4 INTERPOLASI

Interpolasi adalah proses pencarian dan perhitungan nilai suatu fungsi yang grafiknya melewati sekumpulan titik yang diberikan. Titik-titik tersebut mungkin merupakan hasil eksperimen dalam sebuah percobaan, atau diperoleh dari sebuah fungsi yang diketahui. Fungsi interpolasi biasanya dipilih dari sekelompok fungsi tertentu, salah satunya adalah fungsi *polinomial* yang paling banyak dipakai.

Pada bab ini kita membahas beberapa metode numerik untuk mendapatkan fungsi polinomial sebagai hampiran suatu fungsi. Tujuan utama mendapatkan polinomial hampiran ini adalah untuk menggantikan suatu fungsi yang rumit dengan fungsi yang lebih sederhana bentuknya dan mudah dimanipulasi. Di antara fungsi-fungsi yang dapat digunakan sebagai fungsi hampiran adalah fungsi polinomial, fungsi trigonometrik, dan fungsi rasional. Kita hanya akan membahas cara-cara mendapatkan fungsi polinomial hampiran. Fungsi-fungsi polinomial banyak dipakai dalam praktek, karena fungsi-fungsi tersebut mudah dihitung nilainya, diturunkan, diintegralkan, dan perilakunya baik – semua turunannya ada dan kontinyu.

Polinomial banyak dipakai sebagai hampiran fungsi, karena sifatnya yang mudah dihitung nilainya, diturunkan, diintegralkan, dan perilakunya baik – semua turunannya ada dan kontinyu.

Secara umum kita akan membahas masalah penyusunan sebuah polinomial hampiran untuk satu himpunan data titik-titik diskrit. Titik-titik ini, yang biasanya disajikan dalam bentuk tabel, mungkin merupakan hasil eksperimen fisik. Metode tertentu yang dapat digunakan untuk menyusun suatu polinomial hampiran dapat dipilih berdasarkan konteks dari mana data diperoleh.

Interpolasi digunakan untuk menyelesaikan berbagai masalah dalam bidang teori hampiran yang lebih umum. Untuk memberikan beberapa wawasan bagi Anda, berikut disajikan beberapa masalah hampiran (aproksimasi) dan kemungkinan pemakaian interpolasi untuk menyelesaikannya. (1) Diberikan sebuah tabel nilai-nilai suatu fungsi, misalnya $f(x) = \cos(x)$, interpolasi dapat digunakan untuk mencari nilai-nilai $f(x)$ untuk nilai-nilai x yang tidak terdapat di dalam tabel. (2) Diberikan sekumpulan data berupa titik-titik (koordinat), tentukan sebuah fungsi mulus $f(x)$ yang tidak naik-turun (osilatif) dan cocok dengan data tersebut baik secara eksak maupun hampiran. Kasus kecocokan eksak mengarah ke studi fungsi-fungsi interpolasi spline, dan kasus hampiran kecocokan data dikerjakan dengan metode kuadrat terkecil. (3) Diberikan sebuah fungsi $f(x)$, misalkan $f(x) = \log(x)$, dan diperlukan suatu cara untuk menghitung nilai-nilai fungsi tersebut menggunakan komputer. Dalam masalah ini, interpolasi digunakan sebagai alat untuk mendapatkan suatu hampiran yang dapat dihitung. (4) Untuk mengintegralkan atau menurunkan suatu fungsi secara numerik, kita sering mengganti fungsi yang bersangkutan dengan ekspresi hampiran yang lebih sederhana, yang biasanya diperoleh dengan menggunakan interpolasi. Juga, beberapa metode numerik untuk menyelesaikan persamaan diferensial yang dipakai secara luas diperoleh dari hampiran interpolasi.

Beberapa pemakaian interpolasi

4.1 Interpolasi Numerik

Misalkan kita mempunyai data yang disajikan dalam tabel seperti di bawah ini:

x	x_1	x_2	x_3	$\ldots$	x_n
y	y_1	y_2	y_3	$\ldots$	y_n

dengan $x_1 < x_2 < ... < x_n$. Kita ingin mencari sebuah polinomial $P(x)$ sedemikian hingga

$$P(x_i) = y_i \qquad \text{untuk } 1 \leq i \leq n.$$

Polinomial interpolasi

Polinomial ini dikatakan menginterpolasikan nilai-nilai pada tabel. Kita dapat menggunakan polinomial ini untuk menghitung suatu nilai y yang berkaitan dengan suatu x, yang tidak terdapat di dalam tabel tetapi terletak di antara nilai-nilai x pada tabel tersebut. Dengan kata lain, kita dapat menghitung nilai $P(x)$ untuk sebarang nilai x untuk $x_1 \leq x \leq x_n$. Inilah tujuan interpolasi. Titik-titik x_i disebut **simpul**. Polinomial interpolasi tergantung pada nilai-nilai dan banyaknya nilai x dan y yang diberikan.

Jika nilai-nilai y merupakan nilai-nilai fungsi $f(x)$ untuk nilai-nilai x yang bersesuaian, maka polinomial $P(x)$ merupakan hampiran fungsi $f(x)$ pada interval $[x_1, x_n]$. Dalam hal ini fungsi $E(x) = f(x) - P(x)$ disebut **fungsi galat**, yang biasanya melibatkan turunan tingkat tinggi fungsi

Jadi $P_n(x) \equiv Q_n(x)$. ■

4.2 Polinomial-polinomial Interpolasi

Interpolasi biasanya diperkenalkan di dalam aljabar elementer, yakni pada masalah perluasan tabel logaritma atau trigonometri untuk mencari nilai-nilai yang tidak tersedia di dalam tabel. Interpolasi juga diperkenalkan pada saat anda mempelajari metode trapesium untuk menghitung intergral secara numerik. Kedua interpolasi tersebut adalah sama, yakni **interpolasi linier**.

Terdapat dua alasan penting pemakaian polinomial interpolasi. Pertama, polinomial interpolasi digunakan untuk menghitung hampiran nilai suatu fungsi $f(x)$, karena nilai polinomial mudah dihitung, polinomial mudah diturunkan dan diintegralkan. Kedua, polinomial digunakan dalam penentuan **kurva mulus** yang melalui sekumpulan data titik. Dalam hal ini biasanya diinginkan sebuah kurva yang tidak osilatif. Penyelesaiannya mengarah ke **fungsi-fungsi spline**. Polinomial interpolasi digunakan dalam pembahasan interpolasi dengan spline.

Selain cara seperti contoh di atas, untuk mencari polinomial yang menginterpolasikan semua titik yang diberikan kita dapat melakukan perhitungan mulai dari polinomial konstan yang melalui titik pertama, kemudian mencari polinomial linier yang melalui kedua titik pertama, polinomial kuadratik yang melalui ketiga titik pertama, dst. Dalam hal ini, polinomial berderajad yang lebih tinggi ditentukan dengan polinomial berderajad yang lebih rendah dengan menambah satu titik lain yang harus dilalui. Kita mulai pembahasan secara lebih terperinci sebagai berikut.

Kasus 1: Polinomial Konstan
Dalam hal ini kita hanya menggunakan sebuah nilai x yang diberikan pada tabel, misalkan

x	x_1
y	y_1

Misalkan $P_0(x)$ adalah fungsi polinomial interpolasinya. Maka $P_0(x_1) = a_1 = y_1$. Polinomial tersebut melalui sebuah titik (x_1, y_1) yang diberikan pada tabel. Dengan demikian kita dapat memilih sebuah fungsi konstan

$$P_0(x) = a_1 = y_1 \tag{4.1}$$

Dengan demikian $P_k(x)$ menginterpolasikan semua titik yang diinterpolasikan oleh $P_{k-1}(x)$.

Misalkan sekarang diberikan sebuah tabel dengan $k+1$ nilai x:

x	x_1	x_2	x_3	$\ldots$	x_k	x_{k+1}
y	y_1	y_2	y_3	$\ldots$	y_k	y_{k+1}

Syarat agar $P_k(x)$ menginterpolasikan semua titik di dalam tabel tersebut, $P_k(x)$ harus memenuhi titik terakhir $(x_{k+1},\, y_{k+1})$. Jadi,

$$P_k(x_{k+1}) = P_{k-1}(x_{k+1}) + a_{k+1}(x_{k+1} - x_1)(x_{k+1} - x_2)\ldots(x_{k+1} - x_k) \quad (4.5)$$

atau

$$y_{k+1} = P_{k-1}(x_{k+1}) + a_{k+1}(x_{k+1} - x_1)(x_{k+1} - x_2)\ldots(x_{k+1} - x_k) \quad (4.6)$$

atau

$$a_{k+1} = \frac{y_{k+1} - P_{k-1}(x_{k+1})}{(x_{k+1} - x_1)(x_{k+1} - x_2)\ldots(x_{k+1} - x_k)} \quad (4.7)$$

Dengan demikian, polinomial

$$P_k(x) = P_{k-1}(x) + a_{k+1}(x - x_1)(x - x_2)\ldots(x - x_k), \quad (4.8)$$

dengan

Rumus koefisien polinomial Newton

$$a_{k+1} = \frac{y_{k+1} - P_{k-1}(x_{k+1})}{(x_{k+1} - x_1)(x_{k+1} - x_2)\ldots(x_{k+1} - x_k)}, \quad (4.9)$$

menginterpolasikan sebuah tabel dengan $(k+1)$ pasang nilai $(x,\, y)$. $P_k(x)$ dikenal sebagai polinomial interpolasi *Gregory-Newton* dengan k **simpul pusat** $x_1, x_2, \ldots,$ dan x_k.

Secara umum, dari pembahasan di atas kita tahu bahwa polinomial $P_{n-1}(x)$ yang menginterpolasikan n pasang nilai $(x,\, y)$, yakni $(x_i,\, y_i)$ untuk $1 \le i \le n$, dapat diperoleh secara rekursif sbb.:

Polinomial-polinomial Newton dibentuk secara rekursif.

$$\begin{aligned}
P_0(x) &= y_1 = a_1 \\
P_1(x) &= P_0(x) + a_2(x - x_1) \\
P_2(x) &= P_1(x) + a_3(x - x_1)(x - x_2) \\
P_3(x) &= P_2(x) + a_4(x - x_1)(x - x_2)(x - x_3) \\
&\vdots
\end{aligned}$$

Jadi,

$$P_2(x) = -5 + 2x - 4x(x-1)$$

4. *Polinomial kubik yang menginterpolasikan empat titik pertama* $(0,\ -5)$, $(1,\ -3)$, $(-1,\ -15)$, *dan* $(2,\ 39)$ *adalah*

$$\begin{aligned} P_3(x) &= P_2(x) + a_3(x-x_1)(x-x_2)(x-x_3) \\ &= -5 + 2x - 4x(x-1) + a_3(x-0)(x-1)(x+1) \end{aligned}$$

   Agar $P_3(x)$ *menginterpolasikan* $(2,\ 39)$ *harus dipenuhi* $39 = -5 + 2(2) - 4(2)(2-1) + a_3(2)(2-1)(2+1)$ *atau* $6a_3 = 39 + 9$, *sehingga* $a_3 = 48/6 = 8$. *Jadi,*

$$P_3(x) = -5 + 2x - 4x(x-1) + 8x(x-1)(x+1)$$

5. *Akhirnya, polinomial berderajad 4 yang menginterpolasikan kelima titik yang diberikan adalah*

$$\begin{aligned} P_4(x) &= P_3(x) + a_4(x-x_1)(x-x_2)(x-x_3)(x-x_4) \\ &= -5 + 2x - 4x(x-1) + 8x(x-1)(x+1) \\ &\qquad +a_4(x-0)(x-1)(x+1)(x-2). \end{aligned}$$

   Agar $P-4(x)$ *menginterpolasikan* $(-2,\ -9)$ *harus dipenuhi* $-9 = -5 + 2(-2) - 4(-2)(-2-1) + 8(-2)(-2-1)(-2+1) + a_4(-2)(-2-1)(-2+1)(-2-2)$ *atau* $24a_4 = -9 + 9 + 24 + 48$, *sehingga* $a_4 = 72/24 = 3$. *Jadi,*

$$\begin{aligned} P_4(x) = -5 + 2x - 4x(x-1) + 8x(x-1)(x+1) \\ + 3x(x-1)(x+1)(x-2). \quad \square \end{aligned} \tag{4.11}$$

Silakan Anda cek apakah polinomial $P_4(x)$ di atas memenuhi kelima titik yang diberikan! Silakan Anda cek pula bahwa polinomial $P_4(x)$ tersebut persis sama dengan polinomial yang didapat pada Contoh 4.1. Jadi Anda dapat menggunakan metode matriks (SPL) atau metode Newton untuk mencari polinomial yang menginterpolasi sekumpulan titik tertentu.

Perhitungan nilai polinomial Newton dan koefisien-koefisiennya dilakukan secara rekursif. Untuk menghitung koefisien a_k pada suku x^{k-1} diperlukan nilai-nilai $P_1(x)$, ..., $P_{k-1}(x)$. Akan tetapi, nilai-nilai tersebut

```
function a=interpolaN(x,y)
% Untuk menghitung a_1, a_2, ..., a_n pada
% P_{n-1}=a_1+a_2(x-x_1)+...+a_n(x-x_1)...(x_{n-1})
% yang melalui (x_1,y_1), (x_2,y_2), ..., (x_n,y_n)
n=length(x);
a(1)=y(1);
for k=2:n,
    P=nilaiPolaN(a(1:k-1),x(1:k-1),x(k));
    M=prod(x(k)-x(1:k-1));
    a(k)=(y(k)-P)/M;
end
```

Gambar 4.2: Fungsi MATLAB `interpolaN.m` untuk menghitung koefisien-koefisien polinomial Newton $P_{n-1} = a_1 + a_2(x - x_1) + ... + a_n(x - x_1)...(x_{n-1})$

Berikut adalah contoh pemakaian fungsi MATLAB di atas untuk menyelesaikan soal pada Contoh 4.2. Hasil perhitungan dengan MATLAB sama dengan hasil sebelumnya pada Contoh 4.2.

```
>>x=[0 1 -1 2 -2]
 x  =
    0  1  -1  2 -2
>>y=[-5 -3 -15 39 -9]
 y  =
   -5  -3  -15  39 -9
>>a=interpolaN(x,y)'
 a  =
   -5   2  -4   8   3
```

Perhitungan dengan MATLAB di atas menunjukkan bahwa $a_1 = -5$, $a_2 = 2$, $a_3 = -4$, $a_4 = 8$, dan $a_5 = 3$, sehingga

$$P_4(x) = -5 + 2x - 4x(x-1) + 8x(x-1)(x+1) + 3x(x-1)(x+1)(x-2),$$

sama dengan (4.11)

Jadi $P_4(3) = 241$. □

ALGORITMA 4.1 (PERKALIAN TERSARANG NEWTON).
Menghitung nilai polinomial Newton

$$P_n(x) = a_1+(x-x_1)[a_2+(x-x_2)[a_3+(x-x_3)[\ldots[a_n+a_{n+1}(x-x_n)]\ldots]]]$$

jika x, a_k, *dan* x_k, *untuk* $1 \le k \le (n+1)$, *diketahui.*

INPUT: n, x, a_k $(1 \le k \le (n+1))$, *dan* x_k $(1 \le k \le n)$

OUTPUT: $P_n(x)$

LANGKAH-LANGKAH:

1. $S_1 = a_{n+1}$
2. *For* $k = 2,\ 3,\ \ldots,\ n+1$

   $$S_k = a_{n-k+2} + (x - x_{n-k+2})S_{k-1}$$
3. $P_n(x) = S_{n+1}$
4. *STOP.*

Perhitungan tersebut juga dapat dilakukan dengan menggunakan fungsi-fungsi MATLAB `nilaiPolaN` atau `xsarang`[4]:

```
>>nilaiPolaN(a,x,3)
 ans  =
     241
>>xsarang(a,x,3)
 ans  =
     241
```

Hasil perhitungan dengan kedua fungsi tersebut sama dengan hasil perhitungan secara manual. Metode perkalian tersarang lebih efisien, karena memerlukan lebih sedikit operasi hitung.

