M O D U L

PRAKTIKUM
METODE NUMERIK

Penyusun :
Priyo Sidik Sasongko,S.Si,M.Kom

JURUSAN ILMU KOMPUTER /I NFORMATIKA
FAKULTAS SAINS DAN MATEMATIKA
UNIVERSITAS DIPONEGORO
2013

MODUL I
SOLUSI PERSAMAAN NONLINEAR

Tujuan :

1. Dapat menghitung akar persamaan nonlinear dengan metode Biseksi, metode Newton Raphson dan metode Secant
2. Mencari besarnya kesalahan dari suatu perhitungan akar persamaan nonlinear dengan metode Biseksi, metode Newton Raphson, dan metode Secant

Petunjuk Praktikum :

1. Lengkapi penggal program di bawah ini serta cetak keluarannya.
2. Buatlah laporan praktikum. Adapun isi laporan meliputi :
   a. Program dan cetak keluarannya
   b. Pembahasan hasil/keluaran

Pendahuluan

Pencarian akar(penyelesaian) suatu persamaan non-linear, $y = f(x)$, adalah mencari suatu harga x^* , yang apabila disubstitusikan ke dalam persamaan itu, akan memberikan harga fungsi nol. Secara matematika

$$f(x^*) = 0$$

Sebagai contoh, akar dari persamaan $f(x) = x - e^{1/x}$ adalah 1.763223 karena, apabila harga $x^* = 1.763223$ disubstitusikan ke dalam persamaan itu, harga fungsi itu menjadi nol.

Metode-metode numeric yang banyak digunakan untuk menyelesaikan persamaan non-linear adalah metode biseksi, metode regula falsi, secant, newton –Raphson dan titik tetap. Tiga metode akan dibicarakan di sini adalah metode biseksi , metode Newton-Raphson dan metode Secant.

A. Metode Biseksi

Dalam metode Biseksi, interval yang mengandung akar dibagi menjadi dua secara berurutan hinggga ukuran interval mengecil dan akhirnya mencapai harga toleransi kesalahan yang diinginkan. Dalam interval [a,b] terdapat sebuah akar (yang akan dicari), apabila dipenuhi :

$$f(a) * f(b) \leq 0$$

Algoritma :

Masukan :

Batas kiri dan kanan interval, a dan b

Toleransi tol, Maksimum iterasi $maxit$

Fungsi, dinyatakan sebagai $fx(x)$

Keluaran :

Akar pendekatan, x_m

Proses :

1. iter=0
2. $x_l = a, x_r = b$
3. Jika $fx(x_l) * fx(x_m) < 0$, kerjakan langkah 4-langkah 7
4. $x_m = 0.5 * (x_l + x_r)$
5. $iter = iter + 1$
6. $delta = |x_m - x_l|$
7. Selagi $delta > tol \& iter < maxit$ kerjakan :

   7.1. jika $fx(x_l) * fx(x_m) < 0$, maka

   $$x_r = x_m$$

   Jika tidak, $x_l = x_m$

   7.2. baharui harga x_m: $x_m = 0.5 * (x_l + x_r)$
8. Akar pendekatan = x_m
9. Selesai

Tugas01 :

Diberikan fungsi $f(x) = e^x + x^2 - 3x - 2 = 0$ terdapat sebuah akar riil dalam selang [-1.0, 1.0]. Carilah akar tersebut dengan metode Biseksi dengan toleransi kesalahan 1e-5.

Format Luarannya:

Pencarian Akar dari $f(x) = e^x + x^2 - 3x - 2 = 0$

Dengan Metode Biseksi

iter	x_l	x_r	x_m	$delta$	$f(x_l)$	$f(x_m)$	$f(x_l) * f(x_m)$

Akar pendekatannya : ……………..

Dengan Toleransi : …………….

B. Metode NEWTON-RAPHSON

Metode Newton Raphson, dalam mencari akar suatu fungsi nonlinear $y = f(x)$memerlukan evaluasi harga fungsi dan turunannya pada sembarang titik x yang merupakan harga awal tebakan akar fungsi tersebut. Metode ini didasarkan atas perluasan deret Taylor di sekitar suatu titik. Karenanya, apabila harga awal tebakan jauh dari akar sebenarnya, konvergensi akan lambat atau mungkin tidak dicapai sama sekali.

