Interpolasi

Risanuri Hidayat,
Jurusan Teknik Elektro dan Teknologi Informasi
FT UGM,

1.1 INTERPOLASI NEWTON

Interpolasi polinomial Newton mempunyai persamaan sebagai berikut,

$$P(x) = f[x_1] + f[x_1,x_2](x - x_1) + f[x_1,x_2,x_3](x - x_1)(x - x_2) + \cdots \quad (1)$$

$$+f[x_1, \dots, x_3](x - x_1) \dots (x - x_n)$$

Atau dapat dinyatakan dengan persamaan,

$$P(x) = \sum_{k=1}^{n} f[x_1,x_2, \dots, x_k] N_k(x) \quad (2)$$

Dengan

$$N_k(x) = \prod_{i=1}^{k-1} (x - x_i) \quad (3)$$

$$N_1(x) = 1 \quad (4)$$

Jika dijabarkan lebih lanjut $\mathrm{N_k(x)}$ adalah persamaan sebagai berikut,

$$N_2(x) = (x - x_1)$$

$$N_3(x) = (x - x_1)(x - x_2)$$

$$N_k(x) = (x - x_1)(x - x_2) \dots (x - x_{k-1})$$

Sedangkan $f[x_k]$ adalah fungsi y_k sebagai berikut,

$$f[x_1] = y_1$$

$$f[x_2] = y_2$$

$$\vdots$$

$$f[x_k] = y_k \tag{5}$$

Persamaan $f[x_1, x_2, \ldots, x_k]$ disebut dengan *divided difference* (selisih terbagi). Secara rinci *divided difference* dapat dijabarkan seperti berikut ini,

$$f[x_1, x_2] = \frac{f[x_2] - f[x_1]}{x_2 - x_1}$$

$$f[x_2, x_3] = \frac{f[x_3] - f[x_2]}{x_3 - x_2}$$

Atau secara umum dapat dinyatakan dengan,

$$f[x_{k-1}, x_k] = \frac{f[x_k] - f[x_{k-1}]}{x_k - x_{k-1}} \tag{6}$$

Untuk *divided difference* derajat lebih tinggi dapat dinyatakan sebagai berikut,

$$f[x_1, x_2, x_3] = \frac{f[x_3, x_2] - f[x_2, x_1]}{x_3 - x_1}$$

$$f[x_1, x_2, \ldots, x_k] = \frac{f[x_2, \ldots x_k] - f[x_1, x_{k-1}]}{x_k - x_1} \tag{7}$$

Untuk mencari divided difference digunakan matriks segitiga sebagai berikut,

$$\begin{matrix} f[x_1] & 0 & 0 & \ldots & 0 \\ f[x_2] & f[x_1, x_2] & 0 & \ldots & 0 \\ f[x_3] & f[x_2, x_3] & f[x_1, x_2, x_3] & \ldots & 0 \\ \vdots & \vdots & \vdots & \ddots & \\ f[x_k] & f[x_{k-1}, x_k] & f[x_{k-2}, x_{k-1}, x_k] & \cdots & f[x_1, x_2, \ldots, x_k] \end{matrix}$$

Contoh:

Diberikan runtun sebagai berikut:

x	2.00	4.00	5.00	8.00
y=f(x)	0.50	0.25	0.2	0.125

Dari contoh di atas, divided difference dapat dihitung sebagai berikut,

x_k	$f[x_k]$	$f[x_{k-1}, x_k]$	$f[x_{k-2}, x_{k-1}, x_k]$	$f[x_1, x_2, \ldots, x_k]$
2, 000	0,500			
4, 000	0,250	-0,125		
5, 000	0,200	-0,050	0,025	
8, 000	0,125	-0,025	0,00625	-0,003125

