, perror_sin,
    exact_sin, 1)
r_e, error_e, perror_e = op(fexp, d, r_e, error_e, perror_e, exact_e, 1)

#p ler
```

```
##Plot
x = abs(log(d))
#Sin turevi icin eksenler
y_s = abs(log(error_sin[0]))
y1_s = abs(log(error_sin[1]))
y2_s = abs(log(error_sin[2]))

p_s, z = polyfit(x, y_s, 1)
p1_s, z = polyfit(x, y1_s, 1)
p2_s, z = polyfit(x, y2_s, 1)

plt.figure(1)
plt.subplot(211)
plt.xlabel('log(dx)')
plt.ylabel('log(Error)')
plt.title('sin(x) degisen araliklarda hata')

plt.plot(x,y_s,x,y1_s,x,y2_s)

plt.text(6, 14, 'p_i=%f\np_g=%f\np_m=%f'%(p_s,p1_s,p2_s))

plt.legend(['ileri fark','geri fark','merkezi fark'],loc='best',fontsize=10)

#E fonk

y_e = abs(log(error_e[0]))
y1_e = abs(log(error_e[1]))
y2_e = abs(log(error_e[2]))

p_e, z = polyfit(x, y_e, 1)
p1_e, z = polyfit(x, y1_e, 1)
p2_e, z = polyfit(x, y2_e, 1)

plt.figure(1)
plt.subplot(212)
plt.xlabel('log(dx)')
plt.ylabel('log(Error)')
plt.title('e(x) degisen araliklarda hata')

plt.plot(x,y_e,x,y1_e,x,y2_e)

plt.text(6, 12, 'p_i=%f\np_g=%f\np_m=%f'%(p_e,p1_e,p2_e))

plt.legend(['ileri fark','geri fark','merkezi fark'],loc='best',fontsize=10)
plt.show()
```