Algoritma :

Masukan : Fungsi, dinyatakan sebagai $fx(x)$

Turunan fungsi, dinyatakan sebagai $dfx(x)$

Harga Tebakan awal x_c

Toleransi eps, maksimum iterasi $maxit$

Keluaran : Akar pendekatan, x_{c+1}

Proses :

1. iter=0
2. Hitung $fx(x_c)$ dan $dfx(x_c)$
3. iter=iter+1
4. $x_{c+1} = x_c - \frac{fx(x_c)}{dfx(x_c)}$
5. $delta = |x_{c+1} - x_c|$
6. Jika$delta \leq eps$& iter>maxit maka Akar pendekatan = x_{c+1}, Selesai
7. $x_c = x_{c+1}$, kembali ke langkah 2

Tugas02:

Diberikan fungsi $f(x) = e^x + x^2 - 3x - 2 = 0$mempunyai akar riildalam selang [-1.0, 1.0]. Carilah akar tersebut dengan metode Newton Raphson dengan toleransi kesalahan 1e-5.

Format Luarannya:

Pencarian Akar Fungsi $f(x) = e^x + x^2 - 3x - 2 = 0$

Dengan Metode Newton Raphson

iter	x_c	$f(x_c)$	$delta$

Akar pendekatannya : ……………..

Dengan Toleransi : …………….

A. Penggal Program Biseksi:

```
{*****************************************************}
{       Program untuk Menghitung Akar Persamaan Nonlinear        }
{                   dari fungsi : f(x) = x^2 -5                  }
{              dengan Metode  Biseksi                     }
{              Dibuat oleh :                                     }
{                      Nama         :                            }
{                      NIM          :                            }
{                      Prog.Studi   :                            }
{*****************************************************}
program biseksi ;
uses crt;
var   ……

function f(x) : real) : real;
begin
        f := x^2-5;
end

begin
clrscr;
data := sqrt(5);

writeln( ‘Mencari Akar Persamaan Nonlinear         ‘)
writeln(‘ f(x) = x^2-5                                ‘);
writeln( ‘Metode Biseksi                             ‘);
writeln(‘--------------------------------------------------‘);
writeln;

write(‘Masukkan Batas bawah                =’); read(a);
write(‘masukkan Batas atas          =’);read(b);
write(‘Toleransi                           =’);read(tol);
write(‘Jumlah maksimum iterasi             =’);read(maxit);

iter :=0;
F_a :=  f(a);
F_b:= f(b);
If F_a*F_b > 0 then writeln(‘ Nilai F(a) * F(b) > 0 ‘)
else
begin
write(‘ iter    a    m   b        f(a)     f(b)       abs[f(b)-f(a)]/2           galat ’)
epsilon := tol+1;
while ((iter<=maxit) and (epsilon >tol))
do
```

```
    begin
        iter := iter +1;
        m    := (a+b)/2;
        F_m := f(m);
    galat := data – abs(m);
        write ( ‘Tuliskan cetak hasilnya………..’);
        epsilon := abs(m-a);
        If( F_a * F_m <= 0 ) Then
        begin
          b     := m;
          F_b := F_m;
        end
        else
        begin
          a      :=m
          F_a  := F_m;
        end
    end
    if(it<=maxit) then
    begin
          writeln(‘toleransi terpenuhi’);
          writeln(‘hasil akhir     = ‘, m:9:7);
end
else writeln (toleransi tidak terpenuhi ‘);
end;
end.
```

B. Penggal Program Newton Raphson:

```
{*****************************************************}
{        Program untuk Menghitung Akar Persamaan Nonlinear            }
{                        dari fungsi : f(x) = x^2 -5                   }
{                  dengan Metode  Newton Rapshon                    }
{                  Dibuat oleh :                                       }
{                            Nama            :                         }
{                            NIM             :                         }
{                            Prog.Studi      :                         }
{*****************************************************}
program Newton_Rapshon;
uses crt;
var   ……

function f(x) : real) : real;
begin
          f := x^2-5;
end
```

```
function f1(x) : real) : real;
begin
          f 1:= 2*x;
end

begin
clrscr;
data := sqrt(5);

writeln( ‘Mencari Akar Persamaan Nonlinear          ‘)
writeln(‘ f(x) = x^2-5                                   ‘);
writeln( ‘Metode Newton Rapshon                         ‘);
writeln(‘-----------------------------------------------------‘);
writeln;

write(‘Masukkan Nilai awal              =’); read(x0);
write(‘Toleransi                                  =’);read(tol);
write(‘Jumlah maksimum iterasi                    =’);read(maxit);

iter :=0;
epsilon := tol+1;
write(‘ iter     x        f(x)       f’(x)       epsilon        galat ’)
while ((iter<=maxit) and (epsilon >tol))
do
begin
        iter := iter +1;
        xb    := x0 – f(x0)/f1(x0);
    galat := data – abs(m);
        write ( ‘Tuliskan cetak hasilnya………..’);
        x0 := xb;
end;
if(it<=maxit) then
    begin
          writeln(‘toleransi terpenuhi’);
          writeln(‘hasil akhir     = ‘, xb:9:7);
end
else writeln (toleransi tidak terpenuhi ‘);
end.
```

Eksekusi program Newton Raphson dilakukan dalam nilai awal x0 = 1.0, toleransi 10^{-7} dan jumlah iterasi maksimum sebanyak 50.

