



# Teori dan Praktikum METODE NUMERIK

Dilengkapi Panduan Praktikum  
Lab Komputasi dengan Matlab

Julan Hernadi



Teori dan Praktikum

**BUKTI  
TERBIT**

PENERBIT ERLANGGA

# METODE NUMERIK

Dilengkapi Panduan Praktikum  
Lab Komputasi dengan Matlab

$dx$

**BUKTI  
TERBIT**

PENERBIT ERLANGGA

Julan Hernadi

Teori dan Praktikum

# METODE NUMERIK

Dilengkapi Panduan Praktikum  
Lab Komputasi dengan Matlab



**PENERBIT ERLANGGA**  
Jl. H. Baping Raya No. 100  
Ciracas, Jakarta 13740  
(Anggota IKAPI)

**Julan Hernadi**

007-518-001-0

**TEORI DAN PRAKTIKUM  
METODE NUMERIK**

Hak cipta © 2024 pada Penulis  
Hak terbit pada Penerbit Erlangga

Penulis:  
**Julan Hernadi**

Editor:  
Lemeda Simarmata

Buku ini diset dan dilayout oleh bagian produksi Penerbit Erlangga  
dengan Power MacPro

Setting & Layout: Divisi Perti

Desain Sampul: Yudi Nur Riyadi

Percetakan: PT Gelora Aksara Pratama



27 26 25 24 4 3 2 1

*Dilarang keras mengutip, menjiplak, memfotokopi, atau memperbanyak dalam bentuk apa pun, baik sebagian atau keseluruhan isi buku ini, serta memperjualbelikannya tanpa izin tertulis dari Penerbit Erlangga.*

© HAK CIPTA DILINDUNGI OLEH UNDANG-UNDANG

# Daftar Isi

<b>Prakata</b>	<b>v</b>
<b>1 Pendahuluan Komputasi Saintifik</b>	<b>1</b>
1.1 Mengapa Perlu Metode Aproksimasi?	1
1.2 Komputasi Saintifik dan Strategi Aproksimasi	4
1.3 Sumber Kesalahan Aproksimasi	5
1.4 Galat, Galat Mutlak dan Galat Relatif	7
1.5 Sensitivitas Komputasi	8
1.6 Sistem Bilangan pada Komputer	9
1.6.1 Sistem biner titik mengambang	10
1.6.2 Metode pembulatan dan dampaknya	12
1.7 Order Konvergensi	16
1.8 Praktikum Lab Komputasi	19
1.8.1 Memahami simbol bilangan khusus pada Matlab	19
1.8.2 Galat mutlak dan galat relatif: Studi kasus pada aproksimasi faktorial dengan formula Stirling	20
1.8.3 Aproksimasi bilangan natural $e$	22
1.8.4 Aproksimasi nilai $e$ dengan deret Taylor	23
1.8.5 Aproksimasi limit mendekati nol	25
1.9 Soal-soal Latihan Bab 1	27
<b>2 Aproksimasi Akar Persamaan Taklinear</b>	<b>31</b>
2.1 Pendahuluan	31
2.2 Eksistensi Akar Persamaan Taklinear	33
2.3 Metode Bagidua	36
2.3.1 Estimasi galat metode bagidua	38
2.3.2 Permasalahan kritis pada metode bagidua	40
2.4 Metode Secant dan False-Position	41
2.4.1 Metode secant	41
2.4.2 Permasalahan kritis pada metode secant	43
2.4.3 Metode false-position	44
2.5 Praktikum Lab Komputasi	46
2.5.1 Mendeteksi lokasi akar persamaan taklinear	46
2.5.2 Implementasi numerik metode bagidua	48

2.5.3	Implementasi numerik metode secant . . . . .	51
2.5.4	Implementasi numerik metode false-position . . . . .	54
2.6	Metode Newton dan Iterasi Titik Tetap . . . . .	57
2.6.1	Metode Newton . . . . .	57
2.6.2	Permasalahan kritis pada metode Newton . . . . .	59
2.6.3	Estimasi galat metode Newton . . . . .	60
2.6.4	Metode iterasi titik tetap . . . . .	65
2.6.5	Eksistensi dan ketunggalan titik tetap . . . . .	66
2.6.6	Kekonvergenen iterasi titik tetap . . . . .	69
2.7	Praktikum Lab Komputasi . . . . .	71
2.7.1	Implementasi numerik metode Newton . . . . .	71
2.7.2	Implementasi numerik metode iterasi titik tetap . . . . .	76
2.8	Seputar Fungsi <code>fzero</code> pada Matlab . . . . .	77
2.9	Soal-soal Latihan Bab 2 . . . . .	78
<b>3</b>	<b>Polinomial Interpolasi</b>	<b>85</b>
3.1	Pendahuluan . . . . .	85
3.2	Metode Interpolasi Lagrange . . . . .	87
3.2.1	Polinomial Lagrange . . . . .	87
3.2.2	Interpolasi Lagrange . . . . .	91
3.3	Metode Selisih Terbagi . . . . .	95
3.4	Praktikum Lab Komputasi . . . . .	102
3.4.1	Implementasi metode Lagrange . . . . .	102
3.4.2	Prediksi nilai atau data hilang . . . . .	106
3.4.3	Implementasi numerik metode selisih terbagi . . . . .	107
3.5	Interpolasi Hermite . . . . .	112
3.5.1	Pendefinisian polinomial Hermite melalui selisih terbagi . . . . .	114
3.6	Praktikum Lab Komputasi . . . . .	115
3.7	Seputar Fungsi <code>polyval</code> dan <code>polyfit</code> pada Matlab . . . . .	119
3.8	Polinomial Kuadrat Terkecil (Regresi) . . . . .	122
3.9	Soal-soal Latihan Bab 3 . . . . .	125
<b>4</b>	<b>Interpolasi Potongan Polinomial</b>	<b>129</b>
4.1	Pendahuluan . . . . .	129
4.2	Konstruksi interpolasi potongan polinomial . . . . .	130
4.3	Praktikum Lab Komputasi . . . . .	133
4.4	Spline . . . . .	135
4.4.1	Spline kubik . . . . .	136
4.4.2	B-spline . . . . .	141
4.4.3	Spline kardinal . . . . .	144
4.5	Praktikum Lab Komputasi . . . . .	151
4.5.1	Implementasi numerik interpolasi spline kubik . . . . .	151
4.5.2	Implementasi numerik B-spline . . . . .	155
4.5.3	Implementasi numerik spline kardinal . . . . .	162
4.6	Soal-soal Latihan Bab 4 . . . . .	168

<b>5</b>	<b>Aproksimasi Derivatif dan Integral</b>	<b>173</b>
5.1	Pendahuluan . . . . .	173
5.2	Aproksimasi Derivatif . . . . .	174
5.2.1	Aproksimasi derivatif pertama . . . . .	175
5.2.2	Estimasi galat aproksimasi derivatif . . . . .	177
5.2.3	Aproksimasi order tinggi . . . . .	180
5.2.4	Aproksimasi derivatif kedua . . . . .	183
5.3	Praktikum Lab Komputasi . . . . .	185
5.3.1	Rumus fungsi diketahui . . . . .	185
5.3.2	Derivatif data tabular . . . . .	188
5.3.3	Masalah <i>rounding error</i> pada aproksimasi derivatif . . . . .	190
5.3.4	Sensitivitas aproksimasi derivatif dan <i>smoothing</i> . . . . .	191
5.4	Aproksimasi Integral Tertentu . . . . .	196
5.4.1	Formula kuadratur melalui interpolasi . . . . .	197
5.4.2	Metode koefisien taktentu . . . . .	200
5.4.3	Estimasi galat metode kuadratur dasar . . . . .	202
5.4.4	Formula kuadratur bersusun . . . . .	206
5.4.5	Metode integrasi Gauss . . . . .	217
5.5	Praktikum Lab Komputasi . . . . .	222
5.5.1	Formula kuadratur dasar . . . . .	222
5.5.2	Formula kuadratur bersusun . . . . .	223
5.5.3	Integral data tabular . . . . .	226
5.5.4	Implementasi integrasi Gauss . . . . .	227
5.6	Soal-soal Latihan Bab 5 . . . . .	230
<b>6</b>	<b>Sistem Persamaan Linear</b>	<b>233</b>
6.1	Pendahuluan . . . . .	233
6.2	Bentuk-bentuk Sistem Persamaan Linear . . . . .	236
6.3	Metode Eliminasi Gaussian . . . . .	238
6.3.1	Algoritma metode eliminasi Gaussian . . . . .	239
6.4	Strategi Pivoting . . . . .	243
6.4.1	Pivoting parsial . . . . .	243
6.4.2	Pivoting skala parsial . . . . .	246
6.4.3	Pivoting total . . . . .	250
6.5	Kompleksitas Penyelesaian Sistem Persamaan Linear . . . . .	255
6.5.1	Flops pada perkalian matriks . . . . .	255
6.5.2	Flops pada metode eliminasi Gaussian . . . . .	258
6.6	Praktikum Lab Komputasi . . . . .	260
6.6.1	Substitusi mundur dan maju . . . . .	260
6.6.2	Membentuk sistem persamaan linear segitiga . . . . .	262
6.6.3	Implementasi eliminasi Gaussian . . . . .	263
6.6.4	Implementasi strategi pivoting parsial . . . . .	264
6.6.5	Implementasi strategi pivoting skala parsial . . . . .	266
6.6.6	Implementasi strategi pivoting total . . . . .	268

6.6.7	Toolbox Matlab untuk sistem persamaan linear . . . . .	270
6.7	Metode Dekomposisi LU . . . . .	273
6.8	Praktikum Lab Komputasi . . . . .	277
6.8.1	Mendefinisikan matriks eliminasi elementer . . . . .	277
6.8.2	Implementasi metode LU tanpa pivoting . . . . .	278
6.8.3	Implementasi metode LU dengan pivoting . . . . .	281
6.9	Sensitivitas Sistem Persamaan Linear . . . . .	283
6.9.1	Norma vektor dan matriks . . . . .	284
6.9.2	Bilangan kondisi sebagai ukuran sensitivitas . . . . .	285
6.10	Soal-soal Latihan Bab 6 . . . . .	288
<b>7</b>	<b>Aproksimasi Akar Sistem Persamaan Taklinear</b>	<b>293</b>
7.1	Pendahuluan . . . . .	293
7.2	Metode Newton . . . . .	295
7.3	Metode Chord . . . . .	299
7.4	Metode Broyden . . . . .	302
7.5	Metode Penurunan Tercuram . . . . .	305
7.6	Praktikum Lab Komputasi . . . . .	308
7.6.1	Implementasi metode Newton . . . . .	308
7.6.2	Implementasi numerik metode chord dan modifikasinya . . . . .	314
7.6.3	Implementasi numerik metode Broyden . . . . .	317
7.6.4	Implementasi metode penurunan tercuram . . . . .	318
7.7	Analisis penyelesaian dan galat aproksimasi . . . . .	320
7.8	Soal-soal Latihan Bab 7 . . . . .	330
	<b>Daftar Pustaka</b>	<b>335</b>
	<b>Apendiks A Pengenalan Matlab</b>	<b>337</b>
	<b>INDEKS</b>	<b>369</b>

# Prakata

*Alhamdulillah wasy-syukru lillaah* atas kekuatan yang diberikan Allah SWT kepada penulis, akhirnya naskah ini dapat terbit dan hadir di tengah para pembaca sekalian. Semoga buku ini dapat memberikan pencerahan kepada mahasiswa dan pembaca yang ingin mempelajari metode numerik secara lebih komprehensif.

Selama ini sebagaian besar mahasiswa mempelajari metode numerik hanya mengikuti langkah-langkah prosedural seperti diberikan pada setiap metode yang ada. Bahkan sebagian mahasiswa tidak mau memahami bagaimana sebuah metode dikembangkan namun langsung menggunakan alat siap pakai berupa *toolbox* yang telah disediakan oleh masing-masing *software* komputasi. Penggunaan *toolbox* instant tidak memberikan dampak positif pada pengalaman belajar mahasiswa. Buku ini menyediakan materi pembelajaran metode numerik secara komprehensif baik dari aspek teoretis maupun aspek implementasi numeriknya pada komputer. Melalui buku ini diharapkan kemampuan kognitif (*knowledge and understanding*) dan keterampilan (*skills*) dapat dicapai dengan baik. Oleh karenanya buku ini dapat juga dijadikan sebagai landasan teoretis (referensi) penelitian bidang teknik dan sains terapan yang melibatkan metode numerik dalam kajian dan analisis datanya.

Buku ini memuat topik-topik penting dan mendasar pada metode numerik dengan pembahasan cukup komprehensif. Buku ini terdiri atas 7 bab bahan kajian, yaitu Pendahuluan Komputasi Sainifik, Aproksimasi Akar Persamaan Taklinear, Polinomial Interpolasi, Aproksimasi Derivatif dan Integral, Sistem Persamaan Linear, dan Aproksimasi Akar Sistem Persamaan Taklinear. Untuk implementasi numerik setiap metode pada buku ini, digunakan Matlab yang telah dilengkapi oleh sintaksis (m-file) sederhana, mudah dipahami, dan dapat dimodifikasi sesuai kebutuhan. Buku ini menawarkan model pembelajaran teori dan praktikum. Belajar teori dimulai dari motivasi dan ide pengembangan metode, analisis galat dalam bentuk teorema, penyusunan formula aproksimasi, penyusunan algoritma, dan perhitungan manual skala kecil dengan kalkulator. Untuk implementasi numerik skala besar dilakukan dengan menggunakan Matlab dalam kegiatan praktikum lab komputasi. Tentunya akan sangat bagus sekali jika mahasiswa tertarik mempelajari metode numerik, mulai dari konsep yang melatarbelakangi setiap metode sampai dengan implementasi numerik pada komputer.

Buku ini disiapkan untuk mata kuliah metode numerik pada prodi matematika, pendidikan matematika, prodi eksakta lainnya pada bidang sains (statistika, komputer, fisika, kimia, farmasi) dan prodi-prodi pada ilmu keteknikan (elektro, informatika, mesin, dan lain-lain). Idealnya buku ini dipelajari secara terurut agar aspek teori dan aspek komputasinya dapat dipahami dengan baik. Akan ideal lagi jika pembelajaran metode numerik diikuti dengan praktikum lab komputasi. Mahasiswa dengan latar belakang matematika tidak begitu kuat dapat saja melewatkan kajian analisis galat pada berbagai teorema yang sarat analisis real, namun diharapkan tetap memahami kelebihan dan kekurangan metode-metode yang ada. Bahkan mahasiswa yang tidak tertarik dengan uraian matematis dapat langsung mempelajari

cara implementasi numerik termasuk dapat menggunakan toolbox relevan yang diberikan pada akhir kegiatan praktikum. Namun demikian tetap disarankan agar mahasiswa dapat mengikuti pembahasan pada buku secara runtut agar mendapatkan pengalaman belajar yang bermakna dan berkesan. Ketekunan dan kesabaran adalah dua kata kunci untuk memahami isi buku ini. Percayalah, tidak ada sesuatu yang sulit jika dipelajari dengan tekun dan sabar.

Sebagai sumbangsih untuk negeri khususnya kepada mahasiswa harapan bangsa, beberapa buku teks telah penulis hadirkan sebelumnya adalah “Matematika Numerik dengan Implementasi MATLAB” (Andi Offset: Yogyakarta, 2012), “Teori dan Komputasi Numerik Diferensial dan Integral” (Graha Ilmu: Yogyakarta, 2015), “Analisis Real Elementer dengan Ilustrasi Grafis dan Numeris” (Erlangga: Jakarta, 2015), “Fondasi Matematika: Dasar Berpikir Logis dan Kritis” (UMP Press: Ponorogo, 2017), “SKR Kalkulus 1” (Erlangga: Jakarta, 2021), “[E-book] Fondasi Matematika dan Metode Pembuktian” (Erlangga: Jakarta, 2021). Mudah-mudahan kehadiran buku-buku ini dapat memberikan kontribusi dalam mempersiapkan generasi bangsa yang lebih baik.

Pada kesempatan ini penulis menyampaikan terima kasih dan penghargaan kepada istri tercinta Sri Purnama Surya dan ananda kebanggaan Ahmad Zhafir Hernadi atas dorongan semangat yang diberikan selama ini. Kepada adik-adik: Herri Gusmedi, Aprillina dan Yuniarni, penulis menyampaikan penghargaan dan terima kasih atas doa dan harapan kalian selama ini. Terkhusus untuk ayahanda tercinta Ahmad Ismail dan ibunda tercinta Konaria binti Muhammad yang keduanya telah kembali menghadap Sang Khalik, penulis senantiasa berdoa semoga kalian berdua dipertemukan kembali oleh Allah SWT di tempat yang terbaik. *Allahummaghfirlii waliwaalidayya, warham humaa kamaa rabbayaa nii shaghiiraa*. “Wahai Tuhanku, ampunilah aku dan kedua orang tuaku, sayangilah mereka seperti mereka menyayangiku di waktu kecil”.

Penulis mengucapkan terima kasih kepada pihak yang telah membantu penulis dalam menyiapkan naskah ini. Secara khusus penulis mengucapkan terima kasih kepada Mas Ceriawan Hadi Santoso yang telah memberikan berbagai koreksi terhadap beberapa kesalahan-kesalahan yang ditemukan pada naskah ini. Penulis menyadari kemungkinan masih adanya beberapa ketidaksempurnaan isi buku ini dikarenakan keterbatasan penulis. Oleh karena itu, segala saran dan kritik membangun demi perbaikan buku pada penerbitan berikutnya akan diterima dengan senang hati. Akhirnya, semoga karya kecil ini dapat memberikan manfaat bagi orang banyak dan menjadi pemicu lahirnya karya-karya yang lebih besar.

Yogyakarta, September 2022

Julan Hernadi

# BAB 1

## Pendahuluan Komputasi Saintifik

*Dialah yang menjadikan matahari bersinar dan bulan bercahaya dan ditetapkan-Nya manzilah-manzilah (tempat-tempat) perjalanan bulan itu, supaya kamu mengetahui bilangan tahun dan perhitungan (waktu).*

QS 10:5.

### 1.1 Mengapa Perlu Metode Aproksimasi?

Pemodelan matematika merupakan suatu proses di mana permasalahan pada dunia nyata disajikan dalam bentuk permasalahan matematika, seperti sistem persamaan linear, persamaan taklinear, persamaan diferensial yang memuat masalah nilai awal dan syarat batas, persamaan integral, masalah optimasi dan kontrol, dan lain sebagainya. Jadi, model matematika menggambarkan sistem dunia nyata dalam bahasa matematika. Permasalahan matematika tersebut perlu ditentukan penyelesaian atau solusinya yang nantinya digunakan untuk menyelesaikan permasalahan nyata yang terkait.

