

Daftar Isi

Kata Pengantar	iii
Daftar Isi	v
Daftar Padan Kata	xi

1. Metode Numerik Secara Umum	1
--	----------

1.1 Metode Analitik versus Metode Numerik	4
1.2 Metode Numerik dalam Bidang Rekayasa	6
1.3 Apakah Metode Numerik Hanya untuk Persoalan Matematika yang Rumit Saja ?	7
1.4 Peranan Komputer dalam Metode Numerik	8
1.5 Mengapa Kita Harus Mempelajari Metode Numerik ?	9
1.6 Tahap-tahap Memecahkan Persoalan Secara Numerik	10
1.7 Peran Ahli Informatika dalam Metode Numerik	11
1.8 Perbedaan Metode Numerik dengan Analisis Numerik	11
1.9 Materi Apa yang Terdapat di dalam Buku Ini ?	12

2. Deret Taylor Dan Analisis Galat	15
---	-----------

2.1 Deret Taylor	16
2.2 Analisis Galat	21
2.3 Sumber Utama Galat Numerik	23
2.3.1 Galat Pemotongan	23
2.3.2 Galat Pembulatan	26
2.3.3 Galat Total	27
2.4 Orde Penghampiran	28

2.5	Bilangan Titik-Kambang	29
2.5.1	Bilangan Titik-Kambang Ternormalisasi	30
2.5.2	Epsilon Mesin	32
2.5.3	Pembulatan pada Bilangan Titik-Kambang	35
2.5.4	Aritmetika Bilangan Titik-Kambang	38
2.5.4.1	Operasi Penambahan dan Pengurangan	38
2.5.4.2	Operasi Perkalian dan Pembagian	44
2.6	Perambatan Galat	45
2.7	Kondisi Buruk	46
2.8	Bilangan Kondisi	49

3. Solusi Persamaan Nirlanjar 55

3.1	Rumusan Masalah	57
3.2	Metode Pencarian Akar	57
3.3	Meode Tertutup	58
3.3.1	Metode Bagidua	61
3.3.2	Metode Regula Falsi	67
3.4	Metode Terbuka	72
3.4.1	Metode Lelaran Titik-Tetap	72
3.4.2	Metode Newton-Raphson	84
3.4.3	Orde Konvergensi Metode Terbuka	91
3.4.4	Metode Secant	93
3.5	Akar Ganda	97
3.6	Akar-akar Polinom	100
3.6.1	Metode Horner untuk Evaluasi Polinom	100
3.6.2	Pencarian Akar-akar Polinom	101
3.6.3	Lokasi Akar Polinom	106
3.7	Sistem Persamaan Nirlanjar	107
3.7.1	Metode Lelaran Titik-Tetap	107
3.7.2	Metode Newton-Raphson	110
3.8	Contoh Soal Terapan	112

4. Solusi Sistem Persamaan Lanjar 117

4.1	Bentuk Umum Sistem Persamaan Lanjar	119
4.2	Metode Eliminasi Gauss	120
4.2.1	Tata-ancang <i>Pivoting</i>	125
4.2.2	Penskalaan	131
4.2.3	Kemungkinan Solusi SPL	132
4.3	Metode Eleminasi Gauss-Jordan	135
4.4	Metode Matriks Balikan	139
4.5	Metode Dekomposisi LU	140
4.5.1	Pemfaktoran dengan Metode LU Gauss	142
4.5.2	Metode Reduksi Crout	147
4.6	Determinan	150
4.7	Kondisi Buruk	154
4.8	Bilangan Kondisi Matriks	160
4.9	Metode Lelaran Untuk Menyelesaikan SPL	165
4.9.1	Metode Lelaran Jacobi	166
4.9.2	Metode Lelaran Gauss-Seidel	167
4.10	Contoh Soal Terapan	171

5 Interpolasi dan Regresi 179

5.1	Persoalan Interpolasi Polinom	182
5.1.1	Interpolasi Lanjar	183
5.1.2	Interpolasi Kuadratik	185
5.1.3	Interpolasi Kubik	186
5.2	Polinom Lagrange	188
5.3	Polinom Newton	193
5.4	Keunikan Polinom Interpolasi	201
5.5	Galat Intepolasi Polinom	202
5.5.1	Batas Antara Galat Interpolasi Untuk Titik-titik yang Berjarak Sama	207
5.5.2	Taksiran Galat Interpolasi Newton	211
5.5.3	Taksiran Galat Interpolasi Lagrange	212

5.6	Polinom Newton-Gregory	212
5.6.1	Polinom Newton-Gregory Maju.....	213
5.6.1.1	Tabel Sesilih Maju	213
5.6.1.2	Penurunan Rumus Polinom Newton-Gregory Maju.....	214
5.6.1.3	Menghitung Batas Galat Interpolasi Polinom Newton-Gregory Maju	218
5.6.1.4	Taksiran Galat Interpolasi Newton-Gregory Maju	220
5.6.1.5	Manfaat Tabel Selisih Maju	221
5.6.2	Polinom Interpolasi Newton-Gregory Mundur	224
5.7	Ekstrapolasi.....	226
5.8	Interpolasi Dwimatra	226
5.9	Contoh Soal Terapan Interpolasi	229
5.10	Regresi	231
5.10.1	Regresi Lanjar	233
5.10.2	Pelanjaran	236
5.11	Contoh Penerapan Regresi dalam Bidang Rekayasa	241

6 Integrasi Numerik 249

6.1	Terapan Integral dalam Bidang Sains dan Rekayasa	251
6.2	Persoalan Integrasi Numerik	253
6.3	Metode Pias	253
6.3.1	Kaidah Segiempat	254
6.3.2	Kaidah Trapesium	257
6.3.3	Kaidah Titik Tengah	258
6.3.4	Galat Metode Pias	260
6.3.4.1	Galat Kaidah Trapesium	261
6.3.4.2	Galat Kaidah Titik Tengah	264
6.4	Metode Newton-Cotes	265
6.4.1	Kaidah Trapesium	266
6.4.2	Kaidah Simpson 1/3	268
6.4.3	Kaidah Simpson 3/8	276
6.4.4	Metode Integrasi Numerik Untuk h yang Berbeda-beda.....	279
6.4.5	Bentuk Umum Metode Newton-Cotes.....	280
6.5	Singularitas	282
6.6	Penggunaan Ekstrapolasi untuk Integrasi	286
6.6.1	Ekstrapolasi Richardson	286
6.6.2	Metode Romberg	291

6.6.3	Ekstrapolasi Aitken	296
6.7	Integral Ganda	300
6.8	Kuadratus Gauss	303
6.9	Contoh Soal Terapan	311

7 Turunan Numerik **317**

7.1	Persoalan Turunan Numerik	318
7.2	Tiga Pendekatan dalam Menghitung Turunan Numerik	318
7.3	Penurunan Rumus Turunan dengan Deret Taylor	320
7.4	Penurunan Rumus Turunan Numerik dengan Polinom Interpolasi ...	323
7.5	Menentukan Orde Galat	325
7.6	Program Menghitung Turunan	326
7.7	Ringkasan Rumus-rumus Turunan	326
7.8	Contoh Perhitungan Turunan	328
7.9	Ekstrapolasi Richardson	329
7.10	Terapan Turunan Numerik dalam Bidang Pengolahan Citra	332

8 Solusi Persamaan Diferensial Biasa **341**

8.1	Kelompok Persamaan Diferensial	342
8.2	Terapan Persamaan Diferensial	343
8.3	PDB Orde Satu	345
8.4	Metode Euler	346
8.4.1	Tafsiran Geometri Metode PDB	347
8.4.2	Analisis Galat Metode Euler	349
8.5	Metode Heun (Perbaikan Metode Euler)	352
8.5.1	Tafsiran Geometri Metode Heun	353
8.5.2	Galat Metode Heun	353
8.5.3	Perluasan Metode Heun	357
8.6	Metode Deret Taylor	358
8.7	Orde Metode PDB	361
8.8	Metode Runge-Kutta	364
8.8.1	Metode Runge-Kutta Orde Satu	365

8.8.2	Metode Runge-Kutta Orde Dua	365
8.8.3	Metode Runge-Kutta Orde Tiga	368
8.8.4	Metode Runge-Kutta Orde Empat	369
8.9	Ekstrapolasi Richardson	371
8.10	Metode Banyak-Langkah	371
8.10.1	Metode Adams-Bashforth-Moulton	372
8.10.2	Metode Milne-Simpson	376
8.10.3	Metode Hamming	376
8.10.4	Prosedur Pendahuluan	376
8.10.5	Keidealan Metode <i>Predictor-Corrector</i>	377
8.11	Pemilihan Ukuran Langkah yang Optimal	379
8.12	Sistem Persamaan Diferensial	381
8.13	Persamaan Diferensial Orde Lanjut	384
8.14	Ketidakstabilan Metode PDB	387
8.15	Contoh Soal Terapan	389
	Daftar Pustaka	397

Daftar Padan

Bahasa Inggris – Bahasa Indonesia

Inggris	Indonesia
<i>approximate</i> (v)	menghampiri
<i>approximation</i> (n)	hampiran
<i>accurate</i> (adj)	teliti
<i>array</i> (n)	larik
<i>continuous</i> (adj)	menerus, sinambung
<i>continuous function</i> (n)	fungsi menerus
<i>derivative</i> (n)	turunan
<i>discrete</i> (adj)	farik
<i>dimension</i> (n)	matra
<i>design (art)</i> (v)	meripta
<i>domain</i> (n)	ranah
<i>effective</i> (adj)	sangkil
<i>efficient</i> (adj)	mangkus
<i>efficiency</i> (n)	kemangkusian
<i>engineering</i>	rekayasa
<i>engineer</i>	rekayasawan
<i>error</i> (n)	galat
<i>estimate</i> (v)	menaksir, pemperkirakan
<i>estimation</i> (n)	taksiran
<i>exact</i> (adj)	sejati
<i>exact solution</i> (n)	solusi sejati
<i>floating-point numbers</i> (n)	bilangan titik-kembang
<i>generalize</i> (v)	merampaikan
<i>generalization</i> (n)	rampatan
<i>iteration</i> (n)	lelaran
<i>interval</i> (n)	selang
<i>subinterval</i> (n)	upaselang
<i>linier</i> (adj)	lanjar
<i>loop</i> (n)	kalang
<i>manipulate</i> (v)	mengutak-atik
<i>multistep</i> (n)	bahu-langkah

<i>non linier</i> (adj)	nirlanjar
<i>problem</i> (n)	soal, persoalan, masalah
<i>plot</i> (v)	merajah
<i>round-off</i> (n)	pembualatan
<i>rule</i> (n)	aturan
<i>single-precision numbers</i> (n)	bilangan berketelitian tunggal
<i>double-precision numbers</i> (n)	bilangan berketelitian ganda
<i>strategy</i> (n)	tata-ancang
<i>substitute</i> (v)	menyulih
<i>substitution</i> (n)	sulihan
<i>significant figure (digit)</i> (n)	angka bena, angka berarti
<i>subscript</i> (n)	tikalas
<i>stagnant</i> (adj)	mandek
<i>step</i> (n)	langkah
<i>strip</i> (n)	pias
<i>truncation</i> (n)	pemotongan

Bab 1

Metode Numerik Secara Umum

Pengetahuan dimulai dari rasa ingin tahu, kepastian dimulai dengan rasa ragu-ragu, dan filsafat dimulai dengan kedua-duanya.
(Jujun S. Suriasumantri)

Rasa ingin tahu adalah ibu dari semua ilmu pengetahuan
(Anonim)

Persoalan yang melibatkan model matematika banyak muncul dalam berbagai disiplin ilmu pengetahuan, seperti dalam bidang fisika, kimia, ekonomi, atau pada persoalan rekayasa (*engineering*), seperti Teknik Sipil, Teknik Mesin, Elektro, dan sebagainya. Seringkali model matematika tersebut muncul dalam bentuk yang tidak ideal alias rumit. Model matematika yang rumit ini adakalanya tidak dapat diselesaikan dengan metode analitik yang sudah umum untuk mendapatkan **solusi sejatinya (*exact solution*)**. Yang dimaksud dengan metode analitik adalah metode penyelesaian model matematika dengan rumus-rumus aljabar yang sudah baku (*lazim*).

Sebagai contoh ilustrasi, tinjau sekumpulan persoalan matematik di bawah ini. Bagaimana cara anda menyelesaikannya?

- (i) Tentukan akar-akar persamaan polinom:
- $$23.4x^7 - 1.25x^6 + 120x^4 + 15x^3 - 120x^2 - x + 100 = 0$$

- (ii) Tentukan harga x yang memenuhi persamaan:

$$\sqrt{27.8e^{5x} - \frac{1}{x}} = \cos^{-1} \frac{(120x^2 + \sqrt{2}x)}{17x - 65}$$

- (iii) Selesaikan sistem persamaan lanjar (*linear*):

$$\begin{aligned}
 1.2a - 3b - 12c + 12d + 4.8e - 5.5f + 100g &= 18 \\
 0.9a + 3b - c + 16d + 8e - 5f - 10g &= 17 \\
 4.6a + 3b - 6c - 2d + 4e + 6.5f - 13g &= 19 \\
 3.7a - 3b + 8c - 7d + 14e + 8.4f + 16g &= 6 \\
 2.2a + 3b + 17c + 6d + 12e - 7.5f + 18g &= 9 \\
 5.9a + 3b + 11c + 9d - 5e - 25f - 10g &= 0 \\
 1.6a + 3b + 1.8c + 12d - 7e + 2.5f + g &= -5
 \end{aligned}$$

- (iv) Tentukan nilai maksimum fungsi tiga matra (*dimension*):

$$f(x,y) = \cos \frac{x - \sqrt{\sin(x)} + 3}{4 + (xy)^2} + \sin(3xy - 1) - \tan\left(\frac{x(0.08 + \cos(x))}{y}\right)$$

- (v) Bila diperoleh tabulasi titik-titik (x,y) sebagai berikut (yang dalam hal ini rumus fungsi $y = f(x)$ tidak diketahui secara eksplisit):

x	$y = f(x)$
2.5	1.4256
3.0	1.7652
3.5	2.0005
4.4	2.8976
6.8	3.8765

Hitung taksiran nilai y untuk $x = 3.8$!

- (vi) Berdasarkan titik-titik data pada tabel persoalan (v) di atas, berapa nilai $f'(3.5)$ dan nilai $f''(3.5)$?

- (vii) Hitung nilai integral-tentu berikut:

$$\int_{1.2}^{2.5} \left(\sqrt{(45.3e^{7x} + \frac{100}{x})^4 + \frac{4}{(x^2 + 1)}} \right) dx$$

- (viii) Diberikan persamaan differensial biasa (PDB) dengan nilai awal:

$$150y'' + 2y't = \frac{\sqrt{\ln(21t+40)y}}{t^2} + 120 \quad ; y'(0) = 0, y(0) = 1.2$$

Hitung nilai y pada $t = 1.8!$

Menghadapi soal-soal seperti di atas, anda mungkin menyerah, atau mungkin mengatakan bahwa soal-soal tersebut tidak dapat diselesaikan dengan metode analitik yang biasa anda kenal. Soal (i) misalnya, biasanya untuk polinom derajat 2 orang masih dapat mencari akar-akar polinom dengan rumus abc yang terkenal itu yaitu

$$x_{1,2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \quad (\text{P.1.1})$$

namun, untuk polinom derajat > 2 , seperti pada soal (i), tidak terdapat rumus aljabar untuk menghitung akar polinom. Yang mungkin kita lakukan adalah dengan memanipulasi polinom, misalnya dengan memfaktorkan (atau menguraikan) polinom tersebut menjadi perkalian beberapa suku. Semakin tinggi derajat polinom, jelas semakin sukar memfaktorkannya. Ada juga beberapa alternatif lain. Yang pertama dengan cara coba-coba seperti **metode pembagian sintetis Horner**. Dengan metode ini, polinom dibagi dengan sebuah bilangan. Jika sisa pembagiannya nol, maka bilangan tersebut adalah akar polinom. Cara kedua adalah secara grafik, yaitu dengan merajah kurva fungsi di atas kertas grafik, kemudian berdasarkan gambar kurva, kita mengambil tarikan akar secara kasar, yaitu titik poyong kurva dengan sumbu-x. Cara ini, selain kaku dan tidak praktis, ketelitian akar yang diperoleh sangat bergantung pada ketelitian penggambaran kurva.. Lagipula, merajah kurva pada kertas grafik hanya terbatas pada fungsi yang dapat digambarkan pada bidang dua matra atau tiga matra. Untuk fungsi dengan peubah lebih besar dari 3 jelas tidak dapat (malah tidak mungkin) kita gambar kurvanya. Soal nomor (ii) masih sejenis dengan soal (i), yaitu menentukan nilai x yang memenuhi kedua persamaan.

Untuk soal nomor (iii), juga tidak ada rumus yang baku untuk menemukan solusi sistem persamaan lanjar. Apabila sistem persamaannya hanya berupa dua garis lurus dengan dua peubah, kita masih dapat menemukan solusinya (dalam hal ini titik potong kedua garis) dengan menggunakan rumus titik potong dua buah garis atau dengan aturan Cramer. Kita juga dapat menemukan titik potong tersebut dengan menggambar kedua garis pada kertas grafik. Untuk sistem yang terdiri dari tiga buah persamaan lanjar dengan tiga peubah, aturan Cramer masih dapat digunakan untuk memecahkan sistem. Tetapi untuk sistem dengan jumlah persamaan dan jumlah peubah lebih besar dari tiga, tidak ada rumus yang dapat dipakai untuk memecahkannya.

Pada soal nomor (iv), relatif sukar mencari titik optimum fungsi yang memiliki banyak peubah. Untuk menentukan titik optimum (titik ekstrim fungsi), pertama-tama orang harus menentukan turunan fungsi, menjadikan ruas kanannya sama dengan nol, lalu memeriksa jenis titik ekstrimnya. Bila fungsinya cukup rumit dan disusun oleh banyak peubah, menghitung turunan fungsi menjadi pekerjaan yang sukar atau bahkan tidak mungkin dilakukan.

Pertanyaan yang agak klasik sering muncul pada soal nomor (v): bagaimana menghitung nilai sebuah fungsi bila rumus fungsinya sendiri tidak diketahui? Kita semua tahu bahwa nilai fungsi diperoleh dengan cara menyulihkan (*substitute*) harga dari peubahnya ke dalam rumus fungsi. Masalahnya, bagaimana kalau persamaan fungsi tersebut tidak diketahui. Yang tersedia hanyalah beberapa buah data diskrit (*discrete*) dalam bentuk tabel. Persoalan semacam nomor (v) ini acapkali muncul pada pengamatan fenomena alam, baik berupa eksperimen di laboratorium maupun penelitian di lapangan yang melibatkan beberapa parameter (misalnya suhu, tekanan, waktu, dan sebagainya). Pengamat tidak mengetahui relasi yang menghubungkan parameter-parameter itu. Pengamat hanya dapat mengukur nilai-nilai parameter tersebut dengan menggunakan alat ukur seperti sensor, termometer, barometer, dan sebagainya. Tidak satupun metode analitik yang tersedia untuk menyelesaikan persoalan jenis ini. Begitu juga soal nomor (vi) melahirkan pertanyaan yang sama, bagaimana menghitung nilai turunan fungsi bila fungsinya sendiri tidak diketahui?.

Pada soal nomor (vii), tidak ada teknik integrasi yang dapat digunakan untuk fungsi yang bentuknya rumit itu. Begitu juga pada soal nomor (viii), tidak terdapat metode persamaan diferensial untuk menyelesaikannya. Dengan kata lain, persoalan (vii) dan (viii) tidak mempunyai solusi analitik.

1.1 Metode Analitik versus Metode Numerik

Contoh-contoh yang dikemukakan di atas memperlihatkan bahwa kebanyakan persoalan matematika tidak dapat diselesaikan dengan metode analitik. Metode analitik disebut juga **metode sejati** karena ia memberi kita **solusi sejati** (*exact solution*) atau solusi yang sesungguhnya, yaitu solusi yang memiliki **galat** (*error*) sama dengan nol! Sayangnya, metode analitik hanya unggul untuk sejumlah persoalan yang terbatas, yaitu persoalan yang memiliki tafsiran geometri sederhana serta bermatra rendah [CHA88]. Padahal persoalan yang muncul dalam dunia nyata seringkali nirlanjar serta melibatkan bentuk dan proses yang rumit. Akibatnya nilai praktis penyelesaian metode analitik menjadi terbatas.

Bila metode analitik tidak dapat lagi diterapkan, maka solusi persoalan sebenarnya masih dapat dicari dengan menggunakan **metode numerik**. Metode numerik adalah teknik yang digunakan untuk memformulasikan persoalan matematik sehingga dapat dipecahkan dengan operasi perhitungan/aritmetika biasa (tambah, kurang, kali, dan bagi). Metode artinya cara, sedangkan numerik artinya angka. Jadi metode numerik secara harafiah berarti cara berhitung dengan menggunakan angka-angka.

Perbedaan utama antara metode numerik dengan metode analitik terletak pada dua hal. Pertama, solusi dengan menggunakan metode numerik selalu berbentuk angka. Bandingkan dengan metode analitik yang biasanya menghasilkan solusi dalam bentuk fungsi matematik yang selanjutnya fungsi mateamtk tersebut dapat dievaluasi untuk menghasilkan nilai dalam bentuk angka.

Kedua, dengan metode numerik, kita hanya memperoleh solusi yang menghampiri atau mendekati solusi sejati sehingga solusi numerik dinamakan juga **solusi hampiran** (*approxomation*) atau **solusi pendekatan**, namun solusi hampiran dapat dibuat seteliti yang kita inginkan. Solusi hampiran jelas tidak tepat sama dengan solusi sejati, sehingga ada selisih antara keduanya. Selisih inilah yang disebut dengan galat (*error*).

Sebagai contoh ilustrasi penyelesaian dengan metode numerik, pandanglah sebuah persoalan integrasi-tentu berikut

$$I = \int_{-1}^1 (4 - x^2) dx \quad (\text{P.1.2})$$

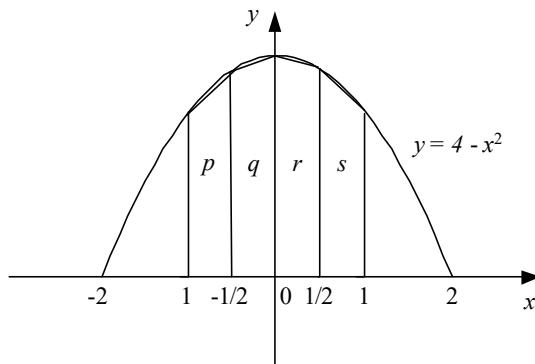
Dengan metode analitik, kita dapat menemukan solusi sejatinya dengan mudah. Di dalam kalkulus integral tentu kita mengetahui teknik pengintegralan untuk fungsi sederhana:

$$\int ax^n dx = \frac{1}{n+1} ax^{n+1} + C \quad (\text{P.1.3})$$

Maka, berdasarkan (P.1.3), kita dapat melakukan pengintegralan suku-suku dari fungsi integralnya lalu menghitung nilai integral-tentunya sebagai berikut:

$$I = \int_{-1}^1 (4 - x^2) dx = [4x - x^3/3] \Big|_{x=-1}^{x=1} = \{4(1) - (1)/3\} - \{4(-1) - (-1)/3\} = 22/3$$

Perhatikanlah bahwa $4x - x^3/3$ adalah solusi analitik dalam bentuk fungsi matematik, sedangkan $22/3$ adalah nilai numerik integral-tentu yang diperoleh dengan cara mengevaluasi fungsi matematik tersebut untuk batas-batas integrasi $x = 1$ dan $x = -1$.



Gambar 1.1 Integrasi $f(x) = 4 - x^2$ secara numerik

Bandingkan penyelesaian di atas bila persoalan integrasi tersebut diselesaikan dengan metode numerik sebagai berikut. Sekali lagi, di dalam kalkulus integral kita tentu masih ingat bahwa interpretasi geometri integral $f(x)$ dari $x = a$ sampai $x = b$ adalah luas daerah yang dibatasi oleh kurva $f(x)$, sumbu- x , dan garis $x = a$ dan $x = b$. Luas daerah tersebut dapat *dihampiri* dengan cara sebagai berikut. Bagilah daerah integrasi $[-1, 1]$ atas sejumlah trapesium dengan lebar 0.5 (Gambar 1.1). Maka, luas daerah integrasi dihampiri dengan luas kempat buah trapesium, atau

$$\begin{aligned}
 I &\approx p + q + r + s & (P.1.4) \\
 &\approx \{[f(-1) + f(-1/2)] \times 0.5/2\} + \{[f(-1/2) + f(0)] \times 0.5/2\} + \\
 &\quad \{[f(0) + f(1/2)] \times 0.5/2\} + \{[f(1/2) + f(1)] \times 0.5/2\} \\
 &\approx 0.5/2 \{f(-1) + 2f(-1/2) + 2f(0) + 2f(1/2) + f(1)\} \\
 &\approx 0.5/2 \{3 + 7.5 + 8 + 7.5 + 3\} \\
 &\approx 7.25
 \end{aligned}$$

yang merupakan solusi hampiran (tanda “ \approx ” artinya “kira-kira”) terhadap solusi sejati ($22/3$). Galat solusi hampiran terhadap solusi sejati adalah

$$\text{galat} = |7.25 - 22/3| = |7.25 - 7.33\dots| = 0.0833\dots$$

Tentu saja kita dapat memperkecil galat ini dengan membuat lebar trapesium yang lebih kecil (yang artinya jumlah trapesium semakin banyak, yang berarti jumlah komputasi semakin banyak). Contoh ini juga memperlihatkan bahwa meskipun solusi dengan metode numerik merupakan hampiran, tetapi hasilnya

dapat dibuat seteliti mungkin dengan mengubah parameter komputasi (pada contoh perhitungan integral di atas, lebar trapesium yang dikurangi).

1.2 Metode Numerik dalam Bidang Rekayasa

Dalam bidang rekayasa, kebutuhan untuk menemukan solusi persoalan secara praktis adalah jelas. Dari kacamata rekayasawan, masih tampak banyak cara penyelesaian persoalan matematik yang dirasa terlalu sulit atau dalam bentuk yang kurang kongkrit. Penyelesaian analitik yang sering diberikan oleh kaum matematika kurang berguna bagi rekayasawan, karena ia harus dapat mentransformasikan solusi matematika yang sejati ke dalam bentuk berwudud yang biasanya meninggalkan kaidah kesejatiannya [BES97]. Solusi hampiran biasanya sudah memenuhi persyaratan rekayasa dan dapat diterima sebagai solusi. Lagipula, banyak persoalan matematika dalam bidang rekayasa yang hanya dapat dipecahkan secara hampiran. Kadang-kadang dapat pula terjadi bahwa metode analitik hanya menjamin keberadaan (atau hanya mengkarakteristikkan beberapa properti umum) solusi, tetapi tidak memberikan cara menemukan solusi tersebut[KRE88].

Bagi rekayasawan, solusi yang diperoleh secara analitik kurang berguna untuk tujuan numerik. Persoalan rekayasa dalam prakteknya tidak selalu membutuhkan solusi dalam bentuk fungsi matematika menerus (*continuous*). Rekayasawan seringkali menginginkan solusi dalam bentuk numerik, misalnya persoalan integral tentu dan persamaan diferensial. Sebuah contoh dalam termodinamika dikemukakan di bawah ini untuk memperjelas pernyataan ini [KRE88].

Sebuah bola logam dipanaskan sampai pada suhu 100°C . Kemudian, pada saat $t = 0$, bola itu dimasukkan ke dalam air yang bersuhu 30°C . Setelah 3 menit, suhu bola berkurang menjadi 70°C . Tentukan suhu bola setelah 22.78 menit menit. Diketahui tetapan pendinginan bola logam itu adalah 0.1865.

Dengan menggunakan hukum pendinginan Newton, laju pendinginan bola setiap detiknya adalah

$$dT/dt = -k(T - 30)$$

yang dalam hal ini k adalah tetapan pendinginan bola logam yang harganya 0.1865. Bagi matematikawan, untuk menentukan suhu bola pada $t = 22.78$ menit, persamaan diferensial tersebut harus diselesaikan terlebih dahulu agar suhu T sebagai fungsi dari waktu t ditemukan. Persamaan diferensial ini dapat diselesaikan dengan metode kalkulus diferensial. Solusi umumnya adalah

$$T(t) = ce^{-kt} + 30$$

Nilai awal yang diberikan adalah $T(0)=100$. Dengan menggunakan nilai awal ini, solusi khusus persamaan diferensial adalah

$$T(t) = 70e^{-0.1865 t} + 30$$

Dengan menyulihkan $t = 22.78$ ke dalam persamaan T , diperoleh

$$T(22.78) = 70e^{-0.1865 \times 22.78} + 30 = 31^{\circ}\text{C}.$$

Jadi, suhu bola setelah 22.78 menit adalah 31°C .

Bagi rekayawan, solusi persamaan diferensial yang berbentuk fungsi menerus ini tidak terlalu penting (bahkan beberapa persamaan diferensial tidak dapat dicari solusi khususnya karena memang tidak ada teknik yang baku untuk menyelesaiakannya). Dalam praktek di lapangan, seringkali para rekayawan hanya ingin mengetahui berapa suhu bola logam setelah t tertentu misalnya setelah 30 menit tanpa perlu mencari solusi khususnya dalam bentuk fungsi terlebih dahulu. Rekayawan cukup memodelkan sistem ke dalam persamaan diferensial, lalu solusi untuk t tertentu dicari secara numerik.

1.3 Apakah Metode Numerik Hanya untuk Persoalan Matematika yang Rumit Saja?

Tentu saja tidak! Anda jangan berpikiran bahwa metode numerik hanya dapat menyelesaikan persoalan rumit saja. Metode numerik berlaku umum, yakni ia dapat diterapkan untuk menyelesaikan persoalan matematika sederhana (yang juga dapat diselesaikan dengan metode analitik) maupun persoalan matematika yang tergolong rumit (yang metode analitik pun belum tentu dapat menyelesaiakannya). Sebagai contoh, dengan metode numerik kita dapat menghitung integral

$$\int_0^\pi \sqrt{1 + \cos^2(x)} dx$$

sama mudahnya menghitung

$$\int_0^1 2x^2 dx$$

1.4 Peranan Komputer dalam Metode Numerik

Komputer berperan besar dalam perkembangan bidang metode numerik. Hal ini mudah dimengerti karena perhitungan dengan metode numerik adalah berupa operasi aritmetika seperti penjumlahan, perkalian, pembagian, plus membuat perbandingan. Sayangnya, jumlah operasi aritmetika ini umumnya sangat banyak dan berulang, sehingga perhitungan secara manual sering menjemuhan. Manusia (yang melakukan perhitungan manual ini) dapat membuat kesalahan dalam melakukannya. Dalam hal ini, komputer berperan mempercepat proses perhitungan tanpa membuat kesalahan.

Penggunaan komputer dalam metode numerik antara lain untuk memprogram. Langkah-langkah metode numerik diformulasikan menjadi program komputer. Program ditulis dengan bahasa pemrograman tertentu, seperti FORTRAN, PASCAL, C, C++, BASIC, dan sebagainya.

Sebenarnya, menulis program numerik tidak selalu diperlukan. Di pasaran terdapat banyak program aplikasi komersil yang langsung dapat digunakan. Beberapa contoh aplikasi yang ada saat ini adalah *MathLab*, *MathCad*, *Maple*, *Mathematica*, *Eureka*, dan sebagainya. Selain itu, terdapat juga *library* yang berisi rutin-rutin yang siap digabung dengan program utama yang ditulis pengguna, misalnya *IMSL (International Mathematical and Statistical Library)* *Math/Library* yang berisi ratusan rutin-rutin metode numerik.

Selain mempercepat perhitungan numerik, dengan komputer kita dapat mencoba berbagai kemungkinan solusi yang terjadi akibat perubahan beberapa parameter. Solusi yang diperoleh juga dapat ditingkatkan ketelitiannya dengan mengubah-ubah nilai parameter.

Kemajuan komputer digital telah membuat bidang metode numerik berkembang secara dramatis. Tidak ada bidang matematika lain yang mengalami kemajuan penting secepat metode numerik. Tentu saja alasan utama penyebab kemajuan ini adalah perkembangan komputer itu sendiri, dari komputer mikro sampai komputer *Cray*, dan kita melihat perkembangan teknologi komputer tidak pernah berakhir. Tiap generasi baru komputer menghadirkan keunggulan seperti waktu, memori, ketelitian, dan kestabilan perhitungan. Hal ini membuat ruang penelitian semakin terbuka luas. Tujuan utama penelitian itu adalah pengembangan algoritma numerik yang lebih baik dengan memanfaatkan keunggulan komputer semaksimal

mungkin. Banyak algoritma baru lahir atau perbaikan algoritma yang lama didukung oleh komputer.

Bagian mendasar dari perhitungan rekayasa yang dilakukan saat ini adalah perhitungan "waktu nyata" (*real time computing*), yaitu perhitungan keluaran (hasil) dari data yang diberikan dilakukan secara simultan dengan *event* pembangkitan data tersebut, sebagaimana yang dibutuhkan dalam mengendalikan proses kimia atau reaksi nuklir, memandu pesawat udara atau roket dan sebagainya [KRE88]. Karena itu, kecepatan perhitungan dan kebutuhan memori komputer adalah pertimbangan yang sangat penting.

Jelaslah bahwa kecepatan tinggi, keandalan, dan fleksibilitas komputer memberikan akses untuk penyelesaian masalah praktik. Sebagai contoh, solusi sistem persamaan lanjar yang besar menjadi lebih mudah dan lebih cepat diselesaikan dengan komputer. Perkembangan yang cepat dalam metode numerik antara lain ialah penemuan metode baru, modifikasi metode yang sudah ada agar lebih mangkus, analisis teoritis dan praktis algoritma untuk proses perhitungan baku, pengkajian galat, dan penghilangan jebakan yang ada pada metode [KRE88].

1.5 Mengapa Kita Harus Mempelajari Metode Numerik?

Seperti sudah disebutkan pada bagian awal bab ini, para rekayasan dan para ahli ilmu alam, dalam pekerjaannya sering berhadapan dengan persamaan matematik. Persoalan yang muncul di lapangan diformulasikan ke dalam model yang berbentuk persamaan matematika. Persamaan tersebut mungkin sangat kompleks atau jumlahnya lebih dari satu. Metode numerik, dengan bantuan komputer, memberikan cara penyelesaian persoalan matematika dengan cepat dan akurat.

Terdapat beberapa alasan tambahan mengapa kita harus mempelajari metode numerik [CHA91]:

1. Metode numerik merupakan alat bantu pemecahan masalah matematika yang sangat ampuh. Metode numerik mampu menangani sistem persamaan besar, kenirlanjaran, dan geometri yang rumit yang dalam praktik rekayasa seringkali tidak mungkin dipecahkan secara analitik.
2. Seperti sudah disebutkan pada upapab 1.4, di pasaran banyak tersedia program aplikasi numerik komersil. Penggunaan aplikasi tersebut menjadi lebih berarti

bila kita memiliki pengetahuan metode numerik agar kita dapat memahami cara paket tersebut menyelesaikan persoalan.

3. Kita dapat membuat sendiri program komputer tanpa harus membeli paket programnya. Seringkali beberapa persoalan matematika yang tidak selalu dapat diselesaikan oleh program aplikasi. Sebagai contoh, misalkan ada program aplikasi tertentu yang tidak dapat dipakai untuk menghitung integrasi lipat dua, \iint , atau lipat tiga, \iiint . Mau tidak mau, kita harus menulis sendiri programnya. Untuk itu, kita harus mempelajari cara pemecahan integral lipat dua atau lebih dengan metode numerik.
4. Metode numerik menyediakan sarana untuk memperkuat kembali pemahaman matematika. Karena, metode numerik ditemukan dengan menyederhanakan matematika yang lebih tinggi menjadi operasi matematika yang mendasar.

1.6 Tahap-Tahap Memecahkan Persoalan Secara Numerik

Ada enam tahap yang dilakukan dalam pemecahan persoalan dunia nyata dengan metode numerik, yaitu

1. Pemodelan

Ini adalah tahap pertama. Persoalan dunia nyata dimodelkan ke dalam persamaan matematika (lihat contoh ilustrasi pada upabab 1.2)

2. Penyederhanaan model

Model matematika yang dihasilkan dari tahap 1 mungkin saja terlalu kompleks, yaitu memasukkan banyak peubah (variable) atau parameter. Semakin kompleks model matematikanya, semakin rumit penyelesaiannya. Mungkin beberapa andaian dibuat sehingga beberapa parameter dapat diabaikan. Contohnya, faktor gesekan udara diabaikan sehingga koefisian gesekan di dalam model dapat dibuang. Model matematika yang diperoleh dari penyederhanaan menjadi lebih sederhana sehingga solusinya akan lebih mudah diperoleh.

3. Formulasi numerik

Setelah model matematika yang sederhana diperoleh, tahap selanjutnya adalah memformulasikannya secara numerik, antara lain:

- a. menentukan metode numerik yang akan dipakai bersama-sama dengan analisis galat awal (yaitu taksiran galat, penentuan ukuran langkah, dan sebagainya).

Pemilihan metode didasari pada pertimbangan:

- apakah metode tersebut teliti?
- apakah metode tersebut mudah diprogram dan waktu pelaksanaannya cepat?
- apakah metode tersebut tidak peka terhadap perubahan data yang cukup kecil?

b. menyusun algoritma dari metode numerik yang dipilih.

4. **Pemrograman**

Tahap selanjutnya adalah menerjemahkan algoritma ke dalam program komputer dengan menggunakan salah satu bahasa pemrograman yang dikuasai.

5. **Operasional**

Pada tahap ini, program komputer dijalankan dengan data uji coba sebelum data yang sesungguhnya.

6. **Evaluasi**

Bila program sudah selesai dijalankan dengan data yang sesungguhnya, maka hasil yang diperoleh diinterpretasi. Interpretasi meliputi analisis hasil *run* dan membandingkannya dengan prinsip dasar dan hasil-hasil empirik untuk menaksir kualitas solusi numerik, dan keputusan untuk menjalankan kembali program dengan untuk memperoleh hasil yang lebih baik.

1.7 Peran Ahli Informatika dalam Metode Numerik

Dari tahap-tahap pemecahan yang dikemukakan di atas, tahap 1 dan 2 melibatkan para pakar di bidang persoalan yang bersangkutan. Kalau persoalannya dalam bidang teknik Sipil, maka orang dari bidang Sipil-lah yang menurunkan model matematikanya. Kalau persoalannya menyangkut bidang Teknik Kimia (TK), maka ahli Teknik Kimia-lah yang mempunyai kemampuan membentuk model matematikanya.

Dimanakah peran orang Informatika? Orang Informatika baru berperan pada tahap 3 dan 4, dan 5. Tetapi, agar lebih memahami dan menghayati persoalan, sebaiknya orang Informatika juga ikut dilibatkan dalam memodelkan, namun perannya hanyalah sebagai pendengar.

Tahap 6 memerlukan kerjasama informatikawan dengan pakar bidang bersangkutan. Bersama-sama dengan pakar, informatikawan mendiskusikan hasil numerik yang diperoleh, apakah hasil tersebut sudah dapat diterima, apakah perlu dilakukan perubahan parameter, dsb.

1.8 Perbedaan Metode Numerik dengan Analisis Numerik

Untuk persoalan tertentu tidaklah cukup kita hanya menggunakan metode untuk memperoleh hasil yang diinginkan; kita juga perlu mengetahui apakah metode tersebut memang memberikan solusi hampiran, dan seberapa bagus hampiran itu [BUC92]. Hal ini melahirkan kajian baru, yaitu **analisis numerik**.

Metode numerik dan analisis numerik adalah dua hal yang berbeda. Metode adalah algoritma, menyangkut langkah-langkah penyelesaian persoalan secara numerik, sedangkan analisis numerik adalah terapan matematika untuk menganalisis metode [NOB72]. Dalam analisis numerik, hal utama yang ditekankan adalah analisis galat dan kecepatan konvergensi sebuah metode. Teorema-teorema matematika banyak dipakai dalam menganalisis suatu metode. Di dalam buku ini, kita akan memasukkan beberapa materi analisis numerik seperti galat metode dan kekonvergenan metode.

Tugas para analis numerik ialah mengembangkan dan menganalisis metode numerik. Termasuk di dalamnya pembuktian apakah suatu metode konvergen, dan menganalisis batas-batas galat solusi numerik. Terdapat banyak sumber galat, diantaranya tingkat ketelitian model matematika, sistem aritmetik komputer, dan kondisi yang digunakan untuk menghentikan proses pencarian solusi. Semua ini harus dipertimbangkan untuk menjamin ketelitian solusi akhir yang dihitung.

1.9 Materi Apa yang Terdapat di dalam Buku Ini?

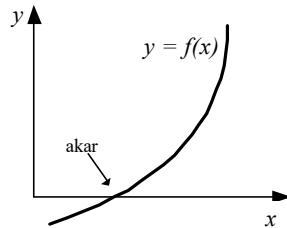
Ada enam pokok bahasan yang ditulis di dalam buku ini:

1. Solusi persamaan nirlanjar.
2. Solusi sistem persamaan lanjar.
3. Interpolasi polinom.
4. Turunan numerik.
5. Integrasi numerik.
6. Solusi persamaan diferensial biasa dengan nilai awal.

Ringkasan masing-masing pokok bahasan 1 sampai 6 adalah sebagai berikut:

1. Solusi persamaan nirlanjar

Selesaikan $f(x) = 0$ untuk x .

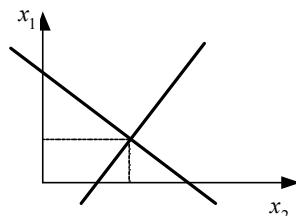


2. Solusi sistem persamaan lanjar

Selesaikan sistem persamaan lanjar

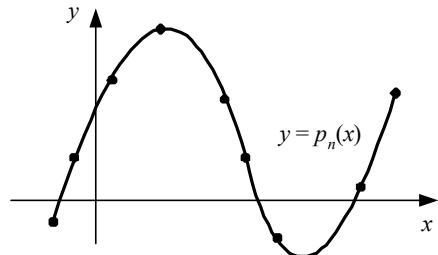
$$\begin{aligned} a_{11}x_1 + a_{12}x_2 &= c_1 \\ a_{21}x_1 + a_{22}x_2 &= c_2 \end{aligned}$$

untuk harga-harga x_1 dan x_2 .



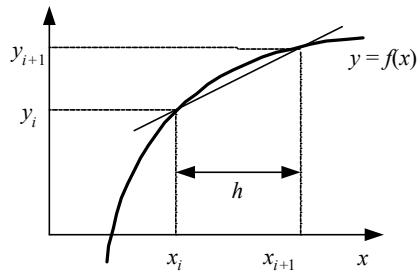
3. Interpolasi polinom

Diberikan titik-titik (x_0, y_0) , (x_1, y_1) , ..., (x_n, y_n) . Tentukan polinom $p_n(x)$ yang melalui semua titik tersebut



4. Turunan numerik

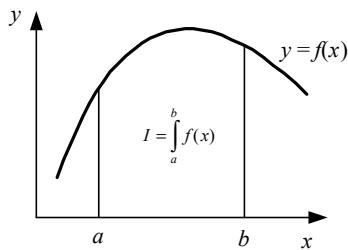
Diberikan titik (x_i, y_i) dan titik (x_{i+1}, y_{i+1}) . Tentukan $f'(x_i)$.



5. Integrasi numerik

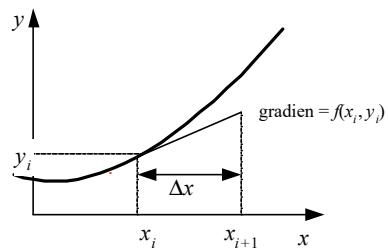
Hitung integral

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



6. Solusi persamaan diferensial biasa dengan nilai awal

Diberikan $dy/dx = f(x,y)$ dan
dengan nilai awal $y_0 = y(x_0)$
Tentukan nilai $y(x_i)$ untuk $x_i \in R$



Sebelum menjelaskan keenam pokok bahasan tersebut, kita perlu terlebih dahulu mengerti konsep galat dalam metode numerik. Konsep galat diberikan sebagai topik tersendiri.

Perjalanan seribu mil dimulai dari satu langkah
(pepatah)

Bab 2

Deret Taylor dan Analisis Galat

Matematik selalu memperlihatkan rasa ingin tahu untuk dapat diterapkan di alam, dan ini dapat mengungkapkan kaitan yang dalam antara pikiran kita dan alam. Kita membicarakan semesta, bagian dari alam. Jadi, tidak mengherankan bahwa sistem logik dan matematika kita bernyanyi seirama dengan alam.
(George Zebrowski)

Columbus menemukan Amerika melalui kesalahan.
(Majalah Intisari)

Prasyarat yang diperlukan untuk mempelajari metode numerik adalah matematika. Matematika adalah ilmu dasar, jadi anda diharapkan sudah memiliki pengetahuan mengenai konsep fungsi, geometri, konsep kalkulus seperti turunan dan integral, dan sebagainya. Tidak paham terlalu dalam tidak apa, yang penting anda mengerti.

Banyak teorema matematika yang dipakai di sini. Dari sekian banyak teorema tersebut, ada satu teorema yang menjadi **kakas** (*tools*) yang sangat penting dalam metode numerik, yaitu teorema **deret Taylor**. Deret Taylor adalah kakas yang utama untuk menurunkan suatu metode numerik.

Pada bagian yang lain, kita akan membahas konsep galat. Seperti sudah dijelaskan di dalam Bab 1, solusi yang diperoleh secara numerik adalah nilai hampiran dari solusi sejati. Ini berarti terdapat galat (*error*) pada solusi hampiran tersebut. Bab 2 ini akan menjelaskan konsep galat, cara mengukur galat, penyebab galat, perambatan galat, dan ketidakstabilan perhitungan akibat galat.

2.1 Deret Taylor

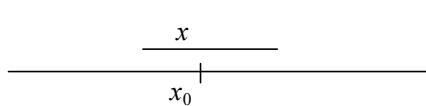
Kebanyakan dari metode-metode numerik yang diturunkan didasarkan pada penghampiran fungsi ke dalam bentuk polinom. Fungsi yang bentuknya kompleks menjadi lebih sederhana bila dihampiri dengan polinom, karena polinom merupakan bentuk fungsi yang paling mudah dipahami kelakuannya.

Kalau perhitungan dengan fungsi yang sesungguhnya menghasilkan solusi sejati, maka perhitungan dengan fungsi hampiran menghasilkan solusi hampiran. Pada bab 1 sudah dikatakan bahwa solusi numerik merupakan pendekatan (hampiran) terhadap solusi sejati, sehingga terdapat galat sebesar selisih antara solusi sejati dengan solusi hampiran. Galat pada solusi numerik harus dihubungkan dengan seberapa teliti polinom menghampiri fungsi sebenarnya. Kertas yang digunakan untuk membuat polinom hampiran adalah **deret Taylor**.

Definisi Deret Taylor

Andaikan f dan semua turunannya, f' , f'' , f''' , ..., menerus di dalam selang $[a, b]$. Misalkan $x_0 \in [a, b]$, maka untuk nilai-nilai x di sekitar x_0 (Gambar 2.1) dan $x \in [a, b]$, $f(x)$ dapat diperluas (diekspansi) ke dalam deret Taylor:

$$f(x) = f(x_0) + \frac{(x-x_0)}{1!} f'(x_0) + \frac{(x-x_0)^2}{2!} f''(x_0) + \dots + \frac{(x-x_0)^m}{m!} f^{(m)}(x_0) + \dots \quad (\text{P.2.1})$$



Gambar 2.1. Nilai-nilai x di sekitar x_0

Persamaan (P.2.1) merupakan penjumlahan dari suku-suku (*term*), yang disebut **deret**. Perhatikanlah bahwa deret Taylor ini panjangnya tidak berhingga sehingga untuk memudahkan penulisan suku-suku selanjutnya kita menggunakan tanda ellipsis (...). Jika dimisalkan $x - x_0 = h$, maka $f(x)$ dapat juga ditulis sebagai

$$f(x) = f(x_0) + \frac{h}{1!} f'(x_0) + \frac{h^2}{2!} f''(x_0) + \frac{h^3}{3!} f'''(x_0) + \dots + \frac{h^m}{m!} f^{(m)}(x_0) + \dots \quad (\text{P.2.2})$$

Contoh 2.1

Hampiri fungsi $f(x) = \sin(x)$ ke dalam deret Taylor di sekitar $x_0 = 1$.

Penyelesaian:

Kita harus menentukan turunan $\sin(x)$ terlebih dahulu sebagai berikut

$$\begin{aligned}f(x) &= \sin(x), \\f'(x) &= \cos(x), \\f''(x) &= -\sin(x), \\f'''(x) &= -\cos(x), \\f^{(4)}(x) &= \sin(x), \\&\text{dan seterusnya.}\end{aligned}$$

Maka, berdasarkan (P.2.1), $\sin(x)$ dihampiri dengan deret Taylor sebagai berikut:

$$\sin(x) = \sin(1) + \frac{(x-1)}{1!} \cos(1) + \frac{(x-1)^2}{2!} (-\sin(1)) + \frac{(x-1)^3}{3!} (-\cos(1)) + \frac{(x-1)^4}{4!} \sin(1) + \dots$$

Bila dimisalkan $x - 1 = h$, maka, berdasarkan (P.2.2),

$$\begin{aligned}\sin(x) &= \sin(1) + h \cos(1) - \frac{h^2}{2} \sin(1) - \frac{h^3}{6} \cos(1) + \frac{h^4}{24} \sin(1) + \dots \\&= 0.8415 + 0.5403h - 0.4208h^2 - 0.0901h^3 + 0.0351h^4 + \dots\end{aligned}$$

■

Kasus khusus adalah bila fungsi diperluas di sekitar $x_0 = 0$, maka deretnya dinamakan **deret Maclaurin**, yang merupakan deret Taylor baku. Kasus $x_0 = 0$ paling sering muncul dalam praktek.

Contoh 2.2

Uraikan $\sin(x)$, e^x , $\cos(x)$, dan $\ln(x+1)$ masing-masing ke dalam deret Maclaurin.

Penyelesaian:

Beberapa turunan $\sin(x)$ sudah dihitung pada Contoh 2.1. Deret Maclaurin dari $\sin(x)$ adalah

$$\begin{aligned}\sin(x) &= \sin(0) + \frac{(x-0)}{1!} \cos(0) + \frac{(x-0)^2}{2!} (-\sin(0)) + \frac{(x-0)^3}{3!} (-\cos(0)) + \frac{(x-0)^4}{4!} \sin(0) + \dots \\&= x - \frac{x^3}{3!} + \frac{x^5}{5!} - \dots\end{aligned}$$

Untuk menentukan deret Maclaurin dari e^x , kita harus menentukan turunan e^x terlebih dahulu sebagai berikut: $f(x) = e^x$, $f'(x) = e^x$, $f''(x) = e^x$, $f'''(x) = e^x$, $f^{(4)}(x) = e^x$, dan seterusnya. Deret Maclaurin dari e^x adalah

$$\begin{aligned} e^x &= e^{(0)} + \frac{(x-0)^1}{1!} e^{(0)} + \frac{(x-0)^2}{2!} e^{(0)} + \frac{(x-0)^3}{3!} e^{(0)} + \frac{(x-0)^4}{4!} e^{(0)} + \dots \\ &= 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!} + \dots \end{aligned}$$

Untuk menentukan deret Maclaurin dari $\cos(x)$, kita harus menentukan turunan $\cos(x)$ terlebih dahulu sebagai berikut: $f(x) = \cos(x)$, $f'(x) = -\sin(x)$, $f''(x) = -\cos(x)$, $f'''(x) = \sin(x)$, $f^{(4)}(x) = \cos(x)$, dan seterusnya. Deret Maclaurin dari $\cos(x)$ adalah

$$= 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \dots$$

Untuk menentukan deret Maclaurin dari $\ln(x+1)$, kita harus menentukan turunan $\ln(x+1)$ terlebih dahulu sebagai berikut: $f(x) = \ln(x+1)$, $f'(x) = (x+1)^{-1}$, $f''(x) = -(x+1)^{-2}$, $f'''(x) = 2(x+1)^{-3}$, $f^{(4)}(x) = -6(x+1)^{-4}$, dan seterusnya. Deret Maclaurin dari $\ln(x+1)$ adalah

$$\begin{aligned} \ln(x+1) &= \ln(0+1) + \frac{(x-0)^1}{1!}(0+1)^{-1} + \frac{(x-0)^2}{2!}(-(0+1)^{-2}) + \frac{(x-0)^3}{3!}2(0+1)^{-3} + \frac{(x-0)^4}{4!}(-6(0+1)^{-4}) + \dots \\ &= x - \frac{x^2}{2} + \frac{x^3}{3} - \frac{x^4}{4} + \dots \end{aligned} \quad \blacksquare$$

Karena suku-suku deret Taylor tidak berhingga banyaknya, maka -untuk alasan praktis- deret Taylor dipotong sampai suku orde tertentu. Deret Taylor yang dipotong sampai suku orde ke- n dinamakan **deret Taylor terpotong** dan dinyatakan oleh:

$$f(x) \approx f(x_0) + \frac{(x-x_0)^1}{1!} f'(x_0) + \frac{(x-x_0)^2}{2!} f''(x_0) + \dots + \frac{(x-x_0)^n}{n!} f^{(n)}(x_0) + R_n(x) \quad (\text{P.2.3})$$

yang dalam hal ini,

$$R_n(x) = \frac{(x-x_0)^{(n+1)}}{(n+1)!} f^{(n+1)}(c), \quad x_0 < c < x \quad (\text{P.2.4})$$

disebut **galat** atau **sisa** (residu).

Dengan demikian deret Taylor yang dipotong sampai suku orde ke- n dapat ditulis sebagai

$$f(x) = P_n(x) + R_n(x) \quad (\text{P.2.5})$$

yang dalam hal ini,

$$P_n(x) = \sum_{k=1}^n \frac{(x - x_0)^k}{k!} f^{(k)}(x_0) \quad (\text{P.2.6})$$

$$R_n(x) = \frac{(x - x_0)^{(n+1)}}{(n+1)!} f^{(n+1)}(c) \quad , \quad x_0 < c < x \quad (\text{P.2.7})$$

Sebagai contoh, $\sin(x)$ pada Contoh 2.1 jika dihampiri dengan deret Taylor orde 4 di sekitar $x_0 = 1$ adalah:

$$\sin(x) = \sin(1) + \frac{(x-1)}{1!} \cos(1) - \frac{(x-1)^2}{2!} \sin(1) - \frac{(x-1)^3}{3!} \cos(1) + \frac{(x-1)^4}{4!} \sin(1) + R_4(x)$$

yang dalam hal ini,

$$R_4(x) = \frac{(x-1)^5}{5!} \cos(c) \quad , \quad 1 < c < x$$

Deret Taylor terpotong di sekitar $x_0 = 0$ disebut deret Maclaurin terpotong. Berdasarkan Contoh 2.2, deret MacLaurin terpotong untuk $\sin(x)$, e^x , $\cos(x)$, dan $\ln(x+1)$ adalah

$$\sin(x) = x - x^3/3! + x^5/5! + R_5(x) ; \quad R_5(x) = \frac{x^6}{6!} \cos(c) \quad (\text{sampai suku orde 5})$$

$$e^x = 1 + x + x^2/2! + x^3/3! + x^4/4! + R_4(x) ; \quad R_4(x) = \frac{x^5}{5!} e^c \quad (\text{sampai suku orde 4})$$

$$\cos(x) = 1 - x^2/4! + x^4/6! - x^6/6! + R_6(x) ; \quad R_6(x) = \frac{x^7}{7!} \cos(c) \quad (\text{sampai suku orde 6})$$

$$\ln(x+1) = x - x^2/2 + x^3/3 - x^4/4; \quad R_4(x) = 24 \frac{x^5}{5!} (c+1)^{-5} \quad (\text{sampai suku orde 4})$$

yang dalam hal ini, $0 < c < x$.

Fungsi pustaka matematika di dalam kalkulator dan *compiler* bahasa pemrograman dapat menggunakan deret Taylor untuk mengevaluasi nilai-nilai fungsi trigonometri dan fungsi transenden yang tidak dapat dihitung secara langsung.

Contoh 2.3

Hitunglah hampiran nilai $\cos(0.2)$, sudut dinyatakan dalam radian, dengan deret Maclaurin sampai suku orde $n = 6$.

Penyelesaian:

Dari hasil pada Contoh 2.2,

$$\cos(0.2) \approx 1 - 0.2^2/2 + 0.2^4/24 - 0.2^6/720 = 0.9800667 \\ (\text{sampai } 7 \text{ angka di belakang koma}) \quad \blacksquare$$

Program 2.1 di bawah ini menghitung nilai hampiran $\cos(x)$. Setiap suku kita hitung nilainya, jika nilainya lebih kecil dari toleransi (*epsilon*) yang kita spesifikasikan, maka perhitungan suku berikutnya dapat dihentikan.

Program 2.1 Program menghitung hampiran nilai $\cos(x)$

```

function cos(x:real):real;
{mengembalikan nilai cosinus sampai nilai suatu suku < ε }
const
    epsilon = 0.0000001; { toleransi nilai tiap suku }
    { jika nilai suku sudah lebih kecil dari epsilon, perhitungan cos dapat
      dihentikan }
var
    tanda, n : integer;
    suku, jumlah : real;

    (* function pangkat(x:real; n:integer):real;
    { Menghitung nilai  $x^n$  }      *)

    (* function faktorial(n:integer):integer;
    { Menghitung  $n!$  }           *)

begin
    suku:=1; {suku pertama deret cosinus}
    tanda:=1; {tanda (+/-) suku pertama}
    n:=0; {orde suku pertama}
    jumlah:=0; {jumlah deret cosinus, inisialisasi dengan 0}
    while abs(suku) >= epsilon do
        begin
            jumlah:=jumlah + suku; {jumlah deret cosinus}
            n:=n+2; {orde suku berikutnya}
            tanda:=-tanda; {tanda suku berikutnya}
        
```

```

        suku:=tanda*pangkat(x,n)/faktorial(n);
end;
{ abs(suku) < epsilon }
cos:=jumlah;
end;

```

Deret Taylor banyak digunakan untuk menurunkan metode-metode numerik. Deret Taylor yang terpotong digunakan sebagai titik awal dalam menurunkan metode. Anda sebaiknya dapat menguasai deret Taylor terlebih dahulu sebagai alat bantu yang penting dalam metode numerik.

2.2 Analisis Galat

Menganalisis galat sangat penting di dalam perhitungan yang menggunakan metode numerik. Galat berasosiasi dengan seberapa dekat solusi hampiran terhadap solusi sejatinya. Semakin kecil galatnya, semakin teliti solusi numerik yang didapatkan. Kita harus memahami dua hal: (a) bagaimana menghitung galat, dan (b) bagaimana galat timbul.

Misalkan \hat{a} adalah nilai hampiran terhadap nilai sejati a , maka selisih

$$\epsilon = a - \hat{a} \quad (\text{P.2.8})$$

disebut **galat**. Sebagai contoh, jika $\hat{a} = 10.5$ adalah nilai hampiran dari $a = 10.45$, maka galatnya adalah $\epsilon = -0.01$. Jika tanda galat (positif atau negatif) tidak dipertimbangkan, maka **galat mutlak** dapat didefinisikan sebagai

$$|\epsilon| = |a - \hat{a}| \quad (\text{P.2.9})$$

Sayangnya, ukuran galat ϵ kurang bermakna sebab ia tidak menceritakan seberapa besar galat itu dibandingkan dengan nilai sejatinya. Sebagai contoh, seorang anak melaporkan panjang sebatang kawat 99 cm, padahal panjang sebenarnya 100 cm. Galatnya adalah $100 - 99 = 1$ cm. Anak yang lain melaporkan panjang sebatang pensil 9 cm, padahal panjang sebenarnya 10 cm, sehingga galatnya juga 1 cm. Kedua galat pengukuran sama-sama bernilai 1 cm, namun galat 1 cm pada pengukuran panjang pensil lebih berarti daripada galat 1 cm pada pengukuran panjang kawat. Jika tidak ada informasi mengenai panjang sesungguhnya, kita mungkin menganggap kedua galat tersebut sama saja. Untuk mengatasi interpretasi nilai galat ini, maka galat harus *dinormalkan* terhadap nilai sejatinya. Gagasan ini melahirkan apa yang dinamakan **galat relatif**.

Galat relatif didefinisikan sebagai

$$\epsilon_R = \frac{\epsilon}{a} \quad (\text{P.2.10})$$

atau dalam persentase

$$\epsilon_R = \frac{\epsilon}{a} \times 100\% \quad (\text{P.2.11})$$

Karena galat dinormalkan terhadap nilai sejati, maka galat relatif tersebut dinamakan juga **galat relatif sejati**. Dengan demikian, pengukuran panjang kawat mempunyai galat relatif sejati $= 1/100 = 0.01$, sedangkan pengukuran panjang pensil mempunyai galat relatif sejati $= 1/10 = 0.1$.

Dalam praktek kita tidak mengetahui nilai sejati a , karena itu galat ϵ seringkali dinormalkan terhadap solusi hampirannya, sehingga galat relatifnya dinamakan **galat relatif hampiran**:

$$\epsilon_{RA} = \frac{\epsilon}{\hat{a}} \quad (\text{P.2.12})$$

Contoh 2.4

Misalkan nilai sejati $= 10/3$ dan nilai hampiran $= 3.333$. Hitunglah galat, galat mutlak, galat relatif, dan galat relatif hampiran.

Penyelesaian:

$$\text{galat} = 10/3 - 3.333 = 10/3 - 3333/1000 = 1/3000 = 0.000333\dots$$

$$\text{galat mutlak} = |0.000333\dots| = 0.000333\dots$$

$$\text{galat relatif} = (1/3000)/(10/3) = 1/1000 = 0.0001$$

$$\text{galat relatif hampiran} = (1/3000)/3.333 = 1/9999$$

■

Galat relatif hampiran yang dihitung dengan persamaan (P.2.12) masih mengandung kelemahan sebab nilai ϵ tetap membutuhkan pengetahuan nilai a (dalam praktek kita jarang sekali mengetahui nilai sejati a). Oleh karena itu, perhitungan galat relatif hampiran menggunakan pendekatan lain. Pada perhitungan numerik yang menggunakan pendekatan lelaran (*iteration*), ϵ_{RA} dihitung dengan cara

$$\epsilon_{RA} = \frac{a_{r+1} - a_r}{a_{r+1}} \quad (\text{P.2.13})$$

yang dalam hal ini a_{r+1} adalah nilai hampiran lelaran sekarang dan a_r adalah nilai hampiran lelaran sebelumnya. Proses lelaran dihentikan bila

$$|\varepsilon_{RA}| < \varepsilon_S$$

yang dalam hal ini ε_S adalah toleransi galat yang dispesifikasikan. Nilai ε_S menentukan ketelitian solusi numerik. Semakin kecil nilai ε_S , semakin teliti solusinya, namun semakin banyak proses lelarannya. Contoh 2.5 mengilustrasikan hal ini.

Contoh 2.5

Misalkan ada prosedur lelaran sebagai berikut

$$x_{r+1} = (-x_r^3 + 3)/6, \quad r = 0, 1, 2, 3, \dots .$$

Lelaran dihentikan bila kondisi $|\varepsilon_{RA}| < \varepsilon_S$, dalam hal ini ε_S adalah toleransi galat yang diinginkan. Misalkan dengan memberikan $x_0 = 0.5$, dan $\varepsilon_S = 0.00001$ kita memperoleh runtunan:

$$\begin{aligned} x_0 &= 0.5 \\ x_1 &= 0.4791667 ; |\varepsilon_{RA}| = |(x_1 - x_0)/x_1| = 0.043478 > \varepsilon_S \\ x_2 &= 0.4816638 ; |\varepsilon_{RA}| = |(x_2 - x_1)/x_2| = 0.0051843 > \varepsilon_S \\ x_3 &= 0.4813757 ; |\varepsilon_{RA}| = |(x_3 - x_2)/x_3| = 0.0005984 > \varepsilon_S \\ x_4 &= 0.4814091 ; |\varepsilon_{RA}| = |(x_4 - x_3)/x_4| = 0.0000693 > \varepsilon_S \\ x_5 &= 0.4814052. ; |\varepsilon_{RA}| = |(x_5 - x_4)/x_5| = 0.0000081 < \varepsilon_S , \text{ berhenti!} \end{aligned}$$

Pada lelaran ke-5, $|\varepsilon_{RA}| < \varepsilon_S$ sudah terpenuhi sehingga lelaran dapat dihentikan. ■

2.3 Sumber Utama Galat Numerik

Secara umum terdapat dua sumber utama penyebab galat dalam perhitungan numerik:

1. **Galat pemotongan** (*truncation error*)
2. **Galat pembulatan** (*round-off error*)

Selain kedua galat ini, masih ada sumber galat lain, antara lain [KRE88]:

- a. **Galat eksperimental**, yaitu galat yang timbul dari data yang diberikan, misalnya karena kesalahan pengukuran, ketidaktelitian alat ukur, dan sebagainya
- b. **Galat pemrograman**. Galat yang terdapat di dalam program sering dinamakan dengan kutu (*bug*), dan proses penghilangan galat ini dinamakan penirkutuan (*debugging*).

Kita tidak akan membicarakan kedua galat terakhir ini karena kontribusinya terhadap galat keseluruhan tidak selalu ada. Akan halnya galat utama, galat pemotongan dan galat pembulatan, keduanya selalu muncul pada solusi numerik. Terhadap kedua galat inilah perhatian kita fokuskan.

2.3.1 Galat Pemotongan

Galat pemotongan mengacu pada galat yang ditimbulkan akibat penggunaan hampiran sebagai pengganti formula eksak. Maksudnya, ekspresi matematik yang lebih kompleks “diganti” dengan formula yang lebih sederhana. Tipe galat pemotongan bergantung pada metode komputasi yang digunakan untuk penghampiran sehingga kadang-kadaang ia disebut juga **galat metode**. Misalnya, turunan pertama fungsi f di x_i dihampiri dengan formula

$$f'(x_i) \approx \frac{f(x_{i+1}) - f(x_i)}{h}$$

yang dalam hal ini h adalah lebar absis x_{i+1} dengan x_i . Galat yang ditimbulkan dari penghampiran turunan tersebut merupakan galat pemotongan.

Istilah “pemotongan” muncul karena banyak metode numerik yang diperoleh dengan penghampiran fungsi menggunakan deret Taylor. Karena deret Taylor merupakan deret yang tak-berhingga, maka untuk penghampiran tersebut deret Taylor kita hentikan/potong sampai suku orde tertentu saja. Penghentian suatu deret atau runtunan langkah-langkah komputasi yang tidak berhingga menjadi runtunan langkah yang berhingga itulah yang menimbulkan galat pemotongan.

Contohnya, hampiran fungsi $\cos(x)$ dengan bantuan deret Taylor di sekitar $x = 0$:

$$\cos(x) \approx 1 - \underbrace{\frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!}}_{\text{nilai hampiran}} \left| + \underbrace{\frac{x^8}{8!} - \frac{x^{10}}{10!} + \dots}_{\text{galat pemotongan}} \right.$$

pemotongan

Deret Taylor fungsi $\cos(x)$ sebenarnya tidak berhingga, namun untuk keperluan praktis, deret tersebut kita potong sampai suku orde tertentu, misalnya sampai suku orde $n = 6$ seperti pada contoh di atas. Kita melihat bahwa menghampiri $\cos(x)$ dengan deret Taylor sampai suku berderajat enam tidak memberikan hasil yang tepat. Galat pada nilai hampiran diakibatkan oleh pemotongan suku-suku deret. Jumlah suku-suku selanjutnya setelah pemotongan merupakan galat pemotongan untuk $\cos(x)$. Kita tidak dapat menghitung berapa persisnya galat pemotongan ini karena jumlah seluruh suku-suku setelah pemotongan tidak

mungkin dapat dihitung. Namun, kita dapat menghampiri galat pemotongan ini dengan rumus suku sisa:

$$R_n(x) = \frac{(x - x_0)^{(n+1)}}{(n+1)!} f^{(n+1)}(c) \quad , \quad x_0 < c < x$$

Pada contoh $\cos(x)$ di atas,

$$R_6(x) = \frac{x^7}{7!} \cos(c) \quad , \quad 0 < c < x$$

Nilai R_n yang tepat hampir tidak pernah dapat kita peroleh, karena kita tidak mengetahui nilai c sebenarnya terkecuali informasi bahwa c terletak pada suatu selang tertentu. Karenanya tugas kita adalah mencari nilai maksimum yang mungkin dari $|R_n|$ untuk c dalam selang yang diberikan itu [PUR84], yaitu:

$$|R_n(x)| < \underset{x_0 < c < x}{\text{Max}} |f^{(n+1)}(c)| \times \frac{(x - x_0)^{n+1}}{n+1!}$$

Contoh komputasi lain yang menghasilkan galat pemotongan adalah perhitungan dengan menggunakan skema lelaran (lihat Contoh 2.5). Sayangnya, tidak seperti deret Taylor, galat pemotongan pada perhitungan dengan skema lelaran tidak ada rumusnya.

Galat pemotongan pada deret Taylor dapat dikurangi dengan meningkatkan orde suku-sukunya, namun jumlah komputasinya menjadi lebih banyak. Pada metode yang menerapkan skema lelaran, galat pemotongan dapat dikurangi dengan memperbanyak lelaran. Hal ini ditunjukkan pada Contoh 2.5 dengan memberikan nilai ε_S yang sekecil mungkin

Contoh 2.6

[PUR89] Gunakan deret Taylor orde 4 di sekitar $x_0 = 1$ untuk menghampiri $\ln(0.9)$ dan berikan taksiran untuk galat pemotongan maksimum yang dibuat.

Penyelesaian:

Tentukan turunan fungsi $f(x) = \ln(x)$ terlebih dahulu

$$\begin{aligned} f(x) &= \ln(x) & \rightarrow f(1) &= 0 \\ f'(x) &= 1/x & \rightarrow f'(1) &= 1 \\ f''(x) &= -1/x^2 & \rightarrow f''(1) &= -1 \\ f'''(x) &= 2/x^3 & \rightarrow f'''(1) &= 2 \end{aligned}$$

$$\begin{aligned} f^{(4)}(x) &= -6/x^4 & \rightarrow f^{(4)}(1) &= -6 \\ f^{(5)}(x) &= 24/x^5 & \rightarrow f^{(5)}(c) &= 24/c^5 \end{aligned}$$

Deret Taylornya adalah

$$\ln(x) = (x - 1) - (x - 1)^2/2 + (x - 1)^3/3 - (x - 1)^4/4 + R_4(x)$$

dan

$$\ln(0.9) = -0.1 - (-0.1)^2/2 + (-0.1)^3/3 - (-0.1)^4/4 + R_4(x) = -0.1053583 + R_4(x)$$

juga

$$|R_5(0.9)| < \underset{0.9 < c < 1}{\text{Max}} \left| \frac{24}{c^5} \right| \times \frac{(-0.1)^5}{5!}$$

dan nilai $\underset{0.9 < c < 1}{\text{Max}} |24/c^5|$ di dalam selang $0.9 < c < 1$ adalah pada $c = 0.9$ (dengan mendasari pada fakta bahwa suatu pecahan nilainya semakin membesar bilamana menyebut dibuat lebih kecil), sehingga

$$|R_4(0.9)| < \underset{0.9 < c < 1}{\text{Max}} \left| \frac{24}{0.9^5} \right| \times \frac{(-0.1)^5}{5!} \approx 0.0000034$$

Jadi $\ln(0.9) = -0.1053583$ dengan galat pemotongan lebih kecil dari 0.0000034. ■

Contoh 2.7

Deret Taylor dapat digunakan untuk menghitung integral fungsi yang sulit diintegalkan secara analitik (bahkan, adakalanya tidak mungkin dihitung secara analitik). Hitunglah hampiran nilai $\int_0^1 e^{x^2} dx$ secara numerik, yaitu fungsi $f(x) = e^{x^2}$ dihampiri dengan deret Maclaurin orde 8.

Penyelesaian:

Deret Maclaurin orde 8 dari fungsi $f(x) = e^{x^2}$ adalah

$$e^{x^2} = 1 + x^2 + x^4/2! + x^6/3! + x^8/4! \quad (\text{silakan memeriksanya kembali sebagai latihan})$$

Dengan demikian, maka

$$\begin{aligned} \int_0^1 e^{x^2} dx &\approx \int_0^1 \left(1 + x^2 + \frac{x^4}{2!} + \frac{x^6}{3!} + \frac{x^8}{4!}\right) dx = x + \frac{x^3}{3} + \frac{x^5}{10} + \frac{x^7}{42} + \frac{x^9}{216} \Big|_{x=0}^{x=1} \\ &= 1 + \frac{1}{3} + \frac{1}{10} + \frac{1}{42} + \frac{1}{216} = \frac{397836}{272160} = 1.4617724 \end{aligned}$$

■

2.3.2 Galat Pembulatan

Perhitungan dengan metode numerik hampir selalu menggunakan bilangan riil. Masalah timbul bila komputasi numerik dikerjakan oleh mesin (dalam hal ini komputer) karena semua bilangan riil tidak dapat disajikan secara tepat di dalam komputer. Keterbatasan komputer dalam menyajikan bilangan riil menghasilkan galat yang disebut **galat pembulatan**. Sebagai contoh $1/6 = 0.166666666\ldots$ tidak dapat dinyatakan secara tepat oleh komputer karena digit 6 panjangnya tidak terbatas. Komputer hanya mampu merepresentasikan sejumlah digit (atau bit dalam sistem biner) saja. Bilangan riil yang panjangnya melebihi jumlah digit (bit) yang dapat direpresentasikan oleh komputer dibulatkan ke bilangan terdekat.

Misalnya sebuah komputer hanya dapat merepresentasikan bilangan riil dalam 6 digit angka berarti, maka representasi bilangan $1/6 = 0.166666666\ldots$ di dalam komputer 6-digit tersebut adalah 0.166667. Galat pembulatannya adalah $1/6 - 0.166667 = -0.000000333$. Contoh dalam sistem biner misalnya $1/10 = 0.000110011001100110011\ldots_2$ direpresentasikan di dalam komputer dalam jumlah bit yang terbatas. Teknik yang digunakan untuk pembulatan bilangan riil dijelaskan di dalam upabab 2.5.3.

Kebanyakan komputer digital mempunyai dua buah cara penyajian bilangan riil, yaitu **bilangan titik-tetap** (*fixed point*) dan bilangan **titik-kambang** (*floating point*) [KRE88]. Dalam format bilangan titik-tetap setiap bilangan disajikan dengan jumlah tempat desimal yang tetap, misalnya 62.358, 0.013, 1.000. Sedangkan dalam format bilangan titik-kambang setiap bilangan disajikan dengan jumlah digit *berarti* yang sudah tetap, misalnya

$$0.6238 \times 10^3 \quad 0.1714 \times 10^{-13}$$

atau ditulis juga

$$0.6238E+03 \quad 0.1714E-13$$

Digit-digit berarti di dalam format bilangan titik-kambang disebut juga **angka bera** (*significant figure*). Konsep angka bera dijelaskan berikut ini.

Angka Bera

Konsep angka bera (*significant figure*) atau angka berarti telah dikembangkan secara formal untuk menandakan keandalan suatu nilai numerik. Angka bera adalah angka bermakna, angka penting, atau angka yang dapat digunakan dengan pasti [CHA91].

Contohnya,

43.123	memiliki 5 angka bena (yaitu 4, 3, 1, 2, 3)
0.1764	memiliki 4 angka bena (yaitu 1, 7, 6, 4)
0.0000012	memiliki 2 angka bena (yaitu 1, 2)
278.300	memiliki 6 angka bena (yaitu 2, 7, 8, 3, 0, 0)
270.0090	memiliki 7 angka bena (yaitu 2, 7, 0, 0, 0, 9, 0)
0.0090	memiliki 2 angka bena (yaitu 9, 0)
1360, 1.360, 0.001360	semuanya memiliki 4 angka bena

Perhatikanlah bahwa angka 0 bisa menjadi angka bena atau bukan. Pada contoh 0.001360, tiga buah angka nol pertama tidak berarti, sedangkan 0 yang terakhir angka berarti karena pengukuran dilakukan sampai ketelitian 4 digit. Jumlah angka bena akan terlihat dengan pasti bila bilangan riil itu ditulis dalam penulisan ilmiah (*scientific notation*), misalnya tetapan dalam kimia dan fisika atau ukuran jarak dalam astronomi. Jumlah angka bena terletak pada jumlah digit *mantis*-nya (tentang mantis ini akan dibahas belakangan):

4.3123×10^1	memiliki 5 angka bena
1.764×10^{-1}	memiliki 4 angka bena
1.2×10^{-6}	memiliki 2 angka bena
2.78300×10^2	memiliki 6 angka bena
0.2700090×10^3	memiliki 7 angka bena
9.0×10^{-3}	memiliki 2 angka bena
$13.60 \times 10^2, 0.1360 \times 10^1, 1.360 \times 10^{-3}$	memiliki 4 angka bena
6.02×10^{23}	memiliki 24 angka bena (bilangan <i>Avogadro</i>)
1.5×10^7	memiliki 8 angka bena (jarak bumi-matahari)

Komputer hanya menyimpan sejumlah tertentu angka bena. Bilangan riil yang jumlah angka benanya melebihi jumlah angka bena komputer akan disimpan dalam sejumlah angka bena komputer itu. Pengabaian angka bena sisanya itulah yang menimbulkan galat pembulatan.

2.3.3 Galat Total

Galat akhir atau galat total atau pada solusi numerik merupakan jumlah galat pemotongan dan galat pembulatan. Misalnya pada Contoh 2.3 kita menggunakan deret Maclaurin orde-4 untuk menghampiri $\cos(0.2)$ sebagai berikut:

$$\cos(0.2) \approx 1 - 0.2^2/2 + 0.2^4/24 \approx 0.9800667$$

↑ ↑
 galat galat
 pemotongan pembulatan

Galat pemotongan timbul karena kita menghampiri $\cos(0.2)$ sampai suku orde empat, sedangkan galat pembulatan timbul karena kita membulatkan nilai hampiran ke dalam 7 digit bena.

2.4 Orde Penghampiran

Di dalam metode numerik, fungsi $f(x)$ sering diganti dengan fungsi hampiran yang lebih sederhana. Satu cara mengungkapkan tingkat ketelitian penghampiran itu adalah dengan menggunakan notasi **O-Besar (Big-Oh)**.

Misalkan $f(h)$ dihampiri dengan fungsi $p(h)$. Jika $|f(h) - p(h)| \leq M|h^n|$, yang dalam hal ini M adalah konstanta riil > 0 , maka kita katakan bahwa $p(h)$ menghampiri $f(h)$ dengan orde penghampiran $O(h^n)$ dan kita tulis

$$f(h) = p(h) + O(h^n) \quad (\text{P.2.14})$$

$O(h^n)$ juga dapat diartikan sebagai orde galat dari penghampiran fungsi. Karena h umumnya cukup kecil yaitu lebih kurang dari 1, maka semakin tinggi nilai n , semakin kecil galat, yang berarti semakin teliti penghampiran fungsinya. Metode yang berorde $O(h^2)$, misalnya, lebih teliti hasilnya daripada metode yang berorde $O(h)$. Juga, pada metode yang berorde $O(h^2)$, jika ukuran h dijadikan setengah kali semula, maka galatnya menjadi seperempat kali galat semula.

Umumnya deret Taylor digunakan untuk penghampiran fungsi. Misalkan,

$$x_{i+1} = x_i + h, \quad i = 0, 1, 2, \dots$$

adalah titik-titik selebar h , maka hampiran $f(x_{i+1})$ dengan deret Taylor di sekitar x_i adalah

$$\begin{aligned} f(x_{i+1}) &= f(x_i) + \frac{(x_{i+1} - x_i)}{1!} f'(x_i) + \frac{(x_{i+1} - x_i)^2}{2!} f''(x_i) + \dots + \frac{(x_{i+1} - x_i)^n}{n!} f^{(n)}(x_i) + R_n(x_{i+1}) \\ &= f(x_i) + \frac{h}{1!} f'(x_i) + \frac{h^2}{2!} f''(x_i) + \dots + \frac{h^n}{n!} f^{(n)}(x_i) + R_n(x_{i+1}) \end{aligned}$$

yang dalam hal ini

$$R_n(x_{i+1}) = \frac{h^{n+1}}{(n+1)!} f^{(n+1)}(t) = O(h^{n+1}) \quad , x_i < t < x_{i+1} \quad (\text{P.2.15})$$

Jadi, kita dapat menuliskan

$$f(x_{i+1}) = \sum_{k=0}^n \frac{h^k}{k!} f^{(k)}(x_i) + O(h^{n+1}) \quad (\text{P.2.16})$$

Persamaan (P.2.16) menyatakan bahwa jika fungsi $f(x)$ dihampiri dengan deret Taylor derajat n , maka suku sisanya cukup dinyatakan dengan lambang $O(h^{n+1})$. Sebagai catatan, suku sisa yang digunakan di dalam notasi O -Besar adalah suku yang dimulai dengan perpangkatan h^{n+1} .

Sebagai contoh,

$$\begin{aligned} e^h &= 1 + h + h^2/2! + h^3/3! + h^4/4! + O(h^5) \\ \ln(x+1) &= x - x^2/2 + x^3/3 - x^4/4 + x^5/5 + O(h^6) \\ \sin(h) &= h - h^3/3! + h^5/5! + O(h^7) \quad (\text{bukan } O(h^6), \text{ karena suku orde } 6 = 0) \\ \cos(h) &= 1 - h^2/2! + h^4/4! - h^6/6! + O(h^8) \quad (\text{bukan } O(h^7), \text{ karena suku orde } 7 \\ &\quad = 0) \end{aligned}$$

2.5 Bilangan Titik-Kambang

Untuk memahami galat pembulatan lebih rinci, kita perlu mengerti cara penyimpanan bilangan riil di dalam komputer. Format bilangan riil di dalam komputer berbeda-beda bergantung pada piranti keras dan compiler bahasa pemrogramannya. Bilangan riil di dalam komputer umumnya disajikan dalam format *bilangan titik-kambang*. Bilangan titik-kambang a ditulis sebagai

$$a = \pm m \times B^p = \pm 0.d_1d_2d_3d_4d_5d_6 \dots d_n \times B^p \quad (\text{P.2.17})$$

yang dalam hal ini,

- m = mantisa (riil), $d_1d_2d_3d_4d_5d_6 \dots d_n$ adalah digit atau bit mantisa yang nilainya dari 0 sampai $B - 1$, n adalah panjang digit (bit) mantisa.
- B = basis sistem bilangan yang dipakai (2, 8, 10, 16, dan sebagainya)
- p = pangkat (berupa bilangan bulat), nilainya dari $-P_{\min}$ sampai $+P_{\max}$

Sebagai contoh, bilangan riil 245.7654 dinyatakan sebagai 0.2457654×10^3 dalam format bilangan titik kambang dengan basis 10. Cara penyajian seperti itu serupa dengan cara penulisan ilmiah. Penulisan ilmiah termasuk ke dalam sistem bilangan titik-kambang.

Sistem bilangan yang kita gunakan setiap hari menggunakan basis sepuluh (disebut juga sistem desimal), $B = 10$. Umumnya komputer menggunakan sistem biner ($B = 2$), tapi beberapa komputer menggunakan basis 8 dan 16. Untuk memudahkan pemahaman –juga karena kita lebih terbiasa sehari-hari dengan bilangan desimal– kebanyakan contoh-contoh bilangan titik-kambang di dalam bab ini disajikan dalam sistem desimal.

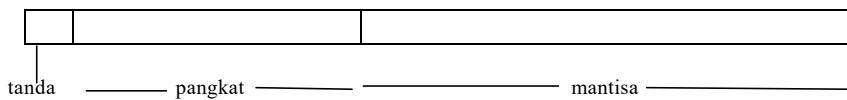
Bilangan titik-kambang di dalam sistem biner biner direpresentasikan oleh komputer dalam bentuk *word* seperti ditunjukkan pada Gambar 2.2. Bit pertama menyatakan tanda (+/-), deretan bit berikutnya menyatakan pangkat bertanda, dan deretan bit terakhir untuk mantisa.

Setiap komputer memiliki panjang *word* yang berbeda-beda. Pada komputer IBM PC, bilangan titik-kambang berketelitian tunggal (*single precision*) disajikan dalam 32 bit yang terdiri atas 1 bit sebagai tanda, 8 bit untuk pangkat dan 23 bit untuk mantisa. Jika dalam bentuk ternormalisasi (akan dijelaskan kemudian), maka bit pertama pada mantisa harus 1, sehingga jumlah bit mantisa efektif adalah 24:

$$a = \pm 0.1b_1b_2b_3b_4b_5b_6 \dots b_{23} \times B^p$$

yang dalam hal ini b menyatakan bit biner (0 atau 1).

Sedangkan pada komputer IBM 370, bilangan titik-kambang berketelitian tunggal disajikan dalam 32 bit yang terdiri dari 1 bit tanda, 7 bit pangkat (basis 16), dan 24 bit mantis (setara dengan 6 sampai 7 digit desimal) [KRE88].



Gambar 2.2 Format bilangan titik-kambang biner (1 word)

2.5.1 Bilangan Titik-kambang Temomatisasi

Representasi bilangan titik-kambang (P.2.17) jauh dari unik, karena, sebagai contoh, kita juga dapat menulisnya sebagai

$$a = \pm (mb) \times B^{p-1} \quad (\text{P.2.18})$$

Misalnya, 245.7654 dapat ditulis sebagai

$$\begin{aligned} & 0.2457654 \times 10^3 \text{ atau} \\ & 2.457654 \times 10^2 \text{ atau} \\ & 0.02457654 \times 10^4, \text{ dan sebagainya} \end{aligned}$$

Agar bilangan titik-kambang dapat disajikan secara seragam, kebanyakan sistem komputer menormalisasikan formatnya sehingga semua digit mantisa selalu angka bena. Karena alasan itu, maka digit pertama mantisa tidak boleh nol. Bilangan titik-kambang yang dinormalisasi ditulis sebagai

$$a = \pm m \times B^p = \pm 0.d_1d_2d_3d_4d_5d_6 \dots d_n \times B^p \quad (\text{P.2.19})$$

yang dalam hal ini, $d_1d_2d_3d_4d_5d_6 \dots d_n$ adalah digit (atau bit) mantisa dengan syarat $1 \leq d_1 \leq b - 1$ dan $0 \leq d_k \leq B-1$ untuk $k > 1$. Pada sistem desimal,

$$1 \leq d_1 \leq 9 \text{ dan } 0 \leq d_k \leq 9,$$

sedangkan pada sistem biner,

$$d_1 = 1 \text{ dan } 0 \leq d_k \leq 1$$

Sebagai contoh, 0.0563×10^{-3} dinormalisasi menjadi 0.563×10^{-4} , 0.00023270×10^6 dinormalisasi menjadi 0.23270×10^3 . Sebagai konsekuensi penormalan, nilai m adalah

$$1/B \leq m < 1$$

Pada sistem desimal ($B = 10$), m akan berkisar dari 0.1 sampai 1, dan pada sistem biner ($B = 10$), antara 0.5 dan 1 [CHA91].

Sebagai catatan, nol adalah kasus khusus. Nol disajikan dengan bagian mantisa seluruhnya nol dan pangkatnya nol. Nol semacam ini tidak dinormalisasi.

Contoh 2.8

[BUC92] Tulislah bilangan e dalam format bilangan titik-kambang ternormalisasi dengan basis 10, basis 2, dan basis 16.

Penyelesaian:

Dalam basis 10 (menggunakan 8 angka bena),

$$e \approx 2.7182818 = 0.27182818 \times 10^1$$

(bilangan titik-kambang desimal ternormalisasi)

Dalam basis 2 (menggunakan 30 bit bena),

$$e \approx 0.10101101111100001010100010110_2 \times 2^2$$

(bilangan titik-kambang biner ternormalisasi)

Dalam basis 16 (gunakan fakta bahwa $16 = 2^4$, sehingga $2^2 = \frac{1}{4} \times 16^1$)

$$\begin{aligned} e &\approx 0.10101101111100001010100010110_2 \times 2^2 \\ &= \frac{1}{4} \times 0.10101101111100001010100010110_2 \times 16^1 \\ &= 0.0010101101111100001010100010110_2 \times 16^1 \\ &= 0.2B7E1516_{16} \times 16^1 \end{aligned}$$

(bilangan titik-kambang heksadesimal ternormalisasi) ■

2.5.2 Epsilon Mesin

Karena jumlah bit yang digunakan untuk representasi bilangan titik-kambang terbatas, maka jumlah bilangan riil yang dapat direpresentasikan juga terbatas. Untuk ilustrasi, tinjau kasus bilangan titik-kambang biner 6-bit *word* (1 bit tanda, 3 bit untuk pangkat bertanda, dan 2 bit mantisa) dengan $B = 2$, dan nilai pangkat dari -2 sampai 3 [GER94]. Karena semua bilangan dinormalisasi, maka bit pertama harus 1, sehingga semua bilangan yang mungkin adalah berbentuk:

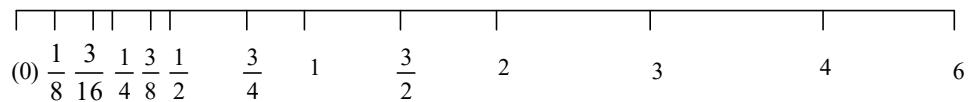
$$\pm 0.10_2 \times 2^p \quad \text{atau} \quad \pm 0.11_2 \times 2^p, \quad -2 \leq p \leq 3$$

Daftar bilangan riil positif yang dapat direpresentasikan adalah

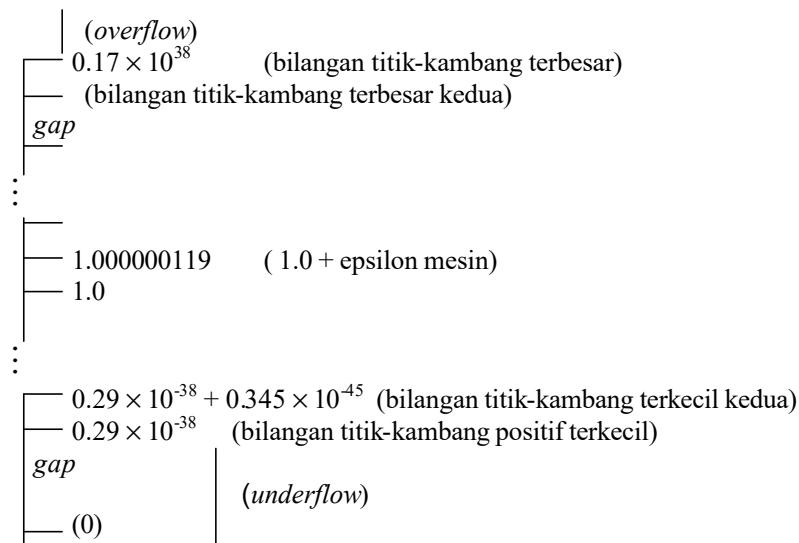
$$\begin{aligned} 0.10_2 \times 2^{-2} &= (1 \times 2^{-1} + 0 \times 2^{-2}) \times 2^{-2} = 1/8 = 0.125_{10} \\ 0.10_2 \times 2^{-1} &= (1 \times 2^{-1} + 0 \times 2^{-2}) \times 2^{-1} = 1/4 = 0.25_{10} \\ 0.10_2 \times 2^0 &= (1 \times 2^{-1} + 0 \times 2^{-2}) \times 2^0 = 1/2 = 0.5_{10} \\ 0.10_2 \times 2^1 &= (1 \times 2^{-1} + 0 \times 2^{-2}) \times 2^1 = 1 = 1.0_{10} \end{aligned}$$

$$\begin{aligned}
 0.10_2 \times 2^2 &= (1 \times 2^{-1} + 0 \times 2^{-2}) \times 2^2 = 2 = 2.0_{10} \\
 0.10_2 \times 2^3 &= (1 \times 2^{-1} + 0 \times 2^{-2}) \times 2^3 = 4 = 4.0_{10} \\
 0.11_2 \times 2^{-2} &= (1 \times 2^{-1} + 1 \times 2^{-2}) \times 2^{-2} = 3/16 = 0.1875_{10} \\
 0.11_2 \times 2^{-1} &= (1 \times 2^{-1} + 1 \times 2^{-2}) \times 2^{-1} = 3/8 = 0.375_{10} \\
 0.11_2 \times 2^0 &= (1 \times 2^{-1} + 1 \times 2^{-2}) \times 2^0 = 3/4 = 0.75_{10} \\
 0.11_2 \times 2^1 &= (1 \times 2^{-1} + 1 \times 2^{-2}) \times 2^1 = 3/2 = 1.5_{10} \\
 0.11_2 \times 2^2 &= (1 \times 2^{-1} + 1 \times 2^{-2}) \times 2^2 = 3 = 3.0_{10} \\
 0.11_2 \times 2^3 &= (1 \times 2^{-1} + 1 \times 2^{-2}) \times 2^3 = 6 = 6.0_{10}
 \end{aligned}$$

Bila kita susun dari nilai positif terkecil ke nilai terbesar, maka seluruh bilangannya digambarkan dalam diagram garis bilangan sebagai berikut:



Pada komputer IBM PC, bilangan titik-kambang berketalitian-tunggal dinyatakan dalam 32-bit *word* (1 bit tanda, 8 bit pangkat, dan 24 bit mantisa). Rentang nilai-nilai positifnya diperlihatkan pada Gambar 2.3.



Gambar 2.3 Rentang bilangan titik-kambang berketalitian tunggal pada *interpreter* Basic

Satu ukuran yang penting di dalam aritmetika komputer adalah seberapa kecil perbedaan antara dua buah nilai yang dapat dikenali oleh komputer. Ukuran yang digunakan untuk membedakan suatu bilangan riil dengan bilangan riil berikutnya adalah **epsilon mesin**. Epsilon mesin distandardisasi dengan menemukan bilangan titik-kambang terkecil yang bila ditambahkan dengan 1 memberikan hasil yang lebih besar dari 1. Dengan kata lain, jika epsilon mesin dilambangkan dengan ϵ maka

$$1 + \epsilon > 1$$

(bilangan yang lebih kecil dari epsilon mesin didefinisikan sebagai nol di dalam komputer).

Epsilon mesin pada sistem bilangan riil yang ditunjukkan pada Gambar 2.3 adalah

$$\epsilon = 1.000000119 - 1.0 = 0.119 \times 10^{-6}$$

Gap (Δx) atau jarak antara sebuah bilangan titik-kambang dengan bilangan titik-kambang berikutnya, yang besarnya adalah

$$\Delta x = \epsilon \times R \quad (\text{P.2.20})$$

yang dalam hal ini R adalah bilangan titik-kambang sekarang. Contohnya, *gap* antara bilangan positif terkecil pertama 0.29×10^{-38} dengan bilangan titik-kambang terkecil kedua pada Gambar 2.3 adalah

$$\Delta x = (0.119 \times 10^{-6}) \times (0.29 \times 10^{-38}) = 0.345 \times 10^{-45}$$

dan dengan demikian bilangan titik-kambang terkecil kedua sesudah 0.29×10^{-38} adalah

$$0.29 \times 10^{-38} + 0.345 \times 10^{-45}$$

Dari persamaan P.2.16 dapat dilihat bahwa *gap* akan bertambah besar dengan semakin besarnya bilangan titik-kambang.

Keadaan *underflow* terjadi bila suatu bilangan titik-kambang tidak dapat dinyatakan di antara 0 dan bilangan positif terkecil (atau antara 0 dan bilangan negatif terbesar). Keadaan *overflow* terjadi bila suatu bilangan titik-kambang lebih besar dari bilangan positif terbesar (atau lebih kecil dari bilangan negatif terkecil)

Jika kita mengetahui jumlah bit mantisa dari suatu bilangan titik-kambang, kita dapat menghitung epsilon mesinnya dengan rumus

$$\epsilon = B^{1-n} \quad (\text{P.2.21})$$

yang dalam hal ini B adalah basis bilangan dan n adalah banyaknya digit (atau bit) bera di dalam mantisa. Pada contoh pertama di atas ($B = 2$ dan $n = 2$),

$$\epsilon = 2^{1-2} = 0.5$$

dan pada contoh bilangan titik-kambang berketelitian tunggal pada komputer IBM PC,

$$\epsilon = 2^{1-24} = 0.00000011920928955078125 = 0.119 \times 10^{-6}$$

Kita juga dapat menemukan perkiraan nilai epsilon mesin dengan prosedur yang sederhana. Gagasananya ialah dengan membagi dua secara terus menerus nilai 1 dan memeriksa apakah 1 ditambah hasil bagi itu lebih besar dari 1. Potongan programnya dituliskan di dalam Program 2.2.

Program 2.2 Program menghitung hampiran nilai epsilon mesin

```

procedure HitungEpsilonMesin1(var eps : real);
{ Prosedur untuk menemukan epsilon mesin
  Keadaan Awal : sembarang
  Keadaan Akhir: eps berisi harga epsilon mesin
}
begin
  eps:=1;
  while eps + 1 > 1 do
    eps:=eps/2;
  {eps + 1 < 1}

  eps:=2*eps;      {nilai epsilon mesin}
end;

```

Hasil pelaksanaan program dengan *compiler* Turbo Pascal dan komputer dengan *processor* 486DX adalah

$$\epsilon = 0.90949470177 \times 10^{-12}$$

Jika yang diinginkan adalah epsilon mesin dalam bentuk perpangkatan dari 2, prosedur untuk menghitungnya dituliskan di dalam Program 2.3.

Program 2.3 Program menghitung hampiran nilai epsilon mesin dalam bentuk 2^k

```
procedure HitungEpsilon_Mesin2(var n : integer);
{ Prosedur untuk menemukan epsilon mesin dalam bentuk perpangkatan 2
  Keadaan Awal : sembarang
  Keadaan Akhir :  $2^{-(n-1)}$  adalah epsilon mesin
}
var
  eps, ne : real;
begin
  eps:=1; ne:=2; n:=0;
  while ne > 1 do
    begin
      eps:=eps/2;
      ne:=1 + eps;
      n:=n + 1;
    end;
  {eps + 1 < 1}
  { Epsilon mesin ialah  $2^{-(n-1)}$  }
end;
```

Hasil pelaksanaan program dengan *compiler* Turbo Pascal dan komputer dengan *processor* 486 adalah

$$\epsilon = 2^{-40} \quad (\text{Keterangan: } 2^{-40} = 0.90949470177 \times 10^{-12})$$

Nilai epsilon mesin yang diperoleh dapat berbeda-beda bergantung pada bahasa pemrograman dan komputer yang digunakan karena beberapa bahasa menggunakan bilangan berketelitian ganda (*double precision*) untuk representasi bilangan titik-kambangnya.

Epsilon dapat digunakan sebagai kriteria berhenti kekonvergenan pada prosedur lelaran yang konvergen. Nilai lelaran sekarang dibandingkan dengan nilai lelaran sebelumnya. Jika selisih keduanya sudah kecil dari epsilon mesin, lelaran dihentikan, tetapi jika tidak, lelaran diteruskan.

2.5.3 Pembulatan pada Bilangan Titik-Kambang

Dari ilustrasi pada upabab 2.5.2 jelaslah bahwa jumlah bilangan riil yang dapat dinyatakan sebagai bilangan titik-kambang terbatas banyaknya (bergantung pada banyaknya bit *word*). Bilangan titik-kambang yang tidak dapat mencocoki satu dari nilai-nilai di dalam rentang pasti terletak di dalam *gap*. Karena itu, bilangan tersebut dibulatkan (atau dikuantisasi) ke salah satu nilai di dalam rentang. Galat yang timbul akibat penghampiran tersebut diacu sebagai galat pembulatan (atau galat kuantisasi). Misalnya, 0.3748 dibulatkan ke 0.375, 3.2 dibulatkan ke 3.0, dan sebagainya.

Ada dua teknik pembulatan yang lazim dipakai oleh komputer, yaitu **pemenggalan** (*chopping*) dan **pembulatan ke digit terdekat** (*in-rounding*). Kedua teknik pembulatan tersebut diilustrasikan di bawah ini.

1. Pemenggalan (*chopping*)

Misalkan a adalah bilangan titik-kambang dalam basis 10:

$$a = \pm 0.d_1d_2d_3\dots d_nd_{n+1}\dots \times 10^p$$

Misalkan n adalah banyak digit mantis komputer. Karena digit mantis a lebih banyak dari digit mantis komputer, maka bilangan a dipotong sampai n digit saja:

$$f_{l_{\text{chop}}}(a) = \pm 0.d_1d_2d_3 \dots d_{n-1}d_n \times 10^p \quad (\text{P.2.21})$$

Sebagai contoh, bilangan $\pi = 3.1459265358\ldots \times 10^0$ di dalam komputer dengan 7 digit mantis disimpan sebagai

$f_{\text{chop}}(\pi) = 0.3141592 \times 10^0$ dengan galat sebesar 0.00000065...

Perhatikan juga bahwa pemenggalan berarti sembarang besaran yang berada pada *gap* sebesar Δx akan disimpan sebagai besaran pada ujung selang yang lebih kecil, sehingga batas atas galat untuk pemenggalan adalah Δx (Gambar 2.4) [CHA91].



Gambar 2.4 Batas atas pemenggalan. Bilangan yang akan dipenggal pada mulanya terletak antara x dan $x + \Delta x$. Setelah pemenggalan, bilangan tersebut = x .

Contoh pemenggalan pada bilangan titik-kambang biner misalnya,

$$1/10 = 0.00011001100110011001100110011 \dots_2$$

Bila digunakan mantis 32 bit, komputer memenggal bilangan $1/10$ di atas menjadi (setelah dinormalkan)

$$1/10 \approx 0.11001100110011001100110011001100, \times 2^{-3}$$

2. Pembulatan ke digit terdekat (*in-rounding*)

Misalkan a adalah bilangan titik-kambang dalam basis 10:

$$a = \pm 0.d_1d_2d_3 \dots d_nd_{n+1} \dots \times 10^p$$

Misalkan n adalah jumlah digit mantis komputer. Karena digit mantis a lebih banyak dari digit mantis komputer, maka bilangan a dibulatkan sampai n digit:

$$fl_{\text{round}}(a) = \pm 0.d_1d_2d_3 \dots \hat{d}_n \times 10^p \quad (\text{P.2.22})$$

yang dalam hal ini

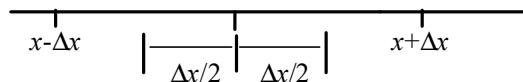
$$\hat{d}_n = \begin{cases} d_n & , \text{jika } d_{n+1} < 5 \\ d_n + 1 & , \text{jika } d_{n+1} > 5 \\ d_n & , \text{jika } d_{n+1} = 5 \text{ dan } n \text{ genap} \\ d_n + 1 & , \text{jika } d_{n+1} = 5 \text{ dan } n \text{ ganjil} \end{cases} \quad (\text{P.2.23})$$

Contohnya, bilangan $\pi = 0.31459265358\dots \times 10^0$ di dalam komputer hipotetis dengan 7 digit mantis dibulatkan menjadi $fl(\pi) = 0.3141593 \times 10^0$ dengan galat sebesar 0.00000035.... Contoh ini memperlihatkan bahwa pembulatan ke digit terdekat menghasilkan galat yang lebih rendah daripada pemenggalan.

Contoh lainnya, nilai $a = 0.5682785715287 \times 10^{-4}$:

- di dalam komputer 7 digit dibulatkan menjadi $fl_{\text{round}}(a) = 0.5682786 \times 10^{-4}$
- di dalam komputer 8 digit dibulatkan menjadi $fl_{\text{round}}(a) = 0.56827857 \times 10^{-4}$
- di dalam komputer 6 digit dibulatkan menjadi $fl_{\text{round}}(a) = 0.568278 \times 10^{-4}$
- di dalam komputer 9 digit dibulatkan menjadi $fl_{\text{round}}(a) = 0.568278572 \times 10^{-4}$

Perhatikan juga bahwa pembulatan ke digit terdekat berarti sembarang besaran yang berada pada *gap* sebesar Δx akan disimpan sebagai bilangan terdekat yang diperbolehkan, sehingga batas atas galat untuk pembulatan adalah $\Delta x/2$ (Gambar 2.5):



Gambar 2.5 Batas atas pembulatan ke digit terdekat

Contoh pembulatan π yang diberikan di atas memperlihatkan bahwa galat pembulatan ke digit terdekat lebih rendah daripada galat pemenggalan, karena itu cara pemenggalan biasanya tidak direkomendasikan untuk dipakai. Namun yang mengherankan, kebanyakan komputer menggunakan cara pemenggalan! Alasannya adalah bahwa algoritma pembulatan ke digit terdekat lebih sukar sehingga membutuhkan waktu lebih lama dari pada waktu untuk pemenggalan. Sedangkan algoritma pemenggalan lebih sederhana sehingga mudah direalisasikan. Pendekatan ini dapat diterima dengan anggapan bahwa jumlah angka benar cukup besar sehingga galat pemenggalan yang dihasilkan biasanya dapat diabaikan [CHA91]. Algoritma pembulatan/pemenggalan dapat diimplementasikan baik di dalam piranti keras atau di dalam rutin perangkat lunak.

Secara umum dapat dinyatakan bahwa satu-satunya cara meminimumkan galat pembulatan adalah menggunakan jumlah angka benar yang lebih banyak. Di dalam program komputer itu artinya kita menggunakan bilangan riil berketelitian ganda (*double precision*) ketimbang bilangan berketelitian tunggal (*single precision*).

2.5.4 Aritmetika Bilangan Titik-Kembang

Selain mengandung galat pembulatan pada representasinya, operasi aritmetika pada bilangan titik-kembang juga menghasilkan galat pembulatan yang lain. Operasi aritmetika pada bilangan titik-kembang meliputi operasi penambahan dan pengurangan, operasi perkalian, dan operasi pembagian.

2.5.4.1 Operasi Penambahan dan Pengurangan

Terdapat dua buah kasus serius yang menyebabkan timbulnya galat pembulatan pada operasi penjumlahan dua buah bilangan titik-kembang:

Kasus 1: Penjumlahan (termasuk pengurangan) bilangan yang sangat kecil ke (atau dari) bilangan yang lebih besar menyebabkan timbulnya galat pembulatan.

Galat pembulatan pada Kasus 1 ini terjadi karena untuk menjumlahkan dua buah bilangan yang berbeda relatif besar, pangkatnya harus disamakan terlebih dahulu (disamakan dengan pangkat bilangan yang lebih besar). Caranya adalah dengan menggeser digit-digit (atau bit) bilangan yang pangkatnya lebih kecil. Pergeseran digit (atau bit) ini mengakibatkan adanya digit (atau bit) yang hilang. Perhatikan contoh berikut.

Contoh 2.9

Misalkan digunakan komputer dengan mantis 4 digit (basis 10). Hitunglah

$$1.557 + 0.04381 = 0.1557 \times 10^1 + 0.4381 \times 10^{-1}$$

Penyelesaian:

$$\begin{aligned} 0.1557 \times 10^1 &= 0.1557 \times 10^1 \\ 0.4381 \times 10^{-1} &= 0.004381 \times 10^1 + (\text{pergeseran digit untuk menyamakan pangkat}) \\ &\underline{\hspace{10em}} \\ &= 0.160081 \times 10^1 \\ &\quad \text{in-rounding} \rightarrow 0.1601 \times 10^1 \\ &\quad \text{chopping} \rightarrow 0.1600 \times 10^1 \end{aligned}$$

Perhatikanlah bahwa dua digit terakhir dari bilangan yang digeser ke kanan pada dasarnya telah hilang dari perhitungan.

$$\text{Galat mutlak pembulatan} = |(0.160081 \times 10^1) - (0.1601 \times 10^1)| = 0.000019$$

$$\text{Galat mutlak pemenggalan} = |(0.160081 \times 10^1) - (0.1600 \times 10^1)| = 0.000081 \blacksquare$$

Galat perhitungan semacam Kasus 1 ini dapat terjadi dalam perhitungan deret tak berhingga yang suku awalnya relatif lebih besar dibandingkan suku berikutnya. Jadi, setelah beberapa suku ditambahkan, kita berada dalam situasi penambahan besaran yang kecil terhadap besaran yang besar. Suatu cara mengurangi galat jenis ini adalah menjumlahkan deret dalam urutan terbalik -yakni dalam urutan yang menaik ketimbang menurun. Dengan cara ini, setiap suku baru akan sebanding besarnya dengan jumlah deret yang terakumulasi [CHA91].

Contoh 2.10

Misalkan digunakan komputer dengan mantis 4 digit (basis 10). Hitunglah

$$3677 - 0.3283 = 0.3677 \times 10^4 - 0.3283 \times 10^0$$

Penyelesaian:

$$\begin{aligned} 0.3677 \times 10^4 &= 0.3677 \times 10^4 \\ 0.3283 \times 10^0 &= 0.00003283 \times 10^4 - (\text{pergeseran digit untuk menyamakan pangkat}) \\ &\underline{\hspace{10em}} \\ &= 0.36766717 \times 10^4 \\ &\quad \text{in-rounding} \rightarrow 0.3677 \times 10^4 \\ &\quad \text{chopping} \rightarrow 0.3676 \times 10^4 \end{aligned}$$

$$\text{Galat mutlak pembulatan} = |(0.36766717 \times 10^4) - (0.3677 \times 10^4)| = 0.00003283$$

$$\text{Galat mutlak pemenggalan} = |(0.36766717 \times 10^4) - (0.3676 \times 10^4)| = 0.00006717 \blacksquare$$

Contoh 2.11

[NAK93] Misalkan digunakan komputer dengan mantis 24 bit (setara dengan 7 tempat desimal). Nilai

$$x = 1.0 + \sum_{i=1}^{10000} 0.00001 = 1.0 + \underbrace{0.00001 + 0.00001 + \dots + 0.00001}_{10000 \text{ kali}}$$

akan dihitung dengan program Pascal yang menggunakan bilangan riil berketelitian-tunggal sebagai berikut:

```
program hitung;
var
    x, jumlah : real;
    i: integer;
begin
    jumlah:=1.0;
    for i:=1 to 10000 do
        jumlah:=jumlah + 0.00001;
    writeln('x=', jumlah);
end.
```

Bila digunakan komputer dengan 7 angka bena, hasil program di atas adalah $x = 1.100136$, padahal seharusnya $x = 1.0 + 10000(0.00001) = 1.0 + 0.1 = 1.1$.

$$\text{Galat mutlak} = |1.1 - 1.100136| = 0.000136$$

$$\text{Galat relatif} = \frac{1.1 - 1.100136}{1.1} \times 100\% = -0.0124$$

Bagaimana kita menjelaskan hasil perhitungan ini? Sumber penyimpangan ini adalah galat pembulatan. Karena bilangan 1 dapat dinyatakan secara persis dalam komputer, maka galatnya nol. Sebaliknya 0.00001 tidak dapat dinyatakan secara tepat dan dikuantisasi oleh nilai yang sedikit berbeda dibandingkan nilai sejatinya. Penyimpangan yang kecil itu dapat diabaikan untuk komputasi yang sedikit, namun terakumulasi setelah penjumlahan yang berulang kali [CHA91]. Untuk jelasnya perhatikan proses perhitungannya dalam sistem biner di bawah ini.

$$\begin{aligned} 1_{10} &= 0.100000000000000000000000_2 \times 2^1 \\ (0.00001)_{10} &= 0.10100111100010110101100_2 \times 2^{-16} \\ &= 0.000000000000000010100111100010110101100_2 \times 2^1 + \\ (1 + 0.00001)_{10} &= 0.100000000000000010100111100010110101100_2 \times 2^1 \\ | & \qquad \qquad \qquad \text{dibulatkan ke mantis 24 bit} \\ (1.00001)_{10} & \qquad \qquad \qquad | \\ (\text{nilai sejati}) & \qquad \qquad \qquad | \\ 0.10000000000000001010100_2 \times 2 &= (1.0000100136)_{10} \\ & \qquad \qquad \qquad (\text{nilai hampiran}) \end{aligned}$$

$$\begin{aligned} \text{galat mutlak} &= |1.00001 - 1.0000100136| = 0.136 \times 10^{-7} \\ \text{galat total} &\approx 10000 \times 0.136 \cdot 10^{-7} = 0.136 \times 10^{-3} = 0.000136 \end{aligned}$$

Satu cara untuk mengurangi galat total ini adalah menjumlahkan suku-suku dalam urutan terbalik, yaitu 0.00001 dijumlahkan terlebih dahulu sebanyak seribu kali, baru kemudian hasilnya dijumlahkan dengan 1.0. Jadi cara perhitungannya adalah

$$x = \left(\sum_{i=1}^{10000} 0.00001 \right) + 1.0 = \underbrace{0.00001 + 0.00001 + \dots + 0.00001}_{10000 \text{ kali}} + 1.0$$

Selain itu, gunakan bilangan berketelitian ganda, sebab galat pembulatannya jauh lebih kecil. ■

Kasus 2: Pengurangan dua buah bilangan yang hampir sama besar (*nearly equal*). Bila dua bilangan titik-kambang dikurangkan, hasilnya mungkin mengandung nol pada posisi digit mantis yang paling berarti (posisi digit paling kiri). Keadaan ini dinamakan **kehilangan angka bena** (*loss of significance*). Baik pemenggalan maupun pembulatan ke digit terdekat menghasilkan jawaban yang sama.

Contoh 2.12

Kurangi 0.56780×10^5 dengan 0.56430×10^5 (5 angka bena)

Penyelesaian:

$$\begin{array}{r} 0.56780 \times 10^5 \\ 0.56430 \times 10^5 - \\ \hline 0.00350 \times 10^5 \end{array} \rightarrow \text{dinormalisasi menjadi } 0.350 \times 10^3 \text{ (3 angka bena)}$$

$$\begin{array}{l} \textit{in-rounding} \rightarrow 0.350 \times 10^3 \\ \textit{chopping} \rightarrow 0.350 \times 10^3 \end{array}$$

Hasil yang diperoleh hanya mempunyai 3 angka bena. Jadi kita kehilangan 2 buah angka bena. Meskipun kita dapat menuliskan hasilnya sebagai 0.35000×10^3 , namun dua nol yang terakhir bukan angka bena tetapi sengaja ditambahkan untuk mengisi kekosongan digit yang hilang. ■

Hasil yang lebih dramatis diperlihatkan pada Contoh 2.13 dan Contoh 2.14 di bawah ini.