A. Metode Secant

Metode Newton memerlukan perhitungan turunan pertama fungsi. Karena itu, tidak semua fungsi mudah dicari turunan pertamanya, terutama fungsi yang rumit. Turunan fungsi tersebut digantikan dengan bentuk yang ekuivalen.

$$f'(x_c) = \frac{f(x_c) - f(x_{c-1})}{x_c - s_{c-1}}$$

Algoritma :

Masukan : Fungsi, dinyatakan sebagai $fx(x)$

Harga Tebakan awal x_0, x_c

Toleransi eps, maksimum iterasi $maxit$

Keluaran : Akar pendekatan, x_{c+1}

Proses :

1. iter=0
2. Hitung $fx(x_0)$,$fx(x_c)$
3. iter=iter+1
4. $x_{c+1} = x_c - \frac{fx(x_c)*(x_c - x_0)}{fx(x_c) - fx(x_0)}$
5. $delta = |x_{c+1} - x_c|$
6. Jika $delta \leq eps$& iter>maxit maka Akar pendekatan = x_{c+1}, Selesai
7. $x_c = x_{c+1}, x_0 = x_c$ kembali ke langkah 2

Tugas03:

Diberikan fungsi $f(x) = e^x + x^2 - 3x - 2 = 0$n mempunyai akar riil dalam selang [-1.0, 1.0]. Carilah akar tersebut dengan metode Secant dengan toleransi kesalahan 1e-5.

Format Luarannya:

Pencarian Akar Fungsi $f(x) = e^x + x^2 - 3x - 2 = 0$

Dengan Metode Secant

iter	x_c	$f(x_c)$	$delta$

Akar pendekatannya : ……………..

Dengan Toleransi : …………….

MODUL II
INTERPOLASI

TUJUAN: Mahasiswa dapat membuat program interpolasi dengan metode polinom Newton dan metode polinom Lagrange

Petunjuk Praktikum :.
Buatlah laporan praktikum. Adapun isi laporan meliputi :

a. Dasar teori untuk menentukan penyelesaian interpolasi Newton dan interpolasi Lagrange tersebut di atas
b. Program dan Pembahasan Program
c. Pembahasan hasil/keluaran

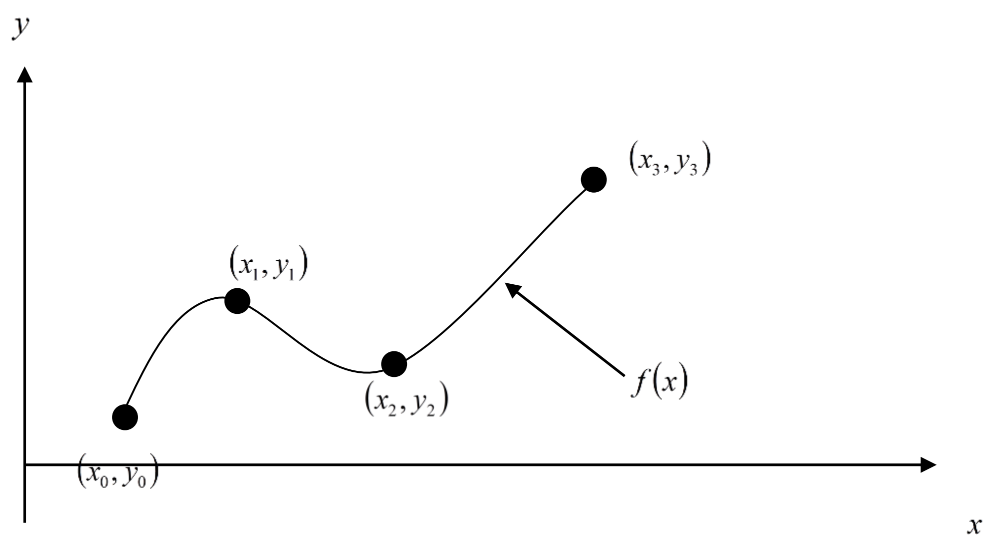
Pendahuluan

Diketahui pasangan data (x_0, y_0), (x_1, y_1), ..., (x_{n-1}, y_{n-1}), (x_n, y_n). Bagaimana mencari y untuk nilai x lain yang dikehendaki? Fungsi kontinu $f(x)$ direpresentasikan $n+1$ data (Gambar 1).