Perkalian tersarang juga dapat digunakan untuk menghitung nilai polinomial berderajad n dalam bentuk umum, yakni

Perkalian tersarang untuk polinomial umum

[4]Pada contoh sebelumnya telah dihitung nilai a_1, ..., a_5 dari masukan vektor x dan y

$$\begin{aligned} P_n(x) &= a_1 + a_2x + a_3x^2 + ... + a_{n+1}x^n \\ &= a_1 + x(a_2 + x(a_3 + ...(a_n + a_{n+1}x)...)). \end{aligned} \tag{4.15}$$

Nilai $P_n(x)$ dapat dihitung secara rekursif sebagai berikut:

$$\begin{aligned} S_1 &= a_{n+1} \\ S_2 &= a_n + xS_1 \\ S_3 &= a_{n-1} + xS_2 \\ S_4 &= a_{n-2} + xS_3 \\ &\vdots \\ S_n &= a_2 + xS_{n-1} \\ S_{n+1} &= a_1 + xS_n. \end{aligned} \tag{4.16}$$

Akhirnya, $P_n(x) = S_{n+1}$.

4.2.2 Polinomial Newton: Selisih Terbagi (*Divided Difference*)

Polinomial interpolasi Newton yang kita peroleh di atas dinyatakan secara rekursif. Oleh karena itu, untuk menghitung suatu nilai dengan menggunakan polinomial berderajad n kita perlu menghitung nilai-nilai polinomial berderajad $(n-1)$, $(n-2)$, $\ldots$, 2, 1.

Sekarang kita akan membahas cara mendapatkan suatu penyajian secara eksplisit suatu polinomial interpolasi Newton dari data yang tertabulasi, dengan menggunakan sebuah metode yang dikenal sebagai metode **selisih terbagi (*divided-difference*)**.

Misalkan kita ingin mencari polinomial interpolasi $P_n(x)$ untuk menghampiri suatu fungsi $f(x)$. Untuk ini, data yang diberikan adalah $(n+1)$ titik, $(x_1,\ f(x_1))$, $(x_2,\ f(x_2))$, $\ldots$, $(x_{n+1},\ f(x_{n+1}))$. Misalkan polinomial interpolasinya kita tulis sebagai

$$\begin{aligned} P_n(x) = a_1 + a_2(x-x_1) + a_3(x-x_1)(x-x_2) + \ldots \\ + a_{n+1}(x-x_1)(x-x_2)\ldots(x-x_n) \end{aligned} \tag{4.17}$$

dan kita ingin mencari nilai-nilai koefisien $a_1, a_2, \ldots, a_n, a_{n+1}$. Perhatikan, bahwa di sini berlaku $P_n(x_k) = f(x_k)$ untuk $1 \leq k \leq (n+1)$.

Jika $x = x_1$ disubstitusikan ke dalam (4.17), maka semua suku pada

Selisih terbagi Newton

1. *Selisih terbagi ke-nol terhadap x_k:*

$$f[x_k] = f(x_k); \qquad k = 1,\ 2,\ 3,\ \ldots,\ (n+1) \tag{4.23}$$

2. *Selisih terbagi pertama terhadap x_k dan x_{k+1}:*

$$f[x_k,\ x_{k+1}] = \frac{f[x_{k+1}] - f[x_k]}{x_{k+1} - x_k}; \qquad k = 1,\ 2,\ 3,\ \ldots n \tag{4.24}$$

3. *Selisih terbagi kedua terhadap x_k, x_{k+1} dan x_{k+2}:*

$$f[x_k,\ x_{k+1},\ x_{k+2}] = \frac{f[x_{k+1},\ x_{k+2}] - f[x_k,\ x_{k+1}]}{x_{k+2} - x_k}; \qquad k = 1,\ 2,\ 3,\ \ldots (n-1) \tag{4.25}$$

4. $\ldots$

5. *Selisih terbagi ke-j terhadap x_k, x_{k+1}, $\ldots$, x_{k+j} didefinisikan secara rekursif:*

$$f[x_k,\ x_{k+1},\ \ldots,\ x_{k+j}] = \frac{f[x_{k+1},\ x_{k+2},\ \ldots,\ x_{k+j}] - f[x_k,\ x_{k+1},\ \ldots,\ x_{k+j-1}]}{x_{k+j} - x_k}; \tag{4.26}$$

untuk $k = 1,\ 2,\ 3,\ \ldots,\ n+1-j; j = 1,\ 2,\ 3,\ \ldots,\ n.$

Selisih terbagi Newton fungsi f dapat dipandang sebagai versi diskrit turunan fungsi f. Perhatikan, dari teorema nilai rata-rata kita tahu bahwa jika $f(x)$ diferensiabel pada interval yang memuat x_1 dan x_2, maka terdapat bilangan c antara x_1 dan x_2 sedemikian hingga

Hubungan selisih terbagi dan turunan

$$f'(c) = \frac{f(x_2) - f(x_1)}{x_2 - x_1} = f[x_1,\ x_2]. \tag{4.27}$$

Jadi $f[x_1,\ x_2]$ dapat dipandang sebagai nilai turunan $f(x)$. Selanjutnya, jika x_1 dan x_2 cukup dekat, maka nilai selisih terbagi pertama $f[x_1,\ x_2]$ dapat digunakan sebagai hampiran yang cukup akurat untuk $f'(\frac{x_1+x_2}{2})$.

Lemma berikut ini, yang dapat dibuktikan dengan induksi matematika (lihat [5] halaman 141 – 143) dapat digunakan untuk menunjukkan

CONTOH 4.5.
Misalkan $f(x) = \cos(x)$*,* $x_1 = 0.2$*,* $x_2 = 0.3$*,* $x_3 = 0.4$*. Dari contoh sebelumnya sudah dihitung* $f[x_1,\ x_2] = -0.2473009$*. Selisih terbagi derajad pertama yang lain adalah*

$$f[x_2,\ x_3] = \frac{\cos(0.4) - \cos(0.3)}{0.4 - 0.3} = -0.3427550.$$

Selanjutnya, selisih terbagi kedua adalah

$$f[x_1, x_2, x_3] = \frac{-0.3427550 - (-0.2473009)}{0.4 - 0.2} = -0.4772703$$

Berdasarkan(4.29)*, untuk* $n = 2$ *dapat dicari nilai*

$$-0.4772703 = \frac{1}{2} f''(c) = -\frac{1}{2}\cos(c),$$

atau $c = \cos^{-1}(0.9545406) = 0.3026814 \approx x_2$. □

Selisih terbagi Newton memiliki beberapa sifat sebagai berikut:

1. Simetris. Jika $(i_1, i_2, ..., i_n)$ menyatakan permutasi (susunan urutan) indeks $(1, 2, ..., n)$, maka

   Sifat simetris selisih terbagi

$$f[x_{i_1}, x_{i_2}, ..., x_{i_n}] = f[x_1, x_2, ..., x_n]. \tag{4.30}$$

   Sifat ini dapat dibuktikan dengan induksi matematika. Cobalah Anda buktikan untuk kausus $n = 2$ dan $n = 3$!

2. Jika didefinisikan

$$f[x_1, x_1] = \lim_{x_2 \to x_1} f[x_1, x_2] = \lim_{x_2 \to x_1} \frac{f(x_2) - f(x_1)}{x_2 - x_1} = f'(x_1),$$

   maka dapat didefinisikan

$$f[\underbrace{x_1, x_1, ..., x_1}_{n \text{ elemen}}] = \frac{1}{n!} f^{(n)}(x_1). \tag{4.31}$$

   Dengan menggunakan sifat simetris, dapat diperluas definisi selisih terbagi untuk beberapa simpul sama dan beberapa simpul lain berbeda.

TEOREMA 4.3 (POLINOMIAL NEWTON).
Misalkan fungsi $f(x)$ terdefinisi pada interval $[a,\ b]$, dan misalkan x_1, x_2, ..., x_{n+1} adalah $(n+1)$ bilangan yang berlainan pada interval $[a,\ b]$. Maka terdapat sebuah polinomial tunggal $P_n(x)$ berderajad paling tinggi n yang memenuhi:

$$f(x_k) = P_n(x_k) \text{ untuk } k = 1,\ 2,\ 3,\ \ldots,\ (n+1).$$

Polinomial Newton ini adalah

Koefisien polinomial Newton merupakah selisih terbagi fungsi yang dihampiri

$$\begin{aligned} P_n(x) = &\ a_1 + a_2(x - x_1) + a_3(x - x_1)(x - x_2) + \ldots \\ &+ a_{n+1}(x - x_1)(x - x_2)\ldots(x - x_n) \end{aligned} \tag{4.35}$$

dengan

$$a_k = f[x_1,\ x_2,\ \ldots,\ x_k] \text{ untuk } k = 1,\ 2,\ 3,\ \ldots,\ (n+1)$$

AKIBAT 4.1 (HAMPIRAN NEWTON).
Misalkan $P_n(x)$ adalah polinomial Newton yang diberikan oleh Teorema 4.3 dan digunakan untuk menghampiri fungsi $f(x)$, yakni

$$f(x) = P_n(x) + E_n(x). \tag{4.36}$$

Jika f mempunyai turunan ke-$(n+1)$ pada interval $[a,\ b]$, maka untuk setiap $x \in [a,\ b]$, terdapat sebuah bilangan $c = c(x) \in [a,\ b]$, sedemikian sehingga fungsi galat $E_n(x)$ berbentuk

Galat hampiran fungsi dengan polinomial Newton

$$E_n(x) = \frac{(x - x_1)(x - x_2)\ldots(x - x_{n+1})f^{(n+1)}(c)}{(n+1)!}. \tag{4.37}$$

Untuk keperluan komputasi, nilai-nilai selisih terbagi pada Tabel 4.1 perlu disimpan ke dalam matriks (*array*), misalkan $D(j,k)$. Jadi koefisien-koefisien a_k pada (4.35) menjadi

Cara menghitung selisih-selisih terbagi Newton suatu fungsi dengan menggunakan tabel

$$D(j,k) = f[x_k,\ x_{k+1},\ \ldots,\ x_{k+j}], \tag{4.38}$$

untuk $1 \le j \le (n+1)$ dan $1 \le k \le [(n+1) - j + 1]$.

Dari (4.26) kita dapatkan rumus rekursif untuk menghitung elemen-elemen matriks D:

$$D(1,k) = f(x_k), \quad \text{untuk} \quad k = 1,\ 2,\ ...,\ (n+1)$$

```
function D=selisihN(x,y)
n=length(x);
D(1,1:n)=y;
for j=2:n,
   for k=1:n-j+1,
      D(j,k)=(D(j-1,k+1)-D(j-1,k))/(x(k+j-1)-x(k));
   end
end
```

Gambar 4.5: Fungsi MATLAB `selisihN` untuk menghitung selisih-selisih terbagi Newton, jika diketahui data berupa pasangan dua buah vektor x dan y

CONTOH 4.6.
Hitunglah selisih-selisih terbagi fungsi f sampai tingkat tiga, jika diketahui data titik-titik sebagai berikut. Selanjutnya, tentukan polinomial Newton yang menginterpolasikan titik-titik tersebut.

x_k	0	1	2	4
$f(x_k)$	1	1	2	5

Penyelesaian:
Dari data pada tabel di atas dapat disusun tabel selisih terbagi Newton untuk fungsi f sebagai berikut. Nilai-nilai selisih terbagi Newton membentuk transpose matriks segitiga atas. Dari hasil perhitungan tersebut, elemen-elemen pada kolom pertama matriks D merupakan koefisien-koefisien polinomial Newton yang menginterpolasikan data tersebut.

x_k	0	1	2	4
$f(x_k)$	1	1	2	5
$D(1,k)$	1	1	2	5
$D(2,k)$	0	1	3/2	
$D(3,k)$	1/2	1/6		
$D(4,k)$	-1/12			


```
  -3   0   15   48   105  192
>>D=selisihN(x,y)
 D  =
  -3   0   15   48   105  192
   3  15   33   57    87    0
   6   9   12   15     0    0
   1   1    1    0     0    0
   0   0    0    0     0    0
   0   0    0    0     0    0
```

CONTOH 4.8.
Buatlah tabel selisih terbagi untuk fungsi $f(x) = \cos(x)$ dengan menggunakan lima titik $(x,\ f(x))$ untuk $x = 0,\ 1,\ 2,\ 3,\ 4.$ Dari tabel tersebut tentukan a_k dan carilah keempat polinomial interpolasi Newton $P_k(x)$ untuk $k = 1,\ 2,\ 3,\ 4.$
Penyelesaian:
Seperti pada contoh sebelumnya kita gunakan fungsi MATLAB `selisihN` *untuk menghasilkan tabel selisih terbagi Newton (matriks D) sebagai berikut:*

```
>>x=0:4
 x  =
   0  1  2  3  4
>>y=cos(x)
 y  =
   1.0000  0.5403  -0.4161  -0.9900  -0.6536
>>D=selisihN(x,y)
 D  =
   1.0000  0.5403  -0.4161  -0.9900  -0.6536
  -0.4597 -0.9564  -0.5738   0.3363   0.0000
  -0.2484  0.1913   0.4551   0.0000   0.0000
   0.1466  0.0879   0.0000   0.0000   0.0000
  -0.0147  0.0000   0.0000   0.0000   0.0000
```

Dengan menggunakan keempat titik pertama $x_1,\ x_2,\ x_3,\ x_4$ dan nilai-nilai a_1, a_2, a_3, a_4, a_5, yakni kolom pertama matriks D pada hasil perhitungan di atas, kita tuliskan keempat polinomial Newton (kita gunakan pendekatan sampai tujuh gunakan fungsi `xsarang`)[5] perhitungan nilai polinomial Newton akan lebih efisien daripada perhitungan secara langsung.