Sebelum dilakukan upaya penentuan penyelesaian yang dimaksud biasanya diselidiki terlebih dulu sifat *well-posedness*-nya, yaitu analisis kewujudan termasuk ketunggalan penyelesaian sebuah permasalahan matematika. Setelah secara teoritis permasalahan matematika ini terjamin mempunyai penyelesaian baru dilakukan upaya mencari penyelesaiannya. Cara pertama yang biasa digunakan adalah cara analitik, yaitu penyelesaian dapat disajikan dalam bentuk formula atau rumus eksplisit yang memuat variabel-variabel terkait. Penyelesaian yang diperoleh dengan cara analitik ini umumnya eksak (*exact solution*). Namun cara analitik ini sangat jarang dapat dilakukan pada bidang matematika terapan dikarenakan rumitnya permasalahan yang dihadapi. Cara kedua adalah dengan metode aproksimasi, yaitu mencari penyelesaian pendekatannya (*approximation solution*) melalui metode numerik tertentu. Penyelesaian ini biasa juga disebut penyelesaian numerik.

Penyelesaian numerik diperoleh dengan memanfaatkan data yang diketahui, biasanya diperoleh dari hasil pengukuran lapangan. Oleh karena adanya keterbatasan data baik secara kuantitas maupun secara kualitas maka penyelesaian yang diperoleh hanyalah sebagai aproksimasi. Konsekuensinya kesalahan aproksimasi tidak dapat dihindari. Kesalahan lain yang tidak dapat dihindari adalah ketika mengembangkan metode numerik tertentu. Banyak persyaratan teoretis tidak dipenuhi namun diasumsikan memenuhi, misalnya sebuah fungsi disyaratkan kontinu atau terdiferensial namun hanya tersedia dalam bentuk data numerik tanpa adanya formula eksplisitnya. Ketika melakukan proses komputasi



Gambar 1.1: Pembuatan seng gelombang dari plat rata

pada komputer maka tidak bisa dihindari dari kemungkinan terjadi kesalahan komputasi (*computational error*). Keterbatasan komputer dalam menyajikan bilangan real ditambah tingginya kompleksitas komputasi yang dihadapi mengakibatkan hasil (*output*) yang diberikan komputer telah mengalami gangguan akurasi dikarenakan sistem pembulatan komputer.

Walaupun banyak kelemahan yang ditemukan pada metode aproksimasi ini namun inilah satu-satunya jalan untuk menyelesaikan sebuah permasalahan matematika yang mana penyelesaian eksak tidak dapat diperoleh. Untuk memperoleh gambaran bahwa metode aproksimasi sangat penting khususnya pada bidang matematika terapan, kita perhatikan beberapa ilustrasi dalam contoh-contoh berikut.

**Contoh 1.1.** Sebuah atap berbentuk seng gelombang terbuat dari plat tipis rata yang dipres seperti terlihat pada Gambar 1.1. Diinginkan membuat seng gelombang yang pajangnya 120 cm, tinggi gelombang dari garis pusat 1 cm, dan setiap gelombang mempunyai periode  $4\pi$  cm, sekitar 15.5 cm. Permasalahannya adalah bagaimana menentukan panjang bahan dasar plat. Permasalahan ini diterjemahkan ke dalam matematika, yaitu bagaimana menghitung panjang busur kurva  $y = f(x) = \sin \frac{x}{2}$  dari  $x = 0$  sampai dengan  $x = 120$ . Berdasarkan pelajaran kalkulus kita harus menyelesaikan integral berikut

$$L = \int_0^{120} \sqrt{1 + (f'(x))^2} dx = \int_0^{120} \sqrt{1 + \frac{1}{4} \cos^2 \frac{x}{2}} dx.$$

Bentuk ini merupakan integral eliptik jenis kedua. Perhitungan integral ini tidak sesederhana seperti pada kalkulus dengan menggunakan teknik integrasi biasa. Oleh karena itu perlu diselesaikan secara aproksimasi dengan metode numerik.

Contoh berikut adalah permasalahan sederhana yang sudah diperkenalkan kepada siswa sekolah menengah pertama SMP atau SMA.

**Contoh 1.2.** Kita ingin membuat sebuah kerucut yang tingginya  $t$  cm terbuat dari bahan tertentu. Kerucut yang terbentuk dibalik, kemudian diisi air sampai penuh. Selanjutnya ke dalamnya dimasukkan sebuah bola padat sampai tenggelam, sehingga sebagian airnya tumpah. Jika diinginkan panjang jari-jari bola setengah jari-jari alas kerucut, berapa jari-jari kerucut ini agar air yang tersisa adalah  $v$  cm<sup>3</sup>. Ilustrasi diberikan pada Gambar 1.2. Permasalahan ini dapat diformulasikan secara sederhana seperti berikut. Misalkan jari-jari kerucut  $r$  cm maka jari-jari bola adalah  $\frac{r}{2}$  cm. Diperoleh:

$$\begin{aligned} V_{kerucut} - V_{bola} &= V_{sisa} \\ \frac{1}{3} (\pi r^2 t) - \frac{4}{3} \pi \left(\frac{r}{2}\right)^3 &= v \\ -\frac{4}{24} \pi r^3 + \frac{1}{3} \pi r^2 t &= v \\ -\frac{1}{6} \pi r^3 + \frac{1}{3} \pi r^2 t - v &= 0. \end{aligned}$$

3. Implementasi algoritma tersebut pada perangkat lunak komputer.
4. Simulasi numerik fenomena fisis tersebut dengan menggunakan perangkat lunak komputer.
5. Representasi hasil terhitung (keluaran komputer) dalam bentuk yang mudah dipahami, biasanya melalui visualisasi grafik.
6. Interpretasi dan validasi hasil terhitung komputer dikaitkan dengan model matematika semula. Pada tahap ini ada kemungkinan diperlukan koreksi atau penyempurnaan terhadap hasil yang diperoleh, bahkan jika perlu dilakukan pengulangan tahapan dari awal.

Dari enam tahapan ini, Heath memandang komputasi saintifik berkenaan dengan tahap 2–4, yaitu pengembangan, implementasi, dan penggunaan algoritma numerik dan perangkat lunak untuk menyelesaikan permasalahan dunia nyata.

Permasalahan dunia nyata yang dihadapi biasanya sangat rumit, oleh karena itu diperlukan strategi dalam melakukan aproksimasi. Lebih lanjut, Heath memberikan strategi umum yang digunakan dalam menentukan penyelesaian suatu model matematika, yaitu dengan cara mentransformasi permasalahan yang awal rumit menjadi permasalahan baru yang lebih sederhana yang penyelesaiannya sama, atau paling tidak cukup mendekati. Secara teknis, pendekatan yang sangat sering digunakan antara lain:

1. Mengganti proses takhingga menjadi proses berhingga, misalnya mengganti integral atau deret takberhingga dengan deret berhingga, mengganti derivatif dengan formula beda hingga.
2. Mengganti fungsi yang rumit dengan fungsi sederhana seperti polinomial atau fungsi trigonometri dasar.
3. Mengganti masalah taklinear dengan masalah linear, misalnya dengan mengambil polinomial Taylor derajat satu.
4. Mengganti persamaan diferensial dengan persamaan aljabar biasa, misalnya dengan menggunakan transformasi Laplace.
5. Mengganti ruang berdimensi takhingga dengan ruang berdimensi hingga, misalnya dengan menggunakan basis berhingga subruang aproksimasinya.

Idealnya, penyelesaian permasalahan baru hasil transformasi identik dengan penyelesaian permasalahan asli tapi keadaan ini sangat jarang terjadi. Sebagai apapun algoritma numerik yang digunakan, hasilnya masih tetap sebagai aproksimasi. Akurasi nilai aproksimasi merupakan isu sentral dalam metode aproksimasi. Keakuratan aproksimasi dan jaminan kekonvergenan barisan aproksimasi ke penyelesaian eksak adalah dua hal penting dalam metode aproksimasi. Jadi, selain bertujuan mendapatkan nilai aproksimasi, juga perlu adanya estimasi terukur terhadap kesalahan yang ditimbulkan oleh aproksimasi tersebut. Walaupun kesalahan sesungguhnya (*true error*) tidak dapat diketahui pasti karena nilai eksak tidak diketahui dengan pasti, namun batas atasnya dapat diestimasi melalui formula yang disebut formula estimasi kesalahan (*error estimation*).

### 1.3 Sumber Kesalahan Aproksimasi

Kesalahan aproksimasi dikategorikan menjadi berupa kesalahan bawaan terjadi sebelum proses komputasi (*inherent error*) dan kesalahan selama proses komputasi (*computational error*).

**Contoh 1.4.** Luas permukaan bumi biasanya dihitung menggunakan formula

$$A = 4\pi r^2$$

dengan  $r$  menyatakan jari-jari bumi. Penggunaan rumus ini adalah sebagai suatu proses aproksimasi yang tentunya akan menghasilkan kesalahan, antara lain:

- bumi dimodelkan sebagai bola, padahal sesungguhnya tidak persis demikian. Ini hanya idealisasi terbaik bentuk bumi dikaitkan dengan bentuk geometri ruang. Bayangkan kalau bumi dimodelkan sebagai kerucut atau bahkan bidang datar yang akhir-akhir ini menjadi viral pada media sosial.
- radius bumi  $r \approx 6370$  km adalah hasil perhitungan atau pengukuran terdahulu, tentunya sebagai nilai pendekatan.
- nilai  $\pi$  diberikan sebagai hasil pemotongan deret takhingga hanya sampai suku tertentu. Bilangan 3.14159265358979 sebagai pengganti nilai  $\pi$  hanyalah akurat dalam 15 digit. Bila dikalikan dengan bilangan besar, misalnya bilangan dengan 20 digit maka kesalahannya menjadi sangat signifikan.
- Nilai numerik dan operasi aritmatika yang dilakukan pada komputer akan mengalami pembulatan yang dilakukan oleh sistem komputer secara otomatis.

## 1.4 Galat, Galat Mutlak dan Galat Relatif

Salah satu kata kunci dalam metode aproksimasi adalah *error* yang digunakan untuk menyatakan ketidakakuratan dan penyimpangan nilai terhitung atau terukur terhadap nilai eksak yang akan diukur. Beberapa istilah dalam bahasa Indonesia untuk *error* antara lain: kesalahan, kekeliruan, atau galat. Pada pembahasan selanjutnya, buku ini akan lebih sering menggunakan istilah galat sebagai terjemahan kata *error*.

**Definisi 1.1.** Misalkan  $p$  adalah nilai eksak yang akan diaproksimasi dan  $p^*$  adalah nilai yang digunakan untuk mengaproksimasi nilai ini maka terdapat tiga ukuran yang digunakan untuk mengukur kualitas aproksimasi ini, yaitu:

1. galat =  $E := p - p^*$ , yaitu selisih nilai eksak dari aproksimasinya.
2. galat mutlak =  $E_M := |p - p^*|$ , yaitu selisih mutlak antara eksak dan aproksimasinya.
3. galat relatif =  $E_R := \frac{|p - p^*|}{|p|} = \frac{\text{galat}}{\text{eksak}}$ , perbandingan galat mutlak nilai yang diaproksimasi. Khusus galat relatif hanya terdefinisi bilamana  $p \neq 0$ . Galat relatif ini kadangkala menggunakan notasi % (persentase), yaitu  $E_R := \frac{|p - p^*|}{|p|} \times 100\%$ .

Bila tanda nilai mutlak definisi (3) diabaikan, diperoleh penjabaran sebagai berikut:

$$\begin{aligned} \text{galat relatif} &= \frac{\text{eksak-aproksimasi}}{\text{eksak}} \\ (\text{galat relatif}) \times (\text{eksak}) &= \text{eksak-aproksimasi} \\ \text{aproksimasi} &= \text{eksak} \times (1 - \text{galat relatif}). \end{aligned}$$

Dalam banyak hal, galat relatif lebih informatif daripada galat mutlak seperti ditunjukkan pada contoh berikut.

- $s$  terdiri dari satu bit 0 atau 1 menyatakan tanda bilangan, positif atau negatif,
- $c$  diperoleh dari 11-bit berikutnya, yang menunjukkan eksponen. Dalam hal ini, basis untuk eksponen adalah 2,
- $f$  diperoleh dari 52-bit terakhir yang merupakan bagian pecahan biner atau **mantisa**.

Dengan format ini, nilai  $c$  berjalan pada interval  $0 < c < 2047$  dan eksponen yang dapat dijangkau berada dalam interval  $(-1023, 1024)$ .

**Contoh 1.8.** Misalkan suatu bilangan dalam sistem 64-bit disajikan sebagai berikut:

0 1000000011 101010010001000

Tentukan bilangan real apa yang sedang disajikan oleh notasi ini.

**PENYELESAIAN.** Bit paling kiri adalah  $s = 0$ . Sebelas bit berikutnya 1000000011 memberikan eksponen, yaitu ekuivalen dengan bilangan desimal

$$c = 1 \times 2^{10} + 0 \times 10^9 + \dots + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 = 1027$$

Selanjutnya, 52-bit terakhir sebagai mantisa berarti

$$f = 1 \times \left(\frac{1}{2}\right)^1 + 1 \times \left(\frac{1}{2}\right)^3 + 1 \times \left(\frac{1}{2}\right)^5 + 1 \times \left(\frac{1}{2}\right)^8 + 1 \times \left(\frac{1}{2}\right)^{12}$$

Jadi bilangan mesin ini sesungguhnya menyajikan bilangan desimal berikut, yaitu

$$\begin{aligned} (-1)^s \times 2^{c-1023} \times (1+f) &= (-1)^0 \times 2^{1027-1023} (1+f) \\ &= 27.56640625 \end{aligned}$$

■

Sebaliknya, contoh berikut menerjemahkan bilangan desimal sehari-hari ke dalam bentuk representasi 64-bit.

**Contoh 1.9.** Sajikan bilangan 32.75 dalam sistem 64-bit (biner).

**PENYELESAIAN.** Pertama bilangan 32.75 disajikan dalam bentuk basis 2, yaitu

$$\begin{aligned} 32.75 &= (-1)^0 \times 2^5 \times \left(1 + \frac{0.75}{2^5}\right) \\ &= (-1)^0 \times 2^5 \times \left(1 + \left(\frac{(1/2) + (1/4)}{2^5}\right)\right) \\ &= (-1)^0 \times 2^5 \times \left(1 + \left(\frac{1}{2}\right)^6 + \left(\frac{1}{2}\right)^7\right) \end{aligned}$$

Secara aljabar kedua formula ekuivalen, namun dalam ketahanan terhadap resiko pembulatan berbeda. Untuk melihat formula mana yang lebih stabil terhadap pembulatan, gunakan data  $x = [10000, 10001, 10002]$ , lakukan perhitungan dalam titik mengambang dengan presisi  $u = 6 \times 10^{-4}$ . Kesimpulan apa yang Anda peroleh dalam eksperimen ini?

16. Untuk fungsi yang sengaja didefinisikan sebagai berikut, berikan alasan mengapa ia sangat rentan terhadap pembulatan untuk  $x$  di sekitar nilai tertentu. Temukan nilai  $x$  yang dimaksud dan perkuat argumen Anda dengan hasil simulasi numerik.

(a)  $f(x) = \frac{\sqrt{x+9}-3}{x}$ .

(b)  $f(x) = \frac{1-\cos x}{x}$ .

(c)  $f(x) = \frac{\ln x - \sin \pi x}{1-x}$ .

(d)  $f(x) = \frac{e^{1+x} - e^{1-x}}{2x}$ .

17. Perhatikan ketiga barisan  $(\sin \frac{1}{n})$ ,  $(\sin \frac{1}{n^2})$ , dan  $(\sin \frac{1}{n})^2$  semua konvergen ke nol untuk  $n \rightarrow \infty$ . Tentukan barisan mana yang kekonvergenan paling cepat. Tentukan order konvergensi masing-masing barisan. Petunjuk: gunakan fakta bahwa  $\sin x \leq x$  untuk  $x \geq 0$  sehingga berlaku  $\sin \frac{1}{n} < \frac{1}{n}$ . Akan sangat membantu jika diamati melalui grafik (plot) fungsi-fungsi tersebut.

18. Tentukan limit barisan  $(\frac{\sin h - h \cos h}{h})$  dan  $(\frac{1-e^h}{h})$  untuk  $h \rightarrow 0$ , kemudian tentukan order konvergensi menuju limit masing-masing. Petunjuk: ambil  $h := \frac{1}{n}$  maka berlaku  $h \rightarrow 0$  bila hanya bila  $n \rightarrow \infty$ . Bila sulit menemukan bentuk  $|\alpha - \alpha_n| \leq \frac{K}{n^p}$  sesuai definisi order konvergensi, Anda dapat membandingkan grafik fungsi  $y = \frac{\sin \frac{1}{n} - \frac{1}{n} \cos \frac{1}{n}}{1/n}$  dan fungsi  $y = \frac{K}{n^p}$  untuk  $p = 1, 2, 3, 4$  dan  $K > 0$  yang memenuhi.

19. Pada contoh soal sebelumnya, kita mengaproksimasi luas permukaan bumi dengan rumus  $A = 4\pi r^2$  di mana jari-jari bumi  $r = 6370$  km.

(a) Hitunglah luas permukaan bumi dalam pembulatan 4 digit signifikan. Dalam hal ini gunakan 3.1416 untuk nilai  $\pi$ .

(b) Seandainya jari-jari bumi bertambah 1 km, berapakah penambahan luas permukaan bumi? Tetap gunakan 4 digit signifikan.

(c) Dari rumus  $A = 4\pi r^2$  diperoleh derivatif  $\frac{dA}{dr} = 8\pi r$ , ini berarti setiap terjadi perubahan pada jari-jari  $r$  sebesar  $h$  maka akan terjadi perubahan luas bumi sebesar  $(8\pi r)h$ . Hitunglah perubahan luas bumi untuk perubahan 1 km pada jari-jarinya. Tetap gunakan 4 digit signifikan. Bagaimana nilai yang diperoleh melalui derivatif dibandingkan dengan menggunakan formula eksak?

(d) Selanjutnya lakukan perhitungan yang sama, kali ini dengan presisi yang lebih tinggi misalnya 9 digit signifikan. Dalam hal ini gunakan nilai 3.14159265 untuk nilai  $\pi$ .

(e) Berikan ulasan terhadap hasil perhitungan yang Anda peroleh pada (a)-(d).

20. Andapun idealisasi bumi sebagai bola dapat dimaklumi, maka menghitung jari-jari bumi bukanlah pekerjaan yang mudah dan tidak mungkin diukur langsung. Cari informasi melalui referensi atau internet, bagaimana orang mengukur jari-jari bumi? Kesalahan apa yang mungkin tidak dapat dihindari dalam proses pengukuran tersebut.

21. Kita telah melakukan eksperimen numerik bahwa nilai  $e$  diaproksimasi dengan menggunakan deret Taylor

$$e^x = \lim_{n \rightarrow \infty} \left( 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \cdots + \frac{x^n}{n!} \right)$$

dengan mengambil  $x = 1$ . Lakukan eksperimen numerik untuk menghitung nilai  $e^{-1}$  dengan mengambil  $x = -1$  dan  $n = 1, 2, \dots, 17$ . Bila diberikan nilai eksak 10 digit signifikan adalah 0.3678794414, buatlah tabel aproksimasi beserta kesalahan relatif untuk tiap-tiap  $n$ .