Contoh 2.13

Kurangi 3.1415926536 dengan 3.1415957341 (11 angka bena).

Penyelesaian:

$$\begin{array}{r} 3.1415926536 = 0.31415926536 \times 10^1 \\ 3.1415957341 = 0.31415957341 \times 10^1 - \\ \hline -0.30805 \times 10^{-5} \quad (5 \text{ angka bena}) \end{array}$$

$$\begin{array}{l} \textit{in-rounding} \rightarrow -0.30805 \times 10^{-5} \\ \textit{chopping} \rightarrow -0.30805 \times 10^{-5} \end{array}$$

Jadi, kita kehilangan 6 angka bena!. ■

Contoh 2.14

Kurangi 0.7642×10^3 dengan 0.7641×10^3 (4 angka bena).

Penyelesaian:

$$\begin{array}{r} 0.7642 \times 10^3 \\ 0.7641 \times 10^3 - \\ \hline 0.0001 \times 10^3 = 0.1 \times 10^0 \quad (1 \text{ angka bena}) \end{array}$$

$$\begin{array}{l} \textit{in-rounding} \rightarrow 0.1 \times 10^0 \\ \textit{chopping} \rightarrow 0.1 \times 10^0 \end{array}$$

Jadi kita kehilangan 3 buah angka bena. ■

Kehilangan angka bena bila mengurangkan dua buah bilangan yang hampir sama besar merupakan sumber galat utama pada operasi bilangan titik-kambang. Kehilangan angka bena dapat dihindari dengan mengubah metode komputasi yang digunakan. Tujuan dari pengubahan metode komputasi adalah menghilangkan operasi pengurangan dua buah bilangan yang hampir sama besar, misalnya dengan pengelompokan suku-suku, perkalian dengan bentuk sekawan, menggunakan deret Taylor, atau manipulasi aljabar lainnya. Contoh 2.15 sampai 2.17 berikut mengilustrasikan cara pengubahan ini.

Contoh 2.15

[MAT92] Diberikan $f(x) = x(\sqrt{x+1} - \sqrt{x})$. Hitunglah $f(500)$ dengan menggunakan 6 angka bena dan pembulatan ke digit terdekat.

Penyelesaian:

$$\begin{aligned}
f(500) &= 500(\sqrt{501} - \sqrt{500}) \\
&= 500(22.3830 - 22.3607) \\
&= 500(0.0223) \\
&= 11.15 \text{ (empat angka bena)} \\
&\text{(solusi sejatinya adalah } 11.174755300747198..\text{)}
\end{aligned}$$

Hasil yang tidak akurat ini disebabkan adanya operasi pengurangan dua bilangan yang hampir sama besar, yaitu 22.3830 - 22.3607. Ketelitian hasil dapat kita tingkatkan bila kita dapat menghilangkan pengurangan tersebut. Caranya adalah mengubah metode komputasi sedemikian sehingga pengurangan dua bilangan yang hampir sama besar menjadi hilang.

Susunlah kembali fungsi $f(x)$ menjadi bentuk yang lebih baik:

$$\begin{aligned}
f(x) &= x(\sqrt{x+1} - \sqrt{x}) \\
&= x(\sqrt{x+1} - \sqrt{x}) \frac{(\sqrt{x+1} + \sqrt{x})}{(\sqrt{x+1} + \sqrt{x})} \\
&= \frac{x[(\sqrt{x+1})^2 - (\sqrt{x})^2]}{(\sqrt{x+1} + \sqrt{x})} \\
&= \frac{x}{\sqrt{x+1} + \sqrt{x}} = p(x)
\end{aligned}$$

sehingga

$$p(500) = \frac{500}{\sqrt{501} + \sqrt{500}} = \frac{500}{22.3830 + 22.3607} = 11.1748$$

Hasil ini jauh lebih baik dibandingkan yang pertama. Solusi sejatinya, 11.174755300747198..., jika dibulatkan sampai 6 angka bena adalah 11.1747, yang lebih dekat ke $p(500)$ daripada ke $f(500)$. ■

Contoh 2.16

Hitunglah akar-akar polinom $x^2 - 40x + 2 = 0$ sampai 4 angka bena.

Penyelesaian:

$$x_{1,2} = \frac{40 \pm \sqrt{(-40)^2 - 8}}{2} = 20 \pm \sqrt{398} = 20.00 \pm 19.95$$

$$x_1 = 20 + 19.95 = 39.95 \text{ (4 angka bena)}$$

$$x_2 = 20 - 19.95 = 0.05 \text{ (1 angka bena)}$$

(kehilangan tiga buah angka bena akibat pengurangan dua buah bilangan yang hampir sama, yaitu $20 - 19.95$)

Nilai x_2 yang lebih akurat dapat diperoleh dengan mengingat lagi pelajaran matematika di sekolah lanjutan bahwa:

jika x_1 dan x_2 adalah akar-akar persamaan $ax^2 + bx + c = 0$ maka $x_1 x_2 = c/a$

Dengan demikian, x_2 dihitung sebagai berikut:

$$39.95x_2 = 2/1 \Rightarrow x_2 = 2.000/39.95 = 0.05006 \text{ (4 angka bena, lebih akurat)} \blacksquare$$

Contoh 2.17

[MAT92] Diberikan $f(x) = \frac{e^x - 1 - x}{x^2}$. Hitung $f(0.01)$ sampai 6 angka bena.

Penyelesaian:

$$f(0.01) = \frac{e^{0.01} - 1 - 0.01}{0.01^2} = \frac{1.01005 - 1 - 0.01}{0.0001} = 0.5 \text{ (1 angka bena)}$$

Hasil yang tidak akurat ini karena adanya kehilangan angka bena akibat pengurangan dua buah nilai yang hampir sama besar, yaitu $1.01005 - 1$. Hasil yang lebih akurat dapat diperoleh dengan menguraikan $f(x)$ ke dalam deret Maclaurin sampai suku orde 2 (Gunakan dalil L'Hospital bila menjumpai pembagian 0/0):

$$f(x) \approx p(x) = 1/2 + x/6 + x^2/24$$

sehingga

$$p(0.01) = 1/2 + 0.01/6 + 0.01^2/24 = 0.5 + 0.001667 + 0.00005 = 0.501671$$

Solusi sejatinya adalah $0.50167084168057542\dots$, yang jika dibulatkan sampai 6 angka bena adalah $0.501671 = p(0.01)$. \blacksquare

2.5.4.2 Operasi Perkalian dan Pembagian

Operasi perkalian dan pembagian dua buah bilangan titik-kambang tidak memerlukan penyamaan pangkat seperti halnya pada penjumlahan. Perkalian dapat dilakukan dengan mengalikan kedua mantis dan menambahkan kedua pangkatnya. Pembagian dikerjakan dengan membagi mantis dan mengurangkan pangkatnya. Selain itu, register dengan panjang ganda dibutuhkan untuk menyimpan hasil antara. Hasil akhir dipotong ke dalam register tunggal.

Contoh 2.18

Hitung perkalian 0.4652×10^4 dengan 0.1456×10^{-1} (4 angka bena).

Penyelesaian:

Kalikan mantis:	0.4652	Jumlahkan pangkat:	4
	$\begin{array}{r} 0.1456 \\ \times \end{array}$		$\begin{array}{r} -1 \\ + \\ \hline 3 \end{array}$
	$\overline{0.06773312}$		

Gabungkan mantis dengan pangkat: 0.06773312×10^3

Normalisasi: 0.6773312×10^2

in-rounding $\rightarrow 0.6773 \times 10^2$

chopping $\rightarrow 0.6773 \times 10^2$

■

Contoh 2.19

Hitung $(0.8675 \times 10^{-5})/0.2543 \times 10^{-2}$ (4 angka bena).

Penyelesaian:

Bagi mantis:	0.8675	Kurangi pangkat:	-4
	$\begin{array}{r} 0.2543 \\ : \end{array}$		$\begin{array}{r} -2 \\ - \\ \hline -2 \end{array}$
	$\overline{3.4113252}$		

Gabungkan mantis dengan pangkat: 3.4113252×10^{-2}

Normalisasi: $0.34113252 \times 10^{-1}$

in-rounding $\rightarrow 0.3411 \times 10^{-1}$

chopping $\rightarrow 0.3411 \times 10^{-1}$

■

2.6 Perambatan Galat

Galat yang dikandung dalam bilangan titik-kambang merambat pada hasil komputasi. Misalkan terdapat dua bilangan a dan b (nilai sejati) dan nilai hampirannya masing-masing \hat{a} dan \hat{b} , yang mengandung galat masing-masing ε_a dan ε_b . Jadi, kita dapat menulis

$$a = \hat{a} + \varepsilon_a$$

dan

$$b = \hat{b} + \varepsilon_b$$

Di bawah ini akan diperlihatkan bagaimana galat merambat pada hasil penjumlahan dan perkalian a dan b .

Untuk penjumlahan,

$$a + b = (\hat{a} + \varepsilon_a) + (\hat{b} + \varepsilon_b) = (\hat{a} + \hat{b}) + (\varepsilon_a + \varepsilon_b) \quad (\text{P.2.24})$$

Jadi, galat hasil penjumlahan sama dengan jumlah galat masing-masing *operand*.

Untuk perkalian,

$$ab = (\hat{a} + \varepsilon_a)(\hat{b} + \varepsilon_b) = \hat{a}\hat{b} + \hat{a}\varepsilon_b + \hat{b}\varepsilon_a + \varepsilon_a\varepsilon_b$$

yang bila kita susun menjadi

$$ab - \hat{a}\hat{b} = \hat{a}\varepsilon_b + \hat{b}\varepsilon_a + \varepsilon_a\varepsilon_b$$

Dengan mengandaikan bahwa $a \neq 0$ dan $b \neq 0$, maka galat relatifnya adalah

$$\begin{aligned} \frac{ab - \hat{a}\hat{b}}{ab} &= \frac{\hat{a}\varepsilon_b + \hat{b}\varepsilon_a + \varepsilon_a\varepsilon_b}{ab} \\ &= \frac{\hat{a}\varepsilon_b}{ab} + \frac{\hat{b}\varepsilon_a}{ab} + \frac{\varepsilon_a\varepsilon_b}{ab} \end{aligned}$$

Dengan mengandaikan bahwa a dan \hat{a} hampir sama besar, yaitu $a \approx \hat{a}$, begitu juga b dan \hat{b} , dan ε_a dan ε_b sangat kecil maka $\hat{a}/a \approx 1$, $\hat{b}/b \approx 1$, dan $(\varepsilon_a/a)(\varepsilon_b/b) \approx 0$. Dengan demikian

$$\frac{ab - \hat{a}\hat{b}}{ab} = \frac{\varepsilon_b}{b} + \frac{\varepsilon_a}{a} = \varepsilon_{Rb} + \varepsilon_{Ra} \quad (\text{P.2.25})$$

Jadi, galat relatif hasil perkalian sama dengan jumlah galat relatif masing-masing *operand*.

Jika operasi aritmetika hanya dilakukan sekali saja, maka kita tidak perlu terlalu khawatir terhadap galat yang ditimbulkannya. Namun, bila operasi dilakukan terhadap seruntunan komputasi, maka galat operasi aritmetika awal akan merambat dalam seruntunan komputasi. Bila hasil perhitungan sebuah operasi aritmetika dipakai untuk operasi selanjutnya, maka akan terjadi penumpukan galat yang semakin besar, yang mungkin mengakibatkan hasil perhitungan akhir menyimpang dari hasil sebenarnya. Ketidakpastian hasil akibat galat pembulatan

yang bertambah besar itu dapat menyebabkan perhitungan menjadi **tidak stabil** (*unstable* atau *instability*), sedangkan lawannya adalah **stabil**, yang merupakan proses numerik yang diinginkan. Metode komputasi dikatakan stabil jika galat pada hasil antara (*intermediate*) hanya sedikit pengaruhnya pada hasil akhir. Jika galat pada hasil antara memberikan pengaruh yang besar pada hasil akhir maka metode komputasinya dikatakan tidak stabil [KRE88]. Ketidakstabilan ini dinamakan "ketidakstabilan numerik", yang dapat dihindari dengan memilih metode komputasi yang stabil (di dalam Bab Solusi Persamaan Diferensial Biasa masalah ini akan dikemukakan lagi). Ketidakstabilan numerik harus dibedakan dengan "ketidakstabilan matematik" dari persoalan yang diberikan. Ketidakstabilan matematik sering dinamakan **kondisi buruk** (*ill conditioned*), yaitu kondisi yang timbul karena hasil perhitungan sangat peka terhadap perubahan kecil data. Kondisi buruk didiskusikan lebih komprehensif di bawah ini.

2.7 Kondisi Buruk

Suatu persoalan dikatakan **berkondisi buruk** (*ill conditioned*) bila jawabannya sangat peka terhadap perubahan kecil data (misalnya perubahan kecil akibat pembulatan). Bila kita mengubah sedikit data, maka jawabannya berubah sangat besar (drastis). Lawan dari berkondisi buruk adalah **berkondisi baik** (*well conditioned*). Suatu persoalan dikatakan berkondisi baik bila perubahan kecil data hanya mengakibatkan perubahan kecil pada jawabannya.

Sebagai contoh, tinjau persoalan menghitung akar persamaan kuadrat $ax^2 + bx + c = 0$ di bawah ini. Di sini kita hanya mengubah nilai tetapan c -nya saja:

$$(i) \quad x^2 - 4x + 3.999 = 0 \Rightarrow \text{akar-akarnya } x_1 = 2.032 \text{ dan } x_2 = 1.968$$

Sekarang, ubah 3.99 menjadi 4.00:

$$(ii) \quad x^2 - 4x + 4.000 = 0 \Rightarrow \text{akar-akarnya } x_1 = x_2 = 2.000$$

Ubah 4.00 menjadi 4.001:

$$(iii) \quad x^2 - 4x + 4.001 = 0 \Rightarrow \text{akar-akarnya imajiner}$$

Kita katakan bahwa persoalan akar-akar persamaan kuadrat di atas berkondisi buruk, karena dengan pengubahan sedikit saja data masukannya (dalam hal ini nilai koefisien c), ternyata nilai akar-akarnya berubah sangat besar.

Kapankah akar persamaan $f(x) = 0$ berkondisi buruk? Misalkan $f(x)$ diubah sebesar ϵ sehingga akarnya berubah sebesar h :

$$f(x + h) + \epsilon = 0 \quad (\text{P.2.26})$$

Bila karena pengubahan ϵ yang sangat kecil mengakibatkan h menjadi besar, dikatakan persoalan mencari akar $f(x) = 0$ berkondisi buruk [NOB72].

Bukti:

Kita menggunakan teorema nilai rata-rata (TNR) di dalam kalkulus diferensial:

$$\frac{f(m) - f(n)}{m - n} = f'(t) \quad , \quad m < t < n$$

Karena $m - n = h \Rightarrow m = n + h$, maka

$$\frac{f(m+h) - f(n)}{h} = f'(t)$$

atau

$$f(p+h) = f(p) + h f'(t) \quad (\text{P.2.27})$$

Terapkan (P.2.27) pada (P.2.26):

$$f(x) + h f'(t) + \epsilon = 0 \quad , \quad x < t < x + h \quad (\text{P.2.28})$$

Pada persoalan pencarian akar, $f(x) = 0$, sehingga

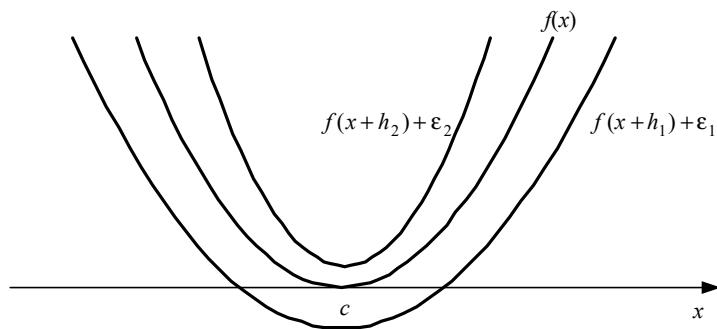
$$h f'(t) + \epsilon = 0 \quad (\text{P.2.29})$$

atau

$$h = \frac{-\epsilon}{f'(t)} \quad (\text{P.2.30})$$

Terlihat dari P.2.30 bahwa akar-akar $f(x) = 0$ akan berkondisi buruk bila $f'(t)$ bernilai sangat kecil. Bila $f'(t)$ sangat kecil nilainya maka h menjadi sangat besar yang berarti akar x bertambah sebesar h tersebut. ■

Masalah seperti ini sering muncul pada pencarian akar kembar (seperti pada contoh di atas). Nilai $f'(x)$ di sekitar akar kembar sangat kecil (mendekati 0), yang berakibat nilai h menjadi sangat besar (lihat Gambar 2.6).



Gambar 2.6 Kondisi buruk pada pencarian akar kembar

Contoh persoalan yang berkondisi buruk lainnya adalah persoalan mencari solusi sistem persamaan lanjut berupa titik potong dari dua buah garis lurus yang berbentuk $ax + by = c$ yang diubah-ubah koefisiennya (b dan c):

$$\begin{aligned} \text{(i)} \quad & x + y = 2 \\ & x + 0.9999y = 1.9999 \\ & \text{Solusi: } x = y = 1.0000 \end{aligned}$$

$$\begin{aligned} \text{(ii)} \quad & x + y = 2 \\ & x + 0.9999y = 2.0010 \\ & \text{Solusi: } x = 12, y = -10 \end{aligned}$$

$$\begin{aligned} \text{(iii)} \quad & x + y = 2 \\ & x + y = 1.9999 \\ & \text{Solusi: tidak ada} \end{aligned}$$

$$\begin{aligned} \text{(iv)} \quad & x + y = 2 \\ & x + y = 2 \\ & \text{Solusi: tidak berhingga, yaitu disepanjang garis } x + y = 2 \end{aligned}$$

Kondisi buruk yang terjadi pada perhitungan titik potong dua buah garis lurus dapat kita analisis sebagai berikut. Sistem persamaan lanjar di atas dapat dituliskan sebagai

$$\begin{array}{l} x + y = 2 \\ x + (1 + \varepsilon)y = 2 + \delta \end{array}$$

yang dalam hal ini ε dan δ dalam orde 10^{-4} .

Solusi sistem persamaan lanjar tersebut adalah

$$\begin{array}{rcl} x + y &= 2 \\ x + (1 + \varepsilon)y &= 2 + \delta & - \\ \hline -\varepsilon y &= -\delta & \Rightarrow y = \delta / \varepsilon \quad \text{dan } x = 2 - \delta / \varepsilon, \varepsilon \neq 0 \end{array}$$

Nilai ε dan δ sangat penting. Perubahan kecil ε mempunyai pengaruh yang besar pada solusi, yang berarti persoalan mencari solusi sistem persamaan lanjar berkondisi buruk.

2.8 Bilangan kondisi

Kondisi komputasi numerik dapat diukur dengan **bilangan kondisi**. Bilangan kondisi merupakan ukuran tingkat sejauh mana ketidakpastian dalam x diperbesar oleh $f(x)$ [CHA88]. Bilangan kondisi dapat dihitung dengan bantuan deret Taylor. Fungsi $f(x)$ diuraikan di sekitar \hat{x} sampai suku orde pertama:

$$f(x) \approx f(\hat{x}) + f'(\hat{x})(x - \hat{x}) \quad (\text{P.2.31})$$

Galat relatif hampiran dari $f(x)$ adalah

$$\varepsilon_{RA}[f(x)] = \frac{f(x) - f(\hat{x})}{f(\hat{x})} \approx \frac{f'(\hat{x})(x - \hat{x})}{f(\hat{x})} \quad (\text{P.2.32})$$

dan galat relatif hampiran dari x adalah

$$\varepsilon_{RA}[x] = \frac{x - \hat{x}}{\hat{x}} \quad (\text{P.2.33})$$

Bilangan kondisi didefinisikan sebagai nisbah (*ratio*) antara P.2.32 dan P.2.33:

$$\text{Bilangan kondisi} = \left| \frac{\epsilon_{RA}[f(x)]}{\epsilon_{RA}[x]} \right| = \left| \frac{\hat{x}f'(\hat{x})}{f(\hat{x})} \right| \quad (\text{P.2.34})$$

Arti dari bilangan kondisi adalah:

- bilangan kondisi = 1 berarti galat relatif hampiran fungsi sama dengan galat relatif x
- bilangan kondisi lebih besar dari 1 berarti galat relatif hampiran fungsi besar
- bilangan kondisi lebih kecil dari 1 berarti galat relatif hampiran fungsi kecil (kondisi baik)

Suatu komputasi dikatakan berkondisi buruk jika bilangan kondisinya sangat besar, sebaliknya berkondisi baik bila bilangan kondisinya sangat kecil.

Contoh 2.20

Misalkan $f(x) = \sqrt{x}$. Tentukan bilangan kondisi perhitungan akar kuadrat x .

Penyelesaian:

Hitung $f'(x)$ terlebih dahulu

$$f'(x) = \frac{1}{2\sqrt{x}}$$

yang akan digunakan untuk menghitung

$$\text{bilangan kondisi} = \left| \frac{\hat{x}/(2\sqrt{\hat{x}})}{\sqrt{\hat{x}}} \right| = \frac{1}{2}$$

Bilangan kondisi ini sangat kecil, yang berarti penarikan akar kuadrat x merupakan proses yang berkondisi baik. Sebagai contoh, $\sqrt{20.999} = 4.5824665$, dan jika 20.999 diubah sedikit (dibulatkan) menjadi 21.000 maka $\sqrt{21.000} = 4.5825756$. Ternyata perubahan kecil pada nilai x hanya berakibat perubahan sedikit pada $f(x)$. ■

Contoh 2.21

Hitung bilangan kondisi $f(x) = \frac{10}{1-x^2}$.

Penyelesaian:

Hitung $f'(x)$ terlebih dahulu

$$f'(x) = \frac{20x}{(1-x^2)^2}$$

yang digunakan untuk menghitung

$$\text{bilangan kondisi} = \left| \frac{\hat{x}[20\hat{x}/(1-\hat{x}^2)^2]}{1/(1-\hat{x}^2)} \right| = \left| \frac{2\hat{x}^2}{1-\hat{x}^2} \right|$$

Bilangan kondisi ini sangat besar untuk $|x| \approx 1$. Jadi, menghitung $f(x)$ untuk x mendekati 1 atau -1 sangat buruk keadaannya, karena galat relatifnya besar. Sebagai contoh, $f(1.009) = -55.306675$, tetapi $f(1.01) = -497.51243$. Ternyata perubahan kecil pada nilai x di sekitar 1 (karena dibulatkan dari 4 angka bena menjadi 3 angka bena), mengakibatkan nilai $f(x)$ berubah sangat besar. Untuk x yang jauh dari 1 atau -1, $f(x)$ berkondisi baik. ■

Contoh 2.22

[CHA91] Hitung bilangan kondisi untuk $f(x) = \tan(x)$.

Penyelesaian:

Hitung $f'(x)$ terlebih dahulu

$$f'(x) = \frac{1}{\cos^2(x)}$$

yang digunakan untuk menghitung

$$\text{bilangan kondisi} = \left| \frac{\hat{x}[1/\cos^2(\hat{x})]}{\tan(\hat{x})} \right|$$

Bilangan kondisi ini sangat besar untuk $x \approx \pi/2$. Misalkan untuk $x = \pi/2 + 0.1(\pi/2)$,

$$\text{bilangan kondisi} = 1.7279(40.86)/-6.314 = -11.2$$

dan untuk $x = \pi/2 + 0.01(\pi/2)$,

$$\text{bilangan kondisi} = 1.5865(4053)/-63.66 = -101$$

■

Orang yang bijaksana belajar dari kesalahan orang lain, hanya orang yang bodohlah yang belajar dari kesalahannya sendiri.
(Pepatah Rusia)

Soal Latihan

1. Tentukan hampiran fungsi di bawah ini ke dalam deret Taylor:
 - (a) $\ln(x)$ sampai orde-4 di sekitar $x_0=1$, lalu hampiri nilai $\ln(0.9)$.
 - (b) $f(x) = e^x - 1$ sampai orde-3 di sekitar $x_0=0$, Lalu hitung nilai $f(0.0001)$ sampai empat angka bena.
 - (c) $\sinh(x) = \frac{1}{2} (e^x - e^{-x})$ di sekitar $x_0 = 0$, lalu hitung nilai hampiran $\int_0^1 \sinh(x) dx$
 - (d) $\sin(x)$ sampai orde-3, lalu tentukan batas atas galat $\sin(x)$ jika $0 \leq x \leq 0.5$.
2. (a) Tentukan polinom Maclarin orde 4 untuk $f(x)$, kemudian gunakan polinom tersebut untuk menghampiri nilai $f(0.23)$, serta tentukan batas atas galatnya.
 - (i) $f(x) = \sin(2x)$
 - (ii) $f(x) = \ln(1+x)$
(b) Cari polinom Taylor orde 3 pada $x_0=1$ untuk $f(x) = x^3 - 2x^2 + 3x + 5$ dan perlihatkan bahwa ia mewakili $f(x)$ secara tepat
(c) Hitunglah
$$\int_0^1 \sin(2x) dx$$

 - (i) secara analitis (solusi sejati)
 - (ii) secara numerik, yang dalam hal ini $\sin(2x)$ dihampiri dengan deret Maclarin yang telah anda dapatkan pada jawaban 2(a)(i) di atas. Hitung galat mutlak dan galat relatif hasilnya. Pakailah enam angka bena untuk, baik untuk setiap hasil maupun hasil akhir.
3. Hitung $\sqrt[3]{10.1} - \sqrt[3]{10}$ secara langsung tetapi hasil setiap perhitungan antara dan hasil akhir dibulatkan sampai empat angka bena. Kemudian, hitunglah $\sqrt[3]{10.1} - \sqrt[3]{10}$ dengan cara yang lebih baik.
4. Carilah akar persamaan kuadrat $x^2 - 10.1x + 1 = 0$ dengan rumus abc, yang setiap hasil perhitungan antara maupun hasil perhitungan akhir dibulatkan dengan teknik:
 - (a) pembulatan ke dalam (*in-rounding*)

(b) pemenggalan (*chopping*)

sampai empat angka bena. Bandingkan hasilnya jika akar terbesar (x_1) dihitung dengan rumus abc dan akar terkecil (x_2) dengan rumus $x_1x_2 = c/a$

5. Diberikan beberapa bilangan titik-kambang yang telah dinormalkan sebagai berikut:

$$a = 0.4523123 \times 10^{-4}$$

$$b = 0.2365401 \times 10^1$$

$$c = 0.4520156 \times 10^{-4}$$

$$d = 0.1234567 \times 10^{-3}$$

Bila mesin yang digunakan untuk operasi aritmetika mempunyai tujuh angka bena, hitung hasil komputasi yang diberikan oleh mesin tersebut (dalam bentuk bilangan titik-kambang ternormalisasi):

- (i) $a + b + c + d$
(ii) $a + c + d + b$
(iii) $a - c$
(iv) $ab - c$
6. Misalkan digunakan mesin hipotetik dengan mantis empat angka bena. Lakukan operasi aritmetika untuk bilangan titik-kambang ternormalisasi berikut. Normalkan hasilnya.

(a) $0.3796 \times 10^2 + 0.9643 \times 10^{-2}$

(b) $0.4561 \times 10^{-2} - 0.6732 \times 10^{-2}$

(c) $0.1234 \times 10^3 \times 0.4321 \times 10^{-1}$

7. Carilah cara yang lebih baik untuk menghitung:

- (i) $f(x) = (x - \sin(x))/\tan(x)$ untuk x mendekati nol
(ii) $f(x) = x - \sqrt{x^2 - a}$ untuk x yang jauh lebih besar dari a
(iii) $f(x) = \cos^2(x) - \sin^2(x)$ untuk x di sekitar $\pi/4$
(iv) $f(x) = \log(x+1) - \log(x)$ untuk x yang besar
(v) $(1 + \alpha)^{1/2} - 1$, $|\alpha| \leq 0.01$ sampai enam angka bena
(vi) $\sin(\alpha + x) - \sin(\alpha)$ untuk x yang kecil
(vii) $(a + x)^n - a^n$ untuk x yang kecil
(viii) $((x^3 - 3x^2) + 3x) - 1$ untuk $x = 2.72$
(ix) $\sqrt[4]{1 + \cos x}/2$ untuk $x \approx \pi/4$

8. Bagaimana cara menghitung

$$\frac{3\alpha}{4} - \sin(\alpha) + \frac{\sin(2\alpha)}{8}$$

sampai 6 angka bena untuk $|\alpha| \leq 0.2$? (Petunjuk : gunakan deret Maclaurin)

9. Diketahui $f(x) = \cos(x)$. Tentukan $f'(1)$ dengan teorema dasar turunan:

$$\lim_{h \rightarrow 0} \frac{f(x+h) - f(x)}{h} = f'(x)$$

Hitung $f'(1)$ dengan bermacam-macam $h = 0.1, 0.01, 0.001, 0.0001, 0.00001, 0.000001$. Untuk memperbaiki hasil perhitungan, hitunglah $f'(1)$ dengan cara yang lebih baik.

10. (a) Hitunglah dengan lima angka bena nilai $f(13.400)$ bila

$$f(x) = x - 1000(\sqrt{x+0.1} - \sqrt{x})$$

(b) Perbaiki hasil perhitungan anda dengan mengubah metode komputasi.

11. Tentukan bilangan kondisi fungsi-fungsi berikut ini dan tentukan apakah fungsi tersebut berkondisi baik atau berkondisi buruk. Jika berkondisi baik, tentukan di x berapakah fungsi ini berkondisi buruk (berikan contoh nilai x untuk memperjelas jawaban anda)

- (a) $f(x) = 1/(1-x)$
- (b) $f(x) = x^2$
- (c) $f(x) = x^n$
- (d) $f(x) = \sin(x)$
- (e) $f(x) = \log(x)$

12. Uraikan $f(x) = \cos(x)$ di sekitar $x = \pi/4$. Hitunglah $f(\pi/3)$ sampai galat relatif hampiran kurang dari 0.5%. (Petunjuk: hitung suku demi suku, setiap kali menambahkan dengan jumlah suku yang lama, hitung galat relatif hampiran)

Esensi dari matematika adalah kebebasannya
(George Cantor)

Bab 3

Solusi Persamaan Nirlanjar

Saya tidak tahu bagaimana saya tampak pada dunia; tetapi bagi saya sendiri
saya nampaknya hanyalah seperti seorang anak laki-laki yang bermain-main
di pantai, dan mengalihkan diri sendiri sekarang dan kemudian menemukan
koral yang lebih halus atau kerang yang lebih indah daripada yang biasa,
sementara samudera besar kebenaran semuanya terbentang di hadapan saya
tak terungkapkan.
(Isaac Newton)

Dalam bidang sains dan rekayasa, para ahli ilmu alam dan rekayasawan sering berhadapan dengan persoalan mencari solusi persamaan –lazim disebut **akar persamaan** (*roots of equation*) atau **nilai-nilai nol**– yang berbentuk $f(x) = 0$. Beberapa persamaan sederhana mudah ditemukan akarnya. Misalnya $2x - 3 = 0$, pemecahannya adalah dengan memindahkan -3 ke ruas kanan sehingga menjadi $2x = 3$, dengan demikian solusi atau akarnya adalah $x = 3/2$. Begitu juga persamaan kuadratik seperti $x^2 - 4x - 5 = 0$, akar-akarnya mudah ditemukan dengan cara pemfaktoran menjadi $(x - 5)(x + 1) = 0$ sehingga $x_1 = 5$ dan $x_2 = -1$.

Umumnya persamaan yang akan dipecahkan muncul dalam bentuk nirlanjar (*non linear*) yang melibatkan bentuk sinus, cosinus, eksponensial, logaritma, dan fungsi transenden lainnya. Misalnya,

1. Tentukan akar riil terkecil dari

$$9.34 - 21.97x + 16.3x^3 - 3.704x^5 = 0$$

2. Kecepatan ke atas sebuah roket dapat dihitung dengan memakai rumus berikut:

$$v = u \ln \left| \frac{m_0}{m_0 - qt} \right| - gt$$

yang dalam hal ini v adalah kecepatan ke atas, u adalah kecepatan pada saat bahan bakar dikeluarkan relatif terhadap roket, m_0 massa awal roket pada saat $t = 0$, q laju pemakaian bahan bakar, dan g percepatan gravitasi ($= 9.8 \text{ m/det}^2$). Jika $u = 2200 \text{ m/det}$, $m_0 = 160000 \text{ kg}$, dan $q = 2680 \text{ kg/det}$, hitunglah waktu saat $v = 1000 \text{ m/det}$. (Nyatakan persamaan dengan ruas kanan sama dengan 0:

$$v - u \ln[m_0/(m_0 - qt)] + gt = 0 \quad)$$

3. Dalam teknik kelautan, persamaan gelombang berdiri yang dipantulkan oleh dermaga pelabuhan diberikan oleh

$$h = h_0 \{ \sin(2\pi x/\lambda) \cos(2\pi tv/\lambda) + e^{-x} \}$$

Tentukan x jika $h = 0.5h_0$, $\lambda = 20$, $t = 10$ dan $v = 50$!

(Nyatakan persamaan dengan ruas kanan sama dengan 0:

$$h - h_0 \{ \sin(2\pi x/\lambda) \cos(2\pi tv/\lambda) + e^{-x} \} = 0 \quad)$$

4. Suatu arus osilasi dalam rangkaian listrik diberikan oleh

$$I = 10e^{-t} \sin(2\pi t)$$

yang dalam hal ini t dalam detik. Tentukan semua nilai t sedemikian sehingga $I = 2$ ampere.

(Nyatakan persamaan dengan ruas kanan sama dengan 0:

$$I - 10e^{-t} \sin(2\pi t) = 0 \quad)$$

5. Dalam bidang teknik lingkungan, persamaan berikut ini dapat digunakan untuk menghitung tingkat oksigen pada hilir sungai dari tempat pembuangan limbah:

$$c = 10 - 15(e^{-0.1x} - e^{-0.5x})$$

yang dalam hal ini x adalah jarak hilir sungai ke tempat pembuangan limbah. Tentukan jarak hilir sungai tersebut bila pembacaan pertama pada alat pengukur tingkat oksigen adalah 4 bila pengukur berada 5 mil dari pembuangan.

(Nyatakan persamaan dengan ruas kanan sama dengan 0:

$$c - 10 - 15(e^{-0.1x} - e^{-0.5x}) = 0 \\)$$

6. Reaksi kesetimbangan



dapat dicirikan oleh hubungan setimbang

$$K = \frac{[C]}{[A]^2 [B]}$$

yang dalam hal ini [...] menyatakan konsentrasi zat kimia. Andaikan bahwa kita mendefenisikan peubah x sebagai jumlah mol C yang dihasilkan. Hukum kekekalan massa dapat dipakai untuk merumuskan ulang hubungan keseimbangan itu sebagai

$$K = \frac{[C_0] + x}{([A_0] - 2x)([B_0] - x)}$$

yang dalam hal ini indeks 0 menunjukkan konsentrasi awal tiap unsur. Jika diketahui tetapan kesetimbangan $K = 1.25 \times 10^{-2}$, dan konsentrasi larutan $[A_0] = 50$, $[B_0] = 40$, dan $[C_0] = 5$, hitunglah x .

(Nyatakan persamaan dengan ruas kanan sama dengan 0:

$$K - \frac{[C_0] + x}{([A_0] - 2x)([B_0] - x)} = 0 \quad)$$

Keenam contoh di atas memperlihatkan bentuk persamaan yang rumit/kompleks yang tidak dapat dipecahkan secara analitik (seperti persamaan kuadratik pada paragraf awal). Bila metode analitik tidak dapat menyelesaikan persamaan, maka kita masih bisa mencari solusinya dengan menggunakan metode numerik.

3.1 Rumusan Masalah

Persoalan mencari solusi persamaan nirlanjar dapat dirumuskan secara singkat sebagai berikut: tentukan nilai x yang memenuhi persamaan

$$f(x) = 0 \tag{P.3.1}$$

yaitu nilai $x = s$ sedemikian sehingga $f(s)$ sama dengan nol.

3.2 Metode Pencarian Akar

Dalam metode numerik, pencarian akar $f(x) = 0$ dilakukan secara lelaran (iteratif). Sampai saat ini sudah banyak ditemukan metode pencarian akar. Secara umum, semua metode pencarian akar tersebut dapat dikelompokkan menjadi dua golongan besar:

1. **Metode tertutup** atau metode pengurung (*bracketing method*)

Metode yang termasuk ke dalam golongan ini mencari akar di dalam selang $[a, b]$. Selang $[a, b]$ sudah dipastikan berisi minimal satu buah akar, karena itu metode jenis ini selalu berhasil menemukan akar. Dengan kata lain, lelarannya selalu konvergen (menuju) ke akar, karena itu metode tertutup kadang-kadang dinamakan juga **metode konvergen**.

2. **Metode terbuka**

Berbeda dengan metode tertutup, metode terbuka tidak memerlukan selang $[a, b]$ yang mengandung akar. Yang diperlukan adalah tebakan (*guess*) awal akar, lalu, dengan prosedur lelaran, kita menggunakan untuk menghitung hampiran akar yang baru. Pada setiap kali lelaran, hampiran akar yang lama dipakai untuk menghitung hampiran akar yang baru. Mungkin saja hampiran akar yang baru mendekati akar sejati (konvergen), atau mungkin juga menjauhinya (divergen). Karena itu, metode terbuka tidak selalu berhasil menemukan akar, kadang-kadang konvergen, kadangkala ia divergen.

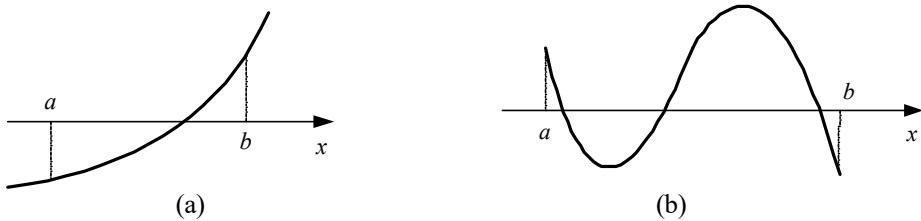
3.3 Metode Tertutup

Seperti yang telah dijelaskan, metode tertutup memerlukan selang $[a,b]$ yang mengandung akar. Sebagaimana namanya, selang tersebut “mengurung” akar sejati. Tata-ancang (*strategy*) yang dipakai adalah mengurangi lebar selang secara sistematis sehingga lebar selang tersebut semakin sempit, dan karenanya menuju akar yang benar.

Dalam sebuah selang mungkin terdapat lebih dari satu buah akar atau tidak ada akar sama sekali. Secara grafik dapat ditunjukkan bahwa jika:

$$(1) f(a)f(b) < 0$$

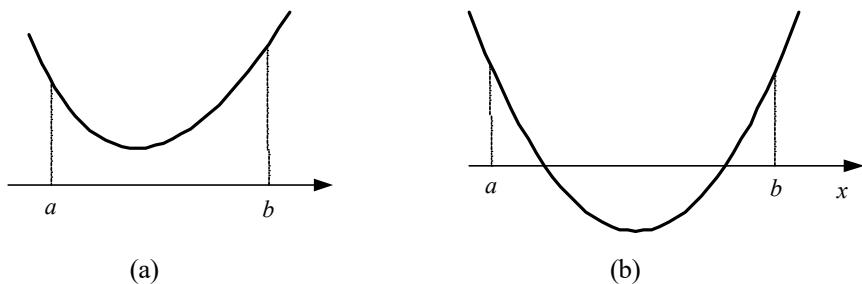
maka terdapat akar sebanyak bilangan ganjil (Gambar 3.1).



Gambar 3.1 Banyaknya akar ganjil

(2) $f(a)f(b) > 0$

maka terdapat akar sebanyak bilangan genap atau tidak ada akar sama sekali (Gambar 3.2).



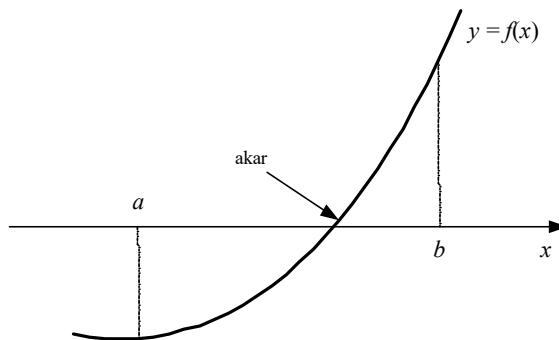
Gambar 3.2 Banyaknya akar genap

Syarat Cukup Keberadaan Akar

Gambar 3.1 memperlihatkan bahwa selalu ada akar di dalam selang $[a, b]$ jika nilai fungsi berbeda tanda ($+/ -$) di $x = a$ dan $x = b$. Tidak demikian halnya jika nilai fungsi di ujung-ujung selang sama tandanya, yang mengisyaratkan mungkin ada akar atau tidak ada sama sekali. Jadi, jika nilai fungsi berbeda tanda tanda di ujung-ujung selang, pastilah terdapat paling sedikit satu buah akar di dalam selang tersebut. Dengan kata lain, syarat cukup keberadaan akar persamaan kita tulis sebagai berikut:

Jika $f(a)f(b) < 0$ dan $f(x)$ menerus di dalam selang $[a, b]$, maka
paling sedikit terdapat satu buah akar persamaan $f(x) = 0$ di dalam selang $[a, b]$.

Syarat ini disebut syarat cukup¹ -bukan syarat perlu- sebab meskipun nilai-nilai di ujung selang tidak berbeda tanda, mungkin saja terdapat akar di dalam selang tersebut (seperti ditunjukkan pada Gambar 3.2). Syarat cukup keberadaan akar ini ditunjukkan pada Gambar 3.3.



Gambar 3.3 Lokasi akar

Ada dua masalah yang terjadi karena ketidaktepatan mengambil selang $[a, b]$. Masalah pertama adalah bila di dalam selang $[a, b]$ terdapat lebih dari satu buah akar. Sekali suatu metode tertutup digunakan untuk mencari akar di dalam selang $[a, b]$, ia hanya menemukan sebuah akar saja. Karena itu, bila kita mengambil selang $[a, b]$ yang mengandung lebih dari satu akar, hanya satu buah akar saja yang berhasil ditemukan (lihat kembali Gambar 3.1(b)).

Masalah kedua adalah bila mengambil selang $[a, b]$ yang tidak memenuhi syarat cukup. Adakalanya kita dapat “kehilangan” akar karena selang $[a, b]$ yang diambil ternyata tidak memenuhi syarat cukup $f(a)f(b) < 0$. Sehingga, kita mungkin sampai pada kesimpulan tidak terdapat akar di dalam selang $[a, b]$ tersebut, padahal seharusnya ada (lihat kembali Gambar 3.2 (b)).

Untuk mengatasi kedua masalah di atas, pengguna metode tertutup disarankan mengambil selang yang berukuran cukup kecil yang memuat hanya satu akar. Ada dua pendekatan yang dapat kita gunakan dalam memilih selang tersebut.

¹ Bentuk implikasi “jika p maka q ” bisa dibaca sebagai “ p adalah syarat cukup untuk q ”. Di dalam kalkulus proposisi, pernyataan “jika p maka q ” (dilambangkan dengan $p \rightarrow q$) adalah benar kecuali jika p benar dan q salah. Jadi, pernyataan tersebut tetap benar meskipun $f(a)f(b) > 0$ dan di dalam selang $[a, b]$ terdapat paling sedikit satu buah akar atau tidak terdapat akar sama sekali. Pernyataan tersebut jelas salah bila $f(a)f(b) > 0$ dan di dalam selang $[a, b]$ terdapat paling sedikit satu buah akar (tidak mungkin).

Pendekatan pertama adalah membuat grafik fungsi di bidang X - Y , lalu melihat di mana perpotongannya dengan sumbu- X . Dari sini kita dapat mengira-ngira selang yang memuat titik potong tersebut. Grafik fungsi dapat dibuat dengan program yang ditulis sendiri, atau lebih praktis menggunakan paket program yang dapat membuat grafik fungsi.

Pendekatan yang kedua adalah dengan mencetak nilai fungsi pada titik-titik absis yang berjarak tetap. Jarak titik ini dapat diatur cukup kecil. Jika tanda fungsi berubah pada sebuah selang, pasti terdapat minimal satu akar di dalamnya. Program 3.1 berisi prosedur untuk menemukan selang yang cukup kecil yang mengandung akar. Program ini mencetak tabel titik-titik sepanjang selang $[a, b]$. Dari tabel tersebut kita dapat menentukan upaselang yang nilai fungsi di ujung-ujungnya berbeda tanda. Keberhasilan dari pendekatan ini bergantung pada jarak antara titik-titik absis. Semakin kecil jarak titik absis, semakin besar peluang menemukan selang yang mengandung hanya sebuah akar.

Program 3.1 Menemukan selang kecil yang mengandung akar

```

procedure Cari_SelangKecilYangMengandungAkar(a, b, h: real);
{ Menentukan dan mencetak nilai-nilai fungsi untuk absis x di dalam
selang [a, b]. Jarak antara tiap absis adalah h.
K.Awal: a dan b adalah ujung-ujung selang, nilainya sudah terdefenisil;
h adalah jarak antara tiap absis x
K.Akhir: tabel yang berisi x dan f(x) dicetak ke layar
}
var
  x : real;
begin
  x:=a;
  writeln('-----');
  writeln('      x          f(x)    ');
  writeln('-----');
  while x <= b do begin
    writeln(x:5:2, f(x):10:6);
    x:=x+h;
  end;
  { x > b }
  writeln('-----');
end;

```

Bila Program 3.1 digunakan untuk mencari selang kecil yang mengandung akar pada fungsi $f(x) = e^x - 5x^2$ mulai dari $a = -0.5$ sampai $b = 1.4$ dengan kenaikan absis sebesar $h = 0.1$, maka hasilnya tampak pada tabel berikut:

x	$f(x)$
-0.50	-0.643469
-0.40	-0.129680
-0.30	0.290818
-0.20	0.618731
-0.10	0.854837
0.00	1.000000
0.10	1.055171
0.20	1.021403
0.30	0.899859
0.40	0.691825
0.50	0.398721
0.60	0.022119
0.70	-0.436247
0.80	-0.974459
0.90	-1.590397
1.00	-2.281718
1.10	-3.045834
1.20	-3.879883
1.30	-4.780703
1.40	-5.744800

Berdasarkan tabel di atas, selang yang cukup kecil yang mengandung akar adalah

$$[-0.40, -0.30] \text{ dan } [0.60, 0.70]$$

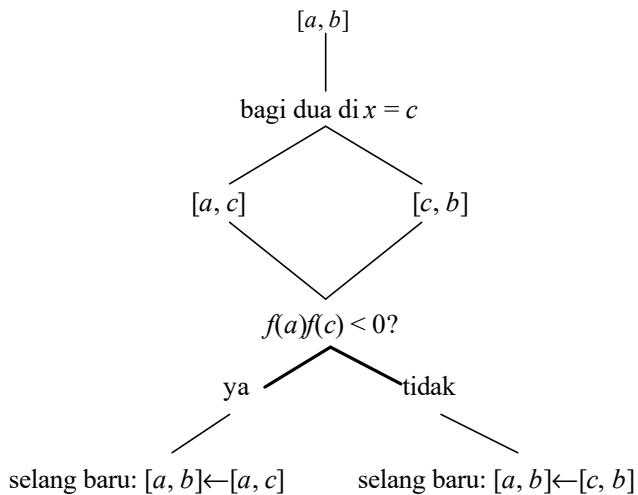
karena nilai fungsi berubah tanda di ujung-ujung selangnya. Selang $[0.00, 1.00]$ juga dapat kita ambil tetapi cukup lebar, demikian juga $[-0.50, 1.40]$, $[-0.30, 0.80]$, dan seterusnya.

Ada dua metode klasik yang termasuk ke dalam metode tertutup, yaitu **metode bagidua** dan **metode regula-falsi**. Masing-masing metode kita bahas lebih rinci di bawah ini.

3.3.1 Metode Bagidua²

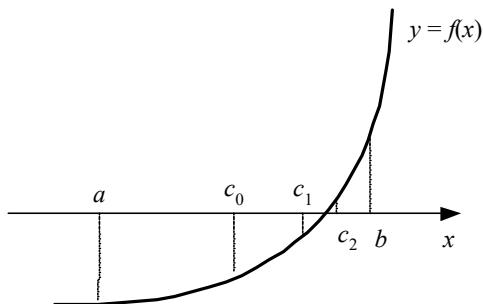
Misalkan kita telah menentukan selang $[a, b]$ sehingga $f(a)f(b) < 0$. Pada setiap kali lelaran, selang $[a, b]$ kita bagi dua di $x = c$, sehingga terdapat dua buah upaselang yang berukuran sama, yaitu selang $[a, c]$ dan $[c, b]$. Selang yang diambil untuk lelaran berikutnya adalah upaselang yang memuat akar, bergantung pada apakah $f(a)f(c) < 0$ atau $f(c)f(b) < 0$.

²Nama lainnya adalah *metode Bolzano*



Selang yang baru dibagi dua lagi dengan cara yang sama. Begitu seterusnya sampai ukuran selang yang baru sudah sangat kecil (lihat Gambar 3.4). Kondisi berhenti lelaran dapat dipilih salah satu dari tiga kriteria berikut:

1. Lebar selang baru: $|a - b| < \varepsilon$, yang dalam hal ini ε adalah nilai toleransi lebar selang yang mengurung akar.
2. Nilai fungsi di hampiran akar: $f(c) = 0$. Beberapa bahasa pemrograman membolehkan pembandingan dua buah bilangan riil, sehingga perbandingan $f(c) = 0$ dibenarkan. Namun kalau kita kembali ke konsep awal bahwa dua buah bilangan riil tidak dapat dibandingkan kesamaannya karena representasinya di dalam mesin tidak tepat, maka kita dapat menggunakan bilangan yang sangat kecil (misalnya epsilon mesin) sebagai pengganti nilai 0. Dengan demikian, menguji kesamaan $f(c) = 0$ dapat kita hampiri dengan $f(c) < \text{epsilon_mesin}$.
3. Galat relatif hampiran akar: $|c_{\text{baru}} - c_{\text{lama}}|/c_{\text{baru}} < \delta$, yang dalam hal ini δ adalah galat relatif hampiran yang diinginkan.



Gambar 3.4 Proses pembagian selang $[a, b]$ dengan metode bagidua

Program 3.2 berisi algoritma metode bagidua. Di dalam algoritma tersebut, format penulisan keluaran tidak dituliskan untuk menghindari kerumitan algoritma dari hal-hal yang tidak esensial.

Program 3.2 Metode bagidua

```

procedure BagiDua(a,b: real);
{ Mencari akar  $f(x)=0$  di dalam selang  $[a,b]$  dengan metode bagidua
K.Awal :  $a$  dan  $b$  adalah ujung-ujung selang sehingga  $f(a)*f(b) < 0$ ,
         nilai  $a$  dan  $b$  sudah terdefinisi.
K.Akhir : Hampiran akar tercetak di layar.
}
const
  epsilon1 = 0.000001;           {batas lebar selang akhir lelaran}
  epsilon2 = 0.00000001;         {bilangan yang sangat kecil, mendekati nol}
begin
  repeat
    c:=(a+b)/2;      { titik tengah  $[a,b]$ }
    if f(a)*f(c) < 0 then
      b:=c           {selang baru  $[a,b]=[a,c]$ }
    else
      a:=c;          {selang baru  $[a,b]=[c,b]$ }
    until (ABS(a-b)< epsilon1) or (f(c)) < epsilon2;
    { c adalah akar persamaan }
    writeln('Hampiran kar = ', x:10:6);
end;

```

Kasus yang Mungkin Terjadi pada Penggunaan Metode Bagidua

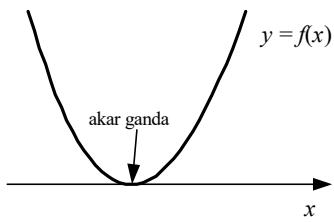
1. Jumlah akar lebih dari satu

Bila dalam selang $[a, b]$ terdapat lebih dari satu akar (banyaknya akar ganjil), hanya satu buah akar yang dapat ditemukan (lihat kembali Gambar 3.1(b)). Cara mengatasinya: gunakan selang $[a,b]$ yang cukup kecil yang memuat hanya satu buah akar.

2. Akar ganda.

Metode bagidua tidak berhasil menemukan akar ganda. Hal ini disebabkan karena tidak terdapat perbedaan tanda di ujung-ujung selang yang baru (Gambar 3.5).

Contoh: $f(x) = (x - 3)^2 = (x - 3)(x - 3)$, mempunyai dua akar yang sama, yaitu $x = 3$.

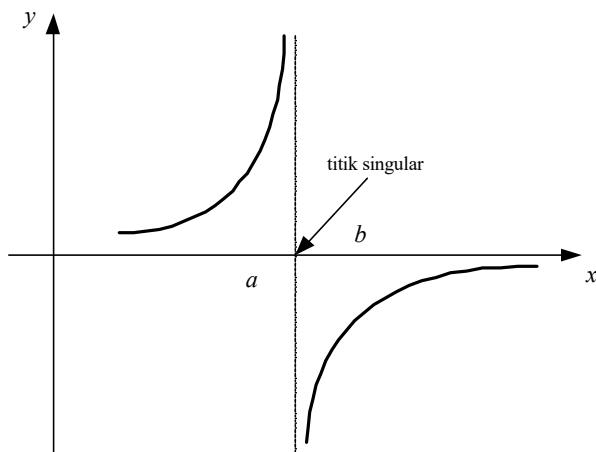


Gambar 3.4 Akar ganda

Cara mengatasinya: akan dibahas pada upabab 3.5.

3. Singularitas.

Pada titik singular, nilai fungsinya tidak terdefinisi. Bila selang $[a, b]$ mengandung titik singular, lelaran metode bagidua tidak pernah berhenti. Penyebabnya, metode bagidua menganggap titik singular sebagai akar karena lelaran cenderung konvergen. Yang sebenarnya, titik singular bukanlah akar, melainkan *akar semu* (Gambar 3.6)



Gambar 3.6 Fungsi singular

Cara mengatasinya: periksa nilai $|f(b) - f(a)|$. Jika $|f(b) - f(a)|$ konvergen ke nol, akar yang dicari pasti akar sejati, tetapi jika $|f(b) - f(a)|$ divergen, akar yang dicari merupakan titik singular (akar semu).

Pada setiap lelaran pada metode bagidua, kita mencatat bahwa selisih antara akar sejati dengan akar hampiran tidak pernah melebihi setengah panjang selang saat itu. Pernyataan ini dinyatakan dengan teorema berikut.

TEOREMA 3.1. Jika $f(x)$ menerus di dalam selang $[a,b]$ dengan $f(a)f(b) < 0$ dan $s \in [a, b]$ sehingga $f(s) = 0$ dan $c_r = (a_r + b_r)/2$, maka selalu berlaku dua ketidaksamaan berikut:

$$(i) |s - c_r| \leq |b_r - a_r|/2$$

dan

$$(ii) |s - c_r| \leq \frac{|b - a|}{2^{r+1}}, \quad r = 0, 1, 2, \dots$$

Bukti:

Misalkan pada lelaran ke- r kita mendapatkan selang $[a_r, b_r]$ yang panjangnya setengah panjang selang sebelumnya, $[a_{r-1}, b_{r-1}]$.

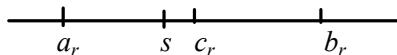
Jadi,

$$|b_r - a_r| = |b_{r-1} - a_{r-1}|/2$$

Jelaslah bahwa

$$\begin{aligned} |b_1 - a_1| &= |b_0 - a_0|/2 = |b - a|/2 \\ |b_2 - a_2| &= |b_1 - a_1|/2 = |b - a|/2^2 \\ |b_3 - a_3| &= |b_2 - a_2|/2 = |b - a|/2^3 \\ &\dots \\ |b_r - a_r| &= |b_{r-1} - a_{r-1}|/2 = |b - a|/2^r \end{aligned}$$

Pada lelaran ke- r , posisi c_r (akar hampiran) dan s (akar sejati) adalah seperti diagram berikut:



Berdasarkan diagram di atas jelaslah bahwa

$$|s - c_r| \leq \frac{|b_r - a_r|}{2}$$

Selanjutnya,

$$|s - c_r| \leq \frac{|b_r - a_r|}{2} = \frac{1}{2} \frac{|b - a|}{2^r} = \frac{|b - a|}{2^{r+1}}$$

■

Jadi, selisih antara akar sejati dengan akar hampiran tidak pernah lebih dari setengah epsilon.

Dengan mengingat kriteria berhenti adalah $|b_r - a_r| < \varepsilon$, maka dari (i) terlihat bahwa

$$|s - c_r| < a / 2$$

sehingga

$$\frac{|b-a|}{2^{r+1}} < \frac{\varepsilon}{2}$$

$$\begin{aligned} &\Leftrightarrow 2^r > |b-a|/\varepsilon \\ &\Leftrightarrow r \ln(2) > \ln(|b-a|) - \ln(\varepsilon) \quad \text{ket: ln adalah logaritma natural} \\ &\Leftrightarrow r > \frac{\ln(|b-a|) - \ln(\varepsilon)}{\ln(2)} \\ &\Leftrightarrow R > \frac{\ln(|b-a|) - \ln(\varepsilon)}{\ln(2)} \end{aligned}$$

yang dalam hal ini R adalah jumlah lelaran (jumlah pembagian selang) yang dibutuhkan untuk menjamin bahwa c adalah hampiran akar yang memiliki galat kurang dari ε .

Contoh 3.1

Temukan akar $f(x) = e^x - 5x^2$ di dalam selang $[0, 1]$ dan $\varepsilon = 0.00001$.

Penyelesaian: Tabel lelaran menggunakan metode bagidua:

r	a	c	b	$f(a)$	$f(c)$	$f(b)$	Selang baru	Lebarnya
0	0.000000	0.500000	1.000000	1.000000	0.398721	-2.281718	[c, b]	0.500000
1	0.500000	0.750000	1.000000	0.398721	-0.695500	-2.281718	[a, c]	0.250000
2	0.500000	0.625000	0.750000	0.398721	-0.084879	-0.695500	[a, c]	0.125000
3	0.500000	0.562500	0.625000	0.398721	0.173023	-0.084879	[c, b]	0.062500
4	0.562500	0.593750	0.625000	0.173023	0.048071	-0.084879	[c, b]	0.031250
5	0.593750	0.609375	0.625000	0.048071	-0.017408	-0.084879	[a, c]	0.015625
6	0.593750	0.601563	0.609375	0.048071	0.015581	-0.017408	[c, b]	0.007813
7	0.601563	0.605469	0.609375	0.015581	-0.000851	-0.017408	[a, c]	0.003906
8	0.601563	0.603516	0.605469	0.015581	0.007380	-0.000851	[c, b]	0.001953
9	0.603516	0.604492	0.605469	0.007380	0.003268	-0.000851	[c, b]	0.000977

10	0.604492	0.604980	0.605469	0.003268	0.001210	-0.000851	[c, b]	0.000488
11	0.604980	0.605225	0.605469	0.001210	0.000179	-0.000851	[c, b]	0.000244
12	0.605225	0.605347	0.605469	0.000179	-0.000336	-0.000851	[a, c]	0.000122
13	0.605225	0.605286	0.605347	0.000179	-0.000078	-0.000336	[a, c]	0.000061
14	0.605225	0.605255	0.605286	0.000179	0.000051	-0.000078	[c, b]	0.000031
15	0.605255	0.605270	0.605286	0.000051	-0.000014	-0.000078	[a, c]	0.000015
16	0.605255	0.605263	0.605270	0.000051	0.000018	-0.000014	[c, b]	0.000008

Jadi, hampiran akarnya adalah $x = 0.605263$

Jumlah lelaran yang dibutuhkan

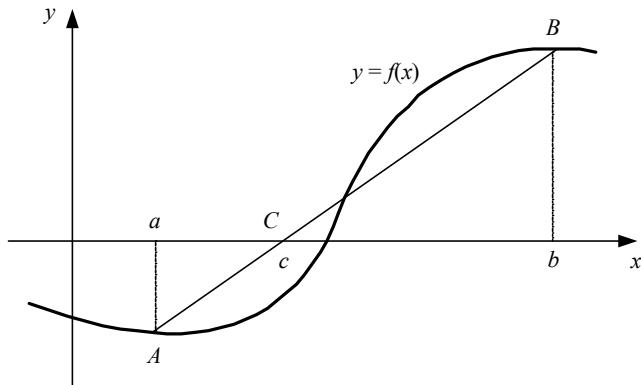
$$R > \frac{\ln(1 - 0) - \ln(0.00001)}{\ln(2)}$$

$$> 16.60964$$

Jadi, dibutuhkan minimal 17 kali lelaran ($r=0$ sampai dengan $r=16$), sesuai dengan jumlah lelaran pada tabel, agar galat akar hampiran kurang dari ϵ . ■

3.3.2 Metode Regula-Falsi

Meskipun metode bagidua selalu berhasil menemukan akar, tetapi kecepatan konvergensi sangat lambat. Kecepatan konvergensi dapat ditingkatkan bila nilai $f(a)$ dan $f(b)$ juga turut diperhitungkan. Logikanya, bila $f(a)$ lebih dekat ke nol daripada $f(b)$ tentu akar lebih dekat ke $x = a$ daripada ke $x = b$. Metode yang memanfaatkan nilai $f(a)$ dan $f(b)$ ini adalah **metode regula-falsi** (bahasa Latin) atau **metode posisi palsu**. (*false position method*). Dengan metode regula-falsi, dibuat garis lurus yang menghubungkan titik $(a, f(a))$ dan $(b, f(b))$. Perpotongan garis tersebut dengan sumbu- x merupakan taksiran akar yang diperbaiki. Garis lurus tadi seolah-olah berlaku menggantikan kurva $y = f(x)$ dan memberikan posisi palsu dari akar.



Gambar 3.7 Metode regula-falsi

Perhatikan Gambar 3.7:

gradien garis AB = gradien garis BC

$$\frac{f(b) - f(a)}{b - a} = \frac{f(b) - 0}{b - c}$$

yang dapat disederhanakan menjadi

$$c = b - \frac{f(b)(b-a)}{f(b)-f(a)} \quad (\text{P.3.2})$$

Algoritma regula-falsi (lihat Program 3.3) hampir sama dengan algoritma bagidua kecuali pada perhitungan nilai c .

Program 3.3 Metode regula-falsi

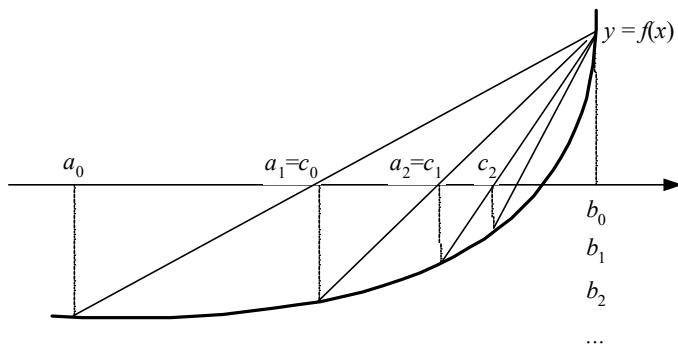
```
procedure regula_falsi(a, b: real);
{ Mencari akar  $f(x)=0$  di dalam selang  $[a,b]$  dengan metode regula falsi
  K.Awal : a dan b adalah ujung-ujung selang sehingga  $f(a)*f(b) < 0$ ,
            harga a dan b sudah terdefenisi
  K.Akhir : Hampiran akar tercetak di layar
}
const
  epsilon1 = 0.00001;           {batas lebar selang akhir larangan}
  epsilon2 = 0.000001;          {bilangan yang sangat kecil, bisa diganti }
begin
  repeat
    c:=b-(f(b)*(b-a)/(f(b)-f(a)));
    if abs(f(c))< epsilon2 then      {f(c) = 0, c adalah akar}
      begin
        a:=c;
        b:=c;
      end
    else
      if f(a)*f(c) < 0 then
        b:=c;    {selang baru [a,b]=[a,c]}
      else
        a:=c;    {selang baru [a,b]=[c,b]}
    until ABS(a-b)< epsilon1;
  { c adalah hampiran akar }
  writeln('Hampiran akar : ', c:10:6);
end;
```

Secara umum, metode regula-falsi lebih cepat konvergensi dibandingkan dengan metode bagidua. Namun, pada beberapa kasus kecepatan konvergensiya justru lebih lambat. Bila kita memakai Program 3.4 untuk menghitung akar $f(x) = e^x - 5x^2$ di dalam selang $[0, 1]$ dan $\epsilon = 0.00001$, maka tabel lelarannya yang dihasilkan adalah sebagai berikut:

r	a	c	b	$f(a)$	$f(c)$	$f(b)$	Selang baru	Lebarnya
0	0.000000	0.304718	1.000000	1.000000	0.891976	-2.281718	[c,b]	0.695282
1	0.304718	0.500129	1.000000	0.891976	0.398287	-2.281718	[c,b]	0.499871
2	0.500129	0.574417	1.000000	0.398287	0.126319	-2.281718	[c,b]	0.425583
3	0.574417	0.596742	1.000000	0.126319	0.035686	-2.281718	[c,b]	0.403258
4	0.596742	0.602952	1.000000	0.035686	0.009750	-2.281718	[c,b]	0.397048
5	0.602952	0.604641	1.000000	0.009750	0.002639	-2.281718	[c,b]	0.395359
6	0.604641	0.605098	1.000000	0.002639	0.000713	-2.281718	[c,b]	0.394902
7	0.605098	0.605222	1.000000	0.000713	0.000192	-2.281718	[c,b]	0.394778
8	0.605222	0.605255	1.000000	0.000192	0.000052	-2.281718	[c,b]	0.394745
9	0.605255	0.605264	1.000000	0.000052	0.000014	-2.281718	[c,b]	0.394736
10	0.605264	0.605266	1.000000	0.000014	0.000004	-2.281718	[c,b]	0.394734
11	0.605266	0.605267	1.000000	0.000004	0.000001	-2.281718	[c,b]	0.394733
12	0.605267	0.605267	1.000000	0.000001	0.000000	-2.281718	[c,b]	0.394733
13	0.605267	0.605267	1.000000	0.000000	0.000000	-2.281718	[c,b]	0.394733
14	0.605267	0.605267	1.000000	0.000000	0.000000	-2.281718	[c,b]	0.394733
15	0.605267	0.605267	1.000000	0.000000	0.000000	-2.281718	[c,b]	0.394733
16	0.605267	0.605267	1.000000	0.000000	0.000000	-2.281718	[c,b]	0.394733
17	0.605267	0.605267	1.000000	0.000000	0.000000	-2.281718	[c,b]	0.394733
18	0.605267	0.605267	1.000000	0.000000	0.000000	-2.281718	[c,b]	0.394733
19	0.605267	0.605267	1.000000	0.000000	0.000000	-2.281718	[c,b]	0.394733
20	0.605267	0.605267	1.000000	0.000000	0.000000	-2.281718	[c,b]	0.394733
21	0.605267	0.605267	1.000000	0.000000	-0.000000	-2.281718	[a,c]	0.000000

Hampiran akar $x = 0.605267$

Jumlah lelaran tabel di atas = 22, lebih banyak daripada jumlah lelaran metode bagidua. Bila diperhatikan, dari lelaran 12 sampai lelaran 21, nilai a , b , c tidak pernah berubah, padahal $f(c)$ sudah sangat kecil (≈ 0). Kasus seperti ini akan terjadi bila kurva fungsinya cekung (konkaf) di dalam selang $[a, b]$. Akibatnya, garis potongnya selalu terletak di atas kurva (bila kurvanya cekung ke atas) atau selalu terletak di bawah kurva (bila kurvanya cekung ke bawah). Perhatikan Gambar 3.8.



Gambar 3.8 Garis potong selalu terletak di atas kurva $y = f(x)$

Pada kondisi yang paling ekstrim, $|b - a_r|$ tidak pernah lebih kecil dari ϵ , sebab salah satu titik ujung selang, dalam hal ini b , selalu tetap untuk setiap lelaran $r = 0, 1, 2, \dots$. Titik ujung selang yang tidak pernah berubah itu dinamakan **titik mandek (stagnant point)**. Pada titik mandek,

$$|b_r - a_r| = |b - a_r| \quad r = 0, 1, 2, \dots$$

yang dapat mengakibatkan program mengalami *looping*. Untuk mengatasi hal ini, kondisi berhenti pada algoritma regula-falsi harus kita tambah dengan memeriksa apakah nilai $f(c)$ sudah sangat kecil sehingga mendekati nol. Jadi, kondisi pada `repeat-until` menjadi

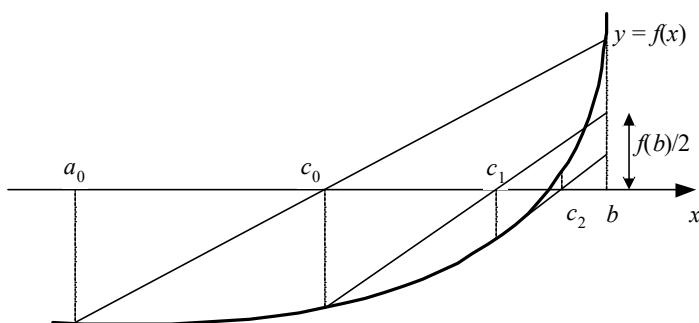
```
until (ABS(a-b) < epsilon1) or (ABS(f(c)) < epsilon2)
```

Bila perubahan ini diterapkan pada soal pencarian akar di atas dengan $\text{epsilon2} = 0.000001$, lelarannya akan berhenti pada $r = 12$ dengan akar $x = 0.605267$.

Perbaikan Metode Regula-Falsi

Untuk mengatasi kemungkinan kasus titik mandek, metode regula-falsi kemudian diperbaiki (*modified false position method*). Caranya, pada akhir lelaran $r = 0$, kita sudah memperoleh selang baru akan dipakai pada lelaran $r = 1$. Berdasarkan selang baru tersebut, tentukan titik ujung selang yang tidak berubah (jumlah perulangan > 1) - yang kemudian menjadi titik mandek. Nilai f pada titik mandek itu diganti menjadi setengah kalinya, yang akan dipakai pada lelaran $r = 1$.

Misalkan fungsi $f(x)$ cekung ke atas di dalam selang $[a, b]$ seperti yang ditunjukkan pada Gambar 3.9.



Gambar 3.9 Perbaikan metode regula-falsi

Setelah menghitung nilai c_0 pada lelaran $r = 0$, ujung selang b untuk lelaran $r = 1$ tidak berubah. Titik b menjadi titik mandek. Karena itu, untuk lelaran $r = 1$, nilai $f(b)$ yang dipakai adalah $f(b)/2$. Begitu juga untuk lelaran $r = 2$, nilai $f(b)$ yang dipakai adalah setengah dari nilai $f(b)$ sebelumnya. Pada akhir lelaran $r = 2$, c_2 sudah terletak di bawah kurva $y = f(x)$. Selang yang dipakai selanjutnya adalah $[c_1, c_2]$. Dengan cara ini kita dapat menghilangkan titik mandek yang berkepanjangan. Program 3.3 kita modifikasi menjadi Program 3.4.

Program 3.4 Metode regula-falsi yang diperbaiki

```

procedure perbaikan_regula_falsi(a, b: real);
{ Mencari akar  $f(x)=0$  di dalam selang  $[a,b]$  dengan metode regula-falsi
yang diperbaiki
K.Awal : a dan b adalah ujung-ujung selang sehingga  $f(a)*f(b) < 0$ ,
harga a dan b sudah terdefenisi
K.Akhir : akar persamaan tercetak di layar
}
const
  epsilon1 = 0.00001; {batas lebar selang akhir lelaran}
  epsilon2 = 0.000001; {batas galat nilai fungsi di hampiran akar}
var
  FA, FB, simpan : real;
  mandek_kiri, mandek_kanan : integer; {jumlah perulangan titik
  ujung selang}
begin
  FA:=f(a); FB:=f(b);
  mandek_kiri:=1; mandek_kanan:=1;
  repeat
    c:=b-(FB*(b-a) / (FB-FA));
    if abs(f(c)) < epsilon2 then { $f(c) = 0$ , c adalah akar}
      begin
        a:=c;
        b:=c;
      end
    else

      begin
        if f(a)*f(c) < 0 then
          begin
            b:=c; {selang baru  $[a,b]=[a,c]$ }
            FB:=f(c);
            mandek_kiri:=mandek_kiri + 1;
            mandek_kanan:=0;
            if mandek_kiri > 1 then
              FA:=FA/2; {a menjadi titik mandek}
          end
        else
          begin
            a:=c; {selang baru  $[a,b]=[c,b]$ }
            FA:=f(c);
            mandek_kanan:=mandek_kanan + 1;
            mandek_kiri:=0;
            if mandek_kanan > 1 then
              FB:=FB/2; {b menjadi titik mandek}
          end;
      end;
  end;
end;

```

```

    end;
until (ABS(a-b)< epsilon1) OR (ABS(f(c)) < epsilon2);
{ c adalah taksiran akar }
writeln('Hampiran akar : ', c:10:6);
end;

```

Tabel lelaran dari Program 3.4 untuk menghitung akar $f(x) = e^x - 5x^2$ di dalam selang $[0, 1]$ dengan $\epsilon = 0.00001$ dan $\delta = 0.000001$ adalah sebagai berikut:

r	a	c	b	$f(a)$	$f(c)$	$f(b)$	Selang baru	Lebarnya
0	0.000000	0.304718	1.000000	1.000000	0.891976	-2.281718 (*2)	[c,b]	0.695282
1	0.304718	0.609797	1.000000	0.891976	-0.019205	-1.140859	[a,c]	0.305079
2	0.304718	0.603367	0.609797	0.891976	0.008005	-0.019205	[c,b]	0.006430
3	0.603367	0.605259	0.609797	0.008005	0.000035	-0.019205 (*2)	[c,b]	0.004538
4	0.605259	0.605275	0.609797	0.000035	-0.000035	-0.009602	[a,c]	0.000017
5	0.605259	0.605267	0.605275	0.000035	0.000000	-0.000035	[c,b]	0.000008

Hampiran akar $x = 0.605267$

Terlihat bahwa jumlah lelarannya berkurang menjadi sepertiga semula. Harus dicatat bahwa metode regula-falsi yang diperbaiki tetap berlaku untuk fungsi yang tidak cekung sekalipun. Jadi, jika anda memprogram dengan metode regula-falsi, pakailah Program 3.4 ini untuk semua kemungkinan kasus fungsi.

3.4 Metode Terbuka

Tidak seperti pada metode tertutup, metode terbuka tidak memerlukan selang yang mengurung akar. Yang diperlukan hanya sebuah tebakan awal akar atau dua buah tebakan yang tidak perlu mengurung akar. Inilah alasan mengapa metodenya dinamakan metode terbuka. Hampiran akar sekarang didasarkan pada hampiran akar sebelumnya melalui prosedur lelaran. Kadangkala lelaran konvergen ke akar sejati, kadangkala ia divergen. Namun, apabila lelarannya konvergen, konvergensinya itu berlangsung sangat cepat dibandingkan dengan metode tertutup.

Yang termasuk ke dalam metode terbuka:

1. Metode lelaran titik-tetap (*fixed-point iteration*)
2. Metode Newton-Raphson
3. Metode *secant*

3.4.1. Metode Lelaran Titik-Tetap

Metode ini kadang-kadang dinamakan juga metode lelaran sederhana, metode langsung, atau metode sulih beruntun. Kesederhanaan metode ini karena pembentukan prosedur lelarannya mudah dibentuk sebagai berikut:

Susunlah persamaan $f(x) = 0$ menjadi bentuk $x = g(x)$. Lalu, bentuklah menjadi prosedur lelaran

$$x_{r+1} = g(x_r) \quad (\text{P.3.3})$$

dan terkalah sebuah nilai awal x_0 , lalu hitung nilai x_1, x_2, x_3, \dots , yang mudah-mudahan konvergen ke akar sejati s sedemikian sehingga

$$f(s) = 0 \text{ dan } s = g(s).$$

Kondisi berhenti lelaran dinyatakan bila

$$|x_{r+1} - x_r| < \epsilon$$

atau bila menggunakan galat relatif hampiran

$$\left| \frac{x_{r+1} - x_r}{x_{r+1}} \right| < \delta$$

dengan ϵ dan δ telah ditetapkan sebelumnya. Program lelaran titik-tetap ditunjukkan oleh Program 3.5.

Program 3.5 Metode lelaran titik-tetap

```
procedure lelaran_titik_tetap(x:real);
{ mencari akar f(x) = 0 dengan metode lelaran titik-tetap
  K.Awal : x adalah tebakan awal akar, nilainya sudah terdefinisi
  K.Akhir: akar persamaan tercetak di layar
}
const
  epsilon = 0.000001;
var
  x_sebelumnya: real;
  function g(x:real): real;
  {mengembalikan nilai g(x). Definisikan g(x), }, lihat Contoh 3.2 }
begin
  repeat
    x_sebelumnya:=x;
```

```

x:=g(x);
until ABS(x-x_sebelumnya) < epsilon;
{ x adalah hampiran akar }
write('Hampiran akar x = ', x:10:6);
end;

```

Program 3.5 hanya menangani lelaran yang konvergen. Program harus dimodifikasi menjadi Program 3.6 untuk menangani lelaran yang divergen. Salah satu cara penanganannya adalah dengan membatasi jumlah maksimum lelaran ($Nmaks$). Jika jumlah lelaran lebih besar dari $Nmaks$, maka diasumsikan lelarannya divergen.

Program 3.6 Metode lelaran titik-tetap (dengan penanganan kasus divergen)

```

procedure lelaran_titik_tetap(x:real);
{ mencari akar f(x) = 0 dengan metode lelaran titik-tetap
  K.Awal : x adalah tebakan awal akar, nilainya sudah terdefinisi
  K.Akhir: akar persamaan tercetak di layar
}
const
  epsilon = 0.000001;
  Nmaks = 30;
var
  x_sebelumnya: real; { hampiran nilai akar pada lelaran sebelumnya }
  i : integer; { pencacah jumlah lelaran }

  function g(x:real): real;
  {mengembalikan nilai g(x). Definisikan g(x) di sini, lihat Contoh 3.2 }

begin
  i:=0;
  repeat
    x_sebelumnya:=x;
    x:=g(x);
    i:=i+1;
  until (ABS(x-x_sebelumnya) < epsilon) or (i > Nmaks);
  { x adalah hampiran akar }
  if i > Nmaks then
    write('Divergen!')
  else
    write('Hampiran akar x = ', x:10:6);
end;

```

Contoh 3.2

Carilah akar persamaan $f(x) = x^2 - 2x - 3 = 0$ dengan metode lelaran titik-tetap. Gunakan $\epsilon = 0.000001$.

Penyelesaian:

Terdapat beberapa kemungkinan prosedur lelaran yang dapat dibentuk.

$$(a) \quad x^2 - 2x - 3 = 0$$

$$\begin{aligned}x^2 &= 2x + 3 \\x &= \sqrt{2x + 3}\end{aligned}$$

Dalam hal ini, $g(x) = \sqrt{2x + 3}$. Prosedur lelarannya adalah $x_{r+1} = \sqrt{2x_r + 3}$. Ambil terkaan awal $x_0 = 4$

Tabel lelarannya:

r	x_r	$ x_{r+1} - x_r $
0	4.000000	-
1	3.316625	0.683375
2	3.103748	0.212877
3	3.034385	0.069362
4	3.011440	0.022945
5	3.003811	0.007629
6	3.001270	0.002541
7	3.000423	0.000847
8	3.000141	0.000282
9	3.000047	0.000094
10	3.000016	0.000031
11	3.000005	0.000010
12	3.000002	0.000003
13	3.000001	0.000001
14	3.000000	0.000000

Hampiran akar $x = 3.000000$ (*konvergen monoton*)

$$\begin{aligned}(b) \quad x^2 - 2x - 3 &= 0 \\x(x-2) &= 3 \\x &= 3/(x - 2)\end{aligned}$$

Dalam hal ini, $g(x) = 3/(x - 2)$. Prosedur lelarannya adalah $x_{r+1} = 3/(x_r - 2)$. Ambil terkaan awal $x_0 = 4$

Tabel lelarannya:

i	x_r	$ x_{r+1} - x_r $
0	4.000000	-
1	1.500000	2.500000
2	-6.000000	7.500000
3	-0.375000	5.625000
4	-1.263158	0.888158
5	-0.919355	0.343803
6	-1.027624	0.108269

7	-0.990876	0.036748
8	-1.003051	0.012175
9	-0.998984	0.004066
10	-1.000339	0.001355
11	-0.999887	0.000452
12	-1.000038	0.000151
13	-0.999987	0.000050
14	-1.000004	0.000017
15	-0.999999	0.000006
16	-1.000000	0.000002
17	-1.000000	0.000001

Hampiran akar $x = -1.000000$ (*konvergen berosilasi*)

$$(c) \quad x^2 - 2x - 3 = 0 \\ x = (x^2 - 3)/2$$

Prosedur lelarannya adalah $x_{r+1} = (x_r^2 - 3)/2$. Ambil terkaan awal $x_0=4$

Tabel lelarannya:

i	x_r	$ x_{r+1} - x_r $
0	4.000000	-
1	6.500000	2.500000
2	19.625000	13.125000
3	191.070313	171.445312
4	18252.432159	18061.361847
...		

Ternyata lelarannya divergen! ■

Contoh 3.3

Apa yang terjadi dengan pemilihan beragam nilai x_0 pada pencarian akar persamaan

$$x^3 + 6x - 3 = 0$$

dengan prosedur lelaran

$$x_{r+1} = \frac{-x_r^3 + 3}{6} \quad [\text{PUR84}]$$

Cobakan dengan: $x_0=0.5$,
 $x_0=1.5$,
 $x_0=2.2$,
 $x_0=2.7$

Penyelesaian:

Tabel larannya adalah sebagai berikut:

r	x_r	r	x_r	r	x_r	r	x_r
0	0.5	0	1.5	0	2.2	0	2.7
1	0.4791667	1	-0.0625	1	-1.2744667	1	-2.7805
2	0.4816638	2	0.5000407	2	0.8451745	2	4.0827578
3	0.4813757	3	0.4791616	3	0.3993792	3	-10.842521
...	...	4	0.4816644	4	0.4893829	4	212.9416
7	0.4814056	5	-16909274.5
8	0.4814056	9	0.4814056	9	0.4814054		
		10	0.4814056	10	0.4814056		
				11	0.4814056		

Terlihat dengan pengambilan x_0 yang cukup dekat ke akar sejati, proses akan konvergen, tetapi jika kita mengambil x_0 terlalu jauh dari akar sejati, ia akan divergen. ■

Kadang-kadang larangan konvergen, kadang-kadang ia divergen. Adakah suatu “tanda” bagi kita untuk mengetahui kapan suatu larangan konvergen dan kapan divergen?

Kriteria konvergensi

Diberikan prosedur larangan

$$x_{r+1} = g(x_r) \quad (\text{P.3.4})$$

Misalkan $x = s$ adalah solusi $f(x) = 0$ sehingga $f(s) = 0$ dan $s = g(s)$. Selisih antara x_{r+1} dan s adalah

$$\begin{aligned} x_{r+1} - s &= g(x_r) - s \\ &= \frac{g(x_r) - s}{(x_r - s)} (x_r - s) \end{aligned} \quad (\text{P.3.5})$$

Terapkan teorema nilai rata-rata pada persamaan (P.3.5) sehingga

$$x_{r+1} - s = g'(t)(x_r - s) \quad (\text{P.3.6})$$

yang dalam hal ini $x_{r+1} < t < s$. Misalkan galat pada lelaran ke- r dan lelaran ke- $(r+1)$ adalah

$$\varepsilon_r = x_r - s \text{ dan } \varepsilon_{r+1} = x_{r+1} - s$$

Persamaan (P.4.6) dapat kita tulis menjadi

$$\varepsilon_{r+1} = g'(t) \varepsilon_r \quad (\text{P.3.7})$$

atau dalam tanda mutlak

$$|\varepsilon_{r+1}| = |g'(t)| |\varepsilon_r| \leq K |\varepsilon_r|$$

Berapakah batas-batas nilai K itu?

Misalkan x_0 dan x berada di dalam selang sejauh $2h$ dari s , yaitu $s - h < x < s + h$. Jika lelaran konvergen di dalam selang tersebut, yaitu $x_0, x_1, x_2, x_3, \dots$ menuju s , maka galat setiap lelaran berkurang. Jadi, haruslah dipenuhi kondisi

$$|\varepsilon_{r+1}| \leq K |\varepsilon_r| \leq K^2 |\varepsilon_{r-1}| \leq K^3 |\varepsilon_{r-2}| \leq \dots \leq K^{r+1} |\varepsilon_0|$$

Kondisi tersebut hanya berlaku jika

$$g'(x) \leq K < 1$$

Karena $K < 1$, maka $K^{r+1} \rightarrow 0$ untuk $r \rightarrow \infty$; di sini $|x_{r+1} - s| \rightarrow 0$.

TEOREMA 3.2. Misalkan $g(x)$ dan $g'(x)$ menerus di dalam selang $[a, b] = [s-h, s+h]$ yang mengandung titik tetap s dan nilai awal x_0 dipilih dalam selang tersebut. Jika $|g'(x)| < 1$ untuk semua $x \in [a, b]$ maka lelaran $x_{r+1} = g(x_r)$ akan konvergen ke s . Pada kasus ini s disebut juga *titik atraktif*. Jika $|g'(x)| > 1$ untuk semua $x \in [a, b]$ maka lelaran $x_{r+1} = g(x_r)$ akan divergen dari s .

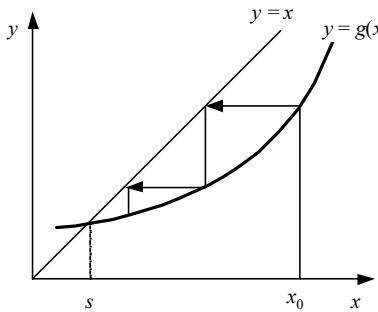
Teorema 3.2 dapat kita sarikan sebagai berikut:

Di dalam selang $I = [s-h, s+h]$, dengan s titik tetap,

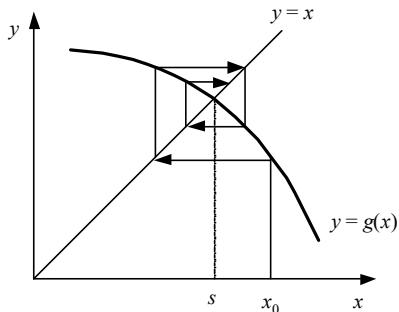
- jika $0 < g'(x) < 1$ untuk setiap $x \in I$, maka lelaran *konvergen monoton*;
- jika $-1 < g'(x) < 0$ untuk setiap $x \in I$, maka lelaran *konvergen bersosilasi*;
- jika $g'(x) > 1$ untuk setiap $x \in I$, maka lelaran *divergen monoton*;
- jika $g'(x) < -1$ untuk setiap $x \in I$, maka lelaran *divergen bersosilasi*.

Semuanya dirangkum seperti pada Gambar 3.10..

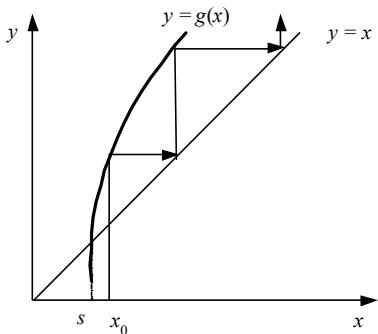
Sebagai catatan, keadaan $|g'(x)| = 1$ tidak didefinisikan. Catat juga bahwa semakin dekat nilai $|g'(x)|$ ke nol di dekat akar, semakin cepat kekonvergenan metode lelaran titik-tetap ini [PUR84].



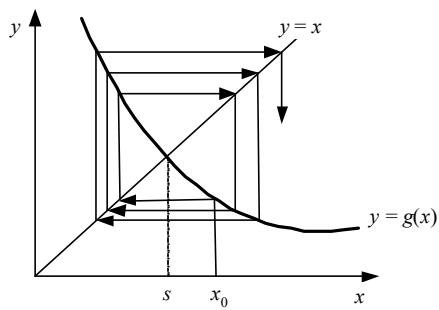
(a) Konvergen monoton: $0 < g'(x) < 1$



(b) Konvergen berosilasi: $-1 < g'(x) < 0$



(c) Divergen monoton: $g'(x) > 1$



(d) Divergen berosilasi: $g'(x) < -1$

Gambar 3.10 Jenis-jenis kekonvergenan

Sekarang, mari kita analisis mengapa pencarian akar persamaan $x^2 - 2x - 3 = 0$ pada Contoh 3.2 dan pencarian akar persamaan $x^3 + 6x - 3 = 0$ pada Contoh 3.3 dengan bermacam-macam prosedur lelaran dan tebakan awal kadang-kadang konvergen dan kadang-kadang divergen.

- (i) Prosedur lelaran pertama: $x_{r+1} = \sqrt{2x_r + 3}$

$$g(x) = \sqrt{2x + 3}$$

$$g'(x) = \frac{1}{2\sqrt{(2x+3)}}$$

Terlihat bahwa $|g'(x)| < 1$ untuk x di sekitar titik-tetap $s = 3$. Karena itu, pengambilan tebakan awal $x_0 = 4$ akan menghasilkan larutan yang konvergen sebab $|g'(4)| = |1/[2\sqrt{(8+3)}]| = 0.1508 < 1$.

(ii) Prosedur larutan kedua: $x_{r+1} = 3/(x_r - 2)$

$$g(x) = 3/(x-2)$$

$$g'(x) = -3/(x-2)^2$$

Terlihat bahwa $|g'(x)| < 1$ untuk x di sekitar titik-tetap $s = 3$. Karena itu, pengambilan tebakan awal $x_0 = 4$ akan menghasilkan larutan yang konvergen sebab

$$|g'(4)| = |-3/(4-2)^2| = 0.75 < 1.$$

(iii) Prosedur larutan ketiga $x_{r+1} = (x_r^2 - 3)/2$

$$g(x) = (x^2 - 3)/2$$

$$g'(x) = x$$

Terlihat bahwa $|g'(x)| > 1$ untuk x di sekitar titik-tetap $s = 3$. Karena itu, pengambilan tebakan awal $x_0 = 4$ akan menghasilkan larutan yang divergen sebab

$$|g'(4)| = |4| = 4 > 1.$$

(iv) Prosedur larutan pada Contoh 3.3: $x_{r+1} = (-x_r^3 + 3)/6$

$$g(x) = (-x^3 + 3)/6$$

$$g'(x) = -x^2/2$$

Terlihat bahwa $|g'(x)| < 1$ untuk x di sekitar titik-tetap $s = 0.48$. Pemilihan $x_0 = 0.5$ akan menjamin larutan konvergen sebab $|g'(x_0)| < 1$. Untuk $x_0 = 1.5$ dan $x_0 = 2.2$ memang nilai $|g'(x_0)| > 1$ tetapi larutannya masih tetap konvergen, namun $x_0 = 2.7$ terlalu jauh dari titik-tetap sehingga larutannya divergen. Dapatkah kita menentukan batas-batas selang yang menjamin prosedur larutan akan konvergen di dalamnya? Temukan jawabannya pada Contoh 3.4 di bawah ini.

Contoh 3.4

Pada Contoh 3.3 di atas, tentukan selang sehingga prosedur lelaran

$$x_{r+1} = (-x_r^3 + 3)/6$$

konvergen?

Penyelesaian:

$$g(x) = (-x^3 + 3)/6$$

$$g'(x) = -x^2/2$$

Syarat konvergen adalah $|g'(x)| < 1$. Jadi,

$$\Leftrightarrow |-x^2/2| < 1$$

$$\Leftrightarrow -1 < -x^2/2 < 1$$

$$\Leftrightarrow 2 > x^2 > -2$$

$$\Leftrightarrow -2 < x^2 < 2$$

Urai satu per satu:

(i) $x^2 > -2$ { tidak ada x yang memenuhi}

(ii) $x^2 < 2$, dipenuhi oleh

$$\Leftrightarrow x^2 - 2 < 0$$

$$\Leftrightarrow -\sqrt{2} < x < \sqrt{2}$$

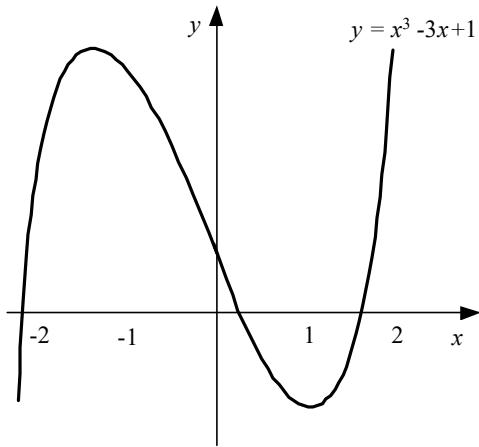
Jadi, prosedur lelaran $x_{r+1} = (-x_r^3 + 3)/6$ konvergen di dalam selang $-\sqrt{2} < x < \sqrt{2}$. Kita dapat memilih x_0 dalam selang tersebut yang menjamin lelaran akan konvergen. ■

Contoh 3.5

Gunakan metode lelaran titik-tetap untuk mencari akar persamaan

$$x^3 - 3x + 1 \text{ dalam selang } [1, 2] \quad [\text{PUR84}]$$

Catatan : selang $[1, 2]$ ini sebenarnya tidak digunakan dalam proses lelaran sebagaimana halnya pada metode bagidua. Selang ini diberikan untuk memastikan bahwa suatu prosedur lelaran titik-tetap konvergen di dalamnya. Kurva fungsi $y = x^3 - 3x + 1$ diperlihatkan pada Gambar 3.11.



Gambar 3.11 Kurva $y = x^3 - 3x + 1$

Penyelesaian:

$$(i) \quad x_{r+1} = (x_r^3 + 1)/3$$

Tetapi, karena $|g'(x)| = |x^2| > 1$ dalam selang $[1, 2]$, maka prosedur lelaran ini tidak digunakan.

$$(ii) \quad x_{r+1} = -1/(x_r^2 - 3)$$

Tetapi, karena $|g'(x)| = |2x / (x^2 - 3)^3| > 1$ dalam selang $[1, 2]$, maka prosedur lelaran ini tidak digunakan.

$$(iii) \quad x_{r+1} = 3/x_r - 1/x_r^2$$

Ternyata $|g'(x)| = |(-3x + 2)/x^3| \leq 1$ di dalam selang $[1, 2]$, yaitu, $g'(x)$ naik dari $g'(1) = -1$ ke $g'(2) = -1/2$. Jadi, $|g'(x)|$ lebih kecil dari 1 dalam selang $[1, 2]$. Dengan mengambil $x = 1.5$, prosedur lelarannya konvergen ke akar $x = 1.5320889$ seperti pada tabel berikut ini.

r	x
0	1.5
1	1.5555556
2	1.5153061
...	...
43	1.5320888
44	1.5320889
45	1.5320889

■

Contoh 3.5 menunjukkan bahwa ada dua hal yang mempengaruhi kekonvergenan prosedur lelaran:

1. Bentuk formula $x_{r+1} = g(x_r)$
2. Pemilihan tebakan awal x

Catatan: Meskipun $|g'(x)| > 1$ menyatakan lelaran divergen, tetapi kita harus hati-hati dengan pernyataan ini. Sebabnya, walaupun x_r divergen dari suatu akar, runtunan lelarannya mungkin konvergen ke akar yang lain. Kasus seperti ini ditunjukkan pada Contoh 3.6 di bawah ini.

Contoh 3.6

Tentukan akar persamaan $f(x) = x^2 - 4x + 3 = 0$ dengan prosedur lelaran

$$x_{r+1} = (x_r^2 + 3)/4$$

Penyelesaian:

Jika prosedur lelaran $x_{r+1} = (x_r^2 + 3)/4$ konvergen ke titik-tetap s , maka

$$\lim_{r \rightarrow \infty} x_r = s$$

sehingga

$$\begin{aligned}s &= (s^2 + 3)/4 \\ s^2 - 4s + 3 &= 0 \\ (s - 3)(s - 1) &= 0\end{aligned}$$

yang memberikan $s_1 = 1$ atau $s_2 = 3$. Jadi, lelaran konvergen ke akar $x = 1$ atau akar $x = 3$. Dari

$$g(x) = (x^2 + 3)/4$$

diperoleh

$$g'(x) = x/2$$

Gambarkan kurva $y = x$ dan $y = (x^2 + 3)/4$ seperti pada Gambar 3.12.

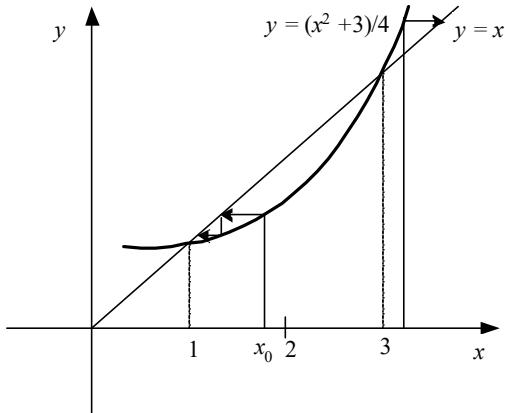
Prosedar lelaran akan konvergen bila

$$\begin{aligned}\Leftrightarrow |g'(x)| &< 1 \\ \Leftrightarrow -1 < x/2 < 1\end{aligned}$$

atau

$$-2 < x < 2$$

Sehingga pemilihan x_0 dalam selang $-2 < x < 2$ menjamin lelaran konvergen ke akar $x = 1$. Dari Gambar 3.12 terlihat bahwa lelaran juga konvergen ke akar $x = 1$ untuk pemilihan x_0 dalam selang $2 < x < 3$. Padahal, kalau dihitung, dalam selang $2 < x < 3$, $|g'(x)| > 1$ yang menyatakan bahwa lelarannya divergen. Lelaran divergen dari akar $x = 3$ tetapi konvergen ke akar $x = 1$. ■



Gambar 3.12 Kurva $y = x$ dan $y = (x^2 + 3)/4$

Sebagai contoh terakhir metode lelaran titik-tetap, mari kita hitung akar fungsi pada Contoh 3.1, yaitu $f(x) = e^x - 5x^2$.

Contoh 3.7

Hitunglah akar $f(x) = e^x - 5x^2$ dengan metode lelaran titik-tetap. Gunakan $\epsilon = 0.00001$. Tebakan awal akar $x_0 = 1$.

Penyelesaian:

Salah satu prosedur lelaran yang dapat dibuat adalah

$$\begin{aligned}e^x - 5x^2 &= 0 \\e^x &= 5x^2 \\x &= \sqrt{e^x/5} \\x_{r+1} &= \text{SQRT}(\text{EXP}(x_r)/5)\end{aligned}$$

Tabel lelarannya:

i	x_r	$ x_{r+1} - x_r $
0	0.500000	-
1	0.574234	0.074234

2	0.595948	0.021714
3	0.602453	0.006506
4	0.604416	0.001963
5	0.605010	0.000593
6	0.605189	0.000180
7	0.605244	0.000054
8	0.605260	0.000016
9	0.605265	0.000005
10	0.605266	0.000002
11	0.605267	0.000000

Hampiran akar $x = 0.605267$

■

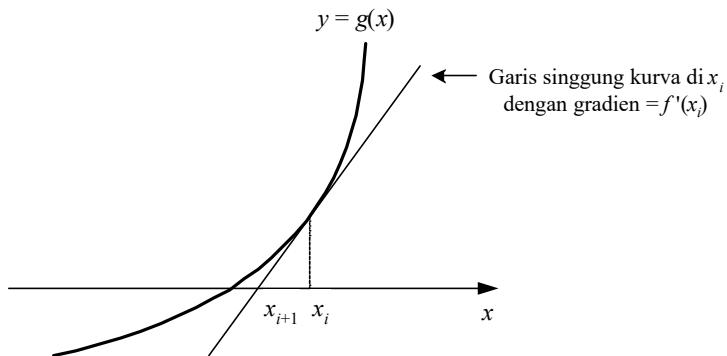
3.4.2 Metode Newton-Raphson³

Di antara semua metode pencarian akar, metode Newton-Raphsonlah yang paling terkenal dan paling banyak dipakai dalam terapan sains dan rekayasa. Metode ini paling disukai karena konvergensiya paling cepat diantara metode lainnya.

Ada dua pendekatan dalam menurunkan rumus metode Newton-Raphson, yaitu:

- (a) penurunan rumus Newton-Raphson secara geometri,
- (b) penurunan rumus Newton-Raphson dengan bantuan deret Taylor.

(a) Penurunan rumus Newton-Raphson secara geometri



Gambar 3.13 Tafsiran geometri metode Newton-Raphson

³Beberapa buku menyebutnya *metode Newton* saja. Joseph Raphson (1648 - 1715) adalah matematikawan Inggris yang mempublikasikan metode Newton.

Dari Gambar 3.13, gradien garis singgung di x_r adalah

$$m = f'(x_r) = \frac{\Delta y}{\Delta x} = \frac{f(x_r) - 0}{x_r - x_{r+1}} \quad (\text{P.3.8})$$

atau

$$f'(x_r) = \frac{f(x_r)}{x_r - x_{r+1}} \quad (\text{P.3.9})$$

sehingga prosedur pelajaran metode Newton-Raphson adalah

$$x_{r+1} = x_r - \frac{f(x_r)}{f'(x_r)}, \quad f'(x_r) \neq 0. \quad (\text{P.3.10})$$

(b) Penurunan rumus Newton-Raphson dengan bantuan deret Taylor

Uraikan $f(x_{r+1})$ di sekitar x_r ke dalam deret Taylor:

$$f(x_{r+1}) \approx f(x_r) + (x_{r+1} - x_r)f'(x_r) + \frac{(x_{r+1} - x_r)^2}{2}f''(t), \quad x_r < t < x_{r+1}$$

(P.3.11)

yang bila dipotong sampai suku orde-2 saja menjadi

$$f(x_{r+1}) \approx f(x_r) + (x_{r+1} - x_r)f'(x_r) \quad (\text{P.3.12})$$

dan karena persoalan mencari akar, maka $f(x_{r+1}) = 0$, sehingga

$$0 = f(x_r) + (x_{r+1} - x_r)f'(x_r) \quad (\text{P.3.13})$$

atau

$$x_{r+1} = x_r - \frac{f(x_r)}{f'(x_r)}, \quad f'(x_r) \neq 0 \quad (\text{P.3.14})$$

yang merupakan rumus metode Newton-Raphson.

Kondisi berhenti lelaran Newton-Raphson adalah bila

$$|x_{r+1} - x_r| < \epsilon$$

atau bila menggunakan galat relatif hampiran

$$\left| \frac{x_{r+1} - x_r}{x_{r+1}} \right| < \delta$$

dengan ϵ dan δ adalah toleransi galat yang diinginkan.

Catatan:

1. Jika terjadi $f'(x_r) = 0$, ulang kembali perhitungan lelaran dengan x_0 yang lain.
2. Jika persamaan $f(x) = 0$ memiliki lebih dari satu akar, pemilihan x_0 yang berbeda-beda dapat menemukan akar yang lain.
3. Dapat pula terjadi lelaran konvergen ke akar yang berbeda dari yang diharapkan (seperti halnya pada metode lelaran titik-tetap).

Program 3.7 Metode Newton-Raphson

```
procedure Newton_Raphson(x:real);
{ Mencari akar persamaan f(x) = 0 dengan metode Newton-Raphson
  K.Awal : x adalah tebakan awal akar, nilainya sudah terdefinisi
  K.Akhir: akar persamaan tercetak di layar
}
const
  epsilon = 0.000001;
var
  x_sebelumnya: real;

  function f(x:real):real;
  { mengembalikan nilai f(x). Definisi f(x) bergantung pada persoalan }

  function f_akses(x:real):real;
  { mengembalikan nilai f'(x). Definisi f'(x) bergantung
    pada persoalan }

begin
  repeat
    x_sebelumnya:=x;
    x:=x - f(x)/f_akses(x);
  until (ABS(x-x_sebelumnya) < epsilon)

  { x adalah hampiran akar persamaan }
  write('Hampiran akar x = ', x:10:6);

end;
```

Catatan: Program 3.7 ini belum menangani kasus pembagian dengan 0 atau ≈ 0 dan kasus divergen. Program 3.8 di bawah ini merupakan modifikasi dari Program 3.7 untuk menangani pembagian dengan 0 dan kasus divergen.

Program 3.8 Metode Newton-Raphson (dengan penanganan kasus divergen dan pembagian dengan 0)

```

procedure Newton_Raphson(x:real);
{ Mencari akar persamaan f(x) = 0 dengan metode Newton-Raphson
  K.Awal : x adalah tebakan awal akar, nilainya sudah terdefinisi
  K.Akhir: akar persamaan tercetak di layar
}
const
  epsilon1 = 0.000001;      { toleransi galat akar hampiran }
  epsilon2 = 0.000000001;   { toleransi nilai yang hampir 0 }
  Nmaks = 30;               { jumlah maksimum lelaran }
var
  x_sebelumnya: real;
  i : integer;
  berhenti : boolean;     { jika f'(x) <<< 0, stop ! }

  function f(x:real):real;
  { mengembalikan nilai f(x). Definisi f(x) bergantung pada persoalan }

  function f_aksi(x:real):real;
  { mengembalikan nilai f'(x). Definisi f'(x) bergantung
    pada persoalan }

begin
  i:=0;
  berhenti:=false;
  repeat
    if ABS(f_aksi(x)) < epsilon2 then
      berhenti:=true; { menghindari pembagian bilangan yang ≈ 0}
    else
      begin
        x_sebelumnya:=x;
        x:=x - f(x) / f_aksi(x);
        i:=i+1;
      end;
  until (ABS(x-x_sebelumnya) < epsilon1) or (berhenti) or (i > Nmaks)

  if berhenti then
    writeln('Pembagian dengan bilangan yang hampir 0')
  else
    if i > Nmaks then
      writeln('Divergen')
    else
      { x adalah hampiran akar persamaan }
      write('Hampiran akar x = ', x:10:6);
  {endif}
  {endif}
end;

```

Contoh 3.8

Hitunglah akar $f(x) = e^x - 5x^2$ dengan metode Newton-Raphson. Gunakan $\varepsilon = 0.00001$. Tebakan awal akar $x_0 = 1$.

Penyelesaian:

$$f(x) = e^x - 5x^2$$

$$f'(x) = e^x - 10x$$

Prosedur lelaran Newton-Raphson:

$$x_{r+1} = x_r - \frac{e^x - 5x^2}{e^x - 10x}$$

Tebakan awal $x_0 = 1$

Tabel larannya:

i	x_r	$ x_{r+1} - x_r $
0	0.500000	-
1	0.618976	0.118976
2	0.605444	0.013532
3	0.605267	0.000177
4	0.605267	0.000000

Hampiran akar $x = 0.605267$ ■

Contoh 3.8 di atas memperlihatkan bahwa metode Newton-Raphson memerlukan sedikit lelaran, dibandingkan dengan metode bagidua, metode regula falsi, dan metode lelaran titik-tetap. Metode Newton-Raphson sangat berguna untuk menghitung fungsi-fungsi dasar, seperti akar bilangan, nilai e , arcsin (x), dan sebagainya. Contoh 3.9 dan Contoh 3.10 memperlihatkan penggunaan metode Newton-Raphson untuk menghitung akar bilangan dan nilai pecahan.

Contoh 3.9

Tentukan bagaimana cara menentukan \sqrt{c} dengan metode Newton-Raphson.

Penyelesaian:

Misalkan $\sqrt{c} = x$. Kuadratkan kedua ruas sehingga $c = x^2 \Leftrightarrow x^2 - c = 0$.

Di sini $f(x) = x^2 - c$ dan $f'(x) = 2x$. Prosedur lelaran Newton-Raphsonnya adalah

$$x_{r+1} = x_r - \frac{x_r^2 - c}{2x_r} = 0.5(x_r + c/x_r)$$

Untuk $c = 2$, dengan memilih $x_0 = 1$ dan $\epsilon = 0.000001$, kita peroleh

$$\begin{aligned}x_1 &= 1.500000 \\x_2 &= 1.416667 \\x_3 &= 1.414216 \\x_4 &= 1.414214\end{aligned}$$

Jadi, $\sqrt{2} \approx 1.414214$

■

Contoh 3.10

Bagaimana menghitung nilai $1/c$ dengan metode Newton-Raphson?

Penyelesaian:

Misalkan $1/c = x \Leftrightarrow 1/x = c \Leftrightarrow 1/x - c = 0$. Di sini $f(x) = 1/x - c$ dan $f'(x) = -1/x^2$.

Prosedur Newton-Raphsonnya adalah

$$x_{r+1} = x_r - \frac{(1/x_r - c)}{-1/x_r^2} = x_r(2 - cx_r)$$

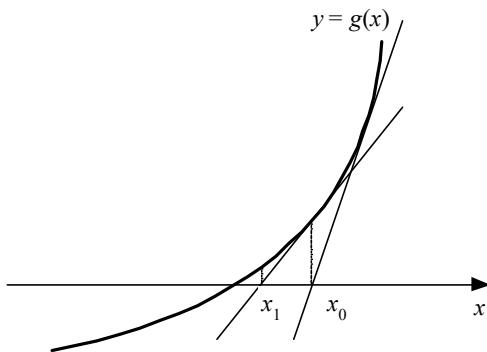
Untuk $c = 7$, dengan memilih $x_0 = 0.2$ dan $\epsilon = 0.0000001$, kita peroleh

$$\begin{aligned}x_1 &= 0.1200000 \\x_2 &= 0.1392000 \\x_3 &= 0.1427635 \\x_4 &= 0.1428570 \\x_5 &= 0.1428751 \\x_6 &= 0.1428571\end{aligned}$$

Jadi, $1/7 \approx 0.1428571$

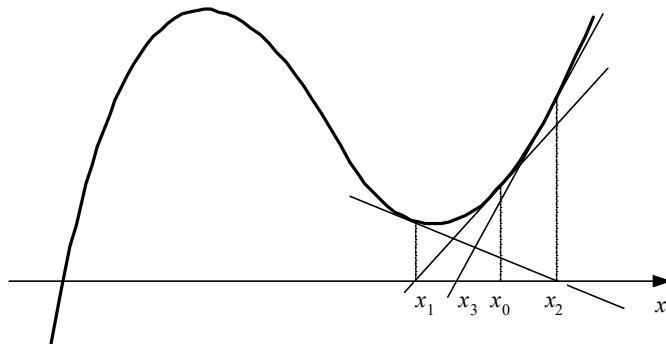
■

Secara umum, bila metode Newton-Raphson konvergen, kekonvergenannya itu berlangsung sangat cepat, seperti yang dilukiskan pada Gambar 3.14. Titik potong garis singgung fungsi dengan sumbu-x semakin cepat bergerak mendekati akar sejati.



Gambar 3.14 Kecepatan konvergensi metode Newton-Raphson sangat cepat

Karena metode Newton-Raphson tergolong metode terbuka, maka dalam beberapa kasus lelarannya mungkin divergen. Bahkan, kalau kurvanya seperti pada Gambar 3.15 serta pemilihan x_0 yang jauh dari akar sejati, lelarannya akan berosilasi di sekitar cekungan lain.



Gambar 3.15 Lelaran metode Newton-Raphson yang divergen

Membuat grafik fungsi sangat membantu dalam pencarian akar. Grafik fungsi dapat memperlihatkan secara visual lokasi akar sejati. Dengan demikian tebakan awal yang bagus untuk akar dapat diturunkan. Pemilihan tebakan awal sebaiknya cukup dekat dengan akar. Selain itu, kita juga dapat mengetahui apakah fungsi tersebut mempunyai akar tidak. Pada kasus tidak ada akar, lelarannya akan divergen berosilasi.

Kriteria konvergensi metode Newton-Raphson

Apakah persyaratan agar metode Newton-Raphson konvergen? Tinjau kembali bentuk umum prosedur lelaran metode terbuka,

$$x_{r+1} = g(x_r)$$

Karena metode Newton-Raphson termasuk metode terbuka, maka dalam hal ini,

$$g(x) = x - \frac{f(x)}{f'(x)}$$

Dengan mengingat syarat perlu agar lelaran konvergen adalah $|g'(x)| < 1$, maka

$$\begin{aligned} g'(x) &= 1 - \frac{[f'(x)f'(x) - f(x)f''(x)]}{[f'(x)]^2} \\ &= \frac{f(x)f''(x)}{[f'(x)]^2} \end{aligned} \tag{P.3.18}$$

Karena itu, metode Newton-Raphson akan konvergen bila

$$\left| \frac{f(x)f''(x)}{[f'(x)]^2} \right| < 1$$

dengan syarat $f''(x) \neq 0$.

3.4.3 Orde Konvergensi Metode Terbuka

Prosedur lelaran pada setiap metode terbuka dapat ditulis dalam bentuk

$$x_{r+1} = g(x_r) \tag{P.3.19}$$

misalnya pada metode Newton-Raphson $g(x_r) = x_r - f(x_r)/f'(x_r)$. Misalkan x_r adalah hampiran terhadap akar sejati s sehingga $s = g(s)$. Maka, berdasarkan konsep galat yang sudah dijelaskan di dalam Bab 2, $s = x_r + \varepsilon_r$ dengan ε_r adalah galat dari x_r . Uraikan $g(s)$ di sekitar x_r :

$$\begin{aligned} g(s) &= g(x_r) + g'(x_r)(s - x_r) + \frac{1}{2} g''(x_r)(s - x_r)^2 + \dots \\ &= g(x_r) + g'(x_r)\varepsilon_r + \frac{1}{2} g''(x_r)\varepsilon_r^2 + \dots \end{aligned} \tag{P.3.20}$$

Kurangi persamaan (P.3.20) dengan persamaan (P.3.19):

$$\begin{aligned} g(s) &= g(x_r) + g'(x_r)\epsilon_r + \frac{1}{2}g''(x_r)\epsilon_r^2 + \dots \\ -x_{r+1} &= g(x_r) \\ \hline g(s) - x_{r+1} &= g'(x_r)\epsilon_r + \frac{1}{2}g''(x_r)\epsilon_r^2 + \dots \end{aligned}$$

Karena $g(s) = s$, maka

$$s - x_{r+1} = g'(x_r)\epsilon_r + \frac{1}{2}g''(x_r)\epsilon_r^2 + \dots$$

Misalkan $s - x_{r+1} = \epsilon_{r+1}$, sehingga

$$\epsilon_{r+1} = g'(x_r)\epsilon_r + \frac{1}{2}g''(x_r)\epsilon_r^2 + \dots \quad (\text{P.3.21})$$

Bilangan pangkat dari ϵ_r menunjukkan orde (atau laju) konvergensi prosedur lelaran:

$$(a) \epsilon_{r+1} \approx g'(t)\epsilon_r, \quad x_r < t < x_{r+1} : \text{prosedur lelaran berorde satu} \quad (\text{P.3.22})$$

$$(b) \epsilon_{r+1} \approx \frac{1}{2}g''(t_r)\epsilon_r^2, \quad x_r < t < x_{r+1} : \text{prosedur lelaran berorde dua} \quad (\text{P.3.23})$$

Metode Newton-Raphson termasuk ke dalam metode terbuka berorde dua. Pernyataan ini kita buktikan di bawah ini.