LATIHAN 4.2

1. Dengan menggunakan simpul-simpul pusat x_1, x_2, x_3, dan x_4, dan koefisien-koefisien a_1, a_2, a_3, a_4, dan a_5 yang diberikan, tentukan polinomial-polinomial Newton $P_1(x)$, $P_2(x)$, $P_3(x)$, dan $P_4(x)$. Selanjutnya, hitunglah nilai-nilai $P_1(c)$, $P_2(c)$, $P_3(c)$, dan $P_4(c)$ untuk nilai c yang diberikan. (*Gunakan perhitungan secara manual dan bandingkan hasilnya dengan menggunakan fungsi-fungsi MATLAB yang diberikan di depan.*)

   (a) $a_1 = 4$, $a_2 = -1$, $a_3 = 0.4$, $a_4 = 0.01$, $a_5 = -0.02$; $x_1 = 1$, $x_2 = 3$, $x_3 = 4$, $x_4 = 4.5$; $c = 2.5$

   (b) $a_1 = 5$, $a_2 = -2$, $a_3 = 0.5$, $a_4 = -0.1$, $a_5 = -0.003$; $x_1 = 0$, $x_2 = 1$, $x_3 = 2$, $x_4 = 3$; $c = 2.5$

   (c) $a_1 = 7$, $a_2 = 3$, $a_3 = 0.1$, $a_4 = 0.05$, $a_5 = -0.04$; $x_1 = -1$, $x_2 = 0$, $x_1 = 4$, $x_4 = 4$; $c = 3$

   (d) $a_1 = -2$, $a_2 = 4$, $a_3 = -0.04$, $a_4 = 0.06$, $a_5 = 0.005$; $x_1 = -3$, $x_2 = -1$, $x_3 = 1$, $x_4 = 4$; $c = 2$

2. Dengan menggunakan data sebagai berikut, (i) Hitunglah tabel selisih terbagi Newton untuk fungsi yang diberikan; (ii) Tuliskan polinomial-polinomial Newton $P_1(x)$, $P_2(x)$, $P_3(x)$, dan $P_4(x)$; (iii) Hitunglah nilai-nilai $P_1(c)$, $P_2(c)$, $P_3(c)$, dan $P_4(c)$ untuk nilai-nilai c yang diberikan; (iv) Bandingkan nilai-nilai $P_1(c)$, $P_2(c)$, $P_3(c)$, dan $P_4(c)$ dengan nilai $f(c)$.

   (a) $f(x) = 3 \times 2^x$, $c = 1.5,\ 2.5$

[5] Ganti input `a` untuk fungsi tersebut dengan `D(:,1)` hasil output fungsi `selisihN`.

(c) $(0,5),\ (1,5),\ (2,3),\ (3,5),\ (4,17),\ (5,45),\ (6,95)$

(d) $(0,1),\ (1,-1),\ (4,1),\ (6,-1)$

(e) $(4,1),\ (1,-1),\ (6,-1),\ (0,1)$

(f) $(0,0),\ (1,16),\ (2,48),\ (4,88),\ (5,0)$

4. Buatlah tabel selisih terbagi dari data berikut ini, kemudian hitung $P_3(1)$:

x_i :	-2	0	3	4
$f(x_i)$:	5	1	55	209

Tambahkan data titik-titik $(-1,-1)$ dan $(2,5)$ satu demi satu dan hitung nilai-nilai $P_4(1)$ dan $P_5(1)$. Berikan komentar terhadap hasil yang Anda peroleh.

5. Berikut adalah nilai-nilai fungsi $f(x) = \ln x$ pada beberapa titik sampel. Gunakan sebanyak mungkin titik untuk menaksir nilai $f(2.6)$ dengan toleransi 0.0000 05. Lakukan perhitungan dengan urutan titik-titik yang berbeda dan cobalah untuk meminimumkan banyaknya titik yang diperlukan.

x_i :	1.5	1.7	1.8	1.9	2.2	2.3	2.5	2.7
$f(x_i)$:	0.40547	0.53063	0.58779	0.64185	0.78846	0.83291	0.91629	0.99325

x_i :	2.9	3.1	3.2	3.4	3.5	3.6	3.7	4.0
$f(x_i)$:	1.06471	1.13140	1.16315	1.22378	1.25276	1.28093	1.30833	1.38629

6. Tunjukkan bahwa, jika x_1, x_2, dan x_3 berbeda maka

$$f[x_1,x_2,x_3] = f[x_2,x_3,x_1] = f[x_3,x_1,x_2].$$

7. Bentuk selisih terbagi polinomial interpolasi $P_2(x)$ adalah

$$P_2(x) = f[x_1] + f[x_1,x_2](x-x_1) + f[x_1,x_2,x_3](x-x_1)(x-x_2).$$

Dengan menyatakan selisih-selisih terbagi tersebut dalam nilai-nilai fungsi $f(x_i)$, (i=1, 2, 3), jelaskan bahwa P_2 melalui titik-titik $(x_i,\ f(x_i))$, (i=1, 2, 3).

8. Perhatikan $(n+1)$ titik $(x_1,y_1),\ (x_2,y_2),\ \ldots,\ (x_{n+1},y_{n+1})$.

ngan mendefinisikan

$$L_1(x) = \frac{(x - x_2)}{(x_1 - x_2)} \text{ dan } L_2(x) = \frac{(x - x_1)}{(x_2 - x_1)}$$

kita dapat menuliskan (4.41) sebagai

Penulisan persamaan garis lurus dalam bentuk polinomial Lagrange

$$P_1(x) = y_1 L_1(x) + y_2 L_2(x). \tag{4.42}$$

Bentuk terakhir ini dapat diperumum untuk $P_2(x)$, $P_3(x)$, $\ldots$ dengan menggunakan tiga, empat titik, dan seterusnya. Hasilnya adalah polinomial-polinomial Lagrange. Polinomial Lagrange dapat digunakan untuk menginterpolasikan tabel dengan n nilai, terutama apabila data titik-titik yang diberikan memiliki interval yang berbeda-beda. Misalkan tabel yang diberikan seperti di bawah ini:

x	x_1	x_2	x_3	$\ldots$	x_n
y	$f(x_1)$	$f(x_2)$	$f(x_3)$	$\ldots$	$f(x_n)$

Sekarang perhatikan polinomial-polinomial berderajad $(n - 1)$ di bawah ini

$$L_k(x) = \frac{(x - x_1)\ldots(x - x_{k-1})(x - x_{k+1})\ldots(x - x_n)}{(x_k - x_1)\ldots(x_k - x_{k-1})(x_k - x_{k+1})\ldots(x_k - x_n)}, \tag{4.43}$$

untuk $k = 1,\ 2,\ ...,\ n$.

Perhatikan juga bahwa faktor-faktor $(x - x_k)$ dan $(x_k - x_k)$ tidak muncul di dalam pembilang dan penyebut pada ruas kanan $L_k(x)$. Notasi lain yang lebih ringkas untuk fungsi $L_k(x)$ adalah[6]

Fungsi kardinal untuk polinomial Lagrange berderajad $(n - 1)$

$$L_k(x) = \prod_{\substack{j=1\\ j\neq k}}^{n} \frac{(x - x_j)}{(x_k - x_j)}. \tag{4.44}$$

Dari definisi L_k di atas, kita lihat bahwa fungsi-fungsi L_k memenuhi sifat

Sifat fungsi kardinal Lagrange

$$L_k(x_j) = \begin{cases} 1 & \text{jika } j = k \\ 0 & \text{jika } j \neq k. \end{cases}$$

[6]Perlu dicatat bahwa definisi fungsi $L_k(x)$ adalah relatif terhadap himpunan simpul pusat yang dipakai. Beberapa buku menggunakan indeks kedua pada L yang menyatakan banyaknya titik yang dipakai, misalkan $L_{n,k}(x)$ sebagai pengganti $L_k(x)$.

daan permasalahan pada keduanya. Jika pada polinomial Newton permasalahan utamanya adalah mencari koefisien-koefisien, maka pada polinomial Lagrange permasalahan utamanya lebih pada perhitungan nilai polinomial itu sendiri[7].

Perhitungan nilai polinomial Lagrange (4.45) yang berderajad $(n-1)$ memerlukan perhitungan n nilai fungsi kardinal $L_k(x)$. Algoritma 4.3 menunjukkan langkah-langkah perhitungan nilai polinomila Lagrange. Implementasikan algoritma tersebut dengan MATLAB ditunjukkan pada Gambar 4.7 dan 4.8.

CONTOH 4.9.
Tentukan polinomial Lagrange yang menginterpolasikan titik-titik $(0,1)$, $(1,1)$, $(2,2)$, *dan* $(4,5)$.
Penyelesaian:
Berdasarkan rumus (4.44) *dan* (4.45), *polinomial Lagrange yang dicari adalah*

$$\begin{aligned} P_3(x) = \frac{(x-1)(x-2)(x-4)}{(0-1)(0-2)(0-4)}1 \quad &+ \quad \frac{x(x-2)(x-4)}{1(1-2)(1-4)}1 \\ &+ \quad \frac{x(x-1)(x-4)}{2(2-1)(2-4)}2 + \frac{x(x-1)(x-2)}{4(4-1)(4-2)}5, \end{aligned}$$

atau

$$P_3(x) = \frac{1}{12}(-x^3 + 9x^2 - 8x + 12).$$

Perhatikan bahwa soal di atas sama dengan soal pada Contoh 4.6. Silakan Anda cek bahwa polinomial Lagrange di atas persis sama dengan polinomial Newton pada contoh 4.6. □

CONTOH 4.10.
Dari tabel

x	*0*	*1*	*-1*	*2*	*-2*	*3*
f(x)	*-3*	*-2*	*5*	*10*	*16*	*-10*

carilah fungsi-fungsi kardinalnya dan polinomial interpolasi Lagrange. Hitung nilai pendekatan $f(-3)$ *dengan menggunakan polinomial tersebut.*

Penyelesaian:

[7] Pembaca yang tertarik akan tertantang untuk menunjukkan apakah polinomial Newton dan Lagrange yang menginterpolasikan titik-titik yang sama adalah identik!

Berikut kita hitung fungsi-fungsi $L_1,\ L_2,\ L_3,\ L_4,\ L_5$, dan L_6:

$$
\begin{aligned}
L_1(x) &= \frac{(x-1)(x+1)(x-2)(x+2)(x-3)}{(0-1)(0+1)(0-2)(0+2)(0-3)} \\
&= \frac{(x-1)(x+1)(x-2)(x+2)(x-3)}{-12} \\
L_2(x) &= \frac{(x-0)(x+1)(x-2)(x+2)(x-3)}{(1-0)(1+1)(1-2)(1+2)(1-3)} \\
&= \frac{x(x+1)(x-2)(x+2)(x-3)}{12} \\
L_3(x) &= \frac{(x-0)(x-1)(x-2)(x+2)(x-3)}{(-1-0)(-1-1)(-1-2)(-1+2)(-1-3)} \\
&= \frac{x(x-1)(x-2)(x+2)(x-3)}{24} \\
L_4(x) &= \frac{(x-0)(x-1)(x+1)(x+2)(x-3)}{(2-0)(2-1)(2+1)(2+2)(2-3)} \\
&= \frac{x(x-1)(x+1)(x+2)(x-3)}{-24} \\
L_5(x) &= \frac{(x-0)(x-1)(x+1)(x-2)(x-3)}{(-2-0)(-2-1)(-2+1)(-2-2)(-2-3)} \\
&= \frac{x(x-1)(x+1)(x-2)(x-3)}{-120} \\
L_6(x) &= \frac{(x-0)(x-1)(x+1)(x-2)(x+2)}{(3-0)(3-1)(3+1)(3-2)(3+2)} \\
&= \frac{x(x-1)(x+1)(x-2)(x+2)}{120}
\end{aligned}
$$

Jadi polinomial Lagrange yang dicari adalah

$$P_5(x) = -3L_1(x) - 2L_2(x) + 5L_3(x) + 10L_4(x) + 16L_5(x) - 10L_6(x).$$

Untuk mencari pendekatan nilai $f(-3)$, kita hitung $P_5(-3)$ sebagai berikut

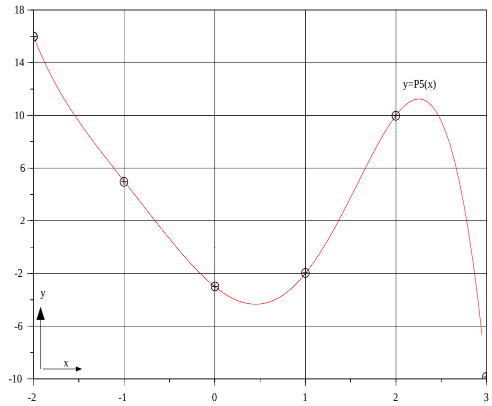
$$P_5(-3) = -3L_1(-3) - 2L_2(-3) + 5L_3(-3) + 10L_4(-3) + 16L_5(-3) - 10L_6(-3)$$

dengan

```
 y  =
   -3  -2  5  10  16 -10
>>L=polag(x,y,-3)
 L  =
    20  -15  -15  6   6  -1
>>P=y*L'
 P  =
     61
```

Cara langsung adalah dengan menggunakan fungsi MATLAB `nilaipolag`*:*

```
>>P=nilaipolag(x,y,-3)
```



Gambar 4.9: Polinomial Lagrange $P_5(x)$ yang menginterpolasikan titik-titik $(0,\ -3)$, $(1,\ -2)$, $(-1,\ 5)$, $(2,\ 10)$, $(-2,\ 16)$, dan $(3,\ -10)$

Gambar 4.9 menyajikan grafik fungsi polinomial Lagrange yang menginterpolasikan keenam titik yang diberikan pada tabel. □

Sekarang marilah kita cari polinomial-polinomial Lagrange lain yang menginterpolasikan dua titik pertama, tiga titik pertama, …dan membandingkannya[8].

[8]Di sini kita gunakan indeks kedua untuk menyatakan banyaknya titik yang dipakai

pada tabel:

$$\begin{aligned}
L_{5,1}(x) &= \frac{(x-1)(x+1)(x-2)(x+2)}{(0-1)(0+1)(0-2)(0+2)} \\
&= \frac{(x-1)(x+1)(x-2)(x+2)}{4} \\
L_{5,2}(x) &= \frac{(x-0)(x+1)(x-2)(x+2)}{(1-0)(1+1)(1-2)(1+2)} \\
&= \frac{-x(x+1)(x-2)(x+2)}{6} \\
L_{5,3}(x) &= \frac{(x-0)(x-1)(x-2)(x+2)}{(-1-0)(-1-1)(-1-2)(-1+2)} \\
&= -x(x-1)(x-2)(x+2)/6 \\
L_{5,4}(x) &= \frac{(x-0)(x-1)(x+1)(x+2)}{(2-0)(2-1)(2+1)(2+2)} \\
&= x(x-1)(x+1)(x+2/24
\end{aligned}$$

$$\begin{aligned}
L_{5,5}(x) &= \frac{(x-0)(x-1)(x+1)(x-2)}{(-2-0)(-2-1)(-2+1)(-2-2)} \\
&= x(x-1)(x+1)(x-2)/24 \\
P_4(x) &= \frac{-3}{4}(x-1)(x+1)(x-2)(x+2) \\
&\quad + \frac{1}{3}x(x+1)(x-2)(x+2) \\
&\qquad - \frac{-5}{6}x(x-1)(x-2)(x+2) + \frac{5}{12}x(x-1)(x+1)(x+2) \\
&\qquad\quad + \frac{2}{3}x(x-1)(x+1)(x-2).
\end{aligned}$$

Untuk membandingkan keempat polinomial Lagrange tersebut dengan polinomial Lagrange yang menginterpolasikan keenam titik yang diberikan, kita lihat grafik keenam fungsi tersebut pada Gambar 4.10. Terlihat bahwa $P_5(x)$ memberikan interpolasi terbaik untuk keenam titik yang diberikan.

dengan $P_n(x)$ adalah suatu polinomial Lagrange hampiran $f(x)$:

$$f(x) \approx P_n(x) = \sum_{k=1}^{n+1} f(x_k)L_k(x), \tag{4.47}$$

maka suku galat $E_n(x)$ berbentuk

Galat hampiran fungsi dengan polinomial Lagrange

$$E_n(x) = \frac{(x-x_1)(x-x_2)\ldots(x-x_{n+1})f^{n+1}(c)}{(n+1)!} \tag{4.48}$$

untuk suatu nilai $c = c(x) \in [a,\ b]$.