22. Barisan Fibonacci didefinisikan secara rekursif sebagai  $F_1 := 1, F_2 := 1$  dan  $F_{n+1} := F_n + F_{n-1}$  untuk  $n = 2, 3, 4, \dots$ . Di lain pihak suku-suku barisan Fibonacci ini dapat disajikan secara eksplisit berdasarkan rumus  $F_n = \frac{1}{\sqrt{5}} \left[ \left( \frac{1+\sqrt{5}}{2} \right)^n - \left( \frac{1-\sqrt{5}}{2} \right)^n \right]$  untuk  $n = 1, 2, 3, \dots$ . Walaupun rumus ini eksak namun dalam implementasi numeriknya akan terjadi pembulatan sehingga hasil yang diperoleh tidak bulat padahal suku-suku barisan Fibonacci ini merupakan bilangan bulat. Lakukan eksperimen numerik dengan Matlab untuk melihat galat mutlak yang dihasilkan oleh formula eksplisit ini dengan nilai eksak merujuk kepada formula rekursif.

## BAB 2

# Aproksimasi Akar Persamaan Taklinear

*Numbers are intellectual witnesses that belong only to mankind.*

Honore De BALZAC

## 2.1 Pendahuluan

Persamaan linear satu variabel sudah diperkenalkan sejak di bangku SMP, yaitu persamaan yang berbentuk

$$ax + b = c \tag{2.1.1}$$

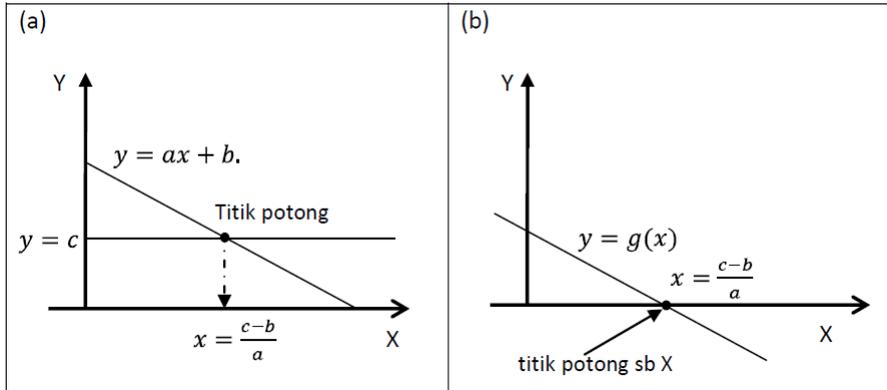
dengan  $a \neq 0$ ,  $b$  dan  $c$  sebagai konstanta yang diberikan, dan  $x$  sebagai variabel yang tidak diketahui. Nilai  $x$  yang memenuhi persamaan ini disebut **penyelesaian**, **solusi** atau **akar** persamaan. Penyelesaian persamaan linear sangat mudah diperoleh, yaitu diberikan oleh

$$x = \frac{c - b}{a}.$$

Persamaan (2.1.1) dikatakan linear karena fungsi yang membentuknya berupa fungsi linear, yaitu  $f(x) = ax + b$  dengan grafik berbentuk garis lurus. Secara geometri, penyelesaian persamaan linear ini adalah absis titik potong antara grafik  $y = ax + b$  dan garis  $y = c$ , seperti ditunjukkan pada Gambar 2.1(a).

Persamaan linear di atas dapat ditulis dalam bentuk umum sebagai  $f(x) = c$  dengan  $f$  sebuah fungsi linear. Selain itu dapat pula disajikan secara implisit  $f(x) - c = 0$  atau disingkat  $g(x) = 0$  dengan  $g(x) := f(x) - c$ . Secara geometris penyelesaian persamaan linear  $g(x) = 0$  adalah absis titik potong grafik  $y = g(x)$  dengan sumbu  $X$ , yaitu nilai  $x$  di mana  $g(x) = 0$ , seperti ditunjukkan pada Gambar 2.1(b). Oleh karena itu, penyelesaian persamaan  $g(x) = 0$  kadang-kadang disebut pembuat nol fungsi  $g$ .

Oleh karena begitu sederhananya masalah mencari penyelesaian persamaan linear maka ia tidak dijadikan topik khusus dalam metode numerik. Permasalahan ini akan menjadi rumit ketika beberapa persamaan linear dengan beberapa variabel terdefinisi secara simultan. Namun demikian, persamaan linear satu variabel sering digunakan dalam memecahkan masalah dalam kehidupan sehari-hari seperti



Gambar 2.1: Ilustrasi geometris akar persamaan linear

diilustrasikan pada contoh berikut.

**Contoh 2.1.** Misalkan ongkos naik taksi diberlakukan dengan sistem biaya buka pintu Rp 10.000,00 dan biaya jarak tempuh dengan tarif Rp 5.000,00 setiap kilomernya. Bila seseorang naik taksi menghabiskan Rp 50.000,00 berapa kilometer jarak yang ditempuh orang tersebut.

**PENYELESAIAN.** Masalah ini diselesaikan dengan menggunakan persamaan linear satu variabel dengan menetapkan  $x$  sebagai jarak tempuh (dalam km). Mudah diperoleh persamaan berikut ini.

$$5000x + 10000 = 50000 \text{ atau disederhanakan menjadi } 5x + 10 = 50.$$

Ini merupakan bentuk persamaan linear (2.1.1) dan penyelesaiannya adalah  $x = \frac{50-10}{5} = 8$ . Jadi, jawabannya adalah 8 km. ■

Persamaan yang bentuknya selain dari (2.1.1) disebut **persamaan taklinear**. Beberapa persamaan taklinear sudah dikenal bahkan sejak di sekolah menengah, diantaranya persamaan kuadrat, persamaan trigonometri dan persamaan logaritma atau eksponen. Beberapa contoh persamaan taklinear adalah sebagai berikut.

**Contoh 2.2.** Beberapa bentuk persamaan taklinear

- **Persamaan kuadrat**

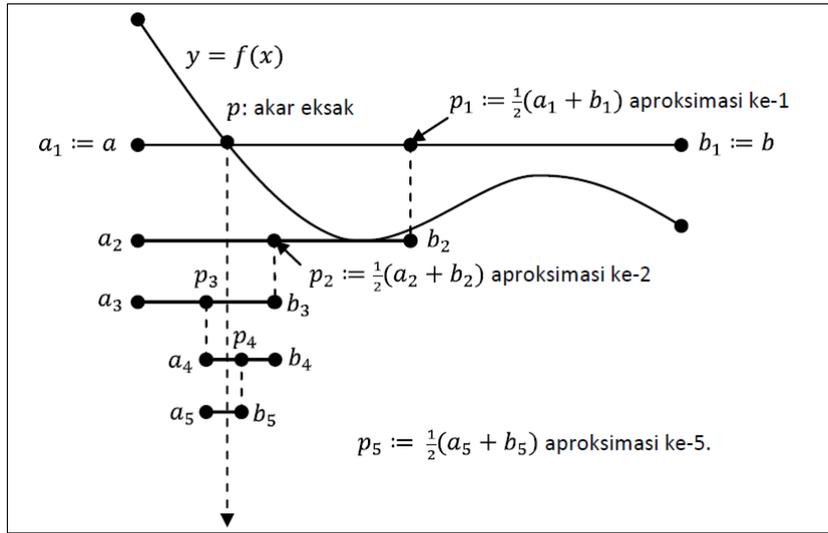
$$ax^2 + bx + c = 0, \quad a \neq 0$$

mempunyai penyelesaian  $x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$ . Penyelesaian ini dikatakan ada jika ia bernilai real, yaitu jika diskriminan  $D := b^2 - 4ac \geq 0$ . Jadi persamaan ini paling banyak mempunyai dua akar.

- **Persamaan suku banyak atau polinomial**

$$a_0 + a_1x + \cdots + a_{n-1}x^{n-1} + a_nx^n = 0, \quad a_n \neq 0.$$

Persamaan polinomial ini paling banyak mempunyai  $n$  akar berbeda. Dapat pula ia tidak mempunyai akar sama sekali, yaitu tak satu pun bilangan real  $x$  yang memenuhi persamaan ini.



Gambar 2.5: Skema metode bagidua

diperkenalkan pada tahap awal adalah metode bagidua (*bisection*), metode secant, metode false position. Selanjutnya diperkenalkan metode yang paling luas digunakan yaitu metode Newton. Sebagai perumuman metode iterasi untuk aproksimasi akar persamaan taklinear juga akan dibahas metode iterasi titik tetap (*fixed-point iteration*).

## 2.3 Metode Bagidua

Metode bagidua (*bisection*) ini merupakan metode paling sederhana yang digunakan untuk mengaproksimasi akar persamaan taklinear. Pada metode ini kita mulai pada suatu interval yang memuat akar. Teorema 2.1 dapat digunakan dalam memilih interval ini. Ide dasar metode ini adalah membagi interval yang memuat akar menjadi dua bagian yang sama panjang, kemudian mempertahankan subinterval yang memuat akar dan membuang subinterval interval lainnya. Proses ini dilakukan terus-menerus sampai terbentuk subinterval yang cukup sempit. Diperoleh barisan interval bersarang (*nested-intervals*) yang kesemuanya memuat akar, yaitu

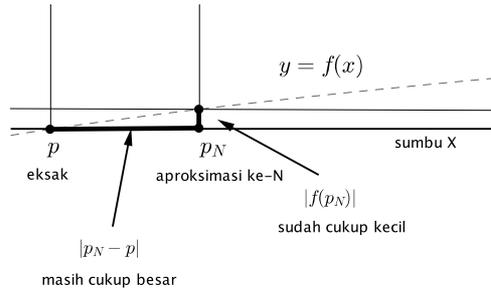
$$[a_1, b_1] \supset [a_2, b_2] \supset \cdots \supset [a_n, b_n] \supset \cdots$$

Karena akarnya terdapat di dalam interval yang sempit maka aproksimasi akar diambil sebagai titik tengah interval ini. Metode ini diilustrasi pada Gambar 2.5.

Berdasarkan skema ini, kita mulai dengan mengambil  $a_1 := a$  dan  $b_1 := b$ , kemudian mengambil  $p_1$  sebagai titik tengah interval, yaitu

$$p_1 := \frac{1}{2}(a_1 + b_1).$$

Bila  $f(p_1) = 0$  maka akarnya adalah  $p_1$ . Tapi bila  $f(p_1) \neq 0$  maka  $f(p_1)$  mempunyai tanda positif atau negatif. Karena  $f(a_1) \neq 0$  maka pasti berlaku salah satu, yaitu  $f(p_1)f(a_1) < 0$  atau  $f(p_1)f(a_1) > 0$ . Bila  $f(p_1)f(a_1) < 0$  maka akarnya pasti terletak pada subinterval  $[a_1, p_1]$  sehingga harus diambil  $a_2 := a_1$  dan



Gambar 2.8: Grafik fungsi datar di sekitar akar eksak

diperoleh relatif sama dengan sebelumnya maka aproksimasi yang kita peroleh sudah mentok ke batas atau limitnya. Secara formal fakta ini menggunakan konsep barisan Cauchy pada bilangan real.

5. Kriteria stopping lainnya adalah lebar interval  $[a_n, b_n]$ . Karena interval ini memuat akar maka jarak antara akar eksak  $p$  dan aproksimasi  $p_n$  tidak akan melebihi setengah panjang interval ini. Jadi cukup dipasang kriteria berhenti ketika  $|b_n - a_n| < TOL$  untuk suatu  $TOL$  yang cukup kecil.

## 2.4 Metode Secant dan False-Position

### 2.4.1 Metode secant

Pada metode bagidua, fungsi  $f$  yang mendefinisikan persamaan  $f(x) = 0$  tidak digunakan dalam menentukan akar aproksimasi. Secara intuitif ketika akarnya terletak di bagian sangat pinggir maka metode bagidua membutuhkan waktu lebih lama untuk mencapai akar ini. Sebuah pertimbangan yang cukup logis bahwa jika  $|f(a)| < |f(b)|$  maka akarnya besar kemungkinan lebih dekat ke  $a$  daripada ke  $b$ . Dalam keadaan seperti ini, absis titik potong garis yang melalui  $(a, f(a))$  dan  $(a, f(b))$  tentunya lebih dekat ke  $a$  daripada ke  $b$ . Inilah ide dasar yang mengilhami metode secant.

Garis secant pada sebuah kurva  $y = f(x)$  adalah garis yang melalui dua titik berbeda pada kurva tersebut, kadang disebut juga tali busur. Sebagai perbandingan, garis yang hanya melalui (menyentuh) sebuah titik pada kurva disebut garis singgung di titik tersebut. Garis singgung disebut juga garis tangen. Dalam metode secant, aproksimasi akar ditentukan dengan mengambil absis titik potong antara garis secant dan sumbu  $X$ . Ilustrasi metode secant diberikan pada Gambar 2.9. Diperhatikan akar eksak  $p$  diaproksimasi berturut-turut oleh  $p_1, p_2$ , dan  $p_3$ . Proses ini diteruskan untuk memperoleh aproksimasi berikutnya. Pada metode secant, fungsi  $f$  dilibatkan dalam menentukan aproksimasi.

Untuk menurunkan formula iterasi pada metode secant kita perlu menentukan terlebih dahulu persamaan garis yang melalui  $(a, f(a))$  dan  $(b, f(b))$ . Untuk keperluan ini kita gunakan formula persamaan garis lurus yang berbentuk  $y - y_1 = m(x - x_1)$  dengan  $m$  gradien garis yang melalui  $(a, f(a))$  dan  $(b, f(b))$ , yaitu

$$m = \frac{f(b) - f(a)}{b - a}.$$

Untuk  $(x_1, y_1)$  kita dapat mengambil salah satu titik  $(a, f(a))$  atau  $(b, f(b))$ . Dalam hal ini kita ambil  $(b, f(b))$  yaitu  $x_1 := b$  dan  $y_1 := f(b)$ . Untuk memudahkan pola pengindekan, selanjutnya tetapkan  $p_{-1} := a$  dan  $p_0 := b$  diperoleh  $m = \frac{f(p_0) - f(p_{-1})}{p_0 - p_{-1}}$ . Jadi, persamaan garis secant yang melalui  $(p_{-1}, f(p_{-1}))$

4. Gunakan m-file bagidua1.m pada interval  $[0, 1]$  dan  $\text{tol} = 0.0001$  untuk menyelesaikan persamaan  $x - e^{-x/c} = 0$  dengan berbagai macam nilai  $c$ ,  $c = 0.1, 1.5, 2.2, 7.8, 13.7, 20, 75$ . Amstrong dan Kulesza (AK) pada 1981 menemukan rumus eksplisit untuk aproksimasi akar persamaan ini sebagai berikut:

$$x = cu \left( 1 - \frac{\ln[(1+c)u]}{1+u} \right) \text{ dengan } u = \ln\left(1 + \frac{1}{c}\right).$$

Bandingkan selisih antara aproksimasi dengan metode bagidua dan formula AK ini.

### 2.5.3 Implementasi numerik metode secant

Pada implementasi ini kita terapkan pada masalah menentukan angka kelahiran  $\lambda$  pada pertumbuhan populasi yang telah diberikan pada Contoh 1.3, yaitu:

$$1000\lambda e^\lambda - 1025\lambda - 435 = 0.$$

Untuk memudahkan menentukan lokasi akar, kita dapat menampilkan grafik fungsi  $f(x) = 1000xe^x - 1025x - 435$  dengan perintah sebagai berikut:

```
>>x=-1:0.0001:1;
>>y=1000*x.*exp(x)-1025*x-435;
>>plot(x,y);grid
```

Untuk menggambarkan dua dot (•) yang menunjukkan lokasi akar, kita gunakan perintah sebagai berikut:

```
>>d=find(abs(y)<0.05)
d =
    2276    15765
>> hold on
>> plot(x(d),y(d),'.','MarkerSize',20)
```

Ternyata ada dua akar yang terdeteksi, yaitu satu di dalam  $[-1, 0]$  dan satu lagi di dalam  $[0, 1]$ . Karena  $\lambda$  menyatakan angka kelahiran maka akar yang relevan adalah akar yang berada di dalam  $[0, 1]$ .

Penyusunan m-file berikut menggunakan formula iterasi secant yang pertama, yaitu  $p_n = p_{n-1} - \frac{f(p_{n-1})(p_{n-1}-p_{n-2})}{f(p_{n-1})-f(p_{n-2})}$  dan kriteria stopping 3 yaitu selisih antara dua iterasi berurutan  $|p_n - p_{n-1}| < TOL$ .

```
function [akar,langkah]=secant3(a,b,tol)
p_1 = a; p0 = b;
f_1 = feval('fun',p_1); f0 = feval('fun',p0);
langkah = 0; akar=[];
while abs(p_1-p0) > tol
    langkah = langkah+1;
    p = p0-f0*(p0-p_1)/(f0-f_1);
    akar =[akar;p];
    p_1 = p0; p0 = p;
    f_1 = feval('fun',p_1);
    f0 = feval('fun',p0);
end
```

```

0.371626952356482
0.415303359884348
0.433754224388734
0.441598179720563
0.444941669570424
0.446368427890417
0.446977554365172
langkah =
8

```

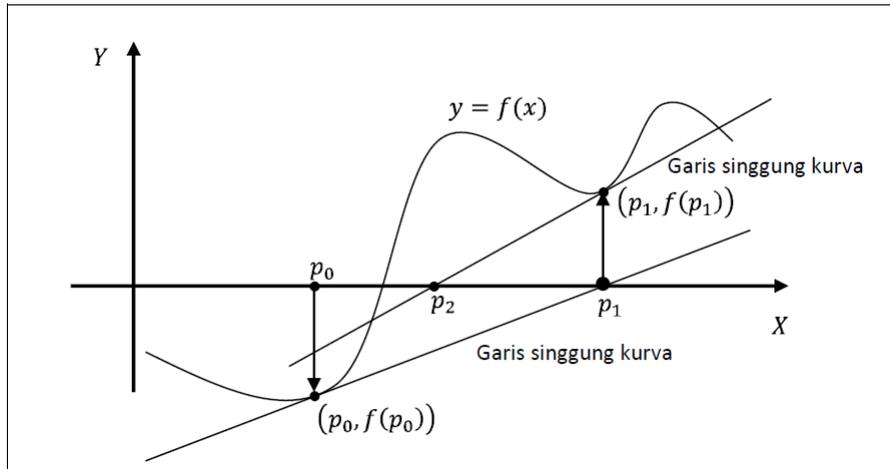
Ternyata metode false-position berhasil menyelesaikan masalah ini yang sebelumnya tidak dapat diselesaikan dengan metode secant.

**Masalah Responsi 2.5.5.** Metode secant dan false-position memiliki kemiripan. Untuk itu pada responsi ini Anda diminta untuk melakukan eksperimen numerik untuk eksplorasi lebih lanjut keunggulan dan kelemahan kedua metode ini. Diberikan persamaan taklinear

$$(x^3 - 4x^2 + x + 1) \sin 3x = 0.$$

Persamaan ini mempunyai cukup banyak akar di dalam interval  $[0, 5]$  seperti ditunjukkan pada Gambar 2.13. Lakukan eksperimen metode false-position dengan menggunakan m-file falsepos3.m.