Dari persamaan (1) polinomial Newton menjadi,

$$P(x) = 0.5 - 0.125(x-2) + 0.025(x-2)(x-4) - 0.003125(x-2)(x-4)(x-5)$$

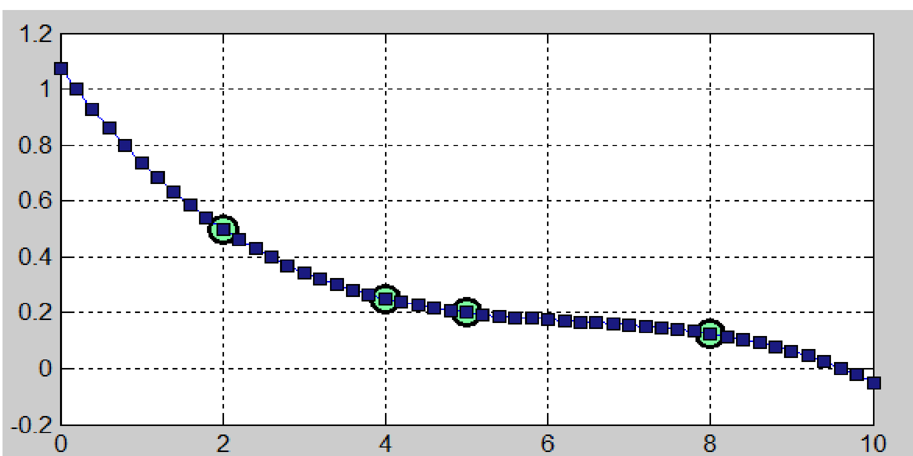
Program MATLAB,

```
function newtonpoly01;

close all;
clear all;
clc;

fy = [ ...
    2.0000    0.50;...
    4.0000    0.25;...
    5.0000    0.20;...
     8.0000    0.125...
    ];

x = fy(:,1); y = fy(:,2);
n = length(x);
D = zeros(n);
D(:,1) = y(1:n);

for (j=2:n)
    for (k=j:n)
        D(k,j) = (D(k,j-1) - D(k-1,j-1))/(x(k) - x(k-j+1));
```

```
    end;
end;

D

% Sekarang akan dihitung sebuah yy=f(xx) dengan rumus polinomial Newton tersebut.
xx = 0:0.2:10;
yy = D(1,1);

for (k=2:n)
    yy = yy+D(k,k).*plinom(xx,x,k);
end;

plot(x,y,'-wo', 'LineWidth',2, 'MarkerEdgeColor','k', 'MarkerFaceColor',[.49 1 .63], 'MarkerSize',12); hold on;
plot(xx,yy,'-bs','LineWidth',1,'MarkerEdgeColor','k','MarkerFaceColor',[.1 .1 .5],'MarkerSize',6); grid on;

 function ypol = plinom(xx,x,k)
        has = 1;
        for (i=2:k)
            has = has.*(xx - x(i-1));
        end;
        ypol = has;
 end;

end
```

Hasil,

```
D =

     0.5000         0          0          0
     0.2500   -0.1250          0          0
     0.2000   -0.0500     0.0250          0
     0.1250   -0.0250     0.00625    -0.003125

>>
```



1.2 INTERPOLASI LAGRANGE

Interpolasi polinomial Lagrange mempunyai persamaan sebagai berikut,

$$P(x) = \sum_{k=1}^{n} f_k L_k(x) \tag{8}$$

Dengan persamaan Lagrange sebagai berikut,

$$L_1(x) = \frac{(x - x_2)(x - x_3) \dots (x - x_k)}{(x_1 - x_2)(x_1 - x_3) \dots (x_1 - x_k)}$$

$$L_2(x) = \frac{(x - x_1)(x - x_3) \dots (x - x_k)}{(x_2 - x_1)(x_2 - x_3) \dots (x_2 - x_k)}$$

$$\vdots$$

$$L_k(x) = \frac{(x - x_1)(x - x_2) \dots (x - x_{i \neq k})}{(x_k - x_1)(x_k - x_2) \dots (x_k - x_{i \neq k})}$$

Atau dapat dinyatakan dengan,

$$L_k(x) = \prod_{\substack{i=1 \\ i \neq k}}^{n} \frac{(x - x_i)}{(x_k - x_i)} \tag{9}$$

Sedangkan f_k adalah nilai fungsi itu sendiri, yaitu

$$f_1 = y_1$$

$$f_2 = y_2$$

$$f_k = y_k \tag{10}$$

Contoh:

Sebagaimana contoh di atas, diberikan runtun sebagai berikut:

x	2.00	4.00	5.00	8.00
y=f(x)	0.50	0.25	0.2	0.125

Dari contoh di atas, nilai Lagrange dapat dihitung sebagai berikut,

$$L_1(x) = \frac{(x-4.0)(x-5.0)(x-8.0)}{(2-4.0)(2-5.0)(2-8.0)} = \frac{(x-4.0)(x-5.0)(x-8.0)}{-36.00}$$

$$L_2(x) = \frac{(x-2.0)(x-5.0)(x-8.0)}{(4.0-2.0)(4.0-5.0)(4.0-8.0)} = \frac{(x-2.0)(x-5.0)(x-8.0)}{8.00}$$

$$L_3(x) = \frac{(x-2.0)(x-4.0)(x-8.0)}{(5.0-2.0)(5.0-4.0)(5.0-8.0)} = \frac{(x-2.0)(x-4.0)(x-8.0)}{-9}$$

$$L_4(x) = \frac{(x-2.0)(x-4.0)(x-5.0)}{(8.0-2.0)(8.0-4.0)(8.0-5.0)} = \frac{(x-2.0)(x-4.0)(x-5.0)}{72}$$

Dari persamaan (8) (9) dan (10), polinomial Langrange menjadi,

$$P(x) = \frac{0.5(x-4.0)(x-5.0)(x-8.0)}{-36.00} + \frac{0.25(x-2.0)(x-5.0)(x-8.0)}{8.00} + \frac{0.2(x-2.0)(x-4.0)(x-8.0)}{-9} + \frac{0.125(x-2.0)(x-4.0)(x-5.0)}{72}$$

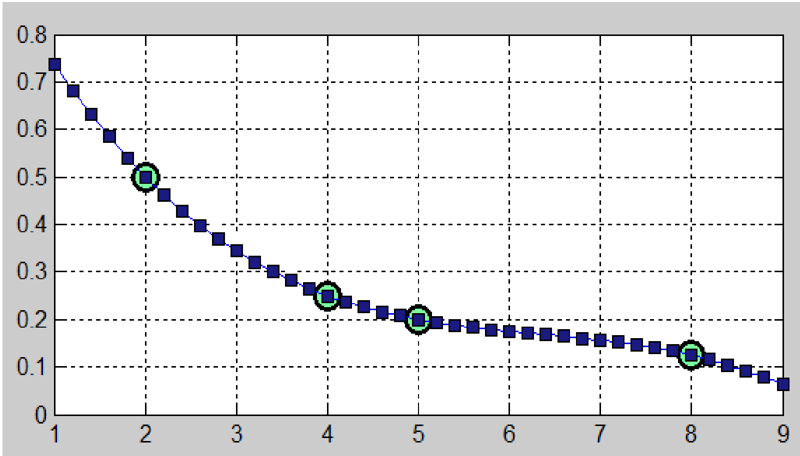
Program MATLAB,

```
function lagrange01;

close all;
clear all;
clc;

fy = [ ...
   2.00 0.500; ...
   4.00 0.250; ...
   5.00 0.200; ...
   8.00 0.125 ...
    ];

x = fy(:,1); y = fy(:,2);


% Menentukan koefisien Lagrange
```

```
L = koef_lagrange(x,y);

% Test fungsi dengan polinomial Lagrange

xx = 1:0.2:9;
yy = plinom_lagrange(L,xx);

function L = koef_lagrange(x,y)
n=length(x);
LL=zeros(1,n);
for k=1:n
   V=1;
   for j=1:n
      if k~=j
         V = V.*(x(k)-x(j));
      end;
   end;
   LL(k)=1/V;
end;

L = LL.*y';

end;

function yy = plinom_lagrange(L,xx);
n = length(L);
VV = 0;
for k=1:n
    VVV(k,:) = ones(1,length(xx));
   for j=1:n
      if k~=j
          VVV(k,:) = VVV(k,:).*(xx-x(j));
      end
   end
   VV = VV + L(k).*VVV(k,:);
end
yy = VV;
end;

plot(x,y,'-wo', 'LineWidth',2, 'MarkerEdgeColor','k', 'MarkerFaceColor',[.49 1 .63], 'MarkerSize',12); hold on;
plot(xx,yy,'-bs','LineWidth',1,'MarkerEdgeColor','k','MarkerFaceColor',[.1 .1 .5],'MarkerSize',6); grid on;

end
```

Hasil,

```
L =

   -0.0139    0.0313   -0.0222    0.0017

>>
```



DAFTAR PUSTAKA

[1] JH. Mathews, KK. Fink, *Numerical Methods Using Matlab*, Prentice Hall, 2004