BUKTI: Oleh karena $P_n(x_k) = f(x_k)$ untuk $k = 1,\ 2,\ \ldots,\ (n+1)$, maka

$$E_n(x_k) = 0 = P_n(x_k) - f(x_k) \quad \text{untuk} \quad k = 1,\ 2,\ \ldots,\ (n+1).$$

Jadi rumus (4.48) benar untuk $x = x_k$, $k = 1,\ 2,\ \ldots,\ (n+1)$. Oleh karena itu kita hanya akan meninjau kasus $x \neq x_k$, $k = 1,\ 2,\ \ldots,\ (n+1)$, dan definisikan fungsi

$$G(t) = f(t) - P_n(t) - \frac{\prod_{k=1}^{n+1}(t-x_k)}{\prod_{k=1}^{n+1}(x-x_k)}E_n(x), \tag{4.49}$$

dengan t bervarisai dan x tetap, keduanya pada $[a,\ b]$. Fungsi $G(t)$ memenuhi sifat-sifat sebagai berikut

1. $G(x_k) = f(x_k) - P_n(x_k) - 0 = f(x_k) - f(x_k) = 0$ untuk $k = 1, 2, \ldots, (n+1)$ dan $G(x) = f(x) - P_n(x) - E_n(x) = 0$ (dari (4.46)).

2. Oleh karena $P_n(t)$ adalah polinomial berderajad n dan $\prod_{k=1}^{n+1}(t-x_k)$ adalah polinomial berderajad $(n+1)$ dalam t, maka dari persamaan (4.49) dan asumsi mengenai f jelas bahwa $G'(t)$, $G''(t)$, ..., $G^{(n+1)}(t)$ ada dan kontinyu pada $[a,\ b]$.

Jadi menurut Teorema Role Umum (lihat Teorema A.8 pada Lampiran A), terdapat $c \in (a,\ b)$ sedemikian hingga $G^{(n+1)}(c) = 0$. Dari persamaan (4.49) diperoleh

$$G^{(n+1)}(t) = f^{(n+1)}(t) - 0 - \frac{(n+1)!E_n(x)}{\prod_{k=1}^{n+1}(x-x_k)}, \tag{4.50}$$

hingga

$$|f^{(2)}(c)| \leq M_2, \quad \text{untuk} \quad x_1 \leq c \leq x_2.$$

Di sisi lain,

$$|t(t-h)| \leq h^2/4, \quad \text{untuk} \quad 0 \leq t \leq h.$$

Jadi

$$|E_1(x)| \leq \frac{h^2 M_2}{2!4} = \frac{h^2 M_2}{8},$$

sesuai dengan pertidaksamaan (4.52). ■

Teorema 4.5 memberikan hubungan antara besar suku-suku galat pada interpolasi linier, kuadratik, dan kubik. Pada setiap kasus batas galat $|E_n(x)|$ tergantung pada h dalam dua hal. Pertama, adanya faktor h^{n+1} sehingga $|E_n(x)|$ sebanding dengan h^{n+1}. Kedua, nilai-nilai M_{n+1} pada umumnya juga tergantung pada h dan nilainya menuju $f^{(n+1)}(x_1)$ jika h menuju nol. Jadi, jika h menuju nol $E_n(x)$ konvergen ke nol secepat h^{n+1} konvergen ke nol. Perilaku seperti ini dinyatakan secara simbolik dengan notasi $O(h^{n+1})$. Sebagai contoh, batas galat (4.52)

$$|E_1(x)| = O(h^2) \quad \text{untuk } x \in [x_1, \ x_2]$$

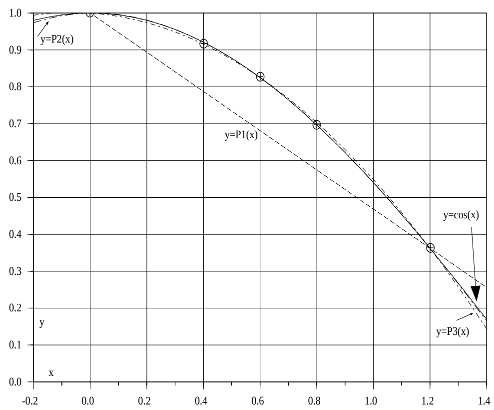
yang berarti batas galat tersebut sebanding dengan kelipatan h^2.

Sebagai konsekuensi hal tersebut, jika turunan $f(x)$ terbatas seragam pada interval tersebut dan $|h| \leq 1$, maka dengan memilih n cukup besar akan membuat nilai h^{n+1} kecil, dan polinomial hampiran berderajad yang lebih tinggi memberikan galat yang lebih kecil.

CONTOH 4.11.

Perhatikan fungsi $y = f(x) = \cos(x)$ *pada interval* $[0.0, \ 1.2]$.

1. *Gunakan titik-titik* $x_1 = 0.0$ *dan* $x_2 = 1.2$ *untuk menentukan polinomial interpolasi linier* $P_1(x)$.
2. *Gunakan titik-titik* $x_1 = 0.0$, $x_2 = 0.6$ *dan* $x_3 = 1.2$ *untuk menentukan polinomial interpolasi kuadratik* $P_2(x)$.
3. *Gunakan titik-titik* $x_1 = 0.0$, $x_2 = 0.4$, $x_3 = 0.8$, *dan* $x_4 = 1.2$ *untuk menentukan polinomial interpolasi kubik* $P_3(x)$.
4. *Tentukan batas-batas galat hampiran yang diberikan oleh* $P_1(x)$, $P_2(x)$, *dan* $P_3(x)$.



Gambar 4.11: Grafik $y = \cos(x)$ dan tiga polinomial Lagrange $P_1(x)$ (berdasarkan $x_1 = 0.0$ dan $x_2 = 1.2$), $P_2(x)$ (berdasarkan $x_1 = 0.0,\ x_2 = 0.6$ dan $x_3 = 1.2$), dan $P_3(x)$ (berdasarkan $x_1 = 0.0,\ x_2 = 0.4,\ x_3 = 0.8$ dan $x_4 = 1.2$)

4. *Untuk menghitung batas-batas $E_1(x),\ E_2(x)$, dan $E_3(x)$ kita perlu menghitung batas-batas $M_2,\ M_3$, dan M_4 untuk $|f^{(2)}(x)|,\ |f^{(3)}(x)|$, dan $|f^{(4)}(x)|$ sbb.: $|f^{(2)}(x)| = |-\cos(x)| \leq |-\cos(0.0)| = 1.000000$ sehingga $M_2 = 1.000000$, $|f^{(3)}(x)| = |\sin(x)| \leq |\sin(1.2)| = 0.932039$ sehingga $M_3 = 0.932039$, $|f^{(4)}(x)| = |\cos(x)| \leq |\cos(0.0)| = 1.000000$ sehingga $M_4 = 1.000000$. Untuk $P_1(x)$ lebar interval antar simpul adalah $h = 1.2$, sehingga batas galatnya adalah*

$$|E_1(x)| \leq \frac{h^2 M_2}{8} = \frac{(1.2)^2(1.000000)}{8} = 0.180000$$

Untuk $P_2(x)$ lebar interval antar simpul adalah $h = 0.6$, sehingga batas galatnya adalah

$$|E_2(x)| \leq \frac{h^3 M_3}{9\sqrt{3}} = \frac{(0.6)^3(0.932039)}{9\sqrt{3}} = 0.012915$$

Jawab:

1. *Dengan menggunakan rumus* (4.48) *galat hampiran tersebut adalah*

$$E_1(x) = e^x - P_1(x) = \frac{(x - x_1)(x - x_2)}{2} e^c,$$

untuk suatu c antara minimum dan maksimum x_1, x_2, dan x. Sesuai dengan tujuan interpolasi, misalkan $0 \le x_1 \le x \le x_2 \le 1$. Fungsi galat tersebut merupakan fungsi kuadrat dalam x dengan pembuat nol x_1 dan x_2, sehingga nilai ekstrimnya terjadi pada $x = (x_1 + x_2)/2$. Jadi

$$|E_1(x)| \le \frac{1}{8} |(x_2 - x_1)(x_1 - x_2)| e^c.$$

Dengan menuliskan $h = x_2 - x_1$ dan memperhatikan bahwa $e^c < e$ untuk $c \in [0,\ 1]$, maka diperoleh

$$|E_1(x)| \le \frac{h^2 e}{8}.$$

2. *Dengan menggunakan polinomial Lagrange linier kita hitung $P_1(0.826)$ sebagai berikut*

$$P_1(0.826) = 0.270500 \frac{(0.826 - 0.83)}{(0.82 - 0.83)} + 2.293319 \frac{(0.826 - 0.82)}{(0.83 - 0.82)} = 2.2841914.$$

Dengan $h = 0.01$, galat hampiran linier memenuhi hubungan

$$E|_1(x)| \le \frac{(0.01)^2 e}{8} = 0.0000340,$$

Galat yang sebenarnya pada hampiran linier tersebut adalah 0.0000276, lebih kecil daripada batas tersebut. □

LATIHAN 4.3

1. Bentuk polinomial interpolasi kubik untuk menghampiri fungsi $f(x)$ dengan menggunakan data berikut ini, kemudian hitung tak-

ii. hitung $P_2(x)$ dengan menggunakan $x_1 = -1$, $x_2 = 0$ dan $x_3 = 1$;
iii. hitung $P_3(x)$ dengan menggunakan $x_1 = -1$, $x_2 = 0$, $x_3 = 1$ dan $x_4 = 2$;
iv. hitung $P_1(x)$ dengan menggunakan $x_1 = 1$ dan $x_2 = 2$;
v. hitung $P_2(x)$ dengan menggunakan $x_1 = 0$, $x_2 = 1$ dan $x_3 = 2$;

(b) $f(x) = x + 2/x$,
   i. hitung $P_2(x)$ dengan menggunakan $x_1 = 0$, $x_2 = 1$ dan $x_3 = 2.5$ untuk menghampiri $f(1.5)$ dan $f(1.2)$;
   ii. hitung $P_3(x)$ dengan menggunakan $x_1 = 0.5$, $x_2 = 1$, $x_3 = 2$ dan $x_4 = 2.5$ untuk menghampiri $f(1.5)$ dan $f(1.2)$;

(c) $f(x) = 8x/2^x$,
   i. hitung $P_2(x)$ dengan menggunakan $x_1 = 0$, $x_2 = 1$ dan $x_3 = 2.5$ untuk menghampiri $f(1.5)$ dan $f(1.3)$;
   ii. hitung $P_3(x)$ dengan menggunakan $x_1 = 0$, $x_2 = 1$, $x_3 = 2$ dan $x_4 = 3$ untuk menghampiri $f(1.5)$ dan $f(1.3)$;

(d) $f(x) = 2\sin(x\pi/6)$, x dalam radian,
   i. hitung $P_2(x)$ dengan menggunakan $x_1 = 0$, $x_2 = 1$ dan $x_3 = 3$ untuk menghampiri $f(2)$, $f(2.4)$, $f(4)$, dan $f(3.5)$;
   ii. hitung $P_3(x)$ dengan menggunakan $x_1 = 0$, $x_2 = 1$, $x_3 = 3$ dan $x_4 = 5$ untuk menghampiri $f(2)$, $f(2.4)$, $f(4)$, dan $f(3.5)$;

6. Buatlah sketsa grafik $L_{5,k}$ dengan $x_k = k$ untuk $k = 1,\ 2,\ 3,\ 4,\ 5$.

7. Tentukan suku galat $E_3(x)$ pada interpolasi kubik terhadap $f(x)$ berdasarkan simpul-simpul $x_1 = -1$, $x_2 = 0$, $x_3 = 3$ dan $x_4 = 4$, untuk fungsi-fungsi
   (a) $f(x) = 4x^3 - 3x + 2$
   (b) $f(x) = x^4 - 2x^3$
   (c) $f(x) = x^5 - 5x^4$.

8. Dengan memperhatikan suku galat pada Teorema 4.4, tunjukkan bahwa polinomial interpolasi $P_n(x)$ akan menghampiri setiap polinomial berderajad paling tinggi n secara eksak. Turunkan bahwa $\sum_{k=1}^{n+1} L_{n,k}(x) = 1$ untuk semua nilai x.

ngan menyelesaikan SPL

$$\begin{aligned} A + x_1 B + y_1 C &= z_1 \\ A + x_2 B + y_2 C &= z_2 \\ A + x_3 B + y_3 C &= z_3. \end{aligned}$$

Hitunglah A, B, dan C jika diketahui $z = P(x, y, z)$ melalui ketiga titik di bawah ini.

(a) $(1, 1, 5)$, $(2, 1, 3)$ dan $(1, 2, 9)$;

(b) $(1, 1, 2.5)$, $(2, 1, 0)$ dan $(1, 2, 4)$;

(c) $(2, 1, 5)$, $(1, 3, 7)$ dan $(3, 2, 4)$;

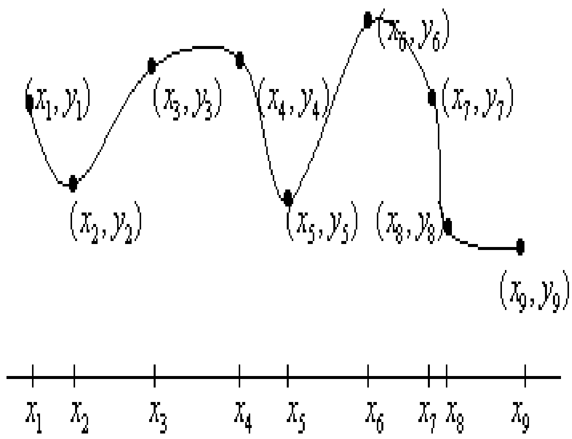
(d) $(1, 2, 5)$, $(3, 2, 7)$ dan $(1, 2, 0)$; Apa yang Anda temukan di sini? Mengapa?.

4.4 Interpolasi dengan Spline

Interpolasi dengan polinomial sering memberikan hasil yang tak dapat diterima. Polinomial interpolasi yang dihasilkan dari sejumlah besar data titik biasanya berderajad tinggi. Polinomial berderajad tinggi biasanya bersifat osilatif (grafiknya naik turun secara cepat). Akibatnya, perubahan data pada interval kecil dapat menyebabkan fluktuasi yang besar pada keseluruhan interval. Karena alasan ini, biasanya interpolasi hanya menggunakan polinomial berderajad rendah.

Polinomial berderajad tinggi tidak cocok untuk interpolasi karena biasanya bersifat osilatif (grafiknya naik turun secara cepat).

Dengan membatasi derajad polinomial interpolasi, diperoleh alternatif lain untuk mendapatkan sebuah kurva mulus yang melalui sejumlah titik. Caranya adalah interval yang memuat data titik dibagi menjadi beberapa subinterval dan pada setiap subinterval disusun polinomial interpolasi. Hasilnya sebuah kurva yang terdiri atas potongan-potongan kurva polinomial yang berderajad sama. Gambar 4.13 menunjukkan sketsa interpolasi dengan spline. Setiap subinterval diinterpolasikan dengan menggunakan suatu polinomial. Bandingkan dengan interpolasi dengan sebuah polinomial tunggal, seperti terlihat pada Gambar 4.14. Kedua interpolasi tersebut memiliki kurva yang sangat berlainan.



Gambar 4.14: Polinomial interpolasi yang melalui sekumpulan titik yang diberikan

dengan $S_k(x) = a_k x + b_k$, k =1, 2, ..., $(n-1)$. Oleh karena $S_k(x)$ linier, $S(x)$ sepotong-sepotong linier.