1. Cobakan interval  $[k, k + 1]$  dengan  $k = 0, 1, 2, 3, 4$  dan  $TOL=0.0001$ . Untuk tiap percobaan, tulis akar aproksimasi yang diperoleh. Urutkan ke-tujuh akar yang terlihat menurut label akar1, akar2, akar3, akar4, akar5, akar6, dan akar7. Ke akar mana sajakah aproksimasi konvergen pada setiap percobaan? Akar mana saja yang tidak dapat dijangkau oleh percobaan ini? Bandingkan hasil metode false-position
2. Kalau pada eksperimen 1 digunakan interval bilangan bulat berurutan, lakukan eksperimen dengan mengambil  $[a, b]$  dengan  $a$  dan  $b$  bilangan bulat sebarang di dalam  $\{0, 1, 2, 3, 4, 5\}$ , misalnya  $[0, 2], [1, 4]$  dan sebagainya. Temukan semua interval  $[a, b]$  yang dapat menjangkau akar yang tidak dapat dijangkau pada eksperimen 1.
3. Salah satu kelebihan metode secant adalah iterasinya dapat dimulai dari interval yang tidak memuat akar, namun barisan aproksimasi yang diperoleh konvergen ke akar eksak. Hal ini sudah kita ujicobakan pada persamaan  $1000\lambda e^\lambda - 1025\lambda - 435 = 0$  dengan mengambil interval  $[1, 2]$  dan  $TOL = 10^{-7}$ , yaitu ketika menjalankan perintah `>>[akar,langkah]=secant3(1,2,1e-7)` diperoleh hasil  $p_8 = 0.576420717744457$ . Lakukan hal yang sama dengan menggunakan metode false-position. Apa yang Anda dapatkan? Mengapa hal ini terjadi?
4. Lakukan eksperimen numerik pada berbagai interval yang memuat akar  $[0, 1]$  dan interval yang tidak memuat akar  $[1, 2], [2, 3], [3, 4]$ , dan  $[4, 5]$ . Apakah metode false-position masih konvergen? Kalau iya, berapa banyak langkah yang dibutuhkan untuk memenuhi toleransi  $10^{-7}$  tersebut. Bandingkan dengan metode secant. Untuk kasus ini, mana yang lebih unggul?
5. Tuliskan sifat-sifat metode false-position yang Anda temukan melalui eksperimen yang telah dilakukan sebelumnya.



Gambar 2.14: Skema metode Newton

## 2.6 Metode Newton dan Iterasi Titik Tetap

### 2.6.1 Metode Newton

Metode ini merupakan metode yang paling populer karena secara umum kekonvergenannya lebih cepat dari metode lainnya dan implementasinya sederhana. Kalau metode secant dan metode false-position menggunakan garis secant kemudian dipotongkan dengan sumbu  $X$  untuk mendapatkan aproksimasinya, maka metode Newton menggunakan garis tangen, yaitu garis singgung kurva di sebuah titik pada kurva tersebut. Pada metode Newton ini hanya dibutuhkan satu titik awal untuk membuat garis tangen.

Misalkan  $p_0$  titik awal yang dipilih maka  $p_1$  diambil sebagai absis titik potong garis singgung kurva  $y = f(x)$  dititik  $(p_0, f(p_0))$ . Selanjutnya, melalui titik  $(p_1, f(p_1))$  dibuat garis singgung untuk mendapatkan  $p_2$ . Untuk lebih jelasnya, skema Newton ini ditunjukkan pada Gambar 2.14.

Untuk membangun formula iterasi metode Newton ini, pertama kita tentukan persamaan garis singgung kurva  $y = f(x)$  yang melalui  $(p_0, f(p_0))$ . Kita gunakan persamaan garis yang berbentuk

$$y - f(p_0) = m(x - p_0)$$

dengan  $m = f'(p_0)$  derivatif pertama  $f$  di titik  $p_0$  adalah gradien garis singgung tersebut. Titik potong garis ini dengan sumbu  $X$  diperoleh dengan menetapkan  $y = 0$ , yaitu

$$\begin{aligned} 0 - f(p_0) &= f'(p_0)(x - p_0) \\ x - p_0 &= -\frac{f(p_0)}{f'(p_0)} \\ x &= p_0 - \frac{f(p_0)}{f'(p_0)}. \end{aligned}$$

Selanjutnya, aproksimasi pertama diperoleh dengan mengambil  $p_1 := x$  sehingga diperoleh

$$p_1 = p_0 - \frac{f(p_0)}{f'(p_0)}.$$

di mana  $\xi_0$  terletak di antara  $\alpha$  dan  $x_0$ , yakni  $\xi_0 \in J$ . Jadi berlaku  $|f''(\xi_0)| \leq \max_{x \in J} |f''(x)|$ . Diperoleh

$$\begin{aligned} |\alpha - x_1| &= \frac{1}{2}(\alpha - x_0)^2 \frac{|f''(\xi_0)|}{|f'(x_0)|} \\ &\leq (\alpha - x_0)^2 \frac{\max |f''(x)|}{2 \min |f'(x)|}, x \in J \\ &= (\alpha - x_0)^2 M. \end{aligned}$$

Jika  $x_0$  dipilih sehingga  $M|\alpha - x_0| < 1$  maka diperoleh  $|\alpha - x_1| < |\alpha - x_0|M|\alpha - x_0| < |\alpha - x_0|$ . Ini berarti terjadi penurunan galat, yakni  $x_1$  lebih dekat ke  $\alpha$  daripada  $x_0$  ke  $\alpha$ . Dengan asumsi  $x_2, x_3, \dots$  kesemuanya berada di dalam  $J$  maka diperoleh bentuk umum

$$\begin{aligned} |\alpha - x_{n+1}| &= \frac{1}{2}(\alpha - x_n)^2 \frac{|f''(\xi_n)|}{|f'(x_n)|} \\ &\leq (\alpha - x_n)^2 M. \end{aligned}$$

Misalkan  $e_n := |\alpha - x_n|$  yakni galat iterasi ke- $n$ , maka diperoleh  $e_{n+1} = e_n^2 M = M^{-1}(Me_n)^2$ . Secara rekursif diperoleh  $e_1 = M^{-1}(Me_0)^2$ ,  $e_2 = M^{-1}(Me_1)^2 = M^{-1}(MM^{-1}(Me_0)^2)^2 = M^{-1}(Me_0)^4$ ,  $e_3 = M^{-1}(Me_2)^2 = M^{-1}(MM^{-1}(Me_0)^4)^2 = M^{-1}(Me_0)^8$ , dan seterusnya diperoleh bentuk umum

$$e_n = M^{-1}(Me_0)^{2^n}.$$

Ketika  $Me_0 = M|\alpha - x_0| < 1$ ; dengan kata lain  $x_0$  cukup dekat dengan  $\alpha$  maka  $e_n \rightarrow 0$  yakni  $x_n \rightarrow \alpha$ . Pembuktian masalah kekonvergenan barisan aproksimasi Newton selesai. Berikutnya dibuktikan ekspresi kedua yang menggambarkan order konvergen metode Newton. Formula estimasi galat metode Newton diberikan oleh

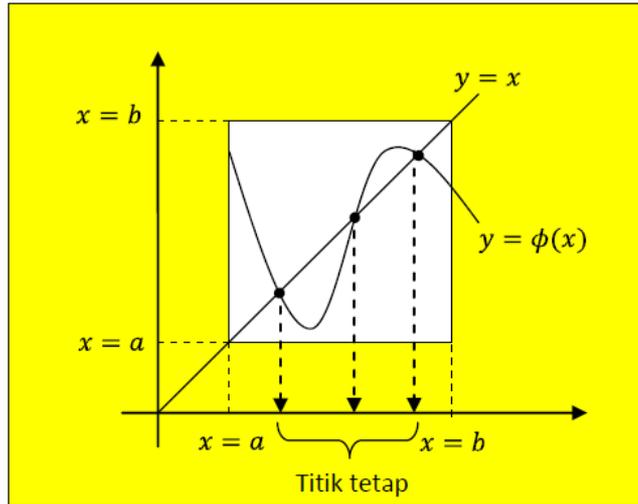
$$\begin{aligned} \alpha - x_{n+1} &= -\frac{1}{2}(\alpha - x_n)^2 \frac{f''(\xi_n)}{f'(x_n)} \\ \frac{\alpha - x_{n+1}}{(\alpha - x_n)^2} &= -\frac{1}{2} \frac{f''(\xi_n)}{f'(x_n)}. \end{aligned}$$

Dengan mengambil limit untuk  $n \rightarrow \infty$  pada kedua ruas, mengingat  $\xi_n$  terletak di antara  $\alpha$  dan  $x_n$ ,  $x_n \rightarrow \alpha$ , dan  $f$  dan  $f''$  kontinu maka diperoleh

$$\begin{aligned} \lim_{n \rightarrow \infty} \frac{\alpha - x_{n+1}}{(\alpha - x_n)^2} &= -\lim_{n \rightarrow \infty} \frac{1}{2} \frac{f''(\xi_n)}{f'(x_n)} \\ &= -\frac{1}{2} \frac{\lim_{n \rightarrow \infty} f''(\xi_n)}{\lim_{n \rightarrow \infty} f'(x_n)} \\ &= -\frac{1}{2} \frac{f''(\lim_{n \rightarrow \infty} \xi_n)}{f'(\lim_{n \rightarrow \infty} x_n)} = -\frac{1}{2} \frac{f''(\alpha)}{f'(\alpha)}. \end{aligned}$$

□

Sekali lagi bahwa teorema ini hanya memberikan landasan teoretis adanya nilai awal  $x_0$  yang dapat menjamin bahwa metode Newton konvergen. Eksistensi nilai awal pada teorema ini tidak konstruktif dalam arti ia tidak dapat ditemukan secara eksplisit sehingga dapat digunakan dalam implementasi. Oleh karena itulah penentuan iterasi awal dalam implementasi metode Newton umumnya menggunakan



Gambar 2.20: Ilustrasi ketidaktunggalan titik tetap

Bukti. Eksistensi titik tetap sudah dijamin oleh Teorema 2.8. Selanjutnya kita tunjukkan bahwa titik tetap ini tunggal. Andai titik tetapnya tidak tunggal, katakan  $\alpha$  dan  $\beta$  dengan  $\alpha \neq \beta$ . Berdasarkan Teorema nilai rerata (TNR) selalu terdapat  $c$  di antara  $\alpha$  dan  $\beta$  sehingga

$$\phi(\alpha) - \phi(\beta) = \phi'(c)(\beta - \alpha).$$

Karena  $\alpha$  dan  $\beta$  diasumsikan titik tetap maka berlaku  $\alpha = \phi(\alpha)$  dan  $\beta = \phi(\beta)$ . Karena diketahui  $|\phi'(c)| < 1$  maka dari relasi sebelumnya diperoleh

$$|\phi(\beta) - \phi(\alpha)| = |\alpha - \beta| = |\phi'(c)||\beta - \alpha| < |\beta - \alpha| = |\alpha - \beta|,$$

yakni diperoleh  $|\alpha - \beta| < |\alpha - \beta|$ , suatu kontradiksi. Jadi titik tetapnya haruslah tunggal.  $\square$

**Contoh 2.11.** Perhatikan persamaan taklinear  $x + \ln x = 0$ . Selesaikan persamaan ini dengan menggunakan iterasi titik tetap.

**PENYELESAIAN.** Persamaan ini dapat ditulis dalam beberapa bentuk yang ekuivalen, di antaranya sebagai berikut

$$x = -\ln x, \quad x = e^{-x}, \quad x = (x + e^{-x})/2.$$

Jadi dalam kasus ini kita mempunyai fungsi iterasi, yaitu  $\phi_1(x) := -\ln x$ ,  $\phi_2(x) := e^{-x}$  dan  $\phi_3(x) := (x + e^{-x})/2$ . Selanjutnya, kita lihat barisan aproksimasi yang dihasilkan oleh ketiga iterasi ini. Untuk keperluan ini kita ambil nilai awal  $x_0 = 0.3$ . Hasilnya sebagai berikut

- Untuk fungsi iterasi  $\phi_1$  diperoleh  $x_1 = \phi_1(0.3) = -\ln 0.3 = 1.2040$ ,  $x_2 = \phi_1(1.2040) = -\ln 1.2040 = -0.1856$
- Untuk fungsi iterasi  $\phi_2$  diperoleh  $x_1 = \phi_2(0.3) = e^{-0.3} = 0.7408$ ,  $x_2 = \phi_2(0.7408) = e^{-0.7408} = 0.4767$

– Untuk fungsi iterasi  $\phi_3$  diperoleh  $x_1 = \phi_3(0.3) = \frac{0.3+e^{-0.3}}{2} = 0.5204$ ,  $x_2 = \phi_3(0.5204) = \frac{0.5204+e^{-0.5204}}{2} = 0.5573$ .

Selanjutnya 7 suku pertama barisan yang dihasilkan oleh ketiga fungsi iterasi ini disajikan pada tabel berikut.

$n$	$x_{n+1} = \phi_1(x_n)$	$x_{n+1} = \phi_2(x_n)$	$x_{n+1} = \phi_3(x_n)$
0	1.2040	0.7408	0.5204
1	-0.1856	0.4767	0.5573
2	1.6840-3.1416i	0.6208	0.5650
3	-1.2710+1.0787i	0.5375	0.5667
4	-0.5111-2.4378i	0.5842	0.5670
5	-0.9126+1.7774i	0.5576	0.5671
6	-0.6922-2.0451i	0.5726	0.5671

Tabel 2.2: Nilai-nilai yang diperoleh melalui fungsi iterasi  $\phi_1$ ,  $\phi_2$ , dan  $\phi_3$ .

Berdasarkan pola bilangan yang terdapat pada tabel ini terlihat bahwa fungsi iterasi pertama tidak memberikan barisan konvergen sebab pada iterasi kedua sudah menghasilkan bilangan imajiner  $i$ . Sedangkan iterasi kedua dan iterasi ketiga menunjukkan konvergen karena cenderung menuju bilangan yang sama. ■

**Contoh 2.12.** Buktikan fungsi iterasi  $\phi(x) = e^{-x}$  mempunyai titik tetap yang tunggal di dalam interval  $[0, 1]$ .

**PENYELESAIAN.** Jelas  $\phi$  kontinu karena ia fungsi eksponensial. Perhatikan  $\phi(0) = e^{-0} = 1$  dan  $\phi(1) = e^{-1} = 0.3679$  sehingga  $\phi(0), \phi(1) \in [0, 1]$ . Jadi asumsi pada Teorema 2.8 terpenuhi sehingga titik tetapnya terjamin ada. Untuk memastikan ketunggalannya, kita tentukan derivatif  $\phi$ , yaitu  $\phi'(x) = -e^{-x}$ . Selanjutnya,  $|\phi'(x)| = e^{-x} = \frac{1}{e^x}$  merupakan fungsi turun sehingga  $|\phi'(x)| < |\phi'(0)| = 1$ , jadi asumsi pada Teorema 2.9 yaitu  $|\phi'(x)| < 1$  untuk setiap  $x \in (0, 1)$  dipenuhi. Akhirnya terbukti bahwa titik tetapnya tunggal. ■

### 2.6.6 Kekonvergenen iterasi titik tetap

Pada contoh sebelumnya kita telah mendefinisikan tiga fungsi iterasi  $\phi_1(x) = -\ln(x)$ ,  $\phi_2(x) = e^{-x}$ , dan  $\phi_3(x) = (e^x + e^{-x})/2$ . Semua fungsi iterasi ini mempunyai titik tetap yang sama dan tunggal yaitu  $\alpha \approx 0.57$  sebagaimana ditunjukkan pada Gambar 2.21. Pada kenyataannya fungsi iterasi  $\phi_2$  dan  $\phi_3$  menghasilkan barisan yang konvergen sedangkan fungsi iterasi  $\phi_1$  tidak. Jadi, ada dan tunggalnya titik tetap tidak menjamin kekonvergenan barisan yang dihasilkan. Pada bagian ini kita membahas syarat cukup agar sebuah fungsi iterasi menghasilkan barisan yang konvergen.

**Teorema 2.10.** Misalkan  $\phi$  kontinu pada  $[a, b]$  dengan  $a \leq \phi(x) \leq b$  dan terdapat  $0 < \gamma < 1$  sehingga  $|\phi(x) - \phi(y)| \leq \gamma|x - y|$  untuk setiap  $x, y \in [a, b]$ . Maka  $\phi$  mempunyai titik tetap tunggal  $\alpha$  dan barisan  $(x_n)$  yang diperoleh dari iterasi  $x_{n+1} = \phi(x_n)$  konvergen ke  $\alpha$  untuk sebarang nilai awal  $x_0 \in [a, b]$ . Lebih lanjut berlaku estimasi galat awal sebagai berikut:

$$|\alpha - x_n| \leq \frac{\gamma^n}{1 - \gamma} |x_1 - x_0|. \quad (2.6.14)$$

## BAB 3

# Polinomial Interpolasi

*Dia menyingsingkan pagi dan menjadikan malam untuk beristirahat, dan (menjadikan) matahari dan bulan untuk perhitungan. Itulah ketentuan Allah Yang Maha Perkasa lagi Maha Mengetahui.*

QS 6:96.

### 3.1 Pendahuluan

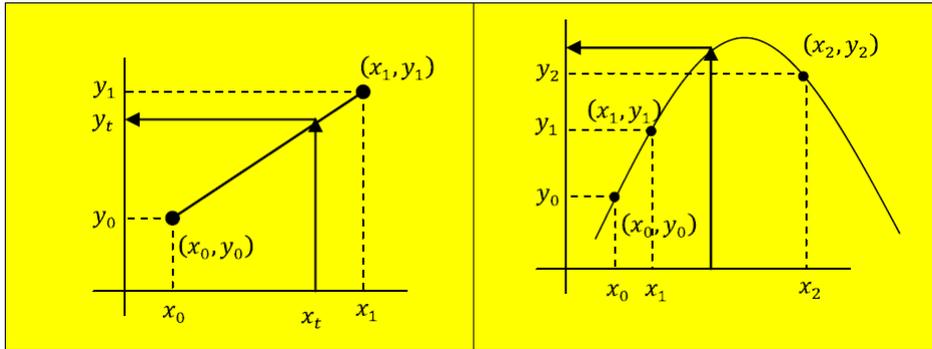
Dalam kehidupan sehari-hari kita sering berjumpa dengan tabel yang memuat angka-angka hasil pengukuran beberapa variabel. Misalnya data mengenai jarak tempuh suatu benda yang diukur setiap menit, frekuensi denyut nadi yang diukur setiap 5 detik, kecepatan angin dalam setiap jam, temperatur dan tekanan udara dalam setiap jam, penambahan penduduk atau besar populasi yang dihitung setiap tahun, peningkatan konsumsi bahan bakar minyak setiap bulan, dan lain sebagainya. Di antara beberapa variabel tersebut biasanya terdapat paling tidak satu variabel bebas. Variabel waktu dalam satuan detik, menit, jam, bulan, atau tahun biasanya ditetapkan sebagai variabel bebas.

