IMPLEMENTASI INTERPOLASI LAGRANGE UNTUK PREDIKSI NILAI DATA BERPASANGAN DENGAN MENGGUNAKAN MATLAB

Krisnawati
STMIK AMIKOM Yogyakarta
e-mail : krisna@amikom.ac.id

ABSTRACT

Interpolasi Lagrange sangat dikenal dalam metode numerik, karena menggunakan fungsi dalam bentuk polinomial. Dari sekumpulan data berpasangan yang ada, dapat diketahui dengan pasti fungsi yang melalui titik-titik tersebut. Dengan menggunakan Matlab dibuat aplikasi untuk membantu proses pencarian fungsi yang dimaksud. Titik-titik yang diketahui haruslah merupakan bilangan real, dalam bidang datar. Berdasarkan fungsi ini dapat diprediksikan nilai selanjutnya untuk kasus yang ada. Interpolasi banyak digunakan untuk memprediksi nilai data berpasangan.

Keywords *: interpolasi, prediksi.*

1. PENDAHULUAN

Aproksimasi merupakan salah satu usaha untuk menyajikan data berbentuk grafis menjadi kalimat matematis.

Secara umum aproksimasi harus mendapatkan suatu fungsi yang melewati semua titik yang diketahui. Aproksimasi ini dikenal sebagai interpolasi. Karena harus melewati semua titik yang ada, maka ada banyak fungsi yang memenuhi, kecuali jika fungsi tersebut mempunyai syarat tertentu.

$$x = xi \rightarrow f(xi) = yi$$

Sedangkan secara khusus aproksimasi tidak mensyaratkan melewati semua titik. Walaupun demikian solusi yang didapat haruslah merupakan hasil terbaik yang mendekati semua titik yang diketahui. Aproksimasi secara khusus lebih dikenal dengan istillah regresi.

$$x = xi \rightarrow f(xi) \approx yi$$

Ada banyak metode interpolasi yang dapat diterapkan, diantaranya adalah:

1. Interpolasi Newton
2. Interpolasi Lagrange
3. Interpolasi Hermite
4. Interpolasi Invers

2. INTERPOLASI LAGRANGE

Merupakan teknik yang popular, karena menggunakan fungsi dalam bentuk polinom.

Jika fungsi yang dicari adalah f(x) dan cacah data n maka :

$$f(x) = \sum_{i=1}^{n} y_i L_i(x)$$

$$Li(x) = \begin{cases} 0, x \neq x_i \\ 1, x = x_i \end{cases}$$

$$= \frac{Q_i(x)}{Q_i(x_i)}$$

dengan :

Qi(x) = (x – x0) (x – x1) (x – x2) ….. (x – xi - 1) (x – xi + 1) …. (x – xn)

Contoh penyelesaian untuk tiga titik diketahui.

3 titik tersebut adalah (1,-1), (3,1/2), (4,0)

$$L_1(x) = \frac{(x-3)(x-4)}{(1-3)(1-4)} = \frac{1}{6}(x^2 - 7x + 12)$$

$$L_2(x) = \frac{(x-1)(x-4)}{(3-1)(3-4)} = -\frac{1}{2}(x^2 - 5x + 4)$$

$$L_3(x) = \frac{(x-1)(x-3)}{(4-1)(4-3)} = \frac{1}{3}(x^2 - 4x + 3)$$

Penyelesaian akhir didapat sebagai berikut:

P(x) = y1L1(x) + y2L2(x) + y3L3(x)

$$= -L_1(x) + \frac{1}{2}L_2(x) + 0$$

$$= \left\{ -\frac{1}{6}(x^2 - 7x + 12) \right\} + \left\{ \frac{1}{2} * (-\frac{1}{2})(x^2 - 5x + 4) \right\}$$

$$= -\frac{1}{6}x^2 + \frac{7}{6}x - 2 - \frac{1}{4}x^2 + \frac{5}{4}x - 3$$

$$= -\frac{5}{12}x^2 + \frac{29}{12}x - 3$$

Contoh penyelesaian untuk empat titik diketahui.