Misalkan $x_1 = a$ dan $x_n = b$, maka domain $S(x)$ adalah $[a,\ b]$. Selanjutnya kita menyaratkan bahwa $S(x)$ kontinyu pada $[a,\ b]$. Jadi, $S(x)$ harus memiliki sifat-sifat sebagai berikut:

1. $S(x)$ sepotong-sepotong linier, dan
2. $S(x)$ kontinyu pada $[a,\ b]$.

Untuk tujuan ekstrapolasi kita mengasumsikan

1. $S(x)$ didefinisikan sama dengan $S_1(x)$ untuk $x < a$, dan
2. $S(x)$ didefinisikan sama dengan $S_{n-1}(x)$ untuk $x > a$.

Konstanta-konstanta a_k dan b_k dipilih sedemikian hingga $S(x)$ kontinyu pada $[a,\ b]$. Syarat kekontinyuan ini bersama dengan syarat interpolasi memberikan persamaan-persamaan

1. $S_k(x_k) = f_k$ atau $a_k x_k + b_k = f_k$, untuk k =1, 2, ..., $(n-1)$;

$(n-1)$,

$$S_k(x) = f_k \frac{x - x_{k+1}}{x_k - x_{k+1}} + f_{k+1} \frac{x - x_k}{x_{k+1} - x_k}, \quad x_k \le x \le x_{k+1}. \tag{4.58}$$

Spline linier dengan polinomial Lagrange linier

```
function [a,b]=spliner(x,f)
% menghitung koefisien-koefisien spliner linier
% S_k(x)=a_kx+b_k, k=1,2,...,(n-1); x_k=< x =<x_{k+1}
n=length(x);
for k=1:(n-1),
    a(k)=(f(k+1)-f(k))/(x(k+1)-x(k));
    b(k)=f(k)-a(k)*x(k);
end
```

Gambar 4.16: Fungsi MATLAB `spliner` untuk menghitung koefisien-koefisien pada spline linier

Ekspresi lain yang ekivalen dengan (4.58) adalah dalam bentuk persamaan garis yang melalui dua titik atau polinomial Newton linier

$$S_k(x) = f_k + m_k(x - x_k), \quad x_k \le x \le x_{k+1} \tag{4.59}$$

dengan $m_k = \frac{f_{k+1}-f_k}{x_{k+1}-x_k}$. Bentuk persamaan (4.59) lebih baik daripada persamaan (4.58) untuk keperluan kalkulasi secara eksplisit nilai-nilai $S(x)$. Perhatikan, bahwa apabila penyelesaian (4.56) dan (4.57) dimasukkan ke dalam persamaan (4.58) akan diperoleh persamaan (4.59). Gambar 4.17 menyajikan fungsi MATLAB `interspliner` yang merupakan implementasi persamaan (4.59) untuk menghitung nilai spline linier $S(z)$. Fungsi `interspliner` dapat digunakan untuk menghitung nilai $S(z)$ untuk z berupa sebuah bilangan atau sebuah vektor (menghitung nilai S di beberapa titik sekaligus).

yang berarti $f(x)$ kontinyu di $x = 0.5$.

Perhatikan lagi,

$$\lim_{x\to 2.0^-} f(x) = \lim_{x\to 2.0^-} 0.5 + 2(x - 0.5) = 3.5,$$

dan

$$\lim_{x\to 2.0^+} f(x) = \lim_{x\to 2.0^+} x + 1.5 = 3.5.$$

Jadi,

$$\lim_{x\to 2.0^-} f(x) = \lim_{x\to 2.0^+} f(x) = f(2.0) = 3.5,$$

yang berarti $f(x)$ kontinyu di $x = 2.0$. Jadi $f(x)$ merupakan sebuah spline linier pada $[-1,\ 4]$. □

Berikut disajikan contoh pemakaian kedua fungsi MATLAB `spliner` dan `interspliner` untuk menghasilkan dan menghitung spline linier dari sekumpulan data titik.

CONTOH 4.14.

Contoh pemakaian fungsi MATLAB `spliner` dan `interspliner`

Tentukan spline linier yang menginterpolasikan data

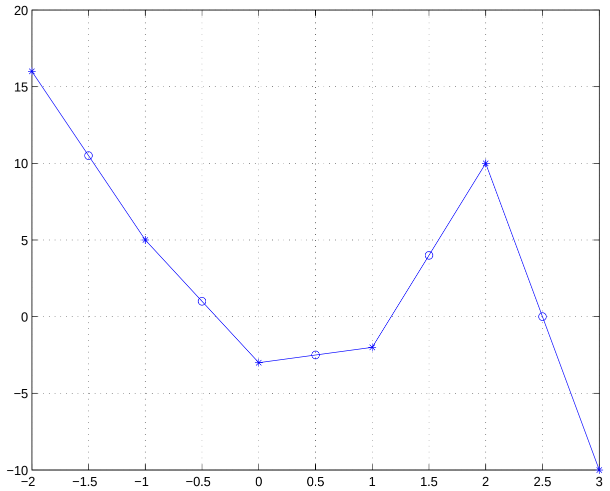
x	-2	-1	0	1	2	3
$f(x)$	16	5	-3	-2	10	-10

dan hitung nilai-nilai $S(z)$ untuk $z = -1.5,\ -0.5,\ 0.5,\ 1.5,\ 2.5$.

Penyelesaian:

Kita gunakan fungsi MATLAB `spliner` dan `interspliner` dengan memasukkan data kedua vektor x dan f sebagai berikut.

```
>>x=[-2 -1 0 1 2 3]
 x =
    -2  -1  0   1   2  3
>>f=[16 5 -3 -2 10 -10]
 f =
    16  5  -3  -2  10 -10
>>[a,b]=spliner(x,f)
 a =
    -11  -8  1  12  -20
 b =
   -6  -3  -3  -14  50
>>x1=[-1.5 -.5 .5 1.5 2.5]
```



Gambar 4.18: Contoh interpolasi dengan spliner linier (simbol 'o') berdasarkan titik-titik '*'

Sekarang kita bahas spline kuadratik atau spline berderajad dua. Suatu fungsi $S(x)$ merupakan sebuah spline berderajad dua pada $[a,\ b]$, jika $S(x)$ memiliki sifat-sifat sebagai berikut:

Syarat-syarat spline kuadratik

1. $S(x)$ sepotong-sepotong kuadratik pada $[a,\ b]$,
2. $S(x)$ kontinyu pada $[a,\ b]$, dan
3. $S'(x)$ kontinyu pada $[a,\ b]$.

Untuk tujuan ekstrapolasi kita mengasumsikan

1. $S(x)$ didefinisikan sama dengan $S_1(x)$ untuk $x < a$, dan
2. $S(x)$ didefinisikan sama dengan $S_{n-1}(x)$ untuk $x > a$.

Dalam hal ini fungsi $S(x)$ didefinisikan sebagai

Definisi spline kuadratik

$$S(x) = \begin{cases} S_1(x) & \text{untuk} \quad x_1 \le x \le x_2 \\ S_2(x) & \text{untuk} \quad x_2 \le x \le x_3 \\ S_3(x) & \text{untuk} \quad x_3 \le x \le x_4 \\ & \vdots \\ S_{n-1}(x) & \text{untuk} \quad x_{n-1} \le x \le x_n \end{cases} \tag{4.60}$$

kuadratik, untuk $k = 1, 2, ..., (n-1)$,

$$S_k(x) = \frac{m_{k+1} - m_k}{2(x_{k+1} - x_k)}(x - x_k)^2 + m_k(x - x_k) + f_k \qquad (4.62)$$

dengan $x_k \le x \le x_{k+1}$.

Bentuk lain persamaan spline kuadratik

Oleh karena $S(x)$ harus kontinyu, maka harus dipenuhi $S_k(x_{k+1}) = f_{k+1}$. Jadi,

$$\frac{m_{k+1} - m_k}{2(x_{k+1} - x_k)}(x_{k+1} - x_k)^2 + m_k(x_{k+1} - x_k) + f_k = f_{k+1},$$

atau

$$\frac{m_{k+1} - m_k}{2} + m_k = \frac{f_{k+1} - f_k}{(x_{k+1} - x_k)},$$

yakni

SPL untuk membentuk spline kuadratik

$$m_k + m_{k+1} = 2\frac{f_{k+1} - f_k}{x_{k+1} - x_k}, \qquad k = 1,\ 2, ..., (n-1).$$

Jadi spline kuadratik sekarang dapat dibentuk dengan menyelesaikan sebuah SPL yang terdiri atas $(n-1)$ persamaan linier dalam n variabel, m_1, m_2, ..., m_n. Apabila disyaratkan $m_1 = 0$ atau $m_n = 0$, maka spline yang didapat disebut **spline kuadratik alami**. Polinomial-polinomial kuadratiknya diperoleh dari persamaan (4.62). Dengan syarat awal $m_1 = 0$ nilai-nilai $m_2, m_3, \ldots, m_n$ dapat dihitung sebagai berikut:

$$m_k = 2\frac{f_k - f_{k-1}}{x_k - x_{k-1}} - m_{k-1} \text{ untuk } 1 \le k \le (n-1). \qquad (4.63)$$

Fungsi MATLAB `spline2` *untuk menghitung nilai-nilai* $m_k = S'(x_k)$ *pada spline kuadratik* $S(x)$

Implementasi persamaan (4.63) sebagai fungsi MATLAB `spline2` ditunjukkan pada Gambar 4.19. Fungsi tersebut menghitung vektor m jika diberikan masukan pasangan vektor x dan f. Fungsi MATLAB `interspline2`, yang ditunjukkan pada Gambar 4.20, berguna untuk menghitung nilai interpolasi dengan spline kuadratik (4.62).

Fungsi MATLAB `interspline2` *untuk menghitung nilai-nilai spline kuadratik alami*

CONTOH 4.15.

Carilah suatu spline kuadratik interpolan untuk data yang diberikan pada tabel berikut ini.

x	-1	0	0.5	1	2	2.5
f	2	1	0	1	2	3

```
function S=interspline2(x,f,z)
% menghitung nilai spline kuadratik alami
%       (m_{k+1}-m_k)
% S(z)= --------------(z-x_k)^2+m_k(z-x_k)+f_k
%       2(x_{k+1}-x_k)
% m_1=0; m_k = 2(f_{k}-f_{k-1})/(x_{k}-x_{k-1})
n=length(x);
m=spline2(x,f); % gunakan fungsi spline2
for j=1:length(z),
    for k=1:(n-1),
        if z(j)>=x(k) & z(j)<=x(k+1),
S(j)=(m(k+1)-m(k))/(2*(x(k+1)-x(k)))*(z(j)-x(k))^2+...
                m(k)*(z(j)-x(k))+f(k);
        end
    end
end
```

Gambar 4.20: Fungsi MATLAB `interspline2`: untuk menghitung nilai-nilai spline kuadratik interpolasi $S(z)$

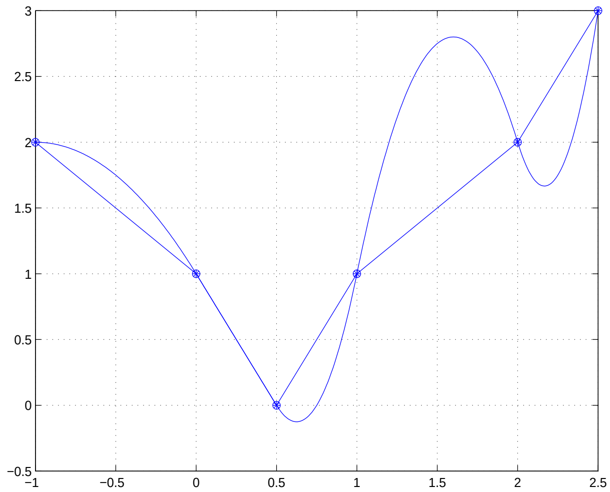
Jadi spline kuadratik yang diinginkan adalah

$$S(x) = \begin{cases} -(x+1)^2+2 & \textit{untuk} \quad -1 \le x \le 0 \\ -2x+1 & \textit{untuk} \quad 0 \le x \le 0.5 \\ 8(x-0.5)^2-2(x-0.5) & \textit{untuk} \quad 0.5 \le x \le 1 \\ -5(x-1)^2+6(x-1)+1 & \textit{untuk} \quad 1 \le x \le 2 \\ 12(x-2)2-4(x-2)+2 & \textit{untuk} \quad 2 \le x \le 2.5. \end{cases}$$

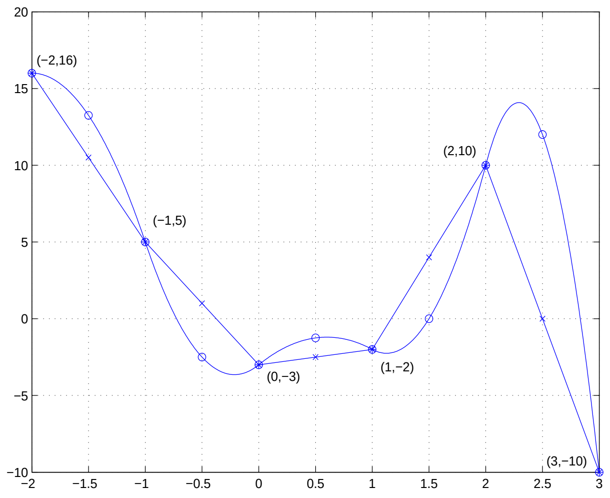
Nilai-nilai m untuk spline $S(x)$ tersebut juga dapat dihitung dengan menggunakan fungsi-fungsi MATLAB `spline2` dan `interspline2` sebagai berikut, sekaligus untuk menguji bahwa spline kuadratik yang dihasilkan benar-benar menginterpolasikan titik-titik yang diketahui.

Contoh pemakaian fungsi MATLAB `spline2` *dan* `interspline2`

```
>>x=[-1 0 .5 1 2 2.5]
 x =
    -1  0  0.5000  1  2   2.5000
>>f=[2 1 0 1 2 3]
```



Gambar 4.21: Spline linier dan kuadratik dengan menggunakan simpul-simpul $(-1,\ 2)$, $(0,\ 1)$, $(0.5,\ 0)$, $(1,\ 1)$, $(2,\ 2)$, $(2.5,\ 3)$.



Gambar 4.22: Interpolasi dengan spline linier (simbol ’*’) dan spline kuadratik (simbol ’o’) berdasarkan data titik yang sama

(Dalam hal ini, $f_k = S(x_k)$, untuk $1 \le k \le n$, adalah nilai-nilai yang sudah diketahui.)

Persyaratan $S'(x)$ harus kontinyu pada simpul-simpul interior $x_2,\ x_3,\ \ldots,\ x_{n-1}$ memberikan $(n-2)$ persamaan lain dalam bentuk

$$S'_k(x_{k+1}) = S'_{k+1}(x_{k+1}), \qquad k = 1,\ 2, ..., (n-2). \tag{4.67}$$

Demikian juga, persyaratan $S''(x)$ harus kontinyu pada simpul-simpul interior $x_2,\ x_3,\ \ldots,\ x_{n-1}$ memberikan $(n-2)$ persamaan lain, yakni

$$S''_k(x_{k+1}) = S''_{k+1}(x_{k+1}), \qquad k = 1,\ 2, ..., (n-2). \tag{4.68}$$

Jadi, dari persamaan-persamaan (4.65), (4.66), (4.67), dan (4.68) kita hanya mempunyai $(2n-4)+2+(n-2)+(n-2) = 4n-6$ persamaan untuk menentukan $4n-4$ konstanta tak diketahui. Oleh karena itu kita akan menggunakan dua buah asumsi (sebagai syarat batas) untuk mendapatkan konstanta-konstanta tersebut. Asumsi ini biasanya menyangkut nilai-nilai $S'(x)$ atau $S''(x)$ di x_1 dan x_n.