Interpolasi Polynomial meliputi pencarian polynomial derajad n yang melalui $n+1$ titik. Metode yang sering dipakai adalah metode interpolasi Newton dann metode interpolasi Lagrange.

Metode Interpolasi Newton

Ilustrasi gambar 1.



Gambar 1 Interpolasi dari data diskrit.

Bentuk Umum metode polinom Newton

Diberikan $n+1$ titik data, $(x_0, y_0),(x_1, y_1),......,(x_{n-1}, y_{n-1}),(x_n, y_n)$, jarak titik x dengan selang yang sama, sebagai

$$P_n(x) = b_0 + b_1(x - x_0) + ... + b_n(x - x_0)(x - x_1)...(x - x_{n-1})$$

Atau dapat ditulis sebagai

$$P_n(x) = P(x_0 + hu) = y_0 + u\Delta^1 y_0 + \frac{u(u-1)}{2!}\Delta^2 y_0 + ... + \frac{u(u-1)...(u-n+1)}{n!}\Delta^n y_0$$

Dengan

$$\frac{x - x_0}{h} = u$$

Algortima Interpolasi Newton ke depan :

1. Baca data masukan pasangan x dan y
2. Berikan nilai x yang dicari y nya.
3. Hitung h
4. Lakukan inisialisasi :
   Sum = y0
   R = (x-x0)/h
5. Lakukan iterasi proses berikut untuk i=1 sampai n-1
   a. Product =1
   b. Untuk j=0 sampai i-1 lakukan perhitungan
   Product = product*(u-j)/(j+1)
   c. Lakukan penjumlahan
   Sum = sum +product * $\Delta^i y_0$
6. Kembalikan nilai sum sebagai hassil perhitungan

Bentuk Umum metode polinom Lagrange

Diberikan $n+1$ titik data, $(x_0, y_0),(x_1, y_1),......,(x_{n-1}, y_{n-1}),(x_n, y_n)$, jarak titik x dengan selang yang tidak sama. Dengan $f(x_i) = y_i$.
Polinom Lagrange diberikan oleh

$$P_n(x) = \sum_{i=0}^{n} L_i(x) f(x_i)$$

dengan n dalam $f_n(x)$ merupakan derajad order n yang mengaproksimasi fungsi $y = f(x)$ diberikan $n+1$ titik data sebagai. $(x_0, y_0),(x_1, y_1),......,(x_{n-1}, y_{n-1}),(x_n, y_n)$, dan

$$L_i(x) = \prod_{\substack{j=0 \\ j \neq i}}^{n} \frac{x - x_j}{x_i - x_j}$$

Algortima Interpolasi Lagrange :

1. Baca data masukan pasangan x dan y
2. Lakukan inisialisasi :
   Sum = y_0
   R = (x-x_0)/h
3. Lakukan iterasi proses berikut untuk i=1 sampai n-1
   a. Product =y_i
   b. Untuk j=0 sampai i-1 lakukan perhitungan
      Product = Product*(x-x_j)/($x_i - x_j$)
   c. Lakukan penjumlahan
   Sum = sum +product * $\Delta^i y_0$
4. Kembalikan nilai sum sebagai hasil perhitungan

Tugas 04a. Gunakan Program Interpolasi Newton untuk menghitung x = 2.5

x	**0.0**	**1.0**	**2.0**	**3.0**	**4.0**
f(x)	**1.0000**	**0.5403**	**-0.4161**	**-0.9900**	**-0.6536**

Tugas 04b. Gunakan Program Interpolasi Lagrange untuk menghitung x = 323.5

x	**321.0**	**322.8**	**324.2**	**325.0**
f(x)	**2.50651**	**2.50893**	**2.51081**	**2.51188**

MODUL III
Menghitung Penyelesaian Sistem Persamaan Linear
Dengan Metode Gauss

Tujuan :
1. Dapat menentukan penyelesaian sistem persamaan linear dengan metode Gauss
2. Mencari besarnya kesalahan dari suatu perhitungan solusi system persamaan linear dengan metode Gauss

Pendahuluan
Banyak system riil dalam berbagai bidang yang dapat dimodelkan dalam bentuk system persamaan linear, dengan melibatkan banyak variabel yang tidak mungkin diselesaikan secara manual. Untuk menangani system seperti itu, terdapat beberapa metode penyelesaian system persamaan linear tersebut secara efisien dan akurat. Dalam modul ini disajikan metode yang pupuleryaitu : metode eliminasi Gauss. System n persamaan dan n variabel dituliskan sebagai berikut :

$$
\begin{aligned}
&a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \ldots + a_{1n}x_n = b_1 \\
&a_{21}x_1 + a_{22}x_2 + a_{23}x_3 + \ldots + a_{2n}x_n = b_2 \\
&. \qquad\qquad . \\
&. \qquad\qquad . \\
&. \qquad\qquad . \\
&a_{n1}x_1 + a_{n2}x_2 + a_{n3}x_3 + \ldots + a_{nn}x_n = b_n
\end{aligned}
$$

Atau $Ax = b$

A. Metode eliminasi Gauss

Dalam eliminasi Gauss, system persamaan ditransformasikan ke dalam system ekivalen dalam bentuk segitiga. Jadi matriks koefisien dikonversikan ke matriks segitiga, bentuk baru ini mudah disubstitusikan balik.