Empat titik tersebut adalah (0,1), (1,2), (3,4), (6,-1)

$$L_1(x) = \frac{(x-1)(x-3)(x-6)}{(0-1)(0-3)(0-6)} = -\frac{1}{18}x^3 + \frac{10}{18}x^2 - \frac{27}{18}x + 1$$

$$L_2(x) = \frac{(x-0)(x-3)(x-6)}{(1-0)(1-3)(1-6)} = \frac{1}{10}x^3 - \frac{9}{10}x^2 + \frac{10}{18}x$$

$$L_3(x) = \frac{(x-0)((x-1)(x-6)}{(3-0)(3-1)(3-6)} = -\frac{1}{18}x^3 + \frac{7}{18}x^2 - \frac{6}{18}x$$

$$L_4(x) = \frac{(x-0)(x-1)(x-3)}{(6-0)(6-1)(6-3)} = \frac{1}{90}x^3 - \frac{4}{90}x^2 + \frac{3}{90}x$$

Penyelesaian akhir didapat sebagai berikut:

P(x)

$$= y_1 L_{1(x)} + y_2 L_{2(x)} + y_3 L_{3(x)} + y_4 L_{4(x)}$$
$$= L_{1(x)} + 2L_{2(x)} + 4L_{3(x)} - L_{4(x)}$$
$$= -\frac{1}{18}x^3 + \frac{10}{18}x^2 - \frac{27}{18}x + 1 + \frac{2}{10}x^3 - \frac{18}{10}x^2 + \frac{20}{18}x - \frac{4}{18}x^3 + \frac{28}{18}x^2 - \frac{24}{18}x$$
$$-\frac{1}{90}x^3 + \frac{4}{90}x^2 - \frac{3}{90}x$$
$$= -\frac{4}{45}x^3 + \frac{16}{45}x^2 + \frac{11}{15}x + 1$$

3. ALGORITMA

Dari manual diatas dapat dituliskan algoritma kasarnya sebagai berikut :

A. Tetapkan jumlah titik yang diketahui.

Untuk menginputkan titik yang diketahui dapat meenggunakan dua array x dan y dengan jumlah data = jumlah titiknya.

Dengan dua array akan lebih mudah mengatur perilaku data didalam program. Bisa juga menggunakan banyak array sejumlah titik yang diketahui, sehingga masing-masing pasang data disimpan dalam satu array. Cara ini terlihat lebih sederhana, tetapi lebih sulit dalam mengatur perilaku data.

Dalam implementasi ini nantinya akan dipilih cara yang pertama, yakni menggunakan dua array x dan y.

B. Mencari Li(x) dan P(x)

Li(x) didapat sejumlah titik yang diketahui, sehingga diperlukan perulangan sebanyak titik yang diketahui. Demikian pula P(x) merupakan jumlahan dari perkalian yi dan Li(x), sehingga memerlukan perulangan yang jumlahnya sana dengan proses pencarian Li(x). Untuk mencari Li(x) diperlukan Qi(x) dan Qi(xi). Karena Qi(x) merupakan hasil perkalian (x-xi) sejumlah titik yang diketahui, maka diperlukan perulangan lagi untuk mencarinya. Tetapi yang harus diingat disini adalah bahwa, untuk (x-xi) tersebut tidak ikut dalam hasil perkalian. Sehingga proses hanya akan dilakukan untuk nilai selain (x-xi). Untuk Qi(xi) dapat dicari setelah Qi(x) diketahui dengan cara mensubstitusi nilai xi ke dalam Qi(x). Setelah Qi(x) dan Qi(xi) diketahui dapat dicari Li(x). Dan untuk selanjutnya mencari P(x).

Misalnya banyaknya titik yang diketahui adalah b, maka algoritma diatas dapat diperhalus menjadi sebagai berikut:

1. Inputkan b.
2. Dari i = 1 s.d b

   Inputkan titik ke i
3. Dari i = 1 s.d b

   Cari Qi(x)

   Cari Qi(xi)

   Cari Li(x)

   Cari P(x)

4. PEMROGRAMAN DAN PENGETESAN

Algoritma diatas diimplementasikan menjadi sebuah program.