Sebagai ilustrasi, misalkan  $s(t)$  jarak tempuh suatu objek bergerak (dalam meter) setelah berjalan selama  $t$  (dalam menit). Dari pengukuran pada 10 menit pertama diperoleh data seperti pada Tabel 3.1. Data ini dapat pula tersajikan dalam bentuk grafik seperti pada Gambar 3.1. Berdasarkan data ini kita dapat menentukan jarak tempuh benda pada waktu-waktu tertentu, misalnya 75 meter setelah berjalan 4 menit, 180 meter setelah berjalan 8 menit. Tetapi kita tidak dapat memastikan jarak yang telah ditempuh benda setelah berjalan  $4\frac{1}{2}$  menit karena jarak tidak diukur pada saat ini. Sebaliknya kita tidak dapat menentukan dengan pasti kapan saat benda tersebut telah menempuh jarak 130 meter karena tidak ada data yang sesuai pada tabel. Untuk kasus ini orang biasanya melakukan perkiraan. Misalkan jarak yang ditempuh oleh objek tersebut setelah berjalan  $4\frac{1}{2}$  menit diperkirakan di antara 75 meter dan 100 meter, katakan 18.5 meter dengan asumsi kecepatannya konstan. Cara ini masih dapat diterima karena fungsi jarak  $s(t)$  merupakan fungsi naik tegas.

Ilustrasi kedua adalah data kecepatan angin yang diukur pada setiap jam seperti disajikan pada

Waktu tempuh ( $t$ )	1	2	3	4	5	6	7	8	9	10
Jarak tempuh ( $s(t)$ )	30	45	65	75	100	120	155	180	200	215

Tabel 3.1: Data jarak tempuh objek bergerak



Gambar 3.2: Interpolasi polinomial linear dan kuadratik

Bab ini membahas berbagai metode interpolasi polinomial seperti metode Lagrange dan metode selisih terbagi atau formula Newton. Selain kedua metode ini, interpolasi diperluas tidak hanya untuk menentukan nilai fungsi atau data yang hilang tetapi juga nilai derivatifnya yaitu pada kajian interpolasi Hermite. Pada bagian akhir bab ini diperkenalkan polinomial kuadrat terkecil sebagai salah satu bentuk regresi yang banyak digunakan dalam statistika. Interpolasi polinomial dan regresi polinomial merupakan dua metode yang sering digunakan untuk mengaproksimasi dan mengestimasi fungsi atau data.

## 3.2 Metode Interpolasi Lagrange

### 3.2.1 Polinomial Lagrange

Secara intuitif, melalui dua titik yang berlainan selalu dapat dibentuk polinomial derajat satu, melalui tiga titik berlainan selalu dapat dibentuk polinomial derajat dua, dan seterusnya. Selanjutnya, polinomial yang melalui titik-titik yang diberikan disebut polinomial interpolasi. Sebelum eksistensi polinomial interpolasi ditunjukkan secara formal, kita perhatikan dulu satu kasus sederhana berikut sebagai ilustrasi.

Misalkan diketahui dua titik berlainan  $(x_0, y_0)$  dan  $(x_1, y_1)$  yakni  $x_0 \neq x_1$  maka berbagai formula dapat digunakan untuk menentukan persamaan garis lurus yang melalui kedua titik ini seperti telah diperkenalkan di sekolah menengah seperti  $y - y_0 = m(x - x_0)$  dengan  $m$  gradien garis atau pun rumus  $\frac{x-x_0}{x_1-x_0} = \frac{y-y_0}{y_1-y_0}$ . Pada kesempatan ini kita tidak menggunakan formula yang sudah ada, tetapi persamaan garis lurus tersebut dibangun secara bertahap. Pertama, dibangun dua buah polinomial derajat satu sebagai berikut:

$$L_0(x) := \frac{x - x_1}{x_0 - x_1} \text{ dan } L_1(x) := \frac{x - x_0}{x_1 - x_0}. \quad (3.2.1)$$

Diperoleh sifat  $L_0(x_0) = 1$ ,  $L_0(x_1) = 0$  dan  $L_1(x_0) = 0$ ,  $L_1(x_1) = 1$ . Sifat ini diilustrasikan pada Gambar 3.3(kiri). Selanjutnya didefinisikan polinomial  $P(x)$  sebagai kombinasi linear kedua polinomial Lagrange tersebut sebagai berikut:

$$P(x) = y_0 L_0(x) + y_1 L_1(x). \quad (3.2.2)$$

Diperoleh bahwa  $P(x)$  merupakan polinomial derajat satu dengan sifat  $P(x_0) = y_0 L_0(x_0) + y_1 L_1(x_0) = y_0 \cdot 1 + y_1 \cdot 0 = y_0$  dan  $P(x_1) = y_0 L_0(x_1) + y_1 L_1(x_1) = y_0 \cdot 0 + y_1 \cdot 1 = y_1$ . Polinomial hasil kombinasi ini tidak lain adalah garis lurus yang melalui  $(x_0, y_0)$  dan  $(x_1, y_1)$  seperti ditunjukkan pada Gambar 3.3 (kanan). Kedua polinomial  $L_0$  dan  $L_1$  disebut polinomial Lagrange untuk  $x_0$  dan  $x_1$ .

Interpolasi	Node yang digunakan	#(interpolasi)
linear	(1.4, 1.6)	1
kuadrat	(1.3, 1.4, 1.6), (1.4, 1.6, 1.8)	2
kubik	(1.1, 1.3, 1.4, 1.6), (1.3, 1.4, 1.6, 1.8), (1.4, 1.6, 1.8, 2.0)	3
derajat 4	(1.0, 1.1, 1.3, 1.4, 1.6), (1.1, 1.3, 1.4, 1.6, 1.8), (1.3, 1.4, 1.6, 1.8, 2.0)	3
derajat 5	(1.0, 1.1, 1.3, 1.4, 1.6, 1.8), (1.1, 1.3, 1.4, 1.6, 1.8, 2.0)	2
derajat 6	(1.0, 1.1, 1.3, 1.4, 1.6, 1.8, 2.0)	1
	Total	12

Tabel 3.3: Semua kemungkinan interpolasi

Jadi terdapat 12 kemungkinan interpolasi untuk mengaproksimasi nilai  $y = f(1.5)$ . ■

**Contoh 3.6.** Tabel berikut menyajikan data populasi suatu negara dari 1940 sampai 1990 yang disajikan setiap 10 tahun. Gunakan polinomial derajat 5 untuk mengestimasi jumlah penduduk pada tahun 1955 dan tahun 1981.

Tahun	1940	1950	1960	1970	1980	1990
Populasi	132165	151326	179323	203302	226542	249633

Tabel 3.4: Data jumlah penduduk dalam ribuan

**PENYELESAIAN.** Untuk mudahnya kita dapat menggunakan transformasi variabel tahun dengan mengambil 1940 sebagai patokan berkoresponden dengan  $t = 0$ . Selanjutnya 1950 berkoresponden dengan  $t = 10$ , dan seterusnya. Tabel di atas dapat ditulis kembali dalam bentuk baru berikut.

$t$	0	10	20	30	40	50
$y$	132165	151326	179323	203302	226542	249633

Pertama, temukan terlebih dahulu polinomial derajat 5 untuk pasangan titik pada tabel ini dengan cara menggunakan (3.3) dan (3.2.6) sehingga diperoleh

$$\begin{aligned}
 P_5(x) &= 132165 \frac{(x-10)(x-20)(x-30)(x-40)(x-50)}{(0-10)(0-20)(0-30)(0-40)(0-50)} \\
 &+ 151326 \frac{(x-0)(x-20)(x-30)(x-40)(x-50)}{(10-0)(10-20)(10-30)(10-40)(10-50)} \\
 &+ 179323 \frac{(x-0)(x-10)(x-30)(x-40)(x-50)}{(20-0)(20-10)(20-30)(20-40)(20-50)} \\
 &+ 203302 \frac{(x-0)(x-10)(x-20)(x-40)(x-50)}{(30-0)(30-10)(30-20)(30-40)(30-50)} \\
 &+ 226542 \frac{(x-0)(x-10)(x-20)(x-30)(x-50)}{(40-0)(40-10)(40-20)(40-30)(40-50)} \\
 &+ 249633 \frac{(x-0)(x-10)(x-20)(x-30)(x-40)}{(50-0)(50-10)(50-20)(50-30)(50-40)}.
 \end{aligned}$$

Untuk meyakinkan bahwa polinomial ini benar-benar melalui titik-titik pada tabel, gambarkan grafiknya, misalnya dengan menggunakan Matlab. Grafiknya ditunjukkan pada Gambar 3.6. Grafik ini

Selisih terbagi tingkat dua ada 3 nilai, yaitu

$$\begin{aligned} f[x_0, x_1, x_2] &= \frac{f[x_1, x_2] - f[x_0, x_1]}{x_2 - x_0} = \frac{1.3520 - 1.1070}{0.4 - 0.0} = 0.6125 \\ f[x_1, x_2, x_3] &= \frac{f[x_2, x_3] - f[x_1, x_2]}{x_3 - x_1} = \frac{1.6515 - 1.3520}{0.6 - 0.2} = 0.7487 \\ f[x_2, x_3, x_4] &= \frac{f[x_3, x_4] - f[x_2, x_3]}{x_4 - x_2} = \frac{2.0170 - 1.6515}{0.8 - 0.4} = 0.9137. \end{aligned}$$

Selisih terbagi tingkat tiga ada 2 nilai, yaitu

$$\begin{aligned} f[x_0, x_1, x_2, x_3] &= \frac{f[x_1, x_2, x_3] - f[x_0, x_1, x_2]}{x_3 - x_0} = \frac{0.7487 - 0.6125}{0.6 - 0.0} = 0.2270 \\ f[x_1, x_2, x_3, x_4] &= \frac{f[x_2, x_3, x_4] - f[x_1, x_2, x_3]}{x_4 - x_1} = \frac{0.9137 - 0.7487}{0.8 - 0.2} = 0.2750. \end{aligned}$$

Akhirnya, selisih terbagi tingkat empat ada 1 nilai, yaitu

$$\begin{aligned} f[x_0, x_1, x_2, x_3, x_4] &= \frac{f[x_1, x_2, x_3, x_4] - f[x_0, x_1, x_2, x_3]}{x_4 - x_0} \\ &= \frac{0.2750 - 0.2270}{0.8 - 0.0} = 0.0600. \end{aligned}$$

Selanjutnya, angka-angka selisih terbagi ini disusun dalam Tabel berikut.

$x_i$	Tingkat nol	Tingkat satu	Tingkat dua	Tingkat tiga	Tingkat empat
0.0	<b>1.0000</b>				
0.2	1.2214	<b>1.1070</b>			
0.4	1.4918	1.3520	<b>0.6125</b>		
0.6	1.8221	1.6515	0.7487	<b>0.2270</b>	
0.8	2.2255	2.0170	0.9137	0.2750	<b>0.0600</b>

Proses penghitungan nilai selisih terbagi lebih cepat menggunakan tabel langsung, yaitu dengan cara menggunakan data pada kolom sebelumnya. Diperhatikan angka 1.1070 diperoleh dari

$$\frac{1.2214 - 1.0000}{0.2 - 0.0}.$$

Juga angka 0.7487 diperoleh dari

$$\frac{1.6515 - 1.3520}{0.6 - 0.2},$$

dan seterusnya. Dilihat dari formulasi (3.3.4) maka hanya bilangan pertama pada tiap-tiap kolom saja (cetak tebal) yang berkontribusi sebagai koefisien pada polinomial interpolasi, yaitu

$$\begin{aligned} P_4(x) &= 1 + 1.1070(x - 0) + 0.6125(x - 0)(x - 0.2) + 0.2270(x - 0)(x - 0.2) \\ &\quad (x - 0.4) + 0.0600(x - 0)(x - 0.2)(x - 0.4)(x - 0.6). \end{aligned}$$

Aproksimasi  $f(0.75)$  diperoleh dengan memasukkan nilai  $x = 0.75$  ke dalam  $P_4(x)$  sehingga diperoleh  $f(0.75) \approx P_4(0.75) = 2.1152$ . ■

```
>> xdata=0:0.2:0.8;
>> ydata=[1 1.2214 1.4918 1.8221 2.2255];
>> y = poli_inter2(0.75,xdata,ydata)
y =
    2.1170
```

Kita telah membahas dua metode untuk menentukan polinomial interpolasi yaitu metode Lagrange dan metode selisih terbagi. Hasil yang diperoleh melalui kedua metode ini sama mengingat sifat ketunggalan polinomial interpolasi. Untuk responsi di bawah ini, Anda boleh menggunakan salah satu dari metode tersebut. Juga, penggunaan metode selisih terbagi boleh dengan formula rekursif ataupun tabel selisih terbagi.

**Masalah Responsi 3.4.1.** Diberikan data seperti pada tabel berikut.

$x_i$	0.1	0.2	0.3	0.4
$y_i = f(x_i)$	-0.6205	-0.2840	0.0066	0.2484

Data ini berasal dari fungsi  $f(x) = x \cos x - 2x^2 + 3x - 1$ .

- Gunakan polinomial derajat 1, 2, dan 3 untuk mengaproksimasi nilai  $f(0.25)$ . Hitunglah galat multaknya. Apakah dapat disimpulkan semakin tinggi derajat polinomial interpolasinya maka aproksimasinya semakin baik?
- Gambarkan grafik polinomial  $y = P_3(x)$  interpolasi dan fungsi  $y = f(x)$  pada interval  $[0, 1]$ . Amati perbedaannya. Di titik mana kedua fungsi ini paling berbeda.
- Perluas domain untuk kedua fungsi menjadi  $[-2, 2]$ , gambarkan grafik kedua fungsi. Fenomena apa yang Anda temukan terkait dengan galat aproksimasi?
- Ambilah xdata sebagai partisi seragam  $x_0 := 0 < x_1 < x_2 < \dots < x_n = 10$ . Sebagai contoh untuk  $n = 10$  maka xdata adalah  $0, 1, 2, \dots, 10$ . Lakukan eksperimen untuk berbagai nilai  $n$  dan bandingkan polinomial interpolasi  $y = P_n(x)$  yang diperoleh dengan fungsi  $y = f(x)$ . Apakah ada indikasi bahwa semakin tinggi derajat polinomial maka semakin dekat pula fungsi dan interpolasinya?

**Masalah Responsi 3.4.2.** Diberikan fungsi  $f(x) = e^{-x^2}$ ,  $x \in [-6, 6]$ .

- Ambilah node seragam pada interval ini, yaitu  $-6, -5, -4, \dots, 0, 1, 2, \dots, 4, 5, 6$ . Temukan polinomial interpolasi pada node ini, berapa derajat polinomialnya? Gambarkan grafik fungsi  $f$  dan juga polinomial interpolasinya? Apakah polinomial ini cukup akurat sebagai aproksimasi fungsi  $f$ ?
- Secara intuitif, pilihlah node sebanyak node pada pertanyaan 1 namun memberikan hasil polinomial interpolasi yang lebih baik dalam arti lebih mirip dengan fungsi  $f$ .

**Masalah Responsi 3.4.3.** Pada panci presto digunakan sistem penguapan (steam), yaitu suhu ditingkatkan mengakibatkan tekanan juga meningkat. Berikut data hubungan antara temperatur (T) dalam derajat Fahrenheit dan tekanan (P) yang dihasilkan dalam *pound/inch*<sup>2</sup>.

T	220	230	240	250	260	270	280	290	300
P	17.19	20.78	24.97	29.82	35.42	41.85	49.18	57.53	66.98

node baru ( $z_k$ ) yang diperoleh dari node asli.

```
function D = hbagi_hermite(xdata,ydata,dydata)
%input: pasangan data (xdata,ydata) dan (xdata,dydata)
n=length(xdata);
zdata=zeros(1,2*n);
zdata(1:2:end-1)=xdata;
zdata(2:2:end)=xdata;
zydata=zeros(1,2*n);
zydata(1:2:end-1)=ydata;
zydata(2:2:end)=ydata;
D=zeros(2*n,2*n);
D(:,1)=zydata';
D(2:2:end,2)=dydata';
dy=ydata(2:end)-ydata(1:end-1);
dx=xdata(2:end)-xdata(1:end-1);
d=dy./dx;
D(3:2:end-1,2)=d';
for j=3:2*n
    v=D(j-1:end,j-1);
    a=v(2:end)-v(1:end-1);
    b=zdata(j:end)-zdata(1:end-j+1);b=b';
    d=a./b;
    D(j:end,j)=d;
end
```

Seperti biasa jangan lupa menyimpan file ini dengan nama `hbagi_hermite.m` sebelum menjalankannya untuk data pada contoh sebelumnya.

```
>>xdata=[0 3 5 8 13];
>>ydata=[0 67 115 186 280];
>>dydata=[22 23 24 22 21];
>> D = hbagi_hermite(xdata,ydata,dydata)
D =
    0         0         0         0         0         0         0         0         0         0
    0 22.0000         0         0         0         0         0         0         0         0
 67.0000 22.3333 0.1111         0         0         0         0         0         0         0
 67.0000 23.0000 0.2222 0.0370         0         0         0         0         0         0
115.0000 24.0000 0.5000 0.0556 0.0037         0         0         0         0         0
115.0000 24.0000         0 -0.2500 -0.0611 -0.0130         0         0         0         0
186.0000 23.6667 -0.1111 -0.0222 0.0456 0.0133 0.0033         0         0         0
186.0000 22.0000 -0.5556 -0.1481 -0.0252 -0.0141 -0.0034 -0.0008         0         0
280.0000 18.8000 -0.6400 -0.0106 0.0172 0.0042 0.0018 0.0004 0.0001         0
280.0000 21.0000 0.4400 0.2160 0.0283 0.0014 -0.0003 -0.0002 -0.0000 -0.0000
```

Hasil keluaran komputer ini menunjukkan kecocokan dengan beberapa perhitungan manual yang telah tersaji pada tabel sebelumnya. Perhatikan bahwa koefisien polinomial Hermite berada pada diagonal utama. Selanjutnya untuk mendefinisikan polinomial Hermite dibutuhkan satu lagi m-file Matlab berikut.

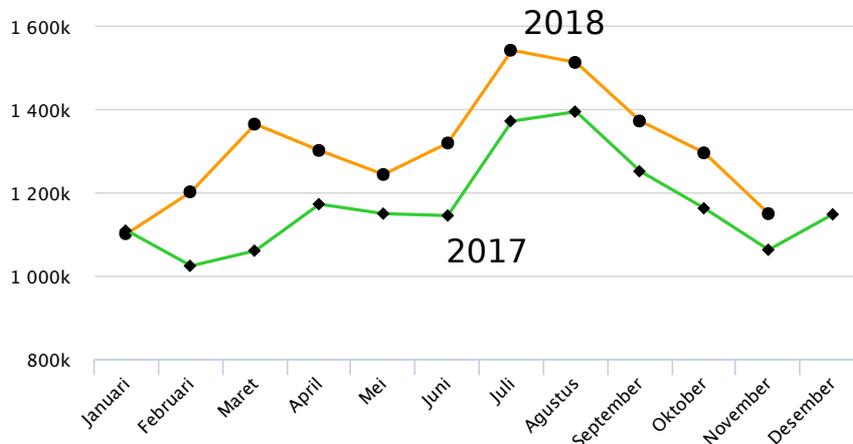
(a)  $x_0, x_1, x_2, x_3$ .(b)  $x_1, x_2, x_3, x_4$ .