Algoritma :
MASUKAN
Dimensi matriks n
Matriks koefisien a
Larik kanan b
KELUARAN : larik penyelesaian x

1. Untuk i=1 sampai n
   1.1 jika $a_{i,j} = 0$
      1.1.1 cari nilai $a_{i+1,j} <> 0$ untuk $i + 1 \leq n$
      1.1.2 jika tak ditemukan $a_{i+1,j} <> 0$, keluar, tidak ada solusi
      1.1.3 untuk j=i sampai n, tukas $a_{i,j}$ dengan $a_{i+1,j}$
   1.2 untuk j=i+1 sampai n
      1.2.1 $m_j = a_{j,i}/a_{i,i}$
      1.2.2 Untuk $k = i$ sampai n
      $$a_{ij,k} = a_{j,k} - m_j * a_{i,k}$$
      1.2.3 $y_j = y_j - m_j * y_i$
2. $x_n = b_n/a_{n,n}$
3. $untuk\ i = n - 1$ sampai i=1

3.1 $ax = 0.0$
3.2 Untuk $j = i + 1$ sampai n
$ax = ax + a_{i,j} * x_j$
3.3 $x_i = (y_i - ax)/a_{i,i}$

4. selesai

Petunjuk Praktikum :

1. Buatlah program serta cetak keluarannya.
2. Buatlah laporan praktikum. Adapun isi laporan meliputi :
   a. Dasar teori untuk menentukan solusi system persamaan linear metode tersebut di atas
   b. Pembahasan Program
   c. Pembahasan hasil/keluaran

Tugas 05:

Diberikan system persamaan linear berikut :

$$3x_1 - 0.1x_2 - 0.2x_3 = 7.85$$

$$0.1x_1 + 7x_2 - 0.3x_3 = -19.3$$

$$0.3x_1 - 0.2x_2 + 10x_3 = 71.4$$

Tentukan solusi dengan metode eliminai Gauss tersebut.

B. Penggal Program Metode Eliminasi Gauss:

```
{*********************************************************}
{      Program untuk Menghitung Solusi Sistem persamaan Linear       }
{                          Ax = B                                    }
{               dengan Metode  Eliminasi Gauss                       }
{          Dibuat oleh :                                             }
{               Nama        :                                        }
{               NIM         :                                        }
{               Prog.Studi  :                                        }
{*********************************************************}
Program eliminasi_Gauss;
uses crt;
type
      Mat55 = Array[0..5,0..5] of real;
      Mat56 = Array[0..5,0..6] of real;
Var
      A       : Mat55;
      G       : Mat56;
      B,x     : Array [0..5] of real;
      Factor, dummy         : real;
      I,j,k,n               : integer;
```

```
        Prosedure Inisialiasi;
        Begin
        { buatlah Matrik A };
        { buatlah Matrik B };
        n := 3;

begin
clrscr;
Inisialisasi;

{ bentuk matriks gabungan }
for i :=0 to n-1 do for j:= 0 to n-1do G[i][j] := A[i][j];
for j :=0 to n-1 do G[j][n] := b[j];
{Proses Eliminasi }
for i:= 0 to n-2 do
begin
  For j:= i+1 to n-1 do
        begin
           factor := G[j][i]/G[i][i];
           For k :=i to n do G[j][k] := G[j][k] –factor*G[i][k];
        end;
end;
{Substitusi Mundur }
x[n-1] := G[n-1][n]/G[n-1][n-1];
for k:= 0 to n-2 do
  begin
   I := n-2-k;
   dummy := G[i][n];

   j := i+1 to n-1 do
        dummy := dummy –G[i][j]*x[j];
        x[i] := dummy /G[i][i];
   end;
writlen( ‘ Hasil akhir :’)
for i := 0 to n-1 do writeln( ‘x[‘,i:2,’[ = ‘, x[i]:10:7);
end.
```

MODUL IV

INTEGRASI NUMERIK

Tujuan :

1. Dapat menentukan penyelesaian Integrasi Numerik dengan metode Trapesium, dan Metode Simpson 1/3
2. Mencari besarnya kesalahan dari suatu perhitungan solusi Integrasi Numerikdengan Dengan metode Trapesium, dan Metode Simpson 1/3.