Berikut kita bahas bagaimana cara memperoleh konstanta-konstanta yang diperlukan untuk membentuk sebuah spline kubik. Oleh karena $S(x)$ sepotong-sepotong merupakan polinomial kubik pada interval $[a,\ b]$, maka turunan keduanya, $S''(x)$, sepotong-sepotong merupakan polinomial linier pada interval $[a,\ b]$. Dengan menggunakan rumus interpolasi Lagrange linier, berdasarkan simpul-simpul x_k dan x_{k+1}, untuk fungsi $S''(x) = S''_k(x)$ kita tahu, bahwa

$$S''_k(x) = S''(x_k)\frac{x - x_{k+1}}{x_k - x_{k+1}} + S''(x_{k+1})\frac{x - x_k}{x_{k+1} - x_k}, \tag{4.69}$$

untuk $x_k \le x \le x_{k+1}$ dan $1 \le k \le (n-1)$. Selanjutnya, kita misalkan $m_k = S''(x_k)$, sehingga (4.69) dapat dituliskan sebagai

$$S''(x) = S''_k(x) = \frac{m_{k+1}}{x_{k+1} - x_k}(x - x_k) + \frac{m_k}{x_{k+1} - x_k}(x_{k+1} - x), \tag{4.70}$$

untuk $x_k \le x \le x_{k+1}$ dan $1 \le k \le (n-1)$. Di sini berlaku $S''_k(x_k) = S''_{k-1}(x_k) = m_k$. Jadi $S''(x)$ kontinyu pada x_k untuk $2 \le k \le n-1$.

Dengan mengintegralkan $S''(x)$ dua kali kita peroleh

$$S_k(x) = \frac{m_{k+1}}{6(x_{k+1} - x_k)}(x - x_k)^3 + \frac{m_k}{6(x_{k+1} - x_k)}(x_{k+1} - x)^3 + c_k x + \delta_k, \tag{4.71}$$

Dengan menuliskan $h_k = x_{k+1} - x_k$ dan $d_k = (f_{k+1} - f_k)/h_k$ kita peroleh

$$\begin{aligned} S_k'(x_k) &= -\frac{m_k}{2}h_k + \frac{f_{k+1}}{h_k} - \frac{h_k}{6}m_{k+1} - \frac{f_k}{h_k} + \frac{h_k}{6}m_k \\ &= -\frac{h_k m_{k+1}}{6} - \frac{h_k m_k}{3} + d_k. \end{aligned} \tag{4.76}$$

Dengan cara yang sama kita dapatkan

$$S_{k-1}'(x_k) = \frac{h_{k-1}m_k}{3} + \frac{h_{k-1}m_{k-1}}{6} + d_{k-1}. \tag{4.77}$$

Dari (4.75), (4.76), dan (4.77) kita peroleh, untuk $2 \le i \le n-1$,

$h_k = x_{k+1} - x_k$, $d_k = (f_{k+1} - f_k)/h_k$, $u_k = 2(h_{k-1} + h_k)$, $v_k = 6(d_k - d_{k-1})$

$$h_{k-1}m_{k-1} + 2h_{k-1}m_k + 6d_{k-1} = -h_k m_{k+1} - 2h_k m_k + 6d_k$$

atau

$$h_{k-1}m_{k-1} + 2(h_{k-1} + h_k)m_k + h_k m_{k+1} = 6(d_k - d_{k-1}).$$

Dengan menuliskan $u_k = 2(h_{k-1}+h_k)$ dan $v_k = 6(d_k-d_{k-1})$, kita dapatkan rumus

SPL untuk menghitung $m_k = S''(x_k)$

$$h_{k-1}m_{k-1} + u_k m_k + h_k m_{k+1} = v_k \qquad \text{untuk} \quad 2 \le k \le n-1. \tag{4.78}$$

Sistem persamaan linier (4.78) merupakan sebuah SPL yang terdiri atas $n-2$ persamaan dalam n variabel $(m_1,\ m_2,\ ...,\ m_n)$. Oleh karena itu diperlukan dua buah persamaan atau syarat untuk mengeliminir m_1 dari persamaan pertama dan m_n dari persamaan ke-($n-2$) pada (4.78). Kedua syarat ini disebut **syarat titik-titik ujung**. Terdapat beberapa strategi untuk memberikan kedua syarat tersebut, sebagaimana disajikan pada Tabel 4.2.

Beberapa metode penentuan spline kubik

Kelima strategi untuk menentukan syarat batas (m_1 dan m_n) di atas akan menentukan persamaan pertama ($k = 2$) dan terakhir ($k = n-1$) pada (4.78), sedangkan persamaan ke-2 sampai $(n-2)$ (yakni untuk $k = 3, ..., (n-2)$) sama untuk strategi manapun yang dipilih.

Secara umum, nilai-nilai m_k, untuk $2 \le k \le n-1$, dihitung dengan cara menyelesaikan sistem persamaan $A\mathbf{m} = \mathbf{v}$, dengan A adalah suatu matriks tridiagonal berukuran $(n-2) \times (n-2)$, $\mathbf{m}$ dan $\mathbf{v}$ adalah dua buah

SPL untuk menghitung m_k merupakan sistem ***trigiagonal***

2. Untuk spline "alami" ($m_1 = m_n = 0$):

$$\begin{array}{lll} u_2^* = u_2, & h_2^* = h_2, & v_2^* = v_2, \\ u_{n-1}^* = u_{n-1}, & h_{n-2}^* = h_{n-2}, & v_{n-1}^* = v_{n-1}. \end{array}$$

Syarat batas untuk spline "alami"

3. Untuk spline "terekstrapolasi" ($m_1 = m_2 - \frac{h_1(m_3 - m_2)}{h_2}, m_n = m_{n-1} + \frac{h_{n-1}(m_{n-1} - m_{n-2})}{h_{n-2}}$):

Syarat batas untuk spline "terekstrapolasi"

$$\begin{array}{lll} u_2^* = 3h_1 + 2h_2 + \frac{h_1^2}{h_2}, & h_2^* = h_2 - \frac{h_1^2}{h_2}, & v_2^* = v_2, \\ u_{n-1}^* = 2h_{n-2} + 3h_{n-1} + \frac{h_{n-1}^2}{h_{n-2}}, & h_{n-2}^* = h_{n-2} - \frac{h_{n-1}^2}{h_{n-2}}, & v_{n-1}^* = v_{n-1}. \end{array}$$

4. Untuk spline "berujung parabolik" ($m_1 = m_2, m_n = m_{n-1}$):

Syarat batas untuk spline "berujung parabolik"

$$\begin{array}{lll} u_2^* = 3h_1 + 2h_2, & h_2^* = h_2, & v_2^* = v_2, \\ u_{n-1}^* = 2h_{n-2} + 3h_{n-1}, & h_{n-2}^* = h_{n-2}, & v_{n-1}^* = v_{n-1}. \end{array}$$

5. Untuk spline "berujung sesuai kurvatur" (m_1 dan m_n ditentukan nilainya):

Syarat batas untuk spline "berujung sesuai kurvatur"

$$\begin{array}{lll} u_2^* = u_2, & h_2^* = h_2, & v_2^* = v_2 - h_1 m_1, \\ u_{n-1}^* = u_{n-1}, & h_{n-2}^* = h_{n-2}, & v_{n-1}^* = v_{n-1} - h_{n-1} m_n. \end{array}$$

Sistem persamaan linier $A\mathbf{m} = \mathbf{v}$ di atas merupakan **sistem tridiagonal** yang bersifat **dominan secara diagonal**, karena $u_k = 2(h_{k-1} + h_k)$ (untuk $3 \le k \le (n-2)$) dan $|u_2^*| \ge |h_2^*|$, $|h_{n-2}^*| \le |u_{n-1}^*|$ untuk setiap strategi, sehingga SPL tersebut mempunyai penyelesaian tunggal. SPL tersebut dapat diselesaikan dengan menggunakan dekomposisi LU maupun eliminasi Gauss guna memperoleh nilai-nilai $m_2,\ m_3,\ \ldots,\ m_{n-1}$.

Setelah mendapatkan nilai-nilai $m_2,\ m_3,\ \ldots,\ m_{n-1}$ kita dapat menghitung nilai-nilai C_k dan D_k, dengan menggunakan rumus (4.74) dan (4.73), dan akhirnya kita peroleh polinomial-polinomial $S_k(x)$ pada interval $[x_k,\ x_{k+1}]$, dengan menggunakan rumus (4.72), untuk $1 \le k \le n-1$.

Secara ringkas langkah-langkah menghitung nilai spline kubik disajikan pada Algoritma 4.4. Perlu dicatat bahwa pada Algoritma 4.4, nilai m_1 dan m_n merupakan masukan yang ditentukan nilainya di luar algoritma. Teknik ini dipilih karena pemilihan strategi penentuan nilai-nilai m_1 dan m_n tidak mempengaruhi langkah-langkah perhitungan spline kubik.

Implementasi Algoritma 4.4 pada MATLAB ditunjukkan pada Gambar 4.23 sebagai fungsi `spline3`. Fungsi `spline3` dapat digunakan untuk menghitung nilai-nilai spline kubik, di beberapa titik (elemen-elemen vektor `z`), yang melalui titik-titik (x_k, y_k), sesuai dengan syarat-syarat batas yang diketahui. Pemakaian fungsi `spline3` memerlukan enam buah masukan, yakni vektor x, y (keduanya seukuran), konstanta atau vektor z, nilai yang menunjukkan jenis spline (1, 2, 3, 4 atau 5) dan kedua syarat batas, $b1$ dan bn.

1. Untuk jenis 1, $S'(x_1) = b1,\ S'(x_n) = bn$.
2. Untuk jenis 2, 3 dan 4, masukan $b1$ dan bn diabaikan.
3. Untuk jenis 5, $m_1 = S''(x_1) = b1,\ m_n = S''(x_n) = bn$.

Jika z berupa konstanta, maka hasilnya sebuah nilai. Jika z berupa sebuah vektor, hasilnya juga berupa sebuah vektor seukuran z. Jika $z = x$, maka hasilnya sama dengan vektor y. Selain menghitung nilai-nilai spline, pemakaian fungsi `spline3` juga akan menampilkan matriks A, dan vektor $\mathbf{V}$, $\mathbf{m}$ serta vektor C dan D.

CONTOH 4.16.

Dari tabel berikut ini, carilah spline kubik alami yang melewati titik-titik tersebut.

x	1	2	3	4	5
y	0	1	0	1	0

Penyelesaian:

Rumus-rumus yang kita miliki dalam hal ini adalah

$$h_k = x_{k+1} - x_k \quad d_k = (y_{k+1} - y_k)/h_k, \qquad k = 1,\ 2,\ 3,\ 4;$$
$$u_k = 2(h_{k-1} + h_k) \quad v_k = 6(d_k - d_{k-1}), \qquad k = 2,\ 3,\ 4, 5,$$

sehingga kita peroleh matriks A dan vektor $\mathbf{V}$:

$$A = \begin{pmatrix} 4 & 1 & 0 \\ 1 & 4 & 1 \\ 0 & 1 & 4 \end{pmatrix} \qquad \mathbf{V} = \begin{pmatrix} -12 \\ 12 \\ -12 \end{pmatrix}$$

Penyelesaikan SPL $A\mathbf{m} = \mathbf{V}$ adalah (diperoleh dengan menggunakan faktorisasi LU atau eliminasi Gauss):

$$\mathbf{m} = (-30/7,\ 36/7,\ -30/7)',$$

sehingga diperoleh

$$\begin{array}{ll} C_1 = 1 - \frac{-30/7}{6} = 12/7, & D_1 = 0; \\ C_2 = 0 - \frac{36/7}{6} = -6/7, & D_2 = 1 - \frac{-30/7}{6} = 12/7; \\ C_3 = 1 - \frac{-30/7}{6} = 12/7, & D_3 = 0 - \frac{36/7}{6} = -6/7; \\ C_4 = 0, & D_4 = 1 - \frac{-30/7}{6} = 12/7. \end{array}$$

Jadi,

$$\begin{aligned} S_1(x) &= \frac{-30/7}{6 \times 1}(x-1)^3 + 0 + \frac{12}{7}(x-1) + 0 = -\frac{5}{7}(x-1)^3 + \frac{12}{7}(x-1), \\ S_2(x) &= \frac{36/7}{6}(x-2)^3 + \frac{-30/7}{6}(3-x)^3 - \frac{6}{7}(x-2) + \frac{12}{7}(3-x), \\ &= \frac{6}{7}(x-2)^3 - \frac{5}{7}(3-x)^3 - \frac{6}{7}(x-2) + \frac{12}{7}(3-x), \\ S_3(x) &= \frac{-30/7}{6}(x-3)^3 + \frac{36/7}{6}(4-x)^3 - \frac{6}{7}(x-3) - \frac{6}{7}(4-x) \\ &= -\frac{5}{7}(x-3)^3 + \frac{6}{7}(4-x)^3 + \frac{12}{7}(x-3) - \frac{6}{7}(4-x), \\ S_4(x) &= 0 + \frac{-30/7}{6}(5-x)^3 + 0 + \frac{12}{7}(5-x) = -\frac{5}{7}(5-x)^3 + \frac{12}{7}(5-x). \end{aligned}$$

Akhirnya kita peroleh spline yang kita cari, yakni

$$S(x) = \begin{cases} -\frac{5}{7}(x-1)^3 + \frac{12}{7}(x-1) & ,\ 1 \le x \le 2 \\ \frac{6}{7}(x-2)^3 - \frac{5}{7}(3-x)^3 - \frac{6}{7}(x-2) + \frac{12}{7}(3-x) & ,\ 2 \le x \le 3 \\ -\frac{5}{7}(x-3)^3 + \frac{6}{7}(4-x)^3 + \frac{12}{7}(x-3) - \frac{6}{7}(4-x) & ,\ 3 \le x \le 4 \\ -\frac{5}{7}(5-x)^3 + \frac{12}{7}(5-x) & ,\ 4 \le x \le 5 \end{cases}$$

Soal tersebut dapat diselesaikan dengan menggunakan fungsi MATLAB `spline3` *sebagai berikut*

Contoh pemakaian fungsi MATLAB `spline3` *untuk menghasilkan lima jenis spline kubik*

```
>>format rat
>>x=0:3
>>x =
      0            1            2            3
>>y=[0 0.5 2 1.5]
y =
      0           1/2           2           3/2
>>z=0:.1:3;
>>s1=spline3(x,y,z,1,0.2,-1);
A =
     7/2           1
      1           7/2
V =
    51/10
   -21/2
m =
    -9/25
    63/25
   -93/25
     9/25
C =
     2/25       131/50        36/25
D =
     3/50          2/25       131/50
>>s2=spline3(x,y,z,2,0,0);
A =
      4            1
      1            4
V =
      6
    -12
m =
      0
    12/5
   -18/5
```

```
A =
      4            1
      1            4
V =
    63/10
  -153/10
m =
    -3/10
    27/10
    -9/2
    33/10
C =
     1/20        11/4         19/20
D =
     1/20         1/20        11/4
>>plot(z,s1,z,s2,z,s3,z,s4,z,s5);grid on
>>hold on
>>plot(x,y,'*');plot(x,y,'o');
>>legend('Spline "Terjepit"','Spline "alami"', ...
  'Spline "terekstrapolasi"',...
  'Spline "berujung parabolik"',...
  'Spline "kurvatur disesuaikan"',2)
```

Gambar 4.25 menyajikan kelima spline kubik yang diminta. Perhatikan bahwa kelima spline kubik tersebut sangat berbeda perilakunya pada interval-interval ujung kiri dan kanan, karena adanya perbedaan syarat batas pada kedua ujung interval interpolasi.