20. Bangunlah polinomial interpolasi Hermite  $H_2$  untuk fungsi  $f(x) = \sqrt{1+x}$  dengan node  $x_0 = 0, x_1 = \pi$ . Gambarkan grafik fungsi galat  $e(x) = |H_2(x) - f(x)|$  dan juga  $e_1(x) := |H_2'(x) - f'(x)|$  pada interval  $[0, \pi]$ . Amati berapa dan di mana terjadi galat maksimum?
21. Diberikan fungsi  $f(x) = e^{-x^2}$ . Definisikan  $P_n(x)$  polinomial interpolasi untuk  $f(x)$  dengan menggunakan  $n+1$  titik node seragam pada interval  $[-1, 1]$ . Definisikan galat aproksimasi fungsi  $f$  oleh  $P_n$  sebagai berikut:

$$e_n = \max_{x \in I \subset [-1, 1]} |f(x) - P_n(x)|.$$

Lakukan eksperimen untuk  $n = 2, 4, 8, 16, 32$ , hitung  $e_n$  untuk  $I = \{-1 + \frac{2k}{500}; k = 0, 1, \dots, 500\}$ . Buatlah plot  $e_n$  versus  $n$ .

22. Berikut data diambil dari Biro Pusat Statistik (BPS) tentang jumlah kunjungan wisatawan mancanegara (dalam ribuan) dari Januari 2017 s.d. Desember 2017: 1107968, 1023388, 1059777, 1171386, 1144858, 1132568, 1370591, 1390343, 1250231, 1161565, 1062030, 1147031. Untuk kunjungan dari Januari 2018 s.d. November 2018 diperoleh data sebagai berikut: 1100677, 1201001, 1363339, 1300277, 1242588, 1318094, 1540549, 1511342, 1370842, 1294463, 1148644.



Gambar 3.14: Data kunjungan wisatawan mancanegara 2017-2018 (BPS)

Dalam hal ini kita memiliki dua kelompok data. Data tahun 2017 terdiri atas 12 pasangan sedangkan data tahun 2018 hanya ada 11 pasang.

- (a) Untuk masing-masing tahun, definisikan polinomial interpolasi dengan melibatkan semua node yang ada, kemudian gambarkan grafiknya. Bagaimana kelakuan grafik polinomial ini pada daerah di antara titik-titik node? Apakah cukup rasional digunakan untuk mengetahui data pada pertengahan bulan. Catat bahwa data yang tercantum dicatat pada akhir bulan.
- (b) Pilihlah 4 data tertentu agar polinomial interpolasi derajat tiga yang diperoleh memberikan hasil “terbaik” dalam arti memberikan jumlah galat kuadrat pada titik node lainnya. Lakukan untuk data tahun 2017 dan 2018.

- (c) Sebagaimana telah dijelaskan bahwa interpolasi digunakan untuk memprediksi data di antara data-data yang diketahui. Dalam hal ini data kunjungan pada bulan Desember 2018 tidak tersedia. Prediksi nilai data di luar rentang data yang ada disebut ekstrapolasi. Melalui polinomial yang diperoleh pada (b) untuk tahun 2017, prediksilah data pada bulan Desember 2018. Apakah hasilnya masuk akal?
- (d) Gunakan polinomial kuadrat terkecil derajat 3 menggunakan semua data. Gambarkan dan bandingkan hasilnya dengan polinomial interpolasi. Berilah ulasan. Gunakan polinomial kuadrat terkecil ini untuk memprediksi data pada bulan Desember 2018.

Untuk mengerjakan soal ini disarankan menggunakan Matlab dan m-file yang telah dibahas pada bagian-bagian sebelumnya.

## BAB 4

# Interpolasi Potongan Polinomial

*Dia mengatur urusan dari langit ke bumi, kemudian (urusan) itu naik kepada-Nya dalam satu hari yang kadarnya adalah seribu tahun menurut perhitunganmu.*

QS 32:5.

### 4.1 Pendahuluan

Kalau dicermati dengan saksama berdasarkan hasil eksperimen sebelumnya bahwa aproksimasi dengan menggunakan polinomial interpolasi hanya baik untuk interval yang cukup sempit. Pengambilan node yang lebih banyak sekaligus meningkatkan derajat polinomial interpolasi tidaklah menjamin kualitas aproksimasi semakin baik. Padahal semakin banyak node atau semakin tinggi derajat polinomial maka semakin tinggi tingkat kompleksitas komputasi dan resiko kesalahan pembulatan pun semakin besar. Oleh karena itu perlu dipertimbangkan interpolasi yang hanya bekerja pada domain yang tidak begitu lebar di mana derajat polinomialnya tidak terlalu besar, misalnya cukup derajat satu (linear), dua (kuadrat), atau tiga (kubik). Inilah ide yang melatarbelakangi metode interpolasi potongan polinomial (*piece-wise polynomial interpolation*).

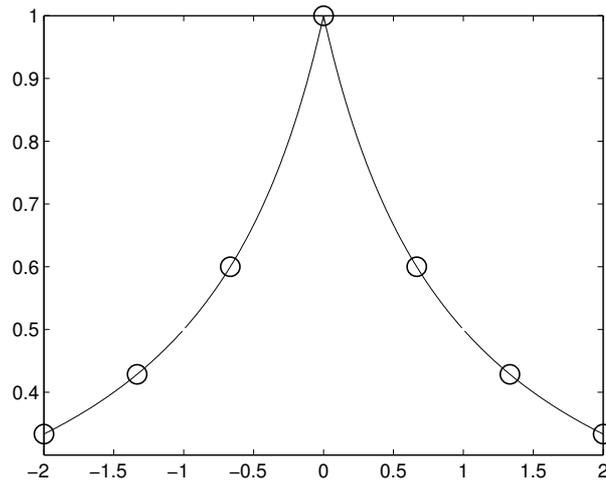
Misalkan kita akan mengaproksimasi fungsi dengan sebuah polinomial pada sebuah domain  $[a, b]$ . Domain ini dipecah menjadi beberapa subdomain yang lebih kecil  $[a_k, b_k]$ ,  $k = 1, 2, \dots, m$ . Masing-masing subdomain dibangun polinomial interpolasi derajat tertentu di mana titik temu antara dua subdomain berdekatan diambil sebagai node bersama. Biasanya digunakan derajat polinomial yang sama untuk setiap subdomain. Sebagai ilustrasi, perhatikan contoh berikut.

**Contoh 4.1.** Diberikan fungsi  $f : [-2, 2] \rightarrow \mathbb{R}$  yang didefinisikan sebagai berikut:

$$f(x) := \begin{cases} \frac{1}{1-x} & \text{jika } -2 \leq x \leq 0 \\ \frac{1}{1+x} & \text{jika } 0 < x \leq 2. \end{cases}$$

Kemudian diambil node seragam  $x_0 = -2, x_1 = -1.3333, x_2 = -0.6667, x_3 = 0, x_4 = 0.6667, x_5 = 1.3333, x_6 = 2$ . Aproksimasilah fungsi ini dengan potongan-potongan polinomial derajat 1 (linear), derajat 2 (kuadrat), dan derajat 3 (kubik) pada setiap subinterval yang terbentuk.

PENYELESAIAN. Grafik fungsi dan pasangan node yang ada ditampilkan pada Gambar 4.1.

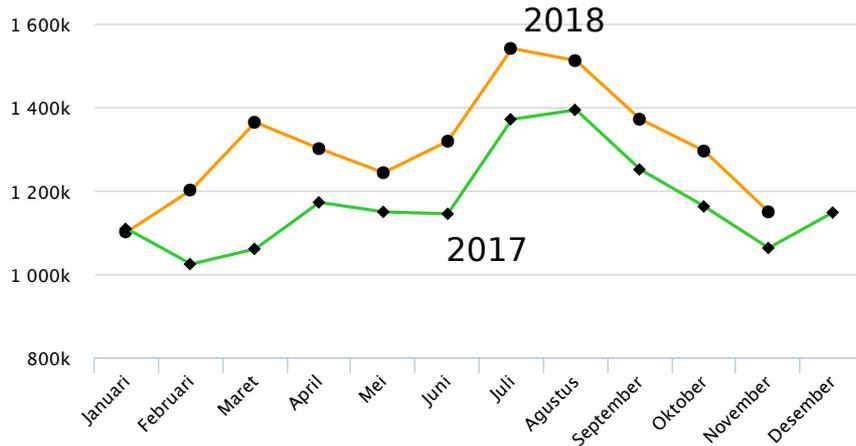
Gambar 4.1: Grafik fungsi  $f$  dan node yang diberikan

- Untuk interpolasi linear kita dapat membangun 6 potongan polinomial linear yang bersesuaian dengan node  $[x_0, x_1]$ ,  $[x_1, x_2]$ ,  $[x_2, x_3]$ ,  $[x_3, x_4]$ ,  $[x_4, x_5]$ , dan  $[x_5, x_6]$ .
- Untuk interpolasi kuadratik kita dapat membangun 3 potongan polinomial kuadratik yang bersesuaian dengan node  $[x_0, x_1, x_2]$ ,  $[x_2, x_3, x_4]$ , dan  $[x_4, x_5, x_6]$ .
- Untuk interpolasi polinomial kubik kita hanya dapat membangun sebuah polinomial kubik yaitu untuk node  $[x_0, x_1, x_2, x_3]$  dan  $[x_3, x_4, x_5, x_6]$ .

Setelah menggunakan m-file yang bersesuaian untuk mendefinisikan para polinomial ini, ketiga interpolasi ini ditunjukkan pada Gambar 4.2. Pada gambar ini diperlihatkan bagaimana fungsi  $f$  diaproksimasi oleh interpolasi potongan polinomial derajat 1 (kiri), derajat 2 (tengah), dan derajat 3 (kanan). Jika semua node digunakan akan terbentuk polinomial interpolasi derajat 6 seperti ditunjukkan pada Gambar 4.3. Berdasarkan visual ini jelas bahwa interpolasi derajat 6 sangat jelek untuk aproksimasi fungsi  $f$  dibandingkan interpolasi potongan polinomial derajat 1, 2, dan 3. Penggunaan interpolasi potongan polinomial biasa dikenal dengan istilah aproksimasi lokal karena domain  $[a, b]$  dipecah menjadi beberapa subdomain yang lebih kecil dan polinomial hanya berlaku secara lokal. Sedangkan penggunaan semua node biasa dikenal dengan istilah aproksimasi global karena satu polinomial berlaku untuk keseluruhan domain.

## 4.2 Konstruksi interpolasi potongan polinomial

Misalkan sebuah fungsi  $f$  akan diaproksimasi dengan potongan-potongan polinomial (*piece-wise polynomials*) pada domain  $[a, b]$  maka perlu ditetapkan derajat polinomial dan banyak polinomial yang akan digunakan. Misalkan kita akan menggunakan  $m$  buah polinomial berderajat  $d$  maka kita perlu menghitung banyak node yang dibutuhkan. Perhatikan bahwa untuk membangun sebuah polinomial berderajat  $d$  dibutuhkan  $d + 1$  titik. Karena titik ke-  $(d + 1)$  menjadi titik pertama untuk polinomial berikutnya maka untuk membangun dua polinomial berderajat  $d$  dibutuhkan  $2(d + 1) - 1 = 2d + 1$  titik. Secara



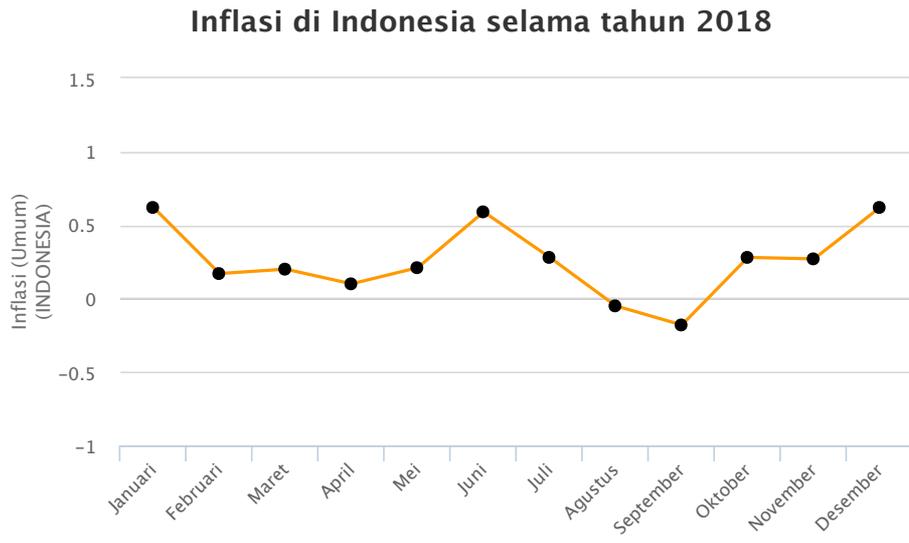
Gambar 4.21: Data kunjungan wisatawan mancanegara 2017-2018 (BPS)

Dalam hal ini kita memiliki dua kelompok data. Data tahun 2017 terdiri atas 12 pasangan sedangkan data tahun 2018 hanya ada 11 pasang.

- Aproksimasilah data ini dengan menggunakan interpolasi B-spline derajat 3.
- Gunakan hasil ini untuk memprediksi kunjungan wisata pada setiap pertengahan bulan.
- Gunakan juga untuk memprediksi jumlah kunjungan pada bulan Desember 2018. Apakah penggunaan polinomial sepotong-sepotong ini lebih cocok daripada interpolasi global seperti dibahas pada bab sebelumnya?
- Fokus pada data 2017, misalkan  $t_k$  menyatakan bulan ke- $k$ ,  $k = 1, 2, \dots, 12$ . Andai hanya tersedia 7 data pada  $t_1, t_3, t_5, t_6, t_7, t_9$ , dan  $t_{11}$  sedangkan data lainnya dianggap tidak ada atau hilang. Gunakan interpolasi B-spline derajat 2 menggunakan data yang tersedia tersebut. Gunakan polinomial ini untuk memprediksi/mengaproksimasi data yang hilang termasuk data ekstrapolasi pada bulan Desember. Seandainya data yang hilang tersebut ketemu kembali persis seperti grafik di atas, hitunglah galat aproksimasinya.

Untuk mengerjakan soal ini disarankan menggunakan Matlab dan m-file yang telah dibahas pada bagian-bagian sebelumnya.

- Berikut data inflasi di Indonesia selama tahun 2018. Nilai numerik data tersebut adalah 0.51, 0.17, 0.20, 0.25, 0.59, 0.28, -0.05, -0.18, 0.28, 0.27, 0.62. Lakukan eksperimen numerik untuk memilih sedikit mungkin node namun menghasilkan polinomial interpolasi terbaik untuk data tersebut. Terbaik dalam arti memberikan galat terkecil untuk data lainnya. Dalam hal ini cukup diambil maksimal 7 data. Pilihlah 7 data untuk membangun polinomial interpolasi B-spline order 2 yang memberikan aproksimasi terbaik. Bandingkan dengan hasil yang diperoleh sebelumnya.



Gambar 4.22: Data inflasi di Indonesia tahun 2018

## BAB 5

# Aproksimasi Derivatif dan Integral

*Karena sesungguhnya sesudah kesulitan itu ada kemudahan, sesungguhnya sesudah kesulitan itu ada kemudahan. QS 94:5-6.*

## 5.1 Pendahuluan

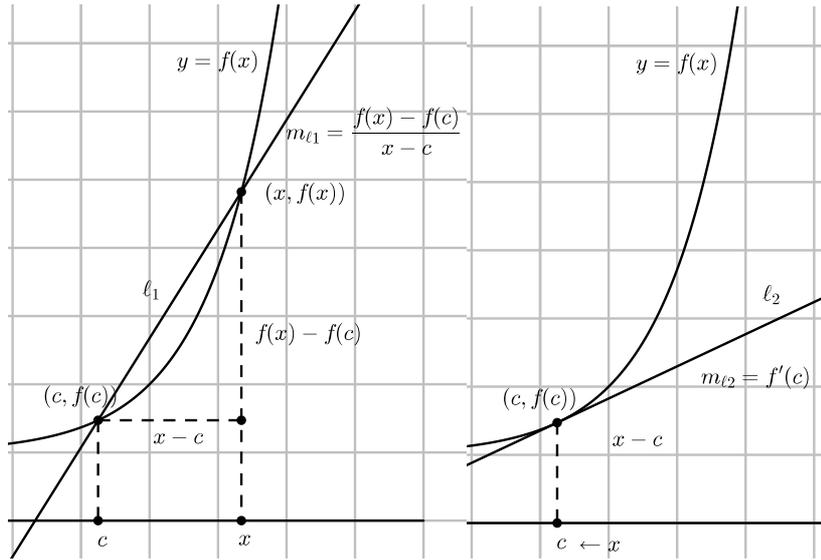
Berdasarkan tampilannya, fungsi dapat dikelompokkan menjadi dua macam yaitu fungsi dengan formula eksplisit, seperti  $f(x) = \sin x$ ,  $g(x) = x^2 + 2x + 4$ ,  $h(x) = x \ln x + e^x$  dan fungsi tabular. Fungsi tabular berupa pasangan data nilai variabel bebas dan variabel terikatnya yang disajikan ke dalam bentuk tabel. Dalam dunia terapan, nilai-nilai ini biasanya diperoleh dari hasil pengukuran variabel terikat berdasarkan runtun waktu. Sebagai contoh, fungsi jarak  $s(t)$  terhadap variabel waktu  $t$  berikut tidak menampilkan formula eksplisit, namun hanya berupa tabel hasil pengukuran untuk beberapa nilai  $t$  tertentu.

$t$	1	2	3	4	5	6	7	8	9	10
$s(t)$	30	40	65	75	100	120	155	180	200	215

Tabel 5.1: Jarak tempuh yang diukur pada saat-saat tertentu

Fungsi seperti ini pada kenyataannya banyak dijumpai pada bidang terapan. Tidak mungkin semua nilai fungsi dapat diketahui melalui alat pengukuran apapun karena domain fungsi biasanya berupa himpunan takberhingga. Misalnya data kecepatan angin, temperatur dan kelembaban udara hanya dicatat setiap jam, tekanan darah dicatat/dicek setiap hari, kenaikan suhu bumi atau kenaikan air laut dicatat per tahun, dan lain sebagainya. Dalam hal formula eksplisit suatu fungsi tidak diketahui maka pengintegralan dan pendiferensialan fungsi ini tidak dapat dilakukan menggunakan rumus yang biasa dipelajari pada kalkulus. Sebagai ilustrasi  $\int_1^{10} s(t)dt$  dan  $s'(5)$  tidak dapat dihitung menggunakan rumus apapun. Untuk mengatasi permasalahan ini diperlukan pendekatan secara numerik.