Petunjuk Praktikum :

1. Lengkapi penggal program di bawah ini serta cetak keluarannya.
2. Buatlah laporan praktikum. Adapun isi laporan meliputi :
   a. Dasar teori untuk menentukan penyelesaian Integrasi Numerik tersebut di atas
   b. Pembahasan Program
   c. Pembahasan hasil/keluaran

Dalam kalkulus dasar kita belajar cara mengevaluasi integral bermacam-macam fungsi dan kita mengenal teknik-teknik integral. Sayangnya tidak semua fungsi dapat dengan mudah diintegrasikan secara analitik. Dengan bantuan computer, kita dapat mengatasi kesulitan itu dengan memanfaatkan metode-metode numeric yang berkaitan dengan integrasi.

Integrasi numeric dikenal juga sebagai kuadratur; persoalan integrasi numeric ialah menghitung secara numeric integral tertentu

$$I = \int_a^b f(x)dx$$

Yang dalm hal ini a dan b adalah batas-batas integral, f adalah fungsi yang dapat diberikan secara eksplisit dalam bentuk persamaan ataupun secara empiric dalam bentuk tabel nilai.Dalam praktikum modul ini membahas teknik integrasi numeric menurut Kaidah Trapesium dan Kaidah Simpson.

a. Kaidah Trapesium

Pandang sebuah pias bernbentuk trapesium dari $x = x_0$sampai $x = x_l$.

Luas satu trapezium adalah

$$\int_{x_0}^{x_l} f(x)dx = \frac{h}{2}[f(x_0) + f(x_l)]$$

Bila selang [a,b] dibagi atas n buah pias trapezium, kaidah integrasi yang diperoleh adalah

$$\int_a^b f(x)dx = \frac{h}{2}[f(x_0) + 2f(x_1) + 2f(x_2) + \cdots + \ 2f(x_{n-1}) + f(x_n)]$$

Algoritma:

Masukan :

fungsi yang diintegrasikan , $y = f(x)$

Batas bawah dan batas atas integral, a, b

Jumlah panel, n

Keluaran :

I = hasil integrasi fungsi.

Proses:

1. Tetapkan lebar panel

   $h = (b - a)/n$

2. Nilai awal total

   $Sum \ = \ f(a)$

3. Untuk i=1 sampai n-1 kerjakan :

   Sum = sum + 2f(a+i*h)

4. Hitung hasil integral

   I = h/2*(sum +f(b))

5. Selesai

b. Kaidah Simpson 1/3

   Menurut kaidah Simpson, luas bidang di bawah kurva $f(x)$ dalam selang [a,b], dapat didekati dengan

$$\int_{x_0}^{x_l} f(x)dx = \ \frac{h}{3}\left[f(a) + 4f\left(\frac{a+b}{2}\right) + f(b)\right]$$

   Bila selang [a,b] dibagi atas n buah pias, kaidah integrasi yang diperoleh adalah

$$\int_a^b f(x)dx = \frac{h}{3}[f(x_0) + 4f(x_1) + 2f(x_2) + \cdots + 2f(x_{n-2}) + \ 4f(x_{n-1}) + f(x_n)]$$

   Algoritma:

   Masukan :

   fungsi yang diintegrasikan , $y = f(x)$

Batas bawah dan batas atas integral, a, b

Jumlah panel, n

Keluaran :

I = hasil integrasi fungsi.

Proses :

1. Tetapkan lebar panel

   $h = (b - a)/n$

   $x = a$

2. Nilai awal total

   $I = f(a) + f(b)$

   $sigma = 0$

3. Untuk i=1 sampai n-1 kerjakan :

   $x = x + h$

   If $i \, mod \, 2 = 1$ then

   $sigma = 4 * f(x)$

   else

   $sigma = 2 * f(x)$

   end

$$I = I + sigma$$

4. Hitung hasil integral

   I = h/3(I)*

5. Selesai

Tugas 06: Dengan kedua metode tersebut,

a. Tentukan integral dari fungsi $y = \mathrm{x} * \sin(x)$ dengan interval [0.0, pi] dengan n= 128.
b. Tentukan galatnya!