Orde konvergensi metode Newton-Raphson

Pada metode Newton-Raphson, $g(x_r) = x_r - f(x_r) / f'(x_r)$. Turunan pertama dari $g(x_r)$ adalah (dari persamaan P.3.18):

$$g'(x_r) = \frac{f(x_r)f''(x_r)}{[f'(x_r)]^2} \quad (\text{P.3.24})$$

Jika x_r adalah akar persamaan $f(x) = 0$, maka $f(x_r) = 0$, sehingga

$$g'(x_r) = 0$$

Ini berarti metode Newton-Raphson paling sedikit berorde dua. Turunan kedua dari $g(x_r)$ adalah

$$g''(x_r) = f''(x_r) / f'(x_r) \quad (\text{P.3.25})$$

Sulihkan (P.3.25) ke dalam (P.3.23):

$$\varepsilon_{r+1} = \frac{f''(x_r) \varepsilon_r^2}{2 f'(x_r)} \quad (\text{P.3.26})$$

Persamaan (P.3.26) ini mempunyai tiga arti:

1. Galat lelaran sekarang sebanding dengan kuadrat galat lelaran sebelumnya. Jika galat lelaran sekarang misalnya 0.001, maka pada lelaran berikutnya galatnya sebanding dengan 0.000001. Hal inilah yang menyebabkan metode Newton-Raphson sangat cepat menemukan akar (jika lelarannya konvergen).
2. Jumlah angka bena akan berlipat dua pada tiap lelaran. Ini merupakan konsekuensi dari hal nomor 1 di atas.
3. Orde konvergensi metode Newton-Raphson adalah kuadratik. sehingga ia dinamakan juga *metode kuadratik*.

Cara lain untuk menemukan orde konvergensi metode Newton-Raphson adalah dengan meneruskan penurunan rumus Newton-Raphson dari deret Taylornya sebagai berikut. Perhatikan kembali persamaan (P.3.11) di atas. Bila $x_{r+1} = s$ sehingga $f(x_{r+1}) = f(s) = 0$, dalam hal ini s adalah akar sejati, sulihkan s ke dalam persamaan (P.3.11) di atas:

$$0 = f(x_r) + (s - x_r)f'(x_r) + \frac{(s - x_r)^2 f''(t)}{2} \quad (\text{P.3.27})$$

Kurangi (P.3.27) dengan (P.3.13):

$$\begin{aligned} 0 &= f(x_r) + (s - x_r)f'(x_r) + \frac{(s - x_r)^2 f''(t)}{2} \\ 0 &= f(x_r) + (x_{r+1} - x_r)f'(x_r) \\ &\hline 0 &= (s - x_{r+1})f'(x_r) + \frac{(s - x_r)^2 f''(t)}{2} \end{aligned} \quad (\text{P.3.28})$$

Misalkan $s - x_{r+1} = \varepsilon_{r+1}$ dan $s - x_r = \varepsilon_r$, maka persamaan (P.3.28) dapat ditulis menjadi

$$\varepsilon_{r+1} f'(x_r) + \frac{\varepsilon_r^2 f''(t)}{2} = 0$$

atau

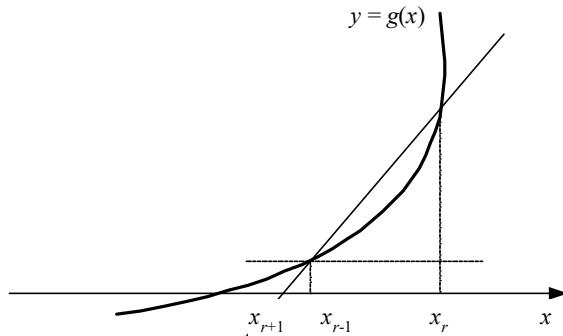
$$\varepsilon_{r+1} = \frac{f''(t)\varepsilon_r^2}{2f'(x_r)} \quad (\text{P.3.29})$$

yang sama dengan (P.3.26), kecuali pada $f''(x_r)$ dan $f''(t)$, tetapi perbedaan ini tidak begitu penting, sebab yang dicari adalah pangkat dari ε_r .

Pada proses pencarian akar dengan metode Newton-Raphson, muncul kesulitan jika $|f'(x)|$ terlalu dekat ke nol, dan kita harus menggunakan bilangan berketelitian ganda untuk memperoleh $f(x)$ dan $f'(x)$ cukup teliti [KRE88]. Persamaan nirlanjar $f(x) = 0$ yang mempunyai kasus seperti ini disebut berkondisi buruk (lihat pembahasan kondisi buruk di dalam Bab 2).

3.4.4 Metode Secant

Prosedur pelajaran metode Newton-Raphson memerlukan perhitungan turunan fungsi, $f'(x)$. Sayangnya, tidak semua fungsi mudah dicari turunannya, terutama fungsi yang bentuknya rumit. Turunan fungsi dapat dihilangkan dengan cara menggantinya dengan bentuk lain yang ekivalen. Modifikasi metode Newton-Raphson ini dinamakan *metode secant*.



Gambar 3.16 Metode Secant

Berdasarkan Gambar 3.16, dapat kita hitung gradien

$$f'(x_r) = \frac{\Delta y}{\Delta x} = \frac{AC}{BC} = \frac{f(x_r) - f(x_{r-1})}{x_r - x_{r-1}} \quad (\text{P.3.30})$$

Sulihkan (P.3.30) ke dalam rumus Newton-Raphson:

$$x_{r+1} = x_r - \frac{f(x_r)}{f'(x_r)}$$

sehingga diperoleh

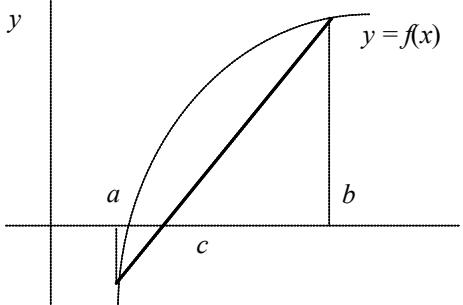
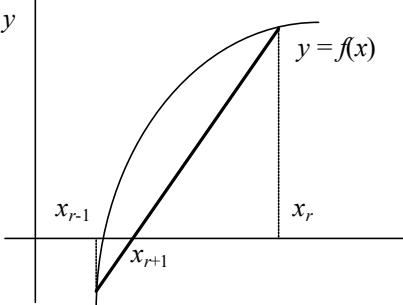
$$x_{r+1} = x_r - \frac{f(x_r)(x_r - x_{r-1})}{f(x_r) - f(x_{r-1})} \quad (\text{P.3.31})$$

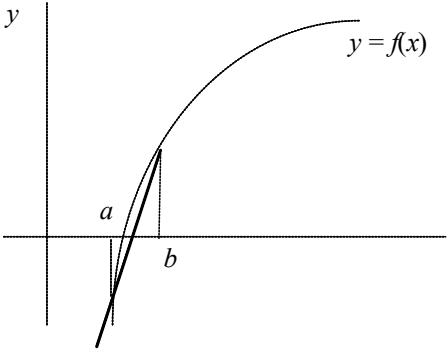
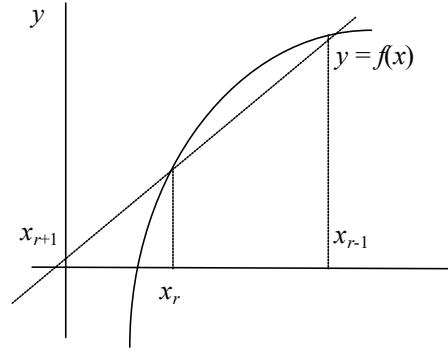
yang merupakan prosedur lelaran metode secant. Dalam hal ini, diperlukan dua buah tebakan awal akar, yaitu x_0 dan x_1 . Kondisi berhenti lelaran adalah bila

$$|x_{r+1} - x_r| < \varepsilon \quad (\text{galat mutlak}) \quad \text{atau} \quad \left| \frac{x_{r+1} - x_r}{x_{r+1}} \right| < \delta \quad (\text{galat hampiran})$$

dengan ε dan δ adalah toleransi galat.

Sepintas metode secant mirip dengan metode regula-falsi, namun sesungguhnya prinsip dasar keduanya berbeda, seperti yang dirangkum pada tabel di bawah ini:

Metode Regula Falsi	Metode Secant
<p>1. Diperlukan dua buah nilai awal a dan b (ujung-ujung selang) sedemikian sehingga $f(a)f(b) < 0$.</p> <p>2. <u>Lelaran pertama:</u></p> 	<p>1. Diperlukan dua buah nilai awal x_0 dan x_1 (tebakan awal akar), tetapi <u>tidak harus</u> $f(x_0)f(x_1) < 0$.</p> <p>2. <u>Lelaran pertama:</u></p> 
<p>Pada lelaran pertama, tidak ada perbedaan antara regula-falsi dan secant.</p> <p>Perbedaan baru muncul pada lelaran kedua.</p>	<p>Pada lelaran pertama tidak ada perbedaan antara secant dan regula falsi.</p> <p>Perbedaan baru muncul pada lelaran kedua.</p>

<u>Lelaran kedua:</u> 	<u>Lelaran kedua:</u> 
Perpotongan garis lurus dengan sumbu x tetap berada di dalam selang yang mengandung akar.	Perpotongan garis lurus dengan sumbu- x mungkin menjauhi akar.
3. Berdasarkan nomor 2 di atas, lelarannya <i>selalu</i> konvergen	3. Berdasarkan nomor 2 di atas, lelarannya <i>mungkin</i> divergen.

Program 3.9 berikut berisi algoritma metode *secant*.

Program 3.9 Metode Secant

```

procedure Secant(x0, x1:real);
{ Mencari akar persamaan  $f(x) = 0$  dengan metode secant
  K.Awal :  $x_0$  dan  $x_1$  adalah tebakan awal akar, terdefenisi nilainya
  K.Akhir: akar persamaan tercetak di layar
}
const
  epsilon = 0.000001;      { toleransi galat akar hampiran }
var
  x_sebelumnya: real;
  function f(x:real):real;
  { mengembalikan nilai  $f(x)$ . Definisi  $f(x)$  bergantung pada persoalan }
begin
  repeat
    x_sebelumnya:=x1;
    x:=x-(f(x1)*(x1 - x0)/(f(x1)-f(x0)));
    x0:=x1;
    x1:=x;
  until (ABS(x-x_sebelumnya) < epsilon);
  { x adalah hampiran akar persamaan }
  write('Hampiran akar x = ', x:10:6);
end;

```

Catatan: Program 3.9 belum menangani kasus pembagian dengan 0 atau ≈ 0 dan kasus divergen. Program harus dimodifikasi untuk menangani pembagian dengan 0 atau ≈ 0 dan kasus divergen menjadi Program 3.10 berikut.

Program 3.10 Perbaikan metode Secant

```

procedure Secant(x0, x1:real);
{ Mencari akar persamaan f(x) = 0 dengan metode secant
  K.Awal : x0 dan x1 adalah tebakan awal akar, terdefenisi nilainya
  K.Akhir: Hampiran akar tercetak di layar
}
const
  epsilon1 = 0.000001;      { toleransi galat akar hampiran }
  epsilon2 = 0.000000001;   { toleransi nilai yang hampir 0 }
  Nmaks = 30;               { jumlah maksimum lelaran }
var
  x_sebelumnya: real;
  berhenti: boolean;
  i : integer;

  function f(x:real):real;
  { mengembalikan nilai f(x). Definisi f(x) bergantung pada persoalan }

begin
  i:=0;
  repeat
    if ABS(f(x1) - f(x0)) < epsilon2 then
      berhenti:=true; { menghindari pembagian bilangan yang ≈ 0}
    else
      begin
        x_sebelumnya:=x1;
        x:=x-(f(x1)*(x1 - x0)/(f(x1)-f(x0)));
        x0:=x1;
        x1:=x;
        i:=i+1;
      end;
    until (ABS(x-x_sebelumnya) < epsilon1) or (berhenti) or (i > Nmaks);

    if berhenti then
      writeln('Pembagian dengan bilangan yang hampir 0')
    else
      if i > Nmaks then
        writeln('Divergen')
      else
        { x adalah hampiran akar persamaan }
        write('Hampiran akar x = ', x:10:6);
    {endif}
  {endif}
end;

```

Contoh 3.11

Hitunglah akar $f(x) = e^x - 5x^2$ dengan metode secant. Gunakan $\epsilon = 0.00001$. Tebakan awal akar $x_0 = 0.5$ dan $x_1 = 1$.

Penyelesaian:

Tabel lelarannya:

i	x_r	$ x_{r+1} - x_r $
0	0.500000	-
1	1.000000	0.500000
3	-0.797042	1.797042
4	10.235035	11.032077
5	-0.795942	11.030977
6	-0.794846	0.001096
7	-0.472759	0.322087
8	-0.400829	0.071930
9	-0.374194	0.026635
10	-0.371501	0.002692
11	-0.371418	0.000083
12	-0.371418	0.000000

Akar $x = -0.371418$

Ternyata lelarannya mengarah ke akar yang lain, yaitu $x = -0.371418$

■

3.5 Akar Ganda

Akar ganda (*multiple roots*) terjadi bila kurva fungsi menyinggung sumbu- x , misalnya:

- (i) $f(x) = x^3 - 5x^2 + 7x - 3 = (x-3)(x-1)(x-1)$ memiliki akar ganda dua di $x = 1$
- (ii) $f(x) = x^4 - 6x^3 + 12x^2 - 10x + 3 = (x-3)(x-1)(x-1)(x-1)$ memiliki akar ganda tiga di $x = 1$.

Pada pembahasan terdahulu kita telah menyinggung bahwa metode bagidua dan metode tertutup lainnya tidak dapat digunakan untuk mencari akar ganda, sebab fungsi tidak berubah tanda di sekeliling akar. Metode terbuka, seperti metode Newton-Raphson, sebenarnya dapat diterapkan di sini. Tetapi, bila digunakan metode Newton-Raphson untuk mencari akar ganda, kecepatan konvergensiya berjalan secara lanjar, tidak lagi kuadratis sebagaimana aslinya. Agar konvergensi metode Newton-Raphson tetap kuadratik untuk akar ganda, maka Ralston dan Rabinowitz mengusulkan alternatif metode Newton-Raphson [CHA91] sebagai berikut:

$$x_{r+1} = x_r - m \frac{f(x_r)}{f'(x_r)} \quad (\text{P.3.32})$$

dengan m adalah **bilangan multiplisitas** akar, misalnya .

- akar tunggal, $m = 1$,
- akar ganda dua, $m = 2$,
- akar ganda tiga, $m = 3$, dan seterusnya.

Namun alternatif ini tidak memuaskan karena kita perlu tahu terlebih dahulu bilangan multiplisitas akar. Disamping itu, untuk x dekat akar ganda, nilai $f(x) \approx 0$ dan juga nilai $f'(x) \approx 0$, yang dapat mengakibatkan pembagian dengan nol. Pembagian dengan nol ini dapat dihindari dengan melihat fakta bahwa $f(x)$ lebih dulu nol sebelum $f'(x)$. Jadi,

if $f(x) \approx 0$ **then** hentikan lelaran

Ralston dan Rabinowitz mengusulkan alternatif lain [CHA91]. Didefinisikan

$$u(x) = f(x) / f'(x) \quad (\text{P.3.33})$$

(Perhatikan, bentuk $u(x)$ ini memiliki akar yang sama dengan $f(x)$, sebab, jika $u(x) = 0$ maka $f(x) = 0$).

Selanjutnya,

$$x_{r+1} = x_r - \frac{u(x_r)}{u'(x_r)} \quad (\text{P.3.34})$$

yang dalam hal ini,

$$u'(x) = [f(x)/f'(x)]' = \frac{f'(x)f'(x) - f''(x)f(x)}{[f'(x)]^2} = \frac{[f'(x)]^2 - f''(x)f(x)}{[f'(x)]^2}$$

$$\quad (\text{P.3.35})$$

sehingga

$$x_{r+1} = x_r - \frac{f(x_r)/f'(x_r)}{\frac{[f'(x_r)]^2 - f''(x_r)f(x_r)}{[f'(x_r)]^2}}$$

atau

$$x_{r+1} = x_r - \frac{f(x_r)f'(x_r)}{[f'(x_r)]^2 - f''(x_r)f(x_r)} \quad (\text{P.3.36})$$

Meskipun rumus (P.3.36) ini lebih disukai untuk akar ganda, namun ia kurang mangkus sebab memerlukan lebih banyak komputasi daripada metode Newton-Raphson yang baku. Rumus (P.3.36) berlaku secara umum, yaitu ia tetap dapat dipakai untuk pencarian akar tidak ganda sekalipun.

Metode secant juga dapat dimodifikasi dengan menyulihkan $u(x) = f(x) / f'(x)$ ke dalam rumusnya. Rumus yang dihasilkan adalah [CHA91]:

$$x_{r+1} = x_r - \frac{u(x_r)(x_{r-1} - x_r)}{u(x_{r-1}) - u(x_r)} \quad (\text{P.3.37})$$

Contoh 3.12

Hitung akar $f(x) = x^3 - 5x^2 + 7x - 3$ dengan metode Newton-Raphson baku dan metode Newton-Raphson yang diperbaiki. Tebakan awal $x_0 = 0$. [CHA91]

Penyelesaian:

$$\begin{aligned} f(x) &= x^3 - 5x^2 + 7x - 3 \\ f'(x) &= 3x^2 - 10x + 7 \\ f''(x) &= 6x - 10 \end{aligned}$$

Dengan metode Newton-Raphson baku:

$$x_{r+1} = \frac{x_r - x_r^3 - 5x_r^2 + 7x_r - 3}{(3x_r^3 - 10x_r + 7)}$$

Dengan metode Newton-Raphson yang dimodifikasi:

$$x_{r+1} = x_r - \frac{(x_r^3 - 5x_r^2 + 7x_r - 3)(3x_r^2 - 10x_r + 7)}{(3x_r^2 - 10x_r + 7)^2 - (6x_r - 10)(x_r^3 - 5x_r^2 + 7x_r - 3)}$$

Tabel lelarannya adalah:

Metode Newton Raphson baku		Metode Newton Raphson yang dimodifikasi	
r	x_r	r	x_r
0	0.000000000	0	0.000000000
1	0.428571429	1	1.105263158
2	0.685714286	2	1.003081664
3	0.832865400	3	1.000002382
4	0.913328983		
5	0.955783293		
6	0.977655101		

Lelaran konvergen ke akar $x = 1$. Terlihat dari tabel di atas bahwa metode Newton yang dimodifikasi memiliki jumlah lelaran lebih sedikit. ■

3.6 Akar-Akar Polinom

Bentuk baku polinom derajat $\leq n$ adalah

$$p(x) = a_0 + a_1 x + a_2 x^2 + \dots + a_n x^n \quad (\text{P.3.38})$$

dengan a_i adalah konstanta riil, $i = 0, 1, 2, \dots, n$, dan $a_n \neq 0$. Polinom $p(x)$ memiliki n buah akar, baik akar nyata maupun akar kompleks. Akar kompleks muncul dalam pasangan konyugasi, $w = u + vi$ dan $w = u - vi$, dengan $i = \sqrt{-1}$. Contohnya, polinom $p(x) = 5 - 4x + x^2$ mempunyai akar $2 + i$ dan $2 - i$.

Semua metode pencarian akar dapat diterapkan pada polinom. Misalnya dengan metode Newton-Raphson,

$$x_{r+1} = x_r - \frac{p(x_r)}{p'(x_r)} \quad (\text{P.3.39})$$

Masalahnya, evaluasi polinom, $p(x_r)$ dan $p'(x_r)$ membutuhkan banyak operasi perkalian (termasuk perpangkatan). Semakin tinggi derajat polinomnya tentu semakin banyak operasi perkalian yang diperlukan, yang berarti semakin besar rambatan galat pembulatannya (ingat, komputer menggunakan bilangan titik-kambang). Karena itu, harus dicari suatu metode perhitungan polinom dengan sedikit operasi perkalian.

3.6.1 Metode Horner untuk Evaluasi Polinom

Menghitung langsung $p(x)$ untuk $x = 1$ tidak mangkus sebab melibatkan banyak operasi perkalian. Metode Horner, atau disebut juga metode perkalian bersarang (*nested multiplication*) menyediakan cara perhitungan polinom dengan sedikit operasi perkalian. Dalam hal ini, polinom $p(x)$ dinyatakan sebagai perkalian bersarang

$$p(x) = a_0 + x(a_1 + x(a_2 + x(a_3 + \dots + x(a_{n-1} + a_n x)))) \quad (\text{P.3.40})$$

Contoh 3.12

Nyatakan $p(x) = 8 + 6x + 2x^2 + 5x^3$ dalam bentuk perkalian bersarang.

Penyelesaian:

$$\begin{aligned} p(x) &= 8 + 6x + 2x^2 + 5x^3 && \text{(6 buah perkalian)} \\ &= 8 + x(6 + x(2 + 5x)) && \text{(hanya 3 buah perkalian)} \end{aligned}$$

Perhitungan $p(x)$ untuk $x = 2$ adalah

$$p(2) = 8 + 2(6 + 2(2 + 5 \cdot 2)) = 68 \quad \blacksquare$$

Metode perkalian bersarang untuk menghitung $p(t)$ seringkali dinyatakan dalam bentuk tabel Horner berikut:

t	a_n	a_{n-1}	a_{n-2}	a_2	a_1	a_0
	tb_n	tb_{n-1}	tb_3	tb_2	tb_1	tb_0

$b_n = a_n$	$b_{n-1} = a_{n-1} + tb_n$	$b_{n-2} = a_{n-2} + tb_{n-1}$	$b_2 = a_2 + tb_3$	$b_1 = a_1 + tb_2$	$b_0 = a_0 + tb_1$
-------------	----------------------------	--------------------------------	--------------------	--------------------	--------------------

—————
polinom sisa
—————

Hasil evaluasi: $p(t) = b_0$

Jadi, untuk Contoh 3.12 di atas,

$$\begin{array}{r|cccc}
 2 & 5 & 2 & 6 & 8 \\
 & 10 & 24 & 60 \\
 \hline
 & 5 & 12 & 30 & 68 = p(2)
 \end{array}$$

dan menghasilkan polinom sisa $5x^2 + 12x + 30$.

Program 3.11 Menghitung $p(x)$ untuk $x = t$ dengan metode Horner

```
{ Dalam program utama telah didefinisikan:  
  const n=...; {derajat polinom}  
  var a, b, c: array[1..n] of real  
 }  
  
function p(t:real):real;  
{ menghitung p(t) dengan metode Horner}  
var  
  k: integer;  
begin  
  b[n]:=a[n];  
  for k:=n-1 downto 0 do  
    b[k]:=a[k] + b[k+1]*t;  
  {end for}  
  p:=b[0];  
end;
```

3.6.2 Pencarian Akar-akar Polinom

Proses perhitungan $p(x)$ untuk $x = t$ dengan menggunakan metode Horner sering dinamakan *pembagian sintetis* $p(x):(x - t)$, menghasilkan $q(x)$ dan sisa b_0 ,

$$\left[\frac{p(x)}{(x-t)} = q(x) \right] + \text{sisa } b_0 \quad (\text{P.3.41})$$

atau

$$p(x) = b_0 + (x-t) q(x) \quad (\text{P.3.42})$$

yang dalam hal ini,

$$q(x) = b_n x^{n-1} + b_{n-1} x^{n-2} + \dots + b_3 x^2 + b_2 x + b_1 \quad (\text{P.3.43})$$

Untuk Contoh 3.12 di atas,

$$p(x) = 8 + 6x + 2x^2 + 5x^3 = 68 + (x-2)(5x^2 + 12x + 30)$$

Jika t adalah hampiran akar polinom $p(x)$ maka

$$p(t) = b_0 + (t - t) q(t) = b_0 + 0 = b_0$$

(Perhatikan, jika t akar sejati, maka $b_0 = 0$)

Akar-akar lain dari $p(x)$ dapat dicari dari polinom $q(x)$ sebab setiap akar $q(x)$ juga adalah akar $p(x)$. Proses reduksi polinom ini disebut *deflasi* (*deflation*). Koefisien-koefisien $q(x)$, yaitu $b_n, b_{n-1}, \dots, b_3, b_2, b_1$ dapat ditemukan langsung dari tabel Horner,

$$\begin{aligned}b_n &= a_n \\b_{n-1} &= a_{n-1} + tb_n \\b_{n-2} &= a_{n-2} + tb_{n-1} \\&\dots \\b_2 &= a_2 + tb_3 \\b_1 &= a_1 + tb_2\end{aligned}$$

Algoritmanya,

```
b[n]:=a[n];
for k:=n-1 downto 1 do
    b[k]:=a[k] + t*b[k+1]
{endfor}
```

Misalkan akar polinom dihitung dengan metode Newton-Raphson,

$$x_{r+1} = x_r - \frac{p(x_r)}{p'(x_r)}$$

maka proses pencarian akar secara deflasi dapat dirumuskan dalam langkah 1 sampai 4 berikut ini.

Langkah 1:

Menghitung $p(x_r)$ dapat dilakukan secara mangkus dengan metode Horner.

Misalkan $t = x_r$ adalah hampiran akar polinom $p(x)$,

$$p(x) = b_0 + (x - x_r) q(x)$$

Perhitungan $p(x_r)$ menghasilkan

$$p(x_r) = b_0 + (x_r - x_r) q(x_r) = b_0$$

Nilai $p(x_r) = b_0$ ini dapat dihitung dengan function p

Langkah 2:

Menghitung $p'(x_r)$ secara mangkus:

Misalkan $t = x_r$ adalah hampiran akar polinom $p(x)$,

$$p(x) = b_0 + (x - x_r) q(x)$$

Turunan dari p adalah

$$p'(x) = 0 + 1 \cdot q(x) + (x - x_r) q'(x) = q(x) + (x - x_r) q'(x)$$

sehingga

$$p'(x_r) = q(x_r) + (x_r - x_r) q'(x_r) = q(x_r)$$

Koefisien polinom $q(x)$ dapat ditentukan dari langkah 1. Selanjutnya $q(x_r)$ dapat dihitung dengan function q berikut:

Program 3.12 Menghitung $p'(t) = q(t)$

```

function q(t:real):real;
{ menghitung  $p'(t)=q(t)$  dengan metode Horner}
var
  k : integer;
begin
  c[n]:=b[n];
  for k:=n-1 downto 1 do
    c[k]:=b[k] + t*c[k+1]
  {endfor}
  q:=c[1];
end;

```

Langkah 3:

$$x_{r+1} = x_r - \frac{p(x_r)}{p'(x_r)}$$

Langkah 4:

Ulangi langkah 1, 2 dan 3 di atas sampai $|x_{r+1} - x_r| < \varepsilon$.

Program 3.13 Prosedur Newton-Raphson untuk menghitung akar polinom

```
procedure Newton_Raphson_untuk_polinom(n:integer; x:real);
{ procedure Newton-Raphson untuk menghitung akar polinom p(x) yang
berderajat n dengan tebakan awal akar x
K.Awal : n adalah derajat polinom; x adalah tebakan awal akar;
kedua nilai sudah terdefinisi
K.Akhir: Hampiran akar polinom tercetak di layar.
}
const
  epsilon = 0.0000001;
var
  x_sebelumnya: real;

  function p(t:real):real;
  {menghitung p(t) dengan metode Horner}
  var
    k: integer;
  begin
    b[n]:=a[n];
    for k:=n-1 downto 0 do
      b[k]:=a[k] + b[k+1]*t;
    {end for}
    p:=b[0];
  end {p};

  function q(t:real):real;
  { menghitung p'(t)=q(t) dengan metode Horner}
  var
    k : integer;
  begin
    c[n]:=b[n];
    for k:=n-1 downto 1 do
      c[k]:=b[k] + t*c[k+1];
    {end for}
    q:=c[1];
  end {q} ;

begin
  repeat
    x_sebelumnya:=x;
    x:=x - p(x)/q(x);
  until ABS(x - x_sebelumnya) < epsilon;
  { x adalah akar polinom }
  writeln('Hampiran akar = ', x:10:6);
end;
```

Program 3.13 ini hanya menemukan satu buah akar polinom. Untuk mencari seluruh akar nyata polinom, harus dilakukan proses deflasi. Setelah akar pertama x_1 diperoleh, polinom $p(x)$ dapat ditulis sebagai

$$p(x) = (x - x_1) q(x) + b_0$$

yang dalam hal ini $q(x) = b_n x^{n-1} + b_{n-1} x^{n-2} + \dots + b_3 x^2 + b_2 x + b_1$. Koefisien-koefisien $q(x)$, yaitu $b_n, b_{n-1}, \dots, b_3, b_2, b_1$ diperoleh di akhir Program 3.11, yang telah tersimpan pada elemen larik $b[n - 1], b[n - 2], \dots, b[2], b[1]$. Selanjutnya panggil Program 3.13 untuk mencari akar polinom $q(x)$ yang berderajat $n-1$ dengan tebakan awalnya dapat digunakan x_1 (atau boleh bilangan lain). Setelah akar kedua x_2 diperoleh, polinom $p(x)$ dapat ditulis sebagai

$$q(x) = (x - x_2) r(x) + b_1$$

yang dalam hal ini

$$r(x) = b_{n-1} x^{n-2} + b_{n-2} x^{n-3} + \dots + b_3 x^2 + b_2 x + b_1.$$

$r(x)$ adalah polinom derajat $n - 2$ dengan koefisien $b_{n-1}, b_{n-2}, \dots, b_3, b_2, b_1$ diperoleh di akhir Program 3.11 pada elemen larik $b[n-1], b[n-2], \dots, b[2], b[1]$. Selanjutnya panggil kembali Program 3.13 untuk mencari akar polinom $r(x)$ yang berderajat $n - 2$ dengan tebakan awalnya dapat digunakan x_2 . Begitu seterusnya sampai polinom sisa yang ditemukan berderajat 0. Atau, dapat juga sampai polinom sisa berderajat dua.

Algoritma selengkapnya adalah:

```

write('Tebakan awal untuk akar pertama: ');
readln(x);
repeat
    Newton_Raphson_untuk_polinom(n, x);

    { salin koefisien b[n], b[n-1], ..., b[1] ke dalam
      a[n-1], a[n-2], ..., a[0] untuk pencarian akar selanjutnya}
    for i:=n downto 1 do
        a[i-1]:=b[i];
    {endfor}
    n:=n-1;    { derajat polinom sisa berkurang satu }
until n=0;

```

Contoh 3.14

[GER85] Temukan seluruh akar nyata polinom

$$p(x) = x^5 - 12x^4 - 293x^3 + 3444x^2 + 20884x - 240240$$

dengan tebakan awal akar $x_0=11$.

Penyelesaian:

Panggil prosedur

```
Newton_Raphson_untuk_polinom(5, 11);
```

untuk mencari akar polinom $p(x)$ berderajat 5 dengan tebakan awal akar $x_0=11$.

Diperoleh akar pertama, yaitu $x_1 = 13.99990$

$$\text{Deflasi} \rightarrow p(x) = (x - x_1) q(x) + b_0$$

$$\text{yang dalam hal ini } q(x) = x^4 + 1999895x^3 - 2650015x^2 - 2659927x + 17160.13$$

Panggil prosedur

```
Newton_Raphson_untuk_polinom(4, 13.99990);
```

untuk mencari akar polinom $q(x)$ berderajat 4 dengan tebakan awal $x_0=13.99990$

Diperoleh akar kedua $x_2 = 12.00016$

$$\text{Deflasi} \rightarrow q(x) = (x - x_2) r(x) + b_1$$

$$\text{yang dalam hal ini } r(x) = x^3 + 1400005x^2 - 9699867x - 1429992$$

Panggil prosedur

```
Newton_Raphson_untuk_polinom(3, 12.00016);
```

untuk mencari akar polinom $r(x)$ berderajat 3 dengan tebakan awal akar $x_0=12.00016$

Diperoleh akar $x_3 = 9.999949$

$$\text{Deflasi} \rightarrow r(x) = (x - x_3) s(x) + b_2$$

$$\text{yang dalam hal ini } s(x) = x^2 + 2396998x + 1429999$$

Demikian seterusnya sampai kita temukan akar keempat dan akar kelima sebagai berikut:

$$x_4 = -12.99991$$

$$x_5 = -11.00006$$

■

3.6.3 Lokasi Akar Polinom

Metode Newton-Raphson memerlukan tebakan awal akar. Bagaimanakah menemukan tebakan awal akar yang bagus untuk polinom? Misalkan akar-akar diberi indeks dan diurut menaik sedemikian sehingga

$$|x_1| \leq |x_2| \leq |x_3| \leq \dots \leq |x_n|$$

Tebakan awal untuk akar terkecil x_1 menggunakan hampiran

$$\begin{aligned} a_0 + a_1 x &\approx 0 \\ x &\approx -a_0/a_1 \end{aligned} \tag{P.3.44}$$

yang dapat dijadikan sebagai tebakan awal untuk menemukan x_1

Tebakan awal untuk akar terbesar x_n menggunakan hampiran

$$\begin{aligned} a_{n-1}x^{n-1} + a_n x^n &\approx 0 \\ x &\approx -a_{n-1}/a_n \end{aligned} \tag{P.3.45}$$

yang dapat dijadikan sebagai tebakan awal untuk menemukan x_n

Contoh 3.15

[NOB72] Tentukan tebakan awal untuk mencari akar polinom $x^2 - 200x + 1 = 0$.

Penyelesaian:

Tebakan awal untuk akar terkecil adalah

$$x_0 = -1/(-200) = 1/200$$

Tebakan awal untuk akar terbesar adalah

$$x_0 = -(-200)/1 = 200$$

■

3.7 Sistem Persamaan Nirlanjard

Di dalam dunia nyata, umumnya model matematika muncul dalam bentuk sistem persamaan. Persamaan yang diselesaikan tidak hanya satu, tetapi dapat lebih dari satu, sehingga membentuk sebuah sistem yang disebut sistem persamaan nirlanjard. Bentuk umum sistem persamaan nirlanjard dapat ditulis sebagai berikut:

$$\begin{aligned}
f_1(x_1, x_2, \dots, x_n) &= 0 \\
f_2(x_1, x_2, \dots, x_n) &= 0 \\
&\dots \\
f_n(x_1, x_2, \dots, x_n) &= 0
\end{aligned} \tag{P.3.46}$$

Penyelesaian sistem ini adalah himpunan nilai x simultan, x_1, x_2, \dots, x_n , yang memenuhi seluruh persamaan. Sistem persamaan dapat diselesaikan secara berlelar dengan metode lelaran titik-tetap atau dengan metode Newton-Raphson.

3.7.1 Metode Lelaran Titik-Tetap

Prosedur lelarannya titik-tetap untuk sistem dengan dua persamaan nirlanjar:

$$\begin{aligned}
x_{r+1} &= g_1(x_r, y_r) \\
y_{r+1} &= g_2(x_r, y_r) \\
r &= 0, 1, 2, \dots
\end{aligned} \tag{P.3.47}$$

Metode lelaran titik-tetap seperti ini dinamakan metode **lelaran Jacobi**. Kondisi berhenti (konvergen) adalah

$$|x_{r+1} - x_r| < \epsilon \text{ dan } |y_{r+1} - y_r| < \epsilon$$

Kecepatan konvergensi lelaran titik-tetap ini dapat ditingkatkan. Nilai x_{r+1} yang baru dihitung langsung dipakai untuk menghitung y_{r+1} . Jadi,

$$\begin{aligned}
x_{r+1} &= g_1(x_r, y_r) \\
y_{r+1} &= g_2(x_{r+1}, y_r) \\
r &= 0, 1, 2, \dots
\end{aligned} \tag{P.3.48}$$

Metode lelaran titik-tetap seperti ini dinamakan metode **lelaran Seidel**. Kondisi berhenti (konvergen) adalah

$$|x_{r+1} - x_r| < \epsilon \text{ dan } |y_{r+1} - y_r| < \epsilon$$

Untuk fungsi dengan tiga persamaan nirlanjar, lelaran Seidel-nya adalah

$$\begin{aligned}
x_{r+1} &= g_1(x_r, y_r, z_r) \\
y_{r+1} &= g_2(x_{r+1}, y_r, z_r) \\
z_{r+1} &= g_3(x_{r+1}, y_{r+1}, z_r) \\
r &= 0, 1, 2, \dots
\end{aligned} \tag{P.3.49}$$

Kondisi berhenti (konvergen) adalah

$$|x_{r+1} - x_r| < \epsilon \text{ dan } |y_{r+1} - y_r| < \epsilon \text{ dan } |z_{r+1} - z_r| < \epsilon$$

Contoh 3.16

[CHA91] Selesaikan sistem persamaan nirlanjar berikut ini,

$$\begin{aligned}f_1(x, y) &= x^2 + xy - 10 = 0 \\f_2(x, y) &= y + 3xy^2 - 57 = 0\end{aligned}$$

(Akar sejatinya adalah $x = 2$ dan $y = 3$)

Penyelesaian:

Prosedur lararan titik-tetapnya adalah

$$x_{r+1} = \frac{10 - x_r^2}{y_r}$$

$$y_{r+1} = 57 - 3x_{r+1}y_r^2$$

Berikan tebakan awal $x_0 = 1.5$ dan $y_0 = 3.5$ dan $\epsilon = 0.000001$

Tabel lararannya:

r	x	y	$ x_{r+1} - x_r $	$ y_{r+1} - y_r $
0	1.500000	3.500000	-	-
1	2.214286	-24.375000	0.714286	27.875000
2	-0.209105	429.713648	2.423391	454.088648
3	0.023170	-12778.041781	0.232275	13207.755429
...				

Ternyata lararannya divergen!

Sekarang kita ubah persamaan prosedur lararannya menjadi

$$x_{r+1} = \sqrt{10 - x_r y_r}$$

$$y_{r+1} = \sqrt{\frac{57 - y_r}{3x_{r+1}}}$$

Tebakan awal $x_0 = 1.5$ dan $y_0 = 3.5$ dan $\epsilon = 0.000001$

Hasilnya,

r	x	y	$ x_{r+1} - x_r $	$ y_{r+1} - y_r $
0	1.500000	3.500000	-	-
1	2.179449	2.860506	0.679449	0.639494
2	1.940534	3.049551	0.238916	0.189045
3	2.020456	2.983405	0.079922	0.066146

4	1.993028	3.005704	0.027428	0.022300
5	2.002385	2.998054	0.009357	0.007650
6	1.999185	3.000666	0.003200	0.002611
7	2.000279	2.999773	0.001094	0.000893
8	1.999905	3.000078	0.000374	0.000305
9	2.000033	2.999973	0.000128	0.000104
10	1.999989	3.000009	0.000044	0.000036
11	2.000004	2.999997	0.000015	0.000012
12	1.999999	3.000001	0.000005	0.000004
13	2.000000	3.000000	0.000002	0.000001
14	2.000000	3.000000	0.000001	0.000000

Akar $x = 2.000000$
 $y = 3.000000$

■

Contoh 3.15 ini memperlihatkan bahwa konvergensi metode lelaran titik-tetap sangat bergantung pada bentuk persamaan prosedur lelaran dan tebakan awal. Syarat perlu kekonvergenan untuk sistem dengan dua persamaan nirlanjar adalah

$$\left| \frac{\partial g_1}{\partial x} \right| + \left| \frac{\partial g_1}{\partial y} \right| < 1$$

dan

$$\left| \frac{\partial g_2}{\partial x} \right| + \left| \frac{\partial g_2}{\partial y} \right| < 1$$

di dalam selang yang mengandung titik tetap (p, q).

3.7.2 Metode Newton-Raphson

Ingatlah kembali bahwa metode Newton-Raphson dapat diturunkan dari deret Taylor,

$$f(x_{r+1}) \approx f(x_r) + (x_{r+1} - x_r) f'(x_r)$$

dan karena persoalan mencari akar, maka $f(x_{r+1}) = 0$, sehingga

$$0 = f(x_r) + (x_{r+1} - x_r) f'(x_r)$$

atau

$$x_{r+1} = x_r - \frac{f(x_r)}{f'(x_r)} \quad , f'(x_r) \neq 0$$

Untuk fungsi dengan dua peubah, deret Taylor orde pertama dapat dituliskan untuk masing-masing persamaan sebagai

$$u_{r+1} = u_r + (x_{r+1} - x_r) \frac{\partial u_r}{\partial x} + (y_{r+1} - y_r) \frac{\partial u_r}{\partial y} \quad (\text{P.3.50})$$

dan

$$v_{r+1} = v_r + (x_{r+1} - x_r) \frac{\partial v_r}{\partial x} + (y_{r+1} - y_r) \frac{\partial v_r}{\partial y} \quad (\text{P.3.51})$$

Karena persoalan mencari akar, maka $u_{r+1} = 0$ dan $v_{r+1} = 0$, untuk memberikan

$$\frac{\partial u_r}{\partial x} x_{r+1} + \frac{\partial u_r}{\partial y} y_{r+1} = -u_r + x_r \frac{\partial u_r}{\partial x} + y_r \frac{\partial u_r}{\partial y}$$

$$\frac{\partial v_r}{\partial x} x_{r+1} + \frac{\partial v_r}{\partial y} y_{r+1} = -v_r + x_r \frac{\partial v_r}{\partial x} + y_r \frac{\partial v_r}{\partial y}$$

Dengan sedikit manipulasi aljabar, kedua persamaan terakhir ini dapat dipecahkan menjadi

$$x_{r+1} = x_r - \frac{u_r \frac{\partial v_r}{\partial y} + v_r \frac{\partial u_r}{\partial y}}{\frac{\partial u_r}{\partial x} \frac{\partial v_r}{\partial y} - \frac{\partial u_r}{\partial y} \frac{\partial v_r}{\partial x}} \quad (\text{P.3.52})$$

dan

$$y_{r+1} = y_r + \frac{u_r \frac{\partial v_r}{\partial x} - v_r \frac{\partial u_r}{\partial x}}{\frac{\partial u_r}{\partial x} \frac{\partial v_r}{\partial y} - \frac{\partial u_r}{\partial y} \frac{\partial v_r}{\partial x}} \quad (\text{P.4.53})$$

Penyebut dari masing-masing persamaan ini diacu sebagai **determinan Jacobi** dari sistem tersebut [CHA91]. Metode Newton-Raphson dapat dirampatkan (*generalization*) untuk sistem dengan n persamaan.

Contoh 3.17

[CHA91] Gunakan metode Newton-Raphson untuk mencari akar

$$\begin{aligned}f_1(x, y) &= u = x^2 + xy - 10 = 0 \\f_2(x, y) &= v = y + 3xy^2 - 57 = 0\end{aligned}$$

dengan tebakan awal $x_0=1.5$ dan $y_0=3.5$

Penyelesaian:

$$\frac{\partial u_o}{\partial x} = 2x + y = 2(1.5) + 3.5 = 6.5$$

$$\frac{\partial u_o}{\partial y} = x = 1.5$$

$$\frac{\partial v_o}{\partial x} = 3y^2 = 3(3.5)^2 = 36.75$$

$$\frac{\partial v_o}{\partial y} = 1 + 6xy = 1 + 6(1.5) = 32.5$$

Determinan Jacobi untuk lelaran pertama adalah

$$6.5(32.5) - 1.5(36.75) = 156.125$$

Nilai-nilai fungsi dapat dihitung dari tebakan awal sebagai

$$u_0 = (1.5)^2 + 1.5(3.5) - 10 = -2.5$$

$$v_0 = (3.5)^2 + 3(1.5)(3.5)^2 - 57 = 1.625$$

Nilai x dan y pada lelaran pertama adalah

$$x_0 = \frac{1.5 - (-2.5)(32.5) - 1.625(1.5)}{156.125} = 2.03603$$

dan

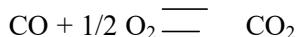
$$y_0 = \frac{3.5 - (-2.5)(36.75) - 1.625(6.5)}{156.125} = 2.84388$$

Apabila lelarannya diteruskan, ia konvergen ke akar sejati $x = 2$ dan $y = 3$. ■

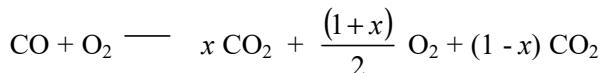
Seperti halnya metode lelaran titik-tetap, metode Newton-Raphson mungkin saja divergen jika tebakan awal tidak cukup dekat ke akar. Penggambaran kurva masing-masing persamaan secara grafik dapat membantu pemilihan tebakan awal yang bagus.

3.8 Contoh Soal Terapan

Dalam suatu proses Teknik Kimia, campuran karbon monoksida dan oksigen mencapai kesetimbangan pada suhu 300°K dan tekanan 5 atm. Reaksi teoritisnya adalah



Reaksi kimia yang sebenarnya terjadi dapat ditulis sebagai



Persamaan kesetimbangan kimia untuk menentukan fraksi mol CO yang tersisa, yaitu x , ditulis sebagai

$$K_p = \frac{(1-x)(3+x)^{\frac{1}{2}}}{x(x+1)^{\frac{1}{2}} p^{\frac{1}{2}}} , \quad 0 < x < 1$$

yang dalam hal ini, $K_p = 3.06$ adalah tetapan kesetimbangan untuk reaksi $\text{CO} + \frac{1}{2} \text{O}_2$ pada 3000°K dan $P = 5$ atm. Tentukan nilai x dengan metode regula falsi yang diperbaiki.

Penyelesaian:

Persoalan ini memang lebih tepat diselesaikan dengan metode tertutup karena x adalah fraksi mol yang nilainya terletak antara 0 dan 1.

Fungsi yang akan dicari akarnya dapat ditulis sebagai

$$f(x) = \frac{(1-x)(3+x)^{\frac{1}{2}}}{x(x+1)^{\frac{1}{2}} p^{\frac{1}{2}}} - K_p , \quad 0 < x < 1$$

dengan $K_p = 3.06$ dan $P = 5$ atm.

Selang yang mengandung akar adalah $[0.1, 0.9]$. Nilai fungsi di ujung-ujung selang adalah

$$f(0.1) = 3.696815 \text{ dan } f(0.9) = -2.988809$$

yang memenuhi $f(0.1)f(0.9) < 0$.

Tabel lelarannya adalah:

<i>r</i>	<i>a</i>	<i>c</i>	<i>b</i>	<i>f(a)</i>	<i>f(c)</i>	<i>f(b)</i>	Selang baru	Lebarnya
0	0.100000	0.542360	0.900000	3.696815	-2.488120	-2.988809	[a, c]	0.442360
1	0.100000	0.288552	0.542360	1.848407	-1.298490	-2.488120	[a, c]	0.188552
2	0.100000	0.178401	0.288552	0.924204	0.322490	-1.298490	[c, b]	0.110151
3	0.178401	0.200315	0.288552	0.322490	-0.144794	-1.298490	[a, c]	0.021914
4	0.178401	0.193525	0.200315	0.322490	-0.011477	-0.144794	[a, c]	0.015124
5	0.178401	0.192520	0.193525	0.161242	0.009064	-0.011477	[c, b]	0.001005
6	0.192520	0.192963	0.193525	0.009064	-0.000027	-0.011477	[a, c]	0.000443
7	0.192520	0.192962	0.192963	0.009064	-0.000000	-0.000027	[a, c]	0.000442

Hampiran akar $x = 0.192962$

Jadi, setelah reaksi berlangsung, fraksi mol CO yang tersisa adalah 0.192962.

Hal-hal kecil membentuk kesempurnaan,
tetapi kesempurnaan bukanlah hal yang kecil.
(Michael Angello)

Soal Latihan

1. Tahun 1225 Leonardo da Pisa mencari akar persamaan

$$f(x) = x^3 + 2x^2 + 10x - 20 = 0$$

dan menemukan $x = 1.368808107$. Tidak seorang pun yang mengetahui cara Leonardo menemukan nilai ini. Sekarang, rahasia itu dapat dipecahkan dengan metode lelaran titik-tetap. Bentuklah semua kemungkinan prosedur lelaran titik-tetap dari $f(x) = 0$, lalu dengan memberikan sembarang tebakan awal (misalnya $x_0 = 1$), tentukan prosedur lelaran mana yang menghasilkan akar persamaan yang ditemukan Leonardo itu.

2. Apa yang terjadi jika persamaan $x^2 = 2$ diatur sebagai $x_{r+1} = 2/x_r$ dan metode lelaran titik-tetap digunakan untuk menemukan akar kuadrat dari 2?
3. Tentukan titik potong kurva $f(x) = e^x$ dengan kurva $g(x) = \sin(x)$ dengan metode Newton-Raphson.

4. Tentukan selang sehingga prosedur lelaran $x_{r+1} = x_r/2 - \cos(2x_r)$ konvergen di dalam selang itu (x dalam radian)
5. Perlihatkan bahwa semua akar $x^{20} - 1 = 0$ berkondisi baik.
6. Gunakan metode
 - (i) bagidua
 - (ii) regula-falsi

untuk menemukan akar persamaan Leonardo dalam selang [1, 1.5], dan juga dengan metode

- (iii) Newton-Raphson, $x_0 = 1$
- (iv) secant, $x_0 = 1$, $x_1 = 1.5$

Untuk semua metode, $\epsilon = 10^{-6}$

7. Diketahui lingkaran $x^2 + y^2 = 2$ dan hiperbola $x^2 - y^2 = 1$. Tentukan titik potong kedua kurva dengan metode lelaran titik-tetap (Soal ini adalah mencari solusi sistem persamaan nirlanjar).
8. Diberikan prosedur lelaran $x_{r+1} = 0.9x_r$ dan nilai awal $x_0 = 1$. Menurut orang matematik, untuk $r = 1, 2, 3, \dots$

$$x_r = (0.9)x_{r-1} = (0.9)(0.9)x_{r-2} = \dots = (0.9)^r x_0 = (0.9)^r$$

dan menyimpulkan bahwa

$$\lim_{r \rightarrow \infty} x_r = 0$$

Bagi orang numerik, harus ditanyakan terlebih dahulu berapa banyak angka bena yang digunakan.

Berapakah

$$\lim_{r \rightarrow \infty} x_r$$

menurut orang numerik bila digunakan:

- (a) satu angka bena
- (b) dua angka bena

(Petunjuk: lakukan sejumlah lelaran tanpa memprogramnya dengan komputer. Setiap perhitungan harus taat asas dengan jumlah angka benar yang digunakan)

9. Misalkan metode bagidua akan digunakan untuk menemukan akar dalam selang [-1,5]. Berapa kali selang harus dibagi dua untuk menjamin bahwa hampiran c_r mempunyai ketelitian 0.5×10^{-9} .
10. Dapatkah metode Newton-Raphson digunakan untuk memecahkan:
 - (i) $f(x) = 0$ jika $f(x) = x^{1/3}$
 - (ii) $f(x) = 0$ jika $f(x) = (x-3)^{1/2}$ dan tebakan awal $x_0 = 4$?

Mengapa?

11. Bagaimana bentuk prosedur lelaran Newton-Raphson untuk menghitung e (bilangan natural?)
12. Nilai arcsin 1 dapat dihitung dari persamaan

$$\sin(x) = 1 \quad \Leftrightarrow \quad \sin(x) - 1 = 0$$

Persamaan ini mempunyai akar ganda di $x = \pi/2$ (periksa!). Tentukan akar ganda itu dengan:

- (a) metode Newton-Raphson baku
 - (b) metode Newton-Raphson dengan faktor multiplisitas akar
 - (c) metode Newton-Raphson yang dimodifikasi untuk menangani kasus akar ganda
13. Gunakan metode Newton-Raphson untuk menghitung $(47)^{1/4}$ sampai enam angka benar.
 14. Perlihatkan bahwa bial metode Newton-Raphson diterapkan pada bermacam-macam $f(x)$ di bawah ini, maka prosedur lelarannya akan mengarah pada pencarian $\sqrt[a]{a}$, untuk $a > 0$.
 - (i) $f(x) = x^2 - a$
 - (ii) $f(x) = 1 - a/x^2$
 15. Misalkan $f(x) = \cos(x)$
 - (a) Tentukan prosedur lelaran Newton-Raphsonnya

- (b) Jika kita ingin menghitung akar $x = 3\pi/2$, dapatkah kita gunakan tebakan awal $x_0 = 3$? Mengapa?
- (c) Seperti (b), bagaimana jika $x_0 = 5$? Mengapa?

Bab 4

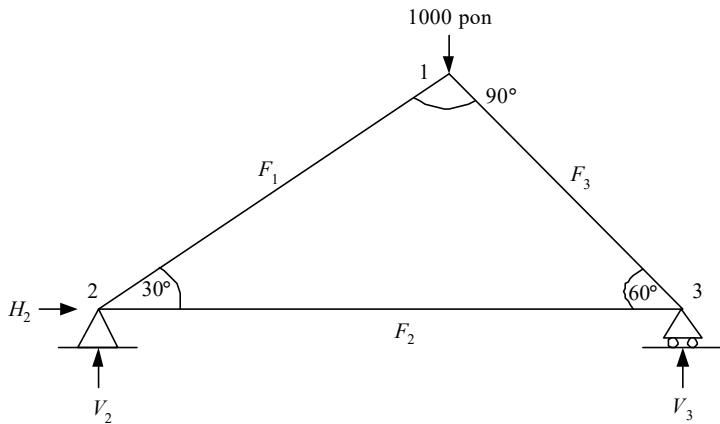
Solusi Sistem Persamaan Lanjar

Saya tidak dapat memastikan bahwa perubahan akan memperbaiki sesuatu, tetapi
saya dapat memastikan bahwa untuk menjadi
lebih baik sesuatu harus berubah
(George C. Lichtenberg)

Dalam praktek rekayasa, perilaku sistem dimodelkan dalam persamaan matematika. Seringkali jumlah persamaan tersebut lebih dari satu dan harus diselesaikan secara serempak atau simultan. Di dalam Bab 3 sudah diberikan contoh penyelesaian sistem dengan dua buah persamaan nirlanjar. Jika sistem persamaan yang dihasilkan berbentuk aljabar lanjar (*linier*), maka diperlukan teknik penyelesaian yang lain. Contoh di bawah ini memberi gambaran sistem persamaan lanjar dalam bidang rekayasa sipil [CHA91].

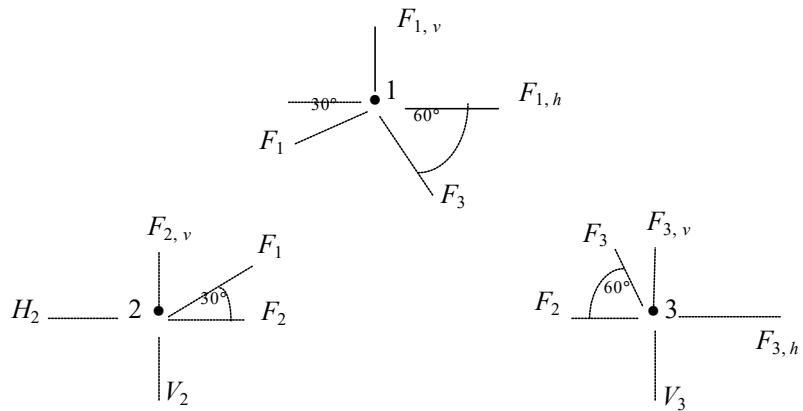
Misalkan seorang insinyur Teknik Sipil merancang sebuah rangka statis yang berbentuk segitiga (Gambar 4.1). Ujung segitiga yang bersudut 30° bertumpu pada sebuah penyangga statis, sedangkan ujung segitiga yang lain bertumpu pada penyangga beroda.

Rangka mendapat gaya eksternal sebesar 1000 pon. Gaya ini disebar ke seluruh bagian rangka. Gaya F menyatakan tegangan atau kompresi pada anggota rangka. Reaksi eksternal (H_2 , V_2 , dan V_3) adalah gaya yang mencirikan bagaimana rangka berinteraksi dengan permukaan pendukung. Engsel pada simpul 2 dapat menjangkitkan gaya mendatar dan tegak pada permukaan, sedangkan gelinding pada simpul 3 hanya menjangkitkan gaya tegak.



Gambar 4.1 Gaya-gaya pada rangka statis tertentu

Struktur jenis ini dapat diuraikan sebagai sistem persamaan aljabar lanjar simultan. Diagram gaya-benda-bebas diperlihatkan untuk tiap simpul dalam Gambar 4.2.



Gambar 4.2 Diagram gaya-benda-bebas untuk simpul-simpul rangka statis

Menurut hukum Newton, resultan gaya dalam arah mendatar maupun tegak harus nol pada tiap simpul, karena sistem dalam keadaan diam (statis). Oleh karena itu, untuk simpul 1,

$$\begin{aligned}\sum F_H &= 0 = -F_1 \cos 30^\circ + F_3 \cos 60^\circ + F_{1,h} \\ \sum F_V &= 0 = -F_1 \sin 30^\circ - F_3 \sin 60^\circ + F_{1,v}\end{aligned}$$

untuk simpul 2,

$$\begin{aligned}\sum F_H &= 0 = F_2 + F_1 \cos 30^\circ + F_{2,h} + H_2 \\ \sum F_V &= 0 = F_1 \sin 30^\circ - F_{2,v} + V_2\end{aligned}$$

dan untuk simpul 3,

$$\begin{aligned}\sum F_H &= 0 = -F_2 - F_3 \cos 60^\circ + F_{3,h} \\ \sum F_V &= 0 = F_3 \sin 60^\circ + F_{3,v} + V_3\end{aligned}$$

Gaya 1000 pon ke bawah pada simpul 1 berpadanan dengan $F_{1,v} = -1000$, sedangkan semua $F_{i,v}$ dan $F_{i,h}$ lainnya adalah nol. Persoalan rangka statis ini dapat dituliskan sebagai sistem yang disusun oleh enam persamaan lanjar dengan 6 peubah yang tidak diketahui:

$$\begin{aligned}\sum F_H &= 0 = -F_1 \cos 30^\circ + F_3 \cos 60^\circ + F_{1,h} = -0.866F_1 + 0.5F_3 \\ \sum F_V &= 0 = -F_1 \sin 30^\circ - F_3 \sin 60^\circ + F_{1,v} = -0.5F_1 - 0.866F_3 + 1000 \\ \sum F_H &= 0 = F_2 + F_1 \cos 30^\circ + F_{2,h} + H_2 = F_2 + 0.866F_1 + 0 + H_2 \\ \sum F_V &= 0 = F_1 \sin 30^\circ - F_{2,v} + V_2 = 0.5F_1 + V_2 \\ \sum F_H &= 0 = -F_2 - F_3 \cos 60^\circ + F_{3,h} = -F_2 - 0.5F_3 \\ \sum F_V &= 0 = F_3 \sin 60^\circ + F_{3,v} + V_3 = 0.866F_3 + V_3\end{aligned}$$

Keenam persamaan di atas ditulis ulang kembali dalam susunan yang teratur berdasarkan urutan peubah $F_1, F_2, F_3, H_2, V_2, V_3$:

$$\begin{array}{rcl} -0.866F_1 & + 0.5F_3 & = 0 \\ -0.5F_1 & - 0.866F_3 & = -1000 \\ -0.866F_1 - F_2 & - H_2 & = 0 \\ -0.5F_1 & - V_2 & = 0 \\ -F_2 - 0.5F_3 & & = 0 \\ -0.866F_3 & - V_3 & = 0 \end{array}$$

atau dalam bentuk matriks:

$$\left[\begin{array}{cccccc} 0.866 & 0 & -0.5 & 0 & 0 & 0 \\ 0.5 & 0 & 0.866 & 0 & 0 & 0 \\ -0.866 & -1 & 0 & -1 & 0 & 0 \\ -0.5 & 0 & 0 & 0 & -1 & 0 \\ 0 & 1 & 0.5 & 0 & 0 & 0 \\ 0 & 0 & -0.866 & 0 & 0 & -1 \end{array} \right] \begin{bmatrix} F_1 \\ F_2 \\ F_3 \\ H_2 \\ V_2 \\ V_3 \end{bmatrix} = \begin{bmatrix} 0 \\ -1000 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Masalah yang ditanyakan adalah nilai F_1, F_2, F_3, H_2, V_2 , dan V_3 yang memenuhi keenam persamaan tersebut secara simultan. Metode penyelesian sistem persamaan lanjar seperti di atas merupakan pokok bahasan Bab 4 ini.

4.1 Bentuk Umum Sistem Persamaan Lanjar

Sistem persamaan lanjar (SPL) dengan n peubah dinyatakan sebagai

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n &= b_1 \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n &= b_2 \\ &\vdots && \vdots \\ &\vdots && \vdots \\ a_{n1}x_1 + a_{n2}x_2 + \dots + a_{nn}x_n &= b_n \end{aligned} \quad (\text{P.4.1})$$

Dengan menggunakan perkalian matriks, kita dapat menulis (P.4.1) sebagai persamaan matriks

$$Ax = b \quad (\text{P.4.2})$$

yang dalam hal ini,

$A = [a_{ij}]$ adalah matriks berukuran $n \times n$

$x = [x_j]$ adalah matriks berukuran $n \times 1$

$b = [b_j]$ adalah matriks berukuran $n \times 1$ (disebut juga vektor kolom)

yaitu

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \dots & a_{2n} \\ a_{31} & a_{32} & a_{33} & \dots & a_{3n} \\ \vdots & & & & \\ a_{n1} & a_{n2} & a_{n3} & \dots & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ \vdots \\ b_n \end{bmatrix}$$

Solusi (P.4.1) adalah himpunan nilai x_1, x_2, \dots, x_n yang memenuhi n buah persamaan. Metode penyelesaian sistem persamaan lanjar dengan determinan (aturan Cramer) tidak praktis untuk sistem yang besar. Beberapa metode penyelesaian praktis sistem persamaan lanjar yang kita bahas di sini adalah:

1. Metode eliminasi Gauss
2. Metode eliminasi Gauss-Jordan
3. Metode matriks balikan
4. Metode dekomposisi LU

5. Metode lelaran Jacobi
6. Metode lelaran Gauss-Seidel.

Walaupun metode penyelesaian SPL beragam, namun sebagian besar metode tersebut, terutama metode 1 sampai 4, tetap didasarkan kepada metode yang paling dasar, yaitu eliminasi Gauss. Metode eliminasi Gauss-Jordan, metode matriks balikan, dan metode dekomposisi *LU* merupakan bentuk variasi lain dari metode eliminasi Gauss. Sedangkan metode lelaran Jacobi dan metode lelaran Gauss-Seidel dikembangkan dari gagasan metode lelaran pada solusi persamaan nirlanjar.

4.2 Metode Eliminasi Gauss

Metode ini berangkat dari kenyataan bahwa bila matriks A berbentuk *segitiga atas* seperti sistem persamaan berikut ini

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1n} \\ 0 & a_{22} & a_{23} & \dots & a_{2n} \\ 0 & 0 & a_{33} & \dots & a_{3n} \\ \vdots & & & & \\ 0 & 0 & 0 & \dots & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ \vdots \\ b_n \end{bmatrix}$$

maka solusinya dapat dihitung dengan **teknik penyulihan mundur (backward substitution)**:

$$\begin{aligned} a_{nn}x_n &= b_n \quad \rightarrow \quad x_n = b_n/a_{nn} \\ a_{n-1,n-1}x_{n-1} + a_{n-1,n}x_n &= b_{n-1} \quad \rightarrow \quad x_{n-1} = \frac{b_{n-1} - a_{n-1,n}x_n}{a_{n-1,n-1}} \\ a_{n-2,n-2}x_{n-2} + a_{n-2,n-1}x_{n-1} + a_{n-2,n}x_n &= b_{n-2} \quad \rightarrow \quad x_{n-2} = \frac{b_{n-2} - a_{n-2,n-1}x_{n-1} - a_{n-2,n}x_n}{a_{n-2,n-2}} \\ &\vdots \\ \text{dst.} & \end{aligned}$$

Sekali $x_n, x_{n-1}, x_{n-2}, \dots, x_{k+1}$ diketahui, maka nilai x_k dapat dihitung dengan

$$x_k = \frac{b_k - \sum_{j=k+1}^n a_{kj}x_j}{a_{kk}}, \quad k = n-1, n-2, \dots, 1 \text{ dan } a_{kk} \neq 0. \quad \text{P.4.3)}$$

Kondisi $a_{kk} \neq 0$ sangat penting, sebab bila $a_{kk} = 0$, persamaan (P.4.3) mengerjakan pembagian dengan nol. Apabila kondisi tersebut tidak dipenuhi, maka SPL tidak mempunyai jawaban.

Di dalam Bab 4 ini, kita menggunakan struktur data matriks untuk semua algoritma yang dijelaskan nanti. Pendeklarasiannya adalah sebagai berikut ini:

```
(* KAMUS GLOBAL *)
const
  n = ... ; { ukuran matriks A }
type
  matriks = array[1..n, 1..n] of real;
  vektor = array[1..n] of real;
var
  { larik/matriks yang digunakan untuk sistem Ax = b }
  A : matriks;
  b : vektor;
  x : vektor;
```

Program 4.1 berikut berisi algoritma penyulihan mundur.

Program 4.1 Penyulihan Mundur

```
procedure Sulih_Mundur(A : matriks; b : vektor; n: integer;
                       var x : vektor);
{ Menghitung solusi sistem persamaan lanjar yang sudah berbentuk matriks
segitiga atas
K.Awal : A adalah matriks yang berukuran  $n \times n$ , elemennya sudah terdefinisi
harganya; b adalah vektor kolom yang berukuran  $n \times 1$ .
K.Akhir: x berisi solusi sistem persamaan lanjar.
}
var
  j, k: integer;
  sigma: real;
begin
  x[n]:=b[n]/a[n,n];
  for k:=n-1 downto 1 do begin
    sigma:=0;
    for j:=k+1 to n do
      sigma:=sigma + a[k, j] * x[j];
    endfor
    x[k]:= (b[k] - sigma )/a[k, k];
  end;
end;
```

Contoh 4.1

[MAT92] Selesaikan sistem persamaan lanjar berikut dengan teknik penyulihan mundur

$$\begin{aligned}4x_1 - x_2 + 2x_3 + 3x_4 &= 20 \\-2x_2 + 7x_3 - 4x_4 &= -7 \\6x_3 + 5x_4 &= 4 \\3x_4 &= 6\end{aligned}$$

Penyelesaian:

$$\begin{aligned}x_4 &= 6/3 = 2 \\x_3 &= \frac{(4 - 5(2))}{6} = -1 \\x_2 &= \frac{-7 - 7(-1) + 4(2)}{-2} = -4 \\x_1 &= \frac{20 + 1(-4) - 2(-1) - 3(2)}{4} = 3\end{aligned}$$

Jadi, solusinya adalah $x = (3, -4, -1, 2)^T$. ■

Metode eliminasi Gauss pada prinsipnya bertujuan mentransformasi sistem $Ax = b$ menjadi sistem

$$Ux = y \quad (\text{P.4.4})$$

dengan U adalah matriks *segitiga atas*. Selanjutnya solusi x dapat dihitung dengan teknik penyulihan mundur. Contohnya pada sistem dengan 4 persamaan lanjar berikut (Elemen matriks A dan vektor kolom b disatukan dalam bentuk satu bentuk matriks):

$$\left[\begin{array}{cccc|c} a_{11} & a_{12} & a_{13} & a_{14} & b_1 \\ a_{21} & a_{22} & a_{23} & a_{24} & b_2 \\ a_{31} & a_{32} & a_{33} & a_{34} & b_3 \\ a_{41} & a_{42} & a_{43} & a_{44} & b_4 \end{array} \right] \xrightarrow{\substack{\text{dieliminasi} \\ \text{menjadi } [U, y]}} \left[\begin{array}{cccc|c} a_{11} & a_{12} & a_{13} & a_{14} & b_1 \\ 0 & a_{22}^{(1)} & a_{23}^{(1)} & a_{24}^{(1)} & b_2^{(1)} \\ 0 & 0 & a_{33}^{(2)} & a_{34}^{(2)} & b_3^{(2)} \\ 0 & 0 & 0 & a_{44}^{(3)} & b_4^{(3)} \end{array} \right]$$

$[A, b] \qquad [U, y]$

Tanda pangkat (1), (2), (3) menunjukkan bahwa elemen matriks A telah berubah satu kali, dua kali, dan tiga kali.

Proses eliminasi terdiri atas tiga operasi baris elementer:

1. *Pertukaran* : Urutan dua persamaan dapat ditukar karena pertukaran tersebut tidak mempengaruhi solusi akhir.
2. *Penskalaan* : Persamaan dapat dikali dengan konstanta bukan nol, karena perkalian tersebut tidak mempengaruhi solusi akhir.
3. *Pengantian* : Persamaan dapat diganti dengan penjumlahan persamaan itu dengan gandaan persamaan lain. Misalnya persamaan diganti dengan selisih persamaan itu dengan dua kali persamaan lain; yaitu

$$\text{baris}_r := \text{baris}_r - m_{p,r} \text{baris}_p \quad (\text{P.4.5})$$

Nilai $a_{r,r}$ pada posisi (r, r) yang digunakan untuk mengeliminasi x_r pada baris $r+1, r+2, \dots, N$ dinamakan elemen *pivot* dan persamaan pada baris ke- r disebut **persamaan pivot** [MAT92]. Ada kemungkinan *pivot* bernilai nol sehingga pembagian dengan nol tidak dapat dilakukan. Tata-ancang eliminasi yang tidak mempedulikan nilai *pivot* adalah tatancang yang naif (*naive*) atau sederhana. Metode eliminasi Gauss seperti ini dinamakan **metode eliminasi Gauss naif** (*naive Gaussian elimination*), karena metodenya tidak melakukan pemeriksaan kemungkinan pembagian dengan nol. Pada metode eliminasi Gauss naif tidak ada operasi pertukaran baris dalam rangka menghindari *pivot* yang bernilai nol itu.

PIVOT: Critical, cardinal, or crucial factor
(Kamus Webster)

Contoh 4.2

Selesaikan sistem persamaan lanjar dengan metode eliminasi Gauss naif:

$$\begin{aligned} 2x_1 + 3x_2 - x_3 &= 5 \\ 4x_1 + 4x_2 - 3x_3 &= 3 \\ -2x_1 + 3x_2 - x_3 &= 1 \end{aligned}$$

Penyelesaian:

$$\left[\begin{array}{ccc|c} 2 & 3 & -1 & 5 \\ 4 & 4 & -3 & 3 \\ -2 & 3 & -1 & 1 \end{array} \right] \xrightarrow{\sim} \left[\begin{array}{ccc|c} 2 & 3 & -1 & 5 \\ 0 & -2 & -1 & -7 \\ 0 & 6 & -2 & 6 \end{array} \right] \xrightarrow{\sim} \left[\begin{array}{ccc|c} 2 & 3 & -1 & 5 \\ 0 & -2 & -1 & -7 \\ 0 & 0 & -5 & -15 \end{array} \right]$$

Keterangan: (i) elemen yang dicetak tebal menyatakan *pivot*.
(ii) simbol “~” menyatakan operasi baris elementer .
(iii) R_i menyatakan baris (*row*) ke- i

- (iv) $R_2 - \frac{4}{2}R_1$ artinya elemen-elemen pada baris kedua dikurangi dengan dua kali elemen-elemen pada baris ke satu.

$$\begin{array}{rcl} R_2 & : & 4 & 4 & -3 & 3 \\ 2R_1 & : & 4 & 6 & -2 & 10 \\ \hline R_2 - \frac{4}{2}R_1 & : & 0 & -2 & -1 & -7 \end{array} \text{ (menjadi elemen baris ke-2)}$$

Solusi sistem diperoleh dengan teknik penyulihan mundur sebagai berikut:

$$\begin{aligned} -5x_3 &= -15 & \rightarrow x_3 &= 3 \\ -2x_2 - x_3 &= -7 & \rightarrow x_2 &= (-7 + 3)/-2 = 2 \\ 2x_1 + 3x_2 - x_3 &= 5 & \rightarrow x_1 &= (5 + 3 - 6)/2 = 1 \end{aligned}$$

Jadi, solusinya adalah $x = (1, 2, 3)^T$

■

Program 4.2 Metode Eliminasi Gauss Naif

```

procedure Eliminasi_Gauss_Naif(A : matriks; b : vektor; n:integer;
                                var x : vektor);
{ Menghitung solusi sistem persamaan lanjar Ax = b
  K.Awal : A adalah matriks yang berukuran  $n \times n$ , elemennya sudah terdefini-
            nisi harganya; b adalah vektor kolom yang berukuran  $n \times 1$ 
  K.Akhir: x berisi solusi sistem
}
var
  i; k, j : integer;
  m: real;
begin
  for k:=1 to n-1 do {mulai dari baris pivot 1 sampai baris pivot n-1}
  begin
    for i:=(k+1) to n do {eliminasi mulai dari baris k+1 sampai baris n}
    begin
      m:=a[i,k]/a[k,k]; {hitung faktor pengali}
      for j:=k to n do {eliminasi elemen dari kolom k sampai kolom n}
        a[i,j]:=a[i,j] - m*a[k,j];
      {endfor}
      b[i]:=b[i] - m*b[k]; {eliminasi elemen vektor b pada baris i}
    end;
  end;
  Sulih_Mundur(A, b, n, x); {dapatkan solusinya dengan teknik penyulihan
                            mundur)
end;

```

Kelemahan eliminasi Gauss naif

Jika $pivot a_{pp} = 0$, baris ke- k tidak dapat digunakan untuk memeliminasi elemen pada kolom p , karena terjadinya pembagian dengan nol. Oleh karena itu, *pivot* yang bernilai nol harus dihindari dengan tata-ancang (*strategy*) *pivoting*.

4.2.1 Tata - a nc a ng *Pivoting*

Prinsip tata-ancang *pivoting* adalah sebagai berikut: jika $a_{p,p}^{(p-1)} = 0$, cari baris k dengan $a_{k,p} \neq 0$ dan $k > p$, lalu pertukarkan baris p dan baris k . Metode eliminasi Gauss dengan tata-ancang pivoting disebut metode eliminasi Gauss yang diperbaiki (*modified Gaussian elimination*).

Contoh 4.3

Selesaikan sistem persamaan lanjar berikut dengan metode eliminasi Gauss yang menerapkan tahapan *pivoting*:

$$\begin{array}{rcl} x_1 + 2x_2 + x_3 & = 2 \\ 3x_1 + 6x_2 & = 9 \\ 2x_1 + 8x_2 + 4x_3 & = 6 \end{array}$$

Setelah operasi baris 1, elemen a_{22} yang akan menjadi *pivot* pada operasi baris 2 ternyata sama dengan nol. Karena itu, pada operasi baris 2, elemen baris 2 dipertukarkan dengan elemen baris 3. Tanda (*) menyatakan pertukaran baris terjadi akibat proses *pivoting*. Sekarang elemen $a_{22} = 4 \neq 0$ sehingga operasi baris elementer dapat diteruskan. Tetapi, karena matriks A sudah membentuk matriks U , proses eliminasi selesai. Solusinya diperoleh dengan teknik penyulihan mundur, yaitu $x_3 = -1$, $x_2 = 1$, dan $x_1 = 1$.

Melakukan pertukarkan baris untuk menghindari *pivot* yang bernilai nol adalah cara *pivoting* yang sederhana (*simple pivoting*). Masalah lain dapat juga timbul bila elemen *pivot* sangat dekat ke nol, karena jika elemen *pivot* sangat kecil dibandingkan terhadap elemen lainnya, maka galat pembulatan dapat muncul [CHA91]. Ingatlah kembali bahwa kita bekerja dengan mesin (komputer) yang beroperasi dengan pembulatan bilangan riil. Jadi, disamping menghindari pembagian dengan nol, tatanan *pivoting* dapat juga diperluas untuk mengurangi galat pembulatan.

Ada dua macam tatancang *pivoting*:

1. *Pivoting sebagian (partial pivoting)*

Pada tatancang *pivoting* sebagian, *pivot* dipilih dari semua elemen pada kolom p yang mempunyai nilai mutlak terbesar,

$$|a_{k,p}| = \max\{|a_{p,p}|, |a_{p+1,p}|, \dots, |a_{n-1,p}|, |a_{n,p}|\}$$

lalu pertukarkan baris ke- k dengan baris ke- p . Misalkan setelah operasi baris pertama diperoleh matriksnya seperti yang digambarkan pada matriks di bawah ini. Untuk operasi baris kedua, carilah elemen x pada kolom kedua, dimulai dari baris ke-2 sampai baris ke-4, yang nilai mutlaknya terbesar, lalu pertukarkan barisnya dengan baris kedua. Elemen x yang nilai mutlaknya terbesar itu sekarang menjadi *pivot* untuk operasi baris selanjutnya.

$$\left[\begin{array}{cccc|c} x & x & x & x & x \\ 0 & x & x & x & x \\ 0 & x & x & x & x \\ 0 & x & x & x & x \end{array} \right]$$

Cari $|x|$ terbesar, lalu pertukarkan
barisnya dengan baris ke-2

Perhatikanlah bahwa teknik *pivoting* sebagian juga sekaligus menghindari pemilihan $pivot = 0$ (sebagaimana pada *simple pivoting*) karena 0 tidak akan pernah menjadi elemen dengan nilai mutlak terbesar, kecuali jika seluruh elemen di kolom yang diacu adalah 0. Apabila setelah melakukan *pivoting* sebagian ternyata elemen $pivot = 0$, itu berarti sistem persamaan lanjar tidak dapat diselesaikan (*singular system*).

2. *Pivoting lengkap (complete pivoting)*

Jika disamping baris, kolom juga diikutkan dalam pencarian elemen terbesar dan kemudian dipertukarkan, maka tatancang ini disebut *pivoting lengkap*. *Pivoting lengkap* jarang dipakai dalam program sederhana karena pertukaran kolom mengubah urutan suku x dan akibatnya menambah kerumitan program secara berarti [CHA91].

Contoh 4.4

Dengan menggunakan empat angka bena, selesaikan sistem persamaan berikut dengan metode eliminasi Gauss:

$$\begin{aligned}0.0003x_1 + 1.566x_2 &= 1.569 \\0.3454x_1 - 2.436x_2 &= 1.018\end{aligned}$$

- (a) tanpa tatancang *pivoting* sebagian (Gauss naif)
- (b) dengan tatancang *pivoting* sebagian (Gauss yang dimodifikasi)

(Perhatikan, dengan 4 angka bena, solusi sejatinya adalah $x_1 = 10.00$ dan $x_2 = 1.00$)

Penyelesaian:

- (a) tanpa tatancang *pivoting* sebagian:

$$\left[\begin{array}{cc|c} 0.0003 & 1.566 & 1.569 \\ 0.3454 & -2.436 & 1.018 \end{array} \right]$$

Operasi baris pertama (0.0003 sebagai *pivot*):

$$R_2 \leftarrow \frac{R_2 - 0.3454R_1}{0.0003} = R_2 - 1151 R_1$$

(tanda “ \leftarrow ” berarti “diisi” atau “diganti dengan”)

Jadi,

$$a_{21} \approx 0$$

$$a_{22} \approx -2.436 - (1151)(1.566) \approx -2.436 - 1802 \approx -1804$$

$$b_2 \approx 1.018 - (1151)(1.569) \approx 1.018 - 1806 \approx -1805$$

$$\left[\begin{array}{cc|c} 0.0003 & 1.566 & 1.569 \\ 0.3454 & -2.436 & 1.018 \end{array} \right] \sim \left[\begin{array}{cc|c} 0.0003 & 1.566 & 1.569 \\ 0 & -1804 & -1805 \end{array} \right]$$

Solusinya diperoleh dengan teknik penyulihan mundur:

$$x_2 = -1805/-1804 = 1.001$$

$$x_1 = \frac{1.569 - (1.566)(1.001)}{0.0003} = \frac{1.569 - 1.568}{0.0003} = \frac{0.001}{0.0003} = 3.333$$

(jauh dari solusi sejati)

Jadi, $x = (3.333, 1.001)^T$. Solusi ini sangat jauh berbeda dengan solusi sejatinya. Kegagalan ini terjadi karena $|a_{11}|$ sangat kecil dibandingkan $|a_{12}|$, sehingga galat

pembulatan yang kecil pada x_2 menghasilkan galat besar di x_1 . Perhatikan juga bahwa $1.569 - 1.568$ adalah pengurangan dua buah bilangan yang hampir sama, yang menimbulkan hilangnya angka bena pada hasil pengurangannya (*loss of significance*).

- (b) dengan tata-ancang *pivoting* sebagian

Baris pertama dipertukarkan dengan baris kedua sehingga 0.3454 menjadi pivot

$$\left[\begin{array}{cc|c} 0 & 0.3454 & -2.436 \\ 0.0003 & 1.566 & 1.569 \end{array} \right] R_2 - \frac{0.0003}{0.3454} R_1 \sim \left[\begin{array}{cc|c} 0.3454 & -2.436 & 1.018 \\ 0 & 1.568 & 1.568 \end{array} \right]$$

Dengan teknik penyulihan mundur diperoleh

$$x_2 = 1.568/1.568 = 1.000$$

$$x_1 = \frac{1.018 - (-2.436)(1.000)}{0.3454} = 10.02 \text{ (lebih baik daripada solusi (a))}$$

Jadi, solusinya adalah $x = (10.02, 1.000)^T$, yang lebih baik daripada solusi (a). Keberhasilan ini karena $|a_{21}|$ tidak sangat kecil dibandingkan dengan $|a_{22}|$, sehingga galat pembulatan yang kecil pada x_2 tidak akan menghasilkan galat yang besar pada x_1 . ■

Contoh 4.5

Dengan menggunakan empat angka bena, selesaikan sistem persamaan berikut ini dengan metode eliminasi Gauss:

$$\begin{aligned} 1.133x_1 + 5.281x_2 &= 6.414 \\ 24.14x_2 - 1.210x_1 &= 22.93 \end{aligned}$$

- (a) tanpa tatacang *pivoting* sebagian (Gauss naif)
 (b) dengan tatacang *pivoting* sebagian

(Perhatikan, dengan 4 angka bena, solusi sejatinya adalah $x_1 = x_2 = 1.000$)

Penyelesaian:

- (a) tanpa tatacang *pivoting* sebagian

$$\left[\begin{array}{cc|c} 1.133 & 5.281 & 6.414 \\ 24.14 & 1.210 & 22.93 \end{array} \right] R_2 - \frac{24.14}{1.133} R_1 \sim \left[\begin{array}{cc|c} 1.133 & 5.281 & 6.414 \\ 0 & -113.7 & -113.8 \end{array} \right]$$

Solusinya diperoleh dengan teknik penyulihan mundur:

$$x_2 = -113.8/-113.7 = 1.001$$

$$x_1 = \frac{6.414 - (5.281)(1.001)}{1.133} = 0.9956$$

Jadi, $x = (0.9956, 1.001)^T$. Solusi ini kurang teliti dibandingkan dengan solusi sejatinya

- (b) dengan tatancang *pivoting* sebagian

Baris ke-1 dipertukarkan dengan baris ke-2, sehingga 24.14 menjadi *pivot*.

$$\left[\begin{array}{cc|c} 24.14 & 1.210 & 22.93 \\ 1.133 & 5.281 & 6.414 \end{array} \right] R_2 - \left(\frac{1.133}{24.14} \right) R_1 \left[\begin{array}{cc|c} 24.14 & -1.210 & 22.93 \\ 0 & 5.338 & 5.338 \end{array} \right]$$

Dengan teknik penyulihan mundur, solusinya adalah

$$x_2 = 5.338/5.338 = 1.000$$

$$x_1 = \frac{22.93 + (1.210)(1.000)}{24.14} = 1.000$$

Jadi, $x = (1.000, 1.000)^T$. Solusi ini tepat sama dengan solusi sejatinya, jadi lebih baik daripada solusi (a) di atas. ■

Contoh 4.4 dan Contoh 4.5 di atas memperlihatkan bahwa dengan tatancang *pivoting* sebagian galat pembulatan dapat dikurangi. Contoh lainnya untuk sistem dengan tiga persamaan berikut:

$$\left[\begin{array}{cccc|c} 0 & 2 & 0 & 1 & 0 \\ 2 & 2 & 3 & 2 & -2 \\ 4 & -3 & 0 & 1 & -7 \\ 6 & 1 & -6 & -5 & 6 \end{array} \right] R_1 \Leftrightarrow R_4 \quad (*) \quad \left[\begin{array}{cccc|c} 6 & 1 & -6 & -5 & 6 \\ 2 & 2 & 3 & 2 & -2 \\ 4 & -3 & 0 & 1 & -7 \\ 0 & 2 & 0 & 1 & 0 \end{array} \right] \quad R_2 - \frac{2}{6}R_1 \quad R_3 - \frac{4}{6}R_1 \quad \sim$$

$$\left[\begin{array}{cccc|c} 6 & 1 & 6 & -5 & 6 \\ 0 & 1.6667 & 5 & 3.666 & -4 \\ 0 & -3.6667 & 4 & 4.333 & -4 \\ 0 & 2 & 0 & 1 & 0 \end{array} \right] R_2 \Leftrightarrow R_3 \quad (*) \quad \left[\begin{array}{cccc|c} 6 & 1 & 6 & -5 & 6 \\ 0 & -3.6667 & 4 & 4.3333 & 11 \\ 0 & 1.6667 & 5 & 3.6667 & -4 \\ 0 & 2 & 0 & 1 & 0 \end{array} \right] \text{dst ...}$$

Metode eliminasi Gauss yang diperbaiki (tidak naif) adalah metode eliminasi Gauss yang melibatkan operasi pertukaran baris dalam rangka memilih elemen *pivot* dengan nilai mutlak terbesar. Program 4.3 berikut berisi algoritma eliminasi Gauss yang diperbaiki.

Program 4.3 Metode Eliminasi Gauss yang diperbaiki (dengan tatancang *pivoting*)

```

procedure Eliminasi_Gauss(A : matriks; b : vektor; n:integer;
                           var x : vektor);
{ Menghitung solusi sistem persamaan lanjar  $Ax = b$  dengan metode eliminasi
Gauss yang diperbaiki.
K.Awal : A adalah matriks yang berukuran  $n \times n$ , elemennya sudah terdefinisi
harganya; b adalah vektor kolom yang berukuran  $n \times 1$ 
K.Akhir: x berisi solusi sistem. Jika tidak ada solusi yang unik, vektor
x diisi dengan nilai -9999
}

var
  i, k, j, r, s, t : integer;
  m, tampung, pivot : real;
  singular : boolean;           { true jika SPL tidak mempunyai solusi }

begin
  k:=1; singular:=false;
  while (k<=n-1) and (not singular) do
    begin
      {cari elemen pivot dengan nilai mutlak terbesar}
      pivot:=a[k,k]; r:=k;   {baris pivot}
      for t:=k+1 to n do {bandingkan dengan elemen pada baris k+1 ..n}
        if ABS(a[t,k]) > ABS(pivot) then
          begin
            pivot:=a[t,k]; r:=t;
            end {if} ;
        {jika pivot=0 maka matriks A singular. Proses dihentikan}
        if pivot = 0 then { atau hampir nol, gunakan suatu epsilon }
          singular:=true
        else
          begin
            if r > k then {jika pivot tetap pada baris k, tidak ada
                pertukaran}
            begin
              {pertukarkan baris k dengan baris r di matriks A}

              for s:=1 to n do
                begin
                  tampung:=a[k,s]; a[k,s]:=a[r,s]; a[r,s]:=tampung;
                end;

              {pertukarkan juga b[k] dengan b[r]}
              tampung:=b[k]; b[k]:=b[r]; b[r]:=tampung;
            end {if} ;
            for i:=(k+1) to n do {eliminasi dari baris k+1 sampai
                baris n}
            begin
              m:=a[i,k]/a[k,k]; {hitung faktor pengali}
              for j:=k to n do {eliminasi dari kolom k sampai kolom n}
                a[i,j]:=a[i,j] - m*a[k,j];
            end
```

```

    {endfor}
    b[i]:=b[i] - m*b[k];      {eliminasi vektor b pada baris i}
  end {for} ;
end {if} ;
k:=k+1;
end {while};
{ k = n or singular }

if not singular then
  Sulih_Mundur(A, b, n, x); {dapatkan solusinya dengan teknik penyulihan
                                mundur)
else
  { solusi tidak ada, tetapi vektor x harus tetap diisi }
  for i:=1 to n do
    x[i]:=-9999;
  {endfor}
{endif}
end;

```

Untuk hasil terbaik, penerapan tatancang *pivoting* dan penggunaan bilangan berketalitian ganda dapat mengurangi galat pembulatan.

Pertukaran elemen baris, sebagai akibat dari pemilihan *pivot*, memakan waktu, khususnya pada SPL yang berukuran besar. Waktu pertukaran ini dapat dikurangi bila elemen-elemen baris tidak benar-benar ditukar secara aktual. Urutan baris dicatat di dalam larik BAR[1..n]. Pertukaran yang dikerjakan hanyalah pertukaran elemen larik BAR. Pada mulanya larik BAR berisi indeks baris matriks:

```
for i:=1 to n do BAR[i]:=i;
```

Elemen matriks diacu sebagai

```
A[BAR[i], k]
```

Maka, pertukaran baris *k* dan baris *r* dikerjakan sebagai

```
tampung:=BAR[r];
BAR[r]:=BAR[k];
BAR[k]:=tampung;
```

4.2.2 Penskalaan

Selain dengan *pivoting* sebagian, penskalaan (*scaling*) juga dapat digunakan untuk mengurangi galat pembulatan pada SPL yang mempunyai perbedaan koefisien yang mencolok. Situasi demikian sering ditemui dalam praktik rekayasa yang menggunakan ukuran satuan yang berbeda-beda dalam menentukan persamaan

simultan. Misalnya pada persoalan rangkaian listrik, tegangan listrik dapat dinyatakan dalam satuan yang berkisar dari mikrovolt sampai kilovolt. Pemakaian satuan yang berbeda-beda dapat menuju ke koefisien yang besarnya sangat berlainan. Ini berdampak pada galat pembulatan, dan karena itu mempengaruhi *pivoting* [CHA91]. Dengan penskalaan berarti kita menormalkan persamaan. Cara menskala adalah membagi tiap baris persamaan dengan nilai mutlak koefisien terbesar di ruas kirinya. Akibat penskalaan, koefisien maksimum dalam tiap baris adalah 1. Cara menskala seperti ini dinamakan dengan **menormalkan SPL**.

Contoh 4.6

Selesaikan sistem persamaan lanjar berikut sampai 3 angka bena dengan menggunakan metode eliminasi Gauss yang menerapkan penskalaan dan tanpa penskalaan:

$$\begin{array}{rcl} 2x_1 + 100000x_2 & = & 100000 \\ x_1 + \quad \quad x_2 & = & 2 \end{array}$$

(Solusi sejatinya dalam 3 angka bena adalah $x_1 = x_2 = 1.00$)

Penyelesaian:

(i) Tanpa penskalaan :

$$\left[\begin{array}{cc|c} 2 & 100000 & 100000 \\ 1 & 1 & 2 \end{array} \right] \xrightarrow{R_2 - 1/2 R_1} \left[\begin{array}{cc|c} 2 & 100000 & 100000 \\ 0 & -50000 & -50000 \end{array} \right]$$

Solusinya adalah

$$\begin{array}{ll} x_2 = 1.00 \\ x_1 = 0.00 \quad (\text{salah}) \end{array}$$

(ii) Dengan penskalaan :

$$\left[\begin{array}{cc|c} 2x_1 + 100000x_2 & = & 100000 \\ x_1 + x_2 & = & 2 \end{array} \right] \xrightarrow{\begin{array}{l} : 100000 \\ : 1 \end{array}} \left[\begin{array}{cc|c} 0.00002 & x_1 + x_2 & = 1 \\ 1 & x_1 + x_2 & = 2 \end{array} \right]$$

$$\left[\begin{array}{cc|c} 0.00002 & 1 & 1 \\ 1 & 1 & 2 \end{array} \right] \xrightarrow{R_1 \leftrightarrow R_2} \left[\begin{array}{cc|c} 1 & 1 & 2 \\ 0.00002 & 1 & 1 \end{array} \right] \sim \left[\begin{array}{cc|c} 1 & 1 & 2 \\ 0 & 1 & 1.00 \end{array} \right]$$

Solusinya,

$$\begin{array}{ll} x_2 = 1.00 \\ x_1 = 1.00 \quad (\text{benar}) \end{array}$$

yang sesuai dengan solusi sejati. Contoh di atas juga memperlihatkan bahwa penskalaan dapat mengubah pemilihan *pivot*. ■

4.2.2 Kemungkinan Solusi SPL

Tidak semua SPL mempunyai solusi. Ada tiga kemungkinan solusi yang dapat terjadi pada SPL:

- (a) mempunyai solusi yang unik,
- (b) mempunyai banyak solusi, atau
- (c) tidak ada solusi sama sekali.

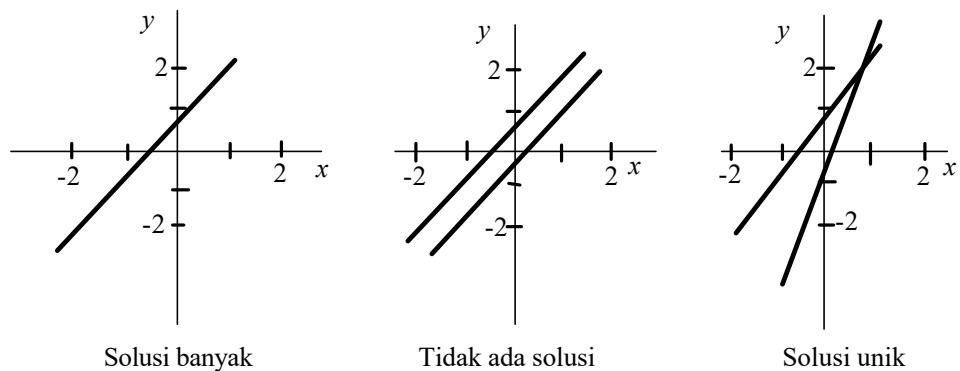
Dengan grafik, ketiga kemungkinan solusi ini diperlihatkan oleh tiga SPL dengan dua persamaan berikut [NAK92]:

$$(i) \begin{aligned} -x + y &= 1 \\ -2x + 2y &= 2 \end{aligned}$$

$$(ii) \begin{aligned} -x + y &= 1 \\ -x + y &= 0 \end{aligned}$$

$$(iii) \begin{aligned} -x + y &= 1 \\ 2x - y &= 0 \end{aligned}$$

Grafik ketiga SPL diperlihatkan pada Gambar 4.3. Grafik pertama memperlihatkan bahwa kedua persamaan berimpit pada satu garis lurus. Solusinya terdapat di sepanjang garis tersebut (banyak solusi). Grafik kedua memperlihatkan kedua persamaan menyatakan dua garis yang sejajar. Tidak ada perpotongan kedua garis tersebut (tidak ada solusi). Sedangkan pada grafik ketiga, kedua persamaan berpotongan pada sebuah titik (solusinya tunggal atau unik).



Gambar 4.3 Kemungkinan solusi sistem persamaan lanjar

Untuk SPL dengan tiga buah persamaan atau lebih (dengan tiga peubah atau lebih), tidak terdapat tafsiran geometrisnya (tidak mungkin dibuat ilustrasi grafisnya) seperti pada SPL dengan dua buah persamaan. Namun, kita masih dapat memeriksa masing-masing kemungkinan solusi itu berdasarkan pada bentuk matriks akhirnya. Agar lebih jelas, tinjau contoh pada SPL yang disusun oleh tiga persamaan.

1. Solusi unik/tunggal

$$\left[\begin{array}{ccc|c} 1 & 1 & 1 & 0 \\ 2 & 3 & 1 & 1 \\ 3 & 1 & 2 & 1 \end{array} \right] \xrightarrow{\text{Eliminasi Gauss}} \left[\begin{array}{ccc|c} 1 & 1 & 1 & 0 \\ 0 & 1 & -1 & 1 \\ 0 & 0 & -3 & 3 \end{array} \right]$$

Solusi: $x_1 = 1, x_2 = 0, x_3 = -1$

2. Solusi banyak/tidak terhingga

$$\left[\begin{array}{ccc|c} 1 & 1 & 2 & 4 \\ 2 & -1 & 1 & 2 \\ 1 & 2 & 3 & 6 \end{array} \right] \xrightarrow{\text{Eliminasi Gauss}} \left[\begin{array}{ccc|c} 1 & 1 & 2 & 4 \\ 0 & -3 & -3 & -6 \\ 0 & 0 & 0 & 0 \end{array} \right]$$

Perhatikan hasil eliminasi Gauss pada baris terakhir. Persamaan yang bersesuaian dengan baris terakhir tersebut adalah

$$0x_1 + 0x_2 + 0x_3 = 0$$

yang dipenuhi oleh banyak nilai x . Solusinya diberikan dalam bentuk parameter:

$$\text{Misalkan } x_3 = k,$$

$$\text{maka } x_2 = -6 + 3k \text{ dan } x_1 = 10 - 5k, \text{ dengan } k \in R.$$

Terdapat tidak berhingga nilai k , berarti solusi SPL banyak sekali.

3. Tidak ada solusi

$$\left[\begin{array}{ccc|c} 1 & 1 & 2 & 4 \\ 2 & -1 & 1 & 2 \\ 1 & 2 & 3 & 7 \end{array} \right] \xrightarrow{\text{Eliminasi Gauss}} \left[\begin{array}{ccc|c} 1 & 1 & 2 & 4 \\ 0 & -3 & -3 & -6 \\ 0 & 0 & 0 & 1 \end{array} \right]$$

Perhatikan hasil eliminasi Gauss pada baris terakhir. Persamaan yang bersesuaian dengan baris terakhir tersebut adalah

$$0x_1 + 0x_2 + 0x_3 = 1$$

yang dalam hal ini, tidak nilai x_i yang memenuhi, $i = 1, 2, 3$

Bentuk akhir matriks setelah eliminasi Gauss untuk ketiga kemungkinan solusi di atas dapat digambarkan sebagai berikut:

Solusi unik

Solusi banyak

Tidak ada solusi

Kita rangkum “pertanda” kemungkinan solusi SPL di bawah ini:

1. Jika pada hasil eliminasi Gauss tidak terdapat baris yang semuanya bernilai 0 (termasuk elemen pada baris yang bersesuaian pada vektor kolom b), maka solusi SPL dipastikan unik.
2. Jika pada hasil eliminasi Gauss terdapat paling sedikit satu baris yang semuanya bernilai 0 (termasuk elemen pada baris yang bersesuaian pada vektor kolom b), maka SPL mempunyai banyak solusi.
3. Jika pada hasil eliminasi Gauss terdapat baris yang semuanya bernilai 0 tetapi elemen pada baris yang bersesuaian pada vektor kolom b tidak 0, maka SPL tidak mempunyai solusi.

Program eliminasi Gauss harus dapat menangani ketiga kemungkinan solusi tersebut.

4.3 Metoda Eliminasi Gauss-Jordan

Metode eliminasi Gauss-Jordan merupakan variasi dari metode eliminasi Gauss. Dalam hal ini, matriks A dieliminasi menjadi matriks identitas I . Di sini tidak diperlukan lagi teknik penyulihan mundur untuk memperoleh solusi SPL. Solusinya langsung diperoleh dari vektor kolom b hasil proses eliminasi.

$$Ax = b \rightarrow Ix = b'$$

Dalam bentuk matriks, eliminasi Gaus-Jordan ditulis sebagai

$$\left[\begin{array}{cccc|c} a_{11} & a_{12} & a_{13} & \dots & a_{1n} & b_1 \\ a_{21} & a_{22} & a_{23} & \dots & a_{2n} & b_2 \\ a_{31} & a_{32} & a_{33} & \dots & a_{3n} & b_3 \\ \vdots & & & & & \vdots \\ a_{n1} & a_{n2} & a_{n3} & \dots & a_{nn} & b_n \end{array} \right] \xrightarrow{\quad} \left[\begin{array}{cccc|c} 1 & 0 & 0 & \dots & 0 & b'_1 \\ 0 & 1 & 0 & \dots & 0 & b'_2 \\ 0 & 0 & 1 & \dots & 0 & b'_3 \\ \vdots & & & & & \vdots \\ 0 & 0 & 0 & \dots & 1 & b'_n \end{array} \right]$$

$$\text{Solusinya: } \begin{aligned} x_1 &= b'_1 \\ x_2 &= b'_2 \\ &\dots \\ x_n &= b'_n \end{aligned}$$

Seperti pada metode eliminasi Gauss naif, metode eliminasi Gauss-Jordan naif tidak menerapkan tata-ancang pivoting dalam proses eliminasinya.

Program 4.4 Metode Eliminasi Gauss-Jordan Naif

```

procedure Eliminasi_Gauss_Jordan_Naif(A : matriks; b: vektor; n:integer;
                                      var x : vektor);
{ Menghitung solusi sistem persamaan lanjar Ax = b dengan metode eliminasi
  Gauss-Jordan.
  K.Awal : A adalah matriks yang berukuran n × n, elemennya sudah terdefinisi
          harganya; b adalah vektor kolom yang berukuran n × 1
  K.Akhir: x berisi solusi sistem
}
var
  i, k, j : integer;
  m, tampung: real;
begin
  for k:=1 to n do
    begin
      tampung:=a[k,k];
      for j:=1 to n do {bagi elemen baris k dengan a[k,k]}
        a[k,j]:=a[k,j]/tampung;
      {endfor}
      b[k]:=b[k]/tampung; {jangan lupa b[k] juga dibagi dengan a[k,k]}
      for i:=1 to n do {eliminasi elemen baris i s/d baris n, i≠k}

```

```

begin
  if i<>k then
    begin
      m:=a[i,k];
      for j:=1 to n do {eliminasi elemen dari kolom 1 s/d kolom n}
        a[i,j]:=a[i,j] - m*a[k,j];
      {endfor}
      b[i]:=b[i] - m*b[k]; {eliminasi elemen vektor b pada baris i}
    end;
  end;
end;
{Solusi langsung didapat dari vektor kolom b}
for i:=1 to n do x[i]:=b[i];
end;

```

Seperti halnya metode eliminasi Gauss, tatanjang *pivoting* dan penskalaan juga dapat diterapkan pada metoda ini untuk memperkecil galat pembulatan.

Contoh 4.7

[CHA91] Selesaikan sistem persamaan lanjar di bawah ini dengan metode eliminasi Gauss- Jordan.

$$\begin{aligned} 3x_1 - 0.1x_2 - 0.2x_3 &= 7.85 \\ 0.1x_1 + 7x_2 - 0.3x_3 &= -19.3 \\ 0.3x_1 - 0.2x_2 + 10x_3 &= 71.4 \end{aligned}$$

Penyelesaian:

$$\left[\begin{array}{ccc|c} 3 & -0.1 & -0.2 & 7.85 \\ 0.1 & 7 & -0.3 & -19.3 \\ 0.3 & -0.2 & 10 & 71.4 \end{array} \right] \sim \left[\begin{array}{ccc|c} 1 & -0.0333333 & -0.0666667 & 2.61667 \\ 0.1 & 7 & -0.3 & -19.3 \\ 0.3 & -0.2 & 10 & 71.4 \end{array} \right]$$

$$\begin{array}{l} R_2 - 0.1 R_1 \\ R_3 - 0.3 R_1 \\ \sim \end{array} \left[\begin{array}{ccc|c} 1 & -0.0333333 & -0.0666667 & 2.61667 \\ 0 & 7.00333 & -0.2933333 & -19.5617 \\ 0 & -0.190000 & 10.0200 & 70.6150 \end{array} \right]$$

$$R_2 / 7.00333 \left[\begin{array}{ccc|c} 1 & -0.0333333 & -0.0666667 & 2.61667 \\ 0 & 1 & -0.0418848 & -2.79320 \\ 0 & -0.190000 & 10.0200 & 70.6150 \end{array} \right]$$

$$\begin{array}{l} R_1 - (-0.003333)R_2 \\ R_3 - (-0.190000)R_2 \\ \sim \end{array} \left[\begin{array}{ccc|c} 1 & 0 & -0.0680629 & 2.52356 \\ 0 & 1 & -0.0418848 & -2.79320 \\ 0 & 0 & 10.01200 & 70.0843 \end{array} \right]$$

$$R_3 / 10.0200 \quad \left[\begin{array}{ccc|c} 1 & 0 & -0.0680629 & 2.52356 \\ 0 & 1 & -0.0418848 & -2.79320 \\ 0 & 0 & 1 & 7.00003 \end{array} \right]$$

$$R_1 - (-0.0680629) R_3 \quad \left[\begin{array}{ccc|c} 1 & 0 & 0 & 3.00000 \\ 0 & 1 & 0 & -2.50001 \\ 0 & 0 & 1 & 7.00003 \end{array} \right]$$

$$R_2 - (-0.0418848) R_2 \quad \left[\begin{array}{ccc|c} 1 & 0 & 0 & 3.00000 \\ 0 & 1 & 0 & -2.50001 \\ 0 & 0 & 1 & 7.00003 \end{array} \right]$$

Solusi: $x_1 = 3.00000$
 $x_2 = -2.50001$
 $x_3 = 7.00003$

■

Penyelesaian SPL dengan metode eliminasi Gauss-Jordan membutuhkan jumlah komputasi yang lebih banyak daripada metode eliminasi Gauss. Karena alasan itu, metode eliminasi Gauss sudah cukup memuaskan untuk digunakan dalam penyelesaian SPL. Namun metode eliminasi Gauss-Jordan merupakan dasar pembentukan matriks balikan yang akan dibahas di bawah ini.

Matriks Balikan (*inverse matrices*)

Matriks balikan, A^{-1} , banyak dipakai dalam pengolahan matriks. Misalnya dalam pengukuran statistik, pencocokan fungsi pada data hasil pengamatan menggunakan metode kuadrat terkecil (*least square*). Di sini, nilai A^{-1} memberikan informasi tentang galat mutlak yang dikandung data. Selain itu, matriks balikan juga dapat dipakai untuk menghitung solusi sistem persamaan lanjar (akan dibahas pada metode matriks balikan). Akan ditunjukkan juga bahwa matriks balikan dapat diperoleh dengan metode eliminasi Gauss-Jordan. Tetapi sebelum membahasnya, ingatlah kembali cara menghitung matriks balikan untuk matriks 2×2 .

Untuk matriks 2×2 ,

$$A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$$

matriks balikannya adalah

$$A^{-1} = \frac{1}{a_{11}a_{22} - a_{12}a_{21}} \begin{bmatrix} a_{22} & -a_{12} \\ -a_{21} & a_{11} \end{bmatrix}, \quad a_{11}a_{22} - a_{12}a_{21} \neq 0.$$

Nilai $a_{11}a_{22} - a_{12}a_{12}$ ini disebut **determinan**. Determinan dilambangkan dengan dua buah garis tegak ($| |$). Lebih jauh tentang determinan ini akan dijelaskan pada bagian lain bab ini. Bila determinan $A = 0$, matriks A tidak mempunya balikan, sehingga dinamakan *matriks singular*. Sistem persamaan lanjar yang mempunyai matriks A singular (sistem singular) tidak mempunyai solusi yang unik, yaitu solusinya banyak atau solusinya tidak ada.

Untuk matriks $n \times n$, matriks balikannya dapat diperoleh dengan metode eliminasi Gauss-Jordan, yaitu:

$$\begin{array}{c} [A \mid I] \xrightarrow{\text{eliminasi G - J}} [I \mid A^{-1}] \\ \left[\begin{array}{cccc|ccc} a_{11} & a_{12} & \dots & a_{1n} & 1 & 0 & \dots & 0 \\ a_{21} & a_{22} & \dots & a_{2n} & 0 & 1 & \dots & 0 \\ \vdots & \vdots & & \vdots & \vdots & & & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} & 0 & 0 & \dots & 1 \end{array} \right] \xrightarrow{\quad} \left[\begin{array}{cccc|ccc} 1 & 0 & \dots & 0 & p_{11} & p_{12} & \dots & p_{1n} \\ 0 & 1 & \dots & 0 & p_{21} & p_{22} & \dots & p_{2n} \\ \vdots & & & \vdots & \vdots & & & \vdots \\ 0 & 0 & \dots & 1 & p_{n1} & p_{n2} & \dots & p_{nn} \end{array} \right] \\ A \qquad \qquad \qquad I \qquad \qquad \qquad I \qquad \qquad \qquad A^{-1} \end{array}$$

Contoh 4.8

Tentukan matriks balikan dari matriks A berikut

$$A = \begin{bmatrix} 1 & -1 & 2 \\ 3 & 0 & 1 \\ 1 & 0 & 2 \end{bmatrix}$$

Penyelesaian:

$$\begin{array}{c} \left[\begin{array}{ccc|ccc} 1 & -1 & 2 & 1 & 0 & 0 \\ 3 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 2 & 0 & 0 & 1 \end{array} \right] R_2 - 3R_1 \sim \left[\begin{array}{ccc|ccc} 1 & -1 & 2 & 1 & 0 & 0 \\ 0 & 3 & -5 & 3 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \end{array} \right] \\ \sim \dots \sim \left[\begin{array}{ccc|ccc} 1 & 0 & 0 & 0 & 0.4 & -0.2 \\ 0 & 1 & 0 & -1 & 0 & 1 \\ 0 & 0 & 1 & 0 & -0.5 & 0.6 \end{array} \right] \end{array}$$

Jadi, matriks balikan dari A adalah

$$A^{-1} = \begin{bmatrix} 0 & 0.4 & -0.2 \\ -1 & 0 & 1 \\ 0 & -0.5 & 0.6 \end{bmatrix}$$

■

Penerapan tata-ancang *pivoting* dan penggunaan bilangan berketalitian ganda dapat memperbaiki hasil matriks balikan.

4.4 Metode Matriks Balikan

Misalkan A^{-1} adalah matriks balikan dari A . Hasil kali A dengan A^{-1} menghasilkan matriks identitas I ,

$$AA^{-1} = A^{-1}A = I \quad (\text{P.4.6})$$

Bila matriks A dikalikan dengan I akan menghasilkan matriks A sendiri,

$$AI = IA = A \quad (\text{P.4.7})$$

Berdasarkan dua kesamaan di atas, sistem persamaan lanjar $Ax = b$ dapat diselesaikan sebagai berikut:

$$\begin{aligned} Ax &= b \\ A^{-1}Ax &= A^{-1}b && \{\text{kalikan kedua ruas dengan } A^{-1}\} \\ Ix &= A^{-1}b \\ x &= A^{-1}b \end{aligned} \quad (\text{P.4.8})$$

Jadi, penyelesaian sistem persamaan lanjar $Ax = b$ adalah $x = A^{-1}b$ dengan syarat A^{-1} ada. Cara penyelesaian dengan mengalikan matriks A^{-1} dengan b itu dinamakan **metode matriks balikan**. Tetapi, penyelesaian dengan SPL metode matriks balikan tidak lebih mangkus daripada metode eliminasi Gauss, sebab lebih banyak proses komputasi yang dibutuhkan. Metode matriks balikan baru mangkus bila digunakan untuk penyelesaian sejumlah SPL dengan matriks A yang sama tetapi dengan vektor kolom b yang berbeda-beda:

$$\begin{aligned} Ax &= b_I \\ Ax &= b_{II} \\ Ax &= b_{III} \\ \dots \text{dst} \end{aligned}$$

Sekali A^{-1} telah diperoleh, maka ia dapat dipakai untuk menyelesaikan sejumlah SPL tersebut.

Contoh 4.9

Selesaikan sistem persamaan lanjar

$$\begin{aligned}x_1 - x_2 + 2x_3 &= 5 \\3x_1 + x_3 &= 10 \\x_1 + 2x_3 &= 5\end{aligned}$$

dengan metode matriks balikan.

Penyelesaian:

$$\left[\begin{array}{ccc|ccc} 1 & -1 & 2 & 1 & 0 & 0 \\ 3 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 2 & 0 & 0 & 1 \end{array} \right] \xrightarrow{\substack{R_2 - 3R_1 \\ R_3 - R_1}} \left[\begin{array}{ccc|ccc} 1 & -1 & 2 & 1 & 0 & 0 \\ 0 & 3 & -5 & -3 & 1 & 0 \\ 0 & 1 & 0 & -1 & 0 & 1 \end{array} \right] \sim \dots \sim \left[\begin{array}{ccc|ccc} 1 & 0 & 0 & 0 & 0.4 & -0.2 \\ 0 & 1 & 0 & -1 & 0 & 1 \\ 0 & 0 & 1 & 0 & -0.2 & 0.6 \end{array} \right] \xrightarrow{-1} A$$

Solusinya adalah $x = A^{-1} b$.

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 0 & 0.4 & -0.2 \\ -1 & 0 & 1 \\ 0 & -0.2 & 0.6 \end{bmatrix} \begin{bmatrix} 5 \\ 10 \\ 5 \end{bmatrix} = \begin{bmatrix} 0 & + & 4 & - & 1 \\ -5 & + & 0 & + & 5 \\ 0 & - & 2 & + & 3 \end{bmatrix} = \begin{bmatrix} 3 \\ 0 \\ 1 \end{bmatrix} \blacksquare$$

Perlu diperhatikan, apabila selama pembentukan matriks balikan terdapat proses *pivoting* (pertukaran baris), baris-baris pada b juga harus dipertukarkan.

4.5 Metode Dekomposisi LU

Jika matriks A non-singular maka ia dapat difaktorkan (diuraikan atau dekomposisi) menjadi matriks segitiga bawah L (*lower*) dan matriks segitiga atas U (*upper*):

$$A = LU \quad (\text{P.4.9})$$

Dalam bentuk matriks, pemfaktoran ini ditulis sebagai

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \dots & a_{2n} \\ a_{31} & a_{32} & a_{33} & \dots & a_{3n} \\ \vdots & \vdots & \vdots & & \vdots \\ a_{n1} & a_{n2} & a_{n3} & \dots & a_{nn} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 \\ l_{21} & 1 & 0 & \dots & 0 \\ l_{31} & l_{32} & 1 & \dots & 0 \\ \vdots & & \vdots & & \vdots \\ l_{n1} & l_{n2} & l_{n3} & \dots & 1 \end{bmatrix} \begin{bmatrix} u_{11} & u_{12} & u_{13} & \dots & u_{1n} \\ 0 & u_{22} & u_{23} & \dots & u_{2n} \\ 0 & 0 & u_{33} & \dots & u_{3n} \\ \vdots & & \vdots & & \vdots \\ 0 & 0 & 0 & \dots & u_{nn} \end{bmatrix}$$

Pada matriks segitiga bawah L , semua elemen diagonal adalah 1, sedangkan pada matriks U tidak ada aturan khusus pada elemen diagonalnya¹.