Dari vektor-vektor m, C, D pada output MATLAB di atas, Anda dapat menyusun kelima spline kubik tersebut dengan menggunakan rumus (4.72) *(Silakan Anda lakukan!).* □

Spline interpolasi harus memiliki sedikit perilaku osilasi (naik turun).

Spline kubik cocok digunakan sebagai fungsi interpolasi.

Suatu fungsi spline yang praktis untuk dipakai adalah spline yang memiliki sedikit perilaku osilasi (naik turun). Dengan kata lain, suatu spline $S(x)$ harus memiliki turunan pertama $S'(x)$ yang nilainya tidak terlalu cepat berubah di antara titik-titik simpul. Hal ini dapat diperoleh dengan menyaratkan turunan keduanya $S''(x)$ harus sekecil mungkin, atau persisnya dengan menyaratkan $\int_a^b [S''(x)]^2\, dx$ sekecil mungkin. Konsekuensinya, diantara semua fungsi $f(x)$ yang memiliki turunan kedua pada $[a,\ b]$ dan menginterpolasikan data titik-titik $\{(x_k,\ y_k)\}_{k=1}^n$, spline

Selanjutnya, dengan menggunakan integral parsial dan syarat batas, diperoleh

$$\begin{aligned}\int_a^b S''(x)[f''(x)-S''(x)]\,dx &= S''(x)[f'(x)-S'(x)]\Big|_{x=a}^{x=b} \\ &\quad -\int_a^b S'''(x)[f'(x)-S'(x)]\,dx \\ &= 0-0-\int_a^b S'''(x)[f'(x)-S'(x)]\,dx.\end{aligned}$$

Karena $S'''(x)$ konstan, $f(b)=S(b)$ dan $f(a)=S(a)$ (syarat interpolasi), maka

$$\int_a^b S'''(x)[f'(x)-S'(x)]\,dx = 0,$$

sehingga

$$\int_a^b S''(x)[f''(x)-S''(x)]\,dx = 0$$

atau

$$\int_a^b f''(x)S''(x)\,dx = \int_a^b [S''(x)]^2\,dx. \tag{4.82}$$

Dari kesamaan (4.82) dan ketaksamaan (4.81) diperoleh

$$0 \le \int_a^b [f''(x)]^2\,dx - \int_a^b [S''(x)]^2\,dx,$$

dan akhirnya akan didapatkan ketaksamaan (4.80). ■

LATIHAN 4.4

1. Diketahui data titik-titik $(0,1),\ (1,1),\ (2,5)$.

   (a) Tentukan spline linier yang menginterpolasikan titik-titik tersebut.

   (b) Tentukan spline kuadratik yang menginterpolasikan titik-titik tersebut.

5. Perhatikan polinomial kubik $S(x) = a_0 + a_1x + a_2x^2 + a_3x^3$. Tunjukkan bahwa syarat $S(1) = 1$,$S'(1) = 0$, $S(2) = 2$, dan $S'(2) = 0$ menghasilkan SPL

$$\begin{aligned} a_0 + a_1 + a_2 + a_3 &= 1 \\ a_1 + 2a_2 + 3a_3 &= 0 \\ a_0 + 2a_1 + 4a_2 + 8a_3 &= 2 \\ a_1 + 4a_2 + 12a_3 &= 0 \end{aligned}$$

   Selesaikan SPL tersebut untuk mendapatkan persamaan $S(x) = 6 - 12x + 9x^2 - 2x^3$.

6. Perhatikan polinomial kubik $S(x) = a_0 + a_1x + a_2x^2 + a_3x^3$. Tunjukkan bahwa syarat $S(1) = 3$,$S'(1) = -4$, $S(2) = 1$, dan $S'(2) = 2$ menghasilkan SPL

$$\begin{aligned} a_0 + a_1 + a_2 + a_3 &= 3 \\ a_1 + 2a_2 + 3a_3 &= -4 \\ a_0 + 2a_1 + 4a_2 + 8a_3 &= 1 \\ a_1 + 4a_2 + 12a_3 &= 2 \end{aligned}$$

   Selesaikan SPL tersebut untuk mendapatkan persamaan $S(x) = 5 + 2x - 6x^2 + 2x^3$.

7. Tunjukkan bahwa fungsi yang didefinisikan sebagai berikut merupakan spline kubik dengan cara menunjukkan bahwa $S_1(2) = S_2(2)$, $S_1'(2) = S_2'(2)$ dan $S_1''(2) = S_2''(2)$.

$$f(x) = \begin{cases} S_1(x) = -\frac{13}{4}x^3 + 15x^2 - \frac{81}{4}x - \frac{19}{2} & \text{untuk} \quad 1 \le x \le 2 \\ S_2(x) = \frac{11}{4}x^3 - 21x^2 + \frac{207}{4}x - \frac{77}{2} & \text{untuk} \quad 2 \le x \le 3 \end{cases}$$

   Gambar grafik fungsi spline tersebut.

8. Tunjukkan bahwa fungsi yang didefinisikan sebagai berikut merupakan spline kubik dengan cara menunjukkan bahwa $S_1(2) = S_2(2)$,

12. Bentuklah kelima jenis spline kubik $S(x)$ yang melalui titik-titik di bawah ini. Untuk spline "terjepit" dan spline "kurvatur disesuaikan" gunakan syarat-syarat batas yang diberikan.

    (a) $(0,1),\ (1,4),\ (2,0),\ (3,-2)$ dengan $S'(0) = 2,\ S'(3) = 2$ dan $''(0) = -1.5,\ S''(3) = 3$.

    (b) $(0,5),\ (1,2),\ (2,1),\ (3,3),\ (4,1)$ dengan $S'(0) = -2,\ S'(4) = -1$ dan $S''(0) = -0.5,\ S''(4) = -1.9$.

    (c) $(0,2),\ (1,3),\ (2,2),\ (3,2),\ (4,1)$ dengan $S'(0) = 1.6,\ S'(4) = -1$ dan $S''(0) = -1.4,\ S''(4) = -1.4$.

    (d) $(1/2, f(1/2)),\ (1, f(1)),\ (3/2, f(3/2)),\ (2, f(2))$, dengan $f(x) = x + 2/x$, $S'(1/2) = -7,\ S'(2) = 1/2$ dan $S''(1/2) = 32,\ S''(2) = 1/2$.

    (e) $(0,1),\ (1,0),\ (2,0),\ (3,1),\ (4,2),\ (5,2),\ (6,1)$ dengan $S'(0) = -0.6,\ S'(6) = -1.8$ dan $S''(0) = 1,\ S''(6) = -1$.

    (f) $(0,0),\ (1,4),\ (2,8),\ (3,9),\ (4,9),\ (5,8),\ (6,6)$ dengan $S'(0) = 1,\ S'(6) = -2$ dan $S''(0) = 1,\ S''(6) = -1$.

    (g) $(0,0),\ (1,0),\ (2,3),\ (3,2),\ (4,2),\ (5,1),\ (6,0)$ dengan $S'(0) = 1.5,\ S'(6) = -0.3$ dan $S''(0) = -1,\ S''(6) = 1$.

13. Apakah fungsi di bawah ini merupakan spline kubik pada interval $0 \le x \le 2$?

$$S(x) = \begin{cases} (x-1)^3, & 0 \le x \le 1 \\ 2(x-1)^3, & 1 \le x \le 2 \end{cases}$$

14. Apakah fungsi di bawah ini merupakan spline kubik alami pada interval $1 \le x \le 3$?

$$S(x) = \begin{cases} x^3 - 3x^2 + 2x + 1, & 1 \le x \le 2 \\ -x^3 + 9x^2 - 22x + 17, & 2 \le x \le 3 \end{cases}$$

15. Diketahui interval $[a,b]$ dan misalkan $a < c < b$.

    (a) Tunjukkan bahwa fungsi

$$\sigma_c = \begin{cases} 0, & a \le x \le c \\ (x-c)^3, & c \le x \le b \end{cases}$$

    merupakan spline kubik pada $[a,b]$.

(melalui) sekumpulan titik adalah identik. Sebuah polinomial berderajad n dapat dibentuk secara tepat dengan menggunakan data $(n+1)$ titik dan dapat dinyatakan sebagai

Melalui $(n+1)$ titik berlainan dapat dibentuk sebuah polinomial yang berderajad paling tinggi n.

$$P_n(x) = a_1x^n + a_2x^{n-1} + ... + a_nx + a_{n+1}. \tag{4.83}$$

Jika diketahui data $(n+1)$ titik (x_1, y_1), (x_2, y_2), ..., (x_{n+1}, y_{n+1}), maka koefisien-koefisien a_k, $1 \leq k \leq (n+1)$ dapat dihitung dengan menyelesaikan SPL $M\mathbf{a} = \mathbf{y}$:

Pengertian matriks Vandermonde

$$\begin{pmatrix} x_1^n & x_1^{n-1} & ... & x_1 & 1 \\ x_2^n & x_2^{n-1} & ... & x_2 & 1 \\ x_3^n & x_3^{n-1} & ... & x_3 & 1 \\ \vdots & \vdots & \ddots & \vdots & \\ x_n^n & x_n^{n-1} & ... & x_n & 1 \\ x_{n+1}^n & x_{n+1}^{n-1} & ... & x_{n+1} & 1 \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \\ a_3 \\ \vdots \\ a_n \\ a_{n+1} \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_n \\ y_{n+1} \end{pmatrix}. \tag{4.84}$$

Matriks M disebut matriks *Vandermonde* dari vektor $\mathbf{x}$. Pada MATLAB matriks Vandermonde dapat dihasilkan dengan fungsi `vander(x)` jika diketahui vektor `x`. Jadi, jika diketahui pasangan vektor $\mathbf{x}$ dan $\mathbf{y}$, maka koefisien-koefisien polinomial (4.83) dapat dihitung dengan mudah sebagai berikut

`vander(x)`

```
>>M=vander(x);
>>a=M\y;
```

Selanjutnya, nilai-nilai polinomial (4.83) dapat dihitung dengan menggunakan fungsi `polyval(a,z)`, dengan `z` dapat berupa konstanta maupun vektor. Hasilnya adalah $P_n(z)$.

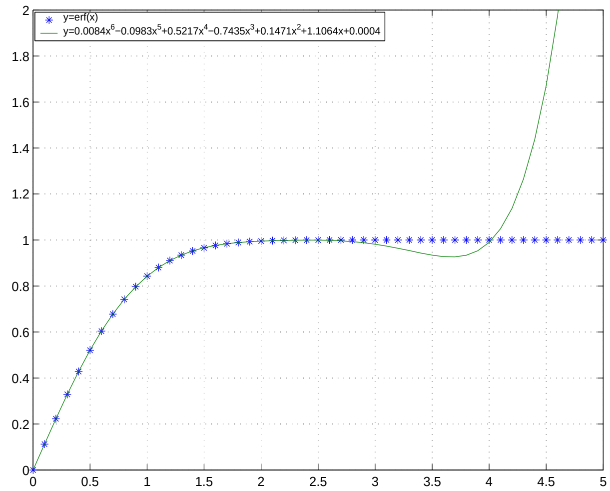
`polyval(a,z)`

Polinomial Lagrange juga dapat dihasilkan dengan fungsi MATLAB `polyfit(x,y,n)`, dengan `n` menyatakan derajad polinomial yang melalui titik-titik (x_k, y_k). Kedua perintah MATLAB di atas ekivalen dengan perintah-perintah MATLAB sebagai berikut:

`polyfit(x,y,n)`

```
>>n=length(x)-1;
>>a=polyfit(x,y,n);
```

CONTOH 4.18.
Dalam contoh ini akan ditunjukkan interpolasi terhadap fungsi galat $f(x) = erf(x) = \frac{2}{\sqrt{\pi}}\int_0^x e^{-t^2}dt$ dengan menggunakan polinomial berderajad 6. Mula-



Gambar 4.26: Polinomial interpolasi berderajad 6 untuk fungsi galat $erf(x) = \frac{2}{\sqrt{\pi}}\int_0^x e^{-t^2}dt$

Sintaks `fz = interp1(x,Y,z)` menghasilkan vektor `fz` yang terdiri atas nilai-nilai interpolasi dengan spline linier yang bersesuaian dengan `z` dan ditentukan berdasarkan pasangan vektor `x` dan `y`. Vektor `x` memuat titik-titik di mana data `y` diberikan. Jika `y` berupa matriks, maka interpolasi dilakukan untuk setiap kolom `y` dan `fz` merupakan matriks berukuran `length(z)`×`size(Y,2)`.

Sintaks `fz = interp1(y,z)` menggunakan vektor `x = 1:n`, dengan `n=length(y)` jika `y` berupa vektor atau `n=size(y,1)` jika `y` berupa matriks. Sintaks `fz = interp1(x,y,z,metode)` menggunakan spline berdasarkan `metode`. Opsion (parameter) `metode` adalah string yang dapat berupa:

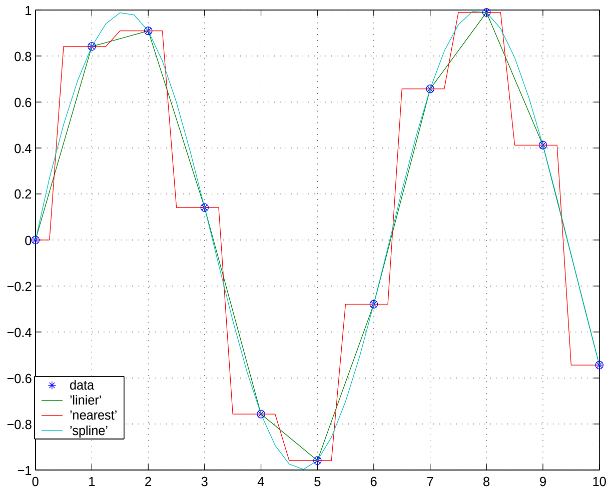
'nearest' untuk interpolasi dengan metode **lingkungan terdekat**.

'linear' untuk interpolasi linier (aslinya menggunakan ini).

'spline' untuk interpolasi dengan spline kubik.

'pchip' untuk interpolasi Hermit kubik sepotong-sepotong.

'cubic' sama dengan 'pchip'.