Format Luaran Program adalah :

```
*****************************************************************
          Hasil Program untuk Menyelesaian Integrasi Numerik
                Int x * sin(x)dx  ; syarat x(0) =0.0 , x(1) = pi
                    dengan Metode  Trapesium dan Simpson 1/3
                    Dibuat oleh :

                              Nama          :  ......................................
                              NIM           :  .....................................
                              Prog.Studi    :  ......................................
--------------------------------------------------------------------------------------------------------------
  N |    H    |   Int  Trapesium     |    Err IT  %      | Int Simpson 1/3      |    Err IS 1/3  %  |
--------------------------------------------------------------------------------------------------------------
### |  #.### |   ########.####    |     #####.##      | #########.####    | ######.##        |
--------------------------------------------------------------------------------------------------------------
```

MODUL V

DIFFERENSI NUMERIK

Tujuan :

1. Dapat menentukan penyelesaian turunan suatu fungsi dari data-data yang diketahui, namun fungsinya tidak diketahui.
2. Mencari besarnya kesalahan dari suatu perhitungan turunan suatu fungsi secara Numerik dengan dengan rumusan interpolasi.

Petunjuk Praktikum :.
Buatlah laporan praktikum. Adapun isi laporan meliputi :

a. Dasar teori untuk menentukan penyelesaian turunan suatu fungsi dari data yang diketahui
b. Program dan Pembahasan Program
c. Pembahasan hasil/keluaran

Differensi numeric adalah proses perhitungan turunan fungsi dari data suatu fungsi. Dengan menggunakan rumusan interpolasi dan turunannya. Turunan suatu fungsi dapat ditentukan secara numeric. Adapun pendekatan perhitungan turunan numeric diperoleh dengan Deret Taylor:

1. Aproksimasi Selisih Maju

$$f_i' = \frac{f_{i+1} - f_i}{h} + O(h)$$

$$f_i'' = \frac{f_{i+2} - 2f_{i+1} + f_i}{h^2} + O(h\quad)$$

2. Aproksimasi Selisih Mundur

$$f_i' = \frac{f_i - f_{i-1}}{h} + O(h)$$

$$f_i'' = \frac{f_{i-2} - 2f_i + f_i}{h^2} + O(h\quad)$$

3. Aproksimasi Selisih Pusat

$$f_i' = \frac{f_{i+1} - f_{i-1}}{2h} + O(h^2)$$

$$f_i'' = \frac{f_{i+1} - 2f_i + f_{i-1}}{h^2} + O(h^2)$$

Adapun pendekatan perhitungan turunan numeric diperoleh dengan Interpolasi Polinomial :

1. Aproksimasi Selisih Maju

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

   $f'(x_0) = \frac{-3f_0 + 4f_1 + f_0}{2h}$, tiga titik

   $f''(x_0) = \dfrac{-f_3 + 4f_2 - 5f_1 + 2f_0}{12h}$

2. Aproksimasi Selisih Mundur

   $f'(x_0) = \frac{f_0 - f_{-1}}{h}$, dua titik

3. Aproksimasi Selisih Pusat

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

Tugas 07.

Diberikan tabel berikut

x	$f(x)$
1.3	3.669
1.5	4.482
1.7	5.474
1.9	6.686
2.1	8.166
2.3	9.974
2.5	12.182

c. Buatlah program untuk menentukan $f'(1.7)$ dan $f'(2.1)$

d. Buatlah program untuk menentukan $f''(1.7)$ dan $f''(2.1)$

Catatan Formula yang digunakan terserah Anda

Format Luaran :

Format Luaran Program adalah :

```
*****************************************************************
      Hasil Program untuk Menyelesaian Differensi Numerik
            denganFormula ################
            Dibuat oleh :
                  Nama          :
                  NIM           :
                  Prog.Studi    :
            ----------------------------------------
            No|  X   |   F(x)|
            -----------------------------------------
            ### |  #.### |    ########.####    |
            -----------------------------------------
Turunan Pertama pada  x = ###.### adalah ########.####
Turunan Kedua pada     x = ###.### adalah ########.####
```

(Catatan : Ketiga Formula, Formula Selisih Maju, Selisih Mundur dan Selisih Sentral dalam satu program)

MODUL VI

Menyelesaikan Persamaan Differensial Biasa
Dengan Metode Euler, dan Metode Runge Kutta

Tujuan :

1. Dapat menentukan penyelesaian Persamaan differensial Biasa secara Numerik dengan metode Euler, metode Heun, dan Metode Runge Kutta
2. Mencari besarnya kesalahan dari suatu perhitungan solusi Persamaan Differensial Biasa secara Numerik dengan Dengan Metode Euler, Metode Heun, dan Metode Runge Kutta

Petunjuk Praktikum :.
Buatlah laporan praktikum. Adapun isi laporan meliputi :

d. Dasar teori untuk menentukan penyelesaian Persamaan Differensial Biasa tersebut di atas
e. Program dan Pembahasan Program
f. Pembahasan hasil/keluaran