Sebagai contoh, matriks 3×3 di bawah ini difaktorkan menjadi :

$$\begin{bmatrix} 2 & -1 & -1 \\ 0 & -4 & 2 \\ 6 & -3 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 3 & 0 & 1 \end{bmatrix} \begin{bmatrix} 2 & -1 & -1 \\ 0 & -4 & 2 \\ 0 & 0 & 4 \end{bmatrix}$$

Metode pemfaktoran A menjadi L dan U akan dijelaskan kemudian. Sekali A difaktorkan menjadi L dan U , kedua matriks tersebut dapat digunakan untuk menyelesaikan $Ax = b$. Metode penyelesaian SPL dengan cara ini dikenal dengan nama **metode dekomposisi LU**. Metode ini dinamakan juga **metode pemfaktoran segitiga** (*triangular factorization*). Nanti akan ditunjukkan bahwa metode eliminasi Guass merupakan suatu dekomposisi LU dari matriks A .

Penyelesaian $Ax = b$ dengan metode dekomposisi LU adalah sebagai berikut. Tinjau sistem persamaan lanjar

$$Ax = b$$

Faktorkan A menjadi L dan U sedemikian sehingga

$$A = LU$$

Jadi,

$$\begin{aligned} Ax &= b \\ LUx &= b \end{aligned} \tag{P.4.10}$$

¹ Pada beberapa buku, yang tertera adalah kebalikannya: semua elemen diagonal dari matriks U adalah 1, sedangkan elemen diagonal matriks L bebas. Hal ini tidak masalah sebab jika L dan U dikalikan, hasilnya tetap sama dengan matriks A .

Misalkan

$$Ux = y \quad (\text{P.4.11})$$

maka

$$Ly = b \quad (\text{P.4.12})$$

Untuk memperoleh y_1, y_2, \dots, y_n , kita menggunakan teknik penyulihan maju (*forward substitution*):

$$Ly = b \rightarrow \begin{bmatrix} 1 & 0 & 0 & \dots & 0 \\ l_{21} & 1 & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ l_{n1} & l_{n2} & l_{n3} & \dots & 1 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix} \rightarrow \begin{array}{l} \text{diperoleh } y_1, y_2, \dots, \\ y_n \text{ dengan teknik} \\ \text{penyulihan maju} \end{array}$$

Dan untuk memperoleh solusi SPL, x_1, x_2, \dots, x_n , kita menggunakan teknik penyulihan mundur (*backward substitution*):

$$Ux = y \rightarrow \begin{bmatrix} u_{11} & u_{12} & u_{13} & \dots & u_{1n} \\ 0 & u_{22} & u_{23} & \dots & u_{2n} \\ \dots & & & \ddots & \\ 0 & 0 & 0 & \dots & u_{nn} \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix} \begin{array}{l} \text{diperoleh} \\ x_1, x_2, \dots, x_n \\ \text{dengan teknik} \\ \text{penyulihan} \\ \text{mundur} \end{array}$$

Jadi, langkah-langkah menghitung solusi SPL dengan metode dekomposisi *LU* dapat diringkas sebagai berikut:

1. Bentuklah matriks *L* dan *U* dari *A*
2. Pecahkan $Ly = b$, lalu hitung y dengan teknik penyulihan maju
3. Pecahkan $Ux = y$, lalu hitung x dengan teknik penyulihan mundur

Sama halnya dengan metode matriks balikan, metode dekomposisi *LU* akan mangkus bila digunakan untuk menyelesaikan sejumlah SPL dengan matriks *A* yang sama tetapi dengan *b* berbeda-beda. Sekali *A* difaktorkan menjadi *L* dan *U*, keduanya dapat digunakan untuk menghitung solusi sejumlah SPL tersebut. Metode dekomposisi *LU* merupakan metode yang paling populer untuk memecahkan sistem persamaan lanjar.

Terdapat dua metode untuk memfaktorkan A atas L dan U :

1. Metode LU Gauss.
2. Metode reduksi Crout.

Masing-masing metode pemfaktoran kita bahas di bawah ini.

4.5.1 Pemfaktoran dengan Metode LU Gauss

Walaupun tidak ada hubungannya dengan dekomposisi LU , metode eliminasi Gauss dapat digunakan untuk memfaktorkan A menjadi L dan U (karena itulah metode pemfaktoran ini kita namakan metode LU Gauss). Di dalam upabab ini juga akan ditunjukkan bahwa sebenarnya metode eliminasi Gauss dapat dinyatakan sebagai dekomposisi LU .

Misalkan matriks A berukuran 4×4 difaktorkan atas L dan U ,

$$A = LU$$

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ m_{21} & 1 & 0 & 0 \\ m_{31} & m_{32} & 1 & 0 \\ m_{41} & m_{42} & m_{43} & 1 \end{bmatrix} \begin{bmatrix} u_{11} & u_{12} & u_{13} & u_{14} \\ 0 & u_{22} & u_{23} & u_{24} \\ 0 & 0 & u_{33} & u_{34} \\ 0 & 0 & 0 & u_{44} \end{bmatrix}$$

Di sini kita menggunakan simbol m_{ij} ketimbang l_{ij} , karena nilai l_{ij} berasal dari faktor pengali (m_{ij}) pada proses eliminasi Gauss. Langkah-langkah pembentukan L dan U dari matriks A adalah sebagai berikut:

1. Nyatakan A sebagai $A = IA$

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \dots & a_{2n} \\ a_{31} & a_{32} & a_{33} & \dots & a_{3n} \\ \vdots & & \vdots & & \vdots \\ a_{n1} & a_{n2} & a_{n3} & \dots & a_{nn} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & & \vdots & & \vdots \\ 0 & 0 & 0 & \dots & 1 \end{bmatrix} \begin{bmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \dots & a_{2n} \\ a_{31} & a_{32} & a_{33} & \dots & a_{3n} \\ \vdots & & \vdots & & \vdots \\ a_{n1} & a_{n2} & a_{n3} & \dots & a_{nn} \end{bmatrix}$$

2. Eliminasikan matriks A di ruas kanan menjadi matriks segitiga atas U . Tempatkan faktor pengali m_{ij} pada posisi l_{ij} di matriks I .

3. Setelah seluruh proses eliminasi Gauss selesai, matriks I menjadi matriks L , dan matriks A di ruas kanan menjadi matriks U .

Di bawah ini diberikan dua contoh pemfaktoran A dengan metode ini, masing-masing untuk kasus tanpa *pivoting* dan dengan *pivoting*.

Contoh 4.10 (LU Gauss naif)

Faktorkan matriks A berikut dengan metode LU Gauss:

$$A = \begin{bmatrix} 4 & 3 & -1 \\ -2 & -4 & 5 \\ 1 & 2 & 6 \end{bmatrix}$$

Penyelesaian:

$$A = \begin{bmatrix} 4 & 3 & -1 \\ -2 & -4 & 5 \\ 1 & 2 & 6 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 4 & 3 & -1 \\ -2 & -4 & 5 \\ 1 & 2 & 6 \end{bmatrix}$$

Eliminasikan matriks A di ruas kanan menjadi matriks segitiga atas U , dan tempatkan faktor pengali m_{ij} pada posisi l_{ij} di matriks L .

$$\begin{bmatrix} 4 & 3 & -1 \\ -2 & -4 & 5 \\ 1 & 2 & 6 \end{bmatrix} \xrightarrow[R_2 - (-2/4)R_1]{\sim} \begin{bmatrix} 4 & 3 & -1 \\ 0 & -2.5 & 4.5 \\ 1 & 2 & 6 \end{bmatrix} \xrightarrow[R_3 - (1/4)R_1]{\sim} \begin{bmatrix} 4 & 3 & -1 \\ 0 & -2.5 & 4.5 \\ 0 & 1.25 & 6.25 \end{bmatrix}$$

Tempatkan $m_{21} = -2/4 = 0.5$ dan $m_{31} = 1/4 = 0.25$ ke dalam matriks L :

$$L = \begin{bmatrix} 1 & 0 & 0 \\ -0.5 & 1 & 0 \\ 0 & m_{32} & 1 \end{bmatrix}$$

Teruskan proses eliminasi Gauss pada matriks A ,

$$\begin{bmatrix} 4 & 3 & -1 \\ 0 & -2.5 & 4.5 \\ 0 & 1.25 & 6.25 \end{bmatrix} \xrightarrow[R_3 - (1.25/-2.5)R_2]{\sim} \begin{bmatrix} 4 & 3 & -1 \\ 0 & -2.5 & 4.5 \\ 0 & 0 & 8.5 \end{bmatrix} = U$$

Tempatkan $m_{32} = 1.25/-2.5 = -0.5$ ke dalam matriks L :

$$L = \begin{bmatrix} 1 & 0 & 0 \\ -0.5 & 1 & 0 \\ 0.25 & -0.5 & 1 \end{bmatrix}$$

Jadi,

$$A = \begin{bmatrix} 4 & 3 & -1 \\ -2 & -4 & 5 \\ 1 & 2 & 6 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ -0.5 & 1 & 0 \\ 0.25 & -0.5 & 1 \end{bmatrix} \begin{bmatrix} 4 & 3 & -1 \\ 0 & -2.5 & 4.5 \\ 0 & 0 & 8.5 \end{bmatrix} \blacksquare$$

Contoh 4.11 (LU Gauss dengan tata-ancang pivoting)

Faktorkan matriks A berikut

$$A = \begin{bmatrix} 1 & 1 & -1 \\ 2 & 2 & 1 \\ -1 & 1 & 1 \end{bmatrix} \quad b = \begin{bmatrix} 1 \\ 5 \\ 1 \end{bmatrix}$$

lalu pecahkan sistem $Ax = b$.

Penyelesaian:

Eliminasikan matriks A di ruas kanan menjadi matriks segitiga atas U , dan tempatkan faktor pengali m_{ij} pada posisi l_{ij} di matriks L .

$$\begin{bmatrix} 1 & 1 & -1 \\ 2 & 2 & 1 \\ -1 & 1 & 1 \end{bmatrix} \quad R_2 - (2)R_1 \quad \begin{bmatrix} 1 & 1 & -1 \\ 0 & 0 & 3 \\ 0 & 2 & 0 \end{bmatrix}$$

$$\quad R_3 - (-1/1)R_1$$

Tempatkan $m_{21} = 2$ dan $m_{31} = 1/1 = 1$ ke dalam matriks L :

$$L = \begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ -1 & m_{32} & 1 \end{bmatrix}$$

Teruskan proses eliminasi Gauss pada matriks A . Dalam hal ini ada *pivoting* karena calon *pivot* bernilai 0, sehingga baris kedua dipertukarkan dengan baris ketiga:

$$\begin{bmatrix} 1 & 1 & -1 \\ 0 & \mathbf{0} & 3 \\ 0 & 2 & 0 \end{bmatrix} \quad R_2 \Leftrightarrow R_3 \quad \begin{bmatrix} 1 & 1 & -1 \\ 0 & 2 & 0 \\ 0 & 0 & 3 \end{bmatrix}$$

Jangan lupa mempertukarkan juga $R_2 \Leftrightarrow R_3$ pada matriks L , kecuali elemen diagonalnya

$$L = \begin{bmatrix} 1 & 0 & 0 \\ \boxed{2} & 1 & 0 \\ -1 & m_{32} & 1 \end{bmatrix} \quad R_2 \Leftrightarrow R_3 \quad \begin{bmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ 2 & m_{32} & 1 \end{bmatrix}$$

Jangan lupa mempertukarkan juga $R_2 \Leftrightarrow R_3$ pada vektor b ,

$$b = \begin{bmatrix} 1 \\ 5 \\ 1 \end{bmatrix} \quad R_2 \Leftrightarrow R_3 \quad \begin{bmatrix} 1 \\ 1 \\ 5 \end{bmatrix}$$

Teruskan proses eliminasi Gauss pada matriks A :

$$R_3 - (\frac{0}{2})R_2 \quad \begin{bmatrix} 1 & 1 & -1 \\ 0 & 2 & 0 \\ 0 & 0 & 3 \end{bmatrix} = U$$

Tempatkan $m_{32} = 0/2 = 0$ ke dalam matriks L :

$$L = \begin{bmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ 2 & 0 & 1 \end{bmatrix}$$

Jadi,

$$A = \begin{bmatrix} 1 & 1 & -1 \\ -1 & 1 & 1 \\ 2 & 2 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ 2 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 & -1 \\ 0 & 2 & 0 \\ 0 & 0 & 3 \end{bmatrix}$$

Berturut-turut dihitung y dan x sebagai berikut:

$$Ly = b \quad \begin{bmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ 2 & 0 & 1 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 5 \end{bmatrix}$$

y_1, y_2 , dan y_3 dihitung dengan teknik penyulihan maju:

$$\begin{array}{lcl} y_1 & = 1 \\ -y_1 + y_2 & = 1 & \rightarrow y_2 = 1 + y_1 = 1 + 1 = 2 \\ 2y_1 + 0y_2 + y_3 & = 5 & \rightarrow y_3 = 5 - 2y_1 = 3 \end{array}$$

$$Ux = y \quad \left[\begin{array}{ccc} 1 & 1 & -1 \\ 0 & 2 & 0 \\ 0 & 0 & 3 \end{array} \right] \left[\begin{array}{c} x_1 \\ x_2 \\ x_3 \end{array} \right] = \left[\begin{array}{c} 1 \\ 2 \\ 3 \end{array} \right]$$

x_1, x_2 , dan x_3 dihitung dengan teknik penyulihan mundur:

$$\begin{array}{lcl} 3x_3 & = 3 & \rightarrow x_3 = 1 \\ 2x_2 + 0x_3 & = 2 & \rightarrow x_2 = 1 \\ x_1 + x_2 - x_3 & = 1 & \rightarrow x_1 = 1 \end{array}$$

Jadi, solusi sistem persamaan lanjar di atas adalah $x = (1, 1, 1)^T$. ■

Pertukaran baris untuk matriks yang berukuran besar diperlihatkan oleh matriks di bawah ini:

$$\left[\begin{array}{cccccc} a_1 & a_2 & a_3 & a_4 & a_5 & a_6 \\ 0 & b_2 & b_3 & b_4 & b_5 & b_6 \\ 0 & 0 & c_3 & c_4 & c_5 & c_6 \\ 0 & 0 & 0 & 0 & d_5 & d_6 \\ 0 & 0 & 0 & e_4 & e_5 & e_6 \\ 0 & 0 & 0 & f_4 & f_5 & f_6 \end{array} \right] \xrightarrow[R_5 \Leftrightarrow R_4]{(*)} \left[\begin{array}{cccccc} a_1 & a_2 & a_3 & a_4 & a_5 & a_6 \\ 0 & b_2 & b_3 & b_4 & b_5 & b_6 \\ 0 & 0 & c_3 & c_4 & c_5 & c_6 \\ 0 & 0 & 0 & e_4 & e_5 & e_6 \\ 0 & 0 & 0 & 0 & d_5 & d_6 \\ 0 & 0 & 0 & f_4 & f_5 & f_6 \end{array} \right]$$

Maka, baris ke-5 dan baris ke-4 pada matriks L juga harus dipertukarkan:

$$\left[\begin{array}{cccccc} 1 & 0 & 0 & 0 & 0 & 0 \\ m_{21} & 1 & 0 & 0 & 0 & 0 \\ m_{31} & m_{32} & 1 & 0 & 0 & 0 \\ \boxed{m_{41} \ m_{42} \ m_{43}} & 1 & 0 & 0 \\ \boxed{m_{51} \ m_{52} \ m_{53}} & x & 1 & 0 \\ m_{61} & m_{62} & m_{63} & x & x & 1 \end{array} \right] \xrightarrow[R_5 \Leftrightarrow R_4]{(*)} \left[\begin{array}{cccccc} 1 & 0 & 0 & 0 & 0 & 0 \\ m_{21} & 0 & 0 & 0 & 0 & 0 \\ m_{31} & m_{32} & 1 & 0 & 0 & 0 \\ \boxed{m_{51} \ m_{52} \ m_{53}} & 1 & 0 & 0 \\ \boxed{m_{41} \ m_{42} \ m_{43}} & x & 1 & 0 \\ m_{61} & m_{62} & m_{63} & x & x & 1 \end{array} \right]$$

4.5.2 Metode Reduksi Crout

Meskipun metode *LU* Gauss dikenal paling baik untuk melakukan dekomposisi *LU*, terdapat metode lain yang digunakan secara luas, yaitu metode reduksi (dekomposisi) Crout (atau **metode reduksi Cholesky** atau **metode Dolittle**).

Dalam membahas metode reduksi Crout, tinjau matriks 3×3 berikut:

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \quad L = \begin{bmatrix} 1 & 0 & 0 \\ l_{21} & 1 & 0 \\ l_{31} & l_{32} & 1 \end{bmatrix} \quad U = \begin{bmatrix} u_{11} & u_{12} & u_{13} \\ 0 & u_{22} & u_{23} \\ 0 & 0 & u_{33} \end{bmatrix}$$

Karena $LU = A$, maka hasil perkalian *L* dan *U* itu dapat ditulis sebagai

$$LU = \begin{bmatrix} u_{11} & u_{12} & u_{13} \\ l_{21}u_{11} & l_{21}u_{12} + u_{22} & l_{21}u_{13} + u_{23} \\ l_{31}u_{11} & l_{31}u_{12} + l_{32}u_{22} & l_{31}u_{13} + l_{32}u_{23} + u_{33} \end{bmatrix} = A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$$

Dari kesamaan dua buah matriks $LU = A$, diperoleh

$$u_{11} = a_{11}, \quad u_{12} = a_{12}, \quad u_{13} = a_{13} \quad \} \quad \text{Baris pertama } U$$

$$\begin{aligned} l_{21}u_{11} &= a_{21} \rightarrow l_{21} = \frac{a_{21}}{u_{11}} \\ l_{31}u_{11} &= a_{31} \rightarrow l_{31} = \frac{a_{31}}{u_{11}} \end{aligned} \quad \} \quad \text{Kolom pertama } L$$

$$\begin{aligned} l_{21}u_{12} + u_{22} &= a_{22} \rightarrow u_{22} = a_{22} - l_{21}u_{12} \\ l_{21}u_{13} + u_{23} &= a_{23} \rightarrow u_{23} = a_{23} - l_{21}u_{13} \end{aligned} \quad \} \quad \text{Baris kedua } U$$

$$l_{31}u_{12} + l_{32}u_{22} = a_{32} \rightarrow l_{32} = \frac{a_{32} - l_{31}u_{12}}{u_{22}} \quad \} \quad \text{Kolom kedua } L$$

$$l_{31}u_{13} + l_{32}u_{23} + u_{33} = a_{33} \rightarrow u_{33} = a_{33} - (l_{31}u_{13} + l_{32}u_{23}) \quad \} \quad \text{Baris ketiga } U$$

Kita perhatikan ada urutan pola teratur dalam menemukan elemen-elemen L dan U , yaitu:

- elemen-elemen baris pertama dari U
- elemen-elemen baris pertama dari L
- elemen-elemen baris kedua dari U
- elemen-elemen baris kedua L
- ...
- elemen-elemen baris ke- k dari U
- elemen-elemen baris ke- k dari L

Rumus umum menghitung u dan l untuk sistem dengan matriks A yang berukuran 3×3 dapat ditulis sebagai berikut:

$$u_{pj} = a_{pj} - \sum_{k=1}^{p-1} l_{pk} u_{kj} \quad \begin{matrix} p = 1, 2, 3, \dots, n \\ j = p, p+1, \dots, n \end{matrix} \quad (\text{P.4.13})$$

dan

$$l_{iq} = \frac{a_{iq} - \sum_{k=1}^{q-1} l_{ik} u_{kj}}{u_{qq}} \quad \begin{matrix} q = 1, 2, 3, \dots, n-1 \\ i = q+1, q+2, \dots, n \\ \text{dengan syarat } u_{qq} \neq 0 \end{matrix} \quad (\text{P.4.14})$$

Contoh 4.12

Selesaikan

$$\begin{aligned} x_1 + x_2 - x_3 &= 1 \\ 2x_1 + 2x_2 + x_3 &= 5 \\ -x_1 + x_2 + 2x_3 &= 5 \end{aligned}$$

dengan metode dekomposisi LU , yang dalam hal ini L dan U dihitung dengan metode reduksi Crout.

Penyelesaian:

$$A = \begin{bmatrix} 1 & 1 & -1 \\ 2 & 2 & 1 \\ -1 & 1 & 1 \end{bmatrix} \quad b = \begin{bmatrix} 1 \\ 5 \\ 1 \end{bmatrix}$$

Diperoleh:

$$\begin{aligned} u_{11} &= a_{11} = 1 \\ u_{12} &= a_{12} = 1 \\ u_{13} &= a_{13} = -1 \end{aligned}$$

$$\begin{aligned} l_{21} &= a_{21}/u_{11} = 2/1 = 2 \\ l_{31} &= a_{31}/u_{11} = -1/1 = -1 \end{aligned}$$

$$u_{22} = a_{22} - l_{21}u_{12} = 2 - 2 \cdot 1 = 0$$

Karena u_{qq} tidak boleh nol, lakukan pertukaran baris, baik untuk matriks A maupun untuk vektor b :

<u>Matriks A</u>	<u>Vektor b</u>
$R_2 \Leftrightarrow R_3$	$R_2 \Leftrightarrow R_3$
$\begin{bmatrix} 1 & 1 & -1 \\ -1 & 1 & 1 \\ 2 & 2 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 1 \\ 5 \end{bmatrix}$

Hitung kembali nilai l_{21} , l_{31} , dan u_{22} (Perhatikan bahwa nilai u_{11}, u_{12}, u_{13} tidak berubah)

$$\begin{aligned} l_{21} &= a_{21}/u_{11} = -1/1 = -1 \\ l_{31} &= a_{31}/u_{11} = 2/1 = 2 \end{aligned}$$

$$\begin{aligned} u_{22} &= a_{22} - l_{21}u_{12} = 1 - (-1)(1) = 1 + 1 = 2 \\ u_{23} &= a_{23} - l_{21}u_{13} = 1 - (-1)(-1) = 1 - 1 = 0 \end{aligned}$$

$$l_{32} = \frac{a_{32} - l_{31}u_{12}}{u_{22}} = \frac{2 - 2(1)}{2} = 0$$

Diperoleh L dan U sebagai berikut,

$$U = \begin{bmatrix} 1 & 1 & -1 \\ 0 & 2 & 0 \\ 0 & 0 & 3 \end{bmatrix} \quad L = \begin{bmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ 2 & 0 & 1 \end{bmatrix} \quad \text{dan } b = \begin{bmatrix} 1 \\ 1 \\ 5 \end{bmatrix}$$

Berturut-turut dihitung y dan x sebagai berikut:

$$Ly = b \longrightarrow \begin{bmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ 2 & 0 & 1 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 5 \end{bmatrix}$$

y_1 , y_2 , dan y_3 dihitung dengan teknik penyulihan maju:

$$\begin{aligned} y_1 &= 1 \\ -y_1 + y_2 &= 1 \rightarrow y_2 = 1 + y_1 = 1 + 1 = 2 \\ 2y_1 + 0y_2 + y_3 &= 5 \rightarrow y_3 = 5 - 2y_1 = 3 \end{aligned}$$

$$Ux = y \quad \left[\begin{array}{ccc} 1 & 1 & -1 \\ 0 & 2 & 0 \\ 0 & 0 & 3 \end{array} \right] \quad \left[\begin{array}{c} x_1 \\ x_2 \\ x_3 \end{array} \right] = \left[\begin{array}{c} 1 \\ 2 \\ 3 \end{array} \right]$$

x_1, x_2 , dan x_3 dihitung dengan teknik penyulihan mundur:

$$\begin{aligned} 3x_3 &= 3 & \rightarrow x_3 &= 1 \\ 2x_2 + 0x_3 &= 2 & \rightarrow x_2 &= 1 \\ x_1 + x_2 - x_3 &= 1 & \rightarrow x_1 &= 1 \end{aligned}$$

Jadi, solusi sistem persamaan lanjar di atas adalah $x = (1, 1, 1)^T$. ■

Jika diamati elemen segitiga bawah pada matriks U semuanya bernilai nol, sehingga ruang yang tidak terpakai itu dapat dipakai untuk menyimpan elemen matriks L . Elemen diagonal matriks L seluruhnya 1, jadi tidak perlu disimpan (*default*). Dengan demikian, penyimpanan elemen L dan U pada satu matriks dapat menghemat penggunaan memori. Selain itu, matriks A hanya dipakai sekali untuk memperoleh L dan U , sesudah itu tidak dipakai lagi. Dengan demikian, setelah L dan U diperoleh, elemennya dapat dipindahkan ke dalam A . Karena alasan ini, maka metode dekomposisi LU dinamakan juga metode kompaksi memori.

4.6 Determinan

Pada pembahasan matriks balikan kita telah menyinggung sedikit mengenai determinan. Menghitung determinan matriks 2×2 sangat mudah dan selalu diajarkan di sekolah menengah. Misalkan A adalah matriks

$$A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$$

maka determinan matriks A adalah

$$\det(A) = \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} = a_{11}a_{22} - a_{12}a_{21}$$

Begitupun menghitung determinan untuk matriks 3×3 ,

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$$

maka determinannya dihitung dengan aturan Cramer:

$$\begin{aligned} \det(A) &= \begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix} \\ &= a_{11} \begin{bmatrix} a_{22} & a_{23} \\ a_{32} & a_{33} \end{bmatrix} - a_{12} \begin{bmatrix} a_{21} & a_{23} \\ a_{31} & a_{33} \end{bmatrix} + a_{13} \begin{bmatrix} a_{21} & a_{22} \\ a_{31} & a_{32} \end{bmatrix} \\ &= a_{11}(a_{22}a_{33} - a_{23}a_{32}) - a_{12}(a_{21}a_{33} - a_{23}a_{31}) + a_{13}(a_{21}a_{32} - a_{22}a_{31}) \end{aligned}$$

Menghitung determinan untuk matriks $n \times n$ dengan aturan Cramer menjadi tidak praktis lagi. Metode eliminasi Gauss dapat diterapkan untuk menghitung determinan matriks $n \times n$. Determinannya dapat dihitung setelah ia ditransformasi menjadi matriks segitiga atas U . Pertama-tama kita lihat dulu dua hukum penting determinan [NAK92]:

$$\text{Hukum 1: } \det(BC) = \det(B) \times \det(C)$$

yaitu, determinan dari perkalian dua buah matriks sama dengan perkalian determinan masing-masing matriks.

$$\text{Hukum 2: } \det(M) = \text{hasil kali semua elemen diagonal } M \text{ jika } M \text{ adalah matriks segitiga atas atau matriks segitiga bawah.}$$

Jadi, jika semua elemen diagonal matriks adalah satu, maka determinannya sama dengan satu. Dalam menghitung determinan, pertimbangkan dua kasus berikut berikut: (i) bila eliminasi Gauss-nya tanpa *pivoting* dan (ii) bila eliminasi Gauss-nya dengan *pivoting*.

Kasus 1: Bila eliminasi Gauss tidak menerapkan tatancang *pivoting*.

Jika *pivoting* tidak diterapkan, determinan matriks A adalah

$$\det(A) = \det(LU) = \det(L) \times \det(U) = \det(U) = u_{11} u_{22} u_{33} \dots u_{nn} \quad (\text{P.4.15})$$

yang dalam hal ini $\det(L) = 1$ sebab semua elemen diagonal L adalah satu.

Kasus 2: Bila eliminasi Gauss menerapkan tatancang *pivoting*.

Tatancang *pivoting* mengakibatkan pertukaran baris. Dekomposisi LU dengan *pivoting* setara dengan mengerjakan dua proses terpisah berikut:

- (1) Transformasikan matriks A menjadi matriks A' dengan cara permutasi baris-baris matriks (sama dengan mengalikan A dengan matriks permutasi P),

$$A' = PA \quad \text{atau setara dengan } A = P^{-1} A' \quad (\text{P.4.16})$$

- (2) Dekomposisi A' menjadi LU tanpa *pivoting*

$$A' = LU$$

Dari (1) dan (2), L dan U dihubungkan dengan A oleh

$$A = P^{-1} A' = P^{-1} LU \quad (\text{P.4.17})$$

Determinan A dapat ditulis sebagai

$$\begin{aligned} \det(A) &= \det(P^{-1}) \times \det(L) \times \det(U) \\ &= \det(P^{-1}) \times 1 \times \det(U) \\ &= \det(P^{-1}) \times \det(U) \\ &= \alpha \det(U) \end{aligned}$$

yang dalam hal ini $\alpha = \det(P^{-1}) = -1$ atau 1 bergantung pada apakah *pivoting* sejumlah bilangan ganjil atau genap. Jika *pivoting* dilakukan sejumlah p kali, maka α dapat ditulis sebagai:

$$\alpha = (-1)^p$$

α bernilai 1 untuk p genap dan -1 untuk p ganjil. Karena itu,

$$\det(A) = (-1)^p \det(U) = (-1)^p u_{11} u_{22} u_{33} \dots u_{nn} \quad (\text{P.4.18})$$

Contoh 4.13

Hitung determinan matriks A berikut:

$$A = \begin{bmatrix} 2 & 3 & -1 \\ 4 & 4 & -3 \\ -2 & 3 & -1 \end{bmatrix}$$

Penyelesaian:

$$\begin{bmatrix} 2 & 3 & -1 \\ 4 & 4 & -3 \\ -2 & 3 & -1 \end{bmatrix} \xrightarrow{R_2 - \frac{4}{2}R_1} \begin{bmatrix} 2 & 3 & -1 \\ 0 & -2 & -1 \\ -2 & 3 & -1 \end{bmatrix} \xrightarrow{R_3 - \frac{-2}{2}R_1} \begin{bmatrix} 2 & 3 & -1 \\ 0 & -2 & -1 \\ 1 & 0 & -2 \end{bmatrix} \xrightarrow{R_3 - \frac{6}{-2}R_2} \begin{bmatrix} 2 & 3 & -1 \\ 0 & -2 & -1 \\ 0 & 0 & -5 \end{bmatrix}$$

Tidak ada proses *pivoting* selama eliminasi Gauss, maka

$$\det(A) = (2)(-2)(-5) = 20$$

■

Contoh 4.14

Hitung determinan matriks berikut

$$\begin{bmatrix} 1 & 2 & 1 \\ 3 & 6 & 0 \\ 2 & 8 & 4 \end{bmatrix}$$

Penyelesaian:

$$\begin{bmatrix} 1 & 2 & 1 \\ 3 & 6 & 0 \\ 2 & 8 & 4 \end{bmatrix} \xrightarrow{R_2 - \frac{3}{1}R_1} \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & -3 \\ 2 & 8 & 4 \end{bmatrix} \xrightarrow{R_3 - \frac{2}{1}R_1} \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & -3 \\ 0 & 4 & 2 \end{bmatrix} \xrightarrow{R_3 \leftrightarrow R_3} \begin{bmatrix} 1 & 2 & 1 \\ 0 & 4 & 2 \\ 0 & 0 & -3 \end{bmatrix} \xrightarrow{(*)} \begin{bmatrix} 1 & 2 & 1 \\ 0 & 4 & 2 \\ 0 & 0 & -3 \end{bmatrix}$$

Pivoting diterapkan satu kali ($p = 1$), sehingga determinan matriks A adalah

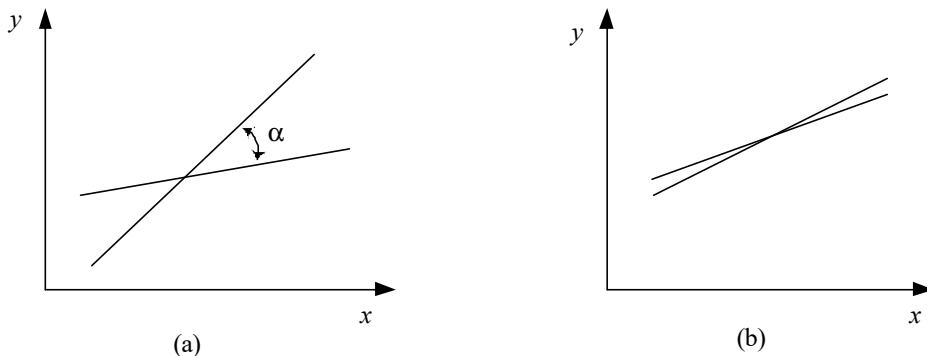
$$\det(A) = (-1)^1 (1)(4)(-3) = 12$$

■

4.7 Kondisi Buruk

Matriks A dikatakan berkondisi buruk (*ill condition*) jika terdapat sebuah vektor kolom b sehingga untuk perubahan kecil A atau b akan menghasilkan perubahan besar pada solusi $x = A^{-1}b$. Sistem $Ax = b$ dikatakan berkondisi buruk bila A berkondisi buruk. Apabila sistem $Ax = b$ berkondisi buruk, hasil perhitungannya mempunyai galat yang besar.

Sebagai contoh, dua persamaan lanjar dengan dua peubah yang tidak diketahui merepresentasikan dua buah garis lurus. Sistem berkondisi baik jika dan hanya jika sudut α antara kedua garis kecil, yaitu jika dan hanya jika kedua garis hampir sejajar. Perubahan kecil pada koefisien dapat menyebabkan pergeseran yang besar pada titik potong kedua garis (Gambar 4.3) [KRE88]. Untuk sistem persamaan yang lebih besar situasi tersebut pada prinsipnya sama, namun sayangnya tidak ada tafsiran geometrinya.



Gambar 4.3 (a) sistem berkondisi baik dan (b) sistem berkondisi buruk

Sebagai contoh, tinjau sistem persamaan lanjar berikut

$$(i) \quad \begin{aligned} x_1 + 2x_2 &= 10 \\ 1.1x_1 + 2x_2 &= 10.4 \end{aligned}$$

yang mempunyai solusi sejati $x_1 = 4$ dan $x_2 = 3$. Jika sekarang $a_{21} = 1.1$ diubah menjadi 1.05,

$$(ii) \quad \begin{aligned} x_1 + 2x_2 &= 10 \\ 1.05x_1 + 2x_2 &= 10.4 \end{aligned}$$

ternyata solusinya jauh berbeda, yaitu $x_1 = 8$ dan $x_2 = 1$. Penambahan sebesar ϵ pada koefisien 1.1 dapat dinyatakan sebagai berikut:

$$\begin{aligned}x_1 + 2x_2 &= 10 \\(1.1 + \epsilon)x_1 + 2x_2 &= 10.4\end{aligned}$$

yang mempunyai solusi

$$x_1 = \frac{0.4}{0.1 + \epsilon}$$

$$x_2 = \frac{0.6 + 10\epsilon}{0.2 + 2\epsilon}$$

Solusi ini memperlihatkan bahwa sistem berkondisi buruk sebab perubahan kecil ϵ menghasilkan perubahan besar pada solusi SPL. Pada contoh di atas, $\epsilon = -0.05$, sehingga $x_1 = 0.4/(0.1 - 0.05) = 8$ dan $x_2 = (0.6 - 10 \times 0.05)/(0.2 - 2 \times 0.05) = 1$.

Misalkan \hat{x} adalah solusi hampiran dari sistem

$$A\hat{x} = b \quad (\text{P.4.19})$$

Terhadap solusi hampiran ini terdapat sisa (residu) sebesar

$$r = b - A\hat{x}$$

Di sini

$$A\hat{x} = b - r \quad (\text{P.4.20})$$

Kurangi (P.4.19) dengan (P.4.20):

$$A(\hat{x} - x) = r \quad (\text{P.4.21})$$

Orang mungkin berpikir bahwa sisa r yang kecil menandakan bahwa \hat{x} lebih dekat ke x . Tetapi, kesimpulan ini ternyata salah. Penyelesaian kembali solusi hampiran ke SPL yang asli tidak dapat memberi petunjuk bahwa sistem berkondisi buruk. Bila $x_1 = 8$ dan $x_2 = 1$ disulih kembali ke dalam SPL (i):

$$\begin{aligned}8 + 2(1) &= 10 \\1.1(8) + 2(1) &= 10.8\end{aligned}$$

Residunya adalah

$$r = b - A\hat{x} = \begin{bmatrix} 10 \\ 10.4 \end{bmatrix} - \begin{bmatrix} 10 \\ 10.8 \end{bmatrix} = \begin{bmatrix} 0 \\ 0.4 \end{bmatrix}$$

ternyata sisa r cukup kecil meskipun $x_1 = 8$ dan $x_2 = 1$ bukan jawaban yang benar untuk masalah semula.

Contoh lainnya, tinjau sistem persamaan lanjar $Ax = b$ dengan

$$A = \begin{bmatrix} 3.02 & -1.05 & 2.53 \\ 4.33 & 0.56 & -1.78 \\ -0.83 & -0.54 & 1.47 \end{bmatrix} \quad \text{dan} \quad b = \begin{bmatrix} -1.61 \\ 7.23 \\ -3.38 \end{bmatrix}$$

Solusi sejatinya adalah $x = (1,2,-1)^T$. Solusi hampirannya, bila dihitung dengan metode eliminasi Gauss, adalah $\hat{x} = (0.880, -2.35, -2.66)^T$. Bila 3.02 pada matriks A diubah menjadi 3.00 diperoleh solusi $\hat{x} = (01.07968, 4.75637, 0.05856)^T$. Kita menyimpulkan bahwa SPL tersebut berkondisi buruk. Jika kita mengalikan A dengan x dengan x adalah solusi sejati $x = (1,2,-1)^T$, kita peroleh

$$Ax = (-1.61, 7.23, -3.38)^T = b$$

tetapi bila kita hitung dengan solusi hampiran $\hat{x} = (0.880, -2.35, -2.66)^T$ kita peroleh

$$A\hat{x} = (-1.6047, 7.2292, -2.66),$$

yang sangar dekat ke b . Penyulihan kembali solusi ke dalam sistem persamaan ternyata tidak dapat dijadikan petunjuk bahwa sistem berkondisi buruk.

Beberapa ukuran untuk kondisi buruk telah dikemukakan para ahli numerik, antara lain $|\det(A)|$ sangat kecil dibandingkan dengan nilai maksimum $|a_{ij}|$ dan $|b_i|$. Misalnya, SPL dengan dua persamaan dapat ditulis sebagai:

$$a_{11}x_1 + a_{12}x_2 = b_1 \rightarrow x_2 = \frac{a_{11}}{a_{12}}x_1 + \frac{b_1}{a_{12}}$$

$$a_{21}x_1 + a_{22}x_2 = b_2 \rightarrow x_2 = \frac{a_{21}}{a_{22}}x_1 + \frac{b_2}{a_{22}}$$

Bila gradien kedua garis tersebut *hampir sama*, maka:

$$\frac{-a_{11}}{a_{12}} \approx \frac{-a_{21}}{a_{22}}$$

dan apabila kedua ruas dikali silang:

$$a_{11}a_{22} \approx a_{12}a_{21}$$

atau dapat ditulis sebagai

$$a_{11}a_{22} - a_{12}a_{21} \approx 0$$

yang dalam hal ini $a_{11}a_{22} - a_{12}a_{21}$ adalah determinan matriks A pada SPL di atas. Sistem persamaan lanjar berkondisi buruk bila determinan matriks A hampir nol. Jika $\det(A) = 0$, maka gradien kedua garis tersebut sama, yang berarti SPL tidak mempunyai jawab yang unik. Determinan matriks A pada contoh di atas adalah $(1)(2) - 2(1.1) = 2 - 2.2 = -0.2$, yang relatif lebih dekat ke nol.

Contoh 4.15

Tentukan solusi $Ax = b$ berikut

$$A = \begin{bmatrix} 3.02 & -1.05 & 2.53 \\ 4.33 & 0.56 & -1.78 \\ -0.83 & -0.54 & 1.47 \end{bmatrix} \quad b = \begin{bmatrix} -1.61 \\ 7.23 \\ -3.38 \end{bmatrix}$$

Penyelesaian:

Matriks akhir hasil eliminasi Gauss-nya sampai 3 angka bena adalah

$$\left[\begin{array}{ccc|c} 4.33 & 0.56 & -1.78 & 7.23 \\ 0 & -1.44 & 3.77 & -6.65 \\ 0 & 0 & \textbf{-0.00362} & 0.00962 \end{array} \right]$$

sangat kecil, rawan pembulatan

Solusi hampirannya adalah $x = (0.880, -2.53, -2.66)^T$ bandingkan dengan solusi sejatinya, $x = (1, 2, -1)^T$

Perbedaan kedua solusi ini tidak mengherankan apabila kita hitung determinan matriks A ,

$$\det(A) = (4.33)(-1.44)(-0.00362) = -0.0226$$

yang nilainya sangat kecil (mendekati nol), yang berarti sistem berkondisi buruk. Bila koefisien a_{11} diubah dari 3.02 menjadi 3.10 memberikan solusi

$$x = (0.05856, 4.75637, 1.07968)^T$$

Solusi yang lebih baik dapat kita peroleh bila menggunakan bilangan berketelitian yang lebih tinggi, misalnya sampai empat angka bena sebagai berikut: $x_1 = 0.9998$, $x_2 = 1.9995$, $x_3 = -1.002$. ■

Sukar dirinci harus *berapa dekat* determinan ke nol untuk menunjukkan adanya kondisi buruk. Ini diperumit dengan kenyataan bahwa determinan dapat diubah dengan mengalikan satu atau beberapa persamaan dengan suatu faktor skala tanpa mengubah penyelesaiannya. Akibatnya, determinan merupakan nilai yang nisbi yang dipengaruhi oleh besarnya koefisien [CHA91]. Ini diperlihatkan oleh contoh berikut.

Contoh 4.16

Tentukan determinan matriks A pada SPL berikut

$$\begin{aligned}x_1 + 2x_2 &= 10 \\1.1x_1 + 2x_2 &= 10.4\end{aligned}$$

bila

- (i) SPL seperti apa adanya,
- (ii) kedua persamaan dikali 10.

Penyelesaian:

- (i) SPL apa adanya

Determinannya,

$$\det(A) = (1)(2) - (1.1)(2) = -0.2$$

yang dekat ke nol, karena itu SPL berkondisi buruk.

- (ii) Kedua persamaan dikali 10,

$$10x_1 + 20x_2 = 100$$

$$11x_1 + 20x_2 = 104$$

Determinannya,

$$\det(A) = (10)(20) - (11)(20) = -20.$$

yang ternyata menjadi lebih besar. Meskipun determinannya besar, namun SPL tetap berkondisi buruk sebab perkalian dengan skala tidak mempengaruhi penyelesaiannya secara grafis. ■

Contoh 4.16 (ii) di atas memperlihatkan bahwa ukuran determinan sukar dikaitkan dengan kondisi buruk. Kesukaran ini dapat diatasi bila SPL dinormalkan sedemikian sehingga koefisien terbesar pada tiap baris persamaan sama dengan 1.

Contoh 4.17

Normalkan SPL pada Contoh 4.16, lalu hitung determinan matriks A .

Penyelesaian:

SPL dinormalkan dengan cara membagi tiap baris dengan koefisien terbesar pada baris itu sehingga koefisien maksimumnya = 1

$$\begin{aligned}0.5x_1 + x_2 &= 5 \\0.55x_1 + x_2 &= 5.2\end{aligned}$$

Determinannya,

$$\det(A) = (0.5)(1) - (1)(0.55) = -0.55$$

yang dekat ke nol, karena itu berkondisi buruk. ■

Pada sistem yang berkondisi baik, penormalan tetap menghasilkan determinan yang jauh dari nol. Hal ini ditunjukkan pada Contoh 4.17 berikut.

Contoh 4.18

Hitung determinan matriks A pada SPL

$$\begin{aligned}3x_1 + 2x_2 &= 18 \\-x_1 + 2x_2 &= 2\end{aligned}$$

bila (i) SPL apa adanya dan bila (ii) SPL dinormalkan[CHA91]

Penyelesaian:

(i) SPL apa adanya

Determinannya,

$$\det(A) = (3)(2) - 2(-1) = 8$$

yang nilainya jauh dari nol, karena itu berkondisi baik.

- (ii) SPL dinormalkan

Penormalan menghasilkan

$$\begin{aligned}x_1 + 0.667x_2 &= 6 \\-0.5x_1 + x_2 &= 1\end{aligned}$$

Determinannya,

$$\det(A) = (1)(1) - (0.667)(-0.5) = 1.333$$

yang nilainya jauh dari nol, karena itu berkondisi baik. ■

Selain dengan menghitung determinan, ada beberapa ukuran lain yang dapat digunakan untuk memeriksa apakah sistem persamaan lanjar berkondisi buruk [NAK92]:

1. Mencoba mengubah koefisien dengan perubahan yang cukup kecil, lalu membandingkan solusinya dengan solusi sistem persamaan yang belum diubah. Jika perubahan kecil koefisien menghasilkan solusi yang sangat berbeda dengan solusi sebelum perubahan, maka sistem berkondisi buruk.
2. Membandingkan solusi berketelitian tunggal dengan solusi berketelitian ganda. Jika kedua solusinya berbeda berarti sistem berkondisi buruk.
3. Skalakan A sehingga elemen terbesar dalam masing-masing baris adalah 1 dan kemudian hitung A^{-1} . Jika elemen A^{-1} beberapa orde lebih besar daripada elemen matriks yang diskala semula, maka sistem berkondisi buruk
4. Menghitung $\det(A) \times \det(A^{-1})$ apakah berbeda jauh dari 1. Jika ya, berarti sistem berkondisi buruk.
5. Menghitung $(A^{-1})^{-1}$ apakah berbeda “jauh” dari A . Jika ya, berarti sistem berkondisi buruk.
6. Menghitung AA^{-1} apakah berbeda “jauh” dari matriks I . Jika ya, berarti sistem berkondisi buruk.
7. Menghitung $(A^{-1})(A^{-1})^{-1}$ apakah berbeda “jauh” dari matriks I . Jika ya, berarti sistem berkondisi buruk.

Walaupun terdapat beragam cara untuk memeriksa kondisi sistem, akan lebih disukai mendapatkan bilangan tunggal yang dapat berlaku sebagai petunjuk adanya kondisi buruk. Bilangan tersebut dinamakan **bilangan kondisi matriks**.

4.8 Bila ng a n Kondisi Matriks

Bilangan kondisi matriks dinyatakan sebagai :

$$Cond(A) = \|A\| \|A^{-1}\| \quad (\text{P.4.22})$$

yang dalam hal ini $\|A\|$ adalah norma (*norm*) tak-hingga (∞) matriks A , yang didefinisikan sebagai:

$$\|A\| = \|A\|_{\infty} = \max_{1 \leq i \leq n} \sum_{j=1}^n |a_{ij}|$$

Sebagai tambahan, perlu kita ketahui sifat-sifat norma matriks berikut :

- (a) $|A| \geq 0$ dan $|A| = 0$ jika dan hanya jika $A = 0$
- (b) $|kA| = k|A|$
- (c) $|A + B| \leq |A| + |B|$
- (d) $|AB| \leq |A||B|$

Contoh 4.19

Hitung bilangan kondisi matriks A berikut

$$A = \begin{bmatrix} 3.02 & -1.05 & 2.53 \\ 4.33 & 0.56 & -1.78 \\ -0.83 & -0.54 & 1.47 \end{bmatrix}$$

Penyelesaian:

Tentukan terlebih dahulu matriks balikannya,

$$A^{-1} = \begin{bmatrix} 5.661 & -7.273 & -18.55 \\ 200.5 & -268.3 & -669.9 \\ 76.85 & -102.6 & -255.9 \end{bmatrix}$$

maka dapat dihitung

$$\begin{aligned}\|A\| &= |4.33| + |0.56| + |1.78| = 6.07 \\ \|A^{-1}\| &= |200.5| + |-268.3| + |-669.9| = 1138.7\end{aligned}$$

sehingga bilangan kondisi matriks A adalah

$$cond(A) = (6.07)(1138.7) = 7595 \quad \blacksquare$$

Bagaimana kita menggunakan bilangan kondisi ini untuk menentukan apakah sistem berkondisi buruk atau berkondisi baik? Ralston dan Rabinowitz (1978) dan Gerald dan Wheatley (1984), memperkenalkan penggunaan bilangan kondisi matriks untuk menjelaskan kasus sistem berkondisi buruk sebagai berikut. Seperti diketahui bahwa kondisi buruk disebabkan oleh kesalahan dalam pengukuran data model atau karena kesalahan pembulatan. Misalkan bahwa kesalahan dalam pengukuran parameter SPL menyebabkan kesalahan pada koefisien $a_{i,j}$, sehingga SPL dipecahkan sebagai $(A + E)\hat{x} = b$, yang dalam hal ini \hat{x} menyatakan solusi SPL yang mengandung galat. Misalkan $\hat{A} = A + E$ menyatakan koefisien matriks yang mengandung kesalahan. Kita ingin menghitung berapa besar selisih $x - \hat{x}$.

Dengan menggunakan $Ax = b$ dan $\hat{A}\hat{x} = b$, dapat kita tulis :

$$\begin{aligned}x &= A^{-1}b = A^{-1}(Ax) = A^{-1}(A + \hat{A} - A)\hat{x} \\ &= [I + A^{-1}(\hat{A} - A)]\hat{x} \\ &= \hat{x} + A^{-1}(\hat{A} - A)\hat{x}\end{aligned}$$

Karena $\hat{A} - A = E$, maka

$$x - \hat{x} = A^{-1}E\hat{x} \quad (\text{P.4.23})$$

Dengan menggunakan norma, kita peroleh :

$$\|x - \hat{x}\| \leq \|A^{-1}\| \|E\| \|x\| = \|A^{-1}\| \|A\| \frac{\|E\|}{\|A\|} \|\hat{x}\|$$

sehingga

$$\frac{\|x - \hat{x}\|}{\|\hat{x}\|} \leq (\text{bilangan kondisi}) \times \frac{\|E\|}{\|A\|} \quad (\text{P.4.24})$$

Persamaan (P.4.24) ini menyatakan bahwa norma galat relatif solusi SPL dapat sebesar norma galat nisbi koefisien matriks A dikali dengan bilangan kondisi. *Jadi, jika bilangan kondisi matriks A besar, maka galat relatif solusi SPL juga akan besar. Sebaliknya, jika bilangan kondisinya kecil, galat relatif solusi SPL juga kecil.* Misalnya jika koefisien A diketahui teliti sampai t angka bena (yakni, galat pembulatan berorde 10^{-t}) dan bilangan kondisi $A = 10^c$, penyelesaian x akan teliti sampai $t - c$ angka bena (galat pembulatan $\geq 10^{c-t}$). Misalnya, jika koefisien A diketahui hanya 4 angka bena dan bilangan kondisi 1000, vektor x hanya mempunyai ketelitian satu angka signifikan

TEOREMA 4.1. Sistem persamaan lanjar $Ax = b$ yang bilangan kondisinya kecil menyatakan sistem berkondisi baik. Bilangan kondisi besar menandakan bahwa sistem berkondisi buruk. [KRE88].

Sistem pada Contoh 4.19 adalah contoh sistem yang berkondisi buruk, karena bilangan kondisinya besar.

Dalam praktek, A^{-1} tidak diketahui, sehingga untuk menghitung bilangan kondisi matriks A kita harus menaksir $\|A^{-1}\|$. Metode untuk penaksiran ini tidak dijelaskan di sini.

Di dalam banyak literatur disebutkan bahwa matriks Hilbert adalah contoh matriks yang berkondisi buruk. Bentuk umum matriks Hilbert orde n adalah

$$H = \begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{3} & \dots & \frac{1}{n} \\ \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \dots & \frac{1}{n+1} \\ \frac{1}{3} & \frac{1}{4} & \frac{1}{5} & \dots & \frac{1}{n+2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \frac{1}{n} & \frac{1}{n+1} & \frac{1}{n+2} & \dots & \frac{1}{2n+1} \end{bmatrix}$$

Contohnya, untuk $n = 3$ matriks Hilbertnya adalah

$$H = \begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{3} \\ \frac{1}{2} & 1 & \frac{1}{4} \\ \frac{1}{3} & \frac{1}{4} & 1 \end{bmatrix}$$

Norma matriks H adalah

$$\|H\|_{\sim} = |1| + |\frac{1}{2}| + |\frac{1}{3}| = \frac{11}{6}$$

Matriks balikannya adalah,

$$H^{-1} = \begin{bmatrix} 9 & -18 & 10 \\ -36 & 96 & -60 \\ 30 & -90 & 60 \end{bmatrix}$$

Elemen matriks H^{-1} jauh lebih besar daripada matriks H , hal ini menandakan bahwa matriks H berkondisi buruk. Dapat dihitung norma matriks H^{-1}

$$\|H^{-1}\|_{\sim} = |36| + |96| + |60| = 192$$

Sehingga bilangan kondisi matriks H adalah

$$cond(H) = \|H\|_{\sim} \|H^{-1}\|_{\sim} = \frac{11}{6} \times 192 = 352$$

yang nilanya sangat besar, sehingga benarlah bahwa matriks H berkondisi buruk.

Sekarang kita buktikan mengapa penyulihan kembali solusi ke dalam SPL tidak dapat dijadikan petunjuk bahwa sistem berkondisi buruk. Tinjau kembali persamaan residu

$$r = b - A\hat{x}. \quad (\text{P.4.25})$$

Pada sistem yang berkondisi buruk nilai r sangat kecil, sehingga kita dapat terkecoh dengan menganggap sistem berkondisi baik. Contoh-contoh sebelum ini memperlihatkan bahwa r bukanlah ukuran yang bagus untuk galat ($e = x - \hat{x}$) pada sistem yang berkondisi buruk. Bila x adalah solusi eksak maka $r = 0$, atau

$$0 = b - Ax \quad (\text{P.4.26})$$

Kurangi (P.4.25) dengan (P.4.26):

$$\begin{aligned}
 (b - A\hat{x}) - (b - Ax) &= r \Leftrightarrow \\
 -A\hat{x} + Ax &= r \Leftrightarrow \\
 A(x - \hat{x}) &= r \Leftrightarrow \\
 Ae &= r
 \end{aligned} \tag{P.4.27}$$

atau

$$e = A^{-1}r \tag{P.4.28}$$

Pada sistem yang berkondisi buruk, elemen matriks A^{-1} relatif besar dibandingkan elemen-elemen A . Dari (P.4.28) terlihat bahwa bila elemen A^{-1} relatif sangat besar dibandingkan nilai r yang kecil, maka e akan besar. Jadi, residu r yang kecil tidak menjamin solusi yang diperoleh adalah benar. Karena itu digunakan hubungan antara nilai mutlak galat solusi dengan nilai mutlak residu. Dari persamaan (P.4.25) kita peroleh :

$$r = b - A\hat{x} = Ax - A\hat{x} = A(x - \hat{x}) = Ae \tag{P.4.29}$$

Disini,

$$e = A^{-1}r \tag{P.4.30}$$

Dari sifat-sifat norma matriks di atas, maka norma untuk persamaan (P.5.27) , dengan menerapkan sifat (d), dapat kita tulis :

$$\|e\| \leq \|A^{-1}\| \|r\| \tag{P.4.31}$$

Dari $r = Ae$, kita juga punya $\|r\| \leq \|A\| \|e\|$, yang bila digabung dengan persamaan (P.5.32) memberikan

$$\frac{\|r\|}{\|A\|} \leq \|e\| \leq \|A^{-1}\| \|r\| \tag{P.4.32}$$

Dengan menggunakan cara yang sama untuk $Ax = b$ dan $x = A^{-1}b$, kita peroleh

$$\frac{\|b\|}{\|A\|} \leq \|x\| \leq \|A^{-1}\| \|b\| \tag{P.4.33}$$

Dari persamaan (P.4.32) dan (P.4.33) kita dapatkan hubungan yang penting :

$$\frac{1}{\|A\| \|A^{-1}\|} \frac{\|r\|}{\|b\|} \leq \frac{\|e\|}{\|x\|} \leq \|A\| \|A^{-1}\| \frac{\|r\|}{\|b\|} \quad (\text{P.4.34})$$

atau

$$\frac{1}{\text{bil. kondisi}} \frac{\|r\|}{\|b\|} \leq \frac{\|e\|}{\|x\|} \leq (\text{bil. kondisi}) \frac{\|r\|}{\|b\|} \quad (\text{P.4.35})$$

Persamaan (P.4.35) memperlihatkan bahwa galat relatif dalam menghitung solusi x dapat sebesar residu relatif dikali dengan bilangan kondisi. Tentu saja juga akan sekecil residu relatif dibagi dengan bilangan kondisi. Karena itu, jika bilangan kondisi besar, residu r hanya memberikan sedikit informasi tentang ketelitian x . Sebaliknya, jika bilangan kondisi dekat ke 1, residu nisbi memberikan ukuran galat nabi x yang bagus.

Rice pada tahun 1983 menyarankan sebuah cara lain untuk menilai kondisi SPL: jalankan pemecahan SPL yang sama pada dua kompiler yang berbeda (atau pada dua mesin yang berbeda). Karena kode yang dihasilkan kemungkinan besar menerapkan perhitungannya secara berbeda. Kondisi buruk akan jelas terlihat dari eksperimen seperti itu [CHA91].

4.9 Metode Lelaran Untuk Menyelesaikan SPL

Metode eliminasi Gauss melibatkan banyak galat pembulatan. Galat pembulatan yang terjadi pada eliminasi Gauss (maupun eliminasi Gauss-Jordan) dapat menyebabkan solusi yang diperoleh “jauh” dari solusi sebenarnya. Gagasan metoda lelaran pada pencarian akar persamaan nirlanjar dapat juga diterapkan untuk menyelesaikan SPL. Dengan metode lelaran, galat pembulatan dapat diperkecil, karena kita dapat meneruskan lelaran sampai solusinya seteliti mungkin, sesuai dengan batas galat yang kita perbolehkan. Dengan kata lain, besar galat dapat dikendalikan sampai batas yang bisa diterima.

Jika metode eliminasi Gauss dan variasi-variasinya serta metode dekomposisi *LU* dinamakan **metode langsung (direct)** -karena solusi SPL diperoleh tanpa lelaran- maka metode lelaran dinamakan **metode tidak langsung (indirect)** atau **metode iteratif**.

Tinjau kembali sistem persamaan lanjar

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n &= b_1 \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n &= b_2 \\ \vdots &\quad \vdots \\ a_{n1}x_1 + a_{n2}x_2 + \dots + a_{nn}x_n &= b_n \end{aligned}$$

Dengan syarat $a_{kk} \neq 0$, $k = 1, 2, \dots, n$, maka persamaan lelarannya dapat ditulis sebagai

$$\begin{aligned} x_1^{(k+1)} &= \frac{b_1 - a_{12}x_2^{(k)} - \dots - a_{1n}x_n^{(k)}}{a_{11}} \\ x_2^{(k+1)} &= \frac{b_2 - a_{21}x_1^{(k)} - a_{23}x_3^{(k)} - \dots - a_{2n}x_n^{(k)}}{a_{22}} \\ \vdots & \\ x_n^{(k+1)} &= \frac{b_n - a_{n1}x_1^{(k)} - a_{n2}x_2^{(k)} - \dots - a_{nn-1}x_{n-1}^{(k)}}{a_{nn}} \end{aligned} \tag{P.4.36}$$

dengan $k = 0, 1, 2, \dots$

Lelaran dimulai dengan memberikan tebakan awal untuk x ,

$$x_0 = \begin{bmatrix} x_1^{(0)} \\ x_2^{(0)} \\ \vdots \\ x_n^{(0)} \end{bmatrix}$$

Sebagai kondisi berhenti lelarannya, dapat digunakan pendekatan galat relatif

$$\left| \frac{x_i^{(k+1)} - x_i^{(k)}}{x_i^{(k+1)}} \right| < \epsilon \quad \text{untuk semua } i = 1, 2, 3, \dots, n$$

Syarat cukup agar lelarannya konvergen adalah sistem **dominan secara diagonal**:

$$|a_{ii}| > \sum_{j=1, j \neq i}^n |a_{ij}| \quad , i = 1, 2, 3, \dots, n$$

Syarat cukup ini berarti bahwa agar lelarannya konvergen, cukup dipenuhi syarat itu. Jika syarat tersebut dipenuhi, kekonvergenan dijamin. Meskipun sistem tidak dominan secara diagonal, lelarannya masih mungkin konvergen (lihat kembali makna syarat cukup pada upabab 3.3). Kekonvergenan juga ditentukan oleh pemilihan tebakan awal. Tebakan awal yang terlalu jauh dari solusi sejatinya dapat menyebabkan lelaran divergen.

Sebagai contoh, SPL berikut

$$\begin{aligned} 3x_1 + x_2 - x_3 &= 1 \\ 2x_1 + 4x_2 + x_3 &= 5 \\ -x_1 + 5x_2 + 8x_3 &= 5 \end{aligned}$$

dominan secara diagonal, karena

$$\begin{aligned} |3| &> |1| + |-1| \\ |4| &> |2| + |1| \\ |8| &> |-1| + |5| \end{aligned}$$

karena itu lelarannya pasti konvergen.

Ada dua metode lelaran yang akan kita bahas di sini:

1. Metode lelaran Jacobi
2. Metode lelaran Gauss-Seidel

4.9.1 Metode Lelaran Jacobi

Persamaan lelarannya adalah seperti yang ditulis di atas. Misalkan diberikan tebakan awal $x^{(0)}$:

$$x^{(0)} = (x_1^{(0)}, x_2^{(0)}, \dots, x_n^{(0)})^T$$

Prosedur lelaran untuk lelaran pertama, kedua, dan seterusnya adalah sebagai berikut:

Lelaran pertama:

$$x_1^{(1)} = \frac{b_1 - a_{12}x_2^{(0)} - a_{13}x_3^{(0)} - \dots - a_{1n}x_n^{(0)}}{a_{11}}$$

$$x_2^{(1)} = \frac{b_2 - a_{21}x_1^{(0)} - a_{23}x_3^{(0)} - \dots - a_{2n}x_n^{(0)}}{a_{22}}$$

$$\vdots$$

$$x_n^{(1)} = \frac{b_n - a_{n1}x_1^{(0)} - a_{n2}x_2^{(0)} - \dots - a_{nn-1}x_{n-1}^{(0)}}{a_{nn}}$$

Lelaran kedua:

$$x_1^{(2)} = \frac{b_1 - a_{12}x_2^{(1)} - a_{13}x_3^{(1)} - \dots - a_{1n}x_n^{(1)}}{a_{11}}$$

$$x_2^{(2)} = \frac{b_2 - a_{21}x_1^{(1)} - a_{23}x_3^{(1)} - \dots - a_{2n}x_n^{(1)}}{a_{22}}$$

$$\vdots$$

$$x_n^{(2)} = \frac{b_n - a_{n1}x_1^{(1)} - a_{n2}x_2^{(1)} - \dots - a_{nn-1}x_{n-1}^{(1)}}{a_{nn}}$$

Rumus umum :

$$x_i^{(k+1)} = \frac{b_i - \sum_{j=1, j \neq i}^n a_{ij}x_j^{(k)}}{a_{ii}}, k = 0, 1, 2, \dots \quad (\text{P.4.37})$$

4.9.2 Metode Lelaran Gauss-Seidel

Kecepatan konvergen pada lelaran Jacobi dapat dipercepat bila setiap harga x_i yang baru dihasilkan segera dipakai pada persamaan berikutnya untuk menentukan harga x_{i+1} yang lainnya.

Lelaran pertama:

$$x_1^{(1)} = \frac{b_1 - a_{12}x_2^{(0)} - a_{13}x_3^{(0)} - a_{14}x_4^{(0)}}{a_{11}}$$

$$x_2^{(1)} = \frac{b_2 - a_{21}x_1^{(1)} - a_{23}x_3^{(0)} - a_{24}x_4^{(0)}}{a_{22}}$$

$$x_3^{(1)} = \frac{b_3 - a_{31}x_1^{(1)} - a_{32}x_2^{(1)} - a_{34}x_4^{(0)}}{a_{33}}$$

$$x_4^{(1)} = \frac{b_4 - a_{41}x_1^{(1)} - a_{42}x_2^{(1)} - a_{43}x_3^{(1)}}{a_{44}}$$

Lelaran kedua:

$$x_1^{(2)} = \frac{b_1 - a_{12}x_2^{(1)} - a_{13}x_3^{(1)} - a_{14}x_4^{(1)}}{a_{11}}$$

$$x_2^{(2)} = \frac{b_2 - a_{21}x_1^{(2)} - a_{23}x_3^{(1)} - a_{24}x_4^{(1)}}{a_{22}}$$

$$x_3^{(2)} = \frac{b_3 - a_{31}x_1^{(2)} - a_{32}x_2^{(2)} - a_{34}x_4^{(1)}}{a_{33}}$$

$$x_4^{(2)} = \frac{b_4 - a_{41}x_1^{(2)} - a_{42}x_2^{(2)} - a_{43}x_3^{(2)}}{a_{44}}$$

Rumus umum:

$$x_i^{(k+1)} = \frac{b_i - \sum_{j=1}^n a_{ij} x_j^{(k+1)} - \sum_{j=i+1}^n a_{ij} x_j^{(k)}}{a_{ii}}, k = 0, 1, 2, \dots$$

Program 4.5 Metode Lelaran Gauss-Seidel (tanpa penanganan kasus divergen)

```

procedure Gauss_Seidel(A : matriks; b: vektor; n:integer;
                      var x : vektor);
{Menghitung solusi SPL Ax = b dengan metode Gauss-Seidel. Diandaikan
lelaran selalu konvergen
K.Awal : A dan b sudah terdefinisi harganya; x sudah berisi vektor
tebakan awal
K.Akhir: x berisi solusi SPL.
}
const
  epsilon = 0.000001;
var
  i, j : integer;
  konvergen : boolean;
  sigma1, sigma2 : real;
  xlama : vektor;
begin
  repeat
    for i:=1 to n do
      begin
        xlama[i]:=x[i]; {simpan nilai x[i] sebelumnya}
        sigma1:=0;
        for j:=1 to i-1 do
          sigma1:=sigma1 + a[i,j]*x[j];
        {endfor}
        sigma2:=0;
        for j:=i+1 to n do
          sigma2:=sigma2 + a[i,j]*x[j];
        {endfor}
        x[i]:=(b[i] - sigma1 - sigma2)/a[i,i]; {a[i,i] <> 0}
      end;
    {periksa kekonvergenan}
    konvergen:=true; i:=1;
    while (konvergen) and (i<=n) do
      begin
        {bila salah satu dari x[i], i=1, 2, ..., n tidak memenuhi
         ABS(xlama[i] - x[i]) < epsilon
         berarti lelaran belum konvergen}
        if ABS(xlama[i] - x[i]) > epsilon then
          konvergen:=false; {belum konvergen}
        {end if}
        i:=i+1;
      end;
    { konvergen or i > n }
  until konvergen;
end;

```

Contoh 4.20

[MAT92] Tentukan solusi SPL

$$\begin{aligned}4x - y + z &= 7 \\4x - 8y + z &= -21 \\-2x + y + 5z &= 15\end{aligned}$$

dengan nilai awal $P_0 = (x_0, y_0, z_0) = (1, 2, 2)$.
(Solusi sejatinya adalah $(2, 4, 3)$)

Penyelesaian:

- (a) Metode lelaran Jacobi
Persamaan lelarannya:

$$x_{r+1} = \frac{7 + y_r - z_r}{4}$$

$$y_{r+1} = \frac{21 + 4x_r - z_r}{8}$$

$$z_{r+1} = \frac{15 + 2x_r - y_r}{5}$$

Lelarannya:

$$x_1 = \frac{7 + 2 - 2}{4} = 1.75$$

$$y_1 = \frac{21 + 4(1) + 2}{8} = 3.375$$

$$z_1 = \frac{15 + 2(1) - 2}{5} = 3.000$$

$$x_2 = \frac{7 + 3.375 - 3.00}{4} = 1.84375$$

$$y_2 = \frac{21 + 4(3.375) - 3.00}{8} = 3.875$$

$$z_2 = \frac{15 + 2(1.75) - 3.375}{5} = 3.025$$

...

$$x_{19} = 2.00000000$$

$$y_{19} = 4.00000000$$

$$z_{19} = 3.00000000$$

- (b) Metode lelaran Gauss-Seidel
Persamaan lelarannya,

$$x_{r+1} = \frac{7 + y_r - z_r}{4}$$

$$y_{r+1} = \frac{21 + 4x_r - z_r}{8}$$

$$z_{r+1} = \frac{15 + 2x_r - y_r}{5}$$

Lelarannya,

$$x_1 = \frac{7 + 2 - 2}{4} = 1.75$$

$$y_1 = \frac{21 + 4(1.75) + 2}{8} = 3.75$$

$$z_1 = \frac{15 + 2(1.75) - 3.75}{5} = 3.000$$

$$x_2 = \frac{7 + 3.75 - 2.95}{4} = 1.95$$

$$y_2 = \frac{7 + 3.75 - 2.95}{8} = 3.96875$$

$$z_2 = \frac{15 + 2(1.95) - 3.96875}{5} = 2.98625$$

...

$$x_{10} = 2.00000000$$

$$y_{10} = 4.00000000$$

$$z_{10} = 3.00000000$$

Jadi, solusi SPL adalah $x = 2.00000000$, $y = 4.00000000$, $z = 3.00000000$

■

4.10 Contoh Soal Terapan

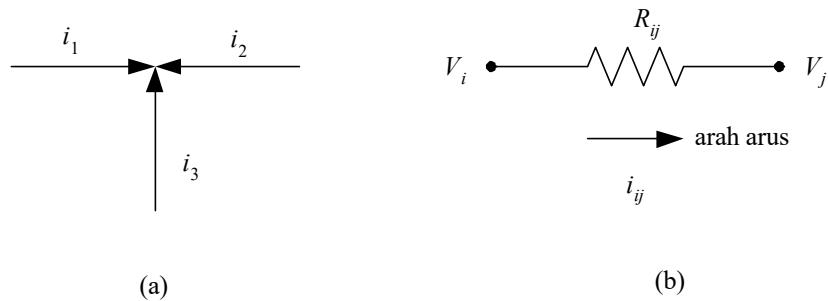
Dalam sebuah rangkaian listrik berlaku hukum-hukum arus Kirchoff menyatakan bahwa jumlah aljabar dari semua arus yang memasuki suatu simpul (Gambar 4.4a) haruslah nol:

$$\sum i = 0 \quad (\text{P.4.38})$$

Dalam hal ini, semua arus i yang memasuki simpul dianggap bertanda positif. Sedangkan hukum Ohm (Gambar 4.4b) menyatakan bahwa arus i yang melalui suatu tahanan adalah :

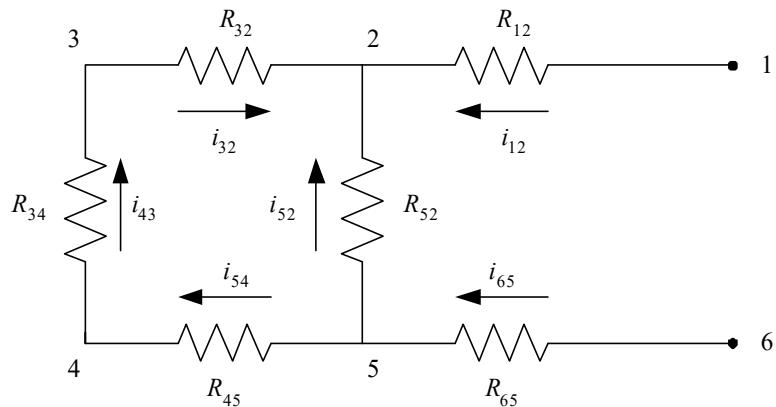
$$i_{ij} = \frac{V_i - V_j}{R_{ij}} \quad (\text{P.4.39})$$

yang dalam hal ini V adalah tegangan dan R adalah tahanan.



Gambar 4.4 (a) Hukum Kirchoff, (b) hukum Ohm

Diberikan sebuah rangkaian listrik dengan 6 buah tahanan seperti pada Gambar 4.5 [CHA91]. Anda diminta menghitung arus pada masing-masing rangkaian.



Gambar 4.5 Rangkaian listrik dengan 6 buah tahanan

Arah arus dimisalkan seperti diatas. Dengan hukum Kirchoff diperoleh persamaan-persamaan berikut :

$$\begin{array}{lll} i_{12} & + i_{52} & + i_{32} = 0 \\ i_{65} & - i_{52} & - i_{54} = 0 \\ i_{43} & - i_{32} & = 0 \\ i_{54} & - i_{43} & = 0 \end{array}$$

Dari hukum Ohm didapat :

$$\begin{array}{lll} i_{32} R_{32} & - V_3 & + V_2 = 0 \\ i_{43} R_{43} & - V_4 & + V_3 = 0 \\ i_{65} R_{65} & & + V_5 = 0 \\ i_{12} R_{12} & & + V_2 = 0 \\ i_{54} R_{54} & - V_5 & + V_4 = 0 \\ i_{52} R_{52} & - V_5 & + V_2 = 0 \end{array}$$

Dengan menyusun kesepuluh persamaan diatas didapatkan SPL sbb :

$$\left[\begin{array}{ccccccccc} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & R_{32} & 0 & 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & R_{43} & 0 & 1 & -1 \\ 0 & 0 & 0 & R_{65} & 0 & 0 & 0 & 0 & 1 \\ R_{12} & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & R_{54} & 0 & 0 & 0 & 1 & -1 \\ 0 & R_{52} & 0 & 0 & 0 & 0 & 1 & 0 & -1 \end{array} \right] \begin{bmatrix} i_{12} \\ i_{52} \\ i_{32} \\ i_{65} \\ i_{54} \\ i_{43} \\ V_2 \\ V_3 \\ V_4 \\ V_5 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ V_6 \\ V_1 \\ 0 \\ 0 \end{bmatrix}$$

Tentukan

$$i_{12}, \quad i_{52}, \quad i_{32}, \quad i_{65}, \quad i_{54}, \quad i_{43}, \quad V_2, \quad V_3, \quad V_4, \quad V_5$$

bila diketahui

$$\begin{aligned} R_{12} &= 5 \text{ ohm}, \quad R_{52} = 10 \text{ ohm}, \quad R_{32} = 10 \text{ ohm} \\ R_{65} &= 20 \text{ ohm}, \quad R_{54} = 15 \text{ ohm}, \quad R_{14} = 5 \text{ ohm.} \\ V_1 &= 200 \text{ volt}, \quad V_6 = 0 \text{ volt.} \end{aligned}$$

Penyelesaian:

Persoalan ini diselesaikan dengan metode eliminasi Gauss. Matriks awal sebelum proses eliminasi Gauss adalah:

$$\left[\begin{array}{cccccccccc|c} 1.000 & 1.000 & 1.000 & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 \\ 0.000 & -1.000 & 0.000 & 1.000 & -1.000 & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 \\ 0.000 & 0.000 & -1.000 & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 \\ 0.000 & 0.000 & 0.000 & 0.000 & 1.000 & -1.000 & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 \\ 0.000 & 0.000 & 10.000 & 0.000 & 0.000 & 0.000 & -1.000 & 1.000 & 0.000 & 0.000 & 0.000 \\ 0.000 & 0.000 & 0.000 & 0.000 & 0.000 & 5.000 & 0.000 & 1.000 & -1.000 & 0.000 & 0.000 \\ 0.000 & 0.000 & 0.000 & 20.000 & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 & 1.000 & 0.000 \\ 5.000 & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 & 1.000 & 0.000 & 0.000 & 0.000 & 200.000 \\ 0.000 & 0.000 & 0.000 & 15.000 & 0.000 & 0.000 & 0.000 & 1.000 & -1.000 & 0.000 & 0.000 \\ 0.000 & 10.000 & 0.000 & 0.000 & 0.000 & 1.000 & 0.000 & 0.000 & 0.000 & -1.000 & 0.000 \end{array} \right]$$

Matriks akhir setelah eliminasi adalah:

$$\left[\begin{array}{cccccccccc|c} 1.000 & 1.000 & 1.000 & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 \\ 0.000 & -1.000 & 0.000 & 1.000 & -1.000 & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 \\ 0.000 & 0.000 & -1.000 & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 \\ 0.000 & 0.000 & 0.000 & 1.000 & -1.000 & 0.000 & 0.100 & 0.000 & 0.000 & -0.100 & 0.000 \\ 0.000 & 0.000 & 0.000 & 0.000 & 1.000 & -1.000 & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 \\ 0.000 & 0.000 & 0.000 & 0.000 & 0.000 & 1.000 & 0.000 & 0.200 & -0.200 & 0.000 & 0.000 \\ 0.000 & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 & -0.100 & -0.200 & 0.200 & 0.150 & 0.000 \\ 0.000 & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 & -0.600 & 0.600 & 0.350 & 40.000 \\ 0.000 & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 & 0.100 & 0.025 & 0.025 & 20.000 \\ 0.000 & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 & -0.200 & -0.200 & -26.667 \end{array} \right]$$

Dengan teknik penyulihan mundur diperoleh solusinya sebagai berikut:

$$\begin{aligned} i_{12} &= 4.444 \text{ ampere}, & i_{52} &= -4.444 \text{ ampere} \\ i_{32} &= 0.000 \text{ ampere}, & i_{65} &= -6.667 \text{ ampere} \\ i_{54} &= -2.222 \text{ ampere}, & i_{43} &= -2.222 \text{ ampere} \\ V_2 &= 177.778 \text{ volt}, & V_3 &= 177.778 \text{ volt} \\ V_4 &= 166.667 \text{ volt}, & V_5 &= 133.333 \text{ volt} \end{aligned}$$

Simplex veri sigillum
Kesederhanaan adalah tanda kebenaran
(Peribahasa Latin)

Kebenaran yang paling agung adalah yang paling sederhana.
Begitu pula orang yang paling agung
(Campbell)

Soal Latihan

1. Pecahkan SPL berikut ini:

$$\begin{aligned}2.51x_1 + 1.48x_2 + 4.53x_3 &= 0.05 \\1.48x_1 + 0.93x_2 - 1.30x_3 &= 1.03 \\2.68x_1 + 3.04x_2 - 1.48x_3 &= -0.53\end{aligned}$$

dengan metode:

- (a) eliminasi Gauss naif (manual, 3 angka bena)
- (b) eliminasi Gauss yang diperbaiki dengan tataancang *pivoting* (manual, 3 angka bena)
- (c) eliminasi Gauss yang diperbaiki dengan tataancang *pivoting* (komputer, jumlah angka bena sesuai dengan komputer yang digunakan).

Sulihkan jawaban masing-masing (a), (b), dan (c) ke dalam SPL, lalu bandingkan hasilnya dengan ruas kanan (vektor b)

2. Diberikan sistem persamaan lanjar $Ax = b$ dengan A dan b sebagai berikut :

$$A = \begin{bmatrix} 1 & 2 & 3 & -1 \\ 2 & 5 & 4 & 8 \\ 4 & 2 & 2 & 1 \\ 6 & 4 & -1 & -2 \end{bmatrix} \quad b = \begin{bmatrix} 10 \\ 8 \\ -2 \\ 4 \end{bmatrix}$$

- (a) Tentukan solusinya dengan metode eliminasi Gauss
- (b) Tentukan determinan matriks A
- (c) Tentukan solusinya dengan metode eliminasi Gauss-Jordan
- (d) Tentukan solusinya dengan metode matriks balikan
- (e) Tentukan solusinya dengan metode dekomposisi LU
- (f) Tentukan solusinya dengan metode lelaran Gauss-Seidell
- (g) Tentukan solusinya dengan metode lelaran Jacobi

Terapkan strategi *pivoting* untuk (a), (b), (c), (d), dan (e).

3. *Pivoting lengkap* jarang diterapkan orang karena kerumitannya. Dari praktik ditemukan bahwa *pivoting lengkap* memberikan hasil yang lebih teliti daripada *pivoting* sebagian meskipun ketelitian ini dibayar dengan waktu komputasi tambahan. Tunjukkan kebenaran pernyataan ini dengan memecahkan SPL berikut :

$$\begin{aligned}0.002110x + 0.08204y &= 0.04313 \\0.3370x + 12.84y &= 6.757\end{aligned}$$

- (a) tanpa *pivoting* (eliminasi Gauss naif)
- (b) dengan *pivoting* sebagian
- (c) dengan *pivoting* lengkap

Semua perhitungan menggunakan empat angka bena (manual).

4. Pecahkan sistem persamaan lanjar $Ax = b$ dengan

$$A = \begin{bmatrix} 8 & 1 & 6 \\ 3 & 5 & 7 \\ 4 & 9 & 2 \end{bmatrix}$$

dan b adalah $(1,0,0)^T$, $(0,1,0)^T$, dan $(0,0,1)^T$. Metode yang digunakan:

- (a) metode eliminasi Gauss yang diperbaiki (sekali jalan).
- (b) metode eliminasi Gauss-Jordan dengan tataancang *pivoting* (sekali jalan)
- (c) metode matriks balikan
- (d) metode dekomposisi LU

Gunakan komputer dan ketelitian hasil semaksimal mungkin (bilangan berketelitian ganda). Hitung juga determinan matriks A .

5. Sekumpulan sistem persamaan linier $Ax = b$ mempunyai matriks A yang sama tetapi vektor b berbeda-beda. Matriks A nya adalah matriks A yang didefinisikan pada soal nomor 2, sedangkan vektor b adalah sbb:

$$b_1 = \begin{bmatrix} 1 \\ 4 \\ -2 \\ 0 \end{bmatrix} \quad b_2 = \begin{bmatrix} -2 \\ 5 \\ 1 \\ 3 \end{bmatrix} \quad b_3 = \begin{bmatrix} 2 \\ -1 \\ 4 \\ 10 \end{bmatrix}$$

- (a) selesaikan dengan metode dekomposisi *LU*
- (b) dengan metode eliminasi Gauss-Jordan, yang dalam hal ini matriks A digabung (*augmented*) dengan semua vektor b .

6. Diberikan SPL $Ax = b$:

$$A = \begin{bmatrix} 2 & 100 & 3000 \\ 50 & 3 & 4 \\ -1 & 2 & 4 \end{bmatrix} \quad b = \begin{bmatrix} 2000 \\ 10 \\ 1 \end{bmatrix}$$

Tentukan solusinya sampai 4 angka bena dengan :

- (a) metode eliminasi Gauss tanpa penskalaan
- (b) metode eliminasi Gauss dengan penskalaan

Dengan penskalaan, bagilah setiap baris i dengan maks $|a_{ij}|$, $j = 1, 2, 3, \dots, n$. Periksa solusi anda dengan menyulihkan kembali kedalam SPL semula.

7. Pada persoalan m persamaan dengan n variabel ($m < n$) , tentukan solusi umum dari $Ax = b$, yang dalam hal ini:

$$A = \begin{bmatrix} 2 & 3 & 1 & 4 & 1 \\ 2 & 3 & 1 & 1 & -1 \\ 4 & 6 & -1 & 1 & 2 \end{bmatrix} \quad \text{dan} \quad b = \begin{bmatrix} 6 \\ 1 \\ 5 \end{bmatrix}$$

8. Pecahkan sistem persamaan lanjar berikut dengan metode eliminasi Gauss :

$$(i) \begin{array}{l} 6.122x + 1500.5y = 1506.622 \\ 2000x + 3y = 2003 \end{array} \quad (ii) \begin{array}{l} 1.001x + 1.5y = 0 \\ 2x + 3y = 1 \end{array}$$

- (a) Tanpa *pivoting* (naif);
- (b) dengan *pivoting*.
- (c) Cek jawaban anda dengan menyulihkan solusi kedalam SPL semula.
Lihat
- (d) bedanya dengan nilai ruas kanan.

Untuk sistem (i) gunakan enam angka bena, dan untuk (ii) gunakan empat angka bena. Ingatlah bahwa setiap komputasi harus dibulatkan ke jumlah angka bena yang diminta (tidak hanya pada hasil akhir saja).

9. Matriks Hilbert adalah contoh klasik matriks yang berkondisi buruk. Misalkan A adalah matriks Hilbert dan diberikan SPL $Ax = b$:

$$\begin{aligned} x_1 + 1/2 x_2 + 1/3 x_3 + 1/4 x_4 &= 1 \\ 1/2 x_1 + 1/3 x_2 + 1/4 x_3 + 1/5 x_4 &= 0 \\ 1/3 x_1 + 1/4 x_2 + 1/5 x_3 + 1/6 x_4 &= 0 \\ 1/4 x_1 + 1/5 x_2 + 1/6 x_3 + 1/7 x_4 &= 0 \end{aligned}$$

Pecahkan $Ax = b$ dengan metode eliminasi Gauss naif dengan ketentuan:

- (a) semua bilangan dalam bentuk pecahan, sehingga tidak ada galat akibat pembulatan. Solusinya eksak, misalkan dilambangkan dengan x . Hitung Ax , dan bandingkan hasilnya dengan b .

- (b) semua bilangan dalam tiga angka bena (manual, tanpa komputer). Solusinya hampiran, misalkan dilambangkan dengan \hat{x} . Hitung $A\hat{x}$, dan bandingkan hasilnya dengan b . Hitung $e = x - \hat{x}$
- (c) semua bilangan berketelitian tinggi (pakai komputer). Solusinya hampiran, misalkan dilambangkan dengan \hat{x} . Hitung $A\hat{x}$, dan bandingkan hasilnya dengan b . Hitung $e = x - \hat{x}$.
10. (a) Dari soal nomor 4 di atas, tentukan determinan matriks A untuk masing-masing ketentuan (a), (b), (c). Apa kesimpulan anda?
- (b) Normalkan matriks A , lalu hitung bilangan kondisi matriks A (gunakan komputer). Apa kesimpulan anda?
11. Pecahkan sistem persamaan lanjar

$$\begin{aligned} 8x_1 + x_2 + 3x_3 + 2x_4 &= 0 \\ 2x_1 + 9x_2 - x_3 - 2x_4 &= 1 \\ x_1 + 3x_2 + 2x_3 - x_4 &= 2 \\ x_1 + 6x_2 + 4x_3 &= 3 \end{aligned}$$

dengan metode:

- (a) dekomposisi LU , yang dalam hal ini L dan U dihitung dengan (i) metode LU Gauss (tidak naif) dan (ii) metode reduksi Crout
- (b) lelaran Jacobi ($\epsilon = 10^{-10}$). Tebakan awal sembarang.
- (c) lelaran Gauss-Seidell ($\epsilon = 10^{-10}$). Tebakan awal sembarang

Gunakan komputer (ketelitian hasil semaksimal mungkin). Untuk (b) dan (c), apakah matriks A dominan secara diagonal?

12. Dapatkah sistem persamaan lanjar berikut :

(a) $5x + 3y = 6$	(b) $5x + 3y = 6$	(c) $2x + y - 5z = 9$
$4x - 2y = 8$	$-6x - 8y = -4$	$x - 5y - z = 14$
		$7x - y - 3z = 26$

diselesaikan dengan metode iterasi Jacobi dan iterasi Gauss-Seidell ? Mengapa ?

13. Matriks Hilbert adalah contoh klasik matriks berkondisi buruk. Diberikan matriks Hilbert berukuran 4×4 :

$$H = \begin{bmatrix} 1 & 1/2 & 1/3 & 1/4 \\ 1/2 & 1/3 & 1/4 & 1/5 \\ 1/3 & 1/4 & 1/5 & 1/6 \\ 1/4 & 1/5 & 1/6 & 1/7 \end{bmatrix}$$

Periksa kondisinya dengan :

- (a) Hitung HH^{-1} apakah berbeda dari matriks identitas
 - (b) Hitung $(H^T)^{-1}$ apakah berbeda dari matriks H
 - (c) Hitung $H^T(H^T)^{-1}$ apakah berbeda dari matriks identitas I dan apakah berbeda dari jawaban (a)
 - (d) Hitung bilangan kondisinya apakah sangat besar dibandingkan dengan (Normalkan terlebih dahulu matriks H)
14. Seperti nomor 13, tetapi matriksnya adalah matriks A pada soal nomor 1.

Bab 5

Interpolasi dan Regresi

Jangan ikuti kemana jalan menuju,¹
tetapi buatlah jalan sendiri dan tinggalkan jejak¹
(Anonim)

Para rekayawan dan ahli ilmu alam sering bekerja dengan sejumlah data diskrit (yang umumnya disajikan dalam bentuk tabel). Data di dalam tabel mungkin diperoleh dari hasil pengamatan di lapangan, hasil pengukuran di laboratorium, atau tabel yang diambil dari buku-buku acuan.

Sebagai ilustrasi, sebuah pengukuran fisika telah dilakukan untuk menentukan hubungan antara tegangan yang diberikan kepada baja tahan-karat dan waktu yang diperlukan hingga baja tersebut patah. Delapan nilai tegangan yang berbeda dicobakan, dan data yang dihasilkan adalah [CHA91]:

Tegangan yang diterapkan, x , kg/mm ²	5	10	15	20	25	30	35	40
Waktu patah, y , jam	40	30	25	40	18	20	22	15

Masalah yang cukup sering muncul dengan data tabel adalah menentukan nilai di antara titik-titik diskrit tersebut (tanpa harus melakukan pengukuran lagi). Misalnya dari tabel pengukuran di atas, rekayawan ingin mengetahui waktu patah y jika tegangan x yang diberikan kepada baja adalah 12 kg/mm². Masalah ini tidak bisa langsung dijawab karena fungsi yang menghubungkan peubah y dengan peubah x tidak diketahui. Salah satu solusinya adalah mencari fungsi yang mencocokkan (*fit*) titik-titik data di dalam tabel tabel. Pendekatan seperti ini di dalam metode numerik dinamakan **pencocokan kurva** (*curve fitting*). Fungsi yang diperoleh dengan pendekatan ini merupakan fungsi hampiran, karena itu nilai fungsinya tidak setepat nilai sejatinya. Namun, cara ini dalam praktek

¹ Terjemahan bebas dari kalimat: "Do not follow where the path may lead. Go, instead, where there is no path and leave a trail"

rekayasa sudah mencukupi karena rumus yang benar-benar menghubungkan dua buah besaran fisik sulit ditemukan.

Pencocokan kurva tidak hanya bertujuan menghitung nilai fungsi, tetapi ia juga digunakan untuk mempermudah perhitungan numerik yang lain seperti menghitung nilai turunan (*derivative*) dan menghitung nilai integral (\int). Misalnya kita dihadapkan dengan fungsi yang bentuknya cukup rumit, seperti fungsi berikut:

$$f(x) = \frac{\ln(2x^{1/2} - 4x^2)^3}{\sqrt{1+2x^5}} \quad (\text{P.5.1})$$

Menghitung turunan fungsi tersebut pada nilai x tertentu, misalnya di $x = a$,

$$f'(a) = ?$$

merupakan pekerjaan yang cukup sulit, apalagi bila turunan yang dibutuhkan semakin tinggi ordennya. Demikian juga dengan menghitung nilai integral fungsi $f(x)$ pada selang integrasi $[a, b]$, misalnya selang $[0, 1]$,

$$\int_0^1 \frac{\ln(2x^{1/2} - 4x^2)^3}{\sqrt{1+2x^5}}$$

merupakan pekerjaan yang tidak mudah, bahkan secara analitik pun belum tentu dapat dilakukan, karena rumus integrasi untuk fungsi semacam ini tidak tersedia. Satu pendekatan untuk melakukan dua perhitungan ini ialah dengan menyederhanakan fungsi $f(x)$ menjadi polinom $p_n(x)$ yang berderajat $\leq n$,

$$f(x) \approx p_n(x)$$

yang dalam hal ini,

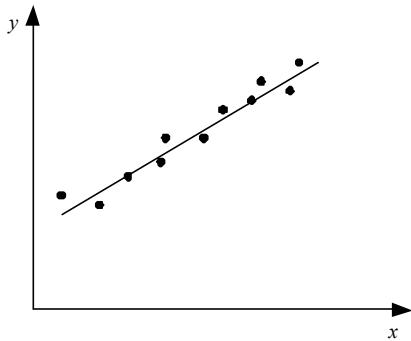
$$p_n(x) = a_0 + a_1x + a_2x^2 + \dots + a_nx^n \quad (\text{P.5.2})$$

Menghitung turunan atau mengintegral suku-suku polinom menjadi lebih mudah karena rumus untuk menghitung turunan atau mengintegrasikan polinom sangat sederhana, yaitu

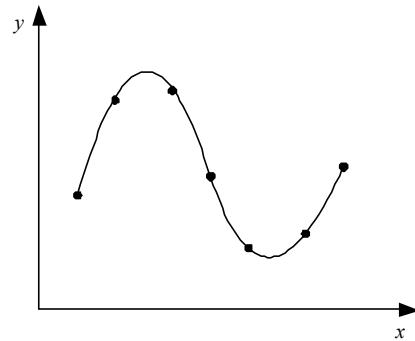
$$(i) \text{ jika } f(x) = ax^n \text{ maka } f'(x) = nax^{n-1}$$

$$(ii) \int ax^n dx = \frac{a}{(n+1)} x^{n+1} + C$$

Untuk membentuk polinom ini, kita mengambil beberapa titik diskrit (yang umumnya berjarak sama) dari fungsi f . Titik-titik tersebut secara alami direpresentasikan dalam bentuk tabel. Selanjutnya titik-titik data ini dicocokkan untuk menentukan polinom $p_n(x)$ yang menghampiri fungsi aslinya.



(a) Regresi



(b) Interpolasi

Gambar 5.1 Pencocokan kurva dengan metode (a) regresi, dan (b) interpolasi

Pencocokan kurva adalah sebuah metode yang memcocokkan titik data dengan sebuah kurva (*curve fitting*) fungsi. Pencocokan kurva dibedakan atas dua metode:

1. *Regresi*.

Data hasil pengukuran umumnya mengandung derau (*noise*) atau galat yang cukup berarti. Karena data ini tidak teliti, maka kurva yang mencocokkan titik data itu tidak perlu melalui semua titik. Tata-ancang yang dipakai adalah menentukan kurva yang mewakili kecenderungan (*trend*) titik data, yakni kurva mengikuti pola titik sebagai suatu kelompok (Gambar 5.1.a). Kurva tersebut dibuat sedemikian sehingga selisih antara titik data dengan titik hampirannya di kurva sekecil mungkin. Metode pencocokan kurva seperti ini dinamakan **regresi kuadrat terkecil** (*least square regression*). Derau pada data mungkin disebabkan oleh kesalahan mengukur, ketidaktelitian pada alat ukur, atau karena kelakuan sistem yang diukur. Contoh data yang mengandung derau adalah tabel tegangan baja di atas.

2. *Interpolasi*

Bila data diketahui mempunyai ketelitian yang sangat tinggi, maka kurva cocokannya dibuat melalui setiap titik, persis sama kalau kurva fungsi yang sebenarnya dirajah melalui tiap titik itu. Kita katakan di sini bahwa kita

menginterpolasi titik-titik data dengan sebuah fungsi (Gambar 5.1.b). Bila fungsi cocokan yang digunakan berbentuk polinom, polinom tersebut dinamakan **polinom interpolasi**. Pekerjaan menginterpolasi titik data dengan sebuah polinom disebut **interpolasi (dengan) polinom**. Contoh data yang berketerkaitan tinggi adalah titik-titik yang dihitung dari fungsi yang telah diketahui (seperti dari persamaan P.5.1), atau data tabel yang terdapat di dalam acuan ilmiah (seperti data percepatan gravitasi bumi sebagai fungsi jarak sebuah titik ke pusat bumi). Selain dengan polinom, interpolasi titik-titik data dapat dilakukan dengan fungsi *spline*, fungsi rasional (pecahan), atau deret *Fourier* [NAK93].

Bab ini dimulai dengan bagian pertama yaitu pencocokan kurva dengan metode interpolasi. Bagian kedua, metode regresi, akan diberikan sebagai akhir bab ini.

Interpolasi memainkan peranan yang sangat penting dalam metode numerik. Fungsi yang tampak rumit menjadi lebih sederhana bila dinyatakan dalam polinom interpolasi. Sebagian besar metode integrasi numerik, metode persamaan diferensial biasa, dan metode turunan numerik didasarkan pada polinom interpolasi. Tidak salah kalau banyak buku acuan menyatakan bahwa interpolasi merupakan pokok bahasan yang fundamental dalam metode numerik.

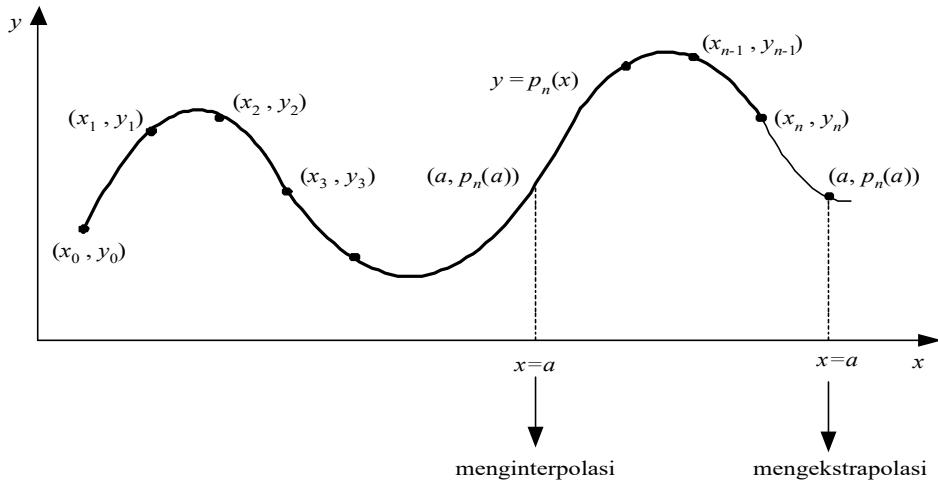
Bagian I: Interpolasi

5.1 Persoalan Interpolasi Polinom

Diberikan $n+1$ buah titik berbeda, $(x_0, y_0), (x_1, y_1), \dots, (x_n, y_n)$. Tentukan polinom $p_n(x)$ yang menginterpolasi (melewati) semua titik-titik tersebut sedemikian rupa sehingga

$$y_i = p_n(x_i) \quad \text{untuk } i = 0, 1, 2, \dots, n$$

Nilai y_i dapat berasal dari fungsi matematika $f(x)$ (seperti $\ln x$, $\sin x$, fungsi Bessel, persamaan P.6.1, dan sebagainya) sedemikian sehingga $y_i = f(x_i)$, sedangkan $p_n(x)$ disebut fungsi hampiran terhadap $f(x)$. Atau, y_i berasal dari nilai empiris yang diperoleh melalui percobaan atau pengamatan.



Gambar 5.2 Interpolasi dan ekstrapolasi

Setelah polinom interpolasi $p_n(x)$ ditemukan, $p_n(x)$ dapat digunakan untuk menghitung perkiraan nilai y di $x = a$, yaitu $y = p_n(a)$. Bergantung pada letaknya, nilai $x = a$ mungkin terletak di dalam rentang titik-titik data ($x_0 < a < x_n$) atau di luar rentang titik-titik data ($a < x_0$ atau $a > x_n$):

- (i) jika $x_0 < a < x_n$ maka $y_k = p(x_k)$ disebut **nilai interpolasi** (*interpolated value*)
- (ii) jika $x_0 < x_k$ atau $x_0 > x_n$ maka $y_k = p(x_k)$ disebut **nilai ekstrapolasi** (*extrapolated value*).

Keduanya, (i) dan (ii), ditunjukkan pada Gambar 5.2.

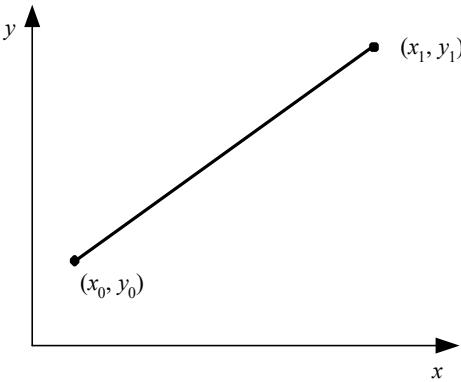
Kita dapat menginterpolasi titik data dengan polinom lanjar, polinom kuadratik, polinom kubik, atau polinom dari derajat yang lebih tinggi, bergantung pada jumlah titik data yang tersedia.

5.1.1 Interpolasi Lanjar

Interpolasi lanjar adalah interpolasi dua buah titik dengan sebuah garis lurus. Misal diberikan dua buah titik, (x_0, y_0) dan (x_1, y_1) . Polinom yang menginterpolasi kedua titik itu adalah persamaan garis lurus yang berbentuk:

$$p_1(x) = a_0 + a_1x \quad (\text{P.5.3})$$

Gambar 5.3 memperlihatkan garis lurus yang menginterpolasi titik-titik (x_0, y_0) dan (x_1, y_1) .



Gambar 5.3 Interpolasi lanjar

Koefisien a_0 dan a_1 dicari dengan proses penyulihan dan eliminasi. Dengan menyulihkan (x_0, y_0) dan (x_1, y_1) ke dalam persamaan (P.5.3), diperoleh dua buah persamaan lanjar:

$$\begin{aligned} y_0 &= a_0 + a_1 x_0 \\ y_1 &= a_0 + a_1 x_1 \end{aligned}$$

Kedua persamaan ini diselesaikan dengan proses eliminasi, yang memberikan

$$a_1 = \frac{y_1 - y_0}{x_1 - x_0} \quad (\text{P.5.4})$$

dan

$$a_0 = \frac{x_1 y_0 - x_0 y_1}{x_1 - x_0} \quad (\text{P.5.5})$$

Sulihkankan (P.5.4) dan (P.5.5) ke dalam (P.5.3) untuk mendapatkan persamaan garis lurus:

$$p_1(x) = \frac{x_1 y_0 - x_0 y_1}{x_1 - x_0} + \frac{(y_1 - y_0)x}{(x_1 - x_0)} \quad (\text{P.5.6})$$

Dengan melakukan sedikit manipulasi aljabar, persamaan (P.5.6) ini dapat disusun menjadi

$$p_1(x) = y_0 + \frac{(y_1 - y_0)}{(x_1 - x_0)}(x - x_0) \quad (\text{P.5.7})$$

Bukti:

$$\begin{aligned} p_1(x) &= \frac{x_1 y_0 - x_0 y_1}{x_1 - x_0} + \frac{(y_1 - y_0)x}{(x_1 - x_0)} \\ \Leftrightarrow p_1(x) &= \frac{x_1 y_0 - x_0 y_1 + x y_1 - x y_0}{x_1 - x_0} \\ \Leftrightarrow p_1(x) &= \frac{x_1 y_0 - x_0 y_1 + x y_1 - x y_0 + x_0 y_0 - x_0 y_1}{x_1 - x_0} \\ \Leftrightarrow p_1(x) &= \frac{(x_1 - x_0)y_0 + (y_1 - y_0)(x - x_0)}{x_1 - x_0} \\ \Leftrightarrow p_1(x) &= y_0 + \frac{(y_1 - y_0)}{(x_1 - x_0)}(x - x_0) \quad \blacksquare \end{aligned}$$

Persamaan (P.5.7) adalah persamaan garis lurus yang melalui dua buah titik, (x_0, y_0) dan (x_1, y_1) . Kurva polinom $p_1(x)$ ini adalah berupa garis lurus (Gambar 5.3).

Contoh 5.1

Perkirakan jumlah penduduk Amerika Serikat pada tahun 1968 berdasarkan data tabulasi berikut [KRE88]:

Tahun	1960	1970
Jumlah penduduk (juta)	179.3	203.2

Penyelesaian:

Dengan menggunakan persamaan (P.5.7), diperoleh

$$p_1(1968) = \frac{179.3 + 203.2 - 179.3(1968 - 1960)}{1970 - 1960} = 198.4$$

Jadi, taksiran jumlah penduduk AS pada tahun 1968 adalah 198.4 juta. ■

Contoh 5.2

Dari data $\ln(9.0) = 2.1972$, $\ln(9.5) = 2.2513$, tentukan $\ln(9.2)$ dengan interpolasi lanjar [KRE88] sampai 5 angka bena. Bandingkan dengan nilai sejati $\ln(9.2) = 2.2192$.

Penyelesaian:

Dengan menggunakan persamaan (P.5.7), diperoleh

$$p_1(9.2) = \frac{2.1972 + 2.1513 - 2.1972(9.2 - 9.0)}{9.5 - 9.0} = 2.2192$$

Galat = $2.2192 - 2.2188 = 0.0004$. Di sini interpolasi lanjar tidak cukup untuk memperoleh ketelitian sampai 5 angka bena. Ia hanya benar sampai 3 angka bena. ■

5.1.2 Interpolasi Kuadratik

Misal diberikan tiga buah titik data, (x_0, y_0) , (x_1, y_1) , dan (x_2, y_2) . Polinom yang menginterpolasi ketiga buah titik itu adalah polinom kuadrat yang berbentuk:

$$p_2(x) = a_0 + a_1x + a_2x^2 \quad (\text{P.5.8})$$

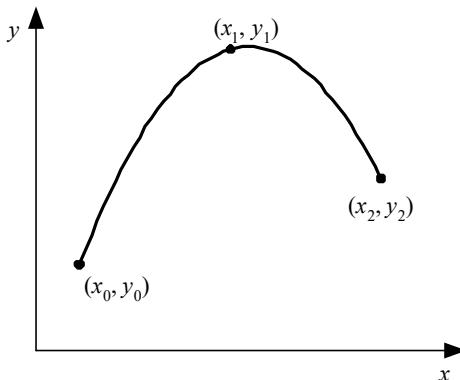
Bila digambar, kurva polinom kuadrat berbentuk parabola (Gambar 5.4).

Polinom $p_2(x)$ ditentukan dengan cara berikut:

- sulihkan (x_i, y_i) ke dalam persamaan (P.5.8), $i = 0, 1, 2$. Dari sini diperoleh tiga buah persamaan dengan tiga buah parameter yang tidak diketahui, yaitu a_0 , a_1 , dan a_2 :

$$\begin{aligned} a_0 + a_1x_0 + a_2x_0^2 &= y_0 \\ a_0 + a_1x_1 + a_2x_1^2 &= y_1 \\ a_0 + a_1x_2 + a_2x_2^2 &= y_2 \end{aligned}$$

- hitung a_0 , a_1 , a_2 dari sistem persamaan tersebut dengan metode eliminasi Gauss.



Gambar 5.4 Interpolasi kuadratik

Contoh 5.3

Diberikan titik $\ln(8.0) = 2.0794$, $\ln(9.0) = 2.1972$, dan $\ln(9.5) = 2.2513$. Tentukan nilai $\ln(9.2)$ dengan interpolasi kuadratik.

Penyelesaian:

Sistem persamaan lanjar yang terbentuk adalah

$$\begin{aligned} a_0 + 8.0a_1 + 64.00a_2 &= 2.0794 \\ a_0 + 9.0a_1 + 81.00a_2 &= 2.1972 \\ a_0 + 9.5a_1 + 90.25a_2 &= 2.2513 \end{aligned}$$

Penyelesaian sistem persamaan dengan metode eliminasi Gauss menghasilkan $a_0 = 0.6762$, $a_1 = 0.2266$, dan $a_2 = -0.0064$. Polinom kuadratiknya adalah

$$p_2(x) = 0.6762 + 0.2266x - 0.0064x^2$$

sehingga

$$p_2(9.2) = 2.2192$$

yang sama dengan nilai sejatinya (5 angka bera).

■

5.1.3 Interpolasi Kubik

Misal diberikan empat buah titik data, (x_0, y_0) , (x_1, y_1) , (x_2, y_2) , dan (x_3, y_3) . Polinom yang menginterpolasi keempat buah titik itu adalah polinom kubik yang berbentuk:

$$p_3(x) = a_0 + a_1x + a_2x^2 + a_3x^3 \quad (\text{P.5.9})$$

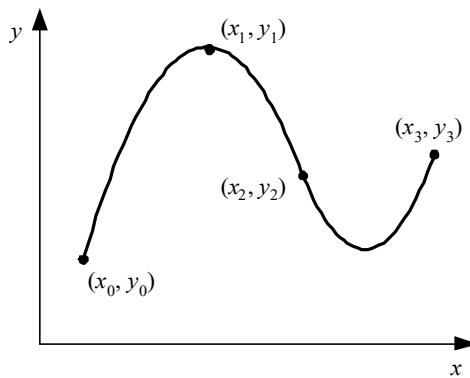
Polinom $p_3(x)$ ditentukan dengan cara berikut:

- sulihkan (x_i, y_i) ke dalam persamaan (P.5.9), $i = 0, 1, 2, 3$. Dari sini diperoleh empat buah persamaan dengan empat buah parameter yang tidak diketahui, yaitu a_0, a_1, a_2 , dan a_3 :

$$\begin{aligned} a_0 + a_1 x_0 + a_2 x_0^2 + a_3 x_0^3 &= y_0 \\ a_0 + a_1 x_1 + a_2 x_1^2 + a_3 x_1^3 &= y_1 \\ a_0 + a_1 x_2 + a_2 x_2^2 + a_3 x_2^3 &= y_2 \\ a_0 + a_1 x_3 + a_2 x_3^2 + a_3 x_3^3 &= y_3 \end{aligned}$$

- hitung a_0, a_1, a_2 , dan a_3 dari sistem persamaan tersebut dengan metode eliminasi Gauss.

Bila digambar, kurva polinom kubik adalah seperti Gambar 5.5.



Gambar 5.5 Interpolasi kubik

Dengan cara yang sama kita dapat membuat polinom interpolasi berderajat n untuk n yang lebih tinggi:

$$p_n(x) = a_0 + a_1 x + a_2 x^2 + \dots + a_n x^n$$

asalkan tersedia $(n+1)$ buah titik data. Dengan menyulihkan (x_i, y_i) ke dalam persamaan polinom di atas $y = p_n(x)$ untuk $i = 0, 1, 2, \dots, n$, akan diperoleh n buah sistem persamaan lanjar dalam $a_0, a_1, a_2, \dots, a_n$,

$$\begin{aligned} a_0 + a_1 x_0 + a_2 x_0^2 + \dots + a_n x_0^n &= y_0 \\ a_0 + a_1 x_1 + a_2 x_1^2 + \dots + a_n x_1^n &= y_1 \\ a_0 + a_1 x_2 + a_2 x_2^2 + \dots + a_n x_2^n &= y_2 \\ \dots &\dots \\ a_0 + a_1 x_n + a_2 x_n^2 + \dots + a_n x_n^n &= y_n \end{aligned}$$

Solusi sistem persamaan lanjar ini diperoleh dengan menggunakan metode eliminasi Gauss yang sudah anda pelajari.

Secara umum, penentuan polinom interpolasi dengan cara yang diuraikan di atas kurang disukai, karena sistem persamaan lanjar yang diperoleh ada kemungkinan berkondisi buruk, terutama untuk derajat polinom yang semakin tinggi.

Beberapa metode perhitungan polinom interpolasi telah ditemukan oleh para numerikawan tanpa menggunakan cara pendekatan di atas. Beberapa diantaranya akan diberikan di sini, yaitu:

1. Polinom Lagrange
2. Polinom Newton
3. Polinom Newton-Gregory (kasus khusus dari polinom Newton)

Untuk sejumlah titik data yang diberikan, metode interpolasi yang berbeda-beda ini tetap menghasilkan polinom yang sama (unik), tetapi dalam bentuk yang berbeda satu sama lain, dan berbeda juga dalam jumlah komputasi yang dilibatkan. Keunikan polinom interpolasi ini akan dibuktikan setelah kita sampai pada polinom Newton.

5.2 Polinom Lagrange

Tinjau kembali polinom lanjar pada persamaan (P.5.7):

$$p_1(x) = y_0 + \frac{(y_1 - y_0)}{(x_1 - x_0)} (x - x_0)$$

Persamaan ini dapat diatur kembali sedemikian rupa sehingga menjadi

$$p_1(x) = y_0 \frac{(x - x_1)}{(x_0 - x_1)} + y_1 \frac{(x - x_0)}{(x_1 - x_0)} \quad (\text{P.5.10})$$

atau dapat dinyatakan dalam bentuk

$$p_1(x) = a_0 L_0(x) + a_1 L_1(x) \quad (\text{P.5.11})$$

yang dalam hal ini

$$a_0 = y_0, \quad L_0(x) = \frac{(x - x_1)}{(x_0 - x_1)}$$

dan

$$a_1 = y_1 \quad , \quad L_1(x) = \frac{(x - x_0)}{(x_1 - x_0)}$$

Persamaan (P.5.11) dinamakan **polinom Lagrange** derajat 1. Nama polinom ini diambil dari nama penemunya, yaitu Joseph Louis Lagrange yang berkebangsaan Perancis.

Bentuk umum polinom Lagrange derajat $\leq n$ untuk $(n + 1)$ titik berbeda adalah

$$p_n(x) = \sum_{i=0}^n a_i L_i(x) = a_0 L_0(x) + a_1 L_1(x) + \dots + a_n L_n(x) \quad (\text{P.5.12})$$

yang dalam hal ini

$$a_i = y_i \quad , \quad i = 0, 1, 2, \dots, n$$

dan,

$$L_i(x) = \prod_{\substack{j=0 \\ j \neq i}}^n \frac{(x - x_j)}{(x_i - x_j)} = \frac{(x - x_0)(x - x_1) \dots (x - x_{i-1})(x - x_{i+1}) \dots (x - x_n)}{(x_i - x_0)(x_i - x_1) \dots (x_i - x_{i-1})(x_i - x_{i+1}) \dots (x_i - x_n)}$$

Mudah dibuktikan, bahwa :

$$L_i(x_j) = \begin{cases} 1 & , i = j \\ 0 & , i \neq j \end{cases}$$

dan polinom interpolasi $p_n(x)$ melalui setiap titik data.

