

LAPORAN

PRAKTIKUM KOMPUTASI BIOMEDIS

Chapter 8 : Numerical Derivate: Finite Difference Approximations

Pelaksanaan Praktikum:

Hari: Selasa

Tanggal: 29 Oktober 2019

Jam ke: 9-10



Oleh:

Nama : M. Thoriqul Aziz E

NIM : 081711733002

Dosen Pembimbing : Osmalina Nur Rahma S.T., M.Si

LABORATORIUM KOMPUTER
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS AIRLANGGA
SURABAYA

2019

A. TUJUAN

Mahasiswa dapat menentukan nilai turunan numerik menggunakan metode *forward*, *backward* dan *central* pendekatan turunan.

B. DASAR TEORI

Dalam keperluan geometri, perubahan nilai per-satuan waktu atau jarak dalam kalkulus yang digunakan adalah differensial(Munir,2010). Dalam kemajuan teknologi saat ini, kebutuhan penghitungan differensial menjadi lebih rumit jika harus dihitung secara manual. Dari kondisi tersebut, maka digagas sebuah metode turunan numerik dengan pendekatan kontinu yang diterapkan pada computer yang kemudian dapat membantu perhitungan turunan secara lebih cepat dan efisien meski menghasilkan nilai galat/ eror hasil perhitungan.

Terdapat tiga metode yang umum diterapkan dalam menghitung turunan numerik yaitu

1. Forward Approximation

$$f'(x_0) = \frac{f(x_0 + h) - f(x_0)}{h} = \frac{f_1 - f_0}{h}$$

2. Backward Approximation

$$f'(x_0) = \frac{f(x_0) - f(x_0 - h)}{h} = \frac{f_0 - f_1}{h}$$

3. Center Approximation

$$f'(x_0) = \frac{f(x_0 + h) - f(x_0 - h)}{2h} = \frac{f_1 - f_{-1}}{2h}$$

C. TUGAS

1. Explain the effect of the change of h towards the error of numerical derivative calculation and give the reason to that!
2. Modify the program that you make to solve the following problem.
Calculate $f'(1.5)$ if the available points are (1.2, 0.8333), (1.4, 0.7143), (1.6, 0.6250), and (1.8, 0.5556).
3. Buat program sedemikian sehingga dapat mengklasifikasikan metode yang digunakan dari nilai x yang dimasukkan!

D. PEMBAHASAN

1. Dalam penentuan nilai turunan numerik dapat dilakukan dengan pendekatan kedepan(forward), belakang(backward), dan metode tengah(central). Berikut adalah contoh kode program dari ketiga metode tersebut dalam Python 3.7:

```
1 import math as mt
2 import matplotlib.pyplot as plt
3 from sympy import *
4 def f(x):
5     return x*mt.sin(x)
6 x0=1
7 h=0.0001
8 fow=(f(x0+h)-f(x0))/h
9 back=(f(x0)-f(x0-h))/h
10 cent=(f(x0+h)-f(x0-h))/2*h
11 print(fow)
12 plt.plot(x0,f(x0))
13 ##### Untuk Mengetahui Persamaan Hasil Turunan #####
14 xsymbol("x")
15 fdx=sin(x)
16 dydx=fd.diff(x)
17 print("turunan pertama =",dydx)
18 ##### Mengetahui Hasil Sebenarnya dari Perhitungan Turunan Kontinu #####
19 fx=lamdbify(x,dydx)
20 print("f'(1) = ",fx(1))
21 erabs=(fx(1)-fow)
22 print("Nilai Error dari hasil perhitungan =", er)
```

Hasil dari perhitungan turunan numerik dari kode pemrograman tersebut adalah sebagai berikut:

```
Run: program derivatif
"C:\Users\Thoriqui Azi\AppData\Local\Programs\Python\Python37\python.exe" "D:/Tugas/Coding/Komputasi Biomedis/Tugas/ch8/program derivatif.py"
1.382917825029833
turunan pertama = x*cos(x) + sin(x)
f'(1) = 1.3817732986768363
Nilai Error dari hasil perhitungan = 0.0011445349538458333
Process finished with exit code 0
```

Dari hasil perhitungan tersebut dapat diamati bahwa nilai eror perhitungan turunan numerik dan turunan kontinu menggunakan python dengan nilai h(lebar bias) adalah 0.01 adalah 0.0011. Jika nilai lebar bias h diperkecil menjadi 0.0001 maka hasil *command window* pada program Python 3.7 :

```
Run: program derivatif
"C:\Users\Thoriqui Azi\AppData\Local\Programs\Python\Python37\python.exe" "D:/Tugas/Coding/Komputasi Biomedis/Tugas/ch8/program derivatif.py"
1.3817852422493804
turunan pertama = x*cos(x) + sin(x)
f'(1) = 1.3817732986768363
Nilai Error dari hasil perhitungan = 1.1951573264168402e-05
Process finished with exit code 0
```

Dapat diamati bahwa ketika bias diperkecil menjadi 0.0001 maka nilai selisih eror antara hasil turunan numerik dan turunan kontinu menjadi

1.1×10^{-5} atau sepuluh kali lebih kecil dari eror sebelumnya. Hal ini disebabkan karena semakin kecil lebar pias, maka luasan fungsi hasil turunan menjadi lebih sempit sehingga hasil nilai turunan menjadi lebih detail dan memiliki eror lebih kecil. Akan tetapi, dengan semakin kecil pias, artinya jumlah perhitungan akan semakin detail sehingga dapat memberatkan sistem yang digunakan.