Pendahuluan

Bentuk umum suatu persamaan diferensial biasa orde n adalah :

$$\frac{d^n y}{dt^n} = f\left(t, y, \frac{dy}{dt}, \frac{d^2 y}{dt^2}, \dots, \frac{d^{n-1} y}{dt^{n-1}}\right), a \leq t \leq b$$

Jika harga y dan turunan pertamanya diberikan pada t = a, maka persamaan diferensial di atas disebut problem harga awal (Initial Value Problem). Jika beberapa suku

$$\left(y, \frac{dy}{dt}, \frac{d^2 y}{dt^2}, \dots, \quad \frac{d^{n-1} y}{dt^{n-1}}\right)$$

Ditentukan pada t = a dan suku-suku lainnya pada t=b, persamaan diferensial itu disebut problem harga batas dua titik (Two-point boundary-value problem).

Dalam modul ini diperkenalkan 3 metode popular untuk menyelesaikan persamaan diferensial biasa: metode Euler, metode Heun, dan metode Runge-Kutta orde empat dengan algoritma masing-masing dan implementasinya.

a. Algoritma Euler

   Masukan

   Waktu awal t_a

   Waktu akhir t_b

Interval dt

Harga fungsi pada $t_a, y(t_a) = y_0$

Keluaran

t, y

Fungsi dy/dt dinyatakan sebagai $dy(t, y)$

2. $n = (t_a - t_b)/dt$
3. $y(0) = y_0, t(0) = t_a$
4. Untuk $i = 0$ sampai $n - 1$

   3.a. $y(i + 1) = y(i) + dt * dy(t(i), y(i))$

   3.b. $t(i + 1) = t(i) + dt$
5. $y\_euler = y(n)$ $\{t_b = t(n), y(t_b)\}$
6. Selesai

b. Algoritma Heun

Masukan

Waktu awal t_a

Waktu akhir t_b

Interval dt

Harga fungsi pada $t_a, y(t_a) = y_0$

Keluaran

t, y

Fungsi dy/dt dinyatakan sebagai $dy(t, y)$

1. $n = (t_a - t_b)/dt$
2. $y(0) = y_0, t(0) = t_a$
3. Untuk $i = 0$ sampai $n - 1$

   3.a. $k1 = dt * dy(t(i), y(i))$

   3.b. $k2 = dt * dy(t(i), y(i) + k1)$

   3.c. $y(i + 1) = y(i) + 0.5 * (k1 + k2)$

   3.d. $t(i + 1) = t(i) + dt$
4. $y_{euler-perbaiki} = y(n)$ $\{t_b = t(n), y(t_b)\}$
5. Selesai

c. Algoritma Runge-Kutta Orde Empat

Masukan

Waktu awal t_a

Waktu akhir t_b

Interval dt

Harga fungsi pada t_a, $y(t_a) = y_0$

Keluaran

t, y

Fungsi dy/dt dinyatakan sebagai $dy(t, y)$

1. n = $(t_a - t_b)/dt$
2. $y(0) = y_0$, $t(0) = t_a$
3. Untuk $i = 0$ sampai $n - 1$
   - 3.a. $k1 = dt * dy(t(i), y(i))$
   - 3.b. $k2 = dt * dy(t(i) + 0.5 * dt, y(i) + 0.5 * k1)$
   - 3.c. $k3 = dt * dy(t(i) + 0.5 * dt, y(i) + 0.5 * k2)$
   - 3.d. $k4 = dt * dy(t(i) + dt, y(i) + k3)$
   - 3.e. $y(i + 1) = y(i) + 1/6 * (k1 + 2 * k2 + 2 * k3 + k4)$
   - 3.f. $t(i + 1) = t(i) + dt$
4. $y\_rungekutta\_orde4 = y(n)$ $\{t_b = t(n)$, y$(t_b)\}$
5. Selesai

Tugas 08 :

Diberikan persamaan differensial biasa derajad satu $\frac{dy}{dx} = x + y,\ y(0) = 1.$ Hitung $y(0.10)$ dengan ukuran langkah h = 0.01. jumlah angka bena =5. Jika diketahui solusi sejati PDB tersebut adalah $y(x) = e^x - x - 1$.

Format Luarannya:

****

Program untuk Menyelesaian Persamaan Differensial

dy/dx = x+y ; syarat y(0) =1

dengan Metode Euler, Heun, dan Runge Kutta4

Dibuat oleh :

NIM :

Prog.Studi :

****

Solusi PDB dy/dx$= x + y, y(0) = 1.$

x	h	$y\_Analitik$	y_Euler	Error %	y_Heun	Error %	$y\_RungeKuta4$	Error %

000   SELAMAT BEKERJA 000