Bukti:

Jika $i = j$, maka

$$\begin{aligned} L_i(x_i) &= \prod_{\substack{j=0 \\ j \neq i}}^n \frac{(x_i - x_j)}{(x_i - x_j)} = \frac{(x_i - x_0)(x_i - x_1) \dots (x_i - x_{i-1})(x_i - x_{i+1}) \dots (x_i - x_n)}{(x_i - x_0)(x_i - x_1) \dots (x_i - x_{i-1})(x_i - x_{i+1}) \dots (x_i - x_n)} \\ &= 1 \quad (\text{karena penyebut = pembilang}) \end{aligned}$$

Jika $i \neq j$, maka

$$\begin{aligned}
L_i(x_j) &= \prod_{\substack{j=0 \\ j \neq i}}^n \frac{(x_j - x_i)}{(x_i - x_j)} \\
&= \frac{(x_j - x_0)(x_j - x_1) \dots (x_j - x_{i-1})(x_j - x_{i+1}) \dots (x_j - x_n)}{(x_i - x_0)(x_i - x_1) \dots (x_i - x_j)(x_i - x_{i-1})(x_i - x_{i+1}) \dots (x_i - x_n)} \\
&= \frac{0}{(x_i - x_0)(x_i - x_1) \dots (x_i - x_j)(x_i - x_{i-1})(x_i - x_{i+1}) \dots (x_i - x_n)} \\
&= 0 \quad (\text{karena pembilang} = 0, \text{ yaitu } (x_j - x_i) = 0)
\end{aligned}$$

Akibatnya,

$$\begin{aligned}
p_n(x_0) &= L_0(x_0) y_0 + L_1(x_0) y_1 + L_2(x_0) y_2 + \dots + L_n(x_0) y_n \\
&= 1 \cdot y_0 + 0 \cdot y_1 + 0 \cdot y_2 + \dots + 0 \cdot y_n \\
&= y_0 \\
p_n(x_1) &= y_1 \\
&\dots \\
p_n(x_n) &= y_n
\end{aligned}$$

Dengan demikian,

$$p_n(x_i) = y_i, \quad i = 0, 1, 2, \dots, n$$

atau dengan kata lain, polinom interpolasi $p_n(x)$ melalui setiap titik data. ■

Contoh 5.4

[MAT92] Hampiri fungsi $f(x) = \cos x$ dengan polinom interpolasi derajat tiga di dalam selang $[0.0, 1.2]$. Gunakan empat titik, $x_0 = 0.0$, $x_1 = 0.4$, $x_2 = 0.8$, dan $x_3 = 1.2$. Perkirakan nilai $p_3(0.5)$, dan bandingkan dengan nilai sejatinya.

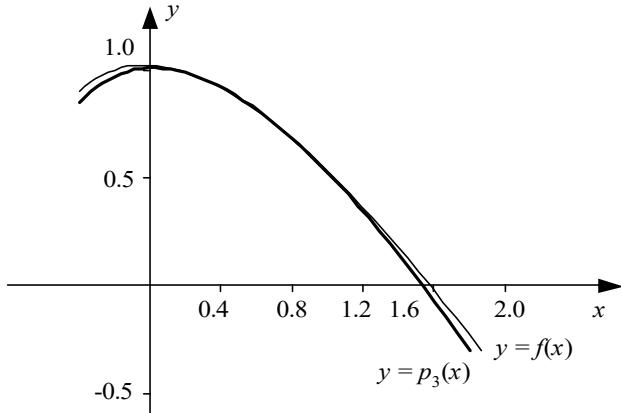
Penyelesaian:

x_i	0.0	0.4	0.8	1.2
y_i	1.000000	0.921061	0.696707	0.362358

Polinom Lagrange derajat 3 yang menginterpolasi keempat titik di tabel adalah

$$\begin{aligned}
 p_3(x) &= a_0 L_0(x) + a_1 L_1(x) + a_2 L_2(x) + a_3 L_3(x) \\
 &= y_0 \frac{(x - x_1)(x - x_2)(x - x_3)}{(x_0 - x_1)(x_0 - x_2)(x_0 - x_3)} + y_1 \frac{(x - x_0)(x - x_2)(x - x_3)}{(x_1 - x_0)(x_1 - x_2)(x_1 - x_3)} + \\
 &\quad y_2 \frac{(x - x_0)(x - x_1)(x - x_3)}{(x_2 - x_0)(x_2 - x_1)(x_2 - x_3)} + y_3 \frac{(x - x_0)(x - x_1)(x - x_2)}{(x_3 - x_0)(x_3 - x_1)(x_3 - x_2)} \\
 &= 1.000000 \frac{(x - 0.4)(x - 0.8)(x - 1.2)}{(0.0 - 0.4)(0.0 - 0.8)(0.0 - 1.2)} + \\
 &\quad 0.921061 \frac{(x - 0.0)(x - 0.8)(x - 1.2)}{(0.4 - 0.0)(0.4 - 0.8)(0.4 - 1.2)} + \\
 &\quad 0.696707 \frac{(x - 0.0)(x - 0.4)(x - 1.2)}{(0.8 - 0.0)(0.8 - 0.4)(0.8 - 1.2)} + \\
 &\quad 0.362358 \frac{(x - 0.0)(x - 0.4)(x - 0.8)}{(1.2 - 0.0)(1.2 - 0.4)(1.2 - 0.8)} \\
 &= \frac{-2.604167(x - 0.4)(x - 0.8)(x - 1.2) + 7.195789(x - 0.0)(x - 0.8)(x - 1.2)}{-5.443021(x - 0.0)(x - 0.4)(x - 1.2) + 0.943640(x - 0.0)(x - 0.4)(x - 0.8)}
 \end{aligned}$$

Untuk mengurangi galat akibat pembulatan, polinom $p_3(x)$ ini tidak perlu disederhanakan lebih jauh. Kurva $y = \cos(x)$ dan $y = p_3(x)$ diperlihatkan pada Gambar 5.6.



Gambar 5.6 Grafik fungsi $y = \cos(x)$ dan $y = p_3(x)$

Dengan menggunakan polinom interpolasi $p_3(x)$ itu kita dapat menaksir nilai fungsi di $x = 0.5$ sebagai berikut:

$$\begin{aligned}
 p_3(0.5) &= -2.604167(0.5 - 0.4)(0.5 - 0.8)(0.5 - 1.2) \\
 &\quad + 7.195789(0.5 - 0.0)(0.5 - 0.8)(0.5 - 1.2) \\
 &\quad - 5.443021(0.5 - 0.0)(0.5 - 0.4)(0.5 - 1.2) \\
 &\quad + 0.943640(0.5 - 0.0)(0.5 - 0.4)(0.5 - 0.8) \\
 &= 0.877221
 \end{aligned}$$

Sebagai perbandingan, nilai sejatinya adalah

$$y = \cos(0.5) = 0.877583 \quad \blacksquare$$

Catatlah bahwa polinom Lagrange tidak hanya berlaku untuk titik-titik yang berjarak sama. Kita juga dapat membentuk polinom Lagrange untuk titik-titik data yang tidak berjarak sama. Perhatikan contoh 5.5 berikut.

Contoh 5.5

Dari fungsi $y = f(x)$, diberikan tiga buah titik data dalam bentuk tabel:

x	1	4	6
y	1.5709	1.5727	1.5751

Tentukan $f(3.5)$ dengan polinom Lagrange derajat 2. Gunakan lima angka bera.

Penyelesaian:

Polinom derajat 2 $\rightarrow n = 2$ (perlu tiga buah titik)

$$p_2(x) = L_0(x)y_0 + L_1(x)y_1 + L_2(x)y_2$$

$$\begin{aligned}
 L_0(x) &= \frac{(x-4)(x-6)}{(1-4)(1-6)} \rightarrow L_0(3.5) = \frac{(3.5-4)(3.5-6)}{(1-4)(1-6)} = 0.083333 \\
 L_1(x) &= \frac{(x-1)(x-6)}{(4-1)(4-6)} \rightarrow L_1(3.5) = \frac{(3.5-1)(3.5-6)}{(4-1)(4-6)} = 1.0417 \\
 L_2(x) &= \frac{(x-1)(x-4)}{(6-1)(6-4)} \rightarrow L_2(3.5) = \frac{(3.5-1)(3.5-4)}{(6-1)(6-4)} = -0.12500
 \end{aligned}$$

Jadi,

$$\begin{aligned}
 p_2(3.5) &= (0.083333)(1.5709) + (1.0417)(1.5727) + (-0.12500)(1.5751) \\
 &= 1.5723 \quad \blacksquare
 \end{aligned}$$

Polinom Lagrange mudah diprogram. Algoritmanya dituliskan pada Program 5.1 berikut ini.

Program 5.1 Polinom Lagrange

```

function Lagrange(x:real; n:integer):real;
{ Menghitung  $y = p_n(x)$ , dengan  $p(x)$  adalah polinom Lagrange derajat  $n$ .
  Titik-titik data telah disimpan di dalam laris  $x[0..n]$  dan  $y[0..n]$ 
}
var
  i, j : integer;
  pi, L : real;
begin
  L:=0;
  for i:=0 to n do
    begin
      pi:=1;
      for j:=0 to n do
        if i<> j then
          pi:=pi*(x - x[j])/(x[i] - x[j]);
        endfor;
      L:=L + y[i]*pi;
    end {for};
    Lagrange:=L;
  end {Lagrange};

```

5.3 Polinom Newton

Polinom Lagrange kurang disukai dalam praktik karena alasan berikut [CHA91]:

1. Jumlah komputasi yang dibutuhkan untuk satu kali interpolasi adalah besar. Interpolasi untuk nilai x yang lain memerlukan jumlah komputasi yang sama karena tidak ada bagian komputasi sebelumnya yang dapat digunakan.
2. Bila jumlah titik data meningkat atau menurun, hasil komputasi sebelumnya tidak dapat digunakan. Hal ini disebakan oleh tidak adanya hubungan antara $p_{n-1}(x)$ dan $p_n(x)$ pada polinom Lagrange.

Polinom Newton dibuat untuk mengatasi kelemahan ini. Dengan polinom Newton, polinom yang dibentuk sebelumnya dapat dipakai untuk membuat polinom derajat yang makin tinggi.

Tinjau kembali polinom lanjar pada persamaan (P.5.7):

$$p_1(x) = y_0 + \frac{(y_1 - y_0)}{(x_1 - x_0)}(x - x_0)$$

Bentuk persamaan ini dapat ditulis sebagai

$$p_1(x) = a_0 + a_1(x - x_0) \quad (\text{P.5.13})$$

yang dalam hal ini

$$a_0 = y_0 = f(x_0) \quad (\text{P.5.14})$$

dan

$$a_1 = \frac{y_1 - y_0}{x_1 - x_0} = \frac{f(x_1) - f(x_0)}{x_1 - x_0} \quad (\text{P.5.15})$$

Persamaan (P.5.15) ini merupakan bentuk selisih-terbagi (*divided-difference*) dan dapat disingkat penulisannya menjadi

$$a_1 = f[x_1, x_0] \quad (\text{P.5.16})$$

Setelah polinom lanjar, polinom kuadratik dapat dinyatakan dalam bentuk

$$p_2(x) = a_0 + a_1(x - x_0) + a_2(x - x_0)(x - x_1) \quad (\text{P.5.17})$$

atau

$$p_2(x) = p_1(x) + a_2(x - x_0)(x - x_1) \quad (\text{P.5.18})$$

Persamaan (P.5.18) memperlihatkan bahwa $p_2(x)$ dapat dibentuk dari polinom sebelumnya, $p_1(x)$. Ini mengarahkan kita pada pembentukan polinom Newton untuk derajat yang lebih tinggi. Nilai a_2 dapat ditemukan dengan menyulihkan $x = x_2$ untuk memperoleh

$$a_2 = \frac{f(x_2) - a_0 - a_1(x_2 - x_0)}{(x_2 - x_0)(x_2 - x_1)} \quad (\text{P.5.19})$$

Nilai a_0 dan nilai a_1 pada persamaan (P.5.14) dan (P.5.15) dimasukkan ke dalam ke dalam persamaan (P.5.19) untuk memberikan

$$a_2 = \frac{\frac{f(x_2) - f(x_0)}{x_2 - x_0} - \frac{f(x_1) - f(x_0)}{x_1 - x_0}}{x_2 - x_1}$$

Dengan melakukan utak-atik aljabar, persamaan terakhir ini lebih disukai ditulis menjadi

$$a_2 = \frac{\frac{f(x_2) - f(x_1)}{x_2 - x_1} - \frac{f(x_1) - f(x_0)}{x_1 - x_0}}{x_2 - x_0} = \frac{f[x_2, x_1] - f[x_1, x_0]}{x_2 - x_0} \quad (\text{P.5.20})$$

Demikianlah seterusnya, kita dapat membentuk polinom Newton secara bertahap: polinom derajat n dibentuk dari polinom derajat $(n-1)$. Polinom Newton dinyatakan dalam hubungan rekursif sebagai berikut:

- (i) rekurens: $p_n(x) = p_{n-1}(x) + a_n(x - x_0)(x - x_1) \dots (x - x_{n-1})$ (P.5.21)
- (ii) basis: $p_0(x) = a_0$

Jadi, tahapan pembentukan polinom Newton adalah sebagai berikut:

$$\begin{aligned} p_1(x) &= p_0(x) + a_1(x - x_0) \\ &= a_0 + a_1(x - x_0) \\ p_2(x) &= p_1(x) + a_2(x - x_0)(x - x_1) \\ &= a_0 + a_1(x - x_0) + a_2(x - x_0)(x - x_1) \\ p_3(x) &= p_2(x) + a_3(x - x_0)(x - x_1)(x - x_2) \\ &= a_0 + a_1(x - x_0) + a_2(x - x_0)(x - x_1) + a_3(x - x_0)(x - x_1)(x - x_2) \\ &\vdots \\ p_n(x) &= p_{n-1}(x) + a_n(x - x_0)(x - x_1) \dots (x - x_{n-1}) \\ &= a_0 + a_1(x - x_0) + a_2(x - x_0)(x - x_1) + a_3(x - x_0)(x - x_1)(x - x_2) + \\ &\quad \dots + a_n(x - x_0)(x - x_1) \dots (x - x_{n-1}) \end{aligned} \quad (\text{P.5.22})$$

Nilai konstanta $a_0, a_1, a_2, \dots, a_n$ merupakan nilai selisih-terbagi, dengan nilai masing-masing:

$$\begin{aligned} a_0 &= f(x_0) \\ a_1 &= f[x_1, x_0] \\ a_2 &= f[x_2, x_1, x_0] \\ &\vdots \\ a_n &= f[x_n, x_{n-1}, \dots, x_1, x_0] \end{aligned}$$

yang dalam hal ini,

$$f[x_i, x_j] = \frac{f(x_i) - f(x_j)}{x_i - x_j} \quad (\text{P.5.23})$$

$$f[x_i, x_j, x_k] = \frac{f[x_i, x_j] - f[x_j, x_k]}{x_i - x_k} \quad (\text{P.5.24})$$

⋮

$$f[x_n, x_{n-1}, \dots, x_1, x_0] = \frac{f[x_n, x_{n-1}, \dots, x_1] - f[x_{n-1}, x_{n-2}, \dots, x_0]}{x_n - x_0} \quad (\text{P.5.25})$$

Dengan demikian polinom Newton pada (P.5.21) dapat ditulis dalam hubungan rekursif sebagai

(i) rekurens:

$$p_n(x) = p_{n-1}(x) + (x - x_0)(x - x_1) \dots (x - x_{n-1}) f[x_n, x_{n-1}, \dots, x_1, x_0] \quad (\text{P.5.26})$$

(ii) basis: $p_0(x) = f(x_0)$

atau dalam bentuk polinom yang lengkap sebagai berikut:

$$\begin{aligned} p_n(x) = & f(x_0) + (x - x_0)f[x_1, x_0] + (x - x_0)(x - x_1)f[x_2, x_1, x_0] \\ & + (x - x_0)(x - x_1) \dots (x - x_{n-1}) f[x_n, x_{n-1}, \dots, x_1, x_0] \end{aligned} \quad (\text{P.5.27})$$

Karena tetapan $a_0, a_1, a_2, \dots, a_n$ merupakan nilai selisih-terbagi, maka polinom Newton dinamakan juga **polinom interpolasi selisih-terbagi Newton**. Nilai selisih terbagi ini dapat dihitung dengan menggunakan tabel yang disebut tabel selisih-terbagi, misalnya tabel selisih-terbagi untuk empat buah titik ($n = 3$) berikut:

i	x_i	$y_i = f(x_i)$	ST-1	ST-2	ST-3
0	x_0	$f(x_0)$	$f[x_1, x_0]$	$f[x_2, x_1, x_0]$	$f[x_3, x_2, x_1, x_0]$
1	x_1	$f(x_1)$	$f[x_2, x_1]$	$f[x_3, x_2, x_1]$	
2	x_2	$f(x_2)$	$f[x_3, x_1]$		
3	x_3	$f(x_3)$			

Keterangan: ST = Selisih-Terbagi

Sekali tabel selisih-terbagi dibentuk, polinom interpolasi yang melewati sekumpulan titik (x_i, y_i) berbeda (misalnya untuk $i = 0, 1, 2$, atau $i = 1, 2, 3$) dapat ditulis dengan mudah. Bila bagian tabel yang diarsir dinyatakan di dalam matriks $ST[0..n, 0..n]$, maka evaluasi $p_n(x)$ untuk $x = t$ dapat dinyatakan sebagai

$$p_n(t) = ST[0,0] + ST[0,1](t - x_0) + ST[0,2](t - x_0)(t - x_1) \\ + \dots + ST[0,n](t - x_0)(t - x_1)\dots(t - x_{n-1})$$

Seperti halnya polinom Lagrange, polinom Newton juga mudah diprogram. Algoritmanya dituliskan pada Program 5.3 di bawah ini.

Program 5.2 Polinom Newton

```

function Newton(x:real; n:integer):real;
{Menghitung  $y = p(x)$ , dengan  $p(x)$  adalah polinom Newton derajat n.
 Titik-titik data telah disimpan di dalam larik  $x[0..n]$  dan  $y[0..n]$ 
}
var
  i, k : integer;
  ST : array[0..30, 0..30] of real; {menyimpan tabel selisih terbagi}
  jumlah, suku: real;

begin
  for k:=0 to n do           { simpan  $y[k]$  pada kolom 0 dari matriks ST }
    ST[k,0]:=y[k];
  {end for}
  for k:=1 to n do          {buat tabel selisih terbagi}
    for i:=0 to n-k do
      ST[i,k]:=(ST[i+1,k-1] - ST[i,k-1])/(x[i+k]-x[i]);
    {end for}
  {end for}
  {hitung  $p(x)$ }
  jumlah:=ST[0,0];
  for i:=1 to n do
    begin
      suku:=ST[0,i];
      for k:=0 to i-1 do
        suku:=suku*(x-x[k]);
      {end for}
      jumlah:=jumlah + suku;
    end;
  Newton:=jumlah;
end;

```

Contoh 5.6

Hitunglah $f(9.2)$ dari nilai-nilai (x, y) yang diberikan pada tabel di bawah ini dengan polinom Newton derajat 3.

Penyelesaian:

Tabel selisih-terbagi:

i	x_i	y_i	ST-1	ST-2	ST-3
0	8.0	2.079442	0.117783	-0.006433	0.000411
1	9.0	2.197225	0.108134	-0.005200	
2	9.5	2.251292	0.097735		
3	11.0	2.397895			

Contoh cara menghitung nilai selisih-terbagi pada tabel adalah:

$$f(x_2, x_1) = \frac{f(x_2) - f(x_1)}{x_2 - x_1} = \frac{2.251292 - 2.197225}{9.5 - 9.0} = 0.108134$$

$$f(x_2, x_1, x_0) = \frac{f[x_2, x_1] - f[x_1, x_0]}{x_2 - x_0} = \frac{0.108134 - 0.117783}{9.5 - 8.0} = -0.006433$$

dan seterusnya.

Nilai-nilai selisih-terbagi yang dibutuhkan untuk membentuk polinom Newton derajat 3 ditandai dengan arsiran.

Polinom Newton-nya (dengan $x_0 = 8.0$ sebagai titik data pertama) adalah:

$$f(x) \approx p_3(x) = 2.079442 + 0.117783(x - 8.0) - 0.006433(x - 8.0)(x - 9.0) + 0.000411(x - 8.0)(x - 9.0)(x - 9.5)$$

Taksiran nilai fungsi pada $x = 9.2$ adalah

$$f(9.2) \approx p_3(9.2) = 2.079442 + 0.141340 - 0.001544 - 0.000030 = 2.219208$$

Nilai sejati $f(9.2) = \ln(9.2) = 2.219203$ (7 angka bena). Catatlah bahwa nilai interpolasi $\ln(9.2)$ semakin teliti dengan meningkatnya orde polinom (Contoh 5.2, Contoh 5.3, dan Contoh 5.6 ini):

$$p_1(9.2) = 2.220782,$$

$$p_2(9.2) = 2.219238,$$

$$p_3(9.2) = 2.219203$$

■

Contoh 5.7

[MAT92] Bentuklah polinom Newton derajat satu, dua, tiga, dan empat yang menghampiri fungsi $f(x) = \cos(x)$ di dalam selang $[0.0, 4.0]$ dan jarak antar titik adalah 1.0. Lalu, taksirlah nilai fungsi di $x = 2.5$ dengan polinom Newton derajat tiga.

Penyelesaian:

Dengan jarak antar titik 1.0, maka titik yang digunakan adalah pada $x_0 = 0.0, x_1 = 1.0, x_2 = 3.0, x_3 = 4.0$. Tabel selisih terbaginya adalah:

<i>i</i>	<i>x_i</i>	<i>f(x_i)</i>	ST-1	ST-2	ST-3	ST-4
0	0.0	1.0000	-0.4597	-0.2484	0.1466	-0.0147
1	1.0	0.5403	-0.9564	0.1913	0.0880	
2	2.0	-0.4161	-0.5739	0.4551		
3	3.0	-0.9900	0.3363			
4	4.0	-0.6536	<i>f(x₃, x₂)</i>			

Contoh cara menghitung nilai selisih-terbagi pada tabel:

$$f[x_1, x_0] = \frac{f(x_1) - f(x_0)}{x_1 - x_0} = \frac{0.5403 - 1.0000}{1.0 - 0.0} = -0.4597$$

$$f[x_2, x_1] = \frac{f(x_2) - f(x_1)}{x_2 - x_1} = \frac{-0.4161 - 0.5403}{2.0 - 1.0} = -0.9564$$

$$f[x_2, x_1, x_0] = \frac{f[x_2, x_1] - f[x_1, x_0]}{x_2 - x_0} = \frac{-0.9564 + 0.4597}{2.0 - 0.0} = -0.2484$$

Maka, polinom Newton derajat 1, 2, dan 3 dengan $x_0 = 0.0$ sebagai titik data pertama adalah

$$\cos(x) \approx p_1(x) = 1.0000 - 0.4597(x - 0.0)$$

$$\cos(x) \approx p_2(x) = 1.0000 - 0.4597(x - 0.0) - 0.2484(x - 0.0)(x - 1.0)$$

$$\cos(x) \approx p_3(x) = 1.0000 - 0.4597(x - 0.0) - 0.2484(x - 0.0)(x - 1.0) + 0.1466(x - 0.0)(x - 1.0)(x - 2.0)$$

$$\cos(x) \approx p_4(x) = 1.0000 - 0.4597(x - 0.0) - 0.2484(x - 0.0)(x - 1.0) + 0.1466(x - 0.0)(x - 1.0)(x - 2.0) - 0.0147(x - 0.0)(x - 1.0)(x - 2.0)(x - 3.0)$$

Grafik $y = \cos(x)$ dan $y = p_1(x)$, $y = p_2(x)$, $y = p_3(x)$, diperlihatkan pada Gambar 5.7. Perhatikan bahwa $y = p_3(x)$ lebih baik dalam menghampiri fungsi $y = \cos(x)$ (kurvanya hampir tepat sama/ berimpit di dalam selang $[0.0, 3.0]$).

Taksiran nilai fungsi di $x = 2.5$ dengan polinom derajat tiga adalah

$$\cos(2.5) \approx p_3(2.5) = 1.0000 - 0.4597(2.5 - 0.0) - 0.2484(2.5 - 0.0)(2.5 - 1.0) + 0.1466(2.5 - 0.0)(2.5 - 1.0)(2.5 - 2.0) \approx -0.8056$$

Nilai sejati $f(2.5)$ adalah

$$f(2.5) = \cos(2.5) = -0.8011$$

sehingga solusi hampiran mengandung galat sejati sebesar

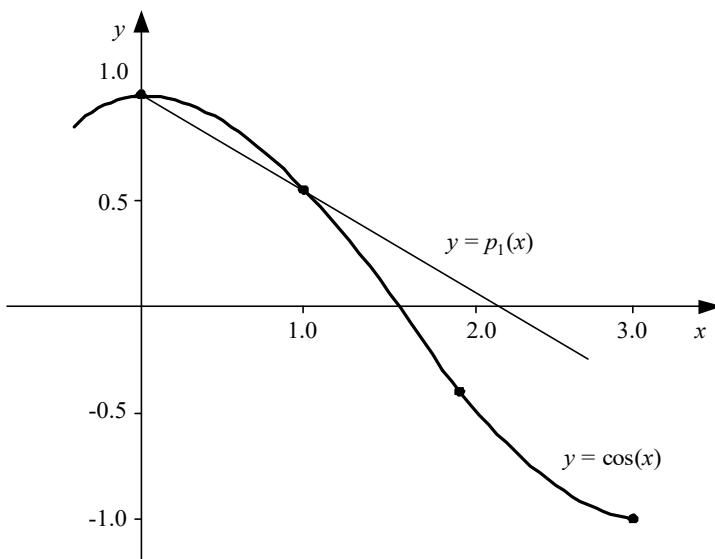
$$\epsilon = -0.8011 - (-0.8056) = -0.0045$$

Catatan:

Titik $x_0 = 0$ tidak selalu harus merupakan ujung selang. Bila $p_3(x)$ didasarkan pada titik $x_0 = 1.0$, $x_1 = 2.0$, $x_3 = 3.0$, dan $x_4 = 4.0$ di dalam selang $[1.0, 4.0]$, maka polinom Newton yang menginterpolasi keempat titik tersebut adalah

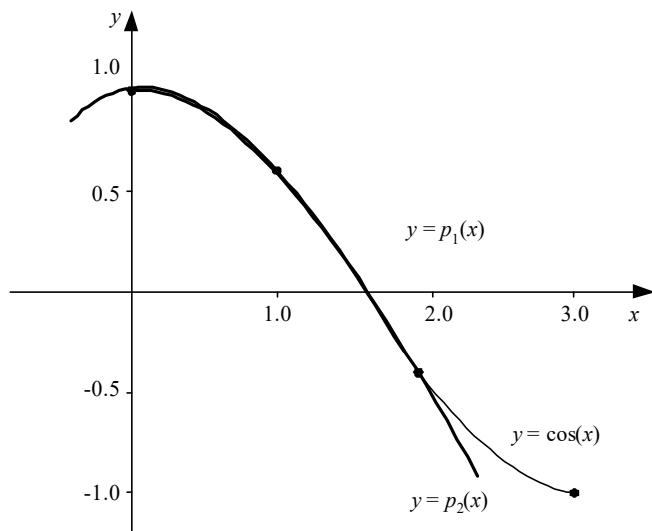
$$p_3(x) = 0.5403 - 0.9564(x - 1.0) + 0.1913(x - 1.0)(x - 2.0) + 0.0880(x - 1.0)(x - 2.0)(x - 3.0)$$

■

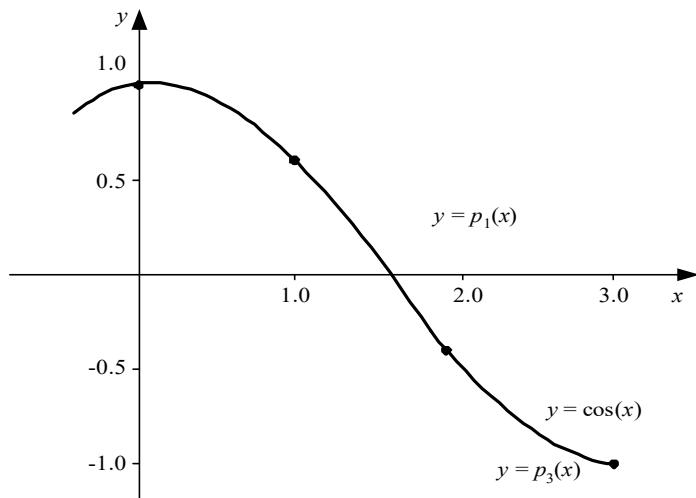


Grafik $y = \cos(x)$ dan polinom Newton derajat 1, $y = p_1(x)$, yang didasarkan pada titik $x_0 = 0.0$ dan $x_1 = 1.0$

Gambar 5.7 Polinom Newton derajat 1 yang menginterpolasi fungsi $y = \cos x$ di dalam selang $[0.0, 4.0]$



Grafik $y = \cos(x)$ dan polinom Newton derajat 2, $y = p_2(x)$, yang didasarkan pada titik $x_0 = 0.0$, $x_1 = 1.0$, $x_2 = 2.0$



Grafik $y = \cos(x)$ dan polinom Newton derajat 3, $y = p_3(x)$, yang didasarkan pada titik $x_0 = 0.0$, $x_1 = 1.0$, $x_2 = 2.0$, dan $x_3 = 3.0$

Gambar 5.7 (lanjutan) Polinom Newton derajat 2 dan 3 yang menginterpolasi fungsi $y = \cos x$ di dalam selang $[0.0, 4.0]$

Kelebihan Polinom Newton

Sekarang kita tuliskan alasan mengapa polinom Newton lebih disukai untuk diprogram, yaitu

1. Karena polinom Newton dibentuk dengan menambahkan satu suku tunggal dengan polinom derajat yang lebih rendah, maka ini memudahkan perhitungan polinom derajat yang lebih tinggi dalam program yang sama [CHA91]. Karena alasan itu, polinom Newton sering digunakan khususnya pada kasus yang derajat polinomnya tidak diketahui terlebih dahulu.
2. Penambahan suku-suku polinom secara beruntun dapat dijadikan kriteria untuk menentukan tercapainya titik berhenti, yaitu apakah penambahan suku-suku yang lebih tinggi tidak lagi secara berarti memperbaiki nilai interpolasi, atau malahan menjadi lebih buruk.
3. Tabel selisih terbagi dapat dipakai berulang-ulang untuk memperkirakan nilai fungsi pada nilai x yang berlainan.

Akan halnya polinom Lagrange, ia disukai karena ia mudah diprogram dan komputasinya tidak memerlukan penyimpanan tabel selisih. Polinom Lagrange biasanya dipakai jika derajat polinom interpolasi diketahui terlebih dahulu.

5.4 Ke unika n Po linom In te rpo la si

Polinom interpolasi hanya ada untuk x_i yang berbeda. Bila terdapat beberapa nilai x yang sama, kita tidak dapat membuat polinom interpolasi yang unik. Misalnya diberikan titik-titik yang ditabulasikan dalam tabel berikut

x	1	2	4	5	6	6
y	4.2	8.5	6.6	5.1	6.3	9.0

Interpolasi keenam titik tersebut dengan polinom derajat lima tidak akan menghasilkan polinom interpolasi yang unik, karena terdapat dua buah titik $x = 6$ dengan nilai y yang berbeda.

Sampai sejauh ini, kita telah membahas dua buah metode polinom interpolasi, yaitu polinom Lagrange dan polinom Newton. Apakah polinom yang dihasilkan oleh kedua metode tersebut sama? Dengan kata lain, apakah polinom interpolasi itu unik (tunggal)? Dapat kita buktikan, bahwa bila polinom interpolasi ada, maka polinom tersebut unik.

Bukti:

Misalkan $p_n(x)$ tidak unik, yang berarti ada polinom lain, misalnya $q_n(x)$, yang juga melewati titik-titik (x_i, y_i) , $i = 0, 1, 2, \dots, n$, yang dalam hal ini

$$p_n(x_i) = q_n(x_i) = y_i$$

Karena $p_n(x)$ dan $q_n(x)$ tidak sama, berarti ada selisih

$$R_n(x) = p_n(x) - q_n(x) \quad (\text{P.5.28})$$

yang dalam hal ini, $R_n(x)$ adalah polinom derajat $\leq n$.
Selanjutnya,

$$R_n(x_i) = p_n(x_i) - q_n(x_i) = y_i - y_i = 0$$

Karena $R_n(x)$ adalah polinom derajat $\leq n$ dan bernilai 0 untuk $(n + 1)$ buah titik, ini mengingatkan kita pada sebuah teorema di dalam kalkulus yang berbunyi:

Polinom derajat $\leq n$ yang mempunyai $(n+1)$ akar berbeda adalah polinom nol (garis $y = 0$)

Jadi, menurut teorema ini,

$$R_n(x) = 0$$

sehingga dengan demikian

$$p_n(x) - q_n(x) = 0$$

atau

$$p_n(x) = q_n(x)$$

Dengan kata lain, $p_n(x)$ unik . ■

Jadi, metode interpolasi apa pun yang kita pakai untuk menginterpolasi $(n+1)$ buah titik data yang sama, polinom interpolasinya -meskipun bentuknya berbeda-beda- bila ditulis ke dalam bentuk baku (P.5.2) adalah sama.

5.5 Galat Interpolasi Polinom

Polinom interpolasi $p_n(x)$ merupakan hampiran terhadap fungsi yang asli $f(x)$. Jadi, $p_n(x)$ tidaklah sama dengan fungsi asli $f(x)$, meskipun pada titik-titik tertentu $f(x)$ dan $p_n(x)$ bersesuaian, yaitu :

$$f(x_i) = p_n(x_i), \quad i = 0, 1, 2, \dots, n$$

Karena $f(x) \neq p_n(x)$, berarti ada selisih (galat) di antara keduanya, sebutlah $E(x)$, yaitu

$$E(x) = f(x) - p_n(x) \quad (\text{P.5.29})$$

Mengingat $f(x_i) = p(x_i)$ untuk $i = 0, 1, 2, \dots, n$, maka harus juga berlaku

$$E(x_i) = f(x_i) - p_n(x_i) = 0$$

yang berarti $E(x)$ mempunyai $(n+1)$ titik nol dari x_0 sampai x_n .

$E(x)$ dapat ditulis sebagai

$$E(x) = f(x) - p_n(x) = (x - x_0)(x - x_1) \dots (x - x_n) R(x) \quad (\text{P.5.30})$$

atau

$$E(x) = Q_{n+1}(x) R(x) \quad (\text{P.5.31})$$

yang dalam hal ini

$$Q_{n+1}(x) = (x - x_0)(x - x_1) \dots (x - x_n) \quad (\text{P.5.32})$$

Catatlah bahwa

$$Q_{n+1}(x_i) = 0 \text{ untuk } i = 0, 1, \dots, n$$

$R(x)$ adalah fungsi yang mencatat nilai-nilai selain dari x_0, x_1, \dots, x_n . Bagaimana menentukan $R(x)$? Jawabannya di bawah ini.

Persamaan (P.5.30) dapat ditulis sebagai

$$f(x) - p_n(x) - (x - x_0)(x - x_1) \dots (x - x_n) R(x) = 0$$

Misal didefinisikan fungsi $W(t)$ sebagai

$$W(t) = f(t) - p_n(t) - (t - x_0)(t - x_1) \dots (t - x_n) R(x) = 0 \quad (\text{P.5.33})$$

Perhatikan di sini bahwa $R(x)$ tidak ditulis sebagai $R(t)$ karena kita akan mencari nilai-nilai x selain t . Persamaan $W(t) = 0$ berarti mempunyai $(n+2)$ titik nol pada $t = x_0, x_1, \dots, x_n$ dan $t = x$. Berdasarkan teorema Rolle yang berbunyi:

Misalkan fungsi f menerus di dalam selang $[a, b]$ dan $f'(x)$ ada untuk semua $a < x < b$. Jika $f(a) = f(b) = 0$, maka terdapat nilai c , dengan $a < c < b$, sedemikian sehingga $f'(c) = 0$.

jika W menerus dan dapat diturunkan pada selang yang berisi $(n+2)$ titik nol, maka :

$$\begin{aligned} W'(t) &= 0 \rightarrow \text{mempunyai } (n+1) \text{ titik nol} \\ W''(t) &= 0 \rightarrow \text{mempunyai } n \text{ titik nol} \\ W'''(t) &= 0 \rightarrow \text{mempunyai } (n-1) \text{ titik nol} \\ \dots \\ W^{(n+1)}(t) &= 0 \rightarrow \text{mempunyai paling sedikit 1 titik nol,} \end{aligned}$$

misal pada $t = c$

$$\begin{aligned} W^{(n+1)}(t) &= 0 = \frac{d^{(n+1)}}{dt^{(n+1)}} [f(t) - p_n(t) - (t - x_0)(t - x_1) \dots (t - x_n) R(x)]_{t=c} \\ &= f^{(n+1)}(c) - 0 - (n+1)! R(x) \end{aligned} \quad (\text{P.5.34})$$

yang dalam hal ini,

$p_n(t)$ adalah polinom derajat n ,

$p_n^{(n)}(t)$ adalah fungsi tetap sehingga $p_n^{(n+1)} = 0$

$Q_{n+1}(t) = (t - x_0)(t - x_1) \dots (t - x_n) = t^{(n+1)} + (\text{suku-suku polinom derajat } \leq n)$

$Q_{n+1}^{(n+1)}(t) = (n+1)! + 0$

$R(x)$ tidak bergantung pada t , jadi ia tidak berubah selama penurunan

Dari persamaan (P.5.34), kita memperoleh

$$R(x) = \frac{f^{(n+1)}(c)}{(n+1)!}, \quad x_0 < c < x_n \quad (\text{P.5.35})$$

Perhatikanlah bahwa persamaan (P.5.35) ini mengingatkan kita pada rumus galat pemotongan pada deret Taylor (lihat Bab 2).

Selanjutnya, sulihkan (P.5.35) ke dalam (P.5.30), menghasilkan

$$E(x) = (x - x_0)(x - x_1) \dots (x - x_n) \frac{f^{(n+1)}(c)}{(n+1)!} \quad (\text{P.5.31})$$

atau

$$E(x) = Q_{n+1}(x) \frac{f^{(n+1)}(c)}{(n+1)!} \quad (\text{P.5.32})$$

dengan

$$Q_{n+1}(x) = (x - x_0)(x - x_1) \dots (x - x_n)$$

Rumus galat ini berlaku untuk semua metode interpolasi polinom, baik polinom Lagrange, polinom Newton, atau polinom interpolasi lainnya. Misalkan kita menginterpolasi dua buah titik dengan polinom Lagrange derajat satu (polinom lanjar). Galat interpolasinya dinyatakan dalam bentuk

$$E(x) = \frac{(x - x_0)(x - x_1)}{2} f''(c)$$

Bila fungsi f diketahui, kita dapat mencari turunannya di $x = c$ untuk menghitung galat interpolasi $E(x)$. Sayangnya, kita tidak mengetahui nilai c ; yang pasti nilai c terletak antara x_0 dan x_n . Jika $f^{(n+1)}$ berubah sangat lambat dalam selang $[x_0, x_n]$, atau $[x_0, x_n]$ adalah selang kecil sedemikian sehingga $f^{(n+1)}$ berubah sangat lambat, maka kita dapat menghampiri $f^{(n+1)}(c)$ dengan $f^{(n+1)}(x_t)$, yang dalam hal ini x_t adalah titik tengah x_0 dan x_n , yaitu $x_t = (x_0 + x_n)/2$. Galat interpolasi dengan menggunakan nilai x_t ini dinamakan **galat rata-rata interpolasi** E_R [NAK93]:

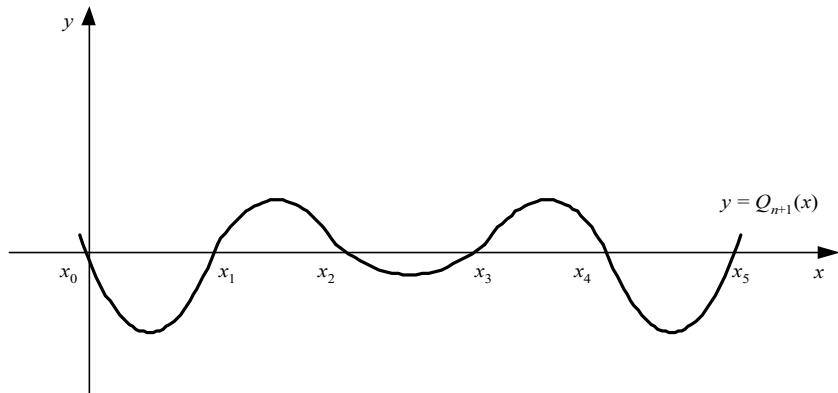
$$E_R(x) = (x - x_0)(x - x_1) \dots (x - x_n) \frac{f^{(n+1)}(x_t)}{(n+1)!} \quad (\text{P.5.33})$$

Dari persamaan (P.5.31) terlihat bahwa galat polinom interpolasi, selain bergantung pada nilai x yang diinterpolasi, juga bergantung pada turunan fungsi semula.

Tinjau kembali Q_{n+1} pada persamaan (P.5.32):

$$Q_{n+1}(x) = (x - x_0)(x - x_1) \dots (x - x_n)$$

Misalkan x_0, x_1, \dots, x_n berjarak sama. Grafik fungsi Q untuk enam titik yang berjarak sama ditunjukkan pada Gambar 5.8.



Gambar 5.8 Grafik fungsi $Q_6(x)$

Berdasarkan $Q_6(x)$ yang berosilasi pada Gambar 5.8 terlihat bahwa:

- di titik-titik data x_i , nilai $Q_6(x_i) = 0$, sehingga galat interpolasi $E(x_i)=0$
- di titik tengah selang, nilai $Q_6(x)$ minimum, sehingga $E(x)$ juga minimum
- di titik-titik sekitar ujung selang, $Q_6(x)$ besar, sehingga $E(x)$ juga besar
- bila ukuran selang $[x_0, x_6]$ semakin besar, amplitudo osilasi meningkat dengan cepat.

Kesimpulan:

Galat interpolasi minimum terjadi untuk nilai x di pertengahan selang. Penjelasannya adalah sebagai berikut.

Nilai-nilai x yang berjarak sama ditulis sebagai

$$x_0, x_1 = x_0 + h, x_2 = x_0 + 2h, \dots, x_n = x_0 + nh$$

atau dengan rumus umum

$$x_i = x_0 + ih \quad , i = 0, 1, 2, \dots, n \quad (\text{P.5.34})$$

Titik yang diinterpolasi dinyatakan sebagai

$$x = x_0 + sh \quad , s \in R \quad (\text{P.5.35})$$

sehingga

$$x - xi = (s - i)h \quad , i = 0, 1, 2, \dots, n \quad (\text{P.5.36})$$

Galat interpolasinya adalah

$$\begin{aligned} E(x) &= (x - x_0)(x - x_1) \dots (x - x_n) \frac{f^{(n+1)}(c)}{(n+1)!} \\ &= (sh)(s-1)h(s-2)h \dots (s-n)h \frac{f^{(n+1)}(c)}{(n+1)!} \\ &= s(s-1)(s-2) \dots (s-n) h^{n+1} \frac{f^{(n+1)}(c)}{(n+1)!} \end{aligned} \quad (\text{P.5.37})$$

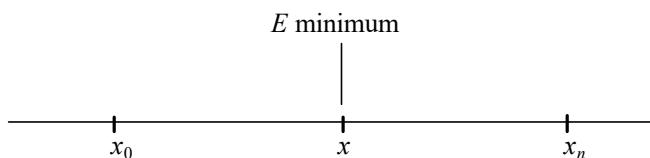
Dapat ditunjukkan bahwa

$$Q_{n+1}(s) = s(s-1)(s-2) \dots (s-n)$$

bernilai minimum bila

$$Q_{n+1}'(s)=0$$

yang dipenuhi untuk $s = n/2$ (buktikan!). Dengan kata lain, $E(x)$ bernilai minimum untuk nilai-nilai x yang terletak di (sekitar) pertengahan selang.



Ingatlah kalimat ini:

Untuk mendapatkan galat interpolasi yang minimum, pilihlah selang $[x_0, x_n]$ sedemikian sehingga x terletak di tengah selang tersebut

Misalkan kepada kita diberikan titik-titik data seperti ini:

x	$f(x)$
0.025	2.831
0.050	3.246
0.075	4.721
0.100	5.210
0.125	6.310
0.150	7.120
0.175	8.512
0.200	9.760
0.225	10.310

Bila anda diminta menghitung $f(0.160)$, maka selang yang digunakan agar galat interpolasi $f(0.160)$ kecil adalah

[0.150, 0.175] → untuk polinom derajat satu

atau

[0.125, 0.200] → untuk polinom derajat tiga

atau

[0.100, 0.225] → untuk polinom derajat lima

5.5.1 Batas Atas Galat Interpolasi Untuk Titik-Titik yang Berjarak Sama

Diberikan absis titik-titik yang berjarak sama:

$$x_i = x_0 + ih, \quad i = 0, 1, 2, \dots, n$$

dan nilai x yang akan diinterpolasikan dinyatakan sebagai

$$x = x_0 + sh, \quad s \in \mathbb{R}$$

Untuk polinom interpolasi derajat 1, 2, dan 3 yang dibentuk dari x_i di atas dapat dibuktikan bahwa

$$(i) |E_1(x)| = |f(x) - p_1(x)| \leq \frac{1}{8} h^2 \underset{x_0 \leq c \leq x_1}{\text{Maks}} |f''(c)| \quad (\text{P.5.38})$$

$$(ii) |E_2(x)| = |f(x) - p_2(x)| \leq \frac{\sqrt{3}}{27} h^3 \underset{x_0 \leq c \leq x_2}{\text{Maks}} |f'''(c)| \quad (\text{P.5.39})$$

$$(iii) |E_3(x)| = |f(x) - p_3(x)| \leq \frac{1}{24} h^4 \underset{x_0 \leq c \leq x_3}{\text{Maks}} |f^{iv}(c)| \quad (\text{P.5.40})$$

Di sini kita hanya membuktikan untuk (i) saja:

Bukti:

Misalkan $x_0 = 0$ dan $x_i = h$, persamaan galatnya adalah

$$E_1(x) = (x - x_0)(x - x_1) \frac{f''(c)}{2!}, \text{ yang dalam hal ini } x_0 < c < x_1$$

$$= \frac{x(x-h)}{2!} f''(c)$$

$$|E_1(x)| = \frac{1}{2} |x^2 - xh f''(c)|$$

$$= \frac{1}{2} |x^2 - xh| |f''(c)| \leq \frac{1}{2} \underset{x_0 \leq x \leq x_1}{\text{Maks}} |x^2 - xh| \underset{x_0 \leq c \leq x_1}{\text{Maks}} |f''(c)|$$

Misalkan

$$\phi(x) = x^2 - xh$$

Di dalam selang $[x_0, x_1]$, nilai maksimum lokal $\phi(x)$ dapat terjadi pada ujung-ujung selang ($x = 0$ atau $x = h$) atau pada titik ekstrim $\phi(x)$. Terlebih dahulu tentukan titik ekstrim $\phi(x)$ dengan cara membuat turunan pertamanya sama dengan 0:

$$\phi'(x) = 2x - h = 0 \rightarrow x = h/2$$

Hitung nilai maksimum lokal $\phi(x)$ di ujung-ujung selang dan titik ekstrim:

- di ujung selang kiri $x = 0, \rightarrow \phi(0) = 0^2 - 0h = 0$
- di ujung selang kanan $x = h \rightarrow \phi(h) = h^2 - h^2 = 0$
- di titik ekstrim $x = h/2 \rightarrow \phi(h/2) = (h/2)^2 - (h/2)h = -1/4 h^2$

Jadi, maksimum $|\phi(x)| = \frac{1}{4} h^2$, sehingga dengan demikian

$$|E_1(x)| = |f(x) - p_1(x)| \leq \frac{1}{8} h^2 \underset{x_0 \leq c \leq x_1}{\text{Maks}} |f''(c)| \quad \blacksquare$$

Contoh 5.8

Tinjaulah kembali tabel yang berisi pasangan titik $(x, f(x))$ yang diambil dari $f(x) = \cos(x)$.

x_i	$f(x_i)$
0.0	1.000000
1.0	0.5403023
2.0	-0.4161468
3.0	-0.9899925
4.0	-0.6536436

- (a) Hitung galat rata-rata interpolasi di titik $x = 0.5$, $x = 1.5$, dan $x = 2.5$, bila x diinterpolasi dengan polinom Newton derajat 3 berdasarkan $x_0 = 0$.
- (b) Hitung batas atas galat interpolasi bila kita melakukan interpolasi titik-titik berjarak sama dalam selang $[0.0, 3.0]$ dengan polinom interpolasi derajat 3.
- (c) Hitung batas atas dan batas bawah galat interpolasi di $x = 0.5$ dengan polinom Newton derajat 3

Penyelesaian:

- (a) Telah diketahui dari Contoh 5.7 bahwa polinom derajat 3 yang menginterpolasi $f(x) = \cos(x)$ dalam selang $[0.0, 3.0]$ adalah :

$$\cos(x) \approx p_3(x) = 1.0000 - 0.4597(x - 0.0) - 0.2485(x - 0.0)(x - 1.0) + 0.1466(x - 0.0)(x - 1.0)(x - 2.0)$$

Menghitung galat rata-rata interpolasi :

Titik tengah selang $[0.0, 3.0]$ adalah di $x_m = (0.0 + 3.0)/2 = 1.5$

Galat rata-rata interpolasi adalah :

$$E_3(x) = \frac{(x - 0.0)(x - 1.0)(x - 2.0)(x - 3.0)}{4!} f^{(4)}(x_m)$$

Hitung turunan keempat dari fungsi $f(x) = \cos(x)$,

$$\begin{aligned} f'(x) &= -\sin(x); \\ f''(x) &= -\cos(x); \\ f'''(x) &= \sin(x) \\ f^{(4)}(x) &= \cos(x) \end{aligned}$$

karena itu,

$$E_3(x) = \frac{(x-0.0)(x-1.0)(x-2.0)(x-3.0)}{4!} (\cos(1.5))$$

Untuk $x = 0.5$, $x = 1.5$, dan $x = 2.5$, nilai-nilai interpolasinya serta galat rata-rata interpolasinya dibandingkan dengan nilai sejati dan galat sejati diperlihatkan oleh tabel berikut :

X	$f(x)$	$p_3(x)$	$E_3(x)$	Galat sejati
0.5	0.8775826	0.8872048	0.0027632	-0.0096222
1.5	0.0707372	0.0692120	-0.0016579	0.0015252
2.5	-0.8011436	-0.8058546	0.0027632	0.0047110

Catatan:

Perhatikan bahwa karena $x = 1.5$ terletak di titik tengah selang, maka galat interpolasinya lebih paling kecil dibandingkan interpolasi x yang lain.

- (b) Telah diketahui bahwa batas atas galat interpolasi dengan polinom derajat 3 adalah

$$E_3(x) = |f(x) - p_3(x)| \leq h^4/24 \operatorname{Max} |f^{(4)}(c)|, x_0 \leq c \leq x_3.$$

Telah diperoleh dari (a) bahwa $f^{(4)}(x) = \cos(x)$, dan dalam selang $[0.0, 3.0]$ nilai $\operatorname{Max} |f^{(4)}(x)|$ terletak di $x = 0.0$. Jadi, $|f^{(4)}(x)| = |\cos(0.0)| = 1.000000$. Untuk $p_3(x)$ dengan jarak antar titik data adalah $h = 1.0$, batas atas galat interpolasinya adalah

$$E_3(x) \leq (1.0)^4 1.000000/24 = 1/24 = 0.0416667.$$

Nilai-nilai $E_3(x)$ pada tabel di atas semuanya di bawah 0.0416667. Jadi, batas atas 0.0416667 berasalan.

$$(c) E_3(x) = \frac{(x-0.0)(x-1.0)(x-2.0)(x-3.0)}{4!} f^{(4)}(1.5)$$

$$E_3(0.5) = \frac{(0.5-0.0)(0.5-1.0)(0.5-2.0)(0.5-3.0)}{4!} (-\cos(c)), 0.0 \leq c \leq 3.0$$

Karena fungsi cosinus monoton dalam selang $[0.0, 3.0]$, maka nilai maksimum dan nilai minimum untuk $\cos(c)$ terletak pada ujung-ujung selang. Untuk $c = 0.0$ maka :

$$\begin{aligned} E_3(0.5) &= \frac{(0.5-0.0)(0.5-1.0)(0.5-2.0)(0.5-3.0)}{4!} (\cos(0.0)) \\ &= -0.0390625 \text{ (minimum)}, \end{aligned}$$

dan untuk $c = 3.0$ maka

$$\begin{aligned}E_3(0.5) &= \frac{(0.5 - 0.0)(0.5 - 1.0)(0.5 - 2.0)(0.5 - 3.0)}{4!} (\cos(3.0)) \\&= 0.0386716 \text{ (maksimum)},\end{aligned}$$

sehingga, batas-batas galat interpolasi di $x = 0.5$ adalah :

$$-0.0390625 \leq E_3(0.5) \leq 0.0386716$$

■

5.5.2 Taksiran Galat Interpolasi Newton

Salah satu kelebihan polinom Newton dibandingkan dengan polinom Lagrange adalah kemudahan menghitung taksiran galat interpolasi meskipun fungsi asli $f(x)$ tidak diketahui, atau kalaupun ada, sukar diturunkan.

Tinjau kembali polinom Newton:

$$p_n(x) = p_{n-1}(x) + (x - x_0)(x - x_1) \dots (x - x_{n-1}) f[x_n, x_{n-1}, \dots, x_1, x_0]$$

Suku

$$(x - x_0)(x - x_1) \dots (x - x_{n-1}) f[x_n, x_{n-1}, \dots, x_1, x_0]$$

dinaikkan dari n sampai $n + 1$ menjadi

$$(x - x_0)(x - x_1) \dots (x - x_{n-1})(x - x_n) f[x_{n+1}, x_n, x_{n-1}, \dots, x_1, x_0]$$

Bentuk terakhir ini bersesuaian dengan rumus galat interpolasi

$$E(x) = (x - x_0)(x - x_1) \dots (x - x_n) \frac{f^{(n+1)}(t)}{(n+1)!}$$

Ekspresi

$$\frac{f^{(n+1)}(t)}{(n+1)!}$$

dapat dihampiri nilainya dengan

$$f[x_{n+1}, x_n, x_{n-1}, \dots, x_1, x_0]$$

yang dalam hal ini $f(x_{n+1}, x_n, x_{n-1}, \dots, x_1, x_0)$ adalah selisih-terbagi ke $(n + 1)$.

Jadi,

$$\frac{f^{(n+1)}(t)}{(n+1)!} \approx f[x_{n+1}, x_n, x_{n-1}, \dots, x_1, x_0] \quad (\text{P.5.41})$$

sehingga taksiran galat interpolasi Newton dapat dihitung sebagai

$$E(x) = (x - x_0)(x - x_1) \dots (x - x_n)f[x_{n+1}, x_n, x_{n-1}, \dots, x_1, x_0] \quad (\text{P.5.42})$$

asalkan tersedia titik tambahan x_{n+1} .

Contoh 5.9

Pada Contoh 5.7, bila digunakan polinom derajat tiga untuk menaksir nilai $f(2.5)$, hitunglah taksiran galat interpolasinya.

Penyelesaian:

Bila digunakan polinom derajat tiga, maka tersedia titik sesudah $x_3 = 3.0$, yaitu $x_4 = 4.0$, dan dari tabel selisih-terbagi ditemukan

$$f[x_4, x_3, x_2, x_1, x_0] = -0.0147$$

sehingga taksiran galat dalam menginterpolasi $f(2.5)$ adalah

$$E(2.5) = (2.5 - 0.0)(2.5 - 1.0)(2.5 - 2.0)(2.5 - 3.0) (-0.0147) = 0.01378125 \quad \blacksquare$$

5.5.3 Taksiran Galat Interpolasi Lagrange

Taksiran galat polinom Lagrange tidak dapat dihitung secara langsung karena tidak tersedia rumus taksiran galat seperti halnya pada interpolasi Newton. Namun, jika tabel selisih-terbagi tersedia, maka taksiran galatnya dapat dihitung dengan rumus taksiran galat polinom Newton:

$$E(x) = (x - x_0)(x - x_1) \dots (x - x_n)f[x_{n+1}, x_n, x_{n-1}, \dots, x_1, x_0]$$

asalkan tersedia titik tambahan x_{n+1} . Meskipun demikian, tabel selisih-terbagi tidak dipakai sebagai bagian dari algoritma Lagrange, ini jarang terjadi [CHA91].

5.6 Polinom Newton-Gregory

Polinom Newton-Gregory merupakan kasus khusus dari polinom Newton untuk titik-titik yang berjarak sama. Pada kebanyakan aplikasi nilai-nilai x berjarak sama, misalnya pada tabel nilai fungsi, atau pada pengukuran yang dilakukan pada selang waktu yang teratur [KRE88].

Untuk titik-titik yang berjarak sama, rumus polinom Newton menjadi lebih sederhana. Selain itu, tabel selisih-terbaginya pun lebih mudah dibentuk. Di sini kita menamakan tabel tersebut sebagai **tabel selisih** saja, karena tidak ada proses pembagian dalam pembentukan elemen tabel.

Ada dua macam tabel selisih, yaitu tabel selisih maju (*forward difference*) dan tabel selisih mundur (*backward difference*). Karena itu, ada dua macam polinom Newton-Gregory, yaitu polinom **Newton-Gregory maju** dan polinom **Newton-Gregory mundur**.

5.6.1 Polinom Newton-Gregory Maju

Polinom Newton-Gregory maju diturunkan dari tabel selisih maju. Sebelum menurunkan rumusnya, kita bahas terlebih dahulu tabel selisih maju.

5.6.1.1 Tabel Selisih Maju

Misal diberikan lima buah titik dengan absis x yang berjarak sama. Tabel selisih maju yang dibentuk dari kelima titik tersebut adalah

x	$f(x)$	Δf	$\Delta^2 f$	$\Delta^3 f$	$\Delta^4 f$
x_0	f_0	Δf_0	$\Delta^2 f_0$	$\Delta^3 f_0$	$\Delta^4 f_0$
x_1	f_1	Δf_1	$\Delta^2 f_1$	$\Delta^3 f_1$	
x_2	f_2	Δf_2	$\Delta^2 f_2$		
x_3	f_3	Δf_3			
x_4	f_4				

Lambang Δ menyatakan selisih maju. Arti setiap simbol di dalam tabel adalah:

$$f_0 = f(x_0) = y_0$$

$$f_1 = f(x_1) = y_1$$

...

$$f_4 = f(x_4)$$

Notasi: $f_p = f(x_p)$

$$\begin{aligned}\Delta f_0 &= f_1 - f_0 \\ \Delta f_1 &= f_2 - f_1 \\ &\dots \\ \Delta f_3 &= f_4 - f_3 \\ \text{Notasi: } \Delta f_p &= f_{p+1} - f_p\end{aligned}$$

$$\begin{aligned}\Delta^2 f_0 &= \Delta f_1 - \Delta f_0 \\ \Delta^2 f_1 &= \Delta f_2 - \Delta f_0 \\ \Delta^2 f_2 &= \Delta f_3 - \Delta f_1 \\ \text{Notasi: } \Delta^2 f_p &= \Delta f_{p+1} - \Delta f_p\end{aligned}$$

$$\begin{aligned}\Delta^3 f_0 &= \Delta^2 f_1 - \Delta^2 f_0 \\ \Delta^3 f_1 &= \Delta^2 f_2 - \Delta^2 f_0 \\ \text{Notasi: } \Delta^3 f_p &= \Delta^2 f_{p+1} - \Delta^2 f_p\end{aligned}$$

Bentuk umum:

$$\Delta^{n+1} f_p = \Delta^n f_{p+1} - \Delta^n f_p, \quad n = 0, 1, 2, \dots \quad (\text{P.5.43})$$

5.6.1.2 Penurunan Rumus Polinom Newton-Gregory Maju

Sekarang kita mengembangkan polinom Newton-Gregory maju yang didasarkan pada tabel selisih maju.

$$\begin{aligned}f[x_1, x_0] &= \frac{f(x_1) - f(x_0)}{x_1 - x_0} \\ &= \frac{\Delta f(x_0)}{h} \\ &= \frac{\Delta f_0}{1! h} \quad (\text{P.5.44})\end{aligned}$$

$$\begin{aligned}f[x_1, x_2, x_0] &= \frac{f[x_2, x_1] - f[x_1, x_0]}{x_2 - x_0} \\ &= \frac{\frac{f(x_2) - f(x_1)}{x_2 - x_1} - \frac{f(x_1) - f(x_0)}{x_1 - x_0}}{x_2 - x_0}\end{aligned}$$

$$\begin{aligned}
& \frac{\Delta f_1 - \Delta f_0}{h} \\
&= \frac{\Delta^2 f_0}{2h} \\
&= \frac{\Delta^2 f_0}{2!h^2}
\end{aligned} \tag{P.5.45}$$

Bentuk umum:

$$f[x_n, \dots, x_1, x_0] = \frac{\Delta^n f(x_0)}{n!h^n} = \frac{\Delta^n f_0}{n!h^n} \tag{P.5.46}$$

Dengan demikian polinom Newton untuk data berjarak sama dapat ditulis sebagai :

$$\begin{aligned}
p_n(x) &= f(x_0) + (x - x_0)f[x_1, x_0] + (x - x_0)(x - x_1)f[x_2, x_1, x_0] + \dots + \\
&\quad (x - x_0)(x - x_1)\dots(x - x_{n-1})f[x_n, x_{n-1}, \dots, x_1, x_0] \\
&= f_0 + (x - x_0)\frac{\Delta f_0}{1!h} + (x - x_0)(x - x_1)\frac{\Delta^2 f_0}{2!h^2} + \dots + \\
&\quad (x - x_0)(x - x_1)\dots(x - x_{n-1})\frac{\Delta^n f_0}{n!h^n}
\end{aligned} \tag{P.5.47}$$

Persamaan (P.5.47) ini dinamakan polinom Newton-Gregory maju. Persamaan (P.5.47) dapat juga ditulis sebagai relasi rekursif:

$$p_n(x) = p_{n-1}(x) + (x - x_0)(x - x_1)\dots(x - x_{n-1})\frac{\Delta^n f_0}{n!h^n} \tag{P.5.48}$$

Jika titik-titik berjarak sama dinyatakan sebagai

$$x_i = x_0 + ih \quad , i = 0, 1, 2, \dots, n$$

dan nilai x yang diinterpolasikan adalah

$$x = x_0 + sh \quad , s \in R$$

maka, persamaan (P.5.47) dapat juga ditulis dalam parameter s sebagai

$$p_n(x) = f_0 + \frac{sh}{1!h} \Delta f_0 + \frac{s(s-1)h^2}{2!h^2} \Delta^2 f_0 + \dots + \\ \frac{s(s-1)(s-2)\dots(s-n+1)h^n}{n!h^n} \Delta^n f_0$$

yang menghasilkan

$$p_n(x) = f_0 + \frac{s}{1!} \Delta f_0 + \frac{s(s-1)}{2!} \Delta^2 f_0 + \dots + \frac{s(s-1)(s-2)\dots(s-n+1)}{n!} \Delta^n f_0 \quad (\text{P.5.49})$$

atau dalam bentuk relasi rekursif,

$$\begin{aligned} \text{(i) rekurens: } p_n(x) &= p_{n-1}(x) + \frac{s(s-1)(s-2)\dots(s-n+1)}{n!} \Delta^n f_0 \\ \text{(ii) basis: } p_0(x) &= f(x_0) \end{aligned} \quad (\text{P.5.50})$$

Seringkali persamaan (P.5.49) dinyatakan dalam bentuk binomial:

$$p_n(x) = \sum_{k=0}^n \binom{s}{k} \Delta^k f_0 \quad (\text{P.5.51})$$

yang dalam hal ini,

$$\binom{s}{0} = 1, \quad \binom{s}{k} = \frac{s(s-1)(s-2)\dots(s-k+1)}{k!} \quad (s > 0, \text{ bilangan bulat})$$

dan

$$k! = 1 \times 2 \times \dots \times k$$

Tahap pembentukan polinom Newton-Gregory maju untuk titik-titik berjarak sama dapat dituliskan sebagai berikut:

$$\begin{aligned} p_0(x) &= f_0 \\ p_1(x) &= p_0(x) + \frac{s}{1!} \Delta f_0 = f_0 + \frac{s}{1!} \Delta f_0 \end{aligned}$$

$$\begin{aligned}
p_2(x) &= p_1(x) + \frac{s(s-1)}{2!} \Delta^2 f_0 \\
&= f_0 + \frac{s}{1!} \Delta f_0 + \frac{s(s-1)}{2!} \Delta^2 f_0 \\
p_3(x) &= p_2(x) + \frac{s(s-1)(s-2)}{2!} \Delta^3 f_0 \\
&= f_0 + \frac{s}{1!} \Delta f_0 + \frac{s(s-1)}{2!} \Delta^2 f_0 + \frac{s(s-1)(s-2)}{2!} \Delta^3 f_0 \\
&\dots \\
p_n(x) &= f_0 + \frac{s}{1!} \Delta f_0 + \frac{s(s-1)}{2!} \Delta^2 f_0 + \frac{s(s-1)(s-2)}{2!} \Delta^3 f_0 + \dots \\
&\quad \frac{s(s-1)(s-2)\dots(s-n+1)}{n!} \Delta^n f_0
\end{aligned}$$

Contoh 5.10

[NOB72] Bentuklah tabel selisih untuk fungsi $f(x) = 1/(x+1)$ di dalam selang $[0.000, 0.625]$ dan $h = 0.125$. Hitung $f(0.300)$ dengan polinom Newton-Gregory maju derajat 3.

Penyelesaian:

Tabel selisih maju:

x	$f(x)$	Δ	Δ^2	Δ^3
0.000	1.000	-0.111	0.022	-0.006
0.125	0.889	-0.089	0.016	-0.003
0.250	0.800	-0.073	0.013	-0.005
0.375	0.727	-0.060	0.008	
0.500	0.667	-0.052		
0.625	0.615			

Untuk memperkirakan $f(0.300)$ dengan polinom Newton-Gregory maju derajat tiga, dibutuhkan 4 buah titik. Ingatlah kembali bahwa galat interpolasi akan minimum jika x terletak di sekitar pertengahan selang. Karena itu, titik-titik yang diambil adalah

$$x_0 = 0.125, x_1 = 0.250, x_2 = 0.375, x_3 = 0.500$$

karena $x = 0.300$ terletak di sekitar pertengahan selang $[0.125, 0.500]$.

Diketahui

$$h = 0.125$$

dan

$$x = x_0 + sh \rightarrow s = \frac{x - x_0}{h} = \frac{0.310 - 0.125}{0.125} = 1.4$$

Nilai $f(0.300)$ dihitung dengan polinom Newton-Gregory maju derajat tiga:

$$\begin{aligned} p_3(x) &\approx f_0 + \frac{s}{1!} \Delta f_0 + \frac{s(s-1)}{2!} \Delta^2 f_0 + \frac{s(s-1)(s-2)}{3!} \Delta^3 f_0 \\ &\approx 0.889 + (1.4)(-0.089) + \frac{(1.4)(0.4)}{2}(0.016) + \\ &\quad \frac{(1.4)(0.4)(-0.6)}{6} (-0.003) \\ &\approx 0.889 - 0.1246 + 0.0045 \\ &\approx 0.769 \end{aligned}$$

Sebagai perbandingan, nilai sejati $f(0.300)$ adalah

$$f(0.300) = 1/(0.300+1) = 0.769$$

■

Program 5.3 Polinom Newton-Gregory Maju

```
function Newton_Gregory_Maju(x:real; n:integer):real;
{ - Menghitung y = p(x), dengan p(x) adalah polinom Newton - Gregory
  maju derajat n.
  - Titik-titik data telah disimpan di dalam larik:
    x[0..n] dan y[0..n]
}
var
  i, k : integer;
  TS : array[0..30, 0..30] of real; {menyimpan tabel selisih}
  h, jumlah, suku, s: real;

  function faktorial(p:integer):integer;
  { menghitung p! }
  var
    k, fak:integer;
  begin
    fak:=1;
    for k:=2 to p do
      fak:=fak*k;
    {end for}
    faktorial:=fak;
  end; {faktorial}
```

```

begin
    for k:=0 to n do          {simpan  $y[k]$  pada kolom 0 matriks  $TS[k, j]$  }
        TS[k, 0]:=y[k];
    {end for}
    for k:=1 to n do          {bentuk tabel selisih}
        for i:=0 to (n-k) do
            TS[i, k]:=TS[i+1, k-1] - TS[i, k-1];
        {end for}
    {end for}

    {hitung  $p(x)$  }
    h:=x[1]-x[0];           { jarak antar titik}

    s:=(x -x[0])/h;
    jumlah:=TS[0, 0];
    for i:=1 to n do
        begin
            suku:=TS[0, i];
            for k:=0 to i-1 do
                suku:=suku*(s-k)
            {end for}
            suku:=suku/faktorial(i);
            jumlah:=jumlah + suku;
        end;
        Newton_Gregory_Maju:=jumlah;
    end;

```

5.6.1.3 Menghitung Batas Galat Interpolasi Newton-Gregory Maju

Seperti halnya pada polinom Newton, kita dapat menghitung batas-batas galat interpolasi Newton-Gregory Maju. Perhatikan Contoh 5.11 di bawah ini.

Contoh 5.11

Misal diberikan tabel selisih yang diambil dari fungsi $f(x) = \sin(x)$ di dalam selang $[0.1, 1.7]$ dan $h = 0.4$.

x	$f(x)$	Δf	$\Delta^2 f$	$\Delta^3 f$
0.1	0.09983	0.37960	-0.07570	-0.04797
0.5	0.47943	0.30390	-0.12367	-0.02846
0.9	0.78333	0.18023	-0.152134	
1.3	0.96356	0.02810		
1.7	0.99166			

Diminta menentukan $f(0.8)$ dengan polinom Newton-Gregory maju derajat dua, dan tentukan juga batas-batas galatnya.

Penyelesaian:

Polinom derajat dua \rightarrow jumlah titik = $2 + 1 = 3$.

Misalkan titik yang diambil adalah $x_0 = 0.1$, $x_1 = 0.5$, dan $x_2 = 0.9$

Titik x yang diinterpolasikan adalah $x = 0.8$

$$s = (x - x_0)/h = (0.8 - 0.1)/0.4 = 1.75$$

Jadi,

$$\begin{aligned} f(0.8) \approx p_2(x) &= f_0 + \frac{s\Delta f_0}{1!} + \frac{s(s-1)\Delta^2 f_0}{2!} \\ &= 0.09983 + \frac{(1.75)}{1} (0.37960) + \frac{(1.75)(0.75)}{2} (-0.07570) \\ &= 0.71445 \end{aligned}$$

Batas-batas galat:

$$\begin{aligned} E(x) &\approx \frac{s(s-1)(s-2)}{(n+1)!} h^3 f^{(n)}(t) \\ E(0.8) &\approx \frac{(1.75)(0.75)(-0.25)(0.4)^3}{3!} [-\cos(t)] \end{aligned}$$

Dalam selang $[0.1, 0.9]$ fungsi cosinus monoton naik, sehingga nilai minimum dan nilai maksimum cosinus terletak di ujung-ujung selang.

Dengan demikian,

$$\text{galat} \leq \frac{(1.75)(0.75)(-0.25)(0.4)^3}{3!} [-\cos(0.1)] = 3.48 \times 10^{-3}$$

$$\text{galat} \geq \frac{(1.75)(0.75)(-0.25)(0.4)^3}{3!} [-\cos(0.9)] = 2.18 \times 10^{-3}$$

Jadi, batas-batas galat dalam menginterpolasi $f(0.8)$ adalah

$$2.18 \times 10^{-3} \leq \text{galat} \leq 3.48 \times 10^{-3}$$

■

5.6.1.4 Taksiran Galat Interpolasi Newton-Gregory Maju

Seperti halnya pada polinom Newton, taksiran galat interpolasi Newton-Gregory dapat dihitung dengan menghampiri turunan fungsi ke $(n+1)$ dengan nilai pada tabel selisih.

Tinjau kembali polinom Newton-Gregory Maju:

$$p_n(x) = p_{n-1}(x) + (x - x_0)(x - x_1)\dots(x - x_{n-1}) \frac{\Delta^n f_0}{n! h^n}$$

Naikkan suku

$$(x - x_0)(x - x_1)\dots(x - x_{n-1}) \frac{\Delta^n f_0}{n! h^n}$$

dari n menjadi $n+1$:

$$(x - x_0)(x - x_1)\dots(x - x_{n-1})(x - x_n) \frac{\Delta^{n+1} f_0}{(n+1)! h^{n+1}}$$

Bentuk terakhir ini bersesuaian dengan rumus galat interpolasi

$$E(x) = (x - x_0)(x - x_1)\dots(x - x_n) \frac{f^{(n+1)}(t)}{(n+1)!}$$

sehingga, $f^{(n+1)}(t)$ dapat dihampiri dengan

$$f^{(n+1)}(t) \approx \frac{\Delta^{n+1} f_0}{h^{n+1}} \quad (\text{P.5.52})$$

Jadi, taksiran galat dalam menginterpolasi $f(x)$ dengan polinom Newton-Gregory maju adalah

$$E(x) = (x - x_0)(x - x_1)\dots(x - x_n) \frac{\Delta^{n+1} f_0}{h^{n+1} (n+1)!} \quad (\text{P.5.53})$$

atau dalam bentuk lain,

$$E(x) = s(s-1)(s-2)\dots(s-n) \frac{\Delta^{n+1} f_0}{(n+1)!} \quad (\text{P.5.54})$$

dengan $s = (x - x_0) / h$.

Contoh 5.12

Dari Contoh 5.11, hitung taksiran galat dalam menginterpolasi $f(0.8)$.

Penyelesaian:

Dengan menggunakan titik tambahan $x = 1.3$, nilai $\Delta^{n+1} f_0$ dapat dihitung, yang pada tabel selisih nilainya sudah ada, yaitu

$$\Delta^{n+1} f_0 = -0.04797$$

sehingga taksiran galat dalam menginterpolasi $f(0.8)$ adalah

$$E(0.8) \approx \frac{s(s-1)(s-2)}{3!} \Delta^3 f_0 = \frac{(1.75)(0.75)(-0.25)(-0.04797)}{3!}$$

$$= 2.62 \times 10^{-3}$$

■

Persamaan (P.5.53) atau (P.5.54) hanya digunakan bila titik x_{n+1} ada (pada Contoh 5.11, tersedia titik sesudah $x_2 = 0.9$, yaitu $x_3 = 1.3$). Bagaimana kalau titik x_{n+1} tidak ada? Untuk kasus ini kita dapat menggunakan $\Delta^{n+1} f_{-1}$ sebagai hampiran $\Delta^{n+1} f_0$ [NAK93].

5.6.1.5 Manfaat Tabel Selisih Maju

Pada contoh-contoh perhitungan yang diberikan sebelum ini, derajat polinom interpolasi ditentukan pada soal. Bila polinom interpolasi derajat n yang diinginkan, maka jumlah titik yang dibutuhkan harus $(n+1)$ buah. Sebaliknya, bila diberikan $(n+1)$ titik, maka kita dapat menginterpolasi titik-titik itu dengan polinom derajat satu (jadi hanya dua titik yang diperlukan), polinom derajat dua (tiga titik), polinom derajat tiga (empat titik) dan maksimal polinom derajat n (jadi semua titik yang dipakai). Timbul pertanyaan, dengan polinom derajat berapakah sekumpulan titik data sebaiknya diinterpolasi agar memberikan galat interpolasi yang minimum?

[NOB72] Misalkan kita membentuk tabel selisih untuk fungsi $f(x) = x$, $f(x) = x^2$, dan $f(x) = x^3$ pada titik-titik x yang berjarak sama, yaitu

$$x_i = x_0 + ih, \quad i = 0, 1, 2, 3, \dots$$

(i)

x	$f(x) = x$	Δf	$\Delta^2 f$	$\Delta^3 f$
0	0	h	0	0
h	h	h	0	
$2h$	$2h$	h		
$3h$	$3h$			

(ii)

x	$f(x) = x^2$	Δf	$\Delta^2 f$	$\Delta^3 f$
0	0	h^2	$2h^2$	0
h	h^2	$3h^2$	$2h^2$	0
$2h$	$4h^2$	$5h^2$	$2h^2$	
$3h$	$9h^2$	$7h^2$		
$4h$	$16h^2$			

(iii)

x	$f(x) = x^3$	Δf	$\Delta^2 f$	$\Delta^3 f$	$\Delta^4 f$
0	0	h^3	$6h^3$	$6h^3$	0
h	h^3	$7h^3$	$12h^3$	$6h^3$	
$2h$	$8h^3$	$19h^3$	$18h^3$		
$3h$	$27h^3$	$37h^3$			
$4h$	$64h^3$				

Apa yang anda temukan dari ketiga tabel di atas? Pada ketiga tabel itu dapat disimpulkan bahwa untuk $f(x) = ax^n$, yang dalam hal ini $a = 1$ dan $n = 1, 2, 3$, diperoleh

$$\Delta^n f(x) = a n! h^n$$

dan

$$\Delta^{n+1} f(x) = 0.$$

Apakah kesimpulan ini benar untuk $n > 3$? Misal diberikan fungsi $f(x)$ dari polinom derajat $\leq n$,

$$f(x) = a_0 + a_1x + a_2x^2 + \dots + a_nx^n$$

dan h adalah jarak antara nilai-nilai x . Selisih orde pertama adalah

$$\begin{aligned} \Delta f(x) &= f(x+h) - f(x) \\ &= \{a_0 + a_1(x+h) + \dots + a_n(x+h)^n\} - \{a_0 + a_1x + \dots + a_nx^n\} \\ &= a_n[(x+h)^n - x^n] + a_{n-1}[(x+h)^{n-1} - x^{n-1}] + \text{suku-suku derajat } \leq n-2 \\ &= a_n[(x^n + nhx^{n-1} + (n-1)x^{n-2}h^2 + \dots + h^n) - x^n] + \\ &\quad a_{n-1}[(x^{n-1} + (n-1)hx^{n-2} + (n-2)x^{n-3}h^2 + \dots + h^{n-1}) - x^{n-1}] + \\ &\quad \text{suku-suku derajat } \leq n-2 \\ &= nha_nx^{n-1} + \text{suku-suku derajat } \leq n-2 \end{aligned}$$

Dengan cara yang sama untuk $\Delta^2 f(x)$, $\Delta^3 f(x)$, ..., kita peroleh

$$\Delta f(x) = nha_n x^{n-1}$$

$$\Delta^2 f(x) = n(n-1) h^2 a_n x^{n-2}$$

$$\Delta^3 f(x) = n(n-1)(n-2) h^3 a_n x^{n-3}$$

...

$$\Delta^n f(x) = n! h^n a_n x^{n-n} = n! h^n a_n = n(n-1)(n-2)\dots(2)(1) h^n a_n x^{n-n}$$

$$\Delta^{n+1} f(x) = 0$$

Jadi kesimpulan kita benar. Apakah kegunaan kesimpulan ini? Bila di dalam tabel selisih ditemukan Δ^k bernilai (hampir) konstan ($\neq 0$) maka polinom yang tepat menginterpolasi titik-titik itu adalah polinom berderajat k . Pada contoh tabel (iii) di atas: Δ^3 konstan, jadi titik-titiknya tepat diinterpolasi dengan polinom derajat tiga (sama dengan fungsi aslinya, $f(x) = x^3$)

Bagaimanakah jika tidak terdapat Δ yang bernilai tetap? Misalnya diberikan tabel selisih di bawah ini:

x	$f(x) = 1/x$	Δf	$\Delta^2 f$	$\Delta^3 f$	$\Delta^4 f$
0.10	10.00	-5.00	3.33	-2.49	1.98
0.20	5.00	-1.67	0.83	-0.51	0.35
0.30	3.33	-0.83	0.33	-0.16	
0.40	2.50	-0.50	0.17		
0.50	2.00	-0.33			
0.60	1.67				

Pada tabel selisih di atas, tidak ada Δ^k yang mendekati nilai tetap. Jadi $f(x) = 1/x$ tidak tepat dihampiri dengan polinom derajat 1, 2, 3, atau 4 di dalam selang [0.10, 0.60]. Tetapi jika selang datanya diperkecil dengan pengambilan h yang lebih kecil dan digunakan empat angka bena sebagai berikut:

x	$f(x) = 1/x$	Δf	$\Delta^2 f$	$\Delta^3 f$
0.25	4.000	-0.154	0.012	-0.003
0.26	3.846	-0.142	0.009	0.001
0.27	3.704	-0.133	0.010	-0.002
0.28	3.571	-0.123	0.008	
0.29	3.448	-0.115		
0.30	3.333			

maka dari tabel ini ditemukan Δ^2 mendekati nilai tetap yaitu sekitar 0.010. Karena itu $f(x) = 1/x$ dapat dihampiri sebanyak empat angka benu dengan polinom kuadratik di dalam selang [0.25, 0.30].

Kesimpulan:

Tabel selisih bermanfaat untuk menentukan

1. Derajat polinom interpolasi
2. Selang data
3. Ketelitian yang diinginkan.

5.6.2 Polinom Interpolasi Newton-Gregory Mundur

Polinom Newton-Gregory mundur (*Newton-Gregory backward*) dibentuk dari tabel selisih mundur. Polinom ini sering digunakan pada perhitungan nilai turunan (*derivative*) secara numerik. Titik-titik yang digunakan berjarak sama, yaitu

$$x_0, x_{-1}, x_{-2}, \dots, x_{-n},$$

yang dalam hal ini,

$$x_i = x_0 + ih \quad , i = 0, -1, -2, \dots, -n$$

dan nilai x yang diinterpolasikan adalah

$$x = x_0 + sh \quad , s \in R$$

Sebagai contoh, tabel selisih mundur untuk 4 titik diperlihatkan oleh tabel berikut:

i	x_i	$f(x)$	∇f	$\nabla^2 f$	$\nabla^3 f$
-3	x_{-3}	f_{-3}			
-2	x_{-2}	f_{-2}	∇f_{-2}		
-1	x_{-1}	f_{-1}	∇f_{-1}	$\nabla^2 f_{-1}$	
0	x_0	f_0	∇f_0	$\nabla^2 f_0$	$\nabla^3 f_0$

Keterangan:

$$\begin{aligned} f_0 &= f(x_0) \\ f_{-1} &= f(x_{-1}) \\ \nabla f_0 &= f_0 - f_{-1} \\ \nabla f_{-1} &= \nabla f_{-1} - \nabla f_{-2} \\ \nabla^2 f_0 &= \nabla f_0 - \nabla f_{-1} \\ \nabla^{k+1} f_i &= \nabla^k f_i - \nabla^k f_{i-1} \end{aligned}$$

Polinom Newton-Gregory mundur yang menginterpolasi $(n+1)$ titik data adalah

$$\begin{aligned}
 f(x) \sim p_n(x) &= \sum_{k=0}^n \binom{s+k-1}{s} \nabla^k f_0 \\
 &= f_0 + \frac{s \nabla f_0}{1!} + \frac{s(s+1) \nabla^2 f_0}{2!} + \dots + \frac{s(s+1)(s+2)\dots(s+n-1) \nabla^n f_0}{n!}
 \end{aligned}
 \tag{P.5.55}$$

Mengenai penurunan rumus Newton-Gregory mundur, ditinggalkan kepada anda sebagai latihan.

Contoh 5.13

Diberikan 4 buah titik data dalam tabel berikut. Hitunglah $f(1.72)$ dengan

- (a) polinom Newton-Gregory maju derajat 3
- (b) polinom Newton-Gregory mundur derajat 3

Misalkan jumlah angka bena yang digunakan adalah 7 digit.

Penyelesaian:

- (a) Polinom Newton-Gregory maju derajat 3

i	x_i	$f(x_i)$	Δf	$\Delta^2 f$	$\Delta^3 f$
0	1.7	0.3979849	-0.0579985	-0.0001693	0.0004093
1	1.8	0.3399864	-0.0581678	0.0002400	
2	1.9	0.2818186	-0.0579278		
3	2.0	0.2238908			

$$s = (x - x_0)/h = (1.72 - 1.70)/0.1 = 0.2$$

Perkiraan nilai $f(1.72)$ adalah

$$\begin{aligned}
 f(1.72) \approx p_3(1.72) &= 0.3979849 + 0.2(-0.0579985) + \frac{0.2(-0.8)}{2} (-0.0001693) \\
 &\quad + \frac{0.2(-0.8)(-1.8)}{6} (0.0004093) \\
 &= 0.3979849 - 0.0115997 + 0.0000135 + 0.0000196 \\
 &= 0.3864183
 \end{aligned}$$

(nilai sejati $f(1.72) = 0.3864185$, jadi $p_3(1.72)$ tepat sampai 6 angka bena)

(b) Polinom Newton-Gregory maju derajat 3

i	x_i	$f(x_i)$	∇	∇^2	∇^3
-3	1.7	0.3979849			
-2	1.8	0.3399864	-0.0579985		
-1	1.9	0.2818186	-0.0581678	-0.0001693	
0	2.0	0.2238908	-0.0579278	0.0002400	0.0004093

Tabel di atas memperlihatkan bahwa tabel selisih mundur sama dengan tabel selisih maju, yang berbeda hanya notasi dan penempatan elemennya.

$$s = (x - x_0)/h = (1.72 - 2.0)/0.1 = -2.8$$

Perkiraan nilai $f(1.72)$ adalah

$$\begin{aligned} f(1.72) \approx p_3(1.72) &= 0.2238908 - 2.8(-0.0579278) + \frac{(-2.8)(-1.8)}{2} (0.0002400) \\ &\quad + \frac{(-2.8)(-1.8)(-0.8)}{6} (0.0004093) \\ &= 0.2238908 + 0.1621978 + 0.0006048 - 0.0002750 \\ &= 0.3864183 \end{aligned}$$

■

Contoh 5.13 memperlihatkan bahwa penyelesaian dengan Newton-Gregory maju atau mundur menghasilkan jawaban yang sama.

5.7 Ekstrapolasi

Pada awal bab sudah disinggung bahwa ekstrapolasi adalah penaksiran nilai $f(x)$ untuk x yang terletak di luar selang titik data. Dari pembahasan galat interpolasi sudah diketahui bahwa galat interpolasi semakin besar pada titik-titik yang jauh dari titik tengah selang. Dengan demikian, penaksiran nilai fungsi di luar selang menghasilkan galat ekstrapolasi yang sangat besar.

5.8 Interpolasi Dwimatra

Adakalanya kita membutuhkan perkiraan nilai fungsi dengan dua peubah. Fungsi dengan dua peubah, x dan y , secara umum dinyatakan sebagai

$$z = f(x, y)$$

Grafik fungsi z adalah berupa permukaan (*surface*) atau selimut kurva dengan alasnya adalah bidang x - y . Jadi, nilai-nilai z terletak pada permukaan tersebut.

Jika z dinterpolasi dengan polinom dua-peubah (interpolasi dwimatra atau dua-dimensi), kita harus menentukan berapa derajat dalam arah- x dan berapa derajat dalam arah- y . Misalnya z dihampiri dengan polinom dua-peubah, yang dalam hal ini derajat 2 dalam arah- x dan derajat 3 dalam arah- y :

$$\begin{aligned} z = f(x, y) \approx & a_0 + a_1 x + a_2 y + a_3 x^2 + a_4 xy + a_5 y^2 + a_6 x^2 y + a_7 x y^2 \\ & + a_8 x y^3 + a_9 y^3 + a_{10} x^2 y^2 + a_{11} x^2 y^3 \end{aligned} \quad (\text{P.5.56})$$

Interpolasi polinom dua-peubah dilakukan dalam dua arah: dalam arah x dan dalam arah- y . Pada setiap arah, kita harus memilih peubah yang dipegang konstan. Dalam arah- y , nilai x dipegang konstan, begitu juga dalam arah x , nilai y dipegang konstan (pemilihan arah mana yang dikerjakan terlebih dahulu memberikan jawaban yang sama). Semua metode interpolasi yang telah dibahas sebelum ini dapat digunakan untuk menginterpolasi polinom dua-peubah.

Contoh 5.14

[MAT92] Diberikan tabel $f(x, y)$ sebagai berikut:

x	y	0.1	0.2	0.3	0.4	0.5	0.6
0.5		0.165	0.428	0.687	0.942	1.190	1.431
1.0		0.271	0.640	1.003	1.359	1.703	2.035
1.5		0.447	0.990	1.524	2.045	2.549	3.031
2.0		0.738	1.568	2.384	3.177	3.943	4.672
2.5		1.216	2.520	3.800	5.044	6.241	7.379
3.0		2.005	4.090	6.136	8.122	10.030	11.841
3.5		3.306	6.679	9.986	13.196	16.277	19.198

Perkirakan nilai $f(1.6, 0.33)$ dengan polinom derajat 2 dalam arah- x dan derajat 3 dalam arah- y .

Penyelesaian:

Kita menggunakan polinom Netwon-Gregory maju untuk interpolasi dalam arah- x dan dalam arah y , karena titik-titiknya berjarak sama. Karena dalam arah- x menggunakan

interpolasi derajat 2, maka kita memilih tiga buah titik di tabel yaitu pada $x = 1.0, 1.5$, dan 2.0 karena $x = 1.6$ terletak paling dekat dengan pertengahan selang $[1.0, 2.0]$. Dalam arah- y , kita memilih empat buah titik (interpolasi derajat 3), yaitu pada $y = 0.2, 0.3, 0.4$, dan 0.5 karena $y = 0.33$ terletak paling dekat dengan pertengahan selang $[0.2, 0.5]$.

Dalam arah- y (x tetap):

	y	z	Δz	$\Delta^2 z$	$\Delta^3 z$
$x = 1.0$	0.2	0.640	0.363	-0.007	-0.005
	0.3	1.003	0.356	-0.012	
	0.4	1.359	0.344		
	0.5	1.703			
$x = 1.5$	0.2	0.990	0.534	-0.013	-0.004
	0.3	1.524	0.521	-0.017	
	0.4	2.045	0.504		
	0.5	2.549			
$x = 2.0$	0.2	1.5680	0.816	-0.023	-0.004
	0.3	2.384	0.793	-0.027	
	0.4	3.177	0.766		
	0.5	3.943			

Jarak antar titik dalam arah- y :

$$h = 0.1$$

dan

$$y = y_0 + sh \rightarrow s = \frac{y - y_0}{h} = \frac{0.33 - 0.2}{0.1} = 1.3$$

Polinom Newton-Gregory maju derajat tiga (dalam arah- y):

$$p_3(y) \approx f_0 + \frac{s}{1!} \Delta f_0 + \frac{s(s-1)}{2!} \Delta^2 f_0 + \frac{s(s-1)(s-2)}{3!} \Delta^3 f_0$$

Untuk $x = 1.0$; $f(x, 0.33) \approx p_3(x, 0.33)$

$$\begin{aligned} p_3(x, 0.33) &\approx 0.640 + \frac{1.3}{1}(0.363) + \frac{(1.3)(1.3-1)}{2}(-0.007) + \frac{(1.3)(1.3-1)(1.3-2)}{6}(-0.005) \\ &= 1.1108 \end{aligned}$$

Untuk $x = 1.5$; $f(x, 0.33) \approx p_3(x, 0.33)$

$$p_3(x, 0.33) \approx 0.990 + \frac{1.3}{1}(0.534) + \frac{(1.3)(1.3-1)}{2}(-0.013) + \frac{(1.3)(1.3-1)(1.3-2)}{6}(-0.004)$$

$$= 1.6818$$

Untuk $x = 2.0$; $f(x, 0.33) \approx p_3(x, 0.33)$

$$p_3(x, 0.33) \approx 1.568 + \frac{1.3}{1}(0.816) + \frac{(1.3)(1.3-1)}{2}(-0.023) + \frac{(1.3)(1.3-1)(1.3-2)}{6}(-0.004)$$

$$= 2.6245$$

Dalam arah-x (y tetap):

x	z	Δz	$\Delta^2 z$
$y = 0.33$	1.0	1.1108	0.5710
	1.5	1.6818	0.9427
	2.0	2.6245	

Jarak antar titik dalam arah-x:

$$h = 0.5$$

dan

$$x = x_0 + sh \rightarrow s = \frac{x - x_0}{h} = \frac{1.6 - 1.0}{0.5} = 1.2$$

Polinom Newton-Gregory maju derajat dua (dalam arah-x):

$$p_3(x) \approx f_0 + \frac{s}{1!} \Delta f_0 + \frac{s(s-1)}{2!} \Delta^2 f_0$$

$$f(1.6, 0.33) \approx p_3(1.6, 0.33) \approx 1.1108 + \frac{1.2}{1}(0.5710) + \frac{(1.2)(1.2-1)}{2}(0.3717)$$

$$= 1.8406$$

Jadi, $f(1.6, 0.33) \approx 1.8406$ (jika dibulatkan ke dalam 4 angka bena adalah 1.841). Tabel di atas diambil dari fungsi $f(x, y) = e^x \sin y + y - 0.1$, yang mana nilai sejati $f(1.6, 0.33) = 1.8350$. Galat interpolasi adalah -0.006 . Galat ini dapat dikurangi jika kita melakukan interpolasi derajat 2 dalam arah-y karena $\Delta^2 y$ kecil dan interpolasi derajat 3 dalam arah-x.

5.9 Contoh Soal Tera pan Inter polasi

Konsentrasi larutan oksigen jenuh dalam air sebagai fungsi suhu dan konsentrasi klorida diberikan dalam bentuk tabel berikut [CHA91]:

Suhu, $^{\circ}\text{C}$	Konsentrasi larutan Oksigen (mg/L) untuk berbagai konsentrasi klorida	
	Klorida = 10 mg/L	Klorida = 20 mg /L
5	11.6	10.5
10	10.3	9.2
15	9.1	8.2
20	8.2	7.4
25	7.4	6.7
30	6.8	6.1

Dengan mengandaikan bahwa data pada tabel berketelitian cukup tinggi, pakailah metode interpolasi untuk menaksir konsentrasi oksigen yang larut untuk $T = 22.4 \text{ } ^{\circ}\text{C}$ pada konsentrasi klorida 10 mg/L dan 20mg/L. Gunakan metode interpolasi Lagrange.

Penyelesaian:

Konsentrasi Klorida = 10 mg/L

<u>T</u>	<u>$C(T)$</u>
5	11.6
10	10.3
15	9.1
20	8.2
25	7.4
30	6.8

Bila digunakan keenam titik data itu, maka polinom interpolasinya adalah polinom Lagrange derajat lima.

$$\begin{aligned}
 p_5(22.4) &= (11.6)L_0(22.4) + (10.3)L_1(22.4) + (9.1)L_2(22.4) + \\
 &\quad (8.2)L_3(22.4) + 7.4)L_4(22.4) + (6.8)L_5(22.4) \\
 &= 7.8125049876 \text{ mg/L}
 \end{aligned}$$

Konsentrasi Klorida = 20 mg/L

T	$C(T)$
5	10.5
10	9.2
15	8.2
20	7.4
25	6.7
30	6.1

Polinom interpolasi Lagrange:

$$\begin{aligned} p_5(22.4) &= (10.5)L_0(22.4) + (9.2)L_1(22.4) + (8.2)L_2(22.4) + \\ &\quad (7.4)L_3(22.4) + (6.7)L_4(22.4) + (6.1)L_5(22.4) \\ &= 7.0550200177 \text{ mg/L} \end{aligned}$$

Bagian II: Regresi

5.10 Regresi

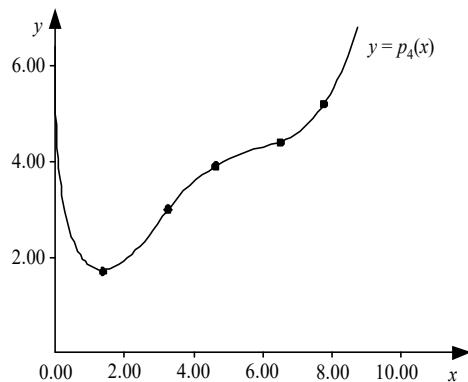
Pada bagian awal bab ini sudah dijelaskan bahwa regresi adalah teknik pencocokan kurva untuk data yang berketelitian rendah. Contoh data yang berketelitian rendah data hasil pengamatan, percobaan di laboratorium, atau data statistik. Data seperti itu kita sebut **data hasil pengukuran**. Galat yang dikandung data berasal dari ketidaktelitian alat ukur yang dipakai, kesalahan membaca alat ukur (paralaks), atau karena kelakuan sistem yang diukur.

Untuk data hasil pengukuran, pencocokan kurva berarti membuat fungsi mengampiri (*approximate*) titik-titik data. Kurva fungsi hampiran tidak perlu melalui semua titik data tetapi dekat dengannya tanpa perlu menggunakan polinom berderajat tinggi.

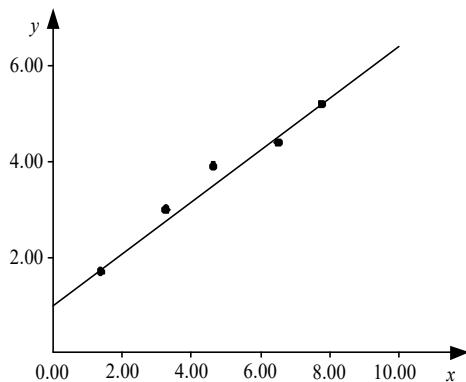
Sebagai contoh ilustrasi, diberikan data jarak tempuh (y) sebuah kendaraan -dalam mil- setelah x bulan seperti pada tabel di bawah ini.

x	1.38	3.39	4.75	6.56	7.76
y	1.83	2.51	3.65	4.10	5.01

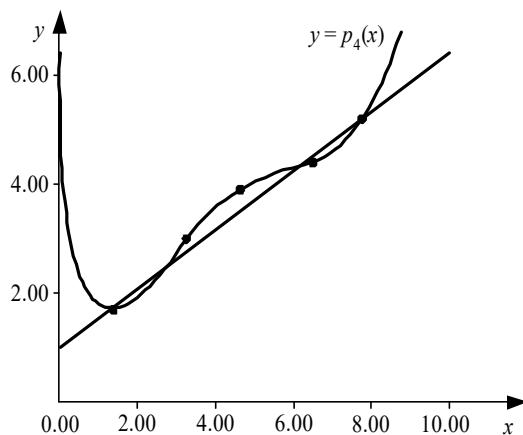
Data di dalam tabel dicocokkan dengan polinom Lagrange (Gambar 5.9(a)), dan dengan fungsi hampiran lanjar (Gambar 5.9(b)). Perbandingan keduanya diperlihatkan pada Gambar 5.9(c).



(a)



(b)



(c)

Gambar 5.9

- (a) Data dicocokkan dengan polinom Lagrange derajat 4
- (b) Data dicocokkan dengan garis lurus
- (c) Perbandingan kedua kurva

Dari kedua pencocokan tersebut, terlihat bahwa garis lurus memberikan hampiran yang *bagus*, tetapi belum tentu yang *terbaik*. Pengertian terbaik di sini bergantung pada cara kita mengukur galat hampiran.

Prinsip penting yang harus diketahui dalam mencocokkan kurva untuk data hasil pengukuran adalah:

1. Fungsi mengandung sesedikit mungkin parameter bebas
2. Deviasi fungsi dengan titik data dibuat minimum.

Kedua prinsip di atas mendasari metode **regresi kuadrat terkecil**. Perbedaan antara metode regresi kuadrat terkecil dengan metode interpolasi polinom adalah:

Regresi kuadrat terkecil	Interpolasi polinom
1. Data berasal dari hasil pengukuran	1. Data berasal dari fungsi yang ingin disederhanakan dengan polinom, dari tabel di literatur, atau dari hasil pengukuran.
2. Data berketelitian rendah (mengandung galat)	2. Data berketelitian tinggi
3. Fungsi kuadrat terkecil tidak perlu melalui setiap titik data. Kurva fungsinya dirancang mengikuti pola titik-titik sebagai suatu kelompok.	3. Fungsi polinom interpolasi harus melalui semua titik data. Semakin banyak datanya, semakin tinggi derajat polinom, dan semakin besar galat pembulatannya
4. Data tidak harus terurut	4. Data harus terurut

Manfaat pencocokan kurva untuk data hasil pengukuran:

1. Bagi ahli sains/rekayasa: mengembangkan formula empirik untuk sistem yang diteliti.
2. Bagi ahli ekonomi: menentukan kurva kecenderungan ekonomi untuk “meramalkan” kecenderungan masa depan.

Teknik regresi yang dibahas di sini hanya regresi lanjar, yaitu pencocokan kurva untuk data yang memiliki hubungan lanjar antara peubah bebas dan peubah terikatnya. Selain regresi lanjar, ada teknik regresi lain, yaitu regresi polinom, regresi ganda, dan regresi nirlanjarn. Mahasiswa dapat mempelajari ketiga teknik regresi yang disebutkan terakhir ini pada buku [CHA91].

5.10.1 Regresi Lanjar

Misalkan (x_i, y_i) adalah data hasil pengukuran. Kita akan menghampiri titik-titik tersebut dengan sebuah garis lurus (Gambar 5.10). Garis lurus tersebut dibuat sedemikian sehingga galatnya sekecil mungkin dengan titik-titik data.

Karena data mengandung galat, maka nilai data sebenarnya, $g(x_i)$, dapat ditulis sebagai

$$g(x_i) = y_i + e_i \quad i = 1, 2, \dots, n \quad (\text{P.5.57})$$

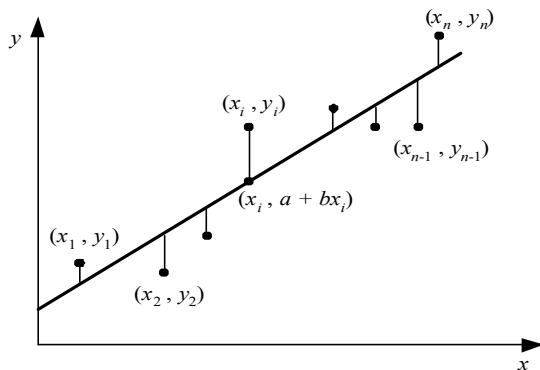
yang dalam hal ini, e_i adalah galat setiap data. Diinginkan fungsi lanjar

$$f(x) = a + bx \quad (\text{P.5.58})$$

yang mencocokkan data sedemikian sehingga deviasinya,

$$r_i = y_i - f(x_i) = y_i - (a + bx_i) \quad (\text{P.5.59})$$

minimum.



Gambar 5.10 Regresi lanjar

Total kuadrat deviasi persamaan (P.5.59) adalah

$$R = \sum_{i=1}^n r_i^2 = \sum_{i=1}^n (y_i - a - bx_i)^2 \quad (\text{P.5.60})$$

Agar R minimum, maka haruslah²

$$\frac{\partial R}{\partial a} = -2 \sum (y_i - a - bx_i) = 0 \quad (\text{P.5.61})$$

$$\frac{\partial R}{\partial b} = -2 \sum x_i(y_i - a - bx_i) = 0 \quad (\text{P.5.62})$$

Penyelesaian:

Masing-masing ruas kedua persamaan dibagi dengan -2:

$$\begin{aligned} \sum (y_i - a - bx_i) &= 0 & \Rightarrow \sum y_i - \sum a - \sum bx_i &= 0 \\ \sum x_i(y_i - a - bx_i) &= 0 & \Rightarrow \sum x_i y_i - \sum ax_i - \sum bx_i^2 &= 0 \end{aligned}$$

Selanjutnya,

$$\begin{aligned} \sum a + \sum bx_i &= \sum y_i \\ \sum ax_i + \sum bx_i^2 &= \sum x_i y_i \end{aligned}$$

atau

$$\begin{aligned} na + b \sum x_i &= \sum y_i \\ a \sum x_i + b \sum x_i^2 &= \sum x_i y_i \end{aligned} \quad (\text{P.5.63}) \quad (\text{P.5.64})$$

Kedua persamaan terakhir ini dinamakan **persamaan normal**, dan dapat ditulis dalam bentuk persamaan matriks:

$$\begin{bmatrix} n & \sum x_i \\ \sum x_i & \sum x_i^2 \end{bmatrix} = \begin{bmatrix} a \\ b \end{bmatrix} \begin{bmatrix} \sum y_i \\ \sum x_i y_i \end{bmatrix}$$

Solusinya, a dan b , dapat diselesaikan dengan metode eliminasi Gauss atau aturan Cramer. Karena data mengandung galat, maka persamaan normal sering berkondisi buruk (*ill-conditioning*). Nilai a dan b juga dapat dicari dengan mengutakatik kedua buah persamaan normal menjadi:

² Untuk selanjutnya, notasi $\sum_{i=1}^n$ ditulis “ Σ ” saja.

$$b = \frac{n \sum x_i y_i - \sum x_i \sum y_i}{n \sum x_i^2 - (\sum x_i)^2} \quad (\text{P.5.65})$$

$$a = y - bx \quad (\text{P.5.66})$$

yang dalam hal ini, y dan x masing-masing adalah nilai rata-rata x dan y .

Untuk menentukan seberapa bagus fungsi hampiran mencocokkan data, kita dapat mengukurnya dengan **galat RMS** (*Root-mean-square error*):

$$E_{RMS} = \left(\frac{1}{n} \sum_{i=1}^n |f(x_i) - y_i|^2 \right)^{\frac{1}{2}} \quad (\text{P.5.67})$$

Semakin kecil nilai E_{RMS} semakin bagus fungsi hampiran mencocokkan titik-titik data.

Contoh 5.15

[NAK93] Tentukan persamaan garis lurus yang mencocokkan data pada tabel di bawah ini. Kemudian, perkiraan nilai y untuk $x = 1.0$.

Penyelesaian:

i	x_i	y_i	x_i^2	$x_i y_i$
1	0.1	0.61	0.01	0.061
2	0.4	0.92	0.16	0.368
3	0.5	0.99	0.25	0.495
4	0.7	1.52	0.49	1.064
5	0.7	1.47	0.49	1.029
6	0.9	2.03	0.81	1.827
	$\sum x_i = 3.3$	$\sum y_i = 7.54$	$\sum x_i^2 = 2.21$	$\sum x_i y_i = 4.844$

Diperoleh sistem persamaan lanjar:

$$\begin{bmatrix} 6 & 3.3 \\ 3.3 & 2.21 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} 7.54 \\ 4.844 \end{bmatrix}$$

Solusi SPL di atas adalah: $a = 0.2862$
 $b = 1.7645$

Persamaan garis regresinya adalah: $f(x) = 0.2862 + 1.7645x$.

Perbandingan antara nilai y_i dan $f(x_i)$:

i	x_i	y_i	$f(x_i) = a + bx_i$	deviasi	$(\text{deviasi})^2$
1	0.1	0.61	0.46261	0.147389	0.02172
2	0.4	0.92	0.99198	-0.07198	0.00518
3	0.5	0.99	1.16843	-0.17844	0.03184
4	0.7	1.52	1.52135	-0.00135	0.00000
5	0.7	1.47	1.52135	-0.05135	0.00264
6	0.9	2.03	1.87426	0.15574	0.02425
					$\Sigma = 0.08563$

Taksiran nilai y untuk $x = 1.0$ adalah

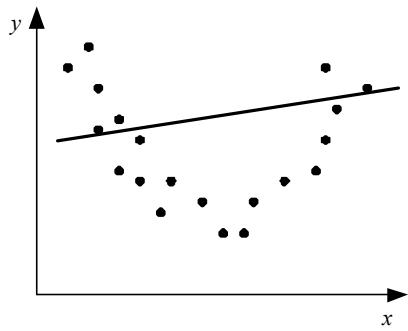
$$y = f(1.0) = 0.2862 + 1.7645(1.0) = 2.0507$$

$$\text{Galat RMS adalah } E_{RMS} = \left(\frac{0.08563}{6}\right)^{1/2} = 0.119464$$
■

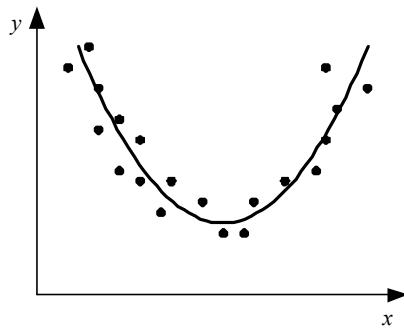
5.10.2 Pelajaran

Regrasi lanjar hanya tepat bila data memiliki hubungan lanjar antara peubah bebas dan peubah terikatnya. Gambar 5.11 memperlihatkan bahwa garis lurus tidak tepat mewakili kecenderungan titi-titik data, dengan kata lain pada kasus ini hubungan x dengan y tidak lanjar. Sebaliknya, fungsi kuadratik lebih tepat menghampiri titik-titik tersebut.

Langkah pertama dalam analisis regresi seharusnya berupa penggambaran titik-titik data pada diagram kartesian dan secara visual memeriksa data untuk memastikan apakah berlaku suatu model lanjar atau model nirlanjar. Penggambaran titik-titik ini sekaligus juga sangat membantu dalam mengetahui fungsi yang tepat untuk mencocokkan data.



(a)



(b)

Gambar 5.11 (a) Data yang tidak cocok untuk regresi lanjar;
 (b) Petunjuk bahwa parabola lebih disenangi [CHA91]

Meskipun fungsi hampiran berbentuk nirlanjar, namun pencocokan kurva dengan fungsi nirlanjar tersebut dapat juga diselesaikan dengan cara regresi lanjar. Misalnya tiga macam fungsi nirlanjar di bawah ini:

1. Persamaan pangkat sederhana

$$y = Cx^b, \text{ } C \text{ dan } b \text{ konstanta.}$$

2. Model eksponensial

$$y = Ce^{bx}, \text{ } C \text{ dan } b \text{ konstanta.}$$

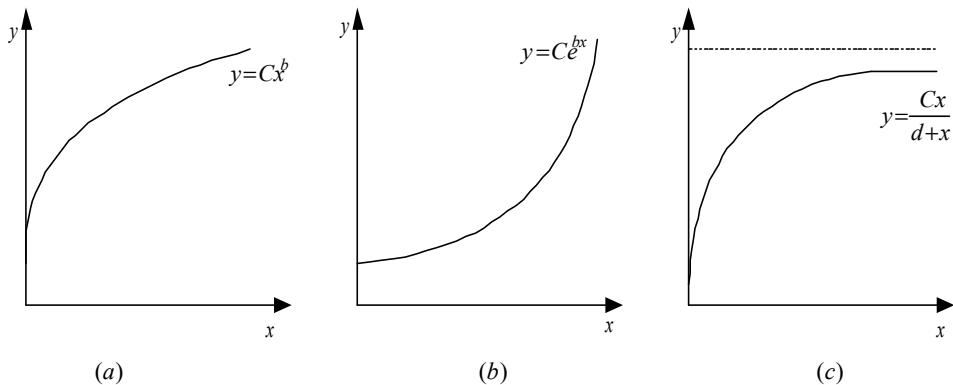
Contoh: - model pertumbuhan populasi
 - model peluruhan zat radioaktif

3. Persamaan laju pertumbuhan jenuh (*saturation growth-rate*)

$$y = \frac{Cx}{d+x}, \text{ } C \text{ dan } d \text{ konstanta.}$$

Contoh: model pertumbuhan bakteri kondisi pembatas
 (misalnya dibatasi oleh jumlah makanan)

Sketsa kurva untuk ketiga macam fungsi nirlanjar di atas diperlihatkan pada Gambar 5.12 berikut ini.



Gambar 5.12 Sketsa kurva (a) $y = Cx^b$ ($C > 0, b > 0$), (b) $y = Ce^{bx}$ ($C > 0, b > 0$), dan
(c) $y = Cx/(d + x)$ ($C > 0, d > 0$)

Pelajaran Persamaan Pangkat Sederhana

Misalkan kita akan mencocokkan data dengan fungsi

$$y = Cx^b \quad (\text{P.5.68})$$

Lakukan pelajaran sebagai berikut:

$$y = Cx^b \quad \Leftrightarrow \quad \ln(y) = \ln(C) + b \ln(x)$$

Definisikan

$$\begin{aligned} Y &= \ln(y) \\ a &= \ln(C) \\ X &= \ln(x) \end{aligned}$$

Persamaan regresi lanjarnya adalah:

$$Y = a + bX$$

Lakukan pengubahan dari (x_i, y_i) menjadi $(\ln(x_i), \ln(y_i))$, lalu hitung a dan b dengan cara regresi lanjar. Dari persamaan $a = \ln(C)$, kita dapat menghitung nilai

$$C = e^a$$

Sulihkan nilai b dan C ke dalam persamaan pangkat $y = Cx^b$.

Contoh 5.16

Cocokkan data berikut dengan fungsi $y = Cx^b$.

Penyelesaian:

i	x_i	y_i	$X_i = \ln(x_i)$	$Y_i = \ln(y_i)$	X_i^2	$X_i Y_i$
1	0.1500	4.4964	-1.8971	1.5033	3.5990	-2.8519
2	0.4000	5.1284	-0.9163	1.6348	0.8396	-1.4980
3	0.6000	5.6931	-0.5108	1.7393	0.2609	-0.8884
4	1.0100	6.2884	0.0100	1.8387	0.0001	0.0184
5	1.5000	7.0989	0.4055	1.9599	0.1644	0.7947
6	2.2000	7.5507	0.7885	2.0216	0.6217	1.5940
7	2.4000	7.5106	0.8755	2.0163	0.7665	1.7653
			$\sum X_i = -1.2447$	$\sum Y_i = 12.7139$	$\sum X_i^2 = 6.2522$	$\sum X_i Y_i = -1.0659$

Diperoleh sistem persamaan lanjar

$$\begin{bmatrix} 7 & -1.2447 \\ -1.2447 & 6.2522 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} 12.7139 \\ -1.0659 \end{bmatrix}$$

Solusi SPL di atas: $a = 1.8515$ dan $b = 0.1981$.

Hitung $C = e^a = e^{1.8515} = 6.369366$

Jadi, titik-titik (x, y) pada tabel di atas dihampiri dengan fungsi pangkat sederhana:

$$y = 6.369366 x^{0.1981}$$

■

Pelajaran Model Eksponensial $y = Ce^{bx}$

Misalkan kita akan mencocokkan data dengan fungsi

$$y = Ce^{bx} \quad (\text{P.5.69})$$

Lakukan pelajaran sebagai berikut:

$$\begin{aligned} y &= Ce^{bx} \\ \Leftrightarrow \ln(y) &= \ln(C) + bx \ln(e) \\ \Leftrightarrow \ln(y) &= \ln(C) + bx \quad (\ln(e) = 1) \end{aligned}$$

Definisikan

$$\begin{aligned}Y &= \ln(y) \\a &= \ln(C) \\X &= x\end{aligned}$$

Persamaan regresi lanjarnya:

$$Y = a + bX$$

Lakukan pengubahan dari (x_i, y_i) menjadi $(x_i, \ln(y_i))$, lalu hitung a dan b dengan cara regresi lanjar.