Strategi paling sederhana dalam melakukan pendekatan numerik adalah dengan cara membawa fungsi tersebut ke dalam bentuk polinomial, misalnya polinomial interpolasi. Kemudian, perhitungan integral dan derivatif untuk fungsi tersebut dikerjakan pada polinomial aproksimasi ini. Hasilnya digunakan sebagai nilai aproksimasi untuk integral atau derivatif fungsi semula. Proses ini disebut pemulusan (*smoothing*) fungsi. Dipilihnya polinomial sebagai fungsi interpolan karena perhitungan derivatif dan integral pada polinomial sangat mudah dilakukan. Bab ini membahas berbagai metode untuk men-



Gambar 5.1: Ilustrasi derivatif secara grafis

gaproksimasi nilai integral tertentu fungsi pada sebuah interval dan derivatif fungsi di titik tertentu. Pembahasannya mencakup penurunan rumusnya, analisis dan implementasi numeriknya.

## 5.2 Aproksimasi Derivatif

Sebelum diturunkan berbagai formula untuk aproksimasi derivatif, perlu dipahami dulu konsep dasar dari derivatif itu sendiri. Sebuah garis lurus dikarakteristisasi oleh gradien atau slope. Gradien merupakan ukuran kemiringan garis, semakin besar gradien semakin tegaklah garis tersebut. Bila garis dibayangkan sebagai seluncuran maka semakin tegak semakin cepat meluncurnya sebuah benda. Bila  $y = \ell(x) = mx + a$  merupakan persamaan garis lurus dan  $\ell(x)$  sebagai nilai fungsi di titik  $x$ , maka slope  $m$  menginformasikan kepada kita tentang seberapa cepat nilai fungsi berubah di sebuah titik. Semakin besar slope semakin cepat fungsi berubah di titik tersebut. Tanda (+) atau (-) pada  $m$  menandakan arah perubahan, bertambah atau berkurang.

Bila dihubungkan dengan kurva lengkung seperti diberikan pada Gambar 5.1 (kiri), maka  $\frac{f(x)-f(c)}{x-c}$  tidak lain adalah gradien tali busur (disebut juga garis secant) yang melalui titik  $(c, f(c))$  dan  $(x, f(x))$ . Ketika titik  $x$  digerakkan terus mendekati  $c$  maka pada akhirnya diperoleh garis singgung kurva (disebut juga garis tangent)  $y = f(x)$  di titik  $c$  seperti terlihat pada Gambar 5.1 (kanan). Nah, gradien garis singgung kurva  $y = f(x)$  di titik  $c$  inilah yang disebut derivatif  $f$  di  $x = c$ , ditulis  $f'(c)$ . Inilah konsep dasar derivatif. Kalau misalnya  $x$  menyatakan waktu tempuh sebuah benda dan  $f(x)$  jarak tempuhnya maka  $\frac{f(x)-f(c)}{x-c}$  dapat diinterpretasikan sebagai kecepatan rerata benda tersebut dalam rentang  $[c, x]$ . Jadi  $f'(c)$  merupakan kecepatan benda tersebut di saat  $c$ . Definisi formal derivatif diberikan sebagai berikut.

**Definisi 5.1.** Misalkan  $I \subset \mathbb{R}$  suatu interval, dan  $f : I \rightarrow \mathbb{R}$ ,  $c \in \mathbb{R}$ . Bilangan real  $L \in \mathbb{R}$  dikatakan

## 5.4 Aproksimasi Integral Tertentu

Aproksimasi numerik suatu integral tertentu dikenal dengan istilah **kuadratur numerik** (*numerical quadrature*). Istilah ini berasal dari metode orang Yunani kuno ketika menghitung luas daerah di dalam sebuah kurva. Contoh yang sangat terkenal adalah masalah penguadratan lingkaran, yaitu menentukan sebuah persegi yang mempunyai luas sama dengan lingkaran yang diberikan. Kuadratur juga dipahami sebagai metode aproksimasi integral dengan menggunakan jumlahan terhingga.

Pada bagian ini kita akan menghitung nilai numerik sebagai aproksimasi integral tertentu berikut

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

dengan  $f$  diasumsikan kontinu dan cukup mulus. Asumsi ini sesungguhnya tidak perlu terlalu ketat sebab keterintegralan suatu fungsi tidak harus kontinu dan mulus. Artinya, fungsi kontinu sepotong-sepotong pun sudah dapat terintegral Riemann. Namun asumsi ini dibutuhkan dalam rangka menurunkan formula galat aproksimasi integral. Berdasarkan teorema fundamental kalkulus (TFK) yang telah dibahas sebelumnya, integral ini dapat diselesaikan dengan mudah dan eksak bilamana kita dapat menemukan antiderivatif  $f$  yaitu  $F$  sehingga  $F'(x) = f(x)$  sehingga diperoleh

$$\int_a^b f(x) dx = F(b) - F(a).$$

Permasalahannya adalah ketika fungsi  $f$  tidak mempunyai antiderivatif, atau kalaupun antiderivatifnya ada maka bentuknya tidak tersajikan dalam bentuk eksplisit. Integral seperti di bawah ini

$$\int_0^2 e^{x^2} dx, \int_0^{5\pi} \sqrt{1 + \cos^2 x} dx, \int_0^1 \int_0^1 \sin(xye^x) dx dy$$

adalah beberapa contoh integral yang tidak dapat diselesaikan dengan menggunakan TFK.

Berbagai definisi integral yang ada dibangun untuk memberikan justifikasi teoritis terhadap berbagai permasalahan yang muncul dalam matematika ataupun dalam ilmu terapan. Namun dalam realisasi numeriknya, integral tersebut tetap harus dibawa ke bentuk jumlahan Riemann sebagai aproksimasinya. Walaupun secara teoretis, integral merupakan proses takhingga namun dalam praktiknya perhitungan yang melibatkan proses takhingga tidak mungkin dilakukan, termasuk oleh komputer. Jadi sangatlah masuk akal bila jumlahan takhingga ini diaproksimasi oleh jumlahan terhingga. Bentuk jumlahan hingga dibentuk dengan cara mengambil nilai integran  $f$  pada beberapa titik di dalam  $[a, b]$  yang banyaknya berhingga. Cara ini dikenal dengan nama aturan kuadratur. Formula kuadratur  $n + 1$  titik diberikan oleh

$$I(f) = \int_a^b f(x) dx = \sum_{i=0}^n w_i f(x_i) + R_n. \quad (5.4.1)$$

Titik-titik  $\{x_i : i = 0, 1, \dots, n\}$  di mana fungsi  $f$  dievaluasi disebut absis atau node, sedangkan koefisien  $\{w_i : i = 0, 1, \dots, n\}$  disebut bobot, dan  $R_n$  disebut sisa atau residu. Kemudian, integral tertentu  $I(f) = \int_a^b f(x) dx$  diaproksimasi dengan aturan atau formula kuadratur yang diberikan oleh

$$I(f) \approx Q(f) := \sum_{i=0}^n w_i f(x_i). \quad (5.4.2)$$

$a$  dan  $b$  diambil sebagai node, sedangkan midpoint tidak termasuk metode Newton-Cotes karena cuma memuat satu node. Selain menggunakan interpolasi, tiga formula kuadratur ini dapat pula diturunkan melalui metode koefisien taktentu (*undetermined coefficient*) sebagai berikut.

### 5.4.2 Metode koefisien taktentu

Metode ini didasarkan pada asumsi bahwa formula kuadratur memberikan hasil eksak untuk semua polinomial derajat  $n$ , yaitu

$$\sum_{k=0}^n w_k f(x_k) = \int_a^b f(x) dx$$

dengan  $f$  adalah polinomial berderajat paling tinggi  $n$ . Untuk memenuhi kondisi ini cukup dipenuhi oleh setiap monomial  $1, x, x^2, \dots, x^n$ . Dalam penjabarannya, cukup diambil interval  $[-1, 1]$ , selanjutnya digunakan transformasi variabel pada domain lebih umum  $[a, b]$ . Untuk midpoint bersesuaian dengan  $n = 0$  dan  $x_0 = \frac{a+b}{2}$ , tidak perlu dijabarkan karena sangat sederhana. Untuk trapesium bersesuaian dengan  $n = 1$  dengan  $x_0 = -1$  dan  $x_1 = 1$ . Jadi dua monomial  $P_0(x) = 1$  dan  $P_1(x) = x$  harus dipenuhi secara eksak formula kuadratur, yaitu untuk  $f(x) = P_0(x) = 1$  diperoleh:

$$\begin{aligned} w_0 P_0(-1) + w_1 P_0(1) &= \int_{-1}^1 1 dx \\ w_0 + w_1 &= 2. \end{aligned}$$

dan juga untuk  $f(x) = P_1(x) = x$  diperoleh:

$$\begin{aligned} w_0 P_1(-1) + w_1 P_1(1) &= \int_{-1}^1 x dx \\ -w_0 + w_1 &= 0. \end{aligned}$$

Setelah diselesaikan diperoleh  $w_0 = w_1 = 1$  sehingga diperoleh formula berikut:

$$\int_{-1}^1 f(x) dx \approx f(-1) + f(1).$$

Dengan transformasi variabel baru  $t := \frac{2x-a-b}{b-a}$  maka diperoleh  $x = \frac{1}{2}((b-a)t+a+b)$  dan  $dx = \frac{1}{2}(b-a)dt$ . Untuk  $x = a$  maka  $t = -1$  dan untuk  $x = b$  maka  $t = 1$ . Sehingga diperoleh aturan trapesium pada domain  $[a, b]$  sebagai berikut

$$\begin{aligned} \int_a^b f(x) dx &= \frac{b-a}{2} \int_{-1}^1 f\left(\frac{(b-a)t+a+b}{2}\right) dt \\ &\approx \frac{b-a}{2} f\left(\frac{(b-a)(-1)+a+b}{2}\right) + f\left(\frac{(b-a)(1)+a+b}{2}\right) \\ T(f) &= \frac{b-a}{2} (f(a) + f(b)). \end{aligned}$$

Hasil ini persis sama dengan pendekatan interpolasi sebelumnya. Untuk Simpson berkaitan dengan  $n = 2$  dengan  $x_0 = -1, x_1 = 0$  dan  $x_2 = 1$ . Tiga monomial yang terlibat adalah  $P_0(x) = 1, P_1(x) = x$  dan  $P_2(x) = x^2$ . Diperoleh sistem persamaan linear sebagai berikut:

### 5.4.3 Estimasi galat metode kuadratur dasar

Secara intuitif maupun secara grafis bahwa semakin sempit interval integrasi  $[a, b]$  maka hasil aproksimasinya semakin akurat, ini berlaku pada setiap metode. Selanjutnya kita perlu memahami pengaruh lebar interval integrasi ini terhadap galat aproksimasi masing-masing metode. Secara eksplisit, kita perlu mengetahui bagaimana dampaknya pada galat aproksimasi jika lebar interval ini diperkecil menjadi setengahnya, seperempatnya, dan seterusnya. Pembuktian teorema berikut membutuhkan teorema rerata integral, yaitu bila  $f, g : [a, b] \rightarrow \mathbb{R}$  kontinu pada  $[a, b]$  dan  $g(x) \geq 0$  pada  $[a, b]$  maka ada  $c \in [a, b]$  sehingga

$$\int_a^b f(x)g(x) dx = f(c) \int_a^b g(x) dx. \quad (5.4.9)$$

Syarat  $g(x) \geq 0$  dapat pula diganti  $g(x) \leq 0$  pada  $[a, b]$ . Intinya  $g(x)$  tidak berubah tanda pada  $[a, b]$ .

**Teorema 5.2.** *Bila  $f \in C^2[a, b]$  maka terdapat  $\xi \in (a, b)$  sehingga metode midpoint memberikan galat sebagai berikut*

$$I(f) - M(f) := \int_a^b f(x) dx - (b-a)f\left(\frac{a+b}{2}\right) = \frac{f''(\xi)}{24}(b-a)^3. \quad (5.4.10)$$

BUKTI. Terapkan teorema Taylor di sekitar  $x_0 = \frac{a+b}{2}$ , yaitu untuk setiap  $x \in [a, b]$  terdapat  $\xi(x) \in (a, b)$  sehingga

$$\begin{aligned} f(x) &= f(x_0) + f'(x_0)(x - x_0) + \frac{f''(\xi(x))}{2}(x - x_0)^2. \\ f(x) - f(x_0) &= f'(x_0)(x - x_0) + \frac{f''(\xi(x))}{2}(x - x_0)^2. \end{aligned}$$

Kedua ruas diintegrasikan dari  $a$  ke  $b$  diperoleh

$$\begin{aligned} I(f) - M(f) &= \int_a^b (f(x) - f(x_0)) dx \\ &= \underbrace{\int_a^b f'(x_0)(x - x_0) dx}_{=0} + \int_a^b \frac{f''(\xi(x))}{2}(x - x_0)^2 dx \\ &= \frac{f''(\xi)}{2} \int_a^b (x - x_0)^2 dx \\ &= \frac{f''(\xi)}{6} [(x - x_0)^3]_a^b \\ &= \frac{f''(\xi)}{6} \left[ \left(\frac{b-a}{2}\right)^3 - \left(\frac{a-b}{2}\right)^3 \right] = \frac{f''(\xi)}{24}(b-a)^3. \end{aligned}$$

Nilai nol integral pada langkah kedua penjabaran di atas dapat diverifikasi dengan mudah dikarenakan  $\int_a^b (x - x_0) dx = 0$ .  $\int_a^b \frac{f''(\xi(x))}{2}(x - x_0)^2 dx = \frac{f''(\xi)}{2} \int_a^b (x - x_0)^2 dx$  diperoleh dengan menerapkan teorema nilai rerata integral yaitu dengan mengambil  $g(x) := (x - x_0)^2 \geq 0$  dan  $f_1(x) := \frac{f''(\xi(x))}{2}$ . Pengambilan  $f_1$  dimaksudkan untuk menghindari tumpang tindih dengan simbol fungsi  $f$  yang sudah ada. Dengan demikian terbukti galat estimasi seperti pada teorema di atas.  $\square$

Interval	Eksak	Midpoint		Trapeسيوم		Simpson	
		Aprok	galat	Aprok	galat	Aprok	galat
[0, 0.25]	0.0322	0.0317	0.0005	0.0333	0.0010	0.0322	0.0000
[0, 0.5]	0.1420	0.1331	0.0090	0.1605	0.0185	0.1422	0.0002
[0, 1]	0.8591	0.6420	0.2171	1.3591	0.5000	0.8811	0.0219
[0, 2]	26.7991	5.4366	21.3625	109.1963	82.3972	40.0231	13.2241
[0, 2.2]	62.7347	8.1154	54.6192	306.0558	243.3212	107.4289	44.6942

kuadratur dasar hanya cocok jika jarak antara kedua batas cukup sempit. Bahkan metode Simpson pun memberikan galat yang sangat besar ketika diterapkan pada interval [0, 2] dan setelahnya. Oleh karena itu aturan kuadratur dasar ini jarang digunakan pada terapan dan sebagai gantinya digunakan aturan kuadratur bersusun.

Untuk memperoleh angka-angka di dalam tabel secara otomatis kita dapat memodifikasi m-file di atas dengan keluaran berupa vektor

```
v=[eksak, Mid, Err_Mid, Trap, Err_Trap, Simp, Err_Simp]
```

sebagai berikut:

```
function v=eksp_kdasar(a,b)
% input: batas integrasi a dan b
% output: vektor v seperti di atas
M = (b-a)*feval('fun', (a+b)/2);
T = (b-a)/2*(feval('fun',a)+feval('fun',b));
S = (b-a)/6*(feval('fun',a)+4*feval('fun', (a+b)/2)+feval('fun',b));
eksak = feval('fun1',b)-feval('fun1',a);
v = [eksak M abs(M-eksak) T abs(T-eksak) S abs(S-eksak)];
%fungsi yang diintegrasikan
function y = fun(x)
y = x*exp(x^2);
%fungsi hasil integral
function y = fun1(x)
y = 0.5*exp(x^2);
```

Tentunya m-file ini hanya akan menghasilkan baris per baris berdasarkan batas integral  $a$  dan  $b$  yang diberikan. Sebagai contoh bila diberikan perintah pada command window

```
>>v=eksp_kdasar(0,1)
```

maka akan muncul keluaran 0.8591 0.6420 0.2171 1.3591 0.5000 0.8811 0.0219.

### 5.5.2 Formula kuadratur bersusun

Dalam bagian ini kita mempunyai tiga formula yang telah ditemukan sebelumnya, yaitu metode midpoint bersusun, trapesium bersusun, dan Simpson bersusun. Untuk implementasi Matlab ini kita gunakan formula berikut.

- Midpoint:  $M(f) = 2h \sum_{k=1}^{n/2} f(a + (2k - 1)h)$ .

dibangun formula iterasi yang dapat meningkatkan order aproksimasi. Metode ini tidak sempat dibahas pada bab ini, namun pembaca yang tertarik mempelajari silakan baca pada buku metode atau analisis numerik lanjutan (Faires & Burden, 2003; Hernadi, 2015).

Aproksimasi integral dengan formula kuadratur yang telah dibahas adalah midpoint, trapesium, dan Simpson baik dalam bentuk dasar (*basic quadrature*) maupun dalam bentuk bersusun (*composite quadrature*). Formula ini didasarkan pada polinomial interpolasi, semakin tinggi derajat polinomial semakin tinggi order konvergensinya. Metode Simpson lebih baik dari midpoint dan trapesium. Formula kuadratur yang didasarkan pada derajat akurasi adalah metode integral Gauss. Metode Gauss ini lebih baik daripada metode Simpson dari aspek derajat akurasi. Metode integral Romberg yang mirip metode ekstrapolasi Richardson pada aproksimasi derivatif memberikan hasil lebih baik dari Gauss, namun tidak sempat dibahas pada bab ini. Pembaca yang tertarik mempelajarinya dapat membacanya pada buku penulis lainnya (Hernadi, 2015). Komputasi numerik pada formula aproksimasi integral tidak begitu sensitif terhadap galat pembulatan komputer seperti halnya pada aproksimasi derivatif.

Selain bertujuan memperoleh aproksimasi derivatif dan integral, tujuan lain yang tidak kalah pentingnya adalah estimasi galat. Melalui formula estimasi galat kita tahu pola kekonvergenan barisan aproksimasi dan dapat mengukur tingkat akurasi yang diperoleh.