Gambar 4.27: Interpolasi nilai-nilai $\sin(x)$ dengan tiga metode interpolasi. Terlihat bahwa interpolasi dengan metode **lingkungan terdekat** menghasilkan nilai konstan di sekitar titik data.

```
ans =
    214.8585
```

Perintah-perintah MATLAB berikut ini menghasilkan kurva pertumbuhan penduduk USA dari tahun 1900 sampai 2002 dengan menggunakan spline kubik berdasarkan data sensus tahun 1900 sampai 1990 di atas. Hasilnya ditunjukkan pada Gambar 4.28. □

```
>>x = 1900:1:2002;
>>y = interp1(t,p,x,'spline');
>>plot(t,p,'o',x,y);grid on
```

MATLAB juga menyediakan fungsi khusus untuk menghasilkan interpolasi dengan spline kubik, yakni fungsi `spline`, yang dapat dipakai dengan cara sebagai berikut. `spline`

```
fz = spline(x,y,z)
a = spline(x,y)
```

CONTOH 4.21.
Perintah-perintah MATLAB berikut menghasilkan nilai-nilai interpolasi $\sin(x)$ *pada interval* $[0, 10]$ *dengan menggunakan spline kubik berdasarkan data nilai-nilai* $\sin(x)$ *di* $x = 0, 1, 2, ..., 10$. □

```
>>x = 0:10;
>>y = sin(x);
>>z = 0:.25:10;
>>fz = spline(x,y,z);
>>plot(x,y,'o',z,fz)
```

CONTOH 4.22.
Perintah-perintah MATLAB berikut menghasilkan nilai-nilai interpolasi dengan spline kubik "terjepit" dengan gradien di kedua ujung nol. Hasilnya terlihat pada Gambar 4.29. □

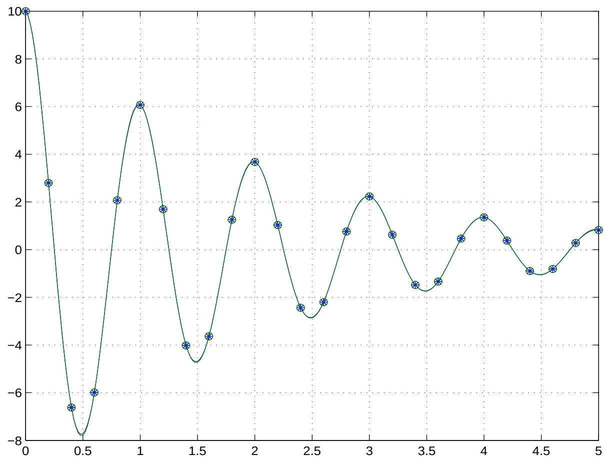
```
>>x = -4:4;
>>y = [0 .15 1.12 2.36 2.36 1.46 .49 .06 0];
>>cs = spline(x,[0 y 0]);
>>z = linspace(-4,4,101);
>>plot(x,y,'o',z,ppval(cs,z),'-');
```

CONTOH 4.23.
Perintah-perintah MATLAB di bawah ini menghasilkan kurva sebuah lingkaran berpusat di $O(0, 0)$ *dan berjari-jari 1 beserta lima buah titik (dua titik berimpit)* $(1, 0)$, $(0, 1)$, $(-1, 0)$, $(0, -1)$, *dan* $(1, 0)$, *seperti terlihat pada Gambar 4.30.*

```
>>x = pi*[0:.5:2];
>>y = [0  1  0 -1  0  1  0;
       1  0  1  0 -1  0  1];
>>a = spline(x,y);
>>fz = ppval(a, linspace(0,2*pi,101));
>>plot(fz(1,:),fz(2,:),'-b',y(1,2:5),y(2,2:5),'or'),
  axis equal
```

Perhatikan, bahwa vektor `y` *memuat dua lebih banyak kolom daripada cacah elemen* `x`, *sehingga nilai-nilai* `y(:,1)` *dan* `y(:,end)` *digunakan sebagai gradien-*

gradien ujung. Perintah MATLAB `linspace` *menghasilkan sejumlah nilai berjarak sama antara dua nilai yang diketahui. Fungsi MATLAB* `ppval` *menghitung nilai-nilai spline kubik, dengan vektor koefisien diketahui, pada nilai-nilai absis yang diberikan.* □



Gambar 4.31: Spline interpolasi kubik untuk fungsi getaran teredam $f(x) = 10e^{-0.5x}\cos(2\pi x)$

CONTOH 4.24 (Perbandingan spline interpolasi dan fungsi eksak).
Diketahui fungsi getaran teredam

$$y(x) = 10e^{-0.5x}\cos 2\pi x.$$

Mula-mula dihasilkan 26 titik sampel `x=[0:.2:5]`*, kemudian dibentuk spline kubik dengan menggunakan titik-titik tersebut. Kurva spline interpolasi dan fungsi aslinya diplot pada sistem koordinat yang sama. Plot pula kurva selisih nilai-nilai spline interpolasi dan fungsi aslinya. Berikut adalah perintah-perintah MATLAB yang melakukan hal-hal tersebut. Hasilnya ditunjukkan pada Gambar 4.31. Walaupun kedua kurva tampak berimpit, namun sebenarnya terdapat perbedaan, seperti yang diperlihatkan oleh kurva galat pada Gambar 4.32.* □

2. Selesaikan soal-soal interpolasi pada LATIHAN 4.4 (nomor 1, 2, 3, 5, 6, dan 12) dengan menggunakan fungsi-fungsi MATLAB `polyfit`, `polyval`, `interp1`, `spline`, dan `ppval`. Bandingkan hasilnya dengan penyelesaian sebelumnya. (Jika perlu plot kurva selisih kedua polinomial atau spline dengan fungsi eksaknya, jika diketahui.)

4.5 Rangkuman

Misalkan kita mempunyai data yang disajikan dalam tabel seperti di bawah ini:

x	x_1	x_2	x_3	$\ldots$	x_n
y	y_1	y_2	y_3	$\ldots$	y_n

dengan $x_1 < x_2 < ... < x_n$. Polinomial $P(x)$ yang memenuhi $P(x_k) = y_k$ untuk $1 \leq k \leq n$ dikatakan menginterpolasikan nilai-nilai pada tabel. Interpolasi adalah menghitung nilai-nilai $P(x)$ untuk sebarang nilai x, yang tidak terdapat di dalam tabel tetapi memenuhi $x_1 < x < x_n$. Titik-titik x_k disebut **simpul**. Jika nilai-nilai y merupakan nilai-nilai fungsi $f(x)$ untuk nilai-nilai x yang bersesuaian, maka polinomial $P(x)$ merupakan hampiran fungsi $f(x)$ pada interval $[x_1,\ x_n]$.

Polinomial Newton Polinomial-polinomial Newton (atau lengkapnya *Gregory – Newton*), yang menginterpolasikan sebuah tabel $(n+1)$ pasang nilai $(x,\ y)$, berbentuk

$$\begin{aligned} P_n(x) &= P_{n-1}(x) + a_{n+1}(x-x_1)(x-x_2)\ldots(x-x_n) \\ &= a_1 + \sum_{k=2}^{n+1} a_k \prod_{j=1}^{k-1}(x-x_j), \end{aligned}$$

dengan

$$\begin{aligned} a_1 &= P_0(x) = y_1, \\ a_k &= \frac{y_k - P_{k-2}(x_k)}{(x_k-x_1)(x_k-x_2)\ldots(x_k-x_{k-1})}, \qquad k = 2,\ 3,\ ...,\ (n+1). \end{aligned}$$

Perkalian tersarang Perhitungan polinomial Newton $P_n(x)$ dapat di-

f relatif terhadap simpul-simpul x_1, x_2, ..., x_{n+1} dapat dinyatakan sebagai

$$f[x_1, x_2, ..., x_{n+1}] = \sum_{k=1}^{n+1} \frac{y_k}{\prod_{\substack{j=1 \\ j \neq k}}^{n+1} (x_k - x_j)}.$$

3. Misalkan $n \geq 1$, dan turunan tingkat ke-n fungsi $f(x)$ ada dan kontinyu pada interval $[a, b]$, dan misalkan $x_1, x_2, \ldots, x_{n+1}$ adalah $(n+1)$ bilangan yang berlainan pada interval $[a, b]$. Maka

$$f[x_1, x_2, ..., x_{n+1}] = \frac{1}{n} f^{(n)}(c)$$

untuk suatu bilangan c di antara nilai minimum dan maksimum x_1, $x_2, \ldots, x_{n+1}$.

4. Selisih terbagi Newton bersifat simetris. Jika $(i_1, i_2, ..., i_n)$ menyatakan permutasi (susunan urutan) indeks $(1, 2, ..., n)$, maka

$$f[x_{i_1}, x_{i_2}, ..., x_{i_n}] = f[x_1, x_2, ..., x_n].$$

5. Jika didefinisikan

$$f[x_1, x_1] = \lim_{x_2 \to x_1} f[x_1, x_2] = \lim_{x_2 \to x_1} \frac{f(x_2) - f(x_1)}{x_2 - x_1} = f'(x_1),$$

maka dapat didefinisikan

$$f[\underbrace{x_1, x_1, ..., x_1}_{n \text{ elemen}}] = \frac{1}{n!} f^{(n)}(x_1).$$

6. Jika $f(x)$ adalah polinomial berderajad n, diketahui $m \leq n$ simpul berlainan x_1, x_2, ..., x_m, dan misalkan $x \neq x_j$, $1 \leq j \leq m$, maka fungsi $g_m(x)$ yang didefinisikan sebagai

$$g_m(x) = f[x_1, x_2, ..., x_m]$$

merupakan polinomial berderajad $n - m$.

7. Misalkan x_1 dan $h > 0$ diketahui dan $x_j = x_1 + hj$ untuk $j = 0, \pm 1, \pm 2, ...$, dan misalkan $f_j = f(x_j)$, dan definisikan **selisih maju**

$k = 1,\ 2,\ ...,\ (n-1)$, dengan

$$S_k(x) = \frac{m_{k+1} - m_k}{2(x_{k+1} - x_k)}(x - x_k)^2 + m_k(x - x_k) + f_k,$$

dengan $m_j = S'(x_j)$, untuk $j = 1,\ 2,\ ..., n$, yang nilai-nilainya dapat diperoleh dengan menyelesaikan SPL

$$m_k + m_{k+1} = 2\frac{f_{k+1} - f_k}{x_{k+1} - x_k}, \qquad k = 1,\ 2, ..., (n-1).$$

SPL tersebut terdiri atas $(n-1)$ persamaan dalam n variabel, sehingga dapat diselesaikan jika salah satu nilai m_k diketahui. Jika diberikan **syarat batas** $m_1 = 0$ atau $m_n = 0$, spline yang dihasilkan disebut **spline kuadratik alami**.

Spline Kubik Spline kubik $S(x)$ yang menginterpolasikan titik-titik $(x_1,\ f_1),\ (x_2,\ f_2),\ \ldots,\ (x_n,\ f_n)$ dengan $x_1 < x_2 < ... < x_n$ dapat dinyatakan sebagai $S(x) = S_k(x)$, untuk $x_k \leq x \leq x_{k+1}$, $k = 1,\ 2,\ ...,\ (n-1)$, dengan

$$S_k(x) = \frac{m_{k+1}(x - x_k)^3}{6(x_{k+1} - x_k)} + \frac{m_k(x_{k+1} - x)^3}{6(x_{k+1} - x_k)} + C_k(x - x_k) + D_k(x_{k+1} - x),$$

$$C_k = \frac{f_{k+1}}{x_{k+1} - x_k} - \frac{x_{k+1} - x_k}{6}m_{k+1},$$

$$D_k = \frac{f_k}{x_{k+1} - x_k} - \frac{x_{k+1} - x_k}{6}m_k.$$

Nilai-nilai m_k, untuk $2 \leq k \leq n-1$, dihitung dengan cara menyelesaikan sistem persamaan $A\mathbf{m} = \mathbf{v}$:,

$$\begin{pmatrix} u_2^* & h_2^* & 0 & 0 & \ldots & 0 \\ h_2 & u_3 & h_3 & 0 & \ldots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \ldots & h_{n-3} & u_{n-2} & h_{n-2} \\ 0 & 0 & \ldots & 0 & h_{n-2}^* & u_{n-1}^* \end{pmatrix} \begin{pmatrix} m_2 \\ m_3 \\ \vdots \\ m_{n-2} \\ m_{n-1} \end{pmatrix} = \begin{pmatrix} v_2^* \\ v_3 \\ \vdots \\ v_{n-2} \\ v_{n-1}^* \end{pmatrix},$$

dengan $h_k = x_{k+1} - x_k$, $d_k = (f_{k+1} - f_k)/h_k$ $(k = 1,\ 2, ..., (n-1))$

dan $u_k = 2(h_{k-1} + h_k)$, $v_k = 6(d_k - d_{k-1})$ $(k = 2, 3, ..., (n-1)$. Nilai-nilai m_1 dan m_n serta elemen-elemen u_2^*, h_2^*, h_{n-2}^*, u_{n-1}^*, v_2^*, dan v_{n-1}^* ditentukan secara berbeda sesuai dengan strategi yang digunakan, yakni:

1. Untuk spline "terjepit" ($m_1 = \frac{3}{h_1}[d_1 - S'(x_1)] - \frac{m_2}{2}$, $m_n = \frac{3}{h_{n-1}}[S'(x_n) - d_{n-1}] - \frac{m_{n-1}}{2}$), dengan $S'(x_1)$ dan $S'(x_n)$ diketahui:

$$\begin{array}{lll} u_2^* = \frac{3}{2}h_1 + 2h_2, & h_2^* = h_2, & v_2^* = v_2 - 3[d_1 - S'(x_1)], \\ u_{n-1}^* = 2h_{n-2} + \frac{3}{2}h_{n-1}, & h_{n-2}^* = h_{n-2}, & v_{n-1}^* = v_{n-1} - 3[S'(x_n) - d_{n-1}]. \end{array}$$

2. Untuk spline "alami" ($m_1 = m_n = 0$):

$$\begin{array}{lll} u_2^* = u_2, & h_2^* = h_2, & v_2^* = v_2, \\ u_{n-1}^* = u_{n-1}, & h_{n-2}^* = h_{n-2}, & v_{n-1}^* = v_{n-1}. \end{array}$$

3. Untuk spline "terekstrapolasi" ($m_1 = m_2 - \frac{h_1(m_3 - m_2)}{h_2}$, $m_n = m_{n-1} + \frac{h_{n-1}(m_{n-1} - m_{n-2})}{h_{n-2}}$):

$$\begin{array}{lll} u_2^* = 3h_1 + 2h_2 + \frac{h_1^2}{h_2}, & h_2^* = h_2 - \frac{h_1^2}{h_2}, & v_2^* = v_2, \\ u_{n-1}^* = 2h_{n-2} + 3h_{n-1} + \frac{h_{n-1}^2}{h_{n-2}}, & h_{n-2}^* = h_{n-2} - \frac{h_{n-1}^2}{h_{n-2}}, & v_{n-1}^* = v_{n-1}. \end{array}$$

4. Untuk spline "berujung parabolik" ($m_1 = m_2, m_n = m_{n-1}$):

$$\begin{array}{lll} u_2^* = 3h_1 + 2h_2, & h_2^* = h_2, & v_2^* = v_2, \\ u_{n-1}^* = 2h_{n-2} + 3h_{n-1}, & h_{n-2}^* = h_{n-2}, & v_{n-1}^* = v_{n-1}. \end{array}$$

5. Untuk spline "berujung sesuai kurvatur" (m_1 dan m_n ditentukan nilainya):

$$\begin{array}{lll} u_2^* = u_2, & h_2^* = h_2, & v_2^* = v_2 - h_1 m_1, \\ u_{n-1}^* = u_{n-1}, & h_{n-2}^* = h_{n-2}, & v_{n-1}^* = v_{n-1} - h_{n-1} m_n. \end{array}$$

Fungsi-fungsi MATLAB di bawah ini berguna untuk melakukan interpolasi dengan menggunakan data sekumpulan titik.

`vander` menghasilkan matriks *Vandermonde*

`polyval` menghitung nilai suatu polinomial