Dari persamaan $a = \ln(C)$, kita dapat menghitung nilai $C = e^a$. Sulihkan nilai b dan C ke dalam persamaan eksponensial $y = Ce^{bx}$.

Pelanjaran Model Laju Pertumbuhan Jenuh $y = \frac{Cx}{d+x}$

Misalkan kita akan mencocokkan data dengan fungsi

$$y = \frac{Cx}{d+x} \quad (\text{P.5.70})$$

Lakukan pelanjaran sebagai berikut:

$$y = \frac{Cx}{d+x}$$

$$\frac{1}{y} = \frac{d}{C} \frac{1}{x} + \frac{1}{C}$$

Definisikan

$$\begin{aligned}Y &= 1/y \\a &= 1/C \\b &= d/C \\X &= 1/x\end{aligned}$$

Persamaan regresi lanjarnya:

$$Y = a + bX$$

Lakukan pengubahan dari (x_i, y_i) menjadi $(1/x_i, 1/y_i)$, lalu hitung a dan b dengan cara regresi lanjar.

Dari persamaan $a = 1/C$, kita dapat menghitung nilai $C = 1/a$.

Dari persamaan $b = d/C$, kita dapat menghitung $d = bC$.

Sulihkan d dan C ke dalam persamaan laju pertumbuhan jenuh $y = Cx/(d+x)$.

Tabel berikut merangkum beberapa fungsi dan pelajarannya [MAT92]:

Fungsi $y = f(x)$	Bentuk lanjar $y = a + bX$	Perubahan peubah dan kontanta
$y = Cx^b$	$\ln(y) = \ln(C) + b \ln(x)$	$Y = \ln(y), X = \ln(x),$ $C = e^a$
$y = Ce^{bx}$	$\ln(y) = \ln(C) + bx$	$Y = \ln(y), X = x, C = e^a$
$y = \frac{Cx}{d+x}$	$\frac{1}{y} = \frac{d}{C} \frac{1}{x} + \frac{1}{C}$	$Y = 1/y, X = 1/x$ $C = 1/a, d = bC$
$y = a + \frac{b}{x}$	$y = a + b \frac{1}{x}$	$Y = y, X = 1/x$
$y = \frac{D}{x+C}$	$y = \frac{D}{C} + \frac{-1}{C}(xy)$	$Y = y, X = xy,$ $C = \frac{-1}{b}, D = \frac{-a}{b}$
$y = \frac{1}{a+bx}$	$\frac{1}{y} = a + bX$	$Y = \frac{1}{y}, X = x$
$y = (a+bx)^{-2}$	$y^{-1/2} = a + bX$	$Y = y^{-1/2}, X = x$
$y = Cxe^{-Dx}$	$\ln\left(\frac{y}{x}\right) = \ln(C) + (-Dx)$	$Y = \ln\left(\frac{y}{x}\right), X = x$ $C = e^a, D = -b$

5.11 Contoh Penerapan Regresi dalam Bidang Rekayasa

Di bawah ini disajikan contoh penerapan regresi dalam bidang teknik kimia. Contoh ini dikutip dari buku [CHA91] dengan beberapa perubahan.

Model Populasi

Model pertumbuhan populasi, misalnya populasi bakteri, adalah penting dalam bidang rekayasa. Yang merupakan dasar terhadap banyak model tersebut adalah andaian bahwa laju perubahan populasi (dp/dt) adalah sebanding dengan populasi sebenarnya (p) pada sembarang waktu (t), atau dalam bentuk persamaan

$$\frac{dp}{dt} = kp \quad (\text{P.5.71})$$

yang dalam hal ini, k adalah kesebandingan yang disebut laju pertumbuhan yang spesifik dan mempunyai satuan waktu⁻¹. Jika k adalah tetapan, maka penyelesaian persamaan (P.5.71) adalah

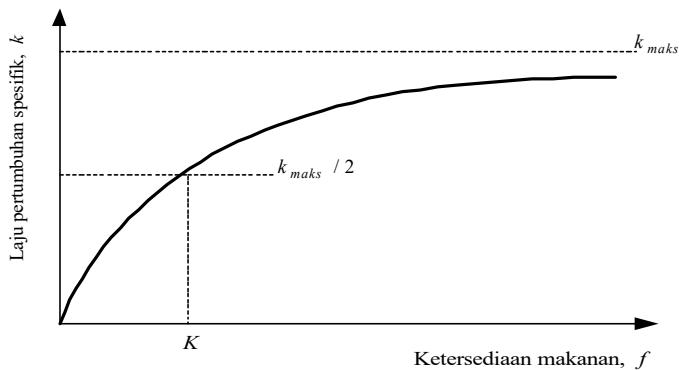
$$p(t) = p_0 e^{kt} \quad (\text{P.5.72})$$

yang dalam hal ini, p_0 adalah populasi pada saat $t = 0$. Terlihat bahwa $p(t)$ dalam persamaan (P.5.72) mendekati tak hingga begitu t menjadi besar. Perilaku ini jelas tidak mungkin untuk sistem yang nyata. Karena itu modelnya harus diubah untuk membuatnya lebih nyata.

Pertama, harus diketahui bahwa laju pertumbuhan yang khusus k tidak dapat berupa tetapan begitu populasi menjadi besar. Ini adalah kasusnya, karena begitu p mendekati tak hingga, organisme yang sedang dimodelkan akan menjadi dibatasi oleh faktor-faktor kekurangan makanan dan produksi sampah beracun. Satu cara untuk mengungkapkan ini secara matematis adalah memakai model laju pertumbuhan jenuh sedemikian sehingga

$$k = \frac{k_{\max}f}{K + f} \quad (\text{P.5.73})$$

dalam hal ini, k_{\max} adalah laju pertumbuhan yang dapat tercapai untuk nilai makanan besar (f) dan K adalah konstanta setengah jenuh. Gambar 5.12 memperlihatkan bahwa pada saat $f = K$, maka $k = k_{\max}/2$. Oleh karena itu, K adalah banyaknya makanan yang menunjang laju pertumbuhan populasi yang sama dengan setengah laju maksimum.



Gambar 5.12 Laju pertumbuhan populasi terhadap ketersediaan makanan

Konstanta K dan k_{maks} adalah nilai-nilai empiris yang didasarkan pada pengukuran k secara eksponensial untuk beragam nilai f . Sebagai contoh, andaikan populasi p menyatakan ragi yang digunakan dalam proses fermentasi alkohol dan f adalah konsentrasi sumber karbon yang harus difermentasikan. Pengukuran k terhadap f untuk ragi diperlihatkan pada tabel berikut:

f , mg/Liter	k , hari ⁻¹
7	0.29
9	0.37
15	0.48
25	0.65
40	0.80
75	0.97
100	0.99
150	1.07
160	1.18
190	1.36
200	1.82

Tabel Pengukuran k terhadap jumlah ketersediaan makanan

Anda diminta menghitung k_{maks} dan K untuk data empirik ini. Mula-mula lakukan pelajaran terhadap persamaan (P.5.73) menjadi

$$Y = a + bX \quad (\text{P.5.74})$$

Transformasikan data dalam bentuk lanjar yang ekivalen itu, lalu hitung A dan B menggunakan metode regresi lanjar. Selanjutnya anda dapat menghitung nilai k_{maks} dan K .

Dengan menyulihkan persamaan (P.5.73) kedalam persamaan (P.5.71) diperoleh

$$\frac{dp}{dt} = \frac{k_{maks}fp}{K + f} \quad (\text{P.5.75})$$

yang dalam hal ini, k_{maks} dan K adalah besaran yang telah ditemukan. Persamaan (P.5.75) ini selanjutnya dapat diselesaikan dengan menggunakan teori diferensial atau menggunakan metode numerik untuk persamaan diferensial biasa (PDB) bila f pada saat t diketahui. Dari persamaan (P.5.75) terlihat bahwa jika f mendekati nol karena p sangat besar, dp/dt mendekati nol dan populasi dikatakan stabil.

Penyelesaian:

Tinjau pencocokan kurva untuk fungsi laju pertumbuhan jenuh saja:

$$k = \frac{k_{maks}f}{(K + f)}$$

Lakukan pelajaran:

$$\frac{1}{k} = \frac{K + f}{k_{maks}f} = \frac{1}{k_{maks}} + \frac{K}{k_{maks}} \frac{1}{f}$$

Persamaan regresi lanjarnya adalah

$$Y = a + bX$$

dengan

$$Y = 1/k$$

$$a = 1/k_{maks}$$

$$b = K/k_{maks}$$

$$X = 1/f$$

Lakukan transformasi dari (f_i, k_i) menjadi $(1/f_i, 1/k_i)$:

$X_i = 1/f_i$	X_i^2	$Y_i = 1/k_i$	$X_i Y_i$
0.14285714	0.02040816	3.44827586	0.49261084
0.11111111	0.01234568	2.70270270	0.30030030
0.06666667	0.00444444	2.08333333	0.13888889
0.04000000	0.00160000	1.53846154	0.06153846
0.02500000	0.00062500	1.25000000	0.03125000
0.01333333	0.00017778	1.03092784	0.01374570
0.01000000	0.00010000	1.01010101	0.01010101
0.00666667	0.00004444	0.93457944	0.00623053
0.00625000	0.00003906	0.84745763	0.00529661
0.00526316	0.00002770	0.73529412	0.00386997
0.00500000	0.00002500	0.54945055	0.00274725
<hr/>			
$\Sigma = 0.43214808 \quad 0.03983727 \quad 16.13058402 \quad 1.06657956$			
<hr/>			

$$\begin{aligned} n &= 11 \\ \sum X_i &= \sum 1/f_i = 0.43214808 & \sum Y_i &= \sum 1/k_i = 16.13058402 \\ \sum X_i^2 &= 0.03983727 & \sum X_i Y_i &= 1.06657956 \end{aligned}$$

Diperoleh sistem persamaan lanjar:

$$\begin{bmatrix} 11 & 0.43214808 \\ 0.43214808 & 0.03983727 \end{bmatrix} \begin{bmatrix} A \\ B \end{bmatrix} = \begin{bmatrix} 16.13058402 \\ 1.06657956 \end{bmatrix}$$

Solusi SPL tersebut adalah

$$\begin{aligned} a &= 0.72249892 \\ b &= 18.93586082 \end{aligned}$$

Dari persamaan laju pertumbuhan jenuh:

$$k = (k_{maks} f) / (K + f)$$

diperoleh:

$$\begin{aligned} k_{maks} &= 1/a = 1.38408511 \\ K &= b k_{maks} = 26.20884299 \end{aligned}$$

Jadi,

$$k = \frac{1.38408511f}{26.20884299+f}$$

Jika kita mengerjakan apa yang bisa kita kerjakan,
maka Tuhan mengerjakan apa yang tidak bisa kita kerjakan.
(Majalah Intisari)

Soal Latihan

1. Diberikan pasangan nilai x dan $f(x)$, $f(x)$ tidak diketahui, sebagai berikut :

x	0.1	0.3	0.5	0.7	0.9	1.1	1.3
$f(x)$	0.003	0.067	0.148	0.248	0.370	0.518	0.697

- (a) Berapa derajat polinom yang dengan *tepat* melalui ketujuh titik data tersebut ?
- (b) Berapa derajat polinom yang *terbaik* untuk menginterpolasi ketujuh titik data tersebut ?
- (c) Dengan derajat terbaik yang anda nyatakan dalam jawaban (b), tentukan nilai fungsi di $x = 0.58$ dengan polinom interpolasi :
- (ii) *Lagrange*
 - (iii) *Newton*
 - (iv) *Newton-Gregory Maju*
 - (v) *Newton-Gregory Mundur*

Pilihlah titik-titik yang meminimumkan galat interpolasi.

- (d) Hitung taksiran galat dalam menghitung $f(0.8)$ dengan polinom (ii), (iii), (iv).
- (e) Dengan teknik interpolasi balikan, tentukan nilai x sehingga $f(x) = 0.200$
2. (a) Perlihatkan untuk interpolasi kubik bahwa :

$$|E_3(x)| = |f(x) - p_3(x)| \leq h^4/27 \max |f^{(4)}(c)|, \quad x_0 \leq c \leq x_3$$

yang dalam hal ini $p_3(x)$ adalah polinom interpolasi derajat 3 yang menginterpolasi

$$x_0 = 0, x_1 = h, x_2 = 2h, \text{ dan } x_3 = 3h.$$

- (b) Tentukan h (jarak antar titik) untuk titik-titik yang berjarak sama dari fungsi $f(x) = \sqrt{x}$ antara $x = 1$ sampai $x = 2$ sehingga interpolasi polinom kubik dalam daftar nilai itu mempunyai galat kurang dari 0.000001. Berapa jumlah titik data dengan h sebesar itu ?
3. Dengan menggunakan jumlah angka bena yang ditentukan, buatlah tabel selisih maju untuk fungsi $f(x) = \sqrt{x}$ pada
- (i) selang $[1.00, 1.06]$, $h = 0.01$, banyak angka bena = 5
 - (ii) selang $[1.00, 1.30]$, $h = 0.01$, banyak angka bena = 6
- Untuk masing-masing (i) dan (ii), dengan polinom derajat berapakah $f(x)$ diinterpolasi secara teliti?
4. Tabel $\tan(x)$, x dalam radian, mempunyai $h = 0.1$. Dekat $x = \pi/2$, fungsi tangen naik sangat cepat dan disini interpolasi lanjar tidak sangat teliti. Dengan polinom derajat berapa yang dibutuhkan untuk menginterpolasi $x = 1.506$ (4 angka bena) ?
5. Tentukan polinom yang menghampiri $f(x) = x/\sqrt{x}$ dalam selang $[2.0, 4.0]$, ambil $h = 0.5$, lima angka bena.
6. Apakah interpolasi Newton-Gregory mundur dapat ditulis dengan menggunakan tabel selisih maju? Jika dapat jelaskan bagaimana caranya.
7. Uraikan cara memperoleh rumus polinom Newton-Gregory Mundur !
8. Misalkan f adalah fungsi yang menerus dan dapat di-diferensialkan dalam $[-h, 2h]$, dan $f_i = f(ih)$. Dengan bantuan deret Taylor, tentukan tetapan a dan b yang meminimumkan galat interpolasi berikut :

$$f_{1/2} \approx a(f_0 + f_1) + b(f_{-1} + f_2)$$

Perlihatkan bahwa galatnya kira-kira $\alpha f_{1/2}^{(4)}$, dengan α bernilai tetap. Tentukan nilai α itu

9. Fungsi

$$f(x) = \frac{x^2 + \sqrt{x}}{e^x + x}$$

tampaknya rumit dan sukar di-integralkan secara analitik dalam selang [0,0.9]. Cara yang mudah adalah dengan menghampiri $f(x)$ dengan polinom interpolasi, misalnya dengan polinom derajat tiga ($p_3(x)$), yang dalam hal ini jarak antar titik adalah

$$h = (0.9 - 0)/3 = 0.3.$$

Selanjutnya

$$\int_0^{0.9} f(x) dx \approx \int_0^{0.9} p_3(x) dx$$

Hitunglah integral $f(x)$ di dalam selang [0,0.9] dengan cara tersebut. Polinom interpolasi yang digunakan terserah anda.

10. Diberikan titik-titik yang absisnya berjarak sama (h), yaitu $(0,f(0))$, $(h,f(h))$, $(2h,f(2h))$, dan $(3h,f(3h))$. Untuk $x = 3h/3$, perlihatkan bahwa nilai interpolasinya dengan polinom Lagrange derajat tiga adalah

$$p_3(3h/3) = -0.0625 \{ f(0) + f(3h) \} + 0.5625 \{ f(h) + f(2h) \}$$

11. Tentukan fungsi lanjar yang mencocokkan titik-titik data berikut dengan metode regresi:

x	1.0	1.5	2.0	2.5	3.0
y	2.0	3.2	4.1	4.9	5.9

12. Diberikan titik-titik (x, y) sebagai berikut:

x	1	2	3	4	5
y	0.6	0.9	4.3	7.6	12.6

- (a) Cocokan titik-titik di tabel masing-masing dengan fungsi $f(x) = Ce^{bx}$ dan $f(x) = Cx^b$
 (b) Hitung deviasi = $y_i - f(x_i)$, kemudian tentukan galat RMS nya. Berdasarkan galat RMS, fungsi hampiran mana yang terbaik?

Bab 6

Integrasi Numerik

Pelajarilah jagad raya ini. Jangan kecewa karena dunia
tidak mengenal anda, tetapi kecewalah karena
anda tidak mengenal dunia.
(Kong Fu Tse - filusuf China)

Di dalam kalkulus, integral adalah satu dari dua pokok bahasan yang mendasar disamping turunan (*derivative*). Dalam kuliah kalkulus integral, anda telah diajarkan cara memperoleh solusi analitik (dan eksak) dari **integral Tak-tentu** maupun **integral Tentu**. Integral Tak-tentu dinyatakan sebagai

$$\int f(x)dx = F(x) + C \quad (\text{P.6.1})$$

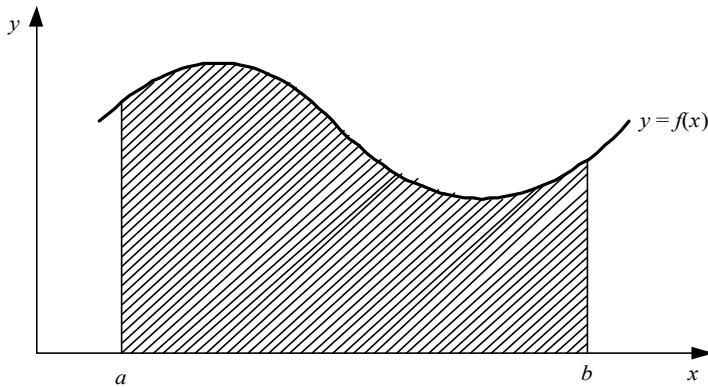
Solusinya, $F(x)$, adalah fungsi menerus sedemikian sehingga $F'(x) = f(x)$, dan C adalah sebuah konstanta. Integral Tentu menangani perhitungan integral di antara batas-batas yang telah ditentukan, yang dinyatakan sebagai

$$I = \int_a^b f(x)dx \quad (\text{P.6.2})$$

Menurut teorema dasar kalkulus integral, persamaan (P.6.2) dihitung sebagai

$$\int_a^b f(x)dx = F(x) \Big|_a^b = F(b) - F(a)$$

Secara geometri, integrasi Tentu sama dengan luas daerah yang dibatasi oleh kurva $y = f(x)$, garis $x = a$ dan garis $x = b$ (Gambar 6.1). Daerah yang dimaksud ditunjukkan oleh bagian yang diarsir.



Gambar 6.1 Tafsiran geometri integral Tentu

Fungsi-fungsi yang dapat diintegrasikan dapat dikelompokkan sebagai

1. Fungsi menerus yang sederhana, seperti polinomial, eksponensial, atau fungsi trigonometri. Misalnya,

$$\int_0^2 (6x^3 - x^2 + \cos(x) - e^x) dx$$

Fungsi sederhana seperti ini mudah dihitung integralnya secara eksak dengan menggunakan metode analitik. Metode-metode analitik untuk menghitung integral fungsi yang demikian sudah tersedia, yaitu

$$\int ax^n dx = ax^{n+1}/(n+1) + C$$

$$\int e^{ax} dx = e^{ax}/a + C$$

$$\int \sin(ax+b) dx = -1/a \cos(ax+b) + C$$

$$\int \cos(ax+b) dx = 1/a \sin(ax+b) + C$$

$$\int dx/x = \ln|x| + C$$

$$\int \ln|x| dx = x \ln|x| - x + C$$

2. Fungsi menerus yang rumit, misalnya

$$\int_0^2 \frac{2 + \cos(1 + x^{3/2})}{\sqrt{1 + 0.5 \sin x}} e^{0.5x} dx$$

Fungsi yang rumit seperti ini jelas sulit, bahkan tidak mungkin, diselesaikan dengan metode-metode integrasi yang sederhana. Karena itu, solusinya hanya dapat dihitung dengan metode numerik.

3. Fungsi yang ditabulasikan, yang dalam hal ini nilai x dan $f(x)$ diberikan dalam sejumlah titik diskrit. Fungsi seperti ini sering dijumpai pada data hasil eksperimen di laboratorium atau berupa data pengamatan di lapangan. Pada kasus terakhir ini, umumnya fungsi $f(x)$ tidak diketahui secara eksplisit. Yang dapat diukur hanyalah besaran fisisnya saja. Misalnya,

x	$f(x)$
0.00	6.0
0.25	7.5
0.50	8.0
0.75	9.0
1.00	8.5

Integrasi fungsi seperti ini jelas harus didikerjakan secara numerik.

6.1 Terapan Integral dalam Bidang Sains dan Rekayasa

Integral mempunyai banyak terapan dalam bidang sains dan rekayasa. Dalam praktek rekayasa, seringkali fungsi yang diintegrasikan (*integrand*) adalah fungsi empirik yang diberikan dalam bentuk tabel, atau *integrand*-nya tidak dalam bentuk fungsi elementer (seperti $\sin x$, fungsi Gamma $\Gamma(\alpha)$, dsb), atau fungsi eksplisit f yang terlalu rumit untuk diintegralkan [KRE88]. Oleh sebab itu, metode numerik dapat digunakan untuk menghampiri integrasi.

Di bawah ini diberikan beberapa contoh persoalan dalam bidang sains dan rekayasa.

1. Dalam bidang fisika, integral digunakan untuk menghitung persamaan kecepatan. Misalkan kecepatan sebuah partikel merupakan fungsi waktu menerus yang diketahui terhadap waktu, $v(t)$. Jarak total d yang ditempuh oleh partikel ini selama waktu t diberikan oleh:

$$d = \int_0^t v(t) dt$$

2. Dalam bidang teknik elektro/kelistrikan, telah diketahui bahwa harga rata-rata suatu arus listrik yang berosilasi sepanjang satu periode boleh nol. Disamping kenyataan bahwa hasil netto adalah nol, arus tersebut mampu menimbulkan kerja dan menghasilkan panas. Karena itu para rekayasaawan listrik sering mencirikan arus yang demikian dengan persamaan

$$I_{RMS} = \sqrt{\frac{\int_0^T i^2(t)dt}{T}}$$

yang dalam hal ini I_{RMS} adalah arus *RMS (root-mean-square)*, T adalah periode, dan $i(t)$ adalah arus pada rangkaian, misalnya

$$\begin{aligned} i(t) &= 5e^{-2t} \sin 2\pi t && \text{untuk } 0 \leq t \leq T/2 \\ &= 0 && \text{untuk } T/2 \leq t \leq T \end{aligned}$$

3. Contoh fungsi dalam bentuk tabel adalah pengukuran fluks panas matahari yang diberikan oleh tabel berikut:

Waktu, jam	Fluks panas q , kalori/cm/jam
0	0.1
1	1.62
2	5.32
3	6.29
4	7.8
5	8.81
6	8.00
7	8.57
8	8.03
9	7.04
10	6.27
11	5.56
12	3.54
13	1.0
14	0.2

Data yang ditabulasikan pada tabel ini memberikan pengukuran fluks panas q setiap jam pada permukaan sebuah kolektor sinar matahari. Diminta

memperkirakan panas total yang diserap oleh panel kolektor seluas 150.000 cm² selama waktu 14 jam. Panel mempunyai kemangkusian penyerapan (*absorption*), e_{ab} , sebesar 45%. Panas total yang diserap diberikan oleh persamaan

$$H = e_{ab} \int_0^t qAdt$$

Demikianlah beberapa contoh terapan integral dalam bidang sains dan rekayasa. Umumnya fungsi yang diintegral bentuknya rumit sehingga sukar diselesaikan secara analitik. Karena itu, perhitungan integral secara numerik lebih banyak dipraktekan oleh para insinyur.

6.2 Persoalan Integrasi Numerik

Persoalan integrasi numerik ialah menghitung secara numerik integral Tentu

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

yang dalam hal ini a dan b batas-batas integrasi, f adalah fungsi yang dapat diberikan secara eksplisit dalam bentuk persamaan ataupun secara empirik dalam bentuk tabel nilai.

Terdapat tiga pendekatan dalam menurunkan rumus integrasi numerik. Pendekatan pertama adalah berdasarkan tafsiran geometri integral Tentu. Daerah integrasi dibagi atas sejumlah pias (*strip*) yang berbentuk segiempat. Luas daerah integrasi dihampiri dengan luas seluruh pias. Rumus, dalam bab ini disebut kaidah, integrasi numerik yang diturunkan dengan pendekatan ini digolongkan ke dalam **metode pias**.

Pendekatan kedua adalah berdasarkan polinom interpolasi. Di sini fungsi *integrand* $f(x)$ dihampiri dengan polinom interpolasi $p_n(x)$. Selanjutnya, integrasi dilakukan terhadap $p_n(x)$ karena polinom lebih mudah diintegral ketimbang mengintegralkan $f(x)$. Rumus integrasi numerik yang diturunkan dengan pendekatan ini digolongkan ke dalam **metode Newton-Cotes**, yaitu metode yang umum untuk menurunkan rumus integarsi numerik..

Pendekatan ketiga sama sekali tidak menggunakan titik-titik diskrit sebagaimana pada kedua pendekatan di atas. Nilai integral diperoleh dengan mengevaluasi nilai fungsi pada sejumlah titik tertentu di dalam selang [-1, 1], mengalikannya

dengan suatu konstanta, kemudian menjumlahkan keseluruhan perhitungan. Pendekatan ketiga ini dinamakan **Kuadratur Gauss**, yang akan dibahas pada bagian akhir bab ini.

6.3 Metode Pias

Pada umumnya, metode perhitungan integral secara numerik bekerja dengan sejumlah titik diskrit. Karena data yang ditabulasikan sudah berbentuk demikian, maka secara alami ia sesuai dengan kebanyakan metode integrasi numerik. Untuk fungsi menerus, titik-titik diskrit itu diperoleh dengan menggunakan persamaan fungsi yang diberikan untuk menghasilkan tabel nilai.

Dihubungkan dengan tafsiran geometri integral Tentu, titik-titik pada tabel sama dengan membagi selang integrasi $[a, b]$ menjadi n buah pias (*strip*) atau segmen (Gambar 6.2). Lebar tiap pias adalah

$$h = \frac{b - a}{n} \quad (\text{P.6.3})$$

Titik absis pias dinyatakan sebagai

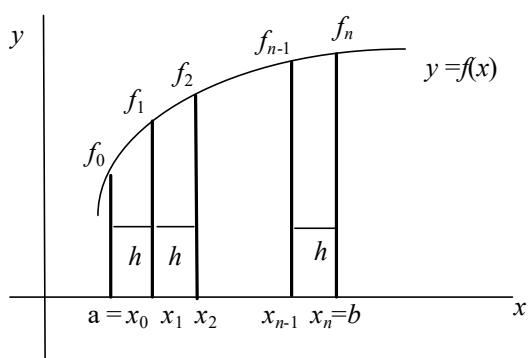
$$x_r = a + rh, \quad r = 0, 1, 2, \dots, n \quad (\text{P.6.4})$$

dan nilai fungsi pada titik absis pias adalah

$$f_r = f(x_r) \quad (\text{P.6.5})$$

Luas daerah integrasi $[a, b]$ dihampiri sebagai luas n buah pias. Metode integrasi numerik yang berbasis pias ini disebut metode pias. Ada juga buku yang menyebutnya **metode kuadratur**, karena pias berbentuk segiempat.

r	x_r	f_r
0	x_0	f_0
1	x_1	f_1
2	x_2	f_2
3	x_3	f_3
4	x_4	f_4
...
$n-2$	x_{n-2}	f_{n-2}
$n-1$	x_{n-1}	f_{n-1}
n	x_n	f_n



Gambar 6.2 Metode pias

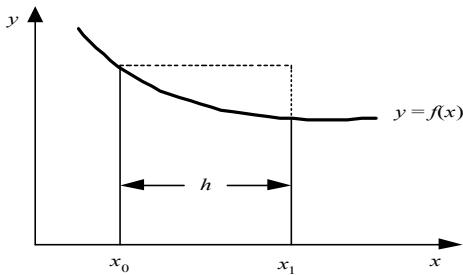
Kaidah integrasi numerik yang dapat diturunkan dengan metode pias adalah:

1. Kaidah segiempat (*rectangle rule*)
2. Kaidah trapesium (*trapezoidal rule*)
3. Kaidah titik tengah (*midpoint rule*)

Dua kaidah pertama pada hakekatnya sama, hanya cara penurunan rumusnya yang berbeda, sedangkan kaidah yang ketiga, kaidah titik tengah, merupakan bentuk kompromi untuk memperoleh nilai hampiran yang lebih baik.

6.3.1 Kaidah Segiempat

Pandang sebuah pias berbentuk empat persegi panjang dari $x = x_0$ sampai $x = x_1$ berikut (Gambar 6.3).



Gambar 6.3 Kaidah segiempat

Luas satu pias adalah (tinggi pias = $f(x_0)$)

$$\int_{x_0}^{x_1} f(x)dx \approx h f(x_0) \quad (\text{P.6.6})$$

atau (tinggi pias = $f(x_1)$)

$$\int_{x_0}^{x_1} f(x)dx \approx h f(x_1) \quad (\text{P.6.7})$$

Jadi,

$$\begin{aligned} \int_{x_0}^{x_1} f(x)dx &\approx hf(x_0) \\ \int_{x_0}^{x_1} f(x)dx &\approx hf(x_1) \\ \hline 2 \int_{x_0}^{x_1} f(x)dx &\approx h [f(x_0) + f(x_1)] \end{aligned}$$

Bagi setiap ruas persamaan hasil penjumlahan di atas dengan 2, untuk menghasilkan

$$\int_{x_0}^{x_1} f(x)dx \approx \frac{h}{2} [f(x_0) + f(x_1)] \quad (\text{P.6.8})$$

Persamaan (P.6.8) ini dinamakan **kaidah segiempat**. Kaidah segiempat untuk satu pias dapat kita perluas untuk menghitung

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

yang dalam hal ini, I sama dengan luas daerah integrasi dalam selang $[a, b]$. Luas daerah tersebut diperoleh dengan membagi selang $[a, b]$ menjadi n buah pias segiempat dengan lebar h , yaitu pias dengan absis $[x_0, x_1], [x_1, x_2], [x_2, x_3], \dots$, dan pias $[x_{n-1}, x_n]$. Jumlah luas seluruh pias segiempat itu adalah hampiran luas I (Gambar 6.4). Kaidah integrasi yang diperoleh adalah **kaidah segiempat gabungan** (*composite rectangle's rule*):

$$\begin{aligned} \int_a^b f(x)dx &\approx hf(x_0) + hf(x_1) + hf(x_2) + \dots + hf(x_{n-1}) \\ \int_a^b f(x)dx &\approx hf(x_1) + hf(x_2) + hf(x_3) + \dots + hf(x_n) \\ \hline 2 \int_a^b f(x)dx &\approx hf(x_0) + 2hf(x_1) + 2hf(x_2) + \dots + 2hf(x_{n-1}) + hf(x_n) \end{aligned}$$

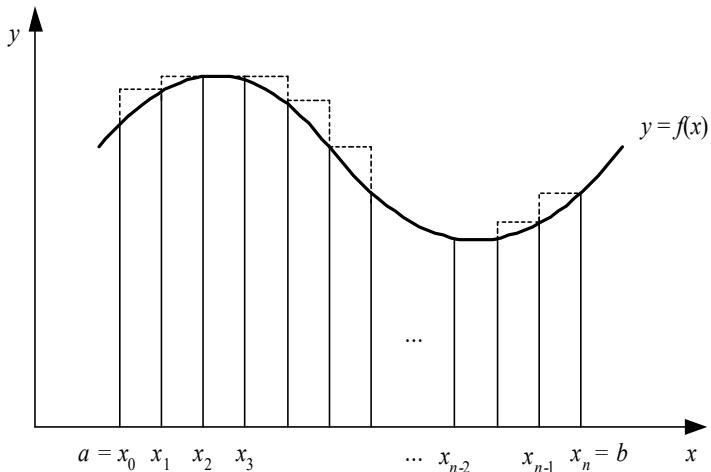
Bagi setiap ruas persamaan hasil penjumlahan di atas dengan 2, untuk menghasilkan

$$\int_a^b f(x)dx \approx \frac{h}{2}f(x_0) + hf(x_1) + hf(x_2) + \dots + hf(x_{n-1}) + \frac{h}{2}f(x_n)$$

Jadi, kaidah segiempat gabungan adalah

$$\int_a^b f(x)dx \approx \frac{h}{2} (f_0 + 2f_1 + 2f_2 + \dots + 2f_{n-1} + f_n) = \frac{h}{2}(f_0 + 2 \sum_{i=1}^{n-1} f_i + f_n) \quad (\text{P.6.9})$$

dengan $f_r = f(x_r)$, $r = 0, 1, 2, \dots, n$.



Gambar 6.4 Kaidah segiempat gabungan

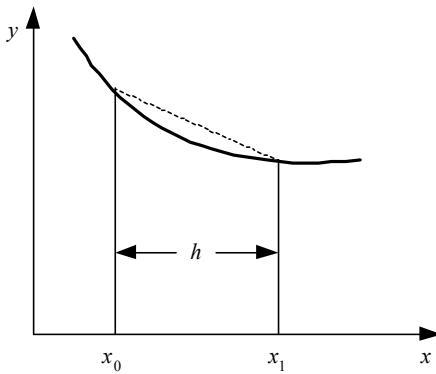
6.3.2 Kaidah Trapezium

Pandang sebuah pias berbentuk trapesium dari $x = x_0$ sampai $x = x_l$ berikut (Gambar 6.5):

Luas satu trapesium adalah

$$\int_{x_0}^{x_1} f(x)dx \approx \frac{h}{2} [f(x_0) + f(x_1)] \quad (\text{P.6.10})$$

Persamaan (P.6.10) ini dikenal dengan nama **kaidah trapesium**. Catatlah bahwa kaidah trapesium sama dengan kaidah segiempat.



Gambar 6.5 Kaidah trapesium

Bila selang $[a, b]$ dibagi atas n buah pias trapesium, kaidah integrasi yang diperoleh adalah **kaidah trapesium gabungan** (*composite trapezoidal's rule*):

$$\begin{aligned}
 \int_a^b f(x)dx &\approx \int_{x_0}^{x_1} f(x)dx + \int_{x_1}^{x_2} f(x)dx + \dots + \int_{x_{n-1}}^{x_n} f(x)dx \\
 &\approx \frac{h}{2} [f(x_0) + f(x_1)] + \frac{h}{2} [f(x_1) + f(x_2)] + \dots + \frac{h}{2} [f(x_{n-1}) + f(x_n)] \\
 &\approx \frac{h}{2} [f(x_0) + 2f(x_1) + 2f(x_2) + \dots + 2f(x_{n-1}) + f(x_n)] \\
 &\approx \frac{h}{2} (f_0 + 2 \sum_{i=1}^{n-1} f_i + f_n)
 \end{aligned} \tag{P.6.11}$$

dengan $f_r = f(x_r)$, $r = 0, 1, 2, \dots, n$.

Program 6.1 Kaidah Trapesium

```
procedure trapesium(a, b : real; n: integer; var I : real);
{ Menghitung integrasi f(x) di dalam selang [a, b] dan jumlah pias
adalah n dengan menggunakan kaidah trapesium.
K.Awal : nilai a, b, dan n sudah terdefinisi
K.Akhir: I adalah hampiran integrasi yang dihitung dengan kaidah
segi-empat.
}
var
  h, x, sigma: real;
  r : integer;
begin
  h:=(b-a)/n;           {lebar pias}
  x:=a;                  {awal selang integrasi}
  I:=f(a) + f(b);
  sigma:=0;
  for r:=1 to n-1 do
    begin
      x:=x+h;
      sigma:=sigma + 2*f(x);
    end;
  I:=(I+sigma)*h/2;     { nilai integrasi numerik}
end;
```

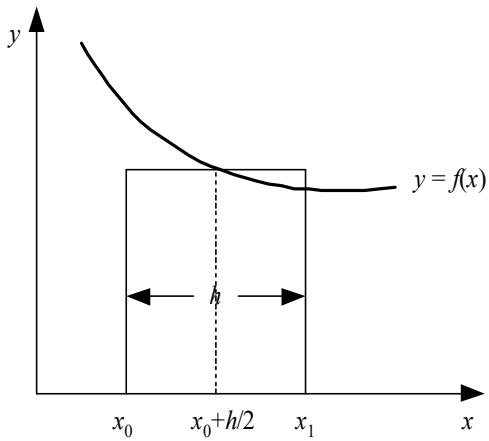
6.3.3 Kaidah Titik Tengah

Pandang sebuah pias berbentuk empat persegi panjang dari $x = x_0$ sampai $x = x_1$ dan titik tengah absis $x = x_0 + h/2$ (Gambar 6.6).

Luas satu pias adalah

$$\int_{x_0}^{x_1} f(x)dx \approx h f(x_0 + h/2) \approx h f(x_{1/2}) \quad \text{P.6.12}$$

Persamaan (P.6.12) ini dikenal dengan nama **kaidah titik-tengah**.



Gambar 6.6 Kaidah titik tengah

Kaidah titik-tengah gabungan adalah (Gambar 6.7):

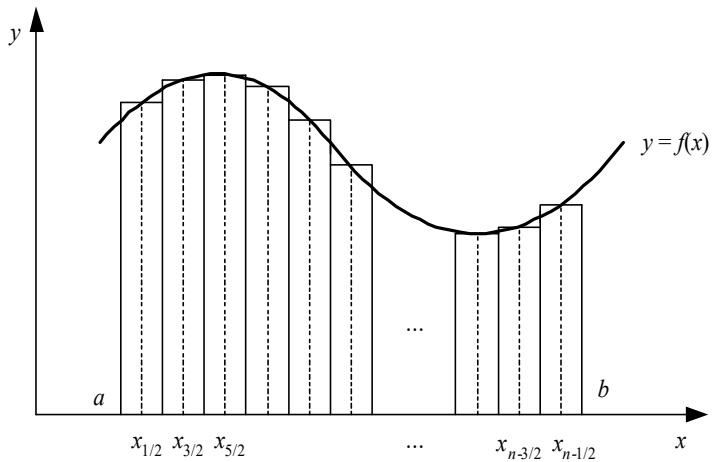
$$\begin{aligned}
 \int_a^b f(x) dx &\approx \int_{x_0}^{x_1} f(x) dx + \int_{x_1}^{x_2} f(x) dx + \dots + \int_{x_{n-1}}^{x_n} f(x) dx \\
 &\approx hf(x_{1/2}) + hf(x_{3/2}) + hf(x_{5/2}) + hf(x_{7/2}) + \dots + hf(x_{n-1/2}) \\
 &\approx h(f_{1/2} + f_{3/2} + \dots + f_{n-1/2}) \approx h \sum_{i=0}^{n-1} f_{i+1/2}
 \end{aligned} \tag{P.6.13}$$

yang dalam hal ini,

$$x_{r+1/2} = a + (r+1/2)h$$

dan

$$f_{r+1/2} = f(x_{r+1/2}) \quad r = 0, 1, 2, \dots, n-1$$



Gambar 6.7 Kaidah titik-tengah gabungan

Program 6.2 KaidahTitik-tengah

```

procedure titik_tengah(a, b : real; n: integer; var I : real);
{ menghitung integrasi f(x) dalam selang [a, b] dengan jumlah pias
sebanyak n.
K.Awal : harga a, b, dan n sudah terdefinisi
K.Akhir: I adalah hampiran integrasi yang dihitung dengan kaidah
titik-tengah
}
var
    h, x, sigma : real;
    r : integer;
begin
    h:=(b-a)/n;           {lebar pias}
    x:= a+h/2;            {titik tengah pertama}
    sigma:=f(x);
    for r:=1 to n-1 do
        begin
            x:=x+h;
            sigma:=sigma + f(x)
        end;
    I:=sigma*h;           { nilai integrasi numerik}
end;

```

6.3.4 Galat Metode Pias

Sekarang akan kita hitung berapa besar galat hasil integrasi untuk masing-masing metode. Misalkan

I adalah nilai integrasi sejati

dan

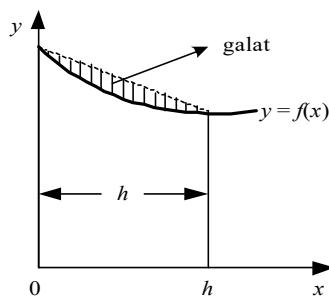
I' adalah integrasi secara numerik

maka galat hasil integrasi numerik didefinisikan sebagai

$$E = I - I' \quad (\text{P.6.14})$$

Untuk penurunan galat, kita tinjau galat integrasi di dalam selang $[0, h]$,

$$I = \int_0^h f(x) dx \quad (\text{P.6.15})$$



Gambar 6.8 Galat kaidah trapesium (bagian yang diarsir)

Untuk setiap kaidah akan kita turunkan galatnya berikut ini.

6.3.4.1 Galat Kaidah Trapezium

Galat untuk satu buah pias (Gambar 6.8) adalah

$$E = \int_0^h f(x)dx - \frac{h}{2} (f_0 + f_1)$$

Uraikan $f(x)$ ke dalam deret Taylor di sekitar $x_0 = 0$

$$f(x) = f_0 + xf'_0 + \frac{1}{2}x^2f''_0 + \frac{1}{6}x^3f'''_0 + \dots$$

Uraikan $f_1 = f(x_1) = f(h)$ ke dalam deret Taylor di sekitar $x_0 = 0$

$$f_1 = f(x_1) = f(h) = f_0 + hf'_0 + \frac{1}{2}h^2f''_0 + \dots$$

Maka,

$$\begin{aligned} E &= \int_0^h [f_0 + xf'_0 + \frac{1}{2}x^2f''_0 + \frac{1}{6}x^3f'''_0 + \dots] dx - \frac{h}{2}f_0 - \frac{h}{2}[f_0 + hf'_0 + \\ &\quad \frac{1}{2}h^2f''_0 + \dots] \\ &= xf'_0 + 1/2x^2f''_0 + 1/6x^3f'''_0 + \dots \Big|_0^h - 1/2hf'_0 - 1/2h^2f''_0 - 1/4h^3f'''_0 - \dots \\ &= (hf'_0 + 1/2h^2f''_0 + 1/6h^3f'''_0 + \dots) - (hf'_0 + 1/2h^2f''_0 + 1/4h^3f'''_0 + \dots) \\ &= -\frac{1}{12}h^3f''(t), \quad 0 < t < h \\ &\approx O(h^3) \end{aligned} \tag{P.6.16}$$

Jadi,

$$\int_0^h f(x)dx \approx \frac{h}{2}(f_0 + f_1) + O(h^3) \tag{P.6.17}$$

Persamaan (P.6.17) ini menyatakan bahwa galat kaidah trapesium sebanding dengan h^3 . Pernyataan ini dibuat dengan andaian bahwa $f(x)$ menerus dalam selang $[0, h]$. Jika tidak, maka galat tidak sebanding dengan h^3 [NAK93].

Untuk n buah pias, galat keseluruhan (total) adalah

$$E_{tot} \approx -\frac{h^3}{12} (f_0'' + f_1'' + f_2'' + \dots + f_{n-1}'')$$

yang dapat disederhanakan dengan teorema nilai antara untuk penjumlahan menjadi

$$\begin{aligned} E_{tot} &\approx -\frac{h^3}{12} \sum_{i=1}^{n-1} f_i'' \\ &\approx -n \frac{h^3}{12} f''(t) \quad , a < t < b \end{aligned} \tag{P.6.18}$$

Mengingat

$$h = \frac{b-a}{n}$$

maka

$$\begin{aligned} E_{tot} &\approx -n \frac{h^3}{12} f''(t) \\ &\approx -n \frac{b-a}{n} \frac{h^3}{12} f''(t) \\ &\approx -\frac{h^3}{12} (b-a) f''(t) \\ &\approx O(h^2) \end{aligned} \tag{P.6.19}$$

Dengan demikian,

$$\int_a^b f(x) dx = \frac{h}{2} (f_0 + 2 \sum_{i=1}^{n-1} f_i + f_n) + O(h^2) \tag{P.6.20}$$

Jadi, galat total integrasi dengan kaidah trapesium sebanding dengan kuadrat lebar pias (h). Semakin kecil ukuran ukuran h , semakin kecil pula galatnya, namun semakin banyak jumlah komputasinya.

Contoh 6.1

[GER85] Hitung integral $\int_{1.8}^{3.4} e^x dx$ dengan kaidah trapesium. Ambil $h = 0.2$. Perkirakan juga batas-batas galatnya. Gunakan 5 angka bena.

Penyelesaian:

Fungsi *integrand*-nya adalah

$$f(x) = e^x$$

Jumlah pias adalah $n = (b-a)/h = (3.4 - 1.8)/0.2 = 8$

Tabel data diskritnya adalah sebagai berikut:

r	x_r	$f(x_r)$	r	x_r	$f(x_r)$
0	1.8	6.050	5	2.8	16.445
1	2.0	7.389	6	3.0	20.086
2	2.2	9.025	7	3.2	24.533
3	2.4	11.023	8	3.4	29.964
4	2.6	13.464			

Nilai integrasinya,

$$\begin{aligned}\int_{1.8}^{3.4} e^x dx &\approx \frac{h}{2} (f_0 + 2f_1 + 2f_2 + \dots + 2f_6 + 2f_7 + f_8) \\ &\approx \frac{0.2}{2} [[6.050 + 2(7.389) + 2(9.025) + \dots + 2(16.445) \\ &\quad + 2(20.086) + 2(24.533) + 29.964]] \\ &\approx 23.994\end{aligned}$$

Nilai integrasi sejatinya adalah

$$\int_{1.8}^{3.4} e^x dx = e^x \Big|_{x=1.8}^{x=3.4} = e^{3.4} - e^{1.8} = 29.964 - 6.050 = 23.914$$

Galat kaidah trapesium:

$$E = -\frac{h^2}{12} (b-a)f''(t), \quad 1.8 < t < 3.4$$

Karena $f(x) = e^x$, $f'(x) = e^x$, dan $f''(x) = e^x$ maka

$$E = -\frac{1}{12} 0.2^2 (3.4 - 1.8)e^x, \quad 1.8 < t < 3.4$$

Karena fungsi $f(x) = e^x$ menaik secara monoton di dalam selang $[1.8, 3.4]$, maka kita dapat menentukan batas-batas galatnya:

$$E = -\frac{1}{12} (0.2)^2 (3.4 - 1.8) \times \begin{cases} e^{1.8} (\min) = -0.0323(\min) \\ e^{3.4} (\max) = -0.1598(\max) \end{cases}$$

atau

$$-0.0323 < E < -0.1598.$$

Di sini nilai sejati I harus terletak di antara

$$23.994 - 0.1598 = 23.834 \text{ dan } 23.994 - 0.0323 = 23.962$$

(nilai integrasi sejatinya adalah 23.914, yang memang terletak di antara 23.834 dan 23.962)

Galat hasil integrasi $\int_{1.8}^{3.4} e^x dx$ adalah

$$23.914 - 23.944 = -0.080$$

yang memang terletak di antara galat minimum dan galat maksimum. ■

6.3.4.2 Galat Kaidah Titik Tengah

Galat untuk satu buah pias adalah

$$E = \int_0^h f(x)dx - hf_{1/2}$$

Dengan cara penurunan yang sama seperti pada kaidah trapesium, dapat dibuktikan bahwa

$$E \approx \frac{h^3}{24} f''(t), \quad 0 < t < h \quad (\text{P.6.20})$$

Galat untuk seluruh pias adalah

$$E_{tot} \approx n \frac{h^3}{24} f''(t), \quad a < t < b$$

$$\begin{aligned} &\approx \frac{h^2}{24} (b - a) f''(t) \\ &= O(h^2) \end{aligned} \quad (\text{P.6.21})$$

Dapat dilihat bahwa galat integrasi dengan kaidah titik tengah sama dengan 1/2 kali galat pada kaidah trapesium namun berbeda tanda. Dengan kata lain, kaidah titik tengah lebih baik daripada kaidah trapesium. Sayangnya, kaidah titik-tengah tidak dapat diterapkan jika fungsi $f(x)$ tidak diketahui secara eksplisit (hanya tersedia data berupa tabel titik-titik saja) sebab kita tidak dapat menghitung nilai tengah, $f_{r+1/2}$.

6.4 Metode Newton-Cotes

Metode *Newton-Cotes* adalah metode yang umum untuk menurunkan kaidah integrasi numerik. Polinom interpolasi menjadi dasar metode Newton-Cotes. Gagasananya adalah menghampiri fungsi $f(x)$ dengan polinom interpolasi $p_n(x)$

$$I = \int_a^b f(x) dx \approx \int_a^b p_n(x) dx \quad (\text{P.6.22})$$

yang dalam hal ini,

$$p_n(x) = a_0 + a_1x + a_2x^2 + \dots + a_{n-1}x^{n-1} + a_nx^n$$

Mengapa polinom interpolasi? Karena suku-suku polinom mudah diintegralkan dengan rumus integral yang sudah baku, yaitu

$$\int ax^n dx = \frac{a}{n+1}x^{n+1} + C$$

Sembarang polinom interpolasi yang telah kita bahas di dalam Bab 5 dapat digunakan sebagai hampiran fungsi, tetapi di dalam bab ini polinom interpolasi yang kita pakai adalah polinom Newton-Gregory maju:

$$\begin{aligned} p_n(x) &= f_0 + (x - x_0) \frac{\Delta f_0}{1!h} + (x - x_0)(x - x_1) \frac{\Delta^2 f_0}{2!h^2} + \dots + \\ &(x - x_0)(x - x_1) \dots (x - x_{n-1}) \frac{\Delta^n f_0}{n!h^n} \end{aligned}$$

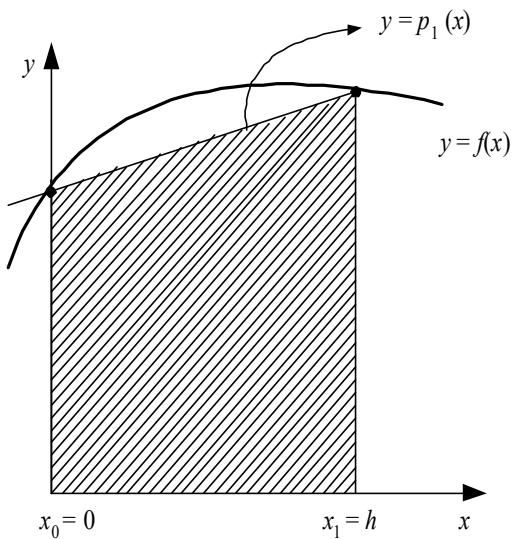
Dari beberapa kaidah integrasi numerik yang diturunkan dari metode Newton-Cotes, tiga di antaranya yang terkenal adalah:

1. Kaidah trapesium (*Trapezoidal rule*)
2. Kaidah Simpson 1/3 (*Simpson's 1/3 rule*)
3. Kaidah Simpson 3/8 (*Simpson's 3/8 rule*)

Sebagai catatan, kaidah trapesium sudah kita turunkan dengan metode pias. Metode Newton-Cotes memberikan pendekatan lain penurunan kaidah trapesium.

6.4.1 Kaidah Trapezium

Diberikan dua buah titik data $(0, f(0))$ dan $(h, f(h))$. Polinom interpolasi yang melalui kedua buah titik itu adalah sebuah garis lurus. Luas daerah yang dihitung sebagai hampiran nilai integrasi adalah daerah di bawah garis lurus tersebut (Gambar 6.9).



Gambar 6.9 Kaidah trapesium

Polinom interpolasi Newton-Gregory derajat 1 yang melalui kedua buah titik itu adalah

$$p_1(x) = f(x_0) + x \frac{\Delta f(x_0)}{h} = f_0 + x \frac{\Delta f_0}{h}$$

Integrasikan $p_1(x)$ di dalam selang $[0,1]$:

$$\begin{aligned}
 I &\approx \int_0^h f(x)dx \quad \approx \int_0^h p_1(x)dx \\
 &\approx \int_0^h \left(f_0 + x \frac{\Delta f_0}{h} \right) dx \\
 &\approx xf_0 + \frac{x^2}{2h} \Delta f_0 \Big|_{x=0}^{x=h} \\
 &\approx hf_0 + \frac{h}{2} \Delta f_0 \\
 &\approx hf_0 + \frac{h}{2} (f_1 - f_0) \quad , \text{ sebab } \Delta f_0 = f_1 - f_0 \\
 &\approx \frac{h}{2} f_0 + \frac{h}{2} f_1 \\
 &\approx \frac{h}{2} (f_0 + f_1)
 \end{aligned}$$

Jadi, kaidah trapesium adalah

$$\int_0^h f(x)dx \approx \frac{h}{2} (f_0 + f_1) \tag{P.6.23}$$

Galat kaidah trapesium sudah kita turunkan sebelumnya pada metode pias, yaitu

$$E = -\frac{1}{12} h^3 f'''(t) = O(h^3) \quad , \quad 0 < t < h$$

Jadi,

$$\int_0^h f(x)dx \approx \frac{h}{2} (f_0 + f_1) + O(h^3) \tag{P.6.24}$$

Kaidah trapesium untuk integrasi dalam selang $[0, h]$ kita perluas untuk menghitung

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

yang dalam hal ini, I sama dengan luas daerah integrasi di dalam selang $[a, b]$. Luas daerah tersebut diperoleh dengan membagi selang $[a, b]$ menjadi n buah upaselang (*subinterval*) dengan lebar tiap upaselang h , yaitu $[x_0, x_1], [x_1, x_2], [x_2, x_3], \dots, [x_{n-1}, x_n]$. Titik-titik ujung tiap upaselang diinterpolasi dengan polinom derajat 1. Jadi, di dalam selang $[a, b]$ terdapat n buah polinom derajat satu yang terpotong-potong (*piecewise*). Integrasi masing-masing polinom itu menghasilkan n buah kaidah trapesium yang disebut kaidah trapesium gabungan. Luas daerah integrasi di dalam selang $[a, b]$ adalah jumlah seluruh luas trapesium, yaitu

$$\begin{aligned} \int_a^b f(x)dx &\approx \int_{x_0}^{x_1} f(x)dx + \int_{x_1}^{x_2} f(x)dx + \dots + \int_{x_{n-1}}^{x_n} f(x)dx \\ &\approx \frac{h}{2} (f_0 + f_1) + \frac{h}{2} (f_1 + f_2) + \dots + \frac{h}{2} (f_{n-1} + f_n) \\ &\approx \frac{h}{2} (f_0 + 2f_1 + 2f_2 + \dots + 2f_{n-1} + f_n) \\ &\approx \frac{h}{2} (f_0 + 2f_i + \sum_{i=1}^{n-1} f_i) \end{aligned} \quad (\text{P.6.25})$$

dengan $f_r = f(x_r)$, $r = 0, 1, 2, \dots, n$.

Galat total kaidah trapesium gabungan sudah kita turunkan pada metode pias, yaitu

$$E_{tot} \approx -\frac{h^2}{12} (b - a)f''(t) = O(h^2), \quad x_0 < t < x_n$$

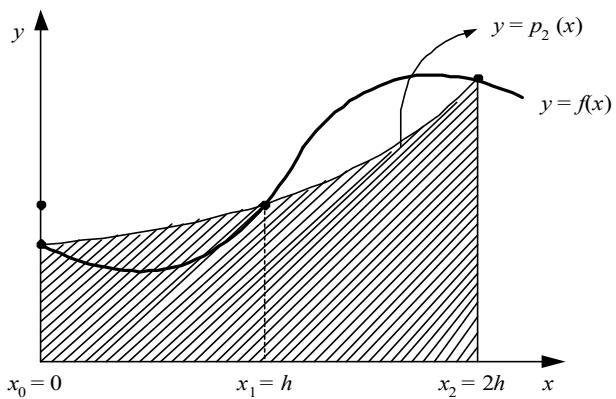
Dengan demikian,

$$\int_a^b f(x)dx = \frac{h}{2} (f_0 + 2\sum_{i=1}^n f_i + f_n) + O(h^2) \quad (\text{P.6.26})$$

Jadi, galat integrasi dengan kaidah trapesium sebanding dengan h^2 .

6.4.2 Kaidah Simpson 1/3

Hampiran nilai integrasi yang lebih baik dapat ditingkatkan dengan menggunakan polinom interpolasi berderajat yang lebih tinggi. Misalkan fungsi $f(x)$ dihampiri dengan polinom interpolasi derajat 2 yang grafiknya berbentuk parabola. Luas daerah yang dihitung sebagai hampiran nilai integrasi adalah daerah di bawah parabola (Gambar 6.10). Untuk itu, dibutuhkan 3 buah titik data, misalkan $(0, f(0))$, $(h, f(h))$, dan $(2h, f(2h))$.



Gambar 6.10 Kaidah Simpson 1/3

Polinom interpolasi Newton-Gregory derajat 2 yang melalui ketiga buah titik tersebut adalah

$$p_2(x) = f(x_0) + \frac{x}{h} \Delta f(x_0) + \frac{x(x-h)}{2!h^2} \Delta^2 f(x_0) = f_0 + x \Delta f_0 + \frac{x(x-h)}{2!h^2} \Delta^2 f_0$$

Integrasikan $p_2(x)$ di dalam selang $[0, 2h]$:

$$\begin{aligned} I &\approx \int_0^{2h} f(x) dx \approx \int_0^{2h} p_2(x) dx \\ &\approx \int_0^{2h} \left(f_0 + \frac{x}{h} \Delta f_0 + \frac{x(x-h)}{2!h^2} \Delta^2 f_0 \right) dx \\ &\approx f_0 x + \frac{1}{2h} x^2 \Delta f_0 + \left(\frac{x^3}{6h^2} - \frac{x^2}{4h} \right) \Delta^2 f_0 \Big|_{x=0}^{x=2h} \end{aligned}$$

$$\begin{aligned}
&\approx 2hf_0 + \frac{4h^2}{2h} \Delta f_0 + \left(\frac{8h^3}{6h^2} - \frac{4h^2}{4h} \right) \Delta^2 f_0 \\
&\approx 2hf_0 + 2h \Delta f_0 + \left(\frac{4h}{3} - h \right) \Delta^2 f_0 \\
&\approx 2hf_0 + 2h \Delta f_0 + \frac{h}{3} \Delta^2 f_0
\end{aligned}$$

Mengingat

$$\Delta f_0 = f_1 - f_0$$

dan

$$\Delta^2 f_0 = \Delta f_1 - \Delta f_0 = (f_2 - f_1) - (f_1 - f_0) = f_2 - 2f_1 + f_0$$

maka, selanjutnya

$$\begin{aligned}
I &\approx 2hf_0 + 2h(f_1 - f_0) + \frac{h}{3}(f_2 - 2f_1 + f_0) \\
&\approx 2hf_0 + 2hf_1 - 2hf_0 + \frac{h}{3}f_2 - \frac{2h}{3}f_1 + \frac{h}{3}f_0 \\
&\approx \frac{h}{3}f_0 + \frac{4h}{3}f_1 + \frac{h}{3}f_2 \\
&\approx \frac{h}{3}(f_0 + 4f_1 + f_2)
\end{aligned} \tag{P.6.27}$$

Persamaan (P.6.27) ini dinamakan **kaidah Simpson 1/3**. Sebutan "1/3" muncul karena di dalam persamaan (P.6.26) terdapat faktor "1/3" (sekaligus untuk membedakannya dengan kaidah Simpson yang lain, yaitu Simpson 3/8).

Misalkan kurva fungsi sepanjang selang integrasi $[a, b]$ kita bagi menjadi $n+1$ buah titik diskrit $x_0, x_1, x_2, \dots, x_n$, dengan n genap, dan setiap tiga buah titik (atau 2 pasang upaselang) di kurva dihampiri dengan parabola (polinom interpolasi derajat 2), maka kita akan mempunyai $n/2$ buah potongan parabola. Bila masing-masing polinom derajat 2 tersebut kita integralkan di dalam upaselang (*sub-interval*) integrasinya, maka jumlah seluruh integral tersebut membentuk **kaidah Simpson 1/3 gabungan**:

$$I_{\text{tot}} = \int_a^b f(x) dx \approx \int_{x_0}^{x_2} f(x) dx + \int_{x_2}^{x_4} f(x) dx + \dots + \int_{x_{n-2}}^{x_n} f(x) dx$$

$$\begin{aligned}
&\approx \frac{h}{3}(f_0 + 4f_1 + f_2) + \frac{h}{3}(f_2 + 4f_3 + f_4) + \dots + \frac{h}{3}(f_{n-2} + 4f_{n-1} + f_n) \\
&\approx \frac{h}{3} (f_0 + 4f_1 + 2f_2 + 4f_3 + 2f_4 + \dots + 2f_{n-2} + 4f_{n-1} + f_n) \\
&\approx \frac{h}{3} (f_0 + 4 \sum_{i=1,3,5}^{n-1} f_i + 2 \sum_{i=2,4,6}^{n-2} f_i + f_n)
\end{aligned} \tag{P.6.28}$$

Persamaan (P.6.28) ini mudah dihafalkan dengan mengingat pola koefisien suku-sukunya:

1, 4, 2, 4, 2, ..., 2, 4, 1

Namun penggunaan kaidah 1/3 Simpson mensyaratkan jumlah upaslang (n) harus genap, ini berbeda dengan kaidah trapesium yang tidak mempunyai persyaratan mengenai jumlah selang.

Program 6.3 Kaidah Simpson 1/3

```

procedure Simpson_sepertiga(a, b : real; n: integer; var I : real);
{ menghitung integrasi f(x) dalam selang [a, b] dengan jumlah pias
sebanyak n (n harus genap)
K.Awal : harga a, b, dan n sudah terdefinisi (n harus genap)
K.Akhir: I adalah hampiran integrasi yang dihitung dengan kaidah
Simpson 1/3
}
var
  h, x, sigma : real;
  r : integer;
begin
  h:=(b-a)/n; {jarak antar titik }
  x:=a; {awal selang integrasi}
  I:=f(a) + f(b);
  sigma:=0;
  for r:=1 to n-1 do
    begin
      x:=x+h;
      if r mod 2 = 1 then { r = 1, 3, 5, ..., n-1 }
        sigma:=sigma + 4*f(x)
      else { r = 2, 4, 6, ..., n-2 }
        sigma:=sigma + 2*f(x);
    end;
  I:=(I+sigma)*h/3; { nilai integrasi numerik}
end;

```

Contoh 6.2

Hitung integral

$$\int_0^1 \frac{1}{1+x} dx \quad [\text{NOB72}]$$

dengan menggunakan

- (a) kaidah trapesium
- (b) kaidah titik-tengah
- (c) kaidah Simpson 1/3

Gunakan jarak antar titik $h = 0.125$.

Penyelesaian:

Jumlah upaselang: $n = (1 - 0)/0.125 = 8$

Tabel titik-titik di dalam selang $[0,1]$:
(untuk kaidah trapesium dan Simpson 1/3)

r	x_r	f_r
0	0	1
1	0.125	0.88889
2	0.250	0.80000
3	0.375	0.72727
4	0.500	0.66667
5	0.625	0.61538
6	0.750	0.57143
7	0.875	0.53333
8	1.000	0.50000

Tabel titik-titik di dalam selang $[0,1]$:
(untuk kaidah titik-tengah)

r	x_r	f_r
1/2	0.063	0.94118
3/2	0.188	0.84211
5/2	0.313	0.76190
7/2	0.438	0.69565
9/2	0.563	0.64000
11/2	0.688	0.59259
13/2	0.813	0.55172
15/2	0.938	0.51613

- (a) dengan kaidah trapesium

$$\begin{aligned} \int_0^1 \frac{1}{1+x} dx &\approx h/2 (f_0 + 2f_1 + 2f_2 + 2f_3 + 2f_4 + 2f_5 + 2f_6 + 2f_7 + f_8) \\ &\approx 0.125/2 [1 + 2(0.88889) + 2(0.80000) + \dots + 0.50000] \\ &\approx 0.69412 \end{aligned}$$

- (b) dengan kaidah titik-tengah

$$\begin{aligned} \int_0^1 \frac{1}{1+x} dx &\approx h (f_{1/2} + f_{3/2} + f_{5/2} + f_{7/2} + f_{9/2} + f_{11/2} + f_{13/2} + f_{15/2}) \\ &\approx 0.125 (5.54128) \end{aligned}$$

(c) dengan kaidah 1/3 Simpson

$$\begin{aligned}\int_0^1 \frac{1}{1+x} dx &\approx h/3 (f_0 + 4f_1 + 2f_2 + 4f_3 + 2f_4 + 4f_5 + 2f_6 + 4f_7 + f_8) \\ &\approx 0.125/3 (16.63568) \\ &\approx 0.69315\end{aligned}$$

Bandingkan solusi (a), (b), dan (c) dengan solusi sejatinya:

$$\int_0^1 \frac{1}{1+x} dx = \ln(1+x) \Big|_{x=0}^{x=1} = \ln(2) - \ln(1) = 0.69314718$$

yang apabila dibulatkan ke dalam 5 angka bena, $f(0.69314718) = 0.69315$, hasilnya tepat sama dengan nilai integrasi yang dihitung dengan kaidah Simpson 1/3. Jadi, kaidah Simpson/3 memang unggul dalam hal ketelitian hasil dibandingkan dua kaidah sebelumnya. ■

Galat Kaidah Simpson 1/3

Galat kaidah Simpson 1/3 untuk dua pasang upaselang adalah

$$E = \int_0^{2h} f(x)dx - \frac{h}{3} (f_0 + 4f_1 + f_2) \quad (\text{P.6.29})$$

Uraikan $f(x)$, f_1 , dan f_2 masing-masing ke dalam deret Taylor di sekitar $x_0 = 0$:

$$f(x) = f_0 + xf'_0 + \frac{x^2}{2} f''_0 + \frac{x^3}{6} f'''_0 + \frac{x^4}{24} f_0^{(iv)} + \dots \quad (\text{P.6.30})$$

$$f_1 = f(h) = f_0 + hf'_0 + \frac{h^2}{2} f''_0 + \frac{h^3}{6} f'''_0 + \frac{h^4}{24} f_0^{(iv)} + \dots \quad (\text{P.6.31})$$

$$f_2 = f(2h) = f_0 + 2hf'_0 + \frac{4h^2}{2} f''_0 + \frac{8h^3}{6} f'''_0 + \frac{16h^4}{24} f_0^{(iv)} + \dots \quad (\text{P.6.32})$$

Sulihkan persamaan (P.6.30), (P.6.31), (P.6.32) ke dalam persamaan (P.6.29):

$$E = \int_0^{2h} \left(f_0 + xf'_0 + \frac{x^2}{2} f''_0 + \frac{x^3}{6} f'''_0 + \frac{x^4}{24} f_0^{(iv)} + \dots \right) dx$$

$$\begin{aligned}
& - \frac{h}{3} [(f_0 + 4f_1 + 4hf'_0 + \frac{4h^2}{2} f_0'' + \frac{4h^3}{6} f_0''' + \frac{4h^4}{24} f_0^{(iv)} + \dots) \\
& + (f_0 + 2hf'_0 + \frac{4h^2}{2} f_0'' + \frac{8h^3}{6} f_0''' + \frac{16h^4}{24} f_0^{(iv)} + \dots)] \\
& = (xf_0 + \frac{x^2}{2} f_0' + \frac{x^3}{6} f_0'' + \frac{x^4}{24} f_0''' + \frac{x^5}{120} f_0^{(iv)} + \dots) \Big|_{x=0} \\
& - \frac{h}{3} (6f_0 + 6hf'_0 + 4h^2 f_0'' + 2h^3 f_0''' + \frac{20h^4}{24} f_0^{(iv)} + \dots) \\
& = (2hf_0 + 2h^2 f_0' + \frac{4h^3}{3} f_0'' + \frac{2h^4}{3} f_0''' + \frac{32h^5}{120} f_0^{(iv)} + \dots) \\
& - (2hf_0 + 2h^2 f_0' + \frac{4h^3}{3} f_0'' + \frac{2h^4}{3} f_0''' + \frac{20h^5}{72} f_0^{IV} + \dots) \\
& = \frac{32h^5}{120} f_0^{(iv)} - \frac{20h^5}{72} f_0^{IV} + \dots \\
& = \left(\frac{8}{30} - \frac{5}{180} \right) h^5 f_0^{(iv)} + \dots \\
& = - \frac{1}{90} h^5 f_0^{(iv)} \\
& = O(h^5)
\end{aligned} \tag{P.6.33}$$

Jadi, kaidah Simpson 1/3 untuk sepasang upaselang ditambah dengan galatnya dapat dinyatakan sebagai

$$\int_0^{2h} f(x) dx = \frac{h}{3} (f_0 + 4f_1 + f_2) + O(h^5)$$

Galat untuk $n/2$ pasang upaselang adalah

$$\begin{aligned}
E_{tot} &= - \frac{1}{90} h^5 (f_0^{(iv)} + f_2^{(iv)} + f_4^{(iv)} + \dots + f_{n-2}^{(iv)}) = - \frac{1}{90} h^5 \sum_{i=0,2,\dots}^{n-2} f_i^{(iv)} \\
&= - \frac{h^5}{90} \cdot \frac{n}{2} \cdot f^{(iv)}(t) \quad , \quad a < t < b \\
&= - \frac{h^4}{180} (b-a) f^{(iv)}(t) \quad , \quad \text{karena } n = (b-a)/h \\
&= O(h^4)
\end{aligned} \tag{P.6.34}$$

Jadi, kaidah Simpson 1/3 gabungan ditambah dengan galatnya dapat dinyatakan sebagai,

$$\int_a^b f(x)dx \approx \frac{h}{3} (f_0 + 4 \sum_{i=1,3,5}^{n-1} f_i + 2 \sum_{i=2,4,6}^{n-2} f_i + f_n) + O(h^4)$$

dengan kata lain, kaidah Simpson 1/3 gabungan berorde 4. Dibandingkan dengan kaidah trapesium gabungan, hasil integrasi dengan kaidah Simpson gabungan jauh lebih baik, karena orde galatnya lebih tinggi. Tapi ada kelemahannya, yaitu kaidah Simpson 1/3 tidak dapat diterapkan bila jumlah upaselang (n) ganjil.

Contoh 6.3

Hitunglah $\int_0^1 \exp(-x^2)dx$ dengan menggunakan kaidah Simpson 1/3 dan jumlah upaselang yang digunakan adalah $n = 10$, lalu taksirlah batas-batas galatnya.

Penyelesaian:

$$h = (1 - 0)/10 = 0.1$$

Tabel titik-titik di dalam selang $[0, 1]$ dengan $h = 0.1$:

r	x_r	f_r
0	0	1.000000
1	0.1	0.990050
2	0.2	0.960789
3	0.3	0.913931
4	0.4	0.852144
5	0.5	0.778801
6	0.6	0.697676
7	0.7	0.612626
8	0.8	0.527292
9	0.9	0.444858
10	1.0	0.367879

Nilai integasi $f(x)$ di dalam selang $[0, 1]$ adalah:

$$\begin{aligned}
I &= \int_0^1 \exp(-x^2) dx \\
&\approx h/3 (f_0 + 4f_1 + 2f_2 + 4f_3 + 2f_4 + 4f_5 + 2f_6 + 4f_7 + 2f_8 + 4f_9 + f_{10}) \\
&\approx 0.1 (1.000000 + 4 \times 0.990050 + 2 \times 0.960789 + \dots + 4 \times 0.444858 + 0.367879) \\
&\approx 0.746825
\end{aligned}$$

Taksiran galatnya:

$$f^{(4)}(x) = 4(4x^4 - 12x^2 + 3)\exp(-x^2)$$

Nilai minimum $f^{(4)}(x)$ adalah pada $x = 2.5 + 0.5\sqrt{10}$, dengan $f^{(4)}(2.5 + 0.5\sqrt{10}) = -7.359$, sedangkan nilai maksimum $f^{(4)}(x)$ adalah pada $x = 0$, dengan $f^{(4)}(0) = 12$, maka batas-batas galatnya adalah

$$\begin{aligned}
E_{tot} &= -\frac{h^4}{180} (b-a)f^{(iv)}(t) \\
&= -\frac{(0.1)^4}{180} (1-0) \times \begin{cases} -7.359(\min) = -0.000004 \\ 12(\max) = 0.000006 \end{cases}
\end{aligned}$$

Jadi, galat integrasinya, E_{tot} , terletak di dalam selang

$$-0.000004 < E_{tot} < 0.000006$$

Di sini nilai sejati I harus terletak di antara

$$0.746825 - 0.000004 = 0.746821 \text{ dan } 0.746825 + 0.000006 = 0.746831$$

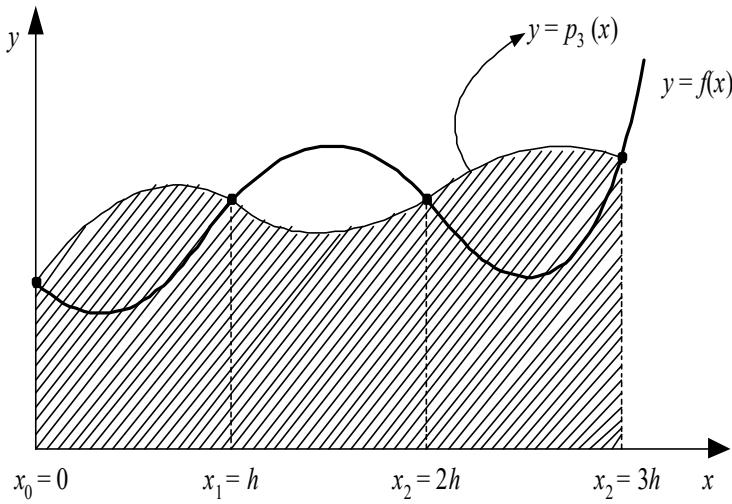
atau

$$0.746821 < I < 0.746831$$

■

6.4.3 Kaidah Simpson 3/8

Seperti halnya pada kaidah Simpson 1/3, hampiran nilai integrasi yang lebih teliti dapat ditingkatkan terus dengan menggunakan polinom interpolasi berderajat lebih tinggi pula. Misalkan sekarang fungsi $f(x)$ kita hampiri dengan polinom interpolasi derajat 3. Luas daerah yang dihitung sebagai hampiran nilai integrasi adalah daerah di bawah kurva polinom derajat 3 tersebut parabola (Gambar 6.11). Untuk membentuk polinom interpolasi derajat 3, dibutuhkan 4 buah titik data, misalkan titik-titik tersebut $(0, f(0))$, $(h, f(h))$, $(2h, f(2h))$, dan $(3h, f(3h))$.



Gambar 6.11 Kaidah Simpson 3/8

Polinom interpolasi Newton-Gregory derajat 3 yang melalui keempat buah titik itu adalah

$$\begin{aligned}
 p_3(x) &= f(x_0) + \frac{x}{h} \Delta f(x_0) + \frac{x(x-h)}{2!h^2} \Delta^2 f(x_0) + \frac{x(x-h)(x-2h)}{3!h^3} \Delta^3 f(x_0) \\
 &= f_0 + \frac{x}{h} \Delta f_0 + \frac{x(x-h)}{2!h^2} \Delta^2 f_0 + \frac{x(x-h)(x-2h)}{3!h^3} \Delta^3 f(x_0)
 \end{aligned} \quad (\text{P.6.35})$$

Integrasi $p_3(x)$ di dalam selang $[0, 3h]$ adalah

$$\begin{aligned}
 I &\approx \int_0^{3h} f(x) dx \approx \int_0^{3h} p_3(x) dx \\
 &\approx \int_0^{3h} \left[f_0 + \frac{x}{h} \Delta f_0 + \frac{x(x-h)}{2!h^2} \Delta^2 f_0 + \frac{x(x-h)(x-2h)}{3!h^3} \Delta^3 f(x_0) \right] dx
 \end{aligned}$$

Dengan cara penurunan yang sama seperti pada kaidah Simpson 1/3, diperoleh

$$\int_0^{3h} f(x) dx \approx \frac{3h}{8} (f_0 + 3f_1 + 3f_2 + f_3) \quad (\text{P.6.36})$$

yang merupakan **kaidah Simpson 3/8**.

Galat kaidah Simpson 3/8 adalah

$$E \approx -\frac{3}{80} h^5 f_0^{(iv)}(t) , \quad 0 < t < 3h \quad (\text{P.6.37})$$

Jadi, kaidah Simpson 3/8 ditambah dengan galatnya dapat dinyatakan sebagai

$$\int_0^{3h} f(x)dx \approx \frac{3h}{8} (f_0 + 3f_1 + 3f_2 + f_3) + O(h^5)$$

Sedangkan kaidah **Simpson 3/8 gabungan** adalah

$$\begin{aligned} \int_a^b f(x)dx &\approx \frac{3h}{8} (f_0 + 3f_1 + 3f_2 + 2f_3 + 3f_4 + 3f_5 + 2f_6 + 3f_7 + 3f_8 + 2f_9 + \dots \\ &\quad + 2f_{n-3} + 3f_{n-2} + 3f_{n-1} + f_n) \\ &\approx \frac{3h}{8} (f_0 + 3 \sum_{\substack{i=1 \\ i \neq 3,6,9,\dots}}^{n-1} f_i + 2 \sum_{i=3,6,9,\dots}^{n-3} f_i + f_n) \end{aligned} \quad (\text{P.6.38})$$

Persamaan (P.6.38) ini mudah dihafalkan dengan mengingat pola suku-sukunya:

$$1, 3, 3, 2, 3, 3, 2, 3, 3, 2, \dots, 2, 3, 3, 1$$

Namun penggunaan kaidah Simpson 3/8 mensyaratkan jumlah upaslang (n) harus kelipatan tiga.

Galat kaidah 3/8 Simpson gabungan adalah

$$\begin{aligned} E_{\text{tot}} &\approx \sum_{i=1}^{n/3} \frac{(-3h^5)}{80} f^{(iv)}(t) \approx -\frac{3h^5}{80} \sum_{i=1}^{n/3} f^{(iv)}(t) \\ &\approx -\frac{3h^5}{80} \cdot \frac{n}{3} \cdot f^{(iv)}(t) \\ &\approx -\frac{h^5}{80} \cdot \frac{(b-a)}{h} f^{(iv)}(t) \end{aligned}$$

$$\approx -\frac{(b-a)h^4}{80} f^{(iv)}(t) , \quad a < t < b \quad (\text{P.6.39})$$

$$= O(h^4)$$

Jadi, kaidah Simpson 3/8 ditambah dengan galatnya dapat dinyatakan sebagai

$$\int_a^b f(x)dx \approx \frac{3h}{8} (f_0 + 3 \sum_{\substack{i=1 \\ i \neq 3,6,9,\dots}}^{n-1} f_i + 2 \sum_{i=3,6,9,\dots}^{n-3} f_i + f_n) + O(h^4)$$

Kaidah Simpson 3/8 memiliki orde galat yang sama dengan orde galat kaidah Simpson 1/3. Namun dalam praktik, kaidah Simpson 1/3 biasanya lebih disukai daripada kaidah Simpson 3/8, karena dengan tiga titik (Simpson 1/3) sudah diperoleh orde ketelitian yang sama dengan 4 titik (Simpson 3/8). Tetapi, untuk n kelipatan tiga, kita hanya dapat menggunakan kaidah Simpson 3/8, dan bukan Simpson 1/3.

Program 6.4 Kaidah Simpson 3/8

```

procedure Simpson_3per8(a, b : real; n: integer; var I : real);
{ menghitung integrasi f(x)dalam selang [a,b]dengan jumlah upa-selang
sebanyak n (n harus kelipatan tiga) }
K.Awal : harga a, b, dan n sudah terdefinisi, n kelipatan 3
K.Akhir: I adalah hampiran integrasi yang dihitung dengan kaidah
3/8 Simpson
}
var
  h, x, sigma : real;
  r : integer;
begin
  h:=(b-a)/n; {jarak antar titik }
  x:=a; {awal selang integrasi}
  I:=f(a) + f(b);
  sigma:=0;
  for r:=1 to n-1 do
    begin
      x:=x+h;
      if r mod 3 = 0 then { r = 3, 6, 9, ..., n-3 }
        sigma:=sigma + 2*f(x)
      else { r ≠ 3, 6, 9, ..., n-1 }
        sigma:=sigma + 3*f(x);
    end;
  I:=(I+sigma)*3*h/8; { nilai integrasi numerik}
end;

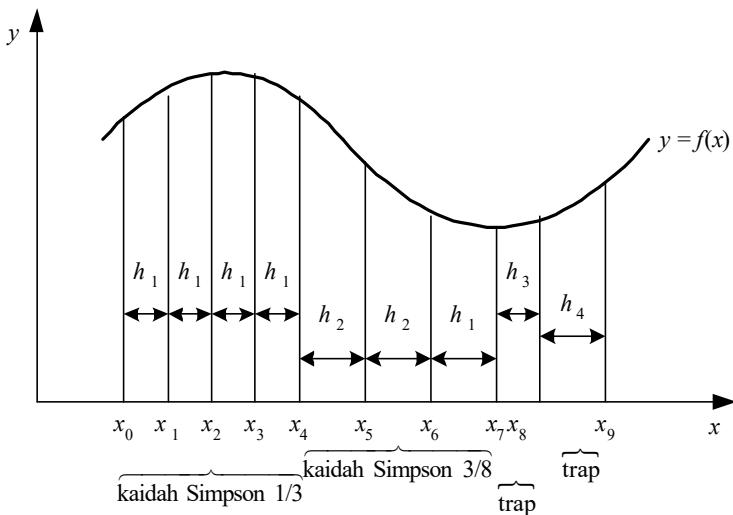
```

6.4.4 Metode Integrasi Numerik Untuk h yang Berbeda-beda

Misalkan jarak antara titik-titik data dalam selang $[a, b]$ tidak seragam. Beberapa titik data mempunyai jarak h_1 , beberapa titik data lain h_2 , sedangkan sisanya berjarak h_3 . Integrasi numerik dalam selang $[a, b]$ dilakukan dengan mengombinasikan kaidah integrasi yang sudah ada, misalnya kombinasi kaidah trapesium, kaidah 1/3 Simpson, dan kaidah 3/8 Simpson. Berdasarkan orde galatnya, kaidah 1/3 Simpson dan 3/8 Simpson lebih teliti daripada kaidah trapesium. Karena itu, kaidah 1/3 Simpson diterapkan bila jumlah upaselang yang bertetangga genap, sedangkan kaidah 3/8 Simpson diterapkan bila jumlah upaselang yang bertetangga ganjil dan kelipatan tiga. Sisanya dihitung dengan kaidah trapesium. Jadi, tata-ancangnya dapat diringkas sebagai berikut :

- untuk sejumlah upaselang berturutan yang berjarak sama adalah genap, gunakan kaidah 1/3 Simpson
- untuk sejumlah upaselang berturutan yang berjarak sama adalah kelipatan tiga, gunakan kaidah 3/8 Simpson
- untuk sejumlah upaselang yang tidak berjarak sama dengan tetangganya, gunakan kaidah trapesium

Contohnya dapat dilihat pada Gambar 6.12. Empat buah upaselang pertama berjarak sama, lebih baik menggunakan kaidah Simpson 1/3 (karena jumlah upaselang genap). Tiga buah upaselang berikutnya berjarak sama, lebih baik menggunakan kaidah Simpson 3/8 (karena jumlah upaselang kelipatan 3). Dua buah upaselang berikutnya masing-masing berbeda lebarnya, maka setiap upaselang dihitung integrasinya dengan kaidah trapesium.



Gambar 6.12 Kaidah 1/3 Simpson gabungan

6.4.5 Bentuk Umum Metode Newton-Cotes

Kaidah trapesium, kaidah Simpson 1/3, dan kaidah Simpson 3/8 adalah tiga buah metode integrasi numerik pertama dari metode Newton-Cotes. Masing-masingnya menghampiri fungsi $f(x)$ dengan polinom interpolasi derajat 1 (lanjar), derajat 2 (kuadratik), dan derajat 3 (kubik). Kita dapat menemukan kaidah-kaidah lainnya dengan menggunakan polinom interpolasi derajat 4, 5, 6, dan seterusnya.

Bentuk umum metode Newton-Cotes dapat ditulis sebagai

$$\int_a^b f(x)dx = \alpha h[w_0f_0 + w_1f_1 + w_2f_2 + \dots + w_nf_n] + E \quad (\text{P.6.40})$$

dalam hal ini $f_r = f(x_r)$, $x_r = a + rh$, dan $h = (b - a)/n$, E menyatakan galat, sedangkan α dan w_i adalah konstanta riil seperti yang didaftarkan pada tabel berikut ini :

n	α	$w_i, i = 1, 2, \dots, n$	E	Nama
1	1/2	1 1	-1/12 $h^3 f''$	Trapesium
2	1/3	1 4 1	-1/90 $h^5 f^{(4)}$	1/3 Simpson
3	3/8	1 3 3 1	-3/80 $h^5 f^{(4)}$	3/8 Simpson
4	2/45	7 32 12 32 7	-8/945 $h^7 f^{(6)}$	Boole
5	5/288	19 75 50 50 75 19	-275/12096 $h^7 f^{(4)}$	
6	1/140	41 216 27 272 27 216 41	-9/1400 $h^9 f^{(8)}$	
7	7/17280	751 3577 1323 2989 2989 1323 3577 751	-8183/518400 $h^9 f^{(8)}$	
8	8/14175	989 5888 -928 10496 -4540 10496 -928 5888 989	-2368/467775 $h^{11} f^{(10)}$	
9	9/89600	2857 15741 1080 19344 5788 5788 19344 1080 15741 2857	-173/14620 $h^{11} f^{(10)}$	
10	5/299376	16067 106300 -48525 272400 -260550 427368 -260550 272400 -48525 106300 16067	-1346350/ 326918592 $h^{13} f^{(12)}$	

Dari tabel di atas nilai-nilai w akan semakin besar dengan membesarnya n . Dari teori galat sudah diketahui bahwa pengurangan bilangan yang besar dengan bilangan yang kecil di dalam komputer dapat menyebabkan galat pembulatan. Karena alasan itu, maka metode-metode Newton-Cotes orde tinggi kurang disukai. Alasan lainnya,

orde metode menyatakan ketelitian hanya jika ranah integrasi $[a, b]$ cukup kecil sehingga turunan fungsi hampir tetap di dalam ranah nilai tersebut. Sebagai implikasinya, metode Newton-Cotes orde tinggi tidak lebih teliti daripada metode orde rendah bila turunannya berubah secara signifikan di dalam ranah tersebut.

Sebagai contoh adalah perhitungan integrasi

$$L = \int_0^{\pi} \sqrt{[(2+2 \cos x)^2 + 4\sin^2 x]} dx = \dots$$

Hasil perhitungan L dengan metode Newton-Cotes orde $n = 2$ sampai orde $n = 10$ adalah:

$n = 2, L = 8.01823$	$n = 7, L = 8.00000$
$n = 3, L = 8.00803$	$n = 8, L = 8.00000$
$n = 4, L = 7.99993$	$n = 9, L = 8.00197$
$n = 5, L = 7.99996$	$n = 10, L = 7.99201$
$n = 6, L = 8.00000$	Nilai integrasi sejati $L = 8.00000$

Hasil di atas menggambarkan hasil integrasi yang nilainya semakin tidak bagus dengan semakin tingginya orde metode Newton-Cotes (n). Dari $n = 2$ sampai $n = 8$, nilai L mendekati hasil sejati, 8.00000. Setelah $n = 8$, galatnya meningkat. Peningkatan galat disebabkan oleh galat penambahan dan pengurangan bilangan-bilangan yang sangat besar di dalam metodenya [NAK92]

6.5 Singula ritas

Kita akan kesulitan melakukan menghitung integrasi numerik apabila fungsi tidak terdefenisi di $x = t$, dalam hal ini $a < t < b$. Misalnya dalam menghitung integrasi

$$I = \int_0^1 \frac{\cos(x)}{\sqrt{x}} dx$$

fungsi $f(x) = \cos x / \sqrt{x}$ jelas tidak terdefinisi di $x = 0$ (ujung bawah selang). Begitu juga apabila perhitungan integrasi

$$I = \int_{0.5}^2 \frac{1}{x-1} dx$$

menggunakan $h = 0.1$, titik diskrit di $x = 1$ tidak dapat dihitung sebab fungsi $f(x) = 1/(x-1)$ tidak terdefinisi di $x = 1$. Fungsi yang tidak terdefinisi di $x = t$, untuk $a \leq t \leq b$, dinamakan fungsi **singular**.

Singularitas juga muncul pada fungsi yang turunannya tidak terdefinisi di $x = t$, untuk $a \leq t \leq b$. Misalnya hasil perhitungan integrasi $\int_0^1 \sqrt{x}$ memperlihatkan hasil yang menyimpang meskipun fungsi $f(x) = \sqrt{x}$ sendiri terdefinisi untuk semua $x = t$, untuk $a \leq t \leq b$. Penyimpangan ini dapat dijelaskan sebagai berikut. Misalkan integral $\int_0^1 \sqrt{x}$ dihitung dengan kaidah trapesium.

Tinjau kembali galat total pada kaidah trapesium:

$$\begin{aligned}
 E_{tot} &\approx -\frac{h^3}{12} (f_0'' + f_1'' + \dots + f_{n-1}'') \\
 &\approx -\frac{h^3}{12} \sum_{i=0}^{n-1} f_i'' \\
 &\approx -\frac{h^3}{12} \int_a^b f(x) dx \\
 &\approx -\frac{h^3}{12} [f'(b) - f'(a)] \tag{P.6.41}
 \end{aligned}$$

Persamaan (P.6.41) menyiratkan bahwa galat integrasi $\int_a^b f(x) dx$ akan besar apabila $f'(a)$ atau $f'(b)$ tidak ada.

Singularitas harus dihilangkan dengan cara memanipulasi persamaan fungsi sedemikian sehingga ia tidak singular lagi.

Contoh 6.4

Ubahlah fungsi integrasi

$$I = \int_0^1 \frac{\cos(x)}{\sqrt{x}} dx$$

sehingga menjadi tidak singular lagi.

Penyelesaian:

Fungsi $f(x) = \cos(x)/\sqrt{x}$ tidak terdefenisi di $x = 0$.

Misalkan

$$x = u^2 \rightarrow dx = 2u du$$

Batas-batas selang integrasi juga berubah

$$\begin{aligned} x = 0 &\rightarrow u = \sqrt{x} = 0 \\ x = 1 &\rightarrow u = \sqrt{x} = 1 \end{aligned}$$

maka

$$\begin{aligned} I &= \int_0^1 \frac{\cos(x)}{\sqrt{x}} dx \\ &= \int_0^1 \frac{\cos(u^2)}{u} (2u) du \\ I &= \int_0^1 2 \cos(u^2) du \quad \rightarrow \text{tidak singular lagi} \end{aligned} \quad \blacksquare$$

Contoh 6.5

Ubahlah fungsi integrasi

$$I = \int_0^1 \sqrt{x} dx$$

sehingga menjadi tidak singular lagi

Penyelesaian:

Fungsi $f(x) = \sqrt{x}$ singular sebab turunannya

$$f'(x) = \frac{1}{2\sqrt{x}}$$

tidak terdefinisi di $x = 0$

Misalkan

$$x = u^2 \rightarrow dx = 2u du$$

Batas-batas selang integrasi juga berubah

$$\begin{aligned}x = 0 &\rightarrow u = \sqrt{x} = 0 \\x = 1 &\rightarrow u = \sqrt{x} = 1\end{aligned}$$

maka

$$I = \int_0^1 \sqrt{x} \, dx = \int_0^1 2u^2 \, du \quad \rightarrow \text{tidak singular lagi}$$
■

Contoh 6.6

Ubahlah fungsi integrasi berikut sehingga menjadi tidak singular:

$$I = \int_0^1 \frac{dx}{\sqrt{(\sin x)(1-x^3)}}$$

Penyelesaian:

Fungsi $f(x) = 1/\sqrt{(\sin x)(1-x^3)}$ tidak terdefenisi di $x = 0$ dan $x = 1$. Pecah integral I menjadi dua bagian, I_1 dan I_2 :

$$I = \int_0^1 \frac{dx}{\sqrt{(\sin x)(1-x^3)}} = \int_0^a \frac{dx}{\sqrt{(\sin x)(1-x^3)}} + \int_a^1 \frac{dx}{\sqrt{(\sin x)(1-x^3)}}$$

I_1 , singular di $x = 0$ I_2 , singular $x = 1$

dengan $0 < a < 1$

Misalkan

$$x = u^2 \rightarrow dx = 2u \, du$$

Batas-batas integrasi

$$\begin{aligned}x = a &\rightarrow u = \sqrt{a} \\x = 0 &\rightarrow u = 0\end{aligned}$$

Maka,

$$I_1 = \int_0^{\sqrt{a}} \frac{2u \, du}{\sqrt{(\sin u^2)(1-u^6)}} = 2 \int_0^{\sqrt{a}} \frac{u/u}{\sqrt{\frac{(\sin u^2)(1-u^6)}{u^2}}} \, du$$

Mengingat

$$\lim_{u \rightarrow 0} \frac{\sin(u^2)}{u^2} = 1$$

maka

$$I_1 = 2 \int_0^{\sqrt{a}} \frac{1}{\sqrt{(1-u^6)}} du \rightarrow \text{tidak singular lagi}$$

$$I_2 = \int_a^1 \frac{1}{\sqrt{(\sin x)(1-x^3)}} dx \rightarrow \text{tidak dapat diterapkan pemisalan } x=u^2$$

Uraikan $(1-x^3)$ menjadi $(1-x)(1+x+x^2)$:

$$I_2 = \int_a^1 \frac{dx}{\sqrt{(\sin x)(1-x)(1+x+x^2)}}$$

Misalkan

$$1-x=u^2 \rightarrow -dx=2u du$$

Batas-batas integrasi :

$$x=1 \rightarrow u = \sqrt{1-x} = 0$$

$$x=a \rightarrow u = \sqrt{1-a}$$

$$\begin{aligned} I_2 &= \int_0^{\sqrt{1-a}} \frac{-2u du}{\sqrt{[\sin(1-u^2)]u^2[1+(1-u^2)+(1-u^2)^2]}} \\ &= 2 \int_0^{\sqrt{1-a}} \frac{u du}{\sqrt{[\sin(1-u^2)][3-3u^2-u^4]}} \\ &= 2 \int_0^{\sqrt{1-a}} \frac{du}{\sqrt{[\sin(1-u^2)][3-3u^2-u^4]}} \quad \rightarrow \text{tidak singular lagi} \quad \blacksquare \end{aligned}$$

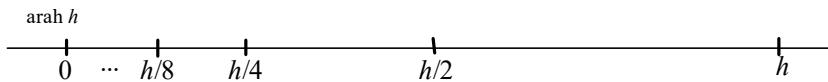
Cara lain penanganan singularitas dapat dilihat di [NAK93] halaman 140.

6.6 Penggunaan Ekstrapolasi untuk Integrasi

Misalkan $I(h)$ adalah perkiraan nilai integrasi dengan jarak antara titik data adalah h ($h < 1$). Dari persamaan galat kaidah integrasi (trapesium, Simpson 1/3, dll) yang dinyatakan dalam notasi orde:

$$E = O(h^p)$$

dapat dilihat bahwa galat E semakin kecil bila digunakan h yang semakin kecil, seperti yang ditunjukkan oleh diagram garis berikut:



Nilai sejati integrasi adalah bila $h = 0$, tetapi pemilihan $h = 0$ tidak mungkin kita lakukan di dalam rumus integrasi numerik sebab ia akan membuat nilai integrasi sama dengan 0. Yang dapat kita peroleh adalah perkiraan nilai integrasi yang lebih baik dengan melakukan ekstrapolasi ke $h = 0$. Ada dua macam metode ekstrapolasi yang digunakan untuk integrasi:

1. Ekstrapolasi Richardson
2. Ekstrapolasi Aitken

6.6.1 Ekstrapolasi Richardson

Pandang kembali kaidah trapesium

$$\int_a^b f(x)dx = \frac{h}{2}(f_0 + 2 \sum_{i=1}^n f_i + f_n) - \frac{(b-a)f''(t)}{12}h^2$$

yang dapat ditulis sebagai

$$\int_a^b f(x)dx = I(h) + Ch^2$$

dengan $I(h)$ adalah integrasi dengan menggunakan kaidah trapesium dengan jarak antar titik selebar h dan $C = \frac{(b-a)f''(t)}{12}$.

Secara umum, kaidah integrasi yang lain dapat kita tulis sebagai

$$\int_a^b f(x)dx = I(h) + Ch^q \quad (\text{P.6.42})$$

dengan C dan q adalah konstanta yang tidak bergantung pada h . Nilai q dapat ditentukan langsung dari orde galat kaidah integrasi, misalnya

$$\text{kaidah trapesium, } O(h^2) \rightarrow q = 2$$

$$\text{kaidah titik-tengah, } O(h^2) \rightarrow q = 2$$

$$\text{kaidah 1/3 Simpson, } O(h^4) \rightarrow q = 4$$

Tujuan ekstrapolasi Richardson ialah menghitung nilai integrasi yang lebih baik (*improve*) dibandingkan dengan I . Misalkan J adalah nilai integrasi yang lebih baik daripada I dengan jarak antar titik adalah h :

$$J = I(h) + Ch^q \quad (\text{P.6.43})$$

Ekstrapolasikan h menjadi $2h$, lalu hitung integrasi numeriknya

$$J = I(2h) + C(2h)^q \quad (\text{P.6.44})$$

Eliminasikan C dari kedua persamaan dengan menyamakan persamaan (P.6.43) dan persamaan (P.6.44):

$$I(h) + Ch^q = I(2h) + C(2h)^q$$

sehingga diperoleh

$$C = \frac{I(h) - I(2h)}{(2^q - 1)h^q} \quad (\text{P.6.45})$$

Sulihkan (P.6.45) ke dalam (P.6.43) untuk memperoleh:

$$J = I(h) + \frac{I(h) - I(2h)}{2^q - 1} \quad (\text{P.6.46})$$

yang merupakan persamaan **ekstrapolasi Richardson**. Ekstrapolasi Richardson dapat kita artikan sebagai berikut:

Mula-mula hitunglah nilai integrasi dengan kaidah yang sudah baku dengan jarak antar titik selebar h untuk mendapatkan $I(h)$, kemudian hitung kembali nilai integrasi dengan jarak antar titik selebar $2h$ untuk memperoleh $I(2h)$. Akhirnya, hitung nilai integrasi yang lebih baik dengan menggunakan persamaan (P.6.46).

Perhatikanlah bahwa jika pernyataan di atas dibalik, kita telah melakukan ekstrapolasi menuju $h = 0$, yaitu kita hitung $I(2h)$ lalu hitung $I(h)$. Urutan penggeraan ($I(2h)$ atau $I(h)$ lebih dulu) tidak mempengaruhi solusi akhirnya.

Sebagai contoh, bila $I(h)$ dan $I(2h)$ dihitung dengan kaidah trapesium ($q = 2$), maka ekstrapolasi Richardson-nya adalah

$$J = I(h) + \frac{1}{3} [I(h) - I(2h)] \quad (\text{P.6.47})$$

dan bila $I(h)$ dan $I(2h)$ dihitung dengan kaidah 1/3 Simpson ($q = 4$), maka ekstrapolasi Richardson-nya adalah

$$J = I(h) + \frac{1}{15} [I(h) - I(2h)] \quad (\text{P.6.48})$$

Perhatikanlah bahwa suku $1/3 [I(h) - I(2h)]$ pada persamaan (P.6.47) dan suku $1/15 [I(h) - I(2h)]$ pada persamaan (P.6.48) merupakan **faktor koreksi**. Artinya, nilai taksiran integrasi $I(h)$ dapat ditingkatkan menjadi nilai yang lebih baik dengan menambahkan faktor koreksi tersebut.

Contoh 6.7

Hitung kembali integral

$$\int_0^1 \frac{1}{1+x} dx$$

dengan menggunakan ekstrapolasi Richardson, yang dalam hal ini $I(h)$ dan $I(2h)$ dihitung dengan kaidah trapesium dan $h = 0.125$.

Penyelesaian:

Jumlah upaslang: $n = (1 - 0)/0.125 = 8$

Tabel titik-titik di dalam selang $[0,1]$ dengan $h = 0.125$:

r	x_r	f_r
0	0	1
1	0.125	0.88889
2	0.250	0.80000
3	0.375	0.72727

4	0.500	0.66667
5	0.625	0.61538
6	0.750	0.57143
7	0.875	0.53333
8	1.000	0.50000

$I(h)$ adalah nilai integrasi dengan kaidah trapesium menggunakan $h = 0.125$:

$$\begin{aligned} I(h) &= \int_0^1 \frac{1}{1+x} dx \approx h/2 (f_0 + 2f_1 + 2f_2 + 2f_3 + 2f_4 + 2f_5 + 2f_6 + 2f_7 + f_8) \\ &\approx 0.125/2 [1 + 2(0.88889) + 2(0.80000) + \dots + 0.50000] \\ &\approx 0.69412 \end{aligned}$$

$I(2h)$ adalah nilai integrasi dengan kaidah trapesium menggunakan $2h = 0.250$:

$$\begin{aligned} I(2h) &= \int_0^1 \frac{1}{1+x} dx \approx (2h)/2 (f_0 + 2f_2 + 2f_4 + 2f_6 + f_8) \\ &\approx 0.250/2 [1 + 2(0.80000) + 2(0.66667) + 2(0.57143) + 0.50000] \\ &\approx 0.69702 \end{aligned}$$

Nilai integrasi yang lebih baik, J , diperoleh dengan ekstrpolasi Richardson:

$$J = I(h) + \frac{I(h) - I(2h)}{2^q - 1}$$

yang dalam hal ini, $q = 2$, karena $I(h)$ dan $I(2h)$ dihitung dengan kaidah trapesium (yang mempunyai orde galat = 2)

$$J = 0.69412 + \frac{0.69412 - 0.69702}{2^2 - 1} = 0.69315$$

Jadi, taksiran nilai integrasi yang lebih baik adalah 0.69315. Bandingkan dengan nilai integrasi sejatinya:

$$\int_0^1 \frac{1}{1+x} dx = \ln(1+x) \Big|_{x=0}^{x=1} = \ln(2) - \ln(1) = 0.69314718$$

yang apabila dibulatkan ke dalam 5 angka bena, $\ln(0.69314718) = 0.69315$, hasilnya tepat sama dengan nilai integrasi yang dihitung dengan ekstrapolasi Richardson. ■

Contoh 6.8

Perlihatkan bahwa bila $I(h)$ dan $I(2h)$ dihitung dengan kaidah trapesium, maka persamaan ekstrapolasi Richardson menyatakan kaidah Simpson 1/3.

Penyelesaian:

Kaidah 1/3 Simpson untuk sepasang upaselang adalah (lihat Gambar 6.10) adalah

$$I = \int_0^{2h} f(x) dx$$

$I(h)$ dan $I(2h)$ adalah perkiraan hasil integrasi dengan kaidah trapesium menggunakan pias masing-masing selebar h dan $2h$:

$$\begin{aligned} I(h) &= \frac{h}{2} (f_0 + f_1) + \frac{h}{2} (f_1 + f_2) = \frac{h}{2} (f_0 + 2f_1 + f_2) \\ I(2h) &= \frac{(2h)}{2} (f_0 + f_2) = h(f_0 + f_2) \end{aligned}$$

Ekstrapolasi Richardson-nya ($q = 2$):

$$\begin{aligned} J &= I(h) + \frac{1}{3} [I(h) - I(2h)] \\ &= \frac{h}{2} (f_0 + 2f_1 + f_2) + \frac{1}{3} \left(\frac{h}{2} (f_0 + 2f_1 + f_2) - h(f_0 + f_2) \right) \\ &= \frac{h}{2} (f_0 + 2f_1 + f_2) + \frac{h}{6} (f_0 + 2f_1 + f_2) - \frac{h}{3} (f_0 + f_2) \\ &= \frac{h}{2} f_0 + hf_1 + \frac{h}{2} f_2 + \frac{h}{6} f_0 + \frac{h}{3} f_1 + \frac{h}{6} f_2 - \frac{h}{3} f_0 - \frac{h}{3} f_2 \\ &= \frac{h}{2} f_0 + \frac{h}{6} f_0 - \frac{h}{3} f_0 + hf_1 + \frac{h}{3} f_1 + \frac{h}{2} f_2 + \frac{h}{6} f_2 - \frac{h}{3} f_2 \\ &= \frac{h}{3} f_0 + \frac{4h}{3} f_1 + \frac{h}{3} f_2 \\ &= \frac{h}{3} (f_0 + 4f_1 + f_2) \end{aligned}$$

yang merupakan kaidah Simpson 1/3. Jadi, berdasarkan definisi ekstrapolasi Richardson, kaidah Simpson 1/3 adalah perkiraan integrasi yang lebih baik daripada kaidah trapesium. Contoh ini bersesuaian dengan jawaban Contoh 6.7, sebab nilai integrasi dengan ekstrapolasi Richardson sama dengan nilai integrasi yang diperoleh dengan kaidah Simpson 1/3 (lihat jawabannya pada Contoh 6.2). ■

Persamaan ekstrapolasi Richardson memenuhi semua kaidah integrasi yang dirurunkan dengan metode pias maupun metode Newton-Cotes. Kita pun dapat menurunkan kaidah integrasi numerik yang baru dengan menerapkan ekstrapolasi Richardson. Misalkan bila $I(h)$ dan $I(2h)$ dihitung dengan kaidah Simpson 1/3, maka ekstrapolasi Richardson menyatakan **kaidah Boole** (buktikan!):

$$J = \int_0^{4h} f(x) dx = \frac{2h}{45} (7f_0 + 32f_1 + 12f_2 + 32f_3 + 7f_4)$$

yang berarti kaidah Boole merupakan taksiran integrasi yang lebih baik daripada kaidah Simpson 1/3.

Bila ekstrapolasi Richardson diterapkan secara terus menerus, akan diperoleh nilai integrasi yang semakin lama semakin baik (galatnya semakin kecil). Metode penerapan ekstrapolasi Richardson seperti ini dinamakan **metode Romberg**.

6.6.2 Metode Romberg

Metode integrasi Romberg didasarkan pada perluasan ekstrapolasi Richardson untuk memperoleh nilai integrasi yang semakin baik. Sebagai catatan, setiap penerapan ekstrapolasi Richardson akan menaikkan order galat pada hasil solusinya sebesar dua:

$$O(h^{2N}) \rightarrow O(h^{2N+2})$$

Misalnya, bila $I(h)$ dan $I(2h)$ dihitung dengan kaidah trapesium yang berorde galat $O(h^2)$, maka ekstrapolasi Richardson menghasilkan kaidah Simpson 1/3 yang berorde $O(h^4)$. Selanjutnya, bila $I(h)$ dan $I(2h)$ dihitung dengan kaidah Simpson 1/3, ekstrapolasi Richardson menghasilkan kaidah Boole yang berorde $O(h^6)$.

Tinjau kembali persamaan ekstrapolasi Richardson:

$$J = I(h) + \frac{I(h) - I(2h)}{2^q - 1}$$

Misalkan I adalah nilai integrasi sejati yang dinyatakan sebagai

$$I = A_k + Ch^2 + Dh^4 + Eh^6 + \dots$$

yang dalam hal ini

$$h = (b - a)/n$$

dan

A_k = Perkiraan nilai integrasi dengan kaidah trapesium dan jumlah pias $n = 2^k$

Orde galat A_k adalah $O(h^2)$.

Sebagai contoh, selang $[a, b]$ dibagi menjadi 64 buah pias atau upaselang:

$$n = 64 = 2^6 \rightarrow k = 6 \ (0, 1, 2, 3, 4, 5, 6)$$

$$k = 0 \ (\text{artinya } n = 2^0 = 1 \text{ pias}, h_0 = (b-a)/1) \rightarrow A_0 = h_0/2 [f_0 + f_{64}]$$

$$k = 1 \ (\text{artinya } n = 2^1 = 2 \text{ pias}, h_1 = (b-a)/2) \rightarrow A_1 = h_1/2 [f_0 + 2f_{32} + f_{64}]$$

$$k = 2 \ (\text{artinya } n = 2^2 = 4 \text{ pias}, h_2 = (b-a)/4) \rightarrow A_2 = h_2/2 [f_0 + 2f_{16} + 2f_{32} + 2f_{48} + f_{64}]$$

$$k = 3 \ (\text{artinya } n = 2^3 = 8 \text{ pias}, h_3 = (b-a)/8) \rightarrow A_2 = h_3/2 [f_0 + 2f_8 + 2f_{16} + 2f_{24} + 2f_{32} + 2f_{40} + 2f_{48} + 2f_{56} + f_{64}]$$

...

$$k = 6 \ (\text{artinya } n = 2^6 = 64 \text{ pias}, h_6 = (b-a)/64) \rightarrow A_6 = h_6/2 [f_0 + 2f_1 + 2f_2 + \dots + 2f_{63} + f_{64}]$$

Arti dari setiap A_k adalah sebagai berikut:

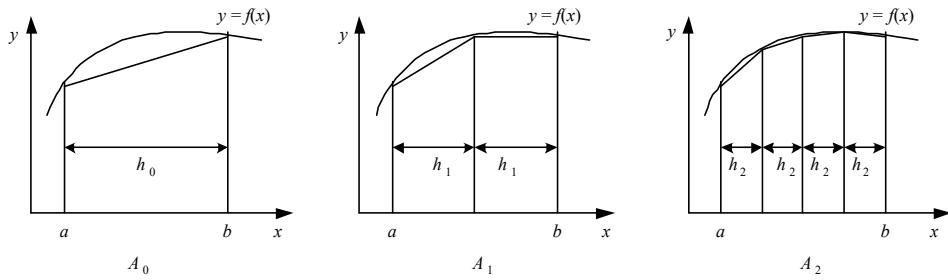
A_0 adalah taksiran nilai integrasi $I = \int_a^b f(x)dx$ dengan menggunakan kaidah trapesium dengan pembagian daerah integrasi menjadi $n = 2^0 = 1$ buah pias;

A_1 adalah taksiran nilai integrasi $I = \int_a^b f(x)dx$ dengan menggunakan kaidah trapesium dengan pembagian daerah integrasi menjadi $n = 2^1 = 2$ buah pias;

A_2 adalah taksiran nilai integrasi $I = \int_a^b f(x)dx$ dengan menggunakan kaidah trapesium dengan pembagian daerah integrasi menjadi $n = 2^2 = 4$ buah pias;

A_6 adalah taksiran nilai integrasi $I = \int_a^b f(x)dx$ dengan menggunakan kaidah trapesium dengan pembagian daerah integrasi menjadi $n = 2^6 = 64$ buah pias;

Tiga A_k yang pertama dilukiskan oleh Gambar 6.13.



Gambar 6.13 Luas daerah A_0, A_1, A_2, \dots , dengan jumlah upaselang masing-masing $n = 1, n = 2, n = 4, \dots$

Gunakan A_0, A_1, \dots, A_k pada persamaan ekstrapolasi Richardson untuk mendapatkan runtunan B_1, B_2, \dots, B_k , yaitu

$$B_k = A_k + \frac{A_k - A_{k-1}}{2^2 - 1}$$

Jadi, nilai I (yang lebih baik) sekarang adalah $I = B_k + D'h^4 + E'h^6 + \dots$ dengan orde galat B_k adalah $O(h^4)$.

Selanjutnya, gunakan B_1, B_2, \dots, B_k pada persamaan ekstrapolasi Richardson untuk mendapatkan runtunan C_2, C_3, \dots, C_k , yaitu

$$C_k = B_k + \frac{B_k - B_{k-1}}{2^4 - 1}$$

Jadi, nilai I (yang lebih baik) sekarang adalah $I = C_k + E''h^6 + \dots$ dengan orde galat C_k adalah $O(h^6)$.

Selanjutnya, gunakan C_2, C_3, \dots, C_k pada persamaan ekstrapolasi Richardson untuk mendapatkan runtunan D_3, D_4, \dots, D_k , yaitu

$$D_k = C_k + \frac{C_k - C_{k-1}}{2^6 - 1}$$

Jadi, nilai I (yang lebih baik) sekarang adalah $I = D_k + E'''h^8 + \dots$ dengan orde galat D_k adalah $O(h^8)$. Demikian seterusnya. Dari runtunan tersebut, diperoleh tabel yang dinamakan **tabel Romberg** seperti berikut ini:

$O(h^2)$	$O(h^4)$	$O(h^6)$	$O(h^8)$	$O(h^{10})$	$O(h^{12})$	$O(h^{14})$
A_0						
A_1	B_1					
A_2	B_2	C_2				
A_3	B_3	C_3	D_3			
A_4	B_4	C_4	D_4	E_4		
A_5	B_5	C_5	D_5	E_5	F_5	
A_6	B_6	C_6	D_6	E_6	F_6	G_6

↓
Nilai integrasi
yang lebih baik

Contoh 6.9

Hitung integral

$$\int_0^1 \frac{1}{1+x} dx$$

dengan metode Romberg ($n = 8$). Gunakan 5 angka bera.

Penyelesaian:

Jarak antar titik: $h = (1 - 0)/8 = 0.125$

Tabel titik-titik di dalam selang $[0,1]$ dengan $h = 0.125$:

r	x_r	f_r
0	0	1.0000
1	0.125	0.88889
2	0.250	0.80000
3	0.375	0.72727
4	0.500	0.66667
5	0.625	0.61538
6	0.750	0.57143
7	0.875	0.53333
8	1.000	0.50000

$$A_0 = h_0/2 [f_0 + f_8] = 1/2 (1 + 0.50000) = 0.75000$$

$$A_1 = h_1/2 [f_0 + 2f_4 + f_8] = 0.5/2[1 + 2(0.66667) + 0.50000] = 0.70833$$

$$A_2 = h_2/2 [f_0 + 2f_2 + 2f_4 + 2f_6 + f_8]$$

$$= 0.250/2[1 + 2(0.80000) + 2(0.66667) + 2(0.57143) + 0.50000] = 0.69702$$

$$A_3 = h_3/2 [f_0 + 2f_1 + 2f_2 + 2f_3 + 2f_4 + 2f_5 + 2f_6 + 2f_7 + f_8]$$

$$= 0.125/2[1 + 2(0.88889) + 2(0.80000) + \dots + 2(0.53333) + 0.50000]$$

$$= 0.69412$$

$$B_1 = A_1 + \frac{A_1 - A_0}{2^2 - 1} = 0.69445 \quad (A_k \text{ berorde } 2, \text{ jadi } q = 2)$$

$$B_2 = A_2 + \frac{A_2 - A_1}{2^2 - 1} = 0.69325$$

$$B_3 = A_3 + \frac{A_3 - A_2}{2^2 - 1} = 0.69315$$

$$C_2 = B_2 + \frac{B_2 - B_1}{2^4 - 1} = 0.69317 \quad (B_k \text{ berorde } 4, \text{ jadi } q = 4)$$

$$C_3 = B_3 + \frac{B_3 - B_2}{2^4 - 1} = 0.69314$$

$$D_3 = C_3 + \frac{C_3 - C_2}{2^6 - 1} = 0.69314 \quad (C_k \text{ berorde } 6, \text{ jadi } q = 6)$$

Tabel Romberg:

k	$O(h^2)$	$O(h^4)$	$O(h^6)$	$O(h^8)$
0	0.75000			
1	0.70833	0.69445		
2	0.69702	0.69325	0.69317	
3	0.69412	0.69315	0.69314	0.69314

Jadi, $\int_0^1 \frac{1}{1+x} dx \approx 0.69314$

(Bandingkan dengan solusi sejati $\int_0^1 \frac{1}{1+x} dx = 0.693145$) ■

6.6.3 Ekstrapolasi Aitken

Kita telah membahas ekstrapolasi Richardson yang dapat diringkas sebagai berikut:

$$\text{Jika } I = \int_a^b f(x)dx \approx I(h) + Ch^q$$

yang dalam hal ini,

h = lebar tiap upaselang atau pias (atau jarak antar titik)

C dan q adalah konstanta dengan q diketahui (C dapat dieliminir)

$I(h)$ adalah hampiran nilai nilai I

Ch^q adalah galat dari hampiran nilai I

maka

$$J = I(h) + \frac{1}{2^q - 1} [I(h) - I(2h)]$$

adalah perkiraan nilai integrasi yang lebih baik (*improve*) daripada I .