Listing programnya sebagai berikut:

```
clc;clear;
%membangun objek simbolik x
syms x;

%menginputkan banyaknya titik
b=input('Banyak titik = ');

%menginputkan masing-masing titik
for i=1:b
   fprintf('x%d',i)
   bx(i)=input(' = ');
   fprintf('y%d',i)
   by(i)=input(' = ');
end
clc;

%menampilkan titik-titik yang sudah diinputkan ke layar
clc;
disp('Titik-titik yang diketahui adalah sebagai berikut:');
for i=1:b
   fprintf('(%d,%1.1f)',bx(i),by(i));
end

%inisialisasi fx
fx=0;
fprintf('\n\n');
disp('Nilai masing-masing L(x)');

% mulai proses pencarian q(x), qx1, lx, dan px
for i=1:b
   %inisialisasi qx
   qx=1;
```

```
    %perulangan untuk mencari qx
    for j=1:b
        if (i~=j)
            qx=qx*(x-bx(j));
        end
    end

    %mencari qx1 dengan substitusi x ke gx
    qx1=subs(qx,x,bx(i));
    %mencari lx
    lx=qx/qx1;
    lx1=collect(lx);
    %menampilkan lx
    fprintf('L%d(x) = ',i);
    disp(lx1);
    %mencari fx
    fx=fx+by(i)*lx;
end

%menyederhanakan fx menjadi px dan menampilkan ke layar
px=collect(fx);
fprintf('Hasilnya = ');
disp(px);
```

Program diatas digunakan untuk menyelesaikan dua permasalahan yang sudah dibahas diatas.

Permasalahan pertama untuk 3 titik diketahui.

Inputnya sebagai berikut :

Banyak titik = 3

x1 = 1

y1 = -1

x2 = 3

y2 = 0.5

x3 = 4

y3 = 0

Input diatas memberikan output sebagai berikut:
Titik-titik yang diketahui adalah sebagai berikut:
(1,-1.0)(3,0.5)(4,0.0)

Nilai masing-masing L(x)
L1(x) = 1/6*x^2-7/6*x+2
L2(x) = -1/2*x^2+5/2*x-2
L3(x) = 1/3*x^2-4/3*x+1

Hasilnya = -5/12*x^2+29/12*x-3
Permasalahan kedua, untuk empat titik diketahui
Inputnya sebagai berikut:
Banyak titik = 4
x1 = 0
y1 = 1
x2 = 1
y2 = 2
x3 = 3
y3 = 4
x4 = 6
y4 = -1

Input diatas memberikan output sebagai berikut:
Titik-titik yang diketahui adalah sebagai berikut:
(0,1.0)(1,2.0)(3,4.0)(6,-1.0)

Nilai masing-masing L(x)
L1(x) = -1/18*x^3+5/9*x^2-3/2*x+1
L2(x) = 1/10*x^3-9/10*x^2+9/5*x
L3(x) = -1/18*x^3+7/18*x^2-1/3*x
L4(x) = 1/90*x^3-2/45*x^2+1/30*x

Hasilnya = -4/45*x^3+16/45*x^2+11/15*x+1

Output program dibandingkan dengan manual yang ada sebelumnya memberikan hasil yang sama. Selain dengan dua contoh diatas program melalui serangkaian tes dengan menggunakan berbagai macam jenis data.

Tetapi perlu menjadi catatan bahwa program hanya digunakan untuk data real, tidak menangani data kompleks

5. KESIMPULAN

Diperlukan teknik tersendiri dalam mengimplementasikan interpolasi Lagrange ke dalam program. Teknik tersebut sebenarnya tidak jauh berbeda dalam mengimplementasikan algoritma lain pada umumnya yakni : pemilihan tipe data yang tepat, yakni pada saat input data dilakukan. Dengan aplikasi ini akan lebih mudah dalam mencari fungsi dari titik-titik yang diketahui untuk memprediksi nilai lainnya.

6. DAFTAR PUSTAKA

Gary J. Lastman & Naresh K. Sinha, 2000, Microcomputer-Based Numerical Methods for Science and Enginering.

MatLab 6 Help.

William J Palm, 2004, Introduction to MatLab 6 for Engineers, The McGraw-Hill Companies, Inc.

http://ft.uns.ac.id/ts/kul_ol/numerik/numerik03_regresi.htm

http://ft.uns.ac.id/ts/kul_ol/numerik/numerik04_interpolasi.htm

http://www.malang.ac.id/e-learning/FMIPA/

http://library.gunadarma.ac.id/files/disk1/9/jbptgunadarma-gdl-course-2004-jackwidjaj-415-met_num_-p.ppt