2. Dari hasil modifikasi program yang mana diketahui nilai hasil fungsi turunan terhadap beberapa nilai masukan x nya. Sedemikian sehingga dapat dibentuk kode program dalam Python 3.7 sebagai berikut:

```

1 from sympy import *
2 A=[1.2,1.4,1.6,1.8]
3 B=[0.8333,0.7143,0.6250,0.5556]
4 h=A[1]-A[0]
5 number=eval(input("masukan nilai x = "))
6 for i in range(0,len(A)):
7     if A[i] >= number:
8         urut=i
9         break
10 print(urut)
11 fow=(B[urut+1]-B[urut])/h
12 print("hasil fow =", fow)

```

Berikut adalah hasil keluaran *command window* jika nilai masukan yang diberikan adalah pada nilai $f'(1.5)$:

```

Run: task 2
"C:\Users\Thoriquil Aziz\AppData\Local\Programs\Python\Python37\python.exe" "D:/Tugas/Coding/Komputasi Biomedis/Tugas/ch8/task 2.py"
masukan nilai x = 1.5
2
hasil fow = -0.34700000000000014
Process finished with exit code 0

```

Dari hasil modifikasi tersebut, diperoleh hasil bawasanya nilai $f'(1.5) = -0.347$. Modifikasi dilakukan dengan konsep menggeser nilai anggota bilangan x sehingga dapat diketahui hasil turunan yang mungkin dari titik titik yang diberikan. Titik nilai f pada beberapa titik tersebut menjadi nilai penentu dalam rumusan *forward derivate approximation* yaitu $(f_1-f_0)/h$. Semula, nilai masukan x harus didefinisikan urutan dalam matriks. Dari urutan keanggotaan matriks x tersebut kemudian dilakukan perhitungan turunan dengan pendekatan maju dengan memasukan nilai y yang bersesuaian dengan urutan pada matriks x . Sehingga, dapat diketahui hasil dari turunan.

3. Dari persoalan tersebut, sehingga dapat dibuat sebuah kode pemrograman pada Python 3.7 sebagai berikut:

```
task2.py | Ch8 Klasifikasi Turunan.py | Coba integrasi numerik.py
1 import math as mt
2 def fungsi(x):
3     return x*mt.sin(x)
4 A=[1.2,1.4,1.6,1.8,2.0]
5 number=eval(input("masukan nilai x = "))
6 for i in range(0,len(A)):
7     if A[i] >= number:
8         urut=i
9         break
10 if number==1.2:
11     h=A[urut]-A[urut-1]
12     fow=(fungsi(number+h)-fungsi(number))/h
13     print('Hasil dengan metode forward= ',fow)
14 elif number>1.2 and number<2.0:
15     h = A[urut] - A[urut - 1]
16     fow = (fungsi(number + h) - fungsi(number)) / h
17     print('Hasil dengan metode forward= ',fow)
18     back=(fungsi(number)-fungsi(number-h))/h
19     print('Hasil dengan metode backward = ',back)
20     center=(fungsi(number+h)-fungsi(number-h))/2*h
21     print('Hasil dengan metode center= ',center)
22 elif number==2.0:
23     h = A[urut] - A[urut - 1]
24     back = (fungsi(number) - fungsi(number - h)) / h
25     print('Hasil dengan metode backward= ',back)
```

Berikut adalah hasil pada *command window* :

```
Run: Ch8 Klasifikasi Turunan
"C:\Users\Thoriqui Azi\AppData\Local\Programs\Python\Python37\python.exe" "D:/Tugas/Coding/Komputasi Biomedis/Tugas/ch8/Ch8 Klasifikasi Turunan.py"
masukan nilai x = 1.2
Hasil dengan metode forward= 1.0984407144128208
Hasil dengan metode backward = 1.3859135941158642
Hasil dengan metode center= 0.04808788617857368
Process finished with exit code 0
```

Dari hasil tersebut, dapat diketahui bahwasanya untuk mengklasifikasikan metode turunan numerik yaitu menggunakan seleksi *if else* pada masukan nilai x nya. Apabila nilai x yang dimasukkan adalah batas atas x atau batas bawah, maka metode yang bisa digunakan adalah metode forward atau backward saja. Jika nilai x berada diantaranya, maka dapat menggunakan ketiga metode secara bersamaan.

E. KESIMPULAN

Dari persoalan diatas dapat diketahui bahwa terdapat 3 metode dalam menentukan turunan numerik yaitu menggunakan metode *forward*, *backward*, dan *center derivate approximation*. Nilai eror dari perhitungan semakin kecil jika nilai h semakin kecil, sehingga perhitungan nilai turunan menjadi semakin sempit dan detail.

F. DAFTAR PUSTAKA

Capra, Steven C and Canale.1991. “**Numerical Methods for Engineers with Personal Computers Applications**”. MacGraw-Hill Book Company.

King M.R and Mody N.A .2010. “**Numerical and Statical Methods for Bioengineering**”.Cambridge University Press. New York.

Munir, Rinaldi.2003.”**Metode Numerik**”. Didownload dari https://kupdf.net/download/metode-numerik-rinaldi-munir-pdf_58eca95edc0d60f81ada9811_pdf