Timbul persoalan, bagaimana jika q tidak diketahui? Untuk kasus ini kita gunakan tiga buah perkiraan nilai I , yaitu $I(h)$, $I(2h)$, dan $I(4h)$:

$$J = I(h) + Ch^q \quad \rightarrow \quad C = \frac{J - I(h)}{h^q} \quad (\text{P.6.49})$$

$$J = I(2h) + C(2h)^q \quad \rightarrow \quad C = \frac{J - I(2h)}{(2h)^q} \quad (\text{P.6.50})$$

$$J = I(4h) + C(4h)^q \quad \rightarrow \quad C = \frac{J - I(4h)}{(4h)^q} \quad (\text{P.6.51})$$

Eliminasikan nilai C dan q dengan menyamakan persamaan (P.6.49) dan (P.6.50)

$$\frac{J - I(h)}{h^q} = \frac{J - I(2h)}{(2h)^q}$$

$$\frac{J - I(h)}{J - I(2h)} = \frac{h^q}{2^q h^q} = \frac{1}{2^q} \quad (\text{P.6.52})$$

dan menyamakan persamaan (P.6.50) dan (P.6.51)

$$\frac{J - I(2h)}{J - I(4h)} = \frac{(2h)^q}{(4h)^q} = \frac{1}{2^q} \quad (\text{P.6.53})$$

Persamaan (P.6.52) sama dengan persamaan (P.6.53):

$$\frac{J - I(h)}{J - I(2h)} = \frac{J - I(2h)}{J - I(4h)} \quad (\text{P.6.54})$$

kali silangkan kedua ruas persamaan (P.6.54)

$$J^2 - J I(h) - J I(4h) + I(h) I(4h) = J^2 - 2J I(2h) + [I(2h)]^2$$

$$J = \frac{I(h) I(4h) - [I(2h)]^2}{I(h) - 2 I(2h) + I(4h)}$$

atau

$$J = I(h) - \frac{[I(h) - I(2h)]^2}{I(h) - 2 I(2h) + I(4h)} \quad (\text{P.6.55})$$

Persamaan (P.6.55) ini dinamakan persamaan **ekstrapolasi Aitken** [NOB72].

Sekarang, tinjau kembali:

$$\begin{aligned} J &= I(h) + Ch^q \\ J &= I(2h) + C(2h)^q \\ \hline 0 &= I(h) - I(2h) + Ch^q - C(2h)^q \\ I(h) - I(2h) &= C(2h)^q - Ch^q \end{aligned} \quad (\text{P.6.56})$$

$$\begin{aligned} J &= I(2h) + C(2h)^q \\ J &= I(4h) + C(4h)^q \\ \hline 0 &= I(2h) - I(4h) + C(2h)^q - C(4h)^q \\ I(2h) - I(4h) &= C(4h)^q - C(2h)^q \end{aligned} \quad (\text{P.6.57})$$

Bagi persamaan (P.6.57) dengan persamaan (P.6.56):

$$\frac{I(2h) - I(4h)}{I(h) - I(2h)} = \frac{C(2h)^q - C(4h)^q}{Ch^q - C(2h)^q} = 2^q \quad (\text{P.6.58})$$

Besaran C pada persamaan (P.6.58) dapat dihilangkan menjadi

$$t = \frac{I(2h) - I(4h)}{I(h) - I(2h)} = 2^q \quad (\text{P.6.59})$$

Tinjau kembali persamaan (P.6.55) yang dapat ditulis ulang sebagai

$$\begin{aligned} J &= I(h) - \frac{I(h) - I(2h)}{\frac{I(h) - 2I(2h) + I(4h)}{I(h) - 2I(h)}} \\ &= I(h) - \frac{I(h) - I(2h)}{\frac{I - I(2h) + I(4h)}{I(h) - I(2h)}} \\ &= I(h) - \frac{I(h) - I(2h)}{1-t} \\ &= I(h) + \frac{I(h) - I(2h)}{t-1} \end{aligned}$$

Jadi,

$$J = I(h) + \frac{I(h) - I(2h)}{t-1} \quad (\text{P.6.60})$$

yang "mirip" dengan persamaan ekstrapolasi Richardson. Ekstrapolasi Aitken akan tepat sama dengan ekstrapolasi Richardson jika nilai teoritis

$$t = 2^q$$

tepat sama dengan nilai empirik

$$t = \frac{I(2h) - I(4h)}{I(h) - I(2h)}$$

Perbedaan antara kedua metode ekstrapolasi muncul bergantung kepada apakah kita mengetahui nilai q atau tidak. Hal ini diringkas dalam prosedur berikut:

Prosedur praktis:

1. Hitung $I(4h)$, $I(2h)$, dan $I(h)$
2. Hitung nilai empirik $t = \frac{I(2h) - I(4h)}{I(h) - I(2h)}$
3. Hitung nilai teoritik $t = 2^q$ (bila q diketahui)
4. Jika t teoritik $\neq t$ empirik harus kita bertanya "mengapa?"
5. Gunakan ekstrapolasi Aitken (P.6.59) dengan nilai empirik t atau ekstrapolasi Richardson (P.6.45) dengan q .

Contoh 6.10

[NOB72] Hitung $\int_0^1 \sqrt{x} dx$ sampai lima angka bena dengan menggunakan kaidah 1/3 Simpson (Gunakan $h = 1/8$)!

Penyelesaian:

$$\begin{aligned} 4h &= 4 \times 1/8 = 1/2 \rightarrow I(4h) = I(1/2) = \frac{1/2 I}{3} (f_0 + 4f_{1/2} + f_1) \\ &= 1/6 (0 + 4\sqrt{1/2} + \sqrt{1}) \\ &= 0.63807 \end{aligned}$$

$$\begin{aligned} 2h &= 2 \times 1/8 = 1/4 \rightarrow I(2h) = I(1/4) = \frac{1/4 I}{3} (f_0 + 4f_{1/4} + 2f_{1/2} + 4f_{3/4} + f_1) \\ &= 1/12 (0 + 4\sqrt{1/4} + 2\sqrt{1/2} + 4\sqrt{3/4} + 1) \\ &= 0.65653 \end{aligned}$$

$$\begin{aligned} h = 1/8 \rightarrow I(h) &= I(1/8) = \frac{1/8 I}{3} (f_0 + 4f_{1/8} + 2f_{2/8} + 4f_{3/8} + 2f_{4/8} + 4f_{5/8} + 2f_{6/8} + 4f_{7/8} + f_1) \\ &= 0.66308 \end{aligned}$$

$$\text{empirik} \rightarrow t = \frac{I(2h) - I(4h)}{I(h) - I(2h)} = \frac{I(1/4) - I(1/2)}{I(1/8) - I(1/4)} = 2.82$$

$$\text{teoritik} \rightarrow t = 2^q = 2^4 = 16 \text{ (yang diharapkan)}$$

Mengapa t teoritik tidak sama dengan t empirik? Perbedaan ini timbul sebab fungsi turunan \sqrt{x} tidak terdefinisi di $x = 0$ (singular). Karena itu, nilai t teoritik ($t = 16$) tidak dapat

dipegang, sehingga ekstrapolasi Richardson (P.6.46) tidak dapat digunakan untuk menghitung perkiraan nilai integrasi yang lebih baik. Jadi, gunakan ekstrapolasi Aitken (P.6.60) dengan nilai t empirik untuk menghitung perkiraan nilai integrasi yang lebih baik:

$$\begin{aligned} J &= I(1/8) + \frac{I(1/8) - I(1/4)}{2.82 - 1} \\ &= 0.66308 + \frac{1}{1.82} [0.66308 - 0.65653] \\ &= 0.66668 \end{aligned}$$

Bandingkan solusi ini dengan solusi sejatinya = 0.66667. Perhatikan, kalau kita menggunakan ekstrapolasi Richardson dengan t teoritik ($t = 16$), maka solusinya

$$\begin{aligned} J &= I(1/8) + \frac{I(1/8) - I(1/4)}{2^4 - 1} \\ &= 0.66308 + \frac{1}{15} [0.66308 - 0.65653] \\ &= 0.66352 \end{aligned}$$

yang cukup berbeda jauh dengan solusi eksak. Karena itu, hasil integrasi dengan ekstrapolasi Aitken yang dapat diterima, yaitu 0.66668. ■

6.7 Integral Ganda

Dalam bidang teknik, integral sering muncul dalam bentuk integral ganda dua (atau lipat dua) atau integral ganda tiga (lipat tiga). Misalkan kita tinjau untuk integral lipat dua. Integral lipat dua didefinisikan sebagai

$$\iint_A f(x, y) dA = \int_a^b \left[\int_c^d f(x, y) dy \right] dx = \int_c^d \left[\int_a^b f(x, y) dx \right] dy \quad (\text{P.6.61})$$

Tafsiran geometri dari integral ganda adalah menghitung volume ruang di bawah permukaan kurva $f(x,y)$ yang alasnya adalah berupa bidang yang dibatasi oleh garis-garis $x = a$, $x = b$, $y = c$, dan $y = d$. Volume benda berdimensi tiga adalah

$$V = \text{luas alas} \times \text{tinggi}$$

Kaidah-kaidah integrasi numerik yang telah kita bahas dapat dipakai untuk menghitung integral ganda. Jika pada fungsi dengan satu peubah, $y = f(x)$, luas daerah dihampiri dengan pias-pias yang berbentuk segiempat atau trapesium, maka pada fungsi dengan dua peubah, $z = f(x, y)$, volume ruang dihampiri dengan balok-balok yang berbentuk segiempat atau trapesium.

Solusi integral lipat dua diperoleh dengan melakukan integrasi dua kali, pertama dalam arah x (dalam hal ini nilai, nilai y tetap), selanjutnya dalam arah y (dalam hal ini, nilai x tetap), atau sebaliknya. Dalam arah x berarti kita menghitung luas alas benda, sedangkan dalam arah y berarti kita mengalikan alas dengan tinggi untuk memperoleh volume benda. Tinggi benda dinyatakan secara tidak langsung dengan koefisien-koefisien w_i pada persamaan (P.6.40).

Misalkan integrasi dalam arah x dihitung dengan kaidah trapesium, dan integrasi dalam arah y dihitung dengan kaidah Simpson 1/3. Maka

$$\begin{aligned}
 \int_c^d \int_a^b [f(x, y)dx]dy &\approx \sum_{j=1}^m v_j \sum_{i=1}^n w_i f_{ij} \\
 &\approx \frac{\Delta y}{3} \left[\frac{\Delta x}{2} (f_{0,0} + 2f_{1,0} + 2f_{2,0} + \dots + 2f_{n-1,0} + f_{n,0}) + \right. \\
 &\quad + 4 \times \frac{\Delta x}{2} (f_{0,1} + 2f_{1,1} + 2f_{2,1} + \dots + 2f_{n-1,1} + f_{n,1}) \\
 &\quad + 2 \times \frac{\Delta x}{2} (f_{0,2} + 2f_{1,2} + 2f_{2,2} + \dots + 2f_{n-1,2} + f_{n,2}) \\
 &\quad \dots \\
 &\quad + 2 \times \frac{\Delta x}{2} (f_{0,m-2} + 2f_{1,m-2} + 2f_{2,m-2} + \dots + 2f_{n-1,m-2} + f_{n,m-2}) \\
 &\quad + 4 \times \frac{\Delta x}{2} (f_{0,m-1} + 2f_{1,m-1} + 2f_{2,m-1} + \dots + 2f_{n-1,m-1} + f_{n,m-1}) \\
 &\quad \left. + \frac{\Delta x}{2} (f_{0,m} + 2f_{1,m} + 2f_{2,m} + \dots + 2f_{n-1,0} + f_{n,m}) \right] \quad (\text{P.6.62})
 \end{aligned}$$

dengan

- Δx = jarak antar titik dalam arah x ,
- Δy = jarak antar titik dalam arah y ,
- n = jumlah titik diskrit dalam arah x ,
- m = jumlah titik diskrit dalam arah y .

Contoh 6.11

Diberikan tabel $f(x,y)$ sebagai berikut:

$x \backslash y$	0.2	0.3	0.4	0.5	0.6
1.5	0.990	1.524	2.045	2.549	3.031
2.0	1.658	2.384	3.177	3.943	4.672
2.5	2.520	3.800	5.044	6.241	7.379
3.0	4.090	6.136	8.122	10.030	11.841

Hitung $\int_{0.2}^{0.6} \int_{1.5}^{3.0} f(x,y) dx dy$ [GER85]

Penyelesaian:

Misalkan

- dalam arah x kita gunakan kaidah trapesium
- dalam arah y kita gunakan kaidah Simpson 1/3

Dalam arah x (y tetap):

$$y = 0.2 ; \int_{1.5}^{3.0} f(x,y) dx \approx \int_{1.5}^{3.0} f(x,0.2) dx$$

$$\approx \Delta x / 2 (f_{0,0} + 2f_{1,0} + 2f_{2,0} + f_{3,0})$$

$$\approx 0.5 / 2 (0.990 + 2 \times 1.658 + 2 \times 2.520 + 4.090)$$

$$\approx 3.3140$$

$$y = 0.3 ; \int_{1.5}^{3.0} f(x,y) dx \approx \int_{1.5}^{3.0} f(x,0.3) dx$$

$$\approx \Delta x / 2 (f_{0,1} + 2f_{1,1} + 2f_{2,1} + f_{3,1})$$

$$\approx 0.5 / 2 (1.524 + 2 (2.384 + 2 \times 3.800 + 6.136))$$

$$\approx 5.0070$$

$$y = 0.4 ; \int_{1.5}^{3.0} f(x,y) dx \approx \int_{1.5}^{3.0} f(x,0.4) dx \approx 6.6522$$

$$y = 0.5 ; \int_{1.5}^{3.0} f(x,y) dx \approx \int_{1.5}^{3.0} f(x,0.5) dx \approx 8.2368$$

$$y = 0.6 ; \int_{1.5}^{3.0} f(x,y) dx \approx \int_{1.5}^{3.0} f(x,0.6) dx \approx 9.7345$$

Dalam arah y :

$$\begin{aligned}\int_{0.2}^{0.6} f(x, y) dy &\approx \Delta y/3 (3.3140 + 4 \times 5.0070 + 2 \times 6.6522 + 4 \times 8.2368 + 9.7435) \\ &\approx 0.1/3 (3.3140 + 4 \times 5.0070 + 2 \times 6.6522 + 4 \times 8.2368 + 9.7435) \\ &\approx 2.6446\end{aligned}$$

Jadi,

$$\int_{0.2}^{0.6} \int_{1.5}^{3.0} f(x, y) dx dy \approx 2.6446 \quad \blacksquare$$

Cara perhitungan integral ganda dua di atas dapat dirampatkan (*generalized*) untuk integral ganda tiga

$$\iiint_R f(x, y, z) dR$$

maupun integral ganda yang lebih tinggi.

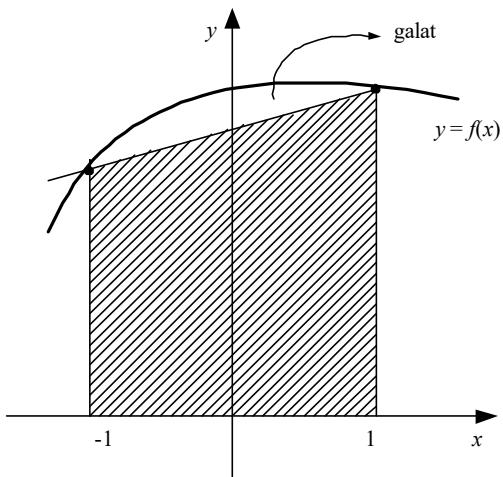
6.8 Kua dra tur G a uss

Sampai saat ini kita telah membahas kaidah integrasi yang berbasis titik-titik data diskrit dengan metode Newton-Cotes. Sebelum melakukan perhitungan integrasi, kita harus membentuk tabulasi titik-titik diskrit yang berjarak sama. Titik-titik diskrit tersebut harus berawal dan berakhir di ujung-ujung selang a dan b . Trapesium-trapesium yang menghampiri daerah integarsi harus berawal dan berakhir di ujung-ujung selang tersebut. Batasan ini mengakibatkan galat yang dihasilkan dengan mekanisme ini ternyata cukup besar.

Misalnya bila kita menggunakan kaidah trapesium untuk menghitung $\int_{-1}^1 f(x) dx$, maka daerah integrasi dalam selang $[-1, 1]$ (Gambar 6.14) dihampiri dengan sebuah trapesium yang luasnya adalah

$$I = \int_{-1}^1 f(x) dx \approx \frac{h}{2} [f(1) + f(-1)] \approx f(1) + f(-1) \quad (\text{P.6.63})$$

dengan $h = (1 - (-1)) = 2$.



Gambar 6.14 Integral $\int_{-1}^1 f(x)dx$ dihampiri dengan trapesium

Perhatikan kembali bahwa persamaan (P.6.63) dapat ditulis sebagai

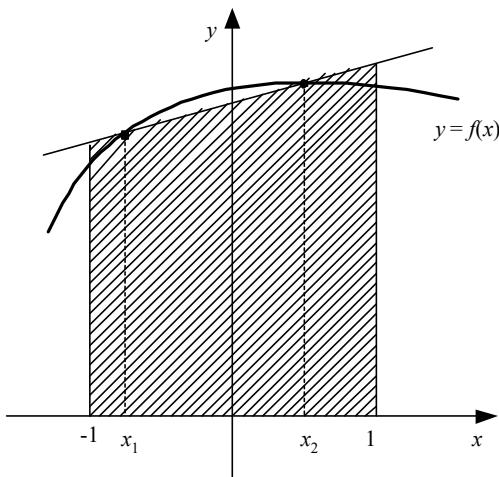
$$I \approx c_1 f(a) + c_2 f(b) \quad (\text{P.6.64})$$

dengan $a = -1$, $b = 1$, $c_1 = c_2 = h/2 = 2/2 = 1$.

Pendekatan integrasi yang berbeda dengan metode Newton-Cotes dikembangkan oleh Gauss dan dinamakan metode **kuadratur Gauss** (*Gaussian Quadrature*). Dengan metode kuadratur Gauss, batasan-batasan yang terdapat pada metode Newton-Cotes kuadratur dihilangkan. Di sini kita tidak perlu lagi menentukan titik-titik diskrit yang berjarak sama, tetapi nilai integrasi numerik cukup diperoleh dengan menghitung nilai fungsi $f(x)$ pada beberapa titik tertentu. Untuk memberi gambaran tentang kuadratur Gauss, perhatikan Gambar 6.15. Sebuah garis lurus ditarik menghubungkan dua titik sembarang pada kurva $y = f(x)$. Titik-titik tersebut diatur sedemikian sehingga garis lurus tersebut menyeimbangkan galat positif dan galat negatif. Luas daerah yang dihitung sekarang adalah luas daerah di bawah garis lurus, yang dinyatakan sebagai

$$I = \int_{-1}^1 f(x)dx \approx c_1 f(x_1) + c_2 f(x_2) \quad (\text{P.6.65})$$

dengan c_1 , c_2 , x_1 , dan x_2 adalah sembarang nilai. Persamaan (P.6.65) ini dinamakan persamaan kuadratur Gauss. Perhatikan bahwa bila dipilih $x_1 = -1$, $x_2 = 1$, dan $c_1 = c_2 = 1$, maka persamaan kuadratur Gauss (P.6.65) menjadi kaidah trapesium (P.6.63). Jadi, kaidah trapesium memenuhi persamaan kuadratur Gauss.



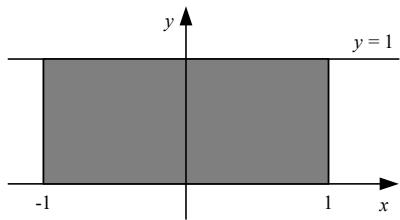
Gambar 6.15 Integral $\int_{-1}^1 f(x)dx$ dihampiri dengan kuadratur Gauss

Persamaan (P.6.65) mengandung empat buah peubah yang tidak diketahui (*unknown*), yaitu x_1 , x_2 , c_1 , dan c_2 . Kita harus memilih x_1 , x_2 , c_1 , dan c_2 sedemikian sehingga galat integrasinya minimum. Karena ada empat buah peubah yang tidak diketahui, maka kita harus mempunyai empat buah persamaan simultan yang mengandung x_1 , x_2 , c_1 , dan c_2 .

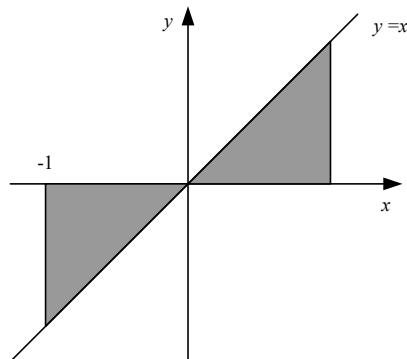
Di atas telah dikatakan bahwa kaidah trapesium bersesuaian dengan kuadratur Gauss. Dapat dilihat bahwa nilai integrasi numerik dengan kaidah trapesium akan tepat (galatnya = 0) untuk fungsi tetap dan fungsi lanjar. Misalnya untuk $f(x) = 1$ dan $f(x) = x$. Perhatikan Gambar 6.16. Dari dua buah fungsi tersebut, diperoleh dua persamaan:

$$f(x) = 1 \rightarrow \int_{-1}^1 1 dx = x \Big|_{x=-1}^{x=1} = 1 - (-1) = 2 = c_1 + c_2 \quad (\text{P.6.66})$$

$$f(x) = x \rightarrow \int_{-1}^1 x dx = \frac{1}{2} x^2 \Big|_{x=-1}^{x=1} = \frac{1}{2}(1)^2 - \frac{1}{2}(-1)^2 = 0 = c_1 x_1 + c_2 x_2 \quad (\text{P.6.67})$$



$$(a) \int_{-1}^1 dx$$



$$(b) \int_{-1}^1 x dx$$

Gambar 6.16 Integrasi yang bernilai sejati dengan kaidah trapesium

Kita memerlukan dua buah persamaan lagi agar x_1 , x_2 , c_1 , dan c_2 dapat ditentukan. Dari penalaran bahwa kaidah trapesium sejati untuk fungsi tetap dan fungsi lanjar, maka penalaran ini juga kita perluas dengan menambahkan anggapan bahwa integrasinya juga sejati untuk

$$f(x) = x^2 \text{ dan } f(x) = x^3.$$

Sekarang kita mendapatkan dua persamaan tambahan, yaitu

$$f(x) = x^2 \rightarrow \int_{-1}^1 x dx = \frac{1}{3} x^3 \Big|_{x=-1}^{x=1} = 2/3 = c_1 x_1^2 + c_2 x_2^2 \quad (\text{P.6.68})$$

dan

$$f(x) = x^3 \rightarrow \int_{-1}^1 x^2 dx = \frac{1}{4} x^4 \Big|_{x=-1}^{x=1} = 0 = c_1 x_1^3 + c_2 x_2^3 \quad (\text{P.6.69})$$

Sekarang, kita sudah mempunyai empat buah persamaan simultan

$$\begin{aligned} c_1 + c_2 &= 2 \\ c_1 x_1 + c_2 x_2 &= 0 \\ c_1 x_1^2 + c_2 x_2^2 &= 2/3 \\ c_1 x_1^3 + c_2 x_2^3 &= 0 \end{aligned}$$

yang bila dipecahkan menghasilkan:

$$\begin{aligned}c_1 &= c_2 = 1 \\x_1 &= 1/\sqrt{3} = 0.577350269 \\x_2 &= -1/\sqrt{3} = -0.577350269\end{aligned}$$

Jadi,

$$\int_{-1}^1 f(x) dx \approx f(1/\sqrt{3}) + f(-1/\sqrt{3}) \quad (\text{P.6.70})$$

Persamaan (P.6.70) dinamakan **kaidah Gauss-Legendre 2-titik**. Dengan kaidah ini, menghitung integral $f(x)$ di dalam selang $[-1, 1]$ cukup hanya dengan mengevaluasi nilai fungsi f di $x = 1/\sqrt{3}$ dan di $x = -1/\sqrt{3}$.

Transformasi $\int_a^b f(x) dx$ Menjadi $\int_{-1}^1 f(t) dt$

Untuk menghitung integrasi

$$I = \int_{-1}^1 f(x) dx$$

kita harus melakukan transformasi:

- a. selang $[a, b]$ menjadi selang $[-1, 1]$
- b. peubah x menjadi peubah t
- c. diferensial dx menjadi dt

Selang $[a, b]$ dan $[-1, 1]$ dilukiskan oleh diagram garis berikut:



Dari kedua diagram garis itu kita membuat perbandingan:

$$\begin{aligned}\Leftrightarrow \frac{x-a}{b-a} &= \frac{t-(-1)}{1-(-1)} \\ \Leftrightarrow \frac{x-a}{b-a} &= \frac{t+1}{2} \\ \Leftrightarrow 2x - 2a &= (t+1)(b-a) \\ \Leftrightarrow 2x &= (t+1)(b-a) + 2a\end{aligned}$$

$$\begin{aligned}
 \Leftrightarrow x &= \frac{bt - at + b - a + 2a}{2} \\
 &= \frac{a + b + bt - at}{2} \\
 \Leftrightarrow x &= \frac{(a+b)+(b-a)t}{2} \tag{P.6.71}
 \end{aligned}$$

Dari persaman (P.6.71), diperoleh diferensialnya

$$dx = \frac{b-a}{2} dt \tag{P.6.72}$$

Transformasikan $\int_a^b f(x)dx$ menjadi $\int_{-1}^1 f(t)dt$ dilakukan dengan menyulihkan (P.6.71) dan (P.6.72) ke dalam $\int_a^b f(x)dx$:

$$\int_a^b f(x)dx = \int_{-1}^1 f\left[\frac{(a+b)+(b-a)t}{2}\right] \frac{(b-a)}{2} dt = \frac{(b-a)}{2} \int_{-1}^1 f\left[\frac{(a+b)+(b-a)t}{2}\right] dt$$

Contoh 6.12

[MAT93] Hitung integral

$$\int_1^2 (x^2 + 1)dx$$

dengan kaidah Gauss-Legendre 2-titik.

Penyelesaian:

$$a = 1, \quad b = 2$$

$$x = \frac{(1+2)+(2-1)t}{2} = 1.5 + 0.5t$$

$$dx = \frac{2-1}{2} dt = 0.5 dt$$

Transformasikan $\int_1^2 f(x)dx$ menjadi $\int_{-1}^1 f(t)dt$:

$$\int_1^2 (x^2 + 1)dx = \int_{-1}^1 [(1.5 + 0.5t)^2 + 1]0.5dt = 0.5 \int_{-1}^1 [(1.5 + 0.5t)^2 + 1]dt$$

Jadi, dalam hal ini

$$f(t) = (1.5 + 0.5t)^2 + 1$$

maka

$$\begin{aligned} f(1/\sqrt{3}) &= (1.5 + 0.5 \times 1/\sqrt{3})^2 + 1 = 4.1993587371 \\ f(-1/\sqrt{3}) &= (1.5 + 0.5 \times -1/\sqrt{3})^2 + 1 = 2.4673079295 \end{aligned}$$

Dengan demikian

$$\begin{aligned} \int_1^2 (x^2 + 1)dx &= 0.5 \int_{-1}^1 (1.5 + 0.52t)^2 + 1 dt \approx 0.5 \times \{f(1/\sqrt{3}) + f(-1/\sqrt{3})\} \\ &\approx 3.333333333 \end{aligned}$$

Nilai integrasi sejatinya adalah:

$$\begin{aligned} \int_1^2 (x^2 + 1)dx &= \frac{1}{3}x^3 + x \Big|_{x=1}^{x=2} = (8/3 + 2) + (1/3 + 1) = (7/3 + 1) \\ &= 3.333333333 \end{aligned}$$

yang untuk kasus ini tepat sama sampai 10 angka bena dengan solusi hampirannya.

Program 6.5 : Integrasi $\int_1^2 (x^2 + 1)dx$ dengan kaidah Gauss-Legendre 2-Titik

```

procedure Gauss_Legendre_2_Titik(a, b: real; var I : real);
{ Menghitung  $\int_a^b f(x)dx$  dengan metode Gauss-Legendre 2-Titik
  K.Awal : harga a dan b sudah terdefenisi
  K.Akhir: I berisi hampiran integrasi
}
var
  f1, f2 : real;

  function f(t:real):real;
  { menghitung nilai f(t) untuk harga t yang telah terdefinisi }
  var
    x:real;
  begin
    x:=((a+b) + (b-a)*t)/2; {transformasi peubah}
    f:=x*x + 1;
  end;

```

```

begin
    f1:=f(sqrt(3)/3);
    f2:=f(-sqrt(3)/3);
    I:=(b-a)/2 * (f1 + f2);
end;

```

Dibandingkan dengan metode Newton-Cotes (trapesium, 1/3 Simpson, dll), kaidah Gauss-Legendre 2-titik lebih sederhana dan lebih mangkus dalam operasi aritmetika, karena Gauss-Legendre 2-titik hanya membutuhkan dua buah evaluasi fungsi. Selain itu, ketelitiannya lebih tinggi dibandingkan dengan metode Newton-Cotes. Namun, kaidah Gauss-Legendre tidak dapat digunakan jika fungsi $f(x)$ tidak

diketahui secara eksplisit, karena kita tidak dapat melakukan transformasi $\int_a^b f(x)dx$

menjadi $\int_{-1}^1 f(t)dt$. Untuk kasus seperti ini, jelas metode Newton-Cotes sebagai jalan keluarnya.

Kaidah Gauss-Legendre 3-Titik

Metode Gauss-Legendre 3-Titik dapat ditulis sebagai

$$I = \int_{-1}^1 f(x)dx \approx c_1 f(x_1) + c_2 f(x_2) + c_3 f(x_3)$$

Parameter x_1 , x_2 , x_3 , c_1 , c_2 , dan c_3 dapat ditemukan dengan membuat penalaran bahwa kuadratur Gauss bernilai tepat untuk 6 buah fungsi berikut:

$$\begin{aligned} f(x) &= 1; & f(x) &= x; & f(x) &= x^2 \\ f(x) &= x^3; & f(x) &= x^4; & f(x) &= x^5 \end{aligned}$$

Dengan cara yang sama seperti pada penurunan kaidah Gauss-Legendre 2-titik, diperoleh 6 buah persamaan simultan yang solusinya adalah

$$\begin{array}{ll} c_1 = 5/9; & x_1 = -\sqrt{3}/5 \\ c_2 = 8/9; & x_2 = 0 \\ c_3 = 5/9; & x_3 = \sqrt{3}/5 \end{array}$$

Jadi,

$$\int_{-1}^1 f(x)dx \approx \frac{5}{9} f\left[-\sqrt{\frac{3}{5}}\right] + \frac{8}{9} f(0) + \frac{5}{9} f\left[\sqrt{\frac{3}{5}}\right] \quad (\text{P.6.73})$$

Kaidah Gauss-Legendre n -Titik

Penurunan kaidah Gauss-Legendre 2-titik dan Gauss-Legendre 3-titik dapat dirampatkan untuk menghasilkan kaidah Gauss-Legendre n-titik

$$\int_{-1}^1 f(x)dt \approx c_1 f(x_1) + c_2 f(x_2) + \dots + c_n f(x_n) \quad (\text{P.6.74})$$

Nilai-nilai c_i dan x_i dapat dilihat pada tabel berikut ini:

Metode Gauss-Legendre n-titik			
<i>n</i>	Faktor bobot	Argumen fungsi	Galat pemotongan
2	$c_1 = 1.000000000$ $c_2 = 1.000000000$	$x_1 = -0.577350269$ $x_2 = 0.577350269$	$\approx f^{(4)}(c)$
3	$c_1 = 0.555555556$ $c_2 = 0.888888889$ $c_3 = 0.555555556$	$x_1 = -0.774596669$ $x_2 = 0$ $x_3 = 0.774596669$	$\approx f^{(6)}(c)$
4	$c_1 = 0.347854845$ $c_2 = 0.652145155$ $c_3 = 0.652145155$ $c_4 = 0.347854845$	$x_1 = -0.861136312$ $x_2 = -0.339981044$ $x_3 = 0.339981044$ $x_4 = 0.861136312$	$\approx f^{(8)}(c)$
5	$c_1 = 0.236926885$ $c_2 = 0.478628670$ $c_3 = 0.568888889$ $c_4 = 0.478628670$ $c_5 = 0.236926885$	$x_1 = -0.906179846$ $x_2 = -0.538469310$ $x_3 = 0$ $x_4 = 0.538469310$ $x_5 = 0.906179846$	$\approx f^{(10)}(c)$
6	$c_1 = 0.171324492$ $c_2 = 0.360761573$ $c_3 = 0.467913935$ $c_4 = 0.467913935$ $c_5 = 0.360761573$ $c_6 = 0.171324492$	$x_1 = -0.932469514$ $x_2 = -0.661209386$ $x_3 = -0.238619186$ $x_4 = 0.238619186$ $x_5 = 0.661209386$ $x_6 = 0.932469514$	$\approx f^{(12)}(c)$

6.9 Contoh Soal Terapan

Seorang penerjun payung terjun dari sebuah pesawat. Kecepatan penerjun sebagai fungsi dari waktu adalah [CHA91]:

$$v(t) = \frac{gm}{c} (1 - e^{-(c/m)t})$$

yang dalam hal ini

v = kecepatan penerjun dalam m/dt

g = tetapan gravitasi = 9.8 m/dt²

m = massa penerjun = 68.1 kg

c = koefisien tahanan udara = 12.5 kg/detik

Misalkan kita ingin mengetahui seberapa jauh penerjun telah jatuh setelah waktu tertentu t . Karena kecepatan merupakan turunan pertama dari fungsi jarak, maka jarak penerjun dari titik terjun ($t = 0$) adalah :

$$d = \int_0^t v(t) dt = \int_0^t \frac{gm}{c} (1 - e^{-(c/m)t}) dt$$

Hitung seberapa jauh penerjun telah jatuh setelah waktu $t=10$ detik dengan bermacam-macam metode integrasi numerik.

Penyelesaian:

Persoalan kita adalah menghitung integrasi

$$d = \int_0^{10} \frac{gm}{c} (1 - e^{-(c/m)t}) dt$$

dengan

v = kecepatan penerjun dalam m/dt

g = percepatan gravitasi = 9.8 m/dt²

m = massa penerjun = 68.1 kg

c = koefisien tahanan udara = 12.5 kg/detik

Nilai d dengan bermacam-macam metode integrasi numerik diringkas dalam tabel berikut:

Metode Integrasi	d (meter)	Keterangan
Trapesium	289.4309571611	$n = 128$
Titik-tengah	289.4372411810	$n = 128$
Simpson 1/3	289.4351464539	$n = 128$
Simpson 3/8	289.4351465013	$n = 243$
Romberg	289.4351465113	$n = 128$
Gauss-Legendre 2-Titik	290.0144778200	
Gauss-Legendre 3-Titik	289.4392972900	
Gauss-Legendre 4-Titik	289.4351622600	

Dari tabel di atas terlihat perbaikan hasil integrasi dimulai setelah kaidah titik-tengah. Mulai dari kaidah Simpson 1/3 sampai metode Romberg, nilai integrasi semakin diperbaiki. Pada contoh ini, hasil integrasi dengan kaidah Simpson 3/8 tidak dapat dibandingkan karena jumlah pias n tidak sama dengan kaidah integrasi lainnya, kecuali jika kita menggunakan n yang sama (n genap tetapi merupakan kelipatan tiga). Sedangkan hasil integrasi dengan kuadratur Gauss memperlihatkan perbaikan dengan semakin tingginya orde metode.

Kebutuhan yang paling mendasar bagi manusia ialah bagaimana mengatasi keterasingannya, untuk meninggalkan penjara kesendiriannya.
 (Erich Fromm - The Art of Loving)

Soal Latihan

1. Diketahui $f(x) = (4t - t^3)\exp(t^2)$, $0 \leq x \leq 2$ dan $n = 256$. Hitunglah $\int_0^2 f(x) dx$ dengan:
 - (a) kaidah trapesium
 - (b) kaidah Simpson 1/3
 - (c) kaidah titik-tengah
 - (d) metode Romberg
 - (e) kaidah Gauss-Legendre 3-titik dan Gauss-Legendre 4-titik.
2. Diketahui $f(x) = x^2 \cos(x^2)$, $1.5 \leq x \leq 2.5$ dan $h = 0.1$. Hitunglah $\int_{1.5}^{2.5} f(x) dx$ dengan:
 - (a) kaidah trapesium
 - (b) kaidah Simpson 1/3
 - (c) kaidah titik-tengah
 - (d) metode Romberg
 - (e) kaidah Gauss-Legendre 3-titik dan Gauss-Legendre 4-titik.
3. Turunkan rumus galat kaidah titik-tengah dan galat totalnya.
4. Turunkan rumus galat kaidah Simpson 3/8 dan galat totalnya.
5. Tentukan n (jumlah upaselang atau pias) sehingga kaidah trapesium memberikan nilai integrasi

$$\int_{-1}^1 \cos(2x) dx$$

kurang dari 0.000001.

6. (a) Dengan menerapkan ekstrapolasi Richardson, turunkan kaidah Boole untuk

$$\int_0^{4h} f(x) dx \text{ bila } I(h) \text{ dan } I(2h) \text{ dihitung dengan kaidah Simpson 1/3.}$$

- (b) Dengan menyatakan $I \approx I(h) + Ch^q$, turunkan rumus ekstrapolasi Richardson untuk menghitung $\int_0^{3h} f(x) dx$ dengan menggunakan titik-titik selebar h dan $3h$.

- (c) Berdasarkan rumus ekstrapolasi Richardson pada jawaban (b) di atas, turunkan kaidah 3/8 Simpson bila $I(h)$ dan $I(3h)$ dihitung dengan kaidah titik tengah.
7. (a) Dengan menyatakan $I \approx I(h) + Ch^q$, turunkan rumus ekstrapolasi Richardson

untuk menghitung $\int_0^{3h} f(x) dx$ dengan menggunakan titik-titik selebar h dan $3h$.

- (b) Dengan menerapkan ekstrapolasi Richardson pada rumus (a), turunkan kaidah integrasi baru yang galatnya berorde $O(h^7)$ untuk

$$\int_0^{6h} f(x) dx$$

bila $I(h)$ dan $I(3h)$ dihitung dengan kaidah Simpson 3/8.

- (c) Berdasarkan rumus ekstrapolasi Richardson pada jawaban (b) di atas, turunkan kaidah Simpson 3/8 bentuk lain bila $I(h)$ dan $I(3h)$ dihitung dengan kaidah titik tengah.

8. Rumus

$$\int_{-1}^1 f(x) (x-1)^2 dx = pf(a) + qf'(a)$$

akan tepat untuk polinom derajat ≤ 3 . Tentukan p , q , a , dan b .

9. Ubahlah bentuk integrasi di bawah ini agar tidak singular lagi :

- (i) $\int_{-1}^0 \cos(x)/x^{2/3} dx$
- (ii) $\int_0^1 dx/(1-x)^{1/2}$
- (iii) $\int_{-1}^2 (3x^2-2x)/(x^3 - x^2 + 2) dx$

10. Nilai integrasi untuk fungsi $f(x) = \sqrt[3]{x}$ singular dekat $x = 0$ dan tidak singular untuk x yang jauh dari nol. Untuk membuktikan pernyataan ini, lakukan perhitungan tangan (tanpa komputer) sampai 5 angka bena pada:

(a) $\int_1^{1.30} \sqrt[3]{x} dx$, kaidah 1/3 Simpson, $h = 0.05$. Bandingkan dengan nilai integrasi sejatinya.

(b) $\int_0^{0.30} \sqrt[3]{x} dx$, kaidah Simpson 1/3, $h = 0.05$. Bandingkan dengan nilai integrasi sejatinya.

Sekarang, ubahlah fungsi $f(x)$ sehingga tidak singular lagi, lalu hitung kembali integrasi soal (a) dan (b) di atas.

11. Hitunglah $\int_0^1 \int_0^2 e^y \cos(x) dx dy$:

- (a) Gunakan kaidah Simpson 1/3 untuk kedua arah, $\Delta x = \Delta y = 0.1$
 (b) Gunakan kaidah Gauss-Legendre 4-titik untuk kedua arah

12. Susunlah rumus integrasi numerik dari bentuk berikut :

$$\int_{-2}^2 f(x) dx = af(-2) + bf(0) + cf(2)$$

yang nilai integrasinya tepat untuk polinom $f(x)$ derajat ≤ 2

13. Perlihatkan bahwa galat kaidah Gauss-Legendre 2-titik sebanding dengan $f^{(4)}(c)$, yang dalam hal ini $-1 < c < 1$. (Petunjuk : gunakan bantuan deret Taylor).

14. Ubahlah bentuk integrasi di bawah ini agar tidak singular lagi :

(i) $\int_{-1}^0 \cos(x)/x^{2/3} dx$

(ii) $\int_0^1 dx/(1-x)^{1/2}$

(iii) $\int_{-1}^2 (3x^2 - 2x)/(x^3 - x^2 + 2) dx$

15. Hitunglah secara analitis $\int_a^b x^3 dx$. Nyatakan jawaban anda dalam a dan b .

Perlihatkan bahwa bila integral tersebut diselesaikan dengan kaidah Simpson 1/3 hasilnya sama dengan nilai integrasi sejatinya.

16. Hitunglah $\int_0^1 \int_1^2 x e^y dx dy$:

- (a) Gunakan kaidah Simpson 1/3 untuk kedua arah, $\Delta x = \Delta y = 0.1$
- (b) Gunakan kaidah Gauss-Legendre 4 titik untuk kedua arah.

Bab 7

Turunan Numerik

Lebih banyak lagi yang terdapat di langit dan di bumi, Horatio, daripada
yang kau mimpikan di dalam filosofimu.
(Hamlet)

Setiap mahasiswa yang pernah mengambil kuliah kalkulus tentu masih ingat dengan turunan fungsi yang didefinisikan sebagai

$$f'(x) = \lim_{h \rightarrow 0} \frac{f(x+h) - f(x)}{h} \quad (\text{P.7.1})$$

Persoalan menghitung turunan fungsi cukup banyak muncul dalam bidang rekayasa. Misalnya dalam bidang pengolahan citra (*image processing*), turunan fungsi diterapkan untuk mendeteksi sisi (*edge*) obyek pada suatu citra (lihat bagian terakhir bab ini). Sementara dalam perhitungan numerik sendiri, turunan fungsi dalam orde yang lebih tinggi, f' , f'' , f''' , ..., kadang-kadang diperlukan. Misalnya untuk menghitung batas-batas galat interpolasi polinom dengan rumus

$$E(x) = \frac{(x-x_0)(x-x_1)(x-x_2)\dots(x-x_n)f^{(n+1)}(\xi)}{(n+1)!}, \quad x_0 \leq \xi \leq x_n$$

atau untuk menghitung galat integrasi numerik dengan aturan trapesium :

$$E(x) = \frac{-1}{12} (b-a) h^2 f''(t), \quad a \leq t \leq b$$

Bila persamaan fungsi $f(x)$ diberikan secara eksplisit, maka kita dapat menentukan fungsi turunannya, $f'(x)$, $f''(x)$, ..., $f^{(n+1)}(x)$, lalu menggunakannya untuk menghitung nilai turunan fungsi di $x = t$.

Seringkali fungsi $f(x)$ tidak diketahui secara eksplisit, tetapi kita hanya memiliki beberapa titik data saja. Pada kasus seperti ini kita tidak dapat menemukan nilai turunan fungsi secara analitik. Sebaliknya, pada kasus lain, meskipun $f(x)$ diketahui secara eksplisit tetapi bentuknya rumit sehingga menentukan fungsi turunannya merupakan pekerjaan yang tidak mangkus dan tidak praktis, misalnya pada fungsi-fungsi berikut ini :

$$(i) \quad f(x) = \frac{\sqrt{\cos(2x^2) + x \tan(3x)}}{\sin(x) + e^x - 2x / \cos(x)},$$

$$(ii) \quad f(x) = x e^{(2x+2)} \ln(4x^2),$$

(iii) dan sebagainya.

Untuk kedua kasus terakhir, perhitungan nilai turunan dapat dikerjakan secara numerik (*numerical differentiation* atau *numerical derivative*). Nilai turunan yang diperoleh merupakan nilai hampiran. Sebagaimana halnya pada integrasi numerik, perhitungan turunan numerik juga menggunakan nilai-nilai diskrit. Karena itu, fungsi dalam bentuk tabel merupakan bentuk alami untuk perhitungan turunan.

7.1 Persoalan Turunan Numerik

Persoalan turunan numerik ialah menentukan hampiran nilai turunan fungsi f yang diberikan dalam bentuk tabel. Meskipun metode numerik untuk menghitung turunan fungsi tersedia, tetapi perhitungan turunan sedapat mungkin dihindari. Alasannya, nilai turunan numerik umumnya kurang teliti dibandingkan dengan nilai fungsinya. Dalam kenyataannya, turunan adalah limit dari hasil bagi selisih: yaitu pengurangan dua buah nilai yang besar ($f(x+h) - f(x)$) dan membaginya dengan bilangan yang kecil (h). Pembagian ini dapat menghasilkan turunan dengan galat yang besar. Lagi pula, jika fungsi f dihampiri oleh polinom interpolasi p , selisih nilai fungsi mungkin kecil tetapi turunannya boleh jadi sangat berbeda dengan nilai turunan sejatinya. Hal ini masuk akal sebab turunan numerik bersifat "halus", dan ini berlawanan dengan integrasi numerik, yang tidak banyak dipengaruhi oleh ketidaktelitian nilai fungsi, karena integrasi pada dasarnya adalah proses penghalusan [KRE88].

7.2 Tiga Pendekatan dalam Menghitung Turunan Numerik

Misal diberikan nilai-nilai x di $x_0 - h$, x_0 , dan $x_0 + h$, serta nilai fungsi untuk nilai-nilai x tersebut. Titik-titik yang diperoleh adalah (x_{-1}, f_{-1}) , (x_0, f_0) , dan (x_1, f_1) , yang dalam hal ini $x_{-1} = x_0 - h$ dan $x_1 = x_0 + h$. Terdapat tiga pendekatan dalam menghitung nilai $f'(x_0)$:

1. Hampiran selisih-maju (*forward difference approximation*)

$$f'(x_0) = \frac{f(x_0 + h) - f(x_0)}{h} = \frac{f_1 - f_0}{h} \quad (\text{P.7.2})$$

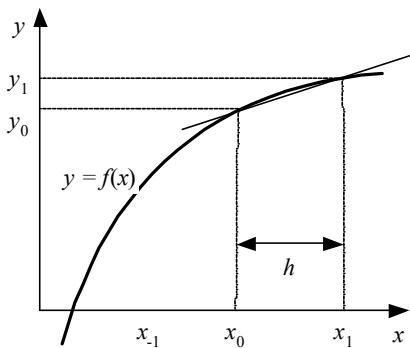
2. Hampiran selisih-mundur (*backward difference approximation*)

$$f'(x_0) = \frac{f(x_0) - f(x_0 - h)}{h} = \frac{f_0 - f_1}{h} \quad (\text{P.7.3})$$

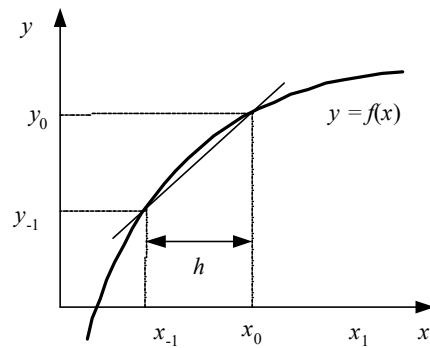
3. Hampiran selisih-pusat (*central difference approximation*)

$$f'(x_0) = \frac{f(x_0 + h) - f(x_0 - h)}{2h} = \frac{f_1 - f_{-1}}{2h} \quad (\text{P.7.3})$$

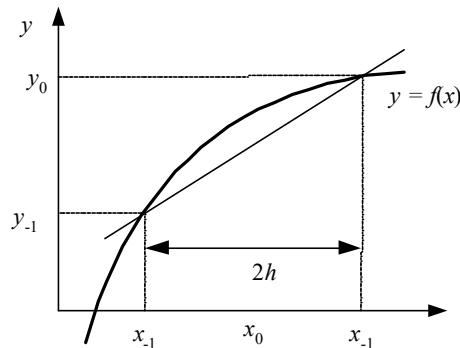
Tafsiran geometri dari ketiga pendekatan di atas diperlihatkan pada Gambar 7.1.



(a) Hampiran selisih-maju



(b) Hampiran selisih-mundur



(c) Hampiran selisih-pusat

Gambar 7.1 Tiga pendekatan dalam perhitungan turunan numerik

Rumus-rumus turunan numerik untuk ketiga pendekatan tersebut dapat diturunkan dengan dua cara, yaitu:

1. Dengan bantuan deret Taylor
2. Dengan hampiran polinom interpolasi

Kedua cara tersebut menghasilkan rumus yang sama.

7.3 Penurunan Rumus Turunan dengan Deret Taylor

Misalkan diberikan titik-titik (x_i, f_i) , $i = 0, 1, 2, \dots, n$, yang dalam hal ini

$$x_i = x_0 + ih$$

dan

$$f_i = f(x_i).$$

Kita ingin menghitung $f'(x)$, yang dalam hal ini $x = x_0 + sh$, $s \in R$ dengan ketiga pendekatan yang disebutkan di atas (maju, mundur, pusat).

(a) Hampiran selisih-maju

Uraikan $f(x_{i+1})$ di sekitar x_i :

$$\begin{aligned} f(x_{i+1}) &= f(x_i) + \frac{(x_{i+1} - x_i)}{1!} f'(x_i) + \frac{(x_{i+1} - x_i)^2}{2!} f''(x_i) + \dots \\ f_{i+1} &= f_i + hf'_i + h^2/2 f''_i + \dots \end{aligned} \quad (\text{P.7.4})$$

$$hf'_i = f_{i+1} - f_i - h^2/2 f''_i + \dots$$

$$f'_i = \frac{f_{i+1} - f_i}{h} - h/2 f''_i$$

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

yang dalam hal ini, $O(h) = h/2 f''(t)$, $x_i < t < x_{i+1}$

Untuk nilai-nilai f di x_0 dan x_1 persamaan rumusnya menjadi:

$$f'_0 = \frac{f_1 - f_0}{h} + O(h) \quad (\text{P.7.5})$$

yang dalam hal ini $O(h) = h/2 f''(t)$, $x_i < t < x_{i+1}$.

(b) Hampiran selisih-mundur

Uraikan $f(x_{i-1})$ di sekitar x_i :

$$\begin{aligned} f(x_{i-1}) &= f(x_i) + \frac{(x_{i+1} - x_i)}{1!} f'(x_i) + \frac{(x_{i+1} - x_i)^2}{2!} f''(x_i) + \dots \\ f_{i-1} &= f_i - hf'_i + h^2/2 f''_i + \dots \quad (\text{P.7.6}) \\ hf'_i &= f_i - f_{i-1} + h^2/2 f''_i + \dots \\ f'_i &= \frac{f_i - f_{i-1}}{h} - h/2 f''_i + \dots \\ f'_i &= \frac{f_i - f_{i-1}}{h} + O(h), \end{aligned}$$

yang dalam hal ini, $O(h) = -h/2 f''(t)$, $x_{i-1} < t < x_i$

Untuk nilai-nilai f di x_0 dan x_{-1} persamaan rumusnya menjadi:

$$f'_0 = \frac{f_0 - f_{-1}}{h} + O(h) \quad (\text{P.7.7})$$

yang dalam hal ini, $O(h) = -h/2 f''(t)$, $x_{i+1} < t < x_i$

(c) Hampiran selisih-pusat

Kurangkan persamaan (P.7.4) dengan persamaan (P.7.6):

$$\begin{aligned} f_{i+1} - f_{i-1} &= 2hf'_i + h^3/3 f'''_i + \dots \\ 2hf'_i &= f_{i+1} - f_{i-1} - h^3/3 f'''_i + \dots \end{aligned}$$

$$f_i' = \frac{f_{i+1} - f_{i-1}}{2h} - h^2/6 f_i''' + \dots$$

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

yang dalam hal ini, $O(h^2) = -h^2/6 f'''(t)$, $x_{i-1} < t < x_{i+1}$

Untuk nilai-nilai f di x_{-1} dan x_1 persamaan rumusnya menjadi:

$$f_o' = \frac{f_1 - f_{-1}}{2h} + O(h^2) \quad (\text{P.7.8})$$

yang dalam hal ini, $O(h^2) = -h/6 f'''(t)$, $x_{i-1} < t < x_{i+1}$.

Perhatikan, bahwa hampiran selisih-pusat lebih baik daripada dua hampiran sebelumnya, sebab orde galatnya adalah $O(h^2)$.

Rumus untuk Turunan Kedua, $f''(x)$, dengan Bantuan Deret Taylor

(a) Hampiran selisih-pusat

Tambahkan persamaan (P.7.4) dengan persamaan (P.7.6) di atas :

$$\begin{aligned} f_{i+1} + f_{i-1} &= 2f_i + h^2 f_i'' + h^4/12 f_i^{(4)} + \dots \\ f_{i+1} - 2f_i + f_{i-1} &= h^2 f_i'' + h^4/12 f_i^{(4)} \\ f_i'' &= \frac{f_{i+1} - 2f_i + f_{i-1}}{h^2} - h^2/12 f_i^{(4)} \end{aligned}$$

Jadi,

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

yang dalam hal ini, $O(h^2) = -h^2/12 f^{(4)}(t)$, $x_{i-1} < t < x_{i+1}$

Untuk nilai-nilai f di x_{-1} , x_0 , dan x_1 persamaan rumusnya menjadi:

$$f_0'' = \frac{f_1 - 2f_0 + f_{-1}}{h^2} + O(h^2) \quad (\text{P.7.9})$$

yang dalam hal ini $O(h^2) = -h^2/12 f^{(4)}(t)$, $x_{i-1} < t < x_{i+1}$.

(b) Hampiran selisih-mundur

Dengan cara yang sama seperti (a) di atas, diperoleh :

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

yang dalam hal ini $O(h) = h f''(t)$, $x_{i-2} < t < x_i$

Untuk nilai-nilai f di x_{-2} , x_{-1} , dan x_0 persamaan rumusnya :

$$f_{-2}'' = \frac{f_{-2} - 2f_{-1} + f_0}{h^2} + O(h), \quad (\text{P.7.10})$$

yang dalam hal ini, $O(h) = h f''(t)$, $x_{i-2} < t < x_i$

(c) Hampiran selisih-maju

Dengan cara yang sama seperti di atas, diperoleh :

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

yang dalam hal ini, $O(h) = -h f''(t)$, $x_i < t < x_{i+2}$

Untuk nilai-nilai f di x_0 , x_1 , dan x_2 persamaan rumusnya :

$$f_0'' = \frac{f_2 - 2f_1 + f_0}{h^2} + O(h), \quad (\text{P.7.11})$$

yang dalam hal ini, $O(h) = -h f''(t)$, $x_1 < t < x_{i+2}$.

7.4 Penurunan Rumus Turunan Numerik dengan Polinom Interpolasi

Misalkan diberikan titik-titik data berjarak sama,

$$x_i = x_0 + ih, \quad i = 0, 1, 2, \dots, n,$$

dan

$$x = x_0 + sh, \quad s \in R$$

adalah titik yang akan dicari nilai interpolasinya. Polinom Newton-Gregory yang menginterpolasi seluruh titik data tersebut adalah :

$$\begin{aligned} f(x) \approx p_n(x) &= f_0 + \frac{s\Delta f_0}{1!} + s(s-1) \frac{\Delta^2 f_0}{2!} + s(s-1)(s-2) \frac{\Delta^3 f_0}{3!} + \\ &\quad s(s-1)(s-2)\dots(s-n+1) \frac{\Delta^n f_0}{n!} \\ &= F(s) \end{aligned}$$

yang dalam hal ini, $s = (x-x_0)/h$.

Turunan pertama dari $f(x)$ adalah :

$$\begin{aligned} f'(x) &= df/dx = \frac{dF}{ds} \frac{ds}{dx} \\ &= (0 + \Delta f_0 + (s-1/2) \Delta^2 f_0 + (s^2/2 - s + 1/3) \Delta^3 f_0 + \dots) 1/h \\ &= 1/h (\Delta f_0 + (s-1/2) \Delta^2 f_0 + \text{galat}) \end{aligned} \quad (\text{P.7.12})$$

Berdasarkan (P.7.12), diperoleh rumus turunan numerik dengan ketiga pendekatan (maju, mundur, pusat) sebagai berikut:

(a) Hampiran selisih-maju

- bila digunakan titik-titik x_0 dan x_1 :

$$f'(x_0) = 1/h (\Delta f_0) = \frac{f_1 - f_0}{h} \quad (\text{P.7.13})$$

- bila digunakan titik-titik x_0 , x_1 , dan x_2 :

$$f'(x_0) = 1/h (\Delta f_0 + (s - 1/2) \Delta^2 f_0)$$

untuk titik $x_0 \rightarrow s = (x_0 - x_0)/h = 0$, sehingga

$$\begin{aligned} f'(x_0) &= 1/h (\Delta f_0 - 1/2 \Delta^2 f_0) \\ &= 1/h (\Delta f_0 - 1/2(\Delta f_1 - \Delta f_0)) \\ &= 1/h (3/2 \Delta f_0 - 1/2 \Delta f_1) \\ &= 1/h (3/2 f_1 - 3/2 f_0 - 1/2 f_2 + 1/2 f_1) \\ &= 1/h (-3/2 f_0 + 2 f_1 - 1/2 f_2) \end{aligned}$$

$$f'(x_0) = \frac{-3f_0 + 4f_1 - f_2}{2h} \quad (\text{P.7.13})$$

(b) Hampiran selisih-mundur

- polinom interpolasi: Newton-Gregory mundur
- bila digunakan titik-titik x_0 dan x_{-1} :

$$f'(x_0) = 1/h (\nabla f_0) = \frac{f_0 - f_{-1}}{h} \quad (\text{P.7.14})$$

(c) Hampiran selisih-pusat

- digunakan tiga titik x_0 , x_1 , dan x_2 :

$$f'(x_0) = 1/h (\Delta f_0 + (s - 1/2) \Delta^2 f_0)$$

untuk titik $x_1 \rightarrow s = (x_1 - x_0)/h = h/h = 1$, sehingga

$$\begin{aligned} f'(x_1) &= 1/h (\Delta f_0 + 1/2 \Delta^2 f_0) \\ &= 1/h (\Delta f_0 + 1/2(\Delta f_1 - \Delta f_0)) \\ &= 1/h (1/2 \Delta f_0 + 1/2 \Delta f_1) \\ &= 1/2h (f_1 - f_0 + f_2 - f_1) \end{aligned}$$

$$= \frac{f_2 - f_0}{2h}$$

untuk titik x_{-1} , x_0 , dan x_1 :

$$f'(x_0) = \frac{f_1 - f_{-1}}{2h} \quad (\text{P.7.15})$$

Rumus untuk Turunan Kedua, $f''(x)$, dengan Polinom Interpolasi

Turunan kedua f adalah

$$\begin{aligned} \frac{d^2 f}{dx^2} &= \frac{d}{ds} \left(\frac{df}{dx} \right) \frac{ds}{dx} \\ &= 1/h (0 + \Delta^2 f_0 + (s - 1) \Delta^3 f_0) \cdot 1/h \\ &= 1/h^2 (\Delta^2 f_0 + (s - 1) \Delta^3 f_0) \end{aligned}$$

Misalkan untuk hampiran selisih-pusat, titik-titik yang digunakan x_0 , x_1 , dan x_2 :

- pada titik $x_1 \rightarrow s = (x_1 - x_0)/h = h/h = 1$, sehingga

$$\begin{aligned} f''(x_1) &= 1/h^2 (\Delta^2 f_0 + (1 - 1) \Delta^3 f_0) \\ &= 1/h^2 (\Delta^2 f_0) \\ &= 1/h^2 (\Delta f_1 - \Delta f_0) \\ &= 1/h^2 (f_2 - f_1 + f_1 - f_0) \\ &= 1/h^2 (f_0 - 2f_1 + f_2) \end{aligned}$$

- untuk titik x_{-1} , x_0 , dan x_1 :

$$f''(x_0) = \frac{f_{-1} - 2f_0 + f_1}{h^2} \quad (\text{P.7.16})$$

7.5 Menentukan Orde Galat

Pada penurunan rumus turunan numerik dengan deret Taylor, kita dapat langsung memperoleh rumus galatnya. Tetapi dengan polinom interpolasi kita harus mencari rumus galat tersebut dengan bantuan deret Taylor.

Contohnya, kita menentukan rumus galat dan orde dari rumus turunan numerik hampir selisih-pusat:

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

Nyatakan E (galat) sebagai ruas kiri persamaan, lalu ekspansi ruas kanan dengan deret Taylor di sekitar x_0 :

$$\begin{aligned} E &= f'(x_0) - (f_1 - f_{-1})/2h \\ &= f'_0 - 1/2h [(f_0 + hf'_0 + h^2/2 f_0'' + h^3/6 f_0''' + \dots) \\ &\quad - (f_0 - hf'_0 + h^2/2 f_0'' - h^3/6 f_0''')] \\ &= f'_0 - 1/2h (2hf'_0 + h^3/3 f_0'''' + \dots) \\ &= f'_0 - f'_0 - h^2/6 f_0'''' + \dots \\ &= -h^2/6 f_0'''' + \dots \\ &= -h^2/6 f'''(t), \quad x_{-1} < t < x_1 \\ &= O(h^2) \end{aligned} \tag{P.7.17}$$

Jadi, hampiran selisih-pusat memiliki galat $E = -h^2/6 f'''(t)$, $x_{-1} < t < x_1$, dengan orde $O(h^2)$.

7.6 Program Menghitung Turunan

Program menghitung turunan numerik sangat sederhana. Rumus-rumus turunan dinyatakan sebagai fungsi. Di bawah ini tiga buah fungsi menghitung turunan pertama dengan rumus hampiran selisih-maju, hampiran selisih mundur, dan hampiran selisih-pusat.

Program 7.1 Menghitung turunan pertama dengan rumus hampiran selisih-maju, hampiran selisih-mundur, dan hampiran selisih-pusat.

```
function fAksen_maju(f0, f1, h : real):real;
{ Menghitung  $f'(x_0)$  dengan rumus hampiran selisih-maju }
begin
  fAksen_maju:=(f1-f0)/h;
end;

function fAksen_mundur(f_1, f0, h : real):real;
{ Menghitung  $f'(x_0)$  dengan rumus hampiran selisih-mundur }
begin
  fAksen_mundur:=(f0-f_1)/h;
end;

function fAksen_pusat(f_1, f1, h : real):real;
{ Menghitung  $f'(x_0)$  dengan rumus hampiran selisih-pusat }
begin
  fAksen_pusat:=(f1-f_1)/(2*h);
end;
```

7.7 Ringkasan Rumus-Rumus Turunan

Di bawah ini dirangkum beberapa rumus perhitungan turunan secara numerik, baik untuk turunan pertama, turunan kedua, dan seterusnya. Disertakan juga orde dari setiap rumus, dalam notasi O -besar. Rumus turunan dengan orde yang semakin tinggi menunjukkan nilai turunannya semakin teliti, namun jumlah komputasinya makin banyak (jumlah titik data yang diperlukan juga lebih banyak).

1. Rumus untuk turunan pertama

$$f'_0 = \frac{f_1 - f_0}{h} + O(h) \quad (\text{selisih-maju})$$

$$f'_0 = \frac{f_0 - f_{-1}}{h} + O(h) \quad (\text{selisih-mundur})$$

$$f'_0 = \frac{f_1 - f_{-1}}{2h} + O(h^2) \quad (\text{selisih-pusat})$$

$$f_0' = \frac{-3f_0 + 4f_1 - f_2}{2h} + O(h^2) \quad (\text{selisih-maju})$$

$$f_0' = \frac{-f_2 + 8f_1 - 8f_{-1} + f_{-2}}{12h} + O(h^4) \quad (\text{selisih-pusat})$$

2. Rumus untuk turunan kedua

$$f_0'' = \frac{f_1 - 2f_0 + f_{-1}}{h^2} + O(h^2) \quad (\text{selisih-pusat})$$

$$f_0'' = \frac{f_{-2} - 2f_{-1} + f_0}{h^2} + O(h) \quad (\text{selisih-mundur})$$

$$f_0'' = \frac{f_2 - 2f_1 + f_0}{h^2} + O(h) \quad (\text{selisih-maju})$$

$$f_0'' = \frac{-f_3 + 4f_2 - 5f_1 + 2f_0}{12h} + O(h^2) \quad (\text{selisih-maju})$$

$$f_0'' = \frac{-f_2 + 16f_1 - 30f_0 + 16f_{-1} - f_{-2}}{12h^2} + O(h^4) \quad (\text{selisih-pusat})$$

3. Rumus untuk turunan ketiga

$$f_0''' = \frac{f_3 - 3f_2 + 3f_1 - f_0}{h^3} + O(h) \quad (\text{selisih-maju})$$

$$f_0''' = \frac{f_2 - 2f_1 + 2f_{-1} - f_{-2}}{2h^3} + O(h^2) \quad (\text{selisih-pusat})$$

4. Rumus untuk turunan keempat

$$f_0^{(iv)} = \frac{f_4 - 4f_3 + 6f_2 - 4f_1 + f_0}{h^4} + O(h) \quad (\text{selisih-maju})$$

$$f_0^{(iv)} = \frac{f_2 - 4f_1 + 6f_0 - 4f_{-1} + f_{-2}}{h^4} + O(h^2) \quad (\text{selisih-pusat})$$

7.8 Contoh Perhitungan Turunan

Contoh 7.1

Diberikan data dalam bentuk tabel sebagai berikut :

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

- Hitunglah $f'(1.7)$ dengan rumus hampiran selisih-pusat orde $O(h^2)$ dan $O(h^4)$
- Hitunglah $f'(1.4)$ dengan rumus hampiran selisih-pusat orde $O(h^2)$
- Rumus apa yang digunakan untuk menghitung $f'(1.3)$ dan $f'(2.5)$?

Penyelesaian:

- Orde $O(h^2)$:

$$f'_0 = \frac{f_1 - f_{-1}}{2h}$$

Ambil titik-titik $x_{-1} = 1.5$ dan $x_1 = 1.9$, yang dalam hal ini $x_0 = 1.7$ terletak di tengah keduanya dengan $h = 0.2$.

$$f'(1.7) = \frac{6.686 - 4.482}{2(0.2)} = 5.510 \quad (\text{empat angka bena})$$

Orde $O(h^4)$:

$$f'_0 = \frac{-f_2 + 8f_1 - 8f_{-1} + f_2}{12h}$$

Ambil titik-titik $x_{-2} = 1.3$ dan $x_{-1} = 1.5$, $x_0 = 1.7$, dan $x_1 = 1.9$, dan $x_2 = 2.1$, yang dalam hal ini $x_0 = 1.7$ terletak di pertengahannya.

$$\begin{aligned} f'(1.7) &= \frac{-8.166 + 8(6.686) - 8(4.482) + 3.669}{12(0.2)} \\ &= 5.473 \quad (4 \text{ angka bena}) \end{aligned}$$

(b) Orde $O(h^2)$:

Ambil titik-titik $x_{-1} = 1.3$ dan $x_1 = 1.5$, yang dalam hal ini $x_0 = 1.4$ terletak di tengahnya dan $h = 0.1$.

$$f'(1.4) = \frac{4.482 - 3.669}{2(0.1)} = 4.065 \quad (4 \text{ angka bena})$$

(c) Untuk menghitung $f'(1.3)$ digunakan rumus hampiran selisih-maju, sebab $x = 1.3$ hanya mempunyai titik-titik sesudahnya (maju), tetapi tidak memiliki titik-titik sebelumnya. Sebaliknya, untuk menghitung nilai $f'(2.5)$ digunakan rumus hampiran selisih-mundur, sebab $x = 2.5$ hanya mempunyai titik-titik sebelumnya (mundur).

Hampiran selisih-maju :

$$\begin{aligned} f'_0 &= \frac{f_1 - f_0}{h} + O(h) \\ f'(1.3) &= \frac{4.482 - 3.669}{0.2} = 4.065 \end{aligned}$$

Hampiran selisih-mundur :

$$\begin{aligned} f'_0 &= \frac{f_0 - f_{-1}}{h} + O(h) \\ f'(2.5) &= \frac{12.182 - 9.974}{0.2} = 11.04 \quad \blacksquare \end{aligned}$$

7.9 Ekstrapolasi Richardson

Ekstrapolasi Richardson juga dapat diterapkan pada turunan numerik untuk memperoleh solusi yang lebih teliti. Misalkan $D(h)$ dan $D(2h)$ adalah hampiran $f'(x_0)$ dengan mengambil titik-titik masing-masing sejarak h dan $2h$. Misalkan untuk menghitung $f'(x_0)$ digunakan rumus hampiran beda-pusat orde $O(h^2)$:

$$\begin{array}{c} h \quad h \\ \hline x_{-1} & x_0 & x_1 \end{array}$$

$$\begin{aligned} D(h) &= 1/2h (f_1 - f_{-1}) + O(h^2) \\ &= f'_0 + Ch^2 + \dots \end{aligned} \quad (\text{P.7.18})$$

$$\begin{array}{c} 2h \quad 2h \\ \hline x_{-2} & x_{-1} & x_0 & x_1 & x_2 \end{array}$$

$$\begin{aligned} D(2h) &= \frac{1}{2(2h)} (f_2 - f_{-2}) + O((2h)^2) \\ &= f'_0 + C(2h)^2 + \dots \\ &= f'_0 + 4Ch^2 + \dots \end{aligned} \quad (\text{P.7.19})$$

Kurangi persamaan (P.7.18) dengan persamaan (P.7.19), menghasilkan :

$$D(h) - D(2h) = -3Ch^2$$

dari sini,

$$C = \frac{D(h) - D(2h)}{-3h^2} \quad (\text{P.7.20})$$

Sulihkan (P.7.20) ke dalam persamaan (P.7.18) :

$$\begin{aligned} D(h) &= f'_0 + \frac{[D(h) - D(2h)]h^2}{-3h^2} \\ &= f'_0 - 1/3 [D(h) - D(2h)] \end{aligned}$$

atau

$$f'_0 = D(h) + 1/3 [D(h) - D(2h)] \quad (\text{P.7.21})$$

Ekstrapolasi Richardson dapat diperluas penggunaannya untuk mendapatkan nilai turunan fungsi yang lebih baik (*improve*). Berdasarkan persamaan (P.7.21) di atas dapat dituliskan aturan:

$$f'_0 = D(h) + \frac{1}{2^n - 1} [D(h) - D(2h)] \quad (\text{P.7.22})$$

yang dalam hal ini n adalah orde galat rumus yang dipakai. Misalnya digunakan rumus hampiran selisih-pusat orde $O(h^2)$ dalam menghitung $D(h)$ dan $D(2h)$, maka $n = 2$, sehingga rumus ekstrapolasi Richardsonnya adalah seperti pada persamaan (P.7.21).

Catat juga bahwa setiap perluasan ekstrapolasi Richardson akan menaikkan orde galat dari $O(h^n)$ menjadi $O(h^{n+2})$ (lihat bahasan ekstrapolasi Richardson pada Bab Integrasi Numerik).

Contoh 7.2

Diberikan data dalam bentuk tabel sebagai berikut :

x	$f(x)$
2.0	0.42298
2.1	0.40051
2.2	0.37507
2.3	0.34718
2.4	0.31729
2.5	0.28587
2.6	0.25337
2.7	0.22008
2.8	0.18649
2.9	0.15290
3.0	0.11963

Tentukan $f'(2.5)$ dengan ekstrapolasi Ricahrdson bila $D(h)$ dan $D(2h)$ dihitung dengan rumus hampiran selisih-pusat orde $O(h^2)$ sampai 5 angka bena.

Penyelesaian:

$D(h) \rightarrow$ selang titik yang dipakai: [2.4 , 2.6] dan $h = 0.1$

$$x_{-1} = 2.4, x_0 = 2.5, x_1 = 2.6$$

$$D(h) = \frac{f_1 - f_{-1}}{2h} = \frac{(0.25337 - 0.31729)}{2(0.1)} = -0.31960$$

$D(2h) \rightarrow$ selang titik yang dipakai: [2.3 , 2.7] dan $h = 0.2$

$$x_{-2} = 2.3, x_0 = 2.5, x_2 = 2.7$$

$$D(2h) = \frac{f_2 - f_{-2}}{2h} = \frac{(0.22008 - 0.34718)}{2(0.2)} = -0.31775$$

$D(4h)$ → selang titik yang dipakai: [2.1, 2.9] dan $h = 0.4$

$$x_{-4} = 2.1, x_0 = 2.5, x_4 = 2.9$$

$$D(4h) = \frac{f_4 - f_{-4}}{2h} = \frac{(0.40051 - 0.15290)}{2(0.4)} = -0.30951$$

$D(h) = -0.31960$ dan $D(2h) = -0.31775$ keduanya dihitung dengan rumus orde $O(h^2)$, maka $n = 2$, sehingga

$$\begin{aligned} f'(2.5) &= f'_0 = D(h) + 1/(2^2 - 1) [D(h) - D(2h)] \\ &= -0.31960 + 1/3 (-0.31960 + 0.31775) \\ &= -0.32022 \rightarrow \text{mempunyai galat orde } O(h^4) \end{aligned}$$

$D(2h) = -0.31775$ dan $D(4h) = -0.30951$ keduanya dihitung dengan rumus orde $O(h^2)$, maka $n = 2$, sehingga

$$\begin{aligned} f'(2.5) &= f'_0 = D(2h) + 1/(2^2 - 1) [D(2h) - D(4h)] \\ &= -0.31775 + 1/3 (-0.31775 + 0.30951) \\ &= -0.32050 \rightarrow \text{mempunyai galat orde } O(h^4) \end{aligned}$$

$D(2h) = -0.32022$ dan $D(4h) = -0.32050$ mempunyai galat orde $O(h^4)$, maka $n = 4$, sehingga

$$\begin{aligned} f'(2.5) &= f'_0 = D(2h) + 1/(2^4 - 1) [D(2h) - D(4h)] \\ &= -0.32022 + 1/15 (-0.32022 + 0.32050) \\ &= -0.32020 \rightarrow \text{mempunyai galat orde } O(h^6) \end{aligned}$$

Tabel Richardson :

h	$O(h^2)$	$O(h^4)$	$O(h^6)$
0.1	-0.31960		
0.2	-0.31775	-0.32022	
0.4	-0.30951	-0.32050	-0.32020

Jadi, $f'(2.5) = -0.32020$. ■

7.10 Terapan Turunan Numerik dalam Bidang Pengolahan Citra

Citra (*image*) merupakan kumpulan elemen gambar (*picture element = pixel*) yang secara keseluruhan merekam suatu adegan (*scene*) melalui pengindera visual (kamera) [DUL96]. Citra intensitas ialah citra yang setiap *pixel* merekam intensitas cahaya yang dipantulkan dari setiap titik di objek, misalnya citra biner, *graylevel*, berwarna, dan banyak-alur (*multi-channel*). Untuk kebutuhan pengolahan dengan komputer, citra disajikan dalam bentuk diskrit yang disebut citra digital. Citra digital dapat disajikan oleh matriks f yang berukuran $M \times N$ dengan bentuk:

$$f = \begin{bmatrix} f_{11} & f_{12} & \dots & f_{1N} \\ f_{21} & f_{22} & \dots & f_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ f_{M1} & f_{M2} & \dots & f_{MN} \end{bmatrix} \quad (\text{P.7.22})$$

Tiap elemen matriks adalah bilangan bulat dalam rentang [0..255] untuk citra 8 bit.

Salah satu proses yang terdapat dalam pengolahan citra ialah pendekatan tepi. Tepi merupakan *feature* yang penting pada suatu citra. Tepi didefinisikan sebagai perubahan intensitas yang besar dalam jarak yang singkat. Perbedaan intensitas inilah yang menampakkan rincian pada gambar. Tepi biasanya terdapat pada batas antara dua daerah berbeda pada suatu citra. Tepi memberikan informasi batas-batas objek dengan lingkungannya atau dengan objek yang lain, *feature* untuk mengidentifikasi objek, dan untuk terapan penapisan citra.

Pendekatan tepi merupakan langkah pertama untuk melingkupi informasi di dalam citra. Tepi mencirikan batas-batas objek dan karena itu tepi berguna untuk proses segmentasi dan identifikasi objek di dalam citra. Tujuan operasi pendekatan tepi adalah untuk meningkatkan penampakan garis batas suatu daerah atau objek di dalam citra.

Salah satu pendekatan yang dipakai dalam pendekatan sisi adalah dengan kemiringan diferensial (*differential gradient*). Secara matematis perubahan intensitas yang besar dalam jarak yang sangat singkat dapat dipandang sebagai suatu fungsi yang memiliki kemiringan yang besar. Pengukuran kemiringan suatu fungsi dilakukan dengan menghitung turunan pertamanya. Dalam citra digital, pendekatan tepi dapat dilakukan dengan cara yang mirip, yaitu dengan turunan pertamanya secara parsial dalam ruang diskrit:

$$\nabla f(x, y) = \begin{bmatrix} \partial f / \partial x \\ \partial f / \partial y \end{bmatrix} = \begin{bmatrix} f_x \\ f_y \end{bmatrix} \quad (\text{P.7.23})$$

yang dalam hal ini kedua turunan parsial didefinisikan sebagai

$$D_1(x) = \frac{\partial f(x, y)}{\partial x} \approx \frac{f(x + \Delta x, y) - f(x, y)}{\Delta x} \quad (\text{P.7.24})$$

$$D_1(y) = \frac{\partial f(x, y)}{\partial y} \approx \frac{f(x, y + \Delta y) - f(x, y)}{\Delta y} \quad (\text{P.7.25})$$

Biasanya $\Delta x = \Delta y = 1$, sehingga persamaan turunan pertama menjadi:

$$D_1(x) = \frac{\partial f(x, y)}{\partial x} = f(x+1, y) - f(x, y)$$

$$D_1(y) = \frac{\partial f(x, y)}{\partial y} = f(x, y+1) - f(x, y)$$

Kekuatan tepi pada setiap *pixel* citra dihitung dengan rumus:

$$G[f(x, y)] = |f_x|^2 + |f_y|^2 \quad (\text{P.7.26})$$

atau dengan rumus

$$G[f(x, y)] = \max(|f_x|^2, |f_y|^2) \quad (\text{P.7.27})$$

Suatu *pixel* dianggap sebagai *pixel* sisi jika kekuatan tepinya di atas nilai ambang (*threshold*) tertentu.

$D_1(x)$ dan $D_1(y)$ merupakan hampiran selisih-maju. Hampiran lain yang dipakai adalah hampiran selisih-pusat, yaitu:

$$D_2(x) = \frac{\partial f(x, y)}{\partial x} \approx \frac{f(x + \Delta x, y) - f(x - \Delta x, y)}{2\Delta x} \quad (\text{P.7.28})$$

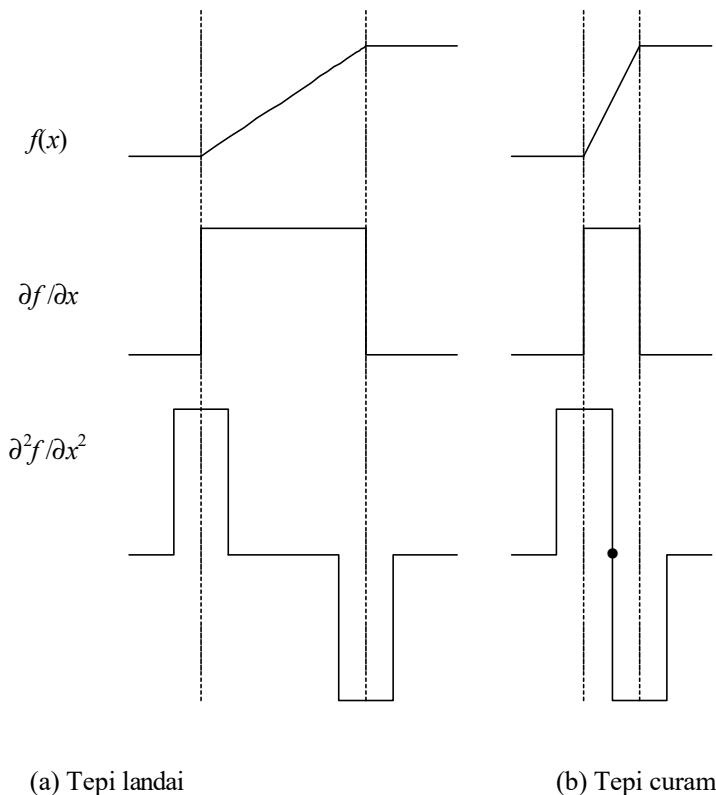
$$D_2(y) = \frac{\partial f(x, y)}{\partial y} \approx \frac{f(x, y + \Delta y) - f(x, y - \Delta y)}{2\Delta y} \quad (\text{P.7.29})$$

Gambar 7.2 adalah contoh hasil deteksi semua tepi citra Lena, citra *Camera*, dan citra botol.



Gambar 7.2 Deteksi semua tepi citra Lena, *camera*, dan botol

Operator lain yang digunakan untuk mendeteksi sisi adalah yang berdasarkan pada operasi turunan kedua (Gambar 7.3), yang dikenal dengan operator Laplace (*Laplacian*). Operator Laplace mendeteksi lokasi tepi lebih akurat khususnya pada tepi yang curam.



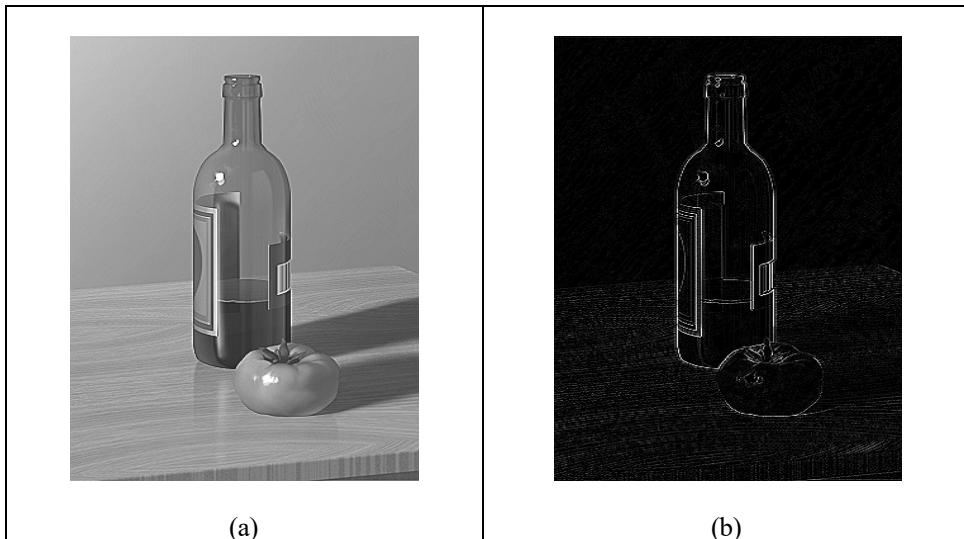
Gambar 7.3 Operator Laplace

Pada Gambar 7.3, kurva pada baris pertama menunjukkan perubahan intensitas suatu tepi. Baris kedua adalah turunan pertamanya, dan baris ketiga adalah turunan keduanya. Kolom kiri (a) adalah untuk sisi yang landai sedangkan kolom (b) untuk sisi yang curam. Dari Gambar 7.3 terlihat juga bahwa turunan kedua dari tepi yang landai tidak terdapat persilangan-nol (*zero crossing*), sedangkan pada tepi yang curam terdapat persilangan-nol yang ditandai dengan titik (●). Persilangan-nol ialah titik perubahan dari nilai positif ke negatif atau sebaliknya.

Jika digunakan hampiran selisih-maju, maka operator Laplace diturunkan sebagai berikut:

$$\begin{aligned}
\nabla^2 f &= \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \\
&= D_1(D_1(x)) + D_1(D_1(y)) \\
&= \frac{1}{\Delta x} D_1(f(x + \Delta x, y) - D_1(f(x, y)) + \frac{1}{\Delta y} D_1(f(x, y + \Delta y) - \\
&\quad D_1(f(x, y))) \\
&= \frac{1}{\Delta x} \left\{ \frac{f(x + \Delta x + \Delta x, y) - f(x + \Delta x, y)}{\Delta x} - \frac{f(x + \Delta x, y) - f(x, y)}{\Delta x} \right\} + \\
&\quad \frac{1}{\Delta y} \left\{ \frac{f(x, y + \Delta y + \Delta y) - f(x, y + \Delta y)}{\Delta y} - \frac{f(x, y + \Delta y) - f(x, y)}{\Delta y} \right\} \\
&= \frac{f(x + 2\Delta x, y) - 2f(x + \Delta x, y) + f(x, y)}{(\Delta x)^2} + \\
&\quad \frac{f(x, y + 2\Delta y) - 2f(x, y + \Delta y) + f(x, y)}{(\Delta y)^2} \tag{P.7.30}
\end{aligned}$$

Biasanya, $\Delta x = \Delta y = 1$ sehingga bentuk (P.7.30) menjadi lebih sederhana. Gambar 7.4 memperlihatkan hasil pendektsian tepi pada citra botol dengan operator Laplace.



Gambar 7.4 (a) citra botol; (b) hasil pendektsian tepi dengan operator Laplace

Tidak ada hal besar yang terjadi dengan tiba-tiba, bahkan yang lebih banyak
daripada setangkai anggur atau sebutir buah ara sekalipun.
Perlu waktu. Mula-mula ia berbunga, kemudian menjadi buah,
dan akhirnya matang.
(Epictetus)

Soal Latihan

1. Jika $x_0 - h, x_0, x_0 + h \in [a,b]$, perlihatkanlah

$$f_0'' = \frac{1}{h^2} (f_1 - 2f_0 + f_{-1}) + E$$

dan

$$E = -\frac{12}{h^2} h^2 f_0^{(iv)} + \dots$$

2. Jika f adalah polinom derajat tiga dan

$$f_0' = a_{-2} f_2 + a_{-1} f_{-1} + a_1 f_1 + a_2 f_2 + E$$

tentukan nilai-nilai tetapan a_{-2} , a_{-1} , a_1 , dan a_2 . Perlihatkan juga bahwa

$$E = \frac{1}{30} h^4 f_0^{(v)} + \dots$$

3. Diberikan tabel yang berisi titik-titik sebuah fungsi f :

x	$f(x)$
1.000	0.54030
1.100	0.45360
1.198	0.36422
1.199	0.36329
1.200	0.36236
1.201	0.36143
1.202	0.36049
1.300	0.26750
1.400	0.16997

- (a) Tentukan nilai $f'(1.2)$ dan $f''(1.2)$ untuk $h = 0.1$ dan $h = 0.001$ dengan rumus hampiran selisih-pusat orde $O(h^2)$.
- (b) Tabel di atas adalah tabel $f(x) = \cos(x)$. Bandingkan jawaban yang anda peroleh dengan nilai sejatinya.
4. Misalkan $f_{1/2}' = f'(x_0 + h/2)$ adalah hampiran nilai turunan dengan rumus selisih-pusat orde $O(h^2)$:

$$f_{1/2}' = \frac{f_1 - f_0}{h} + O(h^2)$$

- (a) Perlihatkan bahwa galat pemotongan rumus tersebut adalah
- $$- \frac{1}{24} h^2 f_0''' + \dots$$
- (b) Hitung $f'(1.5)$ jika diketahui hanya titik-titik berikut:
 $(1.2, 0.8333), (1.4, 0.7143), (1.6, 0.6250)$, dan $(1.8, 0.5556)$
- Gunakan empat angka bena.
5. Misalkan $D(2h)$ dan $D(4h)$ adalah hampiran $f'(x_0)$ dengan lebar selang $2h$ dan $4h$ menggunakan rumus hampiran selisih-pusat orde $O(h^4)$. Turunkan rumus ekstrapolasi Richardson untuk menghitung perkiraan $f'(x_0)$ yang lebih baik adalah:

$$f'(x_0) = D(2h) + \frac{[D(2h) - D(4h)]}{15}$$

Kemudian tentukan hampiran $f'(1.2)$ jika diketahui fungsinya $f(x) = e^x$ dalam selang $[0.8, 1.6]$ dengan $h = 0.1$.

Bab 8

Solusi Persamaan Diferensial Biasa

Penalaran adalah metode yang lambat dan berliku-liku dengan mana mereka yang tidak mengetahui kebenaran menemukannya. Hati mempunyai penalaran sendiri sedangkan penalaran itu tidak mengetahuinya.
(Blaise Pascal)

Persamaan diferensial adalah gabungan antara fungsi yang tidak diketahui secara eksplisit dan turunan (diferensial)-nya. Dalam kuliah Fisika anda tentu masih ingat persamaan gerak sistem pegas.

$$m \frac{d^2x}{dt^2} + c \frac{dx}{dt} + kx = 0 \quad (\text{P.8.1})$$

dengan m adalah massa pegas, k tetapan pegas, c koefisien redaman, dan x posisi sebuah titik pada pegas. Karena x adalah fungsi dari t , maka persamaan (P.8.1) ditulis juga sebagai

$$m x''(t) + c x'(t) + kx(t) = 0$$

atau dalam bentuk yang lebih ringkas,

$$mx'' + cx' + kx = 0.$$

Persamaan (P.8.1) mengandung fungsi $x(t)$ yang tidak diketahui rumus eksplisitnya, turunan pertamanya $x'(t)$, dan turunan kedua $x''(t)$.

Arti fisis diferensial adalah laju *perubahan* sebuah peubah terhadap peubah lain. Pada persamaan (P.8.1), $x'(t)$ menyatakan laju perubahan posisi pegas x terhadap waktu t .

8.1 Kelompok Persamaan Diferensial

Persamaan diferensial dapat dibagi menjadi dua kelompok besar, yaitu persamaan diferensial biasa dan persamaan diferensial parsial.

1. Persamaan diferensial biasa (PDB) - *Ordinary Differential Equations (ODE)*.

PDB adalah persamaan diferensial yang hanya mempunyai satu peubah bebas. Peubah bebas biasanya disimbolkan dengan x .

Contoh 8.1

Contoh-contoh persamaan berikut adalah persamaan diferensial biasa (PDB):

$$(i) \frac{dy}{dx} = x + y$$

$$(ii) y' = x^2 + y^2$$

$$(iii) 2 \frac{dy}{dx} + x^2 y - y = 0$$

$$(iv) y'' + y' \cos x - 3y = \sin 2x$$

$$(v) 2y''' - 23y' = 1 - y''$$

Peubah bebas untuk contoh (i) sampai (v) adalah x , sedangkan peubah terikatnya adalah y , yang merupakan fungsi dari x , atau ditulis sebagai $y = g(x)$. ■

Berdasarkan turunan tertinggi yang terdapat di dalam persamaannya, PDB dapat lagi dikelompokkan menurut ordenya, yaitu:

- a. PDB orde 1, yaitu PDB yang turunan tertingginya adalah turunan pertama.
Contoh (i), (ii), dan (iii) di atas adalah PDB orde 1.
- b. PDB orde 2, yaitu PDB yang turunan tertingginya adalah turunan kedua.
Contoh (iv) adalah PDB orde dua.
- c. PDB orde 3, yaitu PDB yang turunan tertingginya adalah turunan ketiga
Contoh (v) di atas adalah PDB orde tiga.
- d. dan seterusnya untuk PDB dengan orde yang lebih tinggi. PDB orde 2 ke atas dinamakan juga PDB orde lanjut.

2. Persamaan Diferensial Parsial (PDP) - *Partial Differential Equations (PDE)*.

PDP adalah persamaan diferensial yang mempunyai lebih dari satu peubah bebas. Turunan fungsi terhadap setiap peubah bebas dilakukan secara parsial.

Contoh 8.2

Contoh-contoh persamaan berikut adalah persamaan diferensial parsial (PDP):

$$(i) \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 6xye^{x+y} \quad (\text{yang dalam hal ini, } u = g(x,y))$$

$$(ii) \frac{\partial u}{\partial t} = 3\sin(x+t) + \frac{\partial^2 u}{\partial x^2} + (1+x^2) \frac{\partial^2 u}{\partial y^2} \quad (\text{yang dalam hal ini, } u = g(x,y,t))$$

■

Peubah bebas untuk contoh (i) adalah x dan y , sedangkan peubah terikatnya adalah u , yang merupakan fungsi dari x dan y , atau ditulis sebagai $u = g(x,y)$. Sedangkan peubah bebas untuk contoh (ii) adalah x , y , dan t , sedangkan peubah terikatnya adalah u , yang merupakan fungsi dari x , y , dan t , atau ditulis sebagai $u = g(x, y, t)$.

Buku ini hanya membahas metode-metode numerik untuk menyelesaikan PDB, khususnya PDB orde satu. Pada bagian akhir bab akan ditunjukkan bahwa PDB orde lanjut dapat dikembalikan bentuknya menjadi sistem PDB orde satu.

8.2 Terapan Persamaan Diferensial

Persamaan diferensial berperanan penting di alam, sebab kebanyakan fenomena alam dirumuskan dalam bentuk diferensial. Persamaan diferensial sering digunakan sebagai model matematika dalam bidang sains maupun dalam bidang rekayasa. Hukum-hukum dasar fisika, mekanika, listrik, dan termodinamika biasanya didasarkan pada perubahan sifat fisik dan keadaan sistem. Daripada menjelaskan keadaan sistem fisik secara langsung, hukum-hukum tersebut biasanya dinyatakan dalam perubahan *spasial* (koordinat) dan *temporal* (waktu) [CHA91]. Misalnya hukum Newton II menyatakan percepatan sebagai laju perubahan kecepatan setiap waktu, atau $a = dv/dt$, hukum termodinamika (Fluks panas = $-k \frac{\partial T}{\partial x}$, dengan k = konduktivitas panas, dan T = suhu), hukum Faraday (Beda tegangan = $L \frac{di}{dt}$, dengan L = induktansi, dan i = arus). Dengan mengintegralkan persamaan diferensial, dihasilkan fungsi matematika yang menjelaskan keadaan spasial dan temporal sebuah sistem, dinyatakan dalam percepatan, energi, massa, atau tegangan.

Persamaan (P.8.1) adalah terapan PDB dalam bidang fisika. Dalam bidang teknologi pangan, biologi, farmasi, dan teknik kimia, dikenal persamaan yang

menyatakan bahwa laju pertumbuhan bakteri pada waktu t sebanding dengan jumlah bakteri (p) pada saat itu,

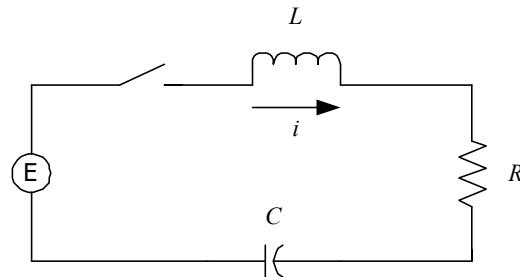
$$\frac{dp}{dt} = kp \quad (\text{P.8.2})$$

dengan k adalah *tetapan kesebandingan*. Dalam bidang kelistrikan (elektro), para rekayasaannya tentu mengetahui benar hukum Kirchoff untuk sebuah rangkaian listrik sederhana *RLC* seperti pada Gambar 8.1.

Hukum tegangan Kirchoff menyatakan bahwa jumlah aljabar dari perubahan tegangan di sekeliling rangkaian tertutup adalah nol,

$$L \frac{di}{dt} + Ri = \frac{q}{C} - E(t) = 0 \quad (\text{P.8.3})$$

dengan $L(di/dt)$ adalah perubahan tegangan antara induktor, L adalah induktansi kumparan (dalam *henry*), R adalah tahanan (dalam *ohm*), q adalah muatan pada kapasitor (dalam *coulomb*), C adalah kapasitansi (dalam *farad*), dan $E(t)$ adalah tegangan yang berubah terhadap waktu.



Gambar 8.1 Rangkaian listrik RLC

Beberapa persamaan diferensial dapat dicari solusinya secara analitis dengan teknik integral. Persamaan (P.8.2) misalnya, solusinya dapat ditemukan sebagai berikut:

$$\begin{aligned} dp/dt &= kp \\ \Leftrightarrow \quad dp/p &= k dt \\ \Leftrightarrow \quad \int dp/p &= \int k dt \\ \Leftrightarrow \quad \ln(p) + C_1 &= kt + C_2 \end{aligned}$$

$$\Leftrightarrow \ln(p) = kt + (C_2 - C_1) = kt + C \quad , \quad \text{dengan } C = C_2 - C_1$$

$$\Leftrightarrow p = e^{kt+C} = e^{kt} e^C = p_0 e^{kt} \quad , \quad \text{dengan } p_0 = e^C$$

Jadi, solusi analitiknya adalah

$$p(t) = p_0 e^{kt}$$

dengan p_0 adalah jumlah bakteri pada waktu $t = 0$. Bila $p_0 = p(0)$ diketahui, maka solusi yang unik dapat diperoleh. Dengan cara yang sama, solusi unik persamaan (P.8.3) juga dapat dihitung secara analitik bila diketahui besar arus pada $t = 0$ adalah $i(0) = 0$, yaitu

$$i(t) = (E/R)(1 - e^{-Rt/L})$$

Setelah persamaan $i(t)$ diperoleh, besar arus pada sembarang waktu t dapat dihitung.

Metode numerik untuk persamaan diferensial memainkan peranan sangat penting bagi rekayasawan, karena dalam prakteknya sebagian besar persamaan diferensial tidak dapat diselesaikan secara analitik. Metode numerik dipakai para rekayasawan untuk memperoleh solusi persamaan diferensial. Bila metode analitik memberikan solusi persamaan diferensial dalam bentuk fungsi menerus, maka metode numerik memberikan solusi persamaan diferensial dalam bentuk farik. Upabab berikut ini membahas berbagai metode numerik untuk menghitung solusi PDB orde satu.

8.3 PDB Orde Satu

Bentuk baku PDB orde satu dengan nilai awal ditulis sebagai

$$y' = f(x, y) \quad \text{dengan nilai awal } y(x_0) = y \quad (\text{P.8.4})$$

Catatan: Kadang-kadang y' ditulis sebagai dy/dx . Jadi, $y' = dy/dx$.

PDB orde satu yang tidak mengikuti bentuk baku tersebut harus ditulis ulang menjadi bentuk persamaan (P.8.4), agar ia dapat diselesaikan secara numerik.

Contoh 8.3

Contoh-contoh persamaan berikut adalah persamaan diferensial biasa dan transformasinya ke dalam bentuk baku PDB orde 1:

$$(i) 2y' + xy = 100; \quad y(0) = 1$$

Bentuk baku: $y' = (100 - xy)/2$; $y(0) = 1$

$$(ii) -xy' + 2y/x = y' - y \quad ; \quad y(1) = -1$$

Bentuk baku: $y' = \frac{2y/x + y}{1+x}$; $y(1) = -1$ ■

Penyelesaian PDB secara numerik berarti menghitung nilai fungsi di $x_{r+1} = x_r + h$, dengan h adalah *ukuran langkah (step)* setiap lelaran. Pada metode analitik, nilai awal berfungsi untuk memperoleh solusi yang unik, sedangkan pada metode numerik nilai awal (*initial value*) pada persamaan (P.8.4) berfungsi untuk memulai lelaran. Terdapat beberapa metode numerik yang sering digunakan untuk menghitung solusi PDB, mulai dari metode yang paling dasar sampai dengan metode yang lebih teliti, yaitu

1. Metode Euler
2. Metode Heun
3. Metode Deret Taylor
4. Metode Runge-Kutta
5. Metode *predictor-corrector*.

8.4 Metode Euler

Diberikan PDB orde satu,

$$y' = dy/dx = f(x, y) \text{ dan nilai awal } y(x_0) = y_0$$

Misalkan

$$y_r = y(x_r)$$

adalah hampiran nilai y di x_r yang dihitung dengan metode Euler. Dalam hal ini

$$x_r = x_0 + rh, \quad r = 0, 1, 2, \dots, n.$$

Metoda Euler diturunkan dengan cara menguraikan $y(x_{r+1})$ di sekitar x_r ke dalam deret Taylor:

$$y(x_{r+1}) = y(x_r) + \frac{(x_{r+1} - x_r)}{1!} y'(x_r) + \frac{(x_{r+1} - x_r)^2}{2!} y''(x_r) + \dots \quad (\text{P.8.5})$$

Bila persamaan (P.8.5) dipotong sampai suku orde tiga, diperoleh

$$y(x_{r+1}) \approx y(x_r) + \frac{(x_{r+1} - x_r)}{1!} y'(x_r) + \frac{(x_{r+1} - x_r)^2}{2!} y''(t) , x_r < t < x_{r+1} \quad (\text{P.8.6})$$

Berdasarkan persamaan (P.8.4),

$$y'(x_r) = f(x_r, y_r)$$

dan

$$x_{r+1} - x_r = h$$

maka persamaan (P.8.6) dapat ditulis menjadi

$$y(x_{r+1}) \approx y(x_r) + hf(x_r, y_r) + \frac{h^2}{2} y''(t) \quad (\text{P.8.7})$$

Dua suku pertama persamaan (P.8.7), yaitu

$$y(x_{r+1}) = y(x_r) + hf(x_r, y_r) ; \quad r = 0, 1, 2, \dots, n \quad (\text{P.8.8})$$

menyatakan **metode Euler** atau **metode Euler-Cauchy**. Metode Euler disebut juga **metode orde-pertama**, karena pada persamaan (P.8.7) kita hanya mengambil sampai suku orde pertama saja. Untuk menyederhanakan penulisan, persamaan (P.8.8) dapat juga ditulis lebih singkat sebagai

$$y_{r+1} = y_r + hf_r$$

Selain dengan bantuan deret Taylor, metode Euler juga dapat diturunkan dengan cara yang berbeda. Sebagai contoh, misalkan kita menggunakan aturan segiempat untuk mengintegrasikan $f(x, y)$ pada persamaan diferensial

$$y' = f(x, y) ; \quad y(x_0) = y_0$$

Integrasikan kedua ruas dalam selang $[x_r, x_{r+1}]$:

$$\int_{x_r}^{x_{r+1}} y'(x) dx = \int_{x_r}^{x_{r+1}} f(x, y(x)) dx$$

Gunakan aturan segiempat untuk mengintegrasikan ruas kanan, menghasilkan:

$$y(x_{r+1}) - y(x_r) = h f(x_r, y(x_r))$$

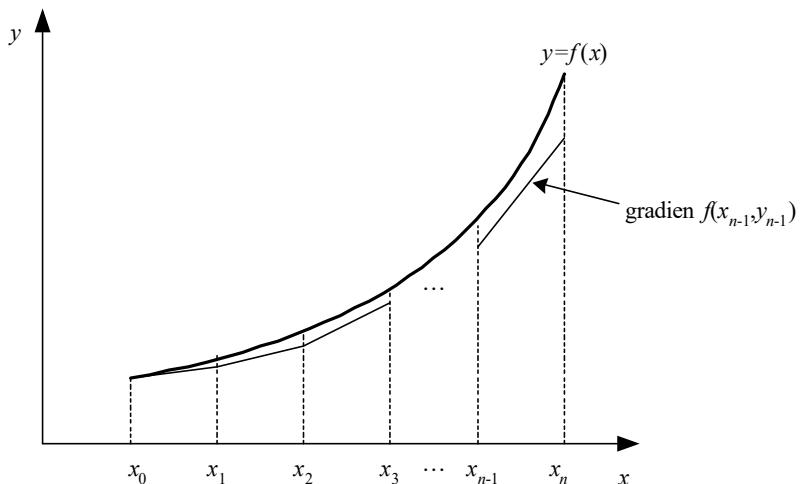
atau

$$y(x_{r+1}) = y(x_r) + h f(x_r, y_r)$$

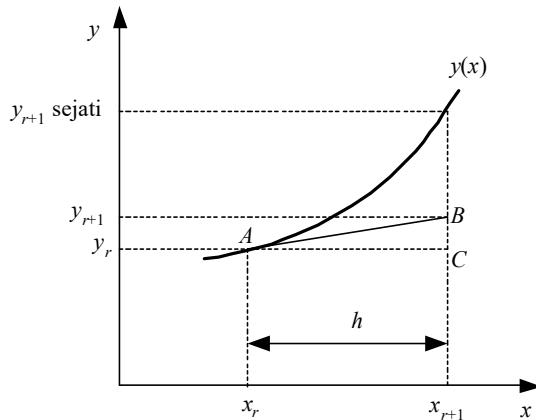
yang merupakan metode Euler.

8.4.1 Tafsiran Geometri Metode PDB

Pikirkanlah kembali bahwa $f(x,y)$ dalam persamaan diferensial menyatakan gradien garis singgung kurva di titik (x,y) . Kita mulai menarik garis singgung dari titik (x_0, y_0) dengan gradien $f(x_0, y_0)$ dan berhenti di titik (x_1, y_1) , dengan y_1 dihitung dari persamaan (P.8.8). Selanjutnya, dari titik (x_1, y_1) ditarik lagi garis dengan gradien $f(x_1, y_1)$ dan berhenti di titik (x_2, y_2) , dengan y_2 dihitung dari persamaan (P.8.8). Proses ini kita ulang beberapa kali, misalnya sampai lelaran ke-n, sehingga hasilnya adalah garis patah-patah seperti yang ditunjukkan pada Gambar 8.2.



Gambar 8.2 Tafsiran geometri metode PDB



Gambar 8.3 Tafsiran geometri untuk penurunan metode Euler

Berdasarkan tafsiran geometri pada Gambar 8.2, kita juga dapat menurunkan metode Euler. Tinjau Gambar 8.3. Gradien (m) garis singgung di x_r adalah

$$m = y'(x_r) = f(x_r, y_r) = \frac{\Delta y}{\Delta x} = \frac{BC}{AB} = \frac{y_{r+1} - y_r}{h}$$

$$\Leftrightarrow y_{r+1} = y_r + hf(x_r, y_r)$$

yang tidak lain adalah persamaan metode Euler.

8.4.2 Analisis Galat Metode Euler

Meskipun metode Euler sederhana, tetapi ia mengandung dua macam galat, *yaitu galat pemotongan (truncation error) dan galat longgokan (cumulative error)*. Galat pemotongan dapat langsung ditentukan dari persamaan (P.8.7), yaitu

$$E_p \approx \frac{1}{2} h^2 y''(t) = O(h^2) \quad (\text{P.8.9})$$

Galat pemotongan ini sebanding dengan kuadrat ukuran langkah h sehingga disebut juga **galat per langkah (error per step)** atau **galat lokal**. Semakin kecil nilai h (yang berarti semakin banyak langkah perhitungan), semakin kecil pula galat hasil perhitungannya. Perhatikan bahwa nilai pada setiap langkah (y_r) dipakai lagi pada langkah berikutnya (y_{r+1}). Galat solusi pada langkah ke- r adalah tumpukan galat dari langkah-langkah sebelumnya. Galat yang terkumpul pada akhir langkah ke- r ini disebut **galat longgokan (cumulative error)**. Jika langkah dimulai dari $x_0 = a$

dan berakhir di $x_n = b$ maka total galat yang terkumpul pada solusi akhir (y_n) adalah

$$E_{total} = \sum_{r=1}^n (1/2)h^2 y''(t) = n \frac{h^2}{2} y''(t) = \frac{(b-a)}{2h} h^2 y''(t) = \frac{(b-a)}{2} y''(t)h \quad (\text{P.8.10})$$

Galat longgokan total ini sebenarnya adalah

$$E_{total} = y(b)_{\text{sejati}} - y(x_n)_{\text{Euler}}$$

Persamaan (P.8.10) menyatakan bahwa galat longgokan sebanding dengan h . Ini berarti metode Euler memberikan hampiran solusi yang buruk, sehingga dalam praktek metode ini kurang disukai, namun metode ini membantu untuk memahami gagasan dasar metode penyelesaian PDB dengan orde yang lebih tinggi. Pengurangan h dapat meningkatkan ketelitian hasil, namun pengurangan h tanpa penggunaan bilangan berketelitian ganda tidaklah menguntungkan karena galat numerik meningkat disebabkan oleh galat pembulatan [NAK93].

Selain galat pemotongan, solusi PDB juga mengandung galat pembulatan, yang mempengaruhi ketelitian nilai y_1, y_2, \dots , semakin lama semakin buruk dengan meningkatnya n (baca kembali Bab 2 Deret Taylor dan Analisis Galat).

Program 8.1 Metode Euler

```

function y_Euler(x0, y0, b, h:real):real;
{menghitung nilai y(b) pada PDB
 y'=f(x,y);    y(x0)=y0
 dengan metode Euler
}
var
 r, n: integer;
 x, y: real;
begin
 n:=(b-x0)/h;      {jumlah langkah}
 y:=y0;            {nilai awal}
 x:=x0;
 for r:=1 to n do
 begin
 y:=y + h*f(x,y);   { hitung solusi y[xr] }
 x:=x + h;           { hitung titik berikutnya }
 end; {for}
 y_Euler:=y;          {y(b)}
end;

```

Contoh 8.4

Diketahui PDB

$$dy/dx = x + y \text{ dan } y(0) = 1$$

Gunakan metode Euler untuk menghitung $y(0,10)$ dengan ukuran langkah $h = 0.05$ dan $h = 0.02$. Jumlah angka benar = 5. Diketahui solusi sejati PDB tersebut adalah $y(x) = e^x - x - 1$.

Penyelesaian:

(i) Diketahui

$$\begin{aligned}a &= x_0 = 0 \\b &= 0.10 \\h &= 0.05\end{aligned}$$

Dalam hal ini, $f(x, y) = x + y$, dan penerapan metode Euler pada PDB tersebut menjadi

$$y_{r+1} = y_r + 0.02(x_r + y_r)$$

Langkah-langkah:

$$x_0 = 0 \rightarrow y_0 = 1$$

$$x_1 = 0.05 \rightarrow y_1 = y_0 + 0.05(x_0 + y_0) = 1 + (0.05)(0 + 1) = 1.0050$$

$$x_2 = 0.10 \rightarrow y_2 = y_1 + 0.05(x_1 + y_1) = 1.0050 + (0.05)(0.05 + 1.0050) = 1.05775$$

Jadi, $y(0.10) \approx 1.05775$.

(Bandingkan dengan nilai solusi sejatinya,

$$y(0.10) = e^{0.10} - 0.01 - 1 = 1.1103$$

sehingga galatnya adalah

$$\text{galat} = 1.1103 - 1.05775 = 0.05255 \quad)$$

(ii) Diketahui

$$\begin{aligned}a &= x_0 = 0 \\b &= 0.10 \\h &= 0.02\end{aligned}$$

Dalam hal ini, $f(x, y) = x + y$, dan penerapan metode Euler pada PDB tersebut menjadi

$$y_{r+1} = y_r + 0.02(x_r + y_r)$$

Langkah-langkah:

$$\begin{aligned}
x_0 &= 0 \rightarrow y_0 = 1 \\
x_1 &= 0.02 \rightarrow y_1 = y_0 + 0.02(x_0 + y_0) = 1 + (0.02)(0 + 1) = 1.0200 \\
x_2 &= 0.04 \rightarrow y_2 = y_1 + 0.02(x_1 + y_1) = 1.0200 + (0.02)(0.02 + 1.0200) = 1.0408 \\
x_3 &= 0.06 \rightarrow y_3 = 1.0624 \\
x_4 &= 0.08 \rightarrow y_4 = 1.0848 \\
x_5 &= 0.10 \rightarrow y_5 = 1.1081
\end{aligned}$$

Jadi, $y(0,10) \approx 1.1081$

(Bandingkan dengan solusi sejatinya, $y(0.10) = 1.1103$, sehingga galatnya adalah
 $\text{galat} = 1.1103 - 1.1081 = 1.1081$) ■

Contoh 8.4 memperlihatkan bahwa kita dapat mengurangi galat dengan memperbanyak langkah (memperkecil h).

8.5 Metode Heun (Perbaikan Metoda Euler)

Metode Euler mempunyai ketelitian yang rendah karena galatnya besar (sebanding dengan h). Buruknya galat ini dapat dikurangi dengan menggunakan metode Heun, yang merupakan perbaikan metode Euler (*modified Euler's method*). Pada metode Heun, solusi dari metode Euler dijadikan sebagai solusi perkiraan awal (*predictor*). Selanjutnya, solusi perkiraan awal ini diperbaiki dengan metode Heun (*corrector*).

Metode Heun diturunkan sebagai berikut: Pandang PDB orde satu

$$y'(x) = f(x, y(x))$$

Integrasikan kedua ruas persamaan dari x_r sampai x_{r+1} :

$$\begin{aligned}
\int_{x_r}^{x_{r+1}} f(x, y(x)) dx &= \int_{x_r}^{x_{r+1}} y'(x) dx \\
&= y(x_{r+1}) - y(x_r) \\
&= y_{r+1} - y_r
\end{aligned}$$

Nyatakan y_{r+1} di ruas kiri dan suku-suku lainnya di ruas kanan:

$$y_{r+1} = y_r + \int_{x_r}^{x_{r+1}} f(x, y(x)) dx \quad (\text{P.8.11})$$

Suku yang mengandung integral di ruas kanan, $\int_{x_r}^{x_{r+1}} f(x, y(x)) dx$, dapat diselesaikan dengan kaidah trapesium menjadi

$$\int_{x_r}^{x_{r+1}} f(x, y(x)) dx \approx \frac{h}{2} [f(x_r, y_r) + f(x_{r+1}, y_{r+1})] \quad (\text{P.8.12})$$

Sulihkan persamaan (P.8.12) ke dalam persamaan (P.8.11), menghasilkan persamaan

$$y_{r+1} = y_r + \frac{h}{2} [f(x_r, y_r) + f(x_{r+1}, y_{r+1})] \quad (\text{P.8.13})$$

yang merupakan **metode Heun**, atau **metode Euler-Cauchy yang diperbaiki**. Dalam persamaan (P.8.13), suku ruas kanan mengandung y_{r+1} . Nilai y_{r+1} ini adalah solusi perkiraan awal (*predictor*) yang dihitung dengan metode Euler. Karena itu, persamaan (P.8.13) dapat ditulis sebagai

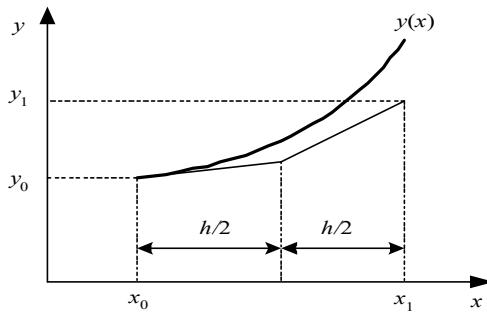
$$\begin{aligned} \text{Predictor : } y^{(0)}_{r+1} &= y_r + hf(x_r, y_r) \\ \text{Corrector : } y_{r+1} &= y_r + \frac{h}{2} [f(x_r, y_r) + f(x_{r+1}, y^{(0)}_{r+1})] \end{aligned} \quad (\text{P.8.14})$$

atau ditulis dalam satu kesatuan,

$$y_{r+1} = y_r + \frac{h}{2} [f(x_r, y_r) + f(x_{r+1}, y_r + hf(x_r, y_r))] \quad (\text{P.8.15})$$

8.5.1 Tafsiran Geometri Metode Heun

Metode ini mempunyai tafsiran geometri yang sederhana. Perhatikanlah bahwa dalam selang x_r sampai $x_r + \frac{1}{2} h$ kita menghampiri solusi y dengan garis singgung melalui titik (x_r, y_r) dengan gradien $f(x_r, y_r)$, dan kemudian meneruskan garis singgung dengan gradien $f(x_{r+1}, y^{(0)}_{r+1})$ sampai x mencapai x_{r+1} [KRE88] (lihat Gambar 8.4 dengan $r = 0$).



Gambar 8.4 Tafsiran geometri metode Heun

8.5.2 Galat Metode Heun

Dari persamaan (P.8.14), suku $\frac{h}{2} [f(x_r, y_r) + f(x_{r+1}, y^{(0)}_{r+1})]$ bersesuaian dengan aturan trapesium pada integrasi numerik. Dapat dibuktikan bahwa galat per langkah metode Heun sama dengan galat kaidah trapesium, yaitu

$$E_p \approx -\frac{h^3}{12} y''(t) \quad , \quad x_r < t < x_{r+1}$$

$$= O(h^3) \quad (\text{P.8.15})$$

Bukti:

Misalkan,

y_{r+1} adalah nilai y sejati di x_{r+1}
 y_{r+1} adalah hampiran nilai y di x_{r+1}

Uraikan y_{r+1} di sekitar x_r :

$$\begin{aligned} Y_{r+1} &= y(x_r) + \frac{(x_{r+1} - x_r)^2}{1!} y'(x_r) + \frac{(x_{r+1} - x_r)^2}{2!} y''(x_r) + \\ &\quad \frac{(x_{r+1} - x_r)^3}{3!} y'''(x_r) + \dots \\ &= y_r + hy'_r + \frac{h^2}{2} y''_r + \frac{h^3}{6} y'''_r + \dots \end{aligned}$$

Dengan menyatakan $y'_r = f(x_r, y_r) = f_r$, maka

$$Y_{r+1} = y_r + hf_r + \frac{h^2}{2} f'_r + \frac{h^3}{6} f''_r + \dots \quad (\text{P.8.16})$$

Dari persamaan (P.8.14),

$$y_{r+1} = y_r + \frac{h}{2} [f(x_r, y_r) + f(x_{r+1}, y^{(0)}_{r+1})]$$

uraikan $f(x_{r+1}, y^{(0)}_{r+1})$ dengan menggunakan deret Taylor di sekitar x_r :

$$f(x_{r+1}, y^{(0)}_{r+1}) = f(x_{r+1}, y_{r+1})$$

$$\begin{aligned}
&= f(x_r, y_r) + \frac{(x_{r+1} - x_r)^2}{2!} f'(x_r, y_r) + \frac{(x_{r+1} - x_r)^2}{2!} f''(x_r, y_r) \\
&\quad + \dots \\
&= f_r + h f'_r + \frac{h^2}{2} f''_r + \dots
\end{aligned}$$

sehingga persamaan (P.8.14) dapat ditulis menjadi:

$$\begin{aligned}
y_{r+1} &= y_r + \frac{h}{2} [f(x_r, y_r) + f(x_{r+1}, y^{(0)}_{r+1})] \\
&= y_r + \frac{h}{2} [f_r + f_r + h f'_r + \frac{h}{2} h^2 f''_r + \dots] \\
&= y_r + h f_r + \frac{h^2}{2} f'_r + \frac{h^3}{4} f''_r + \dots
\end{aligned} \tag{P.8.17}$$

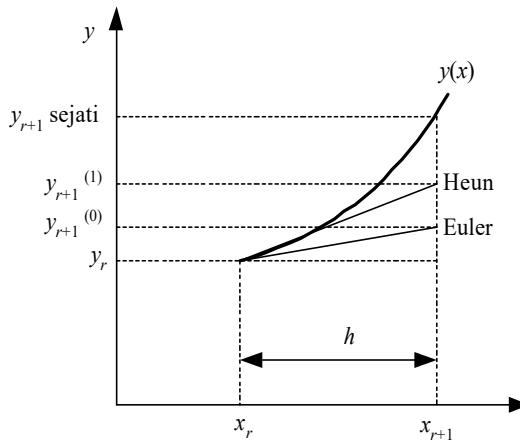
Galat per langkah = nilai sejati - nilai hampiran

$$\begin{aligned}
&= Y_{r+1} - y_{r+1} \\
&= (y_r + h f_r + \frac{h^2}{2} f'_r + \frac{h^3}{6} f'''_r + \dots) - (y_r + h f_r + \frac{h^2}{2} f'_r + \frac{h^3}{4} f''_r + \dots) \\
&= \frac{h^3}{6} f'''_r - \frac{h^3}{4} f'''_r + \dots \\
&= -\frac{h^3}{12} f'''_r + \dots \\
&= -\frac{h^3}{12} f'''(t), \quad x_r < t < x_{r+1} \\
&= O(h^3)
\end{aligned}$$

Galat longgokannya adalah,

$$\begin{aligned}
E_L &= \sum_{r=1}^n -\frac{1}{12} h^3 y'''(t) \\
&= -\frac{(b-a)}{12} h^2 y'''(t) \\
&= O(h^2)
\end{aligned} \tag{P.8.18}$$

Jadi, galat longgokan metode Heun sebanding dengan h^2 . Ini berarti solusi PDB dari metode Heun lebih baik daripada solusi dari metode Euler, namun jumlah komputasinya menjadi lebih banyak dibandingkan dengan metode Euler. Perbandingan metode Heun dengan metode Euler dilukiskan pada Gambar 8.5.



Gambar 8.5 Perbandingan metode Euler dengan metode Heun

Program 8.2 Metode Heun

```

function y_Heun(x0, y0, b, h:real):real;
{menghitung y(b) dengan metode Heun pada PDB
    y'=f(x,y);    y(x0)=y0
}
var
    r, n: integer;
    x, y, y_s : real;
begin
    n:=(b-x0)/h;           {jumlah langkah}
    y:=y0;                  {nilai awal}
    x:=x0;
    for r:=1 to n do
        begin
            y_s:=y;                      { y dari langkah r-1 }
            y:=y + h*f(x,y);             { y(xr) dengan Euler }
            y:=y_s + h/2 * ((f(x,y_s) + f(x+h,y)); { y(xr) dengan Heun }
            x:=x+1;                     { titik berikutnya}
        end;
    y_Heun:=y;
end;

```

Contoh 8.5

Diketahui PDB

$$\frac{dy}{dx} = x + y \quad ; \quad y(0) = 1$$

Hitung $y(0.10)$ dengan metode Heun ($h = 0.02$)

Penyelesaian:

Diketahui

$$\begin{aligned} f(x, y) &= x + y \\ a &= x_0 = 0 \\ b &= 0.10 \\ h &= 0.02 \end{aligned}$$

maka $n = (0.10 - 0)/0.02 = 5$ (jumlah langkah)

Langkah-langkah:

$$\begin{aligned} x_1 &= 0.02 \rightarrow y^{(0)}_1 = y_0 + hf(x_0, y_0) \\ &= 1 + 0.02(0 + 1) \\ &= 1.0200 \end{aligned}$$

$$\begin{aligned} y^{(1)}_1 &= y_0 + (h/2) [f(x_0, y_0) + f(x_1, y^{(0)}_1)] \\ &= 1 + (0.02/2)(0 + 1 + 0.02 + 1.0200) \\ &= 1.0204 \end{aligned}$$

$$\begin{aligned} x_2 &= 0.04 \rightarrow y^{(0)}_2 = y_1 + hf(x_1, y_1) \\ &= 1.0204 + 0.02(0.02 + 1.0204) \\ &= 1.0412 \end{aligned}$$

$$\begin{aligned} y^{(1)}_2 &= y_1 + (h/2) [f(x_1, y_1) + f(x_2, y^{(0)}_2)] \\ &= 1.0204 + (0.02/2)[0.02 + 1.0204 + 0.04 + 1.0412] \\ &= 1.0416 \end{aligned}$$

...

$$\begin{aligned} x_5 &= 0.10 \rightarrow y^{(0)}_5 = y_4 + hf(x_4, y_4) \\ y^{(1)}_5 &= y_4 + (h/2) [f(x_4, y_4) + f(x_5, y^{(0)}_5)] \\ &= 1.1104 \end{aligned}$$

Jadi, $y(0.10) \approx 1.1104$.

Bandingkan:

Nilai sejati	: $y(0.10) = 1.1103$
Euler (Contoh 8.4)	: $y(0.10) = 1.1081$
Heun (Contoh 8.5)	: $y(0.10) = 1.1104 \rightarrow$ lebih baik dari Euler

■

8.5.3 Perluasan Metode Heun

Metode Heun dapat diperluas dengan meneruskan lelarannya sebagai berikut:

$$\begin{aligned}
 y^{(0)}_{r+1} &= y_r + h f(x_r, y_r) \\
 y^{(1)}_{r+1} &= y_r + \frac{h}{2} [f(x_r, y_r) + f(x_{r+1}, y^{(0)}_{r+1})] \\
 y^{(2)}_{r+1} &= y_r + \frac{h}{2} [f(x_r, y_r) + f(x_{r+1}, y^{(1)}_{r+1})] \\
 y^{(3)}_{r+1} &= y_r + \frac{h}{2} [f(x_r, y_r) + f(x_{r+1}, y^{(2)}_{r+1})] \\
 &\dots \\
 y^{(k+1)}_{r+1} &= y_r + \frac{h}{2} [f(x_r, y_r) + f(x_{r+1}, y^{(k)}_{r+1})]
 \end{aligned}$$

Kondisi berhenti adalah bila

$$|y^{(k)}_{r+1} - y^{(k-1)}_{r+1}| < \epsilon$$

dengan ϵ adalah batas galat yang diinginkan. Jika lelarannya dilakukan satu kali (sampai dengan $y^{(1)}_{r+1}$ saja), maka lelarannya dinamakan **lelaran satu lemparan** (*one shot iteration*). Metode Heun adalah lelaran satu lemparan.

Program 8.3: Perluasan Metode Heun

```

function y_Heun2(x0, y0, b, h:real):real;
{menghitung y(b) dengan perluasan metode Heun pada PDB
  y'=f(x,y);    y(x0)=y0
}
const
  epsilon = 0.0000001;
var
  r, n: integer;
  x, y, y_s, tampung : real;
begin
  n:=(b-x0)/h;           {jumlah langkah}
  y:=y0;                  {nilai awal}
  x:=x0;
  for r:=1 to n do
    begin
      y_s:=y;             { y dari langkah r-1 }
      y:=y + h*f(x,y);    { y(xr) dengan Euler }
      repeat
        tampung:=y;
        y:=y_s + h/2 * ((f(x, y_s)+ f(x+h, y)));   {y(xr) dengan Heun}
      until ABS(y-tampung) < epsilon;
      x:=x+h;              { hitung titik berikutnya }
    end;
  y_Heun2:=y;
end;

```

8.6 Metode Deret Taylor

Kita sudah melihat bahwa metode Euler diturunkan dengan menggunakan deret Taylor. Deret Taylor pada penurunan metode Euler dipotong sampai suku orde pertama sehingga solusinya kurang teliti. Kita dapat meningkatkan ketelitian dengan memotong deret sampaisuku yang lebih tinggi lagi. Metode deret Taylor adalah metode yang umum untuk menurunkan rumus-rumus solusi PDB. Metode Euler merupakan metode deret Taylor yang paling sederhana.

Diberikan PDB

$$y'(x) = f(x, y) \text{ dengan kondisi awal } y(x_0) = y_0$$

Misalkan

$$y_{r+1} = y(x_{r+1}), \quad r = 0, 1, \dots, n$$

adalah hampiran nilai y di x_{r+1} . Hampiran ini diperoleh dengan menguraikan y_{r+1} di sekitar x_r sebagai berikut:

$$\begin{aligned} y(x_{r+1}) &= y(x_r) + \frac{(x_{r+1} - x_r)}{1!} y'(x_r) + \frac{(x_{r+1} - x_r)^2}{2!} y''(x_r) + \frac{(x_{r+1} - x_r)^3}{3!} \\ &\quad y'''(x_r) + \dots + \frac{(x_{r+1} - x_r)^n}{n!} y^{(n)}(x_r) \end{aligned}$$

atau

$$y(x_{r+1}) = y(x_r) + hy'(x_r) + \frac{h^2}{2} y''(x_r) + \frac{h^3}{6} y'''(x_r) + \dots + \frac{h^{(n)} y^{(n)}}{n!} x_r \quad (\text{P.8.19})$$

Persamaan (P.8.19) menyiratkan bahwa untuk menghitung hampiran nilai y_{r+1} , kita perlu menghitung $y'(x_r)$, $y''(x_r)$, ..., $y^{(n)}(x_r)$, yang dapat dikerjakan dengan rumus

$$y^{(k)}(x) = P^{(k-1)} f(x, y) \quad (\text{P.8.20})$$

yang dalam hal ini, P adalah operator turunan,

$$P = \left(\frac{\partial}{\partial x} + f \frac{\partial}{\partial y} \right) \quad (\text{P.8.21})$$

Contoh 8.3

Diketahui PDB

$$dy/dx = \frac{1}{2}x - \frac{1}{2}y ; y(0) = 1$$

Tentukan $y(0.50)$ dengan metode deret Taylor ($h = 0.25$).

Penyelesaian:

$$x_0 = 0 \rightarrow y_0 = 1$$

$$x_1 = 0.25 \rightarrow y_1 = ?$$

$$y(x_1) = y(x_0) + hy'(x_0) + \frac{h^2}{2} y''(x_0) + \frac{h^3}{6} y'''(x_0) + \dots + \frac{h^{(n)}}{n!} y^{(n)}(x_0) + \dots$$

Misal kita hanya menghitung $y(x_1)$ sampai suku orde ke-4 saja.

$$y'(x) = \frac{1}{2}x - \frac{1}{2}y$$

$$y''(x) = \frac{d}{dx} (\frac{1}{2}x - \frac{1}{2}y)$$

$$= \frac{1}{2} + f \cdot (-1/2)$$

$$= \frac{1}{2} - (\frac{1}{2}x - \frac{1}{2}y) \cdot \frac{1}{2}$$

$$= \frac{1}{2} - \frac{1}{4}x + \frac{1}{4}y$$

$$y'''(x) = \frac{d}{dx} (\frac{1}{2} - \frac{1}{4}x + \frac{1}{4}y)$$

$$= -1/4 + f \cdot 1/4$$

$$= -1/4 + (\frac{1}{2}x - \frac{1}{2}y) \cdot \frac{1}{4}$$

$$= -1/4 + x/8 - y/8$$

$$y^{(4)}(x) = \frac{d}{dx} (1/4 + 1/8x - 1/8y)$$

$$= 1/8 + f \cdot (-1/8)$$

$$= 1/8 - (x/2 - y/2) \cdot 1/8$$

$$= 1/8 - x/16 + y/16$$

Diperoleh:

$$y(x_0) = y(0) = 1$$

$$y'(x_0) = y'(0) = \frac{1}{2} \times 0 - \frac{1}{2} \times 1 = -1/2$$

$$y''(x_0) = y''(0) = \frac{1}{2} - \frac{1}{4} \times 0 + \frac{1}{4} \times 1 = 3/4$$

$$y'''(x_0) = y'''(0) = -1/4 + 1/8 \times 0 - 1/8 \times 1 = -3/8$$

$$y^{(4)}(x_0) = y^{(4)}(0) = 1/8 - 1/16 \times 0 + 1/16 \times 1 = 3/16$$

sehingga

$$y(x_1) = 1 + 0.25(-1/2) + ((0.25)^2/2)(3/4) + ((0.25)^3/6)(-3/8) + ((0.25)^4/24)(3/16) \\ = 0.8974915$$

$$x_2 = 0.50 \rightarrow y_2 = ?$$

$$y(x_2) = y(x_1) + hy'(x_1) + \frac{h^2}{2} y''(x_1) + \frac{h^3}{6} y'''(x_1) + \dots + \frac{h^{(n)}}{n!} y^{(n)}(x_1) = \dots$$

Diperoleh:

$$\begin{aligned} y(x_1) &= 0.8974915 \\ y'(x_1) &= (1/2)(0.25) - (1/2)(0.8974915) = -0.3237458 \\ y''(x_1) &= \frac{1}{2} - (\frac{1}{4})(0.25) + (\frac{1}{4})(0.8974915) = 0.6618729 \\ y'''(x_1) &= -\frac{1}{4} + (\frac{1}{8})(0.25) - (\frac{1}{8})(0.8974915) = -0.3309634 \\ y^{(4)}(x_1) &= \frac{1}{8} - (\frac{1}{16})(0.25) + (\frac{1}{16})(0.8974915) = 0.1654682 \end{aligned}$$

Sehingga,

$$\begin{aligned} y_2 &= 0.8974915 + 0.25(-0.3237458) + (0.25^2/2)(0.6618729) \\ &\quad + (0.25^3/6)(-0.3309634) + (0.25^4/24)(0.1654682) \\ &= 0.8364037 \end{aligned}$$

Jadi, $y(0.50) \approx 0.8364037$

(Bandingkan dengan solusi sejati, $y(0.50) = 0.8364023$) ■

Galat Metode Deret Taylor

Galat perlengkahan metode deret Taylor setelah pemotongan ke- n adalah

$$\begin{aligned} E_t &\approx \frac{h^{(n+1)} f^{(n+1)}}{(n+1)!}(t), \quad x_0 < t < x_{r+1} \\ &= O(h^{n+1}) \end{aligned} \tag{P.8.22}$$

Pada Contoh 8.1, galat per langkahnya adalah

$$E_p \approx \frac{h^5}{5!} f^{(5)}(t), \quad 0 < t < 0.50$$

Karena t tidak diketahui, kita hanya dapat menghitung batas atas dan batas bawah galat E_p dalam selang-buka $(0, 50)$.

Galat longgokan total metode deret Taylor adalah:

$$\begin{aligned}
E_L &= \frac{h^{(n+1)}}{(n+1)!} f^{(n+1)}(t) \\
&= \frac{b-a}{h} \frac{h^{(n+1)}}{(n+1)!} f^{(n+1)}(t) \\
&= (b-a) \frac{f^{(n+1)}(t)}{(n+1)!} h^n \\
&= O(h^n)
\end{aligned} \tag{P.8.23}$$

8.7 Orde Metode PDB

Orde metode penyelesaian PDB menyatakan ukuran ketelitian solusinya. Makin tinggi orde metode, makin teliti solusinya. Orde metode PDB dapat ditentukan dari persamaan galat per langkah atau dari galat longgokannya.

1. Jika galat longgokan suatu metode PDB berbentuk Ch^p , C tetapan, maka metode tersebut dikatakan berorde p .

Sebagai contoh,

$$\begin{aligned}
\text{metode Euler} \rightarrow \text{Galat longgokan} &= \frac{(b-a)}{2} y''(t) h = Ch, \quad C = \frac{(b-a)}{2} y''(t) \\
&= O(h) \rightarrow \text{orde metode Euler} = 1
\end{aligned}$$

$$\begin{aligned}
\text{metode Heun} \rightarrow \text{Galat longgokan} &= \frac{-(b-a)}{12} y''(t) h^2 = Ch^2, \quad C = \frac{-(b-a)}{12} y''(t) \\
&= O(h^2) \rightarrow \text{orde metode Heun} = 2
\end{aligned}$$

2. Jika galat per langkah suatu metoda PDB berbentuk Bh^{p+1} , B konstanta, maka metode tersebut dikatakan berorder p . Dengan kata lain, jika galat per langkah = $O(h^{p+1})$ maka galat longgokan = $O(h^p)$.

Sebagai contoh,

$$\begin{aligned}
\text{metode Euler} \rightarrow \text{Galat per langkah} &= \frac{1}{2} y''(t) h^2 = Bh^2, \quad B = \frac{1}{2} y''(t) \\
&= O(h^2) \rightarrow \text{orde metode Euler} = 2-1=1
\end{aligned}$$

metode Heun \rightarrow Galat per langkah = $\frac{-1}{12}y''(t)h^3 = Bh^3$, dengan $B = \frac{-1}{12}y''(t)$
 $= O(h^3) \rightarrow$ orde metode Heun = 3-1 = 2

Menentukan Galat per Langkah Metode PDB

Galat per langkah metode PDB diperoleh dengan bantuan deret Taylor. Kita sudah pernah menurunkan galat per langkah metode Heun dengan bantuan deret Taylor. Sekarang, prosedur untuk menentukan galat per langkah suatu metode PDB dapat ditulis sebagai berikut:

- (1) Notasi nilai y hampiran di x_{r+1} adalah y_{r+1}
- (2) Notasi nilai y sejati di x_{r+1} adalah Y_{r+1}
- (3) Uraikan y_{r+1} di sekitar x_r
- (4) Uraikan Y_{r+1} di sekitar x_r
- (5) Galat per langkah adalah = (4) - (3)

Contoh 8.4

Hitung galat per langkah metode PDB

$$y_{r+1} = y_r + hf_r \quad (\text{metode Euler})$$

dan tentukan orde metodenya.

Penyelesaian:

$$\begin{aligned} \text{Hampiran : } & y_{r+1} = y_r + hf_r \\ \text{Sejati : } & Y_{r+1} \end{aligned}$$

Uraikan y_{r+1} hampiran di sekitar x_r :

Ruas kanan persamaan y_{r+1} sudah terdefinisi dalam x_r , jadi y_{r+1} tidak perlu diuraikan lagi.

Uraikan Y_{r+1} di sekitar x_r :

$$\begin{aligned} Y_{r+1} &= Y(x_{r+1}) = y(x_r) + \frac{(x_{r+1} - x_r)}{1!} y'(x_r) + \frac{(x_{r+1} - x_r)^2}{2!} y''(x_r) + \dots \\ &= y_r + hy_r' + \frac{h^2}{2} y_r'' + \dots = y_r + hf_r + \frac{h^2}{2} f_r' + \dots \end{aligned}$$

$$\begin{aligned}
\text{Galat per langkah } E_p &= Y_{r+1} - y_{r+1} \\
&= \frac{h^2}{2} f'_r + \dots \\
&= \frac{h^2}{2} f'(t), \quad x_r < t < x_{r+1} \\
&= O(h^2)
\end{aligned}$$

Orde metode = 2 - 1 = 1 ■

Contoh 8.5

Hitung galat per langkah metode PDB

$$y_{r+1} = y_r + \frac{h}{12} (23f_r - 16f_{r-1} + 5f_{r-2})$$

dan tentukan orde metodenya.

Penyelesaian:

$$\begin{aligned}
\text{Hampiran : } y_{r+1} &= y_r + \frac{h}{12} (23f_r - 16f_{r-1} + 5f_{r-2}) \\
\text{Sejati : } Y_{r+1} &
\end{aligned}$$

Uraikan y_{r+1} di sekitar x_r :

$$\begin{array}{ll}
\begin{array}{l}
23f_r \\
-16f_{r-1} \\
5f_{r-2}
\end{array} & = 23f_r \\
& = -16(f_r - hf'_r + \frac{1}{2}h^2f''_r - \frac{h^3}{6}f'''_r + \dots) \\
& = 5(f_r - 2hf'_r + 4h^2/2f''_r - 8h^3/6f'''_r + \dots) +
\end{array}$$

$$23f_r - 16f_{r-1} + 5f_{r-2} = 12f_r + 6hf'_r + 2h^2f''_r - \frac{24}{6}h^3f'''_r + \dots$$

$$\begin{array}{ll}
y_{r+1} & = y_r + \frac{h}{12} (23f_r - 16f_{r-1} + 5f_{r-2}) \\
& = y_r + \frac{h}{12} (12f_r + 6hf'_r + 2h^2f''_r - \frac{24}{6}h^3f'''_r + \dots) \\
& = y_r + hf'_r + \frac{1}{2}h^2f''_r + \frac{1}{6}h^3f'''_r - \frac{1}{3}h^4f''''_r + \dots
\end{array}$$

Uraikan Y_{r+1} di sekitar x_r :

$$\begin{aligned}
Y_{r+1} &= y_r + hy'_r + h^2/2y''_r + h^3/6y'''_r + h^4/24y^{(4)}_r + \dots \\
&= y_r + hf'_r + \frac{1}{2}h^2f''_r + \frac{1}{6}h^3f'''_r + \frac{1}{24}h^4f''''_r + \dots
\end{aligned}$$

$$\begin{aligned}
\text{Galat per langkah } E_p &= Y_{r+1} - y_{r+1} = \frac{1}{24}h^4f''''_r + \frac{1}{3}h^4f''''_r + \dots \\
&= \frac{9}{24}h^4f''''_r + \dots \\
&= \frac{9}{24}h^4f''''(t), \quad x_{r-2} < t < x_{r+1}
\end{aligned}$$

Orde metode = 4 - 1 = 3 ■

8.8 Metode Runge-Kutta

Penyelesaian PDB dengan metode deret Taylor tidak praktis karena metode tersebut membutuhkan perhitungan turunan $f(x, y)$. Lagipula, tidak semua fungsi mudah dihitung turunannya, terutama bagi fungsi yang bentuknya rumit. Semakin tinggi orde metode deret Taylor, semakin tinggi turunan fungsi yang harus dihitung. Karena pertimbangan ini, metode deret Taylor yang berorde tinggi pun tidak dapat diterima dalam masalah praktek.

Metode Runge-Kutta adalah alternatif lain dari metode deret Taylor yang tidak membutuhkan perhitungan turunan. Metode ini berusaha mendapatkan derajat ketelitian yang lebih tinggi, dan sekaligus menghindarkan keperluan mencari turunan yang lebih tinggi dengan jalan mengevaluasi fungsi $f(x, y)$ pada titik terpilih dalam setiap selang langkah [CON80]. Metode Runge-Kutta adalah metode PDB yang paling populer karena banyak dipakai dalam praktek.

Bentuk umum metoda Range-Kutta orde- n ialah:

$$y_{r+1} = y_r + a_1 k_1 + a_2 k_2 + \dots + a_n k_n \quad (\text{P.8.24})$$

dengan a_1, a_2, \dots, a_n adalah tetapan, dan

$$k_1 = hf(x_r, y_r)$$

$$k_2 = hf(x_r + p_1 h, y_r + q_{11} k_1)$$

$$k_3 = hf(x_r + p_2 h, y_r + q_{21} k_1 + q_{22} k_2)$$

...

$$k_n = hf(x_r + p_{n-1} h, y_r + q_{n-1,1} k_1 + q_{n-1,2} k_2 + \dots + q_{n-1,n-1} k_{n-1})$$

Nilai a_i, p_i, q_{ij} dipilih sedemikian rupa sehingga meminimumkan galat per langkah, dan persamaan (P.8.24) akan sama dengan metode deret Taylor dari orde setinggi mungkin..

Galat per langkah metode Runge-Kutta orde- n : $O(h^{n+1})$

Galat longgokan metode Runge-Kutta orde- n : $O(h^n)$

Orde metode = n

8.8.1 Metode Runge-Kutta Orde Satu

Metode Runge-Kutta orde satu berbentuk

$$\begin{aligned} k_1 &= hf(x_r, y_r) \\ y_{r+1} &= y_r + (a_1 k_1) \end{aligned} \quad (\text{P.8.25})$$

Galat per langkah metode R-K orde satu adalah $O(h^2)$.
Galat longgokan metode R-K orde satu adalah $O(h)$.

Yang termasuk ke dalam metode Runge-Kutta orde satu ialah metode Euler:

$$\begin{aligned} k_1 &= hf(x_r, y_r) \\ y_{r+1} &= y_r + k_1 \end{aligned} \quad (\text{dalam hal ini } a_1 = 1)$$

8.8.2 Metode Runge-Kutta Orde Dua

Metode Runge-Kutta orde dua berbentuk

$$\begin{aligned} k_1 &= hf(x_r, y_r) \\ k_2 &= hf(x_r + p_1 h, y_r + q_{11} k_1) \\ y_{r+1} &= y_r + (a_1 k_1 + a_2 k_2) \end{aligned} \quad (\text{P.8.26})$$

Galat per langkah metode Runge-Kutta orde dua adalah $O(h^3)$.
Galat longgokan metode Runge-Kutta orde dua adalah $O(h^2)$.

Nilai a_1 , a_2 , p_1 , dan q_{11} ditentukan sebagai berikut:

Misalkan

$$\begin{aligned} f_r &= f(x_r, y_r) \\ f_x &= \frac{\partial f(x_r, y_r)}{\partial x}, \text{ dan } f_y = \frac{\partial f(x_r, y_r)}{\partial y} \end{aligned}$$

Uraikan k_2 ke dalam deret Taylor di sekitar (x_r, y_r) sampai suku orde satu saja:

$$\begin{aligned} k_2 &= hf(x_r + p_1 h, y_r + q_{11} k_1) \\ &= h(f + p_1 h f_x + q_{11} k_1 f_y) \\ &= h(f + p_1 h f_x + q_{11} h f f_y) \\ &= h(f + h(p_1 f_x + q_{11} f f_y)) \end{aligned}$$

Sedangkan k_1 tidak perlu diuraikan karena sudah berada dalam bentuk (x_r, y_r) .

Jadi,

$$\begin{aligned}
 y_{r+1} &= y_r + a_1 k_1 + a_2 k_2 \\
 &= y_r + a_1 h f_r + a_2 h f + a_2 h^2 (p_1 f_x + q_{11} f_y) \\
 &= y_r + (a_1 + a_2) h f + a_2 h^2 (p_1 f_x + q_{11} f_y)
 \end{aligned} \tag{P.8.27}$$

Uraikan Y_{+1} sejati di sekitar x_r sampai suku orde dua saja:

$$Y_{r+1} = y_r + h y'_r + \frac{1}{2} h^2 y''_r \tag{P.8.28}$$

Mengingat

$$y'_r = f(x_r, y_r) = f_r$$

dan

$$\begin{aligned}
 y''_r &= f'(x_r, y_r) = \\
 &= \frac{df(x_r, y_r)}{dx} \\
 &= \frac{\partial f}{\partial x} \frac{dx}{dx} + \frac{\partial f}{\partial y} \frac{dy}{dx} \\
 &= f_x + f_y f_r \\
 &= f_x + f f_y
 \end{aligned}$$

maka persamaan (P.8.28) menjadi

$$y_{r+1} = y_r + h f + \frac{1}{2} h^2 (f_x + f_r f_y) \tag{P.8.29}$$

Galat per langkah metode adalah

$$\begin{aligned}
 E_p &= (\text{P.8.29}) - (\text{P.8.27}): \\
 &= \{ y_r + h f + \frac{1}{2} h^2 (f_x + f_r f_y) \} - \{ y_r + (a_1 + a_2) h f_r + a_2 h^2 (p_1 f_x + q_{11} f_r f_y) \} \\
 &= \{ h f + \frac{1}{2} h^2 (f_x + f f_y) \} - \{ (a_1 + a_2) h f_r + a_2 h^2 (p_1 f_x + q_{11} f_y) \}
 \end{aligned}$$

Dengan membuat galat per langkah $E_p = 0$,

$$0 = \{ h f_r + \frac{1}{2} h^2 (f_x + f_r f_y) \} - \{ (a_1 + a_2) h f_r + a_2 h^2 (p_1 f_x + q_{11} f_r f_y) \}$$

atau

$$hf_r + \frac{1}{2} h^2(f_x + f_r f_y) = (a_1 + a_2) hf_r + a_2 h^2(p_1 f_x + q_{11} f_r f_y) \quad (\text{P.8.30})$$

Agar ruas kiri dan ruas kanannya sama, haruslah

$$\begin{aligned} a_1 + a_2 &= 1 \\ a_2 p_1 &= 1/2 \\ a_2 q_{11} &= 1/2 \end{aligned}$$

Karena sistem persamaan di atas terdiri atas tiga persamaan dengan empat peubah yang tidak diketahui, maka solusinya tidak unik, dengan kata lain, solusinya banyak. Solusi yang unik hanya dapat diperoleh dengan memberikan sebuah peubah dengan sebuah harga. Misalkan ditentukan nilai $a_2 = t$, $t \in R$, maka

$$\begin{aligned} a_1 &= 1 - a_2 = 1 - t \\ p_1 &= \frac{1}{2a_2} = \frac{1}{2t} \\ q_{11} &= \frac{1}{2a_2} = \frac{1}{2t} \end{aligned}$$

Karena kita dapat memberikan sembarang nilai t , berarti metode Runge-Kutta Orde dua tidak terhingga banyaknya.

Contoh metode Runge-Kutta orde dua adalah metode Heun, yang dalam hal ini

$$\begin{aligned} a_2 &= 1/2, \\ a_1 &= 1/2, \\ p_1 &= q_{11} = 1 \end{aligned}$$

Dalam bentuk Runge-Kutta orde 2, metode Heun dapat ditulis sebagai

$$\begin{aligned} k_1 &= hf(x_r, y_r) \\ k_2 &= hf(x_r + h, y_r + k_1) \\ y_{r+1} &= y_r + \frac{1}{2}(k_1 + k_2) \end{aligned}$$

Program Heun sudah pernah kita tulis (Program 8.2). Sekarang program tersebut kita tulis lagi dalam bentuk Runge-Kutta orde 2 menjadi Program 8.4 berikut ini.

Program 8.4 Metode Heun

```
function y_Heun(x0, y0, b, h:real):real;
{menghitung y(b) dengan metode Heun pada PDB
    y'=f(x,y);    y(x0)=y0
}
var
    r, n: integer;
    x, y, y_s, x_s : real;
begin
    n:=(b - x0)/h; {jumlah langkah}
    y:=y0;           {nilai awal}
    x:=x0;
    for r:=1 to n do
    begin
        k1:=h*f(x,y);
        k2:=h*f(x+h, y+k1);
        y:=y + (k1 + k2)/2;
        x:=x+1;
    end;
    y_Heun:=y;
end;
```

Contoh metode Runge-Kutta orde dua lainnya ialah **metode Ralston**, yang dalam hal ini

$$\begin{aligned}a_2 &= 2/3 \\a_1 &= 1/3, \\p_1 &= q_{11} = 3/4\end{aligned}$$

sehingga metode Ralston dapat ditulis dalam bentuk Runge-Kutta orde dua sebagai

$$\begin{aligned}k_1 &= hf(x_r, y_r) \\k_2 &= hf(x_r + \frac{3}{4}h, y_r + \frac{3}{4}k_1) \\y_{r+1} &= y_r + (\frac{1}{3}k_1 + \frac{2}{3}k_2)\end{aligned}\tag{P.8.31}$$

Sepintas, metode Runge-Kutta tampaknya rumit, tapi sebenarnya metode Runge-Kutta mudah diprogram. Dengan perhitungan tangan, seringnya menghitung $f(x, y)$ merupakan pekerjaan yang melelahkan. Tetapi dengan komputer, hal ini tidak menjadi masalah.

8.8.3 Metode Runge-Kutta Orde Tiga

Metode Runge-Kutta yang terkenal dan banyak dipakai dalam praktek adalah **metode Runge-Kutta orde tiga** dan **metode Runge-Kutta orde empat**. Kedua metode tersebut terkenal karena tingkat ketelitian solusinya tinggi (dibandingkan metode Runge-Kutta orde sebelumnya, mudah diprogram, dan stabil (akan dijelaskan kemudian).

Metode Runge-Kutta orde tiga berbentuk:

$$\begin{aligned} k_1 &= hf(x_r, y_r) \\ k_2 &= hf(x_r + \frac{1}{2}h, y_r + \frac{1}{2}k_1) \\ k_3 &= hf(x_r + h, y_r - k_1 + 2k_2) \\ y_{r+1} &= y_r + \frac{1}{6}(k_1 + 4k_2 + k_3) \end{aligned} \quad (\text{P.8.32})$$

Galat per langkah metode R-K orde tiga adalah $O(h^4)$.

Galat longgokan metode R-K orde tiga adalah $O(h^3)$.

Program 8.5 Metode Runge-Kutta Orde 3

```
function y_RK3(x0, y0, b, h:real):real;
{menghitung y(b) dengan metode Runge-Kutta orde tiga pada PDB
  y'=f(x,y);    y(x0)=y0
}
var
  r, n: integer;
  x, y, k1, k2, k3: real;
begin
  n:=(b - x0)/h; {jumlah langkah}
  y:=y0;           {nilai awal}
  x:=x0;
  for r:=1 to n do

    begin
      k1:=h*f(x, y);
      k2:=h*f(x + h/2, y + k1/2);
      k3:=h*f(x + h, y - k1 + 2*k2);
      y:=y + (k1 + 4*k2 + k3)/6           { nilai y(xr) }
      x:=x+h;                            { titik berikutnya}
    end;
  y_RK3:=y;
end;
```

8.8.4 Metode Runge-Kutta Orde Empat

Metode Runge-Kutta orde empat adalah

$$\begin{aligned}k_1 &= hf(x_r, y_r) \\k_2 &= hf(x_r + \frac{1}{2}h, y_r + \frac{1}{2}k_1) \\k_3 &= hf(x_r + \frac{1}{2}h, y_r + \frac{1}{2}k_2) \\k_4 &= hf(x_r + h, y_r + k_3) \\y_{r+1} &= y_r + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)\end{aligned}\tag{P.8.33}$$

Galat per langkah metode Runge-Kutta orde empat adalah $O(h^3)$.

Galat longgokan metode Runge-Kutta orde empat adalah $O(h^2)$.

Program 8.6 Metode Runge-Kutta Orde 4

```
function y_RK4(x0, y0, b, h:real):real;
{menghitung y(b) dengan metode Runge-Kutta orde empat pada PDB
y'=f(x,y);      y(x0)=y0
}
var
  r, n: integer;
  x, y, k1, k2, k3, k4: real;
begin
  n:=(b - x0)/h;          {jumlah langkah}
  y:=y0;                   {nilai awal}
  x:=x0;
  for r:=1 to n do
    begin
      k1:=h*f(x, y);
      k2:=h*f(x + h/2, y + k1/2);
      k3:=h*f(x + h/2, y + k2/2);
      k4:=h*f(x + h, y + k3);
      y:=y + (k1 + 2*k2 + 2*k3 + k4)/6      { nilai y(xr) }
      { titik berikutnya}
      x:=x+h;
    end;
  y_RK4:=y;
end;
```

Contoh 8.6

Diketahui PDB

$$= \frac{dy}{dx} 1 + y^2 ; y(0) = 0$$

Tentukan $y(0.20)$ dengan metode Runge-Kutta orde tiga. Gunakan ukuran langkah $h = 0.10$.

Penyelesaian:

Diketahui

$$a = x_0 = 0$$

$$b = 0.20$$

$$h = 0.10$$

maka $n = (0.20 - 0)/0.10 = 2$ (jumlah langkah)

Langkah:

$$\begin{aligned}x_0 &= 0 \rightarrow y_0 = 0 \\x_1 &= 0.10 \rightarrow y_1 = ?\end{aligned}$$

$$\begin{aligned}k_1 &= hf(x_0, y_0) = (0.10)(1 + 0^2) = 0.10 \\k_2 &= hf(x_0 + \frac{1}{2}h, y_0 + \frac{1}{2}k_1) = (0.10)(1 + 0.05^2) = 0.10025 \\k_3 &= hf(x_0 + h, y_0 - k_1 + 2k_2) = (0.10)(1 + 0.1005^2) = 0.10101 \\y_1 &= y_0 + \frac{1}{6}(k_1 + 4k_2 + k_3) \\&= 0 + \frac{1}{6}(0.10 + 4 \times 0.10025 + 0.10101) = 0.10034\end{aligned}$$

$$\begin{aligned}x_2 &= 0.20 \rightarrow y_2 = ? \\k_1 &= hf(x_1, y_1) = (0.10)(1 + 0.10034^2) = 0.10101 \\k_2 &= hf(x_1 + \frac{1}{2}h, y_1 + \frac{1}{2}k_1) = (0.10)(1 + 0.150845^2) = 0.10228 \\k_3 &= hf(x_1 + h, y_1 - k_1 + 2k_2) = (0.10)(1 + 0.20389^2) = 0.10416 \\y_2 &= y_1 + \frac{1}{6}(k_1 + 4k_2 + k_3) \\&= 0.10034 + \frac{1}{6}(0.10101 + 4 \times 0.10228 + 0.10416) \\&= 0.20272\end{aligned}$$

Jadi, $y(0.20) \approx 0.20272$.

Nilai sejati $\rightarrow y(0.20) = 0.20271$. ■

Metode Runge-Kutta orde yang lebih tinggi tentu memberikan solusi yang semakin teliti. Tetapi ketelitian ini harus dibayar dengan jumlah komputasi yang semakin banyak. Jadi ada timbal-balik (*trade-off*) dalam memilih suatu metode Runge-Kutta.

8.9 Ekstrapolasi Richardson

Ekstrapolasi Richardson dapat diterapkan untuk memperbaiki solusi PDB dan memperkirakan galatnya, asal kita mengetahui orde metode PDB. Mula-mula solusi PDB dihitung dengan ukuran langkah h . Kemudian solusinya dihitung lagi tetapi dengan ukuran langkah $2h$. Maka, solusi yang lebih baik adalah

$$y(x) = y(x; h) + \frac{1}{2^p - 1} [y(x; h) - y(x; 2h)] \quad (\text{P.8.34})$$

yang dalam hal ini,

- $y(x; h)$ = solusi PDB di x dengan ukuran langkah h
 $y(x; 2h)$ = solusi PDB di x dengan ukuran langkah $2h$
 $y(x)$ = solusi PDB yang lebih baik.
 p = orde metode PDB yang digunakan

taksiran galatnya adalah

$$\epsilon = \frac{1}{2^p - 1} [y(x; h) - y(x; 2h)] \quad (\text{P.8.35})$$

Bila kita tidak mengetahui p , maka nilai perkiraan ketiga, $y(x; 4h)$ memungkinkan kita menggunakan ekstrapolasi Aitken sebagai pengganti ekstrapolasi Richardson. Lihat kembali Bab Integrasi Numerik.

8.10 Metode Banyak-Langkah

Sampai sejauh ini kita telah mengenal metode Euler, metode Heun, metode deret Taylor, dan metode Runge-Kutta. Semua metode tersebut dikelompokkan ke dalam **metode satu-langkah (one-step)**, sebab untuk menaksir nilai $y(x_{r+1})$ dibutuhkan satu buah taksiran nilai sebelumnya, $y(x_r)$.

Kelompok metode PDB yang lain ialah **metode banyak-langkah (multi-step)**. Pada metode banyak-langkah, perkiraan nilai $y(x_{r+1})$ membutuhkan beberapa taksiran nilai sebelumnya, $y(x_r), y(x_{r-1}), y(x_{r-2}), \dots$. Yang termasuk ke dalam metode banyak-langkah adalah metode *predictor-corrector*. Metode Heun adalah metode *predictor-corrector*, namun metode Heun bukanlah metode banyak-langkah, sebab taksiran nilai $y(x_{r+1})$ hanya didasarkan pada taksiran $y(x_r)$.

Tujuan utama metode banyak-langkah adalah menggunakan informasi dari beberapa titik sebelumnya, y_r , y_{r-1} , y_{r-2} , ..., untuk menghitung taksiran nilai y_{r+1} yang lebih baik.

Beberapa metode *predictor-corrector* (P-C) yang termasuk ke dalam metode banyak-langkah. Pada metode P-C, kita menaksir nilai y_{r+1} dari y_r , y_{r-1} , y_{r-2} , ..., dengan persamaan *predictor*, dan kemudian menggunakan persamaan *corrector* untuk menghitung nilai y_{r+1} yang lebih baik (*improve*).

- | | |
|------------------|---|
| <i>predictor</i> | : Menaksir y_{r+1} dari y_r , y_{r-1} , y_{r-2} , ... |
| <i>corrector</i> | : Memperbaiki nilai y_{r+1} dari <i>predictor</i> |

Metode P-C yang banyak ditulis dalam literatur dan kita bahas di sini adalah:

1. Metode Adams-Bashforth-Moulton.
2. Metode Milne-Simpson
3. Metode Hamming

8.10.1 Metode Adams-Bashforth-Moulton

Tinjau PDB orde satu

$$y'(x) = f(x, y(x))$$

Integrasikan kedua ruas persamaan dari x_r sampai x_{r+1} :

$$\begin{aligned} \int_{x_r}^{x_{r+1}} f(x, y(x)) dx &= \int_{x_r}^{x_{r+1}} y'(x) dx \\ &= y(x) \Big|_{x_r}^{x_{r+1}} \\ &= y(x_{r+1}) - y(x_r) \\ &= y_{r+1} - y_r \end{aligned}$$

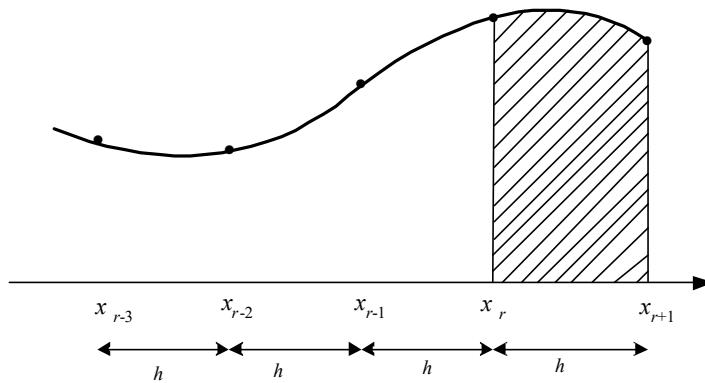
Nyatakan y_{r+1} di ruas kiri persamaan dan suku lainnya di ruas kanan:

$$y_{r+1} = y_r + \int_{x_r}^{x_{r+1}} f(x, y(x)) dx \quad (\text{P.8.34})$$

Persamaan (P.8.34) ini adalah teorema dasar kalkulus (lihat Bab 2, Integral), yang merupakan dasar penurunan persamaan *predictor* dan persamaan *corrector*.

Persamaan Predictor [MAT93]

Persamaan *predictor* diperoleh dengan menghampiri fungsi $f(x, y(x))$ ke dalam polinom interpolasi derajat tiga. Untuk itu, diperlukan empat buah titik yang berjarak sama, yaitu: (x_{r-3}, f_{r-3}) , (x_{r-2}, f_{r-2}) , (x_{r-1}, f_{r-1}) , (x_r, f_r) . Perhatikan Gambar 8.6.



Gambar 8.6 Pembentukan persamaan *predictor*

Dari empat buah titik tersebut, bentuklah polinom interpolasi Lagrange derajat tiga:

$$\begin{aligned}
 f(x, y(x)) \approx & \frac{(x - x_{r-2})(x - x_{r-1})(x - x_r) f_{r-3}}{(x_{r-3} - x_{r-2})(x_{r-3} - x_{r-1})(x_{r-3} - x_r)} + \\
 & \frac{(x - x_{r-3})(x - x_{r-1})(x - x_r) f_{r-2}}{(x_{r-2} - x_{r-3})(x_{r-2} - x_{r-1})(x_{r-2} - x_r)} + \\
 & \frac{(x - x_{r-3})(x - x_{r-2})(x - x_r) f_{r-1}}{(x_{r-1} - x_{r-3})(x_{r-1} - x_{r-2})(x_{r-1} - x_r)} + \\
 & \frac{(x - x_{r-3})(x - x_{r-2})(x - x_{r-1}) f_r}{(x_r - x_{r-3})(x_r - x_{r-2})(x_r - x_{r-1})}
 \end{aligned}$$

$$\begin{aligned}
&\approx -\frac{1}{6h^3} (x - x_{r-2})(x - x_{r-1})(x - x_r) f_{r-3} + \frac{1}{2h^3} (x - x_{r-3})(x - x_{r-1})(x - x_r) f_{r-2} \\
&\quad - \frac{1}{2h^3} (x - x_{r-3})(x - x_{r-2})(x - x_r) f_{r-1} + \frac{1}{2h^3} (x - x_{r-3})(x - x_{r-2})(x - x_{r-1}) f_r
\end{aligned} \tag{P.8.35}$$

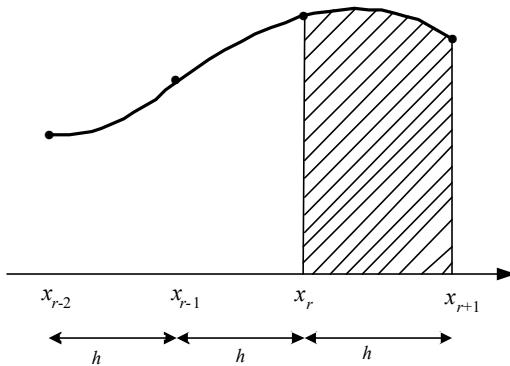
Sulihkan (P.8.35) ke dalam persamaan (P.8.34). Hasil integrasi persamaan (P.8.34) memberikan:

$$y_{r+1}^* = y_r + \frac{h}{24} (-9f_{r-3} + 37f_{r-2} - 59f_{r-1} + 55f_r) \tag{P.8.36}$$

yang merupakan persamaan *predictor*.

Persamaan Corrector [MAT93]

Persamaan *corrector* dibentuk dengan cara yang sama seperti pada persamaan *predictor*. Tetapi, titik-titik yang diperlukan untuk pembentukan polinom interpolasi (Gambar 8.7) ialah (x_{r-2}, f_{r-2}) , (x_{r-1}, f_{r-1}) , (x_r, f_r) , dan titik baru $(x_{r+1}, f_{r+1}) = (x_{r+1}, f(x_{r+1}, y_{r+1}^*))$



Gambar 8.7 Pembentukan persamaan corrector

Dari empat buah titik tersebut, bentuklah polinom interpolasi Lagrange derajat tiga. Kemudian, integrasikan polinom interpolasi tersebut dalam selang $[x_r, x_{r+1}]$, untuk memberikan

$$y_{r+1} = y_r + \frac{h}{24} (f_{r-2} - 5f_{r-1} + 19f_r + 9f_{r+1}^*) \quad (\text{P.8.37})$$

yang merupakan persamaan *corrector*.

Jadi, metode Adams-Bashforth-Moulton dapat diringkas sebagai berikut:

$$\text{predictor : } y_{r+1}^* = y_r + \frac{h}{24} (-9f_{r-3} + 37f_{r-2} - 59f_{r-1} + 55f_r) \quad (\text{P.8.38})$$

$$\text{corrector : } y_{r+1} = y_r + \frac{h}{24} (f_{r-2} - 5f_{r-1} + 19f_r + 9f_{r+1}^*) \quad (\text{P.8.39})$$

Galat per langkah metode Adams-Bashforth-Moulton adalah dalam orde $O(h^5)$, yaitu:

$$\text{predictor : } E_p = Y_{r+1} - y_{r+1}^* \approx \frac{251}{720} h^5 y^{(5)}(t), \quad x_{r-3} < t < x_{r+1}$$

$$\text{corrector : } E_p = Y_{r+1} - y_{r+1} \approx \frac{-19}{720} h^5 y^{(5)}(t), \quad x_{r-3} < t < x_{r+1}$$

dan galat longgokan adalah dalam orde $O(h^4)$. Karena itu, metode Adams-Bashforth-Moulton di atas dinamakan juga **metode Adams-Bashforth-Moulton orde-4**.

Metode yang lebih rendah adalah **metode Adams-Bashforth-Moulton orde-3**:

$$\text{predictor : } y_{r+1}^* = y_r + \frac{h}{12} (23f_r - 16f_{r-1} + 5f_{r-2})$$

$$\text{corrector : } y_{r+1} = y_r + \frac{h}{12} (5f_{r+1}^* + 18f_r - f_{r-1})$$

Pada waktu penurunan persamaan *predictor* Adams-Bashforth-Mouton orde-3 ini, polinom interpolasinya memerlukan tiga buah titik, yaitu (x_{r-2}, f_{r-2}) , (x_{r-1}, f_{r-1}) , (x_r, f_r) , sedangkan pada waktu penurunan persamaan *predictor*, polinom interpolasinya memerlukan titik-titik (x_{r-1}, f_{r-1}) , (x_r, f_r) , (x_{r+1}, f_{r+1}^*) .

Galat per langkahnya adalah dalam orde $O(h^4)$, yaitu:

$$\text{predictor : } E_p = Y_{r+1} - y_{r+1}^* \approx \frac{9}{24} h^4 y''(t), \quad x_{r-2} < t < x_{r+1}$$

$$\text{corrector : } E_p = Y_{r+1} - y_{r+1} \approx \frac{-1}{24} h^4 y''(t), \quad x_{r-2} < t < x_{r+1}$$

dan galat longgokan adalah dalam orde $O(h^3)$.

Cara menurunkan persamaan galat metode *predictor-corrector* sama seperti cara yang sudah dijelaskan sebelumnya. Misalnya kita akan menurunkan persamaan galat metode Adams-Basforth-Moulton orde-3 sebagai berikut:

Uraikan persamaan predictor, *corrector* dan y_{r+1} sejati di sekitar x_r .

Predictor

$$\begin{aligned}
 \text{Hampiran : } y_{r+1}^* &= y_r + \frac{h}{12} (23f_r - 16f_{r-1} + 5f_{r-2}) \\
 &= y_r + \frac{h}{12} [23f_r - 16(f_r - hf'_r + \frac{1}{2}h^2f''_r - \frac{1}{6}h^3f'''_r + \dots) \\
 &\quad + 5(f_r - 2hf'_r + 2h^2f''_r - \frac{8}{6}h^3f'''_r + \dots)] \\
 &= y_r + \frac{h}{12} [12f_r + 6hf'_r + 2h^2f''_r - 4h^3f'''_r + \dots] \\
 &= y_r + hf'_r + \frac{1}{2}h^2f''_r + \frac{1}{6}h^3f'''_r - \frac{1}{3}h^4f''''_r + \dots
 \end{aligned}$$

$$\begin{aligned}
 \text{Sejati: } Y_{r+1} &= y_r + hy'_r + \frac{1}{2}h^2y''_r + \frac{1}{6}h^3y'''_r + \frac{1}{24}h^4y^{(4)}_r + \dots \\
 &= y_r + hf'_r + \frac{1}{2}h^2f''_r + \frac{1}{6}h^3f'''_r + \frac{1}{24}h^4f''''_r + \dots
 \end{aligned}$$

Galat per langkah *predictor*:

$$\begin{aligned}
 E_p &= \text{sejati} - \text{hampiran} \\
 &= \frac{1}{24}h^4f''''_r + \frac{1}{3}h^4f''''_r + \dots \\
 &= \frac{9}{24}h^4f''''_r = \frac{9}{24}h^4y''(t) \quad , \quad x_{r-2} < t < x_{r+1} \\
 &= O(h^4)
 \end{aligned}$$

Corrector

$$\begin{aligned}
 \text{Hampiran : } y_{r+1} &= y_r + \frac{h}{12} (5f_{r+1}^* + 8f_r - f_{r-1}) \\
 &= y_r + \frac{h}{12} [5(f_r + hf'_r + \frac{1}{2}h^2f''_r + \frac{1}{6}h^3f'''_r + \dots) + 8f_r \\
 &\quad (f_r - hf'_r + \frac{1}{2}h^2f''_r - \frac{1}{6}h^3f'''_r + \dots)] \\
 &= y_r + \frac{h}{12} (12f_r + 6hf'_r + 2h^2f''_r + h^3f'''_r + \dots) \\
 &= y_r + hf'_r + \frac{1}{2}h^2f''_r + \frac{1}{6}h^3f'''_r + \frac{1}{12}h^4f''''_r + \dots
 \end{aligned}$$

Galat per langkah *corrector*:

$$\begin{aligned}
 E_p &= \text{sejati} - \text{hampiran} \\
 &= \frac{1}{24}h^4f''''_r - \frac{1}{12}h^4f''''_r + \dots
 \end{aligned}$$

$$\begin{aligned}
&= -1/24 h^4 f_r''' = -1/24 h^4 y'''(t) \quad , \quad x_{r-2} < t < x_{r+1} \\
&= O(h^4)
\end{aligned}$$

Orde metode = 4 - 1 = 3.

8.10.2 Metode Milne-Simpson

Metode Milne-Simpson didasarkan pada integrasi $f(x, y(x))$ pada selang $[x_{r-3}, x_{r+1}]$:

$$y(x_{r+1}) = y(x_{r-3}) + \int_{x_{r-3}}^{x_{r+1}} f(x, y(x)) dx \quad (\text{P.8.40})$$

Persamaan *predictor* dan *corrector* metode Milne-Simpson adalah

$$\text{predictor : } y_{r+1}^* = y_{r-3} + \frac{4h}{3} (2f_{r-2} - f_{r-1} + 2f_r) \quad (\text{P.8.41})$$

$$\text{corrector : } y_{r+1} = y_{r-1} + \frac{h}{3} (f_{r-1} + 4f_r + f_{r+1}) \quad (\text{P.8.42})$$

dan galat per langkahnya adalah dalam orde $O(h^5)$, yaitu:

$$\text{predictor : } E_p = Y_{r+1} - y_{r+1}^* \approx \frac{28h^5}{90} y^{(5)}(t)$$

$$\text{corrector : } E_p = Y_{r+1} - y_{r+1} \approx \frac{-1h^5}{90} y^{(5)}(t)$$

untuk $x_{r-3} < t < x_{r+1}$.

8.10.3 Metode Hamming

Persamaan *predictor* dan *corrector* metode Hamming adalah

$$\text{predictor : } y_{r+1}^* = y_{r-3} + \frac{4h}{3} (2f_{r-2} - f_{r-1} + 2f_r) \quad (\text{P.8.43})$$

$$\text{corrector : } y_{r+1} = \frac{-y_{r-2}}{8} + \frac{9y_r}{8} + \frac{3h}{8} (-f_{r-1} + 2f_r + f_{r+1}) \quad (\text{P.8.44})$$

8.10.4 Prosedur Pendahuluan

PDB hanya mempunyai satu nilai awal, yaitu $y_0 = y(x_0)$. Dengan demikian, metode banyak-langkah tidak swa-mulai (*self-start*), sehingga tidak dapat diterapkan langsung, sebab metode tersebut memerlukan beberapa buah nilai awal. Inilah kelemahan metode banyak-langkah.

Misalkan *predictor* mempunyai persamaan

$$y^*_{r+1} = y_r + \frac{h}{12} (23f_r - 16f_{r-1} + 5f_{r-2})$$

Untuk menghitung y^*_3 , kita harus mempunyai nilai y_0, y_1 , dan y_2 agar nilai

$$f_0 = f(x_0, y_0), f_1 = f(x_1, y_1), f_2 = f(x_2, y_2)$$

dapat ditentukan. Untuk mendapatkan beberapa nilai awal yang lain, kita harus melakukan prosedur pendahuluan (*starting procedure*) dengan metode PDB yang bebas. Metode PDB yang sering dijadikan sebagai prosedur pendahuluan adalah:

- metode Euler
- metode Runge-Kutta
- metode deret Taylor

Jadi, untuk contoh *predictor* di atas, y_1 dan y_2 dihitung terlebih dahulu dengan salah satu prosedur pendahuluan. Selanjutnya, metode P-C dapat dipakai untuk menghitung y_3, y_4, \dots, y_n .

Program 8.7 Metode Adams-Bashforth-Moulton

```
function y_Adams_Bashforth_Moulton(x0, y0, b, h:real):real;
  {menghitung y(b) dengan metode Adams_Bashforth_moulton pada PDB
   y' = f(x,y);      y(x0) = y0 }
var
  r, n: integer;
  x, y, y0, y1, y2, y3 : real;
begin
  n:=(b-x0)/h; {jumlah langkah}
  y0:=y0;          {nilai awal dari PDB}

  {Prosedur pendahuluan untuk menghitung nilai awal lain, y1, y2, y3}
  y1:=y_RK3(x0, y0, x0+h, h);           {y(x1)}
  y2:=y_RK3(x0, y0, x0+2*h, h);         {y(x2)}
  y3:=y_RK3(x0, y0, x0+3*h, h);         {y(x3)}
  x:=x0 + 3*h;                          {x3 }

  for r:=4 to n do
    begin
      y:=y3 + h/24*(-9*f(x-3*h, y0) + 37*f(x-2*h, y1) - 59*f(x-h, y2)
                    + 55*f(x, y3));
      y:=y3 + h/24*(f(x-2*h, y1) - 5*f(x-h, y2) + 19*f(x, y3)
                    + 9*f(x+h, y));
      y0:=y1;
      y1:=y2;
```

```

y2:=y3;
y3:=y;
x:=x+h;           ( titik berikutnya )
end;
y_Adams_Bashforth_Moulton:=y;
end;

```

8.10.5 Ke idealan Metode Predictor-Corrector

Metode *predictor-corrector* dikatakan ideal jika galat per langkah *predictor* mempunyai orde yang sama dengan galat per langkah *corrector*:

$$\text{galat per langkah } \textit{predictor} : Y_{r+1} - y^*_{r+1} \approx A_r h^p$$

$$\text{galat per langkah } \textit{corrector} : Y_{r+1} - y_{r+1} \approx \alpha A_r h^p$$

dengan α adalah tetapan yang diketahui. Metode Adams-Bashforth-Moulton, metode Milne-Simpson, dan metode Hamming adalah metode P-C yang ideal. Metode Heun adalah metode P-C yang tidak ideal, karena

$$\text{galat per langkah } \textit{predictor} : E_p = Y_{r+1} - y_{r+1} \approx \frac{1}{2} y''(t)h^2 \approx Ah^2$$

$$\text{galat per langkah } \textit{corrector} : E_p = Y_{r+1} - y_{r+1} \approx -\frac{1}{12} y'''(t)h^3 \approx Bh^3$$

Jika sebuah metode P-C ideal, kita dapat memperoleh nilai y_{r+1} yang lebih baik (*improve*) sebagai berikut:

$$\bar{y}_{r+1} - y^*_{r+1} = A_r h^p \quad (\text{P.8.45})$$

$$\bar{y}_{r+1} - y_{r+1} = \alpha A_r h^p \quad (\text{P.8.46})$$

dengan \bar{y}_{r+1} adalah taksiran yang lebih baik dari pada y_{r+1} .

Rumus \bar{y}_{r+1} dapat diperoleh dengan membagi persamaan (P.8.45) dengan persamaan (P.8.46):

$$\begin{aligned} \frac{\bar{y}_{r+1} - y^*_{r+1}}{\bar{y}_{r+1} - y_{r+1}} &= \frac{A_r h^p}{\alpha A_r h^p} = \frac{1}{\alpha} \\ \Leftrightarrow \bar{y}_{r+1} - Y_{r+1} &= \alpha y_{r+1} - \alpha y^*_{r+1} \end{aligned}$$

$$\begin{aligned}
&\Leftrightarrow \bar{y}_{r+1} (1 - \alpha) = y_{r+1} - \alpha y_{r+1}^* \\
&\Leftrightarrow \bar{y}_{r+1} = \frac{y_{r+1} - \alpha y_{r+1}^*}{1 - \alpha} \\
&\Leftrightarrow \bar{y}_{r+1} = \frac{y_{r+1}}{1 - \alpha} - \frac{\alpha y_{r+1}^*}{1 - \alpha} \\
&\Leftrightarrow \bar{y}_{r+1} = \frac{(1 - \alpha)y_{r+1} + \alpha y_{r+1}^*}{1 - \alpha} - \frac{\alpha y_{r+1}^*}{1 - \alpha} \\
&\Leftrightarrow \bar{y}_{r+1} = \frac{(1 - \alpha)y_{r+1}}{1 - \alpha} + \frac{y_{r+1}}{1 - \alpha} - \frac{\alpha y_{r+1}^*}{1 - \alpha} \\
&\Leftrightarrow \bar{y}_{r+1} = y_{r+1} + \frac{\alpha}{1 - \alpha} (y_{r+1} - y_{r+1}^*) \tag{P.8.47}
\end{aligned}$$

Suku $\frac{\alpha}{(1-\alpha)} (y_{r+1} - y_{r+1}^*)$ pada persamaan (P.8.47) merupakan taksiran galat per langkah untuk menghitung \bar{y}_{r+1} , dan menyatakan faktor koreksi terhadap nilai y_{r+1} . Jadi, untuk mendapatkan taksiran nilai y_{r+1} yang lebih baik, tambahkan y_{r+1} dengan faktor koreksi tersebut.

Contoh 8.7

Tentukan perkiraan galat per langkah untuk nilai y_{r+1} yang lebih baik dengan metode Adams-Bashforth-Moulton.

Penyelesaian:

$$\begin{aligned}
\text{galat per langkah predictor: } E_p &= y_{r+1} - y_{r+1}^* \approx \frac{251}{720} y^{(5)}(t) h^5 \\
\text{galat per langkah corrector: } E_p &= y_{r+1} - y_{r+1} \approx \frac{-19}{720} y^{(5)}(t) h^5
\end{aligned}$$

Dari persamaan galat di atas, diperoleh

$$A_r = 251/720 \quad \text{dan} \quad \alpha A_r = -19/720$$

Nilai α ditentukan sebagai berikut:

$$\begin{aligned}
&\Leftrightarrow \alpha A_r = -19/720 \\
&\Leftrightarrow \alpha(251/720) = (-19/720) \\
&\Leftrightarrow \alpha = -19/251
\end{aligned}$$

Sehingga,

$$\begin{aligned}\bar{y}_{r+1} &= y_{r+1} + \frac{(-19/251)}{1+19/251} (y_{r+1} - y^*_{r+1}) \\ &= y_{r+1} - \frac{19}{270} (y_{r+1} - y^*_{r+1})\end{aligned}$$

Jadi, taksiran galat per langkah untuk nilai y_{r+1} adalah

$$E_p \approx -19/270 (y_{r+1} - y^*_{r+1})$$

■

8.11 Pemilihan Ukuran Langkah yang Optimal

Ukuran langkah h adalah persoalan yang penting pada metode PDB yang berdasarkan langkah per langkah ini. Jika h terlalu kecil, jumlah langkahnya semakin banyak dan galat pembulatannya semakin besar. Sebaliknya, jika h terlalu besar, galat pemotongannya juga bertambah besar karena galat pemotongan sebanding dengan h . Timbul pertanyaan: berapakah ukuran langkah yang optimal agar galat per langkah metode PDB dapat dipertahankan kurang dari ε ?

Misalkan kita menghitung solusi PDB dengan metode Runge-Kutta orde-4. Kita ingin galat per langkahnya kurang dari ε . Galat per langkah metode Runge-Kutta orde-4 berbentuk

$$E_p(h) = Bh^5 \quad (\text{P.8.48})$$

dengan B adalah konstanta yang bergantung pada soal yang diberikan. Agar $E_t(h)$ kurang dari ε ,

$$Bh^5 < \varepsilon$$

maka ukuran langkah h haruslah

$$h < (\varepsilon/B)^{1/5} \quad (\text{P.8.49})$$

Konstanta B ditentukan dengan cara percobaan sebagai berikut:

1. Hitung $y(x_1)$ dengan ukuran langkah h (disimbolkan dengan $y(x_1;h)$). Galat per langkahnya dinyatakan oleh persamaan (P.8.48).

2. Hitung kembali $y(x_1)$ dengan ukuran langkah $h/2$ (disimbolkan dengan $y(x_2; h/2)$). Jadi, perlu dua langkah untuk menghitung $y(x_1)$ dengan galat tiap langkah per langkah seluruhnya adalah:

$$\begin{aligned} \Leftrightarrow E_p(h/2) + E_p(h/2) &= B(h/2)^5 + B(h/2)^5 \\ \Leftrightarrow 2 E_p(h/2) &= 2B(h/2)^5 = \frac{Bh^5}{16} \end{aligned} \quad (\text{P.8.50})$$

3. Kurangi (P.8.48) dengan (P.8.50):

$$E_p(h) - 2 E_p(h/2) = B(h)^5 - \frac{1}{16} Bh^5 = \frac{15}{16} Bh^5 \quad (\text{P.8.51})$$

4. Ruas kiri persamaan (P.8.51) dihitung sebagai

$$E_p(h) - 2 E_p(h/2) = y(x_1; h) - y(x_2; h/2) \quad (\text{P.8.52})$$

5. Samakan persamaan (P.8.51) dengan persamaan (P.8.52):

$$\frac{15}{16} Bh^5 = y(x_1; h) - y(x_2; h/2)$$

sehingga diperoleh

$$B = \frac{16}{15} \frac{y(x_1; h) - y(x_2; h/2)}{h^5} \quad (\text{P.8.53})$$

6. Sulihkan nilai B dalam ketidaksamaan (P.8.49) sehingga diperoleh batas maksimum nilai ukuran langkah h .

Contoh 8.8

Diberikan PDB

$$y' = y/(1+x^2), y(0)=1$$

Tentukan ukuran langkah h agar galat per langkah kurang dari 0.00001.

Penyelesaian:

Diketahui:

$$\begin{aligned}x_0 &= 0 \rightarrow y_0 = 1 \\ \epsilon &= 0.00001\end{aligned}$$

Dengan ukuran langkah $h = 1$ dan $h = 0.5$, metode Runge-Kutta orde-4 menghasilkan

$$\begin{aligned}y_1 &= y(1; 1) = 0.4566667 \\ y_1 &= y(1; 0.5) = 0.4559973\end{aligned}$$

Nilai B dihitung dengan persamaan (P.8.53):

$$B = \frac{16}{15} \frac{(0.4566667 - 0.4559973)}{1^5} = 0.00063$$

Jadi, ukuran langkah yang optimal agar galat per langkah metode Runge-Kutta orde-4 kurang dari ϵ ialah

$$h < (0.00001/0.00063)^{1/5} = 0.44$$

■

8.12 Sistem Persamaan Diferensial

Dalam bidang sains dan rekayasa, persamaan diferensial banyak muncul dalam bentuk simultan, yang dinamakan **sistem persamaan diferensial**, sebagai berikut:

$$\begin{aligned}y'_1 &= \frac{dy_1}{dx} = f_1(x, y_1, y_2, \dots, y_n) & , y_1(x_0) &= y_{10} \\ y'_2 &= \frac{dy_2}{dx} = f_2(x, y_1, y_2, \dots, y_n) & , y_2(x_0) &= y_{20} \\ &\vdots \\ y'_n &= \frac{dy_n}{dx} = f_n(x, y_1, y_2, \dots, y_n) & , y_n(x_0) &= y_{n0}\end{aligned} \quad (\text{P.8.54})$$

Sistem persamaan diferensial tersebut dapat ditulis dalam notasi vektor sebagai berikut:

$$\mathbf{y}' = \mathbf{f}(x, \mathbf{y}) \quad , \quad \mathbf{y}(\mathbf{x}_0) = \mathbf{y}_0 \quad (\text{P.8.55})$$

yang dalam hal ini,

$$\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ \vdots \\ y_n \end{bmatrix}, \quad \mathbf{y}' = \begin{bmatrix} y'_1 \\ y'_2 \\ \vdots \\ \vdots \\ y'_n \end{bmatrix}, \quad \mathbf{f} = \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ \vdots \\ f_n \end{bmatrix}, \quad \mathbf{y}_0 = \begin{bmatrix} y_{1_0} \\ y_{2_0} \\ \vdots \\ \vdots \\ y_{n_0} \end{bmatrix}$$

Semua metode yang telah dijelaskan untuk persamaan tunggal (Euler, Runge-Kutta, dll.) dapat diterapkan pada sistem persamaan di atas.

Contoh 8.9

Diketahui sistem PDB orde-1

$$\frac{dy}{dt} = -0.5y, \quad y(0) = 4$$

$$\frac{dz}{dt} = 4 - 0.3z - 0.1y, \quad z(0) = 6$$

Hitung $y(0.5)$ dan $z(0.5)$ dengan (a) metode Euler, dan (b) metode Runge-Kutta orde 3. Ambil $h = 0.5$.

Penyelesaian:

$$\mathbf{y} = \begin{bmatrix} y \\ z \end{bmatrix}, \quad \mathbf{y}' = \begin{bmatrix} y' \\ z' \end{bmatrix}, \quad \mathbf{f} = \begin{bmatrix} f_1 \\ f_2 \end{bmatrix} = \begin{bmatrix} -0.5y \\ 4 - 0.3z - 0.1y \end{bmatrix}, \quad \mathbf{y}_0 = \begin{bmatrix} 4 \\ 6 \end{bmatrix}$$

Sistem PDB di atas dapat ditulis menjadi $\mathbf{y}' = \mathbf{f}(t, \mathbf{y}), \mathbf{y}(t_0) = \mathbf{y}_0$

(a) Dengan metode Euler $y_{r+1} = y_r + h f(t_r, y_r)$:

$$y_{r+1} = y_r + h f_1(t_r, y_r, z_r)$$

$$z_{r+1} = z_r + h f_2(t_r, y_r, z_r)$$

$$t_0 = 0 \rightarrow y_0 = 4 \text{ dan } z_0 = 6$$

$$t_0 = 0.5 \rightarrow y_1 = y(0.5) = y_0 + h f_1(t_0, y_0, z_0) = 4 + (0.5)\{(-0.5)(4)\} = 3$$

$$z_1 = z(0.5) = z_0 + h f_2(t_0, y_0, z_0)$$

$$= 6 + (0.5)\{4 - (0.3)(6) - (0.1)(4)\} = 6.9$$

(b) Dengan metode Runge-Kutta orde-3,

$$\begin{aligned}k_1 &= hf(t_r, y_r), \\k_2 &= hf(t_r + h/2, y_r + k_1/2) \\k_3 &= hf(t_r + h, y_r - k_1 + 2k_2) \\y_{r+1} &= y_r + (1/6)(k_1 + 4k_2 + k_3)\end{aligned}$$

$$\begin{aligned}t_0 &= 0 &\rightarrow y_0 = 4 \\t_1 &= 0.5 &\rightarrow y_1 = ?\end{aligned}$$

$$\begin{aligned}k_1 &= hf_1(t_0, y_0, z_0) \\&= 0.5 \{(-0.5)(4)\} = -1 \\k_2 &= hf_1(t_0 + h/2, y_0 + k_1/2, z_0 + k_1/2) \\&= (0.5)f_1(0.25, 3.5, 5.5) \\&= (0.5)\{(-0.5)(3.5)\} \\&= -0.875\end{aligned}$$

$$\begin{aligned}k_3 &= hf_1(t_0 + h, y_0 - k_1 + 2k_2, z_0 - k_1 + 2k_2) \\&= 0.5 f_1(0.5, 3.25, 6.815) \\&= 0.5 \{(-0.5)(3.25)\} \\&= -0.8125\end{aligned}$$

sehingga

$$\begin{aligned}y_1 &= y(0.5) = y_0 + \frac{1}{6}(k_1 + 4k_2 + k_3) \\&= 4 + \frac{1}{6} \{-1 + 4(-0.875) + (-0.8125)\} \\&= 3.114583\end{aligned}$$

$$\begin{aligned}t_0 &= 0 &\rightarrow z_0 = 6 \\t_1 &= 0.5 &\rightarrow z_1 = ?\end{aligned}$$

$$\begin{aligned}k_1 &= hf_2(t_0, y_0, z_0) \\&= 0.5 \{4 - (0.3)(6) - (0.1)(4)\} = 0.9 \\k_2 &= hf_2(t_0 + h/2, y_0 + k_1/2, z_0 + k_1/2) \\&= (0.5)f_2(0.25, 4.45, 6.45) \\&= (0.5)\{4 - (0.3)(6.45) - (0.1)(4.45)\} \\&= 0.81\end{aligned}$$

$$\begin{aligned}k_3 &= hf_2(t_0 + h, y_0 - k_1 + 2k_2, z_0 - k_1 + 2k_2) \\&= 0.5 f_2(0.5, 4.72, 6.72) \\&= 0.5 \{4 - (0.3)(6.72) - (0.1)(4.72)\} \\&= 0.756\end{aligned}$$

sehingga

$$\begin{aligned}z_1 &= z(0.5) = z_0 + (1/6)(k_1 + 4k_2 + k_3) \\&= 6 + (1/6) \{0.9 + 4(0.81) + 0.756\} \\&= 6.816\end{aligned}$$

■

8.13 Persamaan Diferensial Orde Lanjut

Persamaan differensial orde lanjut adalah persamaan diferensial dengan orde yang lebih besar dari satu. Persamaan diferensial ini dapat ditulis kembali sebagai sistem persamaan diferensial orde-1.

Misalkan kepada kita diberikan PDB orde-2

$$y'' = f(x, y, y') \quad \text{dengan nilai awal } y(x_0) = y_0 \text{ dan } y'(x_0) = z_0$$

Untuk mengubah PDB orde-2 tersebut menjadi sistem PDB orde-1, misalkan

$$y' = z$$

maka

$$z' = y'' = f(x, y, y') = f(x, y, z) ; y(x_0) = y_0 \text{ dan } z(x_0) = z_0$$

Dengan demikian, persamaan $y'' = f(x, y, y')$ dapat ditulis menjadi sistem persamaan diferensial biasa:

$$\frac{dy}{dx} = z , \quad y(x_0) = y_0$$

$$\frac{dz}{dx} = f(x, y, z) , \quad z(x_0) = z_0$$

atau dalam notasi vektor:

$$\mathbf{y}' = \mathbf{f}(x, \mathbf{y}) ; \mathbf{y}(\mathbf{x}_0) = \mathbf{y}_0$$

yang dalam hal ini

$$\mathbf{y} = \begin{bmatrix} y \\ z \end{bmatrix} , \quad \mathbf{y}' = \mathbf{f} = \begin{bmatrix} z \\ f(x, y, z) \end{bmatrix} , \quad \mathbf{y}(\mathbf{x}_0) = \begin{bmatrix} y_0 \\ z_0 \end{bmatrix}$$

Selanjutnya sistem persamaan diferensial biasa ini sifat diselesaikan seperti pada Contoh 8.9 terdahulu.

Contoh 8.10

Nyatakan PDB orde-2 berikut:

$$y'' - 3y' - 2y = 0 ; y(0) = 1 \text{ dan } y'(0) = 0.5$$

ke dalam sistem persamaan diferensial biasa orde-1.

Penyelesaian:

Diketahui PDB orde-2:

$$y'' = 3y' - 2y = f(x, y, y')$$

Misalkan

$$y' = z$$

maka

$$z' = y'' = f(x, y, y') = f(x, y, z) = 3z - 2y$$

dan

$$\begin{aligned} y(0) &= 1, \\ z(0) &= 0.5; \end{aligned}$$

sehingga diperoleh sistem PDB orde-1

$$\frac{dy}{dx} = z , \quad y(0) = 1$$

$$\frac{dz}{dx} = 3z - 2y , \quad z(0) = 0.5$$

atau dalam notasi vektor:

$$\mathbf{y}' = \mathbf{f}(x, \mathbf{y}) ; \mathbf{y}(0) = \mathbf{y}_0$$

yang dalam hal ini,

$$\mathbf{y} = \begin{bmatrix} y \\ z \end{bmatrix} , \quad \mathbf{f} = \begin{bmatrix} z \\ 3z - 2y \end{bmatrix} , \quad \mathbf{y}_0 = \begin{bmatrix} 1 \\ 0.5 \end{bmatrix}$$

■

Contoh 8.11

Nyatakan PDB orde-3 berikut:

$$y''' - x - y^2 + y' + 3y'' = 0 \quad ; \quad y(0) = 0; y'(0) = 0.5, y''(0) = 1$$

ke dalam sistem persamaan diferensial biasa orde-1.

Penyelesaian:

$$y''' = x - y^2 + y' + 3y'' = f(x, y, y', y'')$$

Misalkan

$$y' = z$$

dan

$$y'' = z' = t$$

maka

$$t' = y''' = f(x, y, y', y'') = f(x, y, z, t) = x - y^2 + z - 3t$$

dan

$$\begin{aligned} y(0) &= 0, \\ z(0) &= 0.5, \\ t(0) &= 1; \end{aligned}$$

sehingga diperoleh sistem PDB orde-1:

$$\frac{dy}{dx} = z \quad , \quad y(0) = 0$$

$$\frac{dz}{dx} = t \quad , \quad z(0) = 0.5$$

$$\frac{dt}{dx} = x - y^2 + z - 3t \quad , \quad t(0) = 1$$

atau dalam notasi vektor

$$\mathbf{y}' = \mathbf{f}(x, \mathbf{y}) \quad , \quad \mathbf{y}(0) = \mathbf{y}_0$$

yang dalam hal ini,

$$\mathbf{y} = \begin{bmatrix} y \\ z \\ t \end{bmatrix}, \quad \mathbf{f} = \begin{bmatrix} z \\ t \\ x - y^2 + z - 3t \end{bmatrix}, \quad \mathbf{y}(0) = \begin{bmatrix} 0 \\ 0.5 \\ 1 \end{bmatrix} \quad \blacksquare$$

Contoh 8.12

Nyatakan PDB orde-2 berikut:

$$2x''(t) - 5x'(t) - 3x(t) = 45e^{2t}, \quad x(0.5) = 2 \text{ dan } x'(0.5) = 1$$

ke dalam sistem PDB orde-1

Penyelesaian:

$$x''(t) = \frac{45}{2} e^{2t} + \frac{5}{2} x'(t) + \frac{3}{2} x(t)$$

Misalkan

$$x'(t) = z(t)$$

maka

$$z'(t) = x''(t) = f(t, x(t), z(t)) = \frac{45}{2} e^{2t} + \frac{5}{2} z(t) + \frac{3}{2} x(t)$$

dan

$$\begin{aligned} x(0) &= 2, \\ z(0) &= 1; \end{aligned}$$

sehingga diperoleh sistem PDB orde-1:

$$\begin{aligned} \frac{dx}{dt} &= z(t) &&, x(0.5) = 2; \\ \frac{dz}{dt} &= \frac{45}{2} e^{2t} + \frac{5}{2} z(t) + \frac{3}{2} x(t) &&, z(0.5) = 1 \end{aligned}$$

atau dalam notasi vektor:

$$\mathbf{y}' = \mathbf{f}(t, \mathbf{y}), \quad \mathbf{y}(0.5) = \mathbf{y}_0$$

yang dalam hal ini,

$$\mathbf{y} = \begin{bmatrix} x(t) \\ z(t) \end{bmatrix}, \quad \mathbf{f} = \begin{bmatrix} z(t) \\ \frac{45}{2} e^{2t} + \frac{5}{2} z(t) + \frac{3}{2} x(t) \end{bmatrix}, \quad \mathbf{y}(0.5) = \begin{bmatrix} 2 \\ 1 \end{bmatrix} \blacksquare$$

8.14 Ke tidakstabilitan Metode PDB

Pada bab-bab sebelumnya kita telah menyenggung ketidakstabilan pada metode numerik. Karena solusi PDB diperoleh secara lelaran, yang setiap lelaran menghasilkan galat pemotongan, maka ada kemungkinan solusi pada lelaran terakhir menyimpang cukup berarti terhadap solusi sejatinya. Untuk jelasnya perhatikan metode PDB berikut:

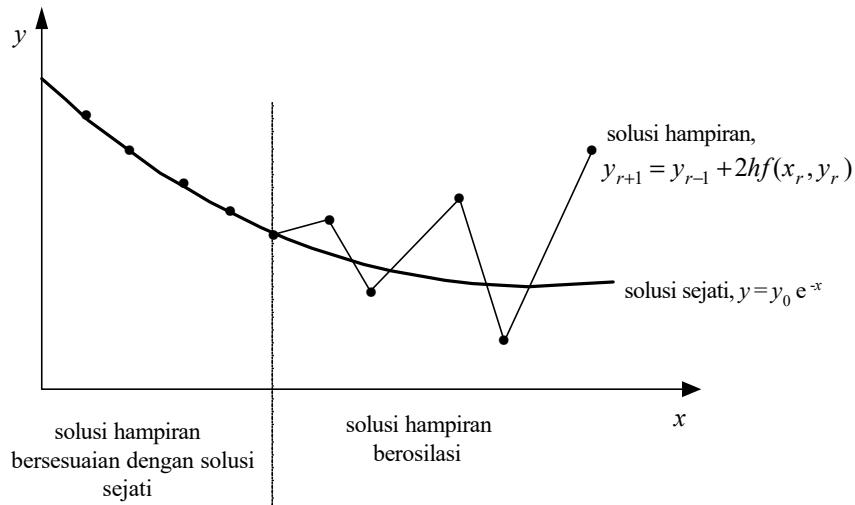
$$y_{r+1} = y_{r-1} + 2hf(x_r, y_r) \quad (\text{P.8.56})$$

Dengan bantuan deret Taylor, galat per langkah metode ini adalah dalam orde $O(h^3)$, yang berarti metode PDB tersebut berorde dua. Kesimpulan sementara kita, metode (P.8.56) menghasilkan solusi yang lebih teliti daripada solusi dengan metoda Euler (yang berorde satu).

Bila metode (P.8.56) diterapkan pada PDB

$$y' = \frac{dy}{dx} = -\alpha y, \quad y(0) = y_0 > 0 \quad \text{dan } \alpha > 0$$

kita memperoleh hasil yang diperlihatkan oleh Gambar 8.8.



Gambar 8.8 Ketidakstabilan metode PDB

Perhatikan bahwa solusi analitik PDB ini adalah $y = y_0 e^{-\alpha x}$, yang mendekati nol dengan peningkatan x . Tetapi solusi numeriknya menjadi tidak stabil dengan

peningkatan x . Ketidakstabilan ini disebabkan oleh penumpukan galat per langkah yang "tumbuh" secara tidak terbatas dengan meningkatnya jumlah langkah. Untuk jumlah langkah yang sedikit, solusinya masih stabil. Tetapi dengan meningkatnya jumlah langkah, solusinya menjadi tidak stabil. Bahkan, untuk jumlah langkah yang tidak terhingga, solusinya tumbuh secara tidak terbatas. Jadi, ada metode PDB yang hanya baik untuk x yang kecil, tetapi buruk untuk x yang besar. Ketidakstabilan ditandai oleh solusi yang berosilasi, tetapi ini tidak selalu demikian.

Dalam praktik, hindari penggunaan metode yang tidak stabil. Contoh metode PDB yang tidak stabil untuk x yang besar adalah metode Euler dan metode Milne-Simpson. Metode Heun, metode Runge-Kutta orde-3, metode Runge-Kutta orde-4, dan metode Adams-Bashforth-Moulton adalah metode PDB yang stabil.

8.15 Contoh Soal Terapan

Pada rangkaian listrik, arus yang mengalir tidaklah tetap, tetapi berubah terhadap waktu. Tinjau kembali rangkaian RLC pada Gambar 8.1.

Hukum Kirchoff menyatakan bahwa jumlah aljabar dari perubahan tegangan di sekeliling rangkaian tertutup adalah nol. Selain dalam bentuk PDB orde-1 (P.8.3), hukum Kirchoff kadang-kadang disajikan dalam bentuk PDB orde-2:

$$L \frac{d^2q}{dt^2} + R \frac{dq}{dt} + q/C - E(t) = 0 \quad (\text{P.8.57})$$

yang dalam hal ini, L adalah induktansi (dalam henry), R adalah tahanan (dalam ohm), q adalah muatan pada kapasitor (dalam coulomb), C adalah kapsitasitansi (dalam farad), $E(t)$ adalah tegangan yang berubah terhadap waktu (dalam volt). Persamaan (P.8.56) adalah PDB orde-2 yang dapat dipecah menjadi sistem PDB orde-1:

$$\begin{aligned} \frac{di}{dt} &= -Ri/L - q/CL + E(t)/L & , i(0) = 0 \\ \frac{dq}{dt} &= i & , q(0) = 0 \end{aligned} \quad (\text{P.8.58})$$

Misalkan $L = 1$ henry, $C = 0.25$ coulomb, $E(t) = E_0 \sin \omega t$, $E_0 = 1$ volt, $\omega = 1.8708$ detik, dan $R = 0$. Hitunglah muatan kapasitor setelah 10 detik dengan metode Euler dan metode Runge-Kutta orde empat (gunakan ukuran langkah $h =$

0.1 detik). Bandingkan jawaban anda dengan solusi analitiknya yang diturunkan sbb :

$$q(t) = \frac{-E_0}{L(p^2 - \omega^2)} \frac{\omega}{p} \sin pt + \frac{E_0}{L(p^2 - \omega^2)} \sin \omega t \quad (\text{P.8.59})$$

dengan $p = 1/\sqrt{LC}$. Dengan menyulihkan besaran-besaran di atas diperoleh persamaan $q(t)$, yaitu :

$$q(t) = -1.8708 \sin 2t + 2 \sin (1.8708t)$$

Penyelesaian:

Persoalan ini adalah pencarian solusi PDB

$$\begin{aligned} \frac{di}{dt} &= -Ri/L - q/CL + E(t)/L & , i(0) = 0 \\ \frac{dq}{dt} &= i & , q(0) = 0 \end{aligned}$$

dengan $L = 1$ henry, $C = 0.25$ coulomb, $E(t) = E_0 \sin \omega t$, $E_0 = 1$ volt, $\omega = 1.8708$ detik, dan $R = 0$. Diminta menentukan $q(10)$, dalam coulomb, dengan metode Euler dan metode Runge-Kutta orde-4. Solusi dengan kedua metode PDB tersebut diperlihatkan oleh tabel berikut:

Metode Euler	Metode Runge-Kutta Orde-4	Nilai sejati
$q[0.000000000] = 0.0000000000$	$q[0.000000000] = 0.0000000000$	$q[0.000000000] = 0.0000000000$
$q[0.100000000] = 0.0000000000$	$q[0.100000000] = 0.0003113455$	$q[0.100000000] = 0.0003106990$
$q[0.200000000] = 0.0000000000$	$q[0.200000000] = 0.0024585603$	$q[0.200000000] = 0.0024577123$
$q[0.300000000] = 0.0018599064$	$q[0.300000000] = 0.0081397788$	$q[0.300000000] = 0.0081396320$
$q[0.400000000] = 0.0073747206$	$q[0.400000000] = 0.0187855018$	$q[0.400000000] = 0.0187873479$
$q[0.500000000] = 0.0181375023$	$q[0.500000000] = 0.0354437446$	$q[0.500000000] = 0.0354491638$
$q[0.600000000] = 0.0354093809$	$q[0.600000000] = 0.0586868743$	$q[0.600000000] = 0.0586976091$
$q[0.700000000] = 0.0600041247$	$q[0.700000000] = 0.0885456437$	$q[0.700000000] = 0.0885634533$
$q[0.800000000] = 0.0921942750$	$q[0.800000000] = 0.1244745566$	$q[0.800000000] = 0.1245010622$
$q[0.900000000] = 0.1316449743$	$q[0.900000000] = 0.1653511112$	$q[0.900000000] = 0.1653876410$
$q[1.000000000] = 0.1773804195$	$q[1.000000000] = 0.2095097383$	$q[1.000000000] = 0.2095571811$
$q[1.100000000] = 0.2277863746$	$q[1.100000000] = 0.2548094562$	$q[1.100000000] = 0.2548681346$

$q[1.2000000000] = 0.2806504669$	$q[1.2000000000] = 0.2987325039$	$q[1.2000000000] = 0.2988020746$
$q[1.3000000000] = 0.3332401203$	$q[1.3000000000] = 0.3385095552$	$q[1.3000000000] = 0.3385889443$
$q[1.4000000000] = 0.3824160480$	$q[1.4000000000] = 0.3712656561$	$q[1.4000000000] = 0.3713530345$
$q[1.5000000000] = 0.4247773155$	$q[1.5000000000] = 0.3941798264$	$q[1.5000000000] = 0.3942726302$
$q[1.6000000000] = 0.4568321853$	$q[1.6000000000] = 0.4046503946$	$q[1.6000000000] = 0.4047453899$
$q[1.7000000000] = 0.4751873622$	$q[1.7000000000] = 0.4004576274$	$q[1.7000000000] = 0.4005510206$
$q[1.8000000000] = 0.4767469470$	$q[1.8000000000] = 0.3799151173$	$q[1.8000000000] = 0.3800027049$
$q[1.9000000000] = 0.4589114608$	$q[1.9000000000] = 0.3420016950$	$q[1.9000000000] = 0.3420790492$
$q[2.0000000000] = 0.4197667741$	$q[2.0000000000] = 0.2864663543$	$q[2.0000000000] = 0.2865290349$
$q[2.1000000000] = 0.3582527055$	$q[2.1000000000] = 0.2138997625$	$q[2.1000000000] = 0.2139435459$
$q[2.2000000000] = 0.2743014806$	$q[2.2000000000] = 0.1257673604$	$q[2.2000000000] = 0.1257884744$
$q[2.3000000000] = 0.1689371454$	$q[2.3000000000] = 0.0244007565$	$q[2.3000000000] = 0.0243961093$
$q[2.4000000000] = 0.0443284075$	$q[2.4000000000] = -0.0870539817$	$q[2.4000000000] = -0.0870865913$
$q[2.5000000000] = -0.0962108212$	$q[2.5000000000] = -0.2047305158$	$q[2.5000000000] = -0.2047922177$
$q[2.6000000000] = -0.2482767430$	$q[2.6000000000] = -0.3241762399$	$q[2.6000000000] = -0.3242669531$
$q[2.7000000000] = -0.4064879707$	$q[2.7000000000] = -0.4405214587$	$q[2.7000000000] = -0.4406398018$
$q[2.8000000000] = -0.5646532983$	$q[2.8000000000] = -0.5486772875$	$q[2.8000000000] = -0.5488205463$
$q[2.9000000000] = -0.7159907450$	$q[2.9000000000] = -0.6435542527$	$q[2.9000000000] = -0.6437184101$
$q[3.0000000000] = -0.8533910308$	$q[3.0000000000] = -0.7202923292$	$q[3.0000000000] = -0.7204721585$
$q[3.1000000000] = -0.9697161675$	$q[3.1000000000] = -0.7744922968$	$q[3.1000000000] = -0.7746815187$
$q[3.2000000000] = -1.0581216764$	$q[3.2000000000] = -0.8024378810$	$q[3.2000000000] = -0.8026293781$
$q[3.3000000000] = -1.1123891644$	$q[3.3000000000] = -0.8012981891$	$q[3.3000000000] = -0.8014842725$
$q[3.4000000000] = -1.1272547310$	$q[3.4000000000] = -0.7693004777$	$q[3.4000000000] = -0.7694731949$
$q[3.5000000000] = -1.0987179945$	$q[3.5000000000] = -0.7058642733$	$q[3.5000000000] = -0.7060157456$
$q[3.6000000000] = -1.0243164985$	$q[3.6000000000] = -0.6116892854$	$q[3.6000000000] = -0.6118120597$
$q[3.7000000000] = -0.9033509087$	$q[3.7000000000] = -0.4887913460$	$q[3.7000000000] = -0.4888787465$
$q[3.8000000000] = -0.7370477501$	$q[3.8000000000] = -0.3404827158$	$q[3.8000000000] = -0.3405291792$
$q[3.9000000000] = -0.5286484334$	$q[3.9000000000] = -0.1712954258$	$q[3.9000000000] = -0.1712968034$
$q[4.0000000000] = -0.2834159319$	$q[4.0000000000] = 0.0131512245$	$q[4.0000000000] = 0.0131974120$
$q[4.1000000000] = -0.0085536027$	$q[4.1000000000] = 0.2063354087$	$q[4.1000000000] = 0.2064297785$
$q[4.2000000000] = 0.2869658080$	$q[4.2000000000] = 0.4010637627$	$q[4.2000000000] = 0.4012049494$
$q[4.3000000000] = 0.5926591052$	$q[4.3000000000] = 0.5897404236$	$q[4.3000000000] = 0.5899250402$
$q[4.4000000000] = 0.8968737130$	$q[4.4000000000] = 0.7646640123$	$q[4.4000000000] = 0.7648866983$
$q[4.5000000000] = 1.1872011323$	$q[4.5000000000] = 0.9183409247$	$q[4.5000000000] = 0.9185944813$
$q[4.6000000000] = 1.4509493520$	$q[4.6000000000] = 1.0438022210$	$q[4.6000000000] = 1.0440778328$
$q[4.7000000000] = 1.6756574551$	$q[4.7000000000] = 1.1349108485$	$q[4.7000000000] = 1.1351983842$
$q[4.8000000000] = 1.8496328856$	$q[4.8000000000] = 1.1866459239$	$q[4.8000000000] = 1.1869343067$

$q[4.9000000000] = 1.9624897591$	$q[4.9000000000] = 1.1953513582$	$q[4.9000000000] = 1.1956289919$
$q[5.0000000000] = 2.0056653360$	$q[5.0000000000] = 1.1589372228$	$q[5.0000000000] = 1.1591924575$
$q[5.1000000000] = 1.9728914217$	$q[5.1000000000] = 1.0770238867$	$q[5.1000000000] = 1.0772455061$
$q[5.2000000000] = 1.8605980828$	$q[5.2000000000] = 0.9510210585$	$q[5.2000000000] = 0.9511987667$
$q[5.3000000000] = 1.6682286867$	$q[5.3000000000] = 0.7841363494$	$q[5.3000000000] = 0.7842612382$
$q[5.4000000000] = 1.3984478724$	$q[5.4000000000] = 0.5813107578$	$q[5.4000000000] = 0.5813757328$
$q[5.5000000000] = 1.0572275755$	$q[5.5000000000] = 0.3490814358$	$q[5.5000000000] = 0.3490815810$
$q[5.6000000000] = 0.6538005648$	$q[5.6000000000] = 0.0953751228$	$q[5.6000000000] = 0.0953079874$
$q[5.7000000000] = 0.2004759507$	$q[5.7000000000] = -0.1707614026$	$q[5.7000000000] = -0.1708956119$
$q[5.8000000000] = -0.2876833746$	$q[5.8000000000] = -0.4394847308$	$q[5.8000000000] = -0.4396830618$
$q[5.9000000000] = -0.7933156150$	$q[5.9000000000] = -0.7005204937$	$q[5.9000000000] = -0.7007772731$
$q[6.0000000000] = -1.2973356744$	$q[6.0000000000] = -0.9435576587$	$q[5.9999999999] = -0.9438646315$
$q[6.1000000000] = -1.7796142292$	$q[6.1000000000] = -1.1586562582$	$q[6.0999999999] = -1.1590028396$
$q[6.2000000000] = -2.2197377833$	$q[6.2000000000] = -1.3366524173$	$q[6.1999999999] = -1.3370260507$
$q[6.3000000000] = -2.5978226591$	$q[6.3000000000] = -1.4695442264$	$q[6.2999999999] = -1.4699308346$
$q[6.4000000000] = -2.8953522153$	$q[6.4000000000] = -1.5508423588$	$q[6.3999999999] = -1.5512268737$
$q[6.5000000000] = -3.0960040093$	$q[6.5000000000] = -1.5758703518$	$q[6.4999999999] = -1.5762373017$
$q[6.6000000000] = -3.1864323062$	$q[6.6000000000] = -1.5420011309$	$q[6.5999999999] = -1.5423352630$
$q[6.7000000000] = -3.1569713867$	$q[6.7000000000] = -1.4488185980$	$q[6.6999999999] = -1.4491055107$
$q[6.8000000000] = -3.0022265956$	$q[6.8000000000] = -1.2981958585$	$q[6.7999999999] = -1.2984226178$
$q[6.9000000000] = -2.7215230006$	$q[6.9000000000] = -1.0942848179$	$q[6.8999999999] = -1.0944405314$
$q[7.0000000000] = -2.3191858552$	$q[7.0000000000] = -0.8434153189$	$q[6.9999999999] = -0.8434916419$
$q[7.1000000000] = -1.8046326632$	$q[7.1000000000] = -0.5539055852$	$q[7.0999999999] = -0.5538971368$
$q[7.2000000000] = -1.1922633539$	$q[7.2000000000] = -0.2357893358$	$q[7.1999999999] = -0.2356940044$
$q[7.3000000000] = -0.5011426825$	$q[7.3000000000] = 0.0995315992$	$q[7.2999999999] = 0.0997124765$
$q[7.4000000000] = 0.2455228159$	$q[7.4000000000] = 0.4396971489$	$q[7.3999999999] = 0.4399587439$
$q[7.5000000000] = 1.0211024606$	$q[7.5000000000] = 0.7718465789$	$q[7.4999999999] = 0.7721806721$
$q[7.6000000000] = 1.7964342962$	$q[7.6000000000] = 1.0831067579$	$q[7.5999999999] = 1.0835019807$
$q[7.7000000000] = 2.5408657292$	$q[7.7000000000] = 1.3610925048$	$q[7.6999999999] = 1.3615347167$
$q[7.8000000000] = 3.2234070736$	$q[7.8000000000] = 1.5943995153$	$q[7.7999999999] = 1.5948723054$
$q[7.9000000000] = 3.8139568316$	$q[7.9000000000] = 1.7730702468$	$q[7.8999999999] = 1.7735555409$
$q[8.0000000000] = 4.2845525957$	$q[8.0000000000] = 1.8890137961$	$q[7.9999999999] = 1.8894925477$
$q[8.1000000000] = 4.6105981677$	$q[8.1000000000] = 1.9363622321$	$q[8.0999999999] = 1.9368151703$
$q[8.2000000000] = 4.7720160509$	$q[8.2000000000] = 1.9117479983$	$q[8.1999999999] = 1.9121564038$
$q[8.3000000000] = 4.7542750480$	$q[8.3000000000] = 1.8144898094$	$q[8.2999999999] = 1.8148362870$
$q[8.4000000000] = 4.5492453352$	$q[8.4000000000] = 1.6466778292$	$q[8.3999999999] = 1.6469470453$
$q[8.5000000000] = 4.1558380966$	$q[8.5000000000] = 1.4131527092$	$q[8.4999999999] = 1.4133320623$

$q[8.6000000000] = 3.5803934777$	$q[8.6000000000] = 1.1213771469$	$q[8.5999999999] = 1.1214573435$
$q[8.7000000000] = 2.8367890832$	$q[8.7000000000] = 0.7812028329$	$q[8.7000000000] = 0.7811783415$
$q[8.8000000000] = 1.9462512330$	$q[8.8000000000] = 0.4045398276$	$q[8.8000000000] = 0.4044091900$
$q[8.9000000000] = 0.9368623754$	$q[8.9000000000] = 0.0049393809$	$q[8.9000000000] = 0.0047053655$
$q[9.0000000000] = -0.1572299770$	$q[9.0000000000] = -0.4028951852$	$q[9.0000000000] = -0.4032255991$
$q[9.1000000000] = -1.2968851068$	$q[9.1000000000] = -0.8036518155$	$q[9.1000000000] = -0.8040676248$
$q[9.2000000000] = -2.4392919003$	$q[9.2000000000] = -1.1819871163$	$q[9.2000000000] = -1.1824736482$
$q[9.3000000000] = -3.5395012337$	$q[9.3000000000] = -1.5231159385$	$q[9.3000000000] = -1.5236553627$
$q[9.4000000000] = -4.5521161863$	$q[9.4000000000] = -1.8133904192$	$q[9.4000000000] = -1.8139623973$
$q[9.5000000000] = -5.4330795579$	$q[9.5000000000] = -2.0408457557$	$q[9.5000000000] = -2.0414282084$
$q[9.6000000000] = -6.1414914505$	$q[9.6000000000] = -2.1956908998$	$q[9.6000000000] = -2.1962608590$
$q[9.7000000000] = -6.6413853504$	$q[9.7000000000] = -2.2707241281$	$q[9.7000000000] = -2.2712586439$
$q[9.8000000000] = -6.9033895287$	$q[9.8000000000] = -2.2616560433$	$q[9.8000000000] = -2.2621331086$
$q[9.9000000000] = -6.9062018290$	$q[9.9000000000] = -2.1673258748$	$q[9.9000000000] = -2.1677253308$
$q[10.000000000] = -6.6378101261$	$q[10.000000000] = -1.9898008772$	$q[10.000000000] = -1.9901052620$

Perbandingan solusi:

	Euler	Runge-Kutta Orde-4	Sejati
$q(10)$	-6.6378101261	-1.9898008772	-1.9901052620

Untuk menghitung $q(10)$ dengan $h = 0.1$ diperlukan sejumlah

$$n = (10 - 0)/0.1 = 100$$

langkah. Karena itu, dapatlah dimengerti mengapa metode PDB yang berorde rendah seperti metode Euler memperlihatkan hasil yang sangat menyimpang (divergen) dengan solusi sejatinya ketika jumlah langkahnya membesar, sedangkan solusi dengan metode Runge-Kutta memperlihatkan kestabilannya pada setiap langkah (bandingkan dengan solusi sejati pada setiap langkah). Ini disebabkan galat per langkah pada metode Euler semakin menumpuk dengan bertambahnya jumlah langkah. Jadi, metode dengan orde tinggi seperti metode Runge-Kutta orde-4 lebih disukai untuk masalah ini.

Tidak ada alasan bagi kita meremehkan hal-hal kecil, karena bukankah sutera itu berasal dari ulat?
(Anonim)

Soal Latihan

1. Diberikan persamaan diferensial berikut : $dy/dx = -2xy^2$, $y(0) = 1$. Lakukan perhitungan numerik untuk menaksir nilai y pada nilai-nilai x dalam selang $[0,5]$ (ambil ukuran langkah $h = 0.2$) :
 - (a) metode Euler
 - (b) metode Heun
 - (c) metode deret Taylor
 - (d) metode Runge-Kutta orde-3
 - (e) metode Runge-Kutta orde-4
 - (f) metode Adams-Bashforth-Moulton
2. Diberikan persamaan diferensial berikut : $dy/dx = x^2y^2$, $y(1) = 0$. Tentukan nilai (1.4) dengan metode-metode (ambil ukuran langkah $h = 0.2$) :
 - (a) metode Euler
 - (b) metode Heun
 - (c) metode deret Taylor
 - (d) metode Runge-Kutta orde 3
 - (e) metode predictor-corrector Milne
3. Nyatakan dalam sistem persamaan diferensial biasa orde satu :
 - (a) $4y'' + 3xy' + 5y' + xy = 10$; $y(1) = 0$, $y'(1) = 1$
 - (b) $Ak d^2T/dx^2 + P\sigma(T^4 - 273^4) = Q$; $T'(1) = 0$, $T(1) = 0$
 - (c) model matematika rangkaian listrik : $0.5 d^2Q/dt^2 + 6dQ/dt + 50Q = 24 \sin(10t)$ dengan $Q = 0$ dan $i = dQ/dt = 0$ pada $t=0$.
 - (d) $x''(t) - x(t) = 6 \cos(t)$; $x(0) = 2$ dan $x''(0) = 3$
 - (e) $2y'' + (y')^2 + y = 0$, $y(0) = 0$, $y'(0) = 1$
 - (f) $y''' = -y + y'$, $y(0) = 1$, $y'(0) = y''(0) = 0$
4. Perlihatkan bahwa metode Runge-Kutta berikut ini :
$$k_1 = hf(x_r, y_r)$$
$$k_2 = hf(x_r + \alpha h, y_r + \alpha k_1)$$

$$y_{r+1} = y_r + \left[\left(1 - \frac{1}{2\alpha} \right) k_1 + \frac{1}{2\alpha} k_2 \right]$$

adalah berorde dua untuk sembarang tetapan $\alpha (\alpha \neq 0)$.

5. *Ekstrapolasi Richardson* yang telah anda kenal di integrasi numerik dapat juga diterapkan pada solusi PDB, yang bertujuan untuk memperbaiki hasil metode Runge-Kutta orde-4. Jika metode Runge-Kutta orde-4 digunakan dengan ukuran langkah h , maka nilai hampiran $y(x)$ adalah:

$$y(x; h) = y_h + Ch^4 \quad (1)$$

dan jika digunakan ukuran step $2h$, maka nilai hampiran $y(x)$ adalah :

$$y(x) = y(x; 2h) + 16Ch^4 \quad (2)$$

Perlihatkanlah bahwa hampiran $y(x)$ yang lebih baik (*improve*) adalah :

$$y(x) = \frac{1}{15} [16y(x; h) - y(x; 2h)] \quad (3)$$

Kemudian hitunglah y (1.4) menggunakan persamaan (3) di atas bila PDB yang digunakan adalah seperti soal nomor 1 di atas.

6. Masih berkaitan dengan *ekstrapolasi Richardson*. Jika metode Heun digunakan dengan ukuran langkah h , maka nilai hampiran $y(x)$ adalah:

$$y(x) = y_h + Ch^2 \quad (1)$$

dan jika digunakan ukuran step $2h$, maka nilai aproksimasi $y(x)$ adalah :

$$y(x) = y_{2h} + 4Ch^2 \quad (2)$$

Perlihatkanlah bahwa hampiran $y(x)$ yang lebih baik (*improve*) adalah :

$$y(x) = \frac{1}{3} (4y_h - y_{2h}) \quad (3)$$

Kemudian hitunglah y (1.4) menggunakan persamaan (3) di atas bila PDB yang digunakan adalah seperti soal nomor 1 di atas.

7. Dengan menggunakan PDB orde 2 pada soal nomor 3(c) di atas, tentukan muatan listrik Q dan arus I pada saat $t = 0.2$. Metode yang digunakan : Runge-Kutta orde 3 dan ukuran langkah $h = 0.1$.

8. (a) Perlihatkan galat per langkah metode *predictor-corrector* Milne adalah :
galat per langkah *predictor* :

$$y_{r+1} - y^*_{r+1} \approx \frac{28}{90} h^5 y^{(5)}(t)$$

galat per langkah *corrector* :

$$y_{r+1} - y_{r+1}^* \approx \frac{-1}{90} h^5 y^{(5)}(t)$$

- (b) Tentukan orde metode Milne
 (c) Tentukan taksiran y_{r+1} (yaitu nilai yang lebih baik daripada y_{r+1})
 (d) Tentukan galat per langkah untuk y_{r+1}

7. Dengan mengingat defenisi kalukulus untuk turunan adalah

$$y'(x) = \lim_{h \rightarrow 0} \frac{y(x+h) - y(x)}{h}$$

- (a) Turunkan metode Euler dari defenisi turunan tersebut
 (b) Bila nilai y dihitung pada $x+h$ dan $x-h$, turunkan metode baru, yaitu metode titik-tengah
10. Turunkanlah persamaan *predictor* pada metode P-C Adams-Bashforth-Moulton bila titik-titik datanya diinterpolasi dengan polinom Newton-Gregory mundur
11. Bila PDB-nya adalah seperti pada soal nomor 1, tentukan ukuran langkah yang optimal agar galat per langkah pada solusi PDB dengan metode Runge-Kutta orde-4 kurang dari 0.000001.
12. Diberikan PDB $y' = -y$, $y(0) = 1$. Dengan mengambil ukuran langkah $h = 0.1$, periksa kestabilan metode Euler, metode Runge-Kutta orde-3, metode titik-tengah (lihat jawaban soal 7b), dan metode Milne pada penaksiran nilai $y(10)$

Daftar Pustaka

- [BES97] Besari, Mohamad Sahari, *Interaksi Matematika, Sains, dan Rekayasa*, makalah pada seminar Penyempurnaan Pengajaran Matematika Layanan di ITB, 9 - 11 Juni 1997
- [BUC92] Buchanan, James I., *Numerical Methods and Analysis*, McGraw-Hill International Editions, 1992
- [CHA91] Chapra, Steven C dan Canale, Raymond P, *Numerical Methods for Engineers with Personal Computer Applications*, MacGraw-Hill Book Company, 1991
- [CON92] Conte, Samuel D dan De Boor, Carl, *Elementary Numerical Analysis, An Algorithmic Approach, 3rd Edition*, MacGraw-Hills, Inc, 1992.
- [DUL96] Dulimarta, Hansye S., *Diktat Kuliah Pengolahan Citra Digital*, Teknik Informatika ITB, 1996
- [GER85] Gerald, Curtis F dan Wheatley, Patrick O, *Applied Numerical Analysis, 3rd Edition*, Addison-Wesley Publishing Company, 1985
- [KRE88] Kreyszic, Erwin, *Advanced Engineering Mathematics*, John Wiley and Sons, 1988
- [MAT92] Mathews, Johh. H, *Numerical Methods for Mathematics, Science and Engineering, 2nd Edition*, Prentice-Hall International, 1993
- [MAN82] Mano, M. Moris, *Computer System Architecture 2nd*, Prentice-Hall International, 1982
- [NAK93] Nakamura, Shoichiro Nakamura, *Applied Numericak Methods in C*, Prentice-Hall Int. Series, 1993
- [NOB72] Noble, Ben, *A Videotape Course on Elementary Numerical Analysis*, Oberlin College, 1972
- [PUR89] Purcell, Edwin J. dan Verberg, Dale, *Kalkulus dan Geometri Analitis* (terjemahan), Penerbit Erlangga, 1989

Catatan :

Catatan :

Catatan :

Catatan :