## 5.6 Soal-soal Latihan Bab 5

1. Mengapa kita perlu mengembangkan formula aproksimasi untuk derivatif dan juga untuk integral padahal rumus-rumus derivatif dan integral fungsi sudah tersedia pada pelajaran kalkulus?
2. Apakah setiap fungsi yang mempunyai rumus eksplisit dapat didiferensialkan? Apakah juga dapat diintegrasikan?
3. Apa yang dimaksud dengan estimasi galat dan apa manfaatnya dalam aproksimasi derivatif ataupun integral?
4. Berdasarkan eksperimen numerik, metode Simpson lebih unggul daripada metode midpoint dan trapesium. Dapatkah Anda memberi contoh kasus di mana Simpson tidak lebih baik daripada trapesium ataupun midpoint?
5. Diberikan fungsi  $f(x) := x \cos x - x^2 \sin x$  yang dievaluasi pada  $x = 1.1, 1.2, 1.3$ , dan  $1.4$ . Hitunglah aproksimasi  $f'(x)$  menggunakan formula selisih terbagi dua titik atau tiga titik tergantung pada node yang ada. Hitunglah galat nyata masing-masing formula. Formula aproksimasi mana yang paling baik?
6. Pada kalkulus sudah dikenal definisi derivatif, yaitu  $f'(x) := \lim_{h \rightarrow 0} \frac{f(x+h) - f(x)}{h}$ . Selanjutnya definisikan barisan fungsi ( $f'_n$ ) dengan  $f'_n(x) := \frac{f(x+10^{-n}) - f(x)}{10^{-n}}$ . Jelas bahwa  $\lim_{n \rightarrow \infty} f'_n(x) = f'(x)$ . Lakukan eksperimen numerik untuk  $n = 1, 2, 3, \dots, 20$  pada fungsi  $f$  dan  $x$  sebagai berikut:
  - (a)  $f(x) = e^x$ ,  $x = 1$ ;
  - (b)  $f(x) = x^2 e^{-x}$ ,  $x = 0.5$ ;
  - (c)  $f(x) = 1/(1+x^2)$ ,  $x = 2$ ;
  - (d)  $f(x) = \sqrt{1-x^2}$ ,  $x = 0.25$ .

14. Untuk integral yang sama seperti soal sebelumnya, gunakan kuadratur bersusun mesh seragam untuk beberapa metode berikut:

- (a) Simpson dengan  $n = 4$ ;
- (b) Gauss dengan  $n = 3$ ;
- (c) Gauss dengan  $n = 4$ .

Metode mana yang paling akurat?

15. Diberikan integral sebagai berikut:

$$\int_0^1 \frac{1}{1+x^3} dx.$$

Nilai eksak integral ini adalah  $\frac{1}{3} \ln 2 + \frac{\pi}{9} \sqrt{3}$ .

- (a) Hitunglah nilai aproksimasi metode Simpson bersusun dengan  $h = 0.25$ ;
- (b) Hitunglah estimasi galatnya;
- (c) Hitunglah galat nyatanya. Apakah konsisten dengan estimasi?

16. Terapkan metode integral Gauss bersusun order 2 pada soal dan pertanyaan yang sama seperti soal no 15.

17. Melalui lingkaran satuan (yakni, jari-jarinya 1), buktikan bahwa

$$\pi = 2 \int_{-1}^1 \sqrt{1-x^2} dx.$$

Gunakan metode Simpson untuk menghitung nilai  $\pi$  ini dengan akurasi 6 digit. Banyak partisi dapat dilakukan coba-coba dimulai dari  $n = 2, 4, 6, 8 \dots$  dan berhentilah ketika galatnya tidak melebihi 0.0000001.

18. Untuk pertanyaan yang sama seperti soal no 17, terapkan metode Gauss bersusun order 2. Apakah dibutuhkan jumlah partisi yang sama seperti metode Simpson?

19. Temukan absis dan bobot pada formula integral Gauss order  $n = 3, 4, 5$  dengan menggunakan polinomial Legendre. Kemudian bandingkan hasilnya dengan angka-angka yang ada pada Tabel 5.2.

20. Tabel berikut menyajikan power yang disuplai kepada roda mobil yang sedang berjalan, bergantung pada kecepatan  $v$ .

$v(\text{m/s})$	0	1.0	1.8	2.4	3.5	4.4	5.1	6.0
$P(\text{kW})$	0	4.7	12.2	19.0	31.8	40.1	43.8	43.2

Jika massa mobil adalah  $m = 2000$  kg, tentukan waktu  $\Delta t$  yang dibutuhkan mobil untuk berakselerasi dari 1 meter per detik menjadi 6 meter per detik. Petunjuk: dengan menggunakan hukum Newton  $F = m \left( \frac{dv}{dt} \right)$  dan definisi power  $P = Fv$  maka diperoleh

$$\Delta t = m \int_{1s}^{6s} (v/P) dv$$

## BAB 6

# Sistem Persamaan Linear

*Dialah yang menjadikan bumi sebagai hamparan bagimu dan langit sebagai atap, dan Dia menurunkan air (hujan) dari langit, lalu Dia menghasilkan dengan hujan itu segala buah-buahan sebagai rejeki untukmu; karena itu janganlah kamu mengadakan sekutu-sekutu bagi Allah, padahal kamu mengetahui.*

QS 2:22.

## 6.1 Pendahuluan

Sistem persamaan linear (SPL) merupakan salah satu pokok bahasan matematika yang sangat penting karena sering digunakan untuk menyelesaikan masalah dalam kehidupan nyata mulai dari masalah yang sangat sederhana sampai masalah yang sangat rumit. Oleh karena itulah materi ini sudah diperkenalkan sejak SMP dan SMA yaitu berupa sistem persamaan linear dua variabel (SPLDV) dan tiga variabel (SPLTV). Tiga metode yang sudah diperkenalkan kepada siswa SMP dan SMA adalah metode eliminasi, substitusi, dan campuran.

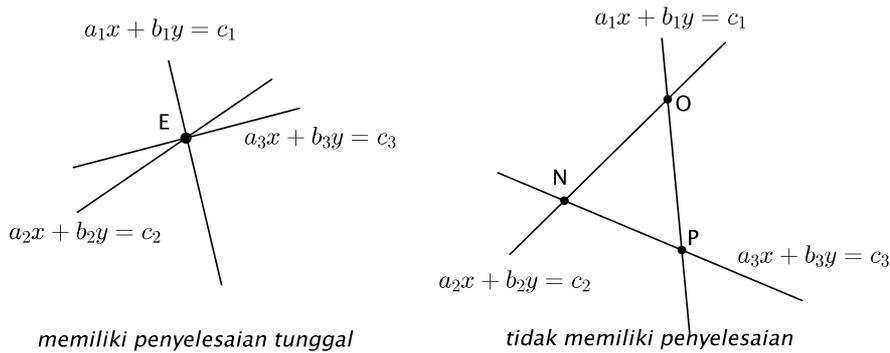
Bentuk umum sistem persamaan linear dengan  $m$  persamaan dan  $n$  variabel dapat disajikan sebagai berikut:

$$\begin{aligned}a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n &= b_1 \\a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n &= b_2 \\&\vdots \\a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n &= b_m.\end{aligned}$$

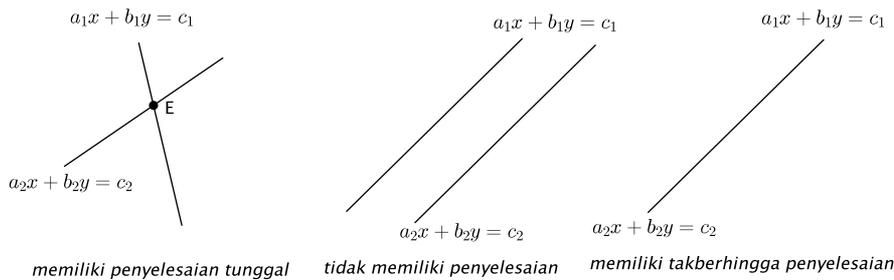
Bentuk ini dapat pula disajikan dalam bentuk matriks-vektor  $Ax = b$  dengan

$$A := \begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \cdots & a_{2n} \\ \vdots & \vdots & \vdots & \ddots & \\ a_{m1} & a_{m2} & a_{m3} & \cdots & a_{mn} \end{bmatrix}, x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}, \text{ dan } b = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix}.$$

Permasalahan dalam sistem persamaan linear adalah menentukan vektor  $x$  yang memenuhi persamaan



Gambar 6.1: Situasi sistem persamaan linear 3 persamaan dan 2 variabel



Gambar 6.2: Situasi sistem persamaan linear 2 persamaan 2 dua variabel

ini jika diberikan matriks  $A$  dan vektor  $b$ . Dalam hal ini  $A$  disebut matriks koefisien dan  $x$  disebut vektor penyelesaian. Untuk kasus  $m < n$  yakni persamaan lebih sedikit dari variabel maka selalu terdapat takberhingga banyak penyelesaian. Sebagai contoh, persamaan  $ax + by = c$  mempunyai penyelesaian  $(\lambda, \frac{c-a\lambda}{b})$  dengan  $\lambda$  konstanta sebarang. Secara geometris, semua titik yang terletak pada garis  $ax + by = c$  adalah penyelesaian. Untuk  $m > n$ , yakni persamaan lebih banyak dari variabel maka sistem persamaan linear umumnya tidak memiliki penyelesaian kecuali pada kasus yang sangat khusus seperti diilustrasikan pada Gambar 6.1 (kiri). Keadaan yang paling sering terjadi adalah sistem ini tidak mempunyai penyelesaian seperti ditunjukkan pada Gambar 6.1 (kanan). Dalam bidang aplikasi, sistem persamaan dengan  $m > n$  muncul pada masalah estimasi parameter yang melahirkan istilah penyelesaian optimal seperti pada metode kuadrat terkecil.

Untuk pembahasan pada bab ini dibatasi pada sistem persegi  $m = n$ , yaitu banyak variabel dan banyak persamaan adalah sama. Ada 3 kemungkinan eksistensi penyelesaian pada sistem persegi, yaitu memiliki penyelesaian tunggal, tidak memiliki penyelesaian, dan mempunyai takberhingga banyak penyelesaian. Ilustrasi keadaan ini untuk  $m = n = 2$  diberikan pada Gambar 6.2.

Jadi, pembahasan selanjutnya  $A$  diasumsikan berupa matriks persegi. Dalam pelajaran aljabar linear, syarat cukup agar sistem persamaan linear  $Ax = b$  mempunyai penyelesaian adalah matriks  $A$  taksingular, yakni memiliki invers  $A^{-1}$  atau invertibel sehingga penyelesaiannya dapat ditulis  $x = A^{-1}b$ . Sebuah matriks  $A$  berukuran  $n \times n$  dikatakan singular jika salah satu dari kondisi berikut dipenuhi, yaitu

1.  $A$  tidak mempunyai invers, yakni tidak ada matriks  $M$  sehingga  $AM = MA = I$  dengan  $I$  sebuah

**Mengapa tidak menggunakan invers matriks?** Secara teoretis, sistem persamaan linear  $Ax = b$  mempunyai penyelesaian tunggal jika dan hanya jika invers matriks  $A$  ada dan penyelesaian diberikan secara eksplisit oleh  $x = A^{-1}b$ . Pertanyaannya, mengapa kita tidak menggunakan formula invers ini langsung? Apakah menghitung invers matriks  $A$  lebih sulit daripada menjalankan metode eliminasi Gaussian? Kita berikan prosedur mencari invers matriks dari aspek kompleksitasnya. Misalkan matriks  $X$  adalah invers dari  $A$  maka haruslah dipenuhi

$$AX = I$$

dengan  $I$  sebuah matriks identitas. Menentukan invers  $A$  sama saja dengan mencari penyelesaian sistem persamaan ini. Perhatikan bentuk ini adalah kumpulan sistem persamaan linear

$$AX_1 = I_1, AX_2 = I_2, \dots, AX_n = I_n$$

dengan  $X_k$  dan  $I_k$  adalah vektor-vektor kolom matriks  $X$  dan  $I$ . Untuk menyelesaikan sekumpulan sistem persamaan linear ini, perlu modifikasi algoritma Gaussian dengan menambahkan satu loop lagi untuk mengakomodasi banyaknya vektor ruas kanan. Menurut Epperson (2013) flops untuk menyelesaikan algoritma modifikasi ini adalah  $\frac{5}{6}n^3 + \mathcal{O}(n^2)$  untuk eliminasi dan  $\frac{1}{2}n^3 + \mathcal{O}(n^2)$  untuk substitusi mundur sehingga total flops adalah  $\frac{4}{3}n^3 + \mathcal{O}(n^2)$ . Fakta ini menunjukkan kompleksitas metode invers ini empat kali lipat lebih tinggi daripada metode eliminasi Gaussian.

Selain dari masalah kompleksitas, penggunaan metode invers berdampak pada akurasi penyelesaian. Contoh sederhananya perhatikan persamaan  $3x = 6$ . Kalau menggunakan metode eliminasi diperoleh  $3x = 6 \leftrightarrow \frac{3}{3}x = \frac{6}{3} \leftrightarrow x = 6/3 = 2$  (hasilnya eksak) sedangkan pada metode invers diperoleh  $x = 3^{-1} \times 6 = \frac{1}{3} \times 6 = 0.3333 \times 6 = 1.9998$  (memuat kesalahannya). Inilah alasan mengapa metode invers hanya digunakan untuk analisis teoretis tapi tidak digunakan dalam implementasi numerik.

## 6.6 Praktikum Lab Komputasi

Sebagaimana telah dijelaskan bahwa metode eliminasi Gaussian membutuhkan dua tahap yaitu membawa sistem persamaan linear menjadi bentuk segitiga atas atau bawah kemudian menerapkan metode substitusi mundur atau maju. Implementasi numerik ini diawali dengan memecah tahapan ini menjadi tiga tahap, yaitu substitusi mundur atau maju, pembentukan matriks segitiga, dan gabungan keduanya.

### 6.6.1 Substitusi mundur dan maju

Untuk substitusi mundur diketahui  $A$  matriks segitiga atas dan vektor  $b$  sebagai berikut:

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ 0 & a_{22} & a_{23} & \cdots & a_{2n} \\ 0 & 0 & a_{33} & \cdots & a_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ \vdots \\ b_n \end{bmatrix}$$

dengan iterasi  $x_n = \frac{b_n}{a_{nn}}$ ,  $x_k = \frac{1}{a_{kk}} \left( b_k - \sum_{j=k+1}^n a_{kj} x_j \right)$ ,  $k = n-1, n-2, \dots, 1$ . Berdasarkan pseudo code yang sudah dibahas sebelumnya, berikut m-file Matlabnya.

### 6.6.6 Implementasi strategi pivoting total

Kalau pada strategi pivoting parsial baik dengan skala maupun tanpa skala kita hanya perlu melakukan pertukaran baris, namun pada strategi pivoting total kita perlu juga melakukan pertukaran kolom. Permasalahannya kalau pertukaran baris tidak mengubah penyelesaian tapi pertukaran kolom mengakibatkan pertukaran penyelesaian. Pada iterasi ke- $i$  perlu diidentifikasi bilangan bulat  $p$  yaitu kolom yang memuat elemen dengan magnitudo terbesar untuk ditukarkan dengan kolom ke- $i$ . Setelah dilakukan pertukaran kolom, proses selanjutnya mengikuti strategi pivoting parsial.

Kolom-kolom hasil identifikasi perlu direkam sebagai proses transformasi dengan matriks permutasi. Sebagai contoh misalkan untuk sistem berukuran  $n = 4$  diperoleh rekaman pertukaran kolom sebagai berikut:

$$\text{mutan} = [ p_1 \quad p_2 \quad p_3 ]$$

maka telah terjadi mutasi sistem persamaan linear dari  $Ax = b$  menjadi  $APx = b$  dengan  $P = P_1P_2P_3$  di mana  $P_i$  matriks permutasi hasil pertukaran kolom ke-  $i$  dan kolom ke- $p_i$  matriks identitas. Sebagai contoh  $p_1 = 3, p_2 = 4, p_3 = 4$  maka diperoleh

$$P_1 = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, P_2 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}, P_3 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}.$$

Algoritma strategi pivoting total ini untuk sistem  $Ax = b$  dapat diringkas sebagai berikut:

1. Untuk  $i = 1, 2, \dots, n - 1$ ,
  - (a) perhatikan submatriks  $A_i := A(i : n, i : n)$  dengan  $A$  matriks hasil eliminasi. Tentukan kolom  $p_i$  matriks  $A_i$  yang memuat elemen dengan magnitudo terbesar.
  - (b) pertukarkan kolom matriks  $A$  dengan  $(K_i) \leftrightarrow (K_{p_i})$ .
  - (c) terapkan eliminasi dengan strategi pivoting parsial.

Setelah looping diselesaikan diperoleh sistem persamaan  $\Delta z = b_1$  dengan  $\Delta$  matriks segitiga atas dan rekaman kolom yang dipertukarkan yaitu  $\text{mutan} = [p_1, p_2, \dots, p_{n-1}]$ .

2. Lakukan substitusi mundur untuk memperoleh  $z = [z_1, z_2, \dots, z_n]^T$ .
3. Peroleh kembali penyelesaian sistem persamaan semula  $x = Pz$  dengan  $P = P_1P_2P_3$ .

Adapun m-file yang bersesuaian dengan algoritma ini diberikan sebagai berikut.

```
function x=total_pivoting(A,b)
n=length(b);
mutan=zeros(1,n-1);%menyimpan data kolom yang dipertukarkan.
for i=1:n-1
    d=max(abs(A(i:end,i:end)));%magnitudo terbesar setiap kolom
    pp=find(d==max(d));
    p=pp(1)+i-1;
    mutan(i)=p;
```

```

AA=A;
AA(:,i)=A(:,p);AA(:,p)=A(:,i);%pertukaran kolom;
%pertukaran baris untuk memilih pivot terbesar
t=find(abs(AA(i:end,i))==max(abs(AA(i:end,i)))));
p1=t(1)+i-1;
A2=AA; b1=b;
A2(i,:)=AA(p1,:);A2(p1,:)=AA(i,:);
b1(i)=b(p1);b1(p1)=b(i);
%eliminasi per kolom
for j=i+1:n
    lambda=A2(j,i)/A2(i,i);
    for k=i:n
        A2(j,k)=A2(j,k)-lambda*A2(i,k);
    end
    b1(j)=b1(j)-lambda*b1(i);
end
A=A2;b=b1;
end
%menjalankan substitusi mundur
z=zeros(n,1);%menyiapkan vektor penyelesaian
z(n)=b1(n)/A2(n,n);
for k=n-1:-1:1
    sum=0;
    for j=k+1:n
        sum=sum+A2(k,j)*z(j);
    end
    z(k)=(b1(k)-sum)/A2(k,k);
end
%transformasi dengan matriks permutasi
P=eye(n);
for k=1:n-1
    P1=matriks_perm(n,k,mutan(k));
    P=P*P1;
end
x = P*z;

```

Untuk uji coba kita ambil sistem persamaan linear yang sama ketika mengimplementasikan strategi pivoting skala parsial sebelumnya sekaligus kita bandingkan hasilnya.

```

>> A=[3.3333 15920 10.333;2.222 16.710 9.612;-1.5611 5.1792 -1.6855];
>> b=[7953;0.965;2.714];
>> x=total_pivoting(A,b)
x =
    0.999999869458794
    0.499999981142246

```

## BAB 7

# Aproksimasi Akar Sistem Persamaan Taklinear

*Dan apakah orang-orang yang kafir tidak mengetahui bahwasanya langit dan bumi itu keduanya dahulu adalah suatu yang padu, kemudian Kami pisahkan antara keduanya. Dan dari air Kami jadikan segala sesuatu yang hidup. Maka mengapakah mereka tiada juga beriman?*

QS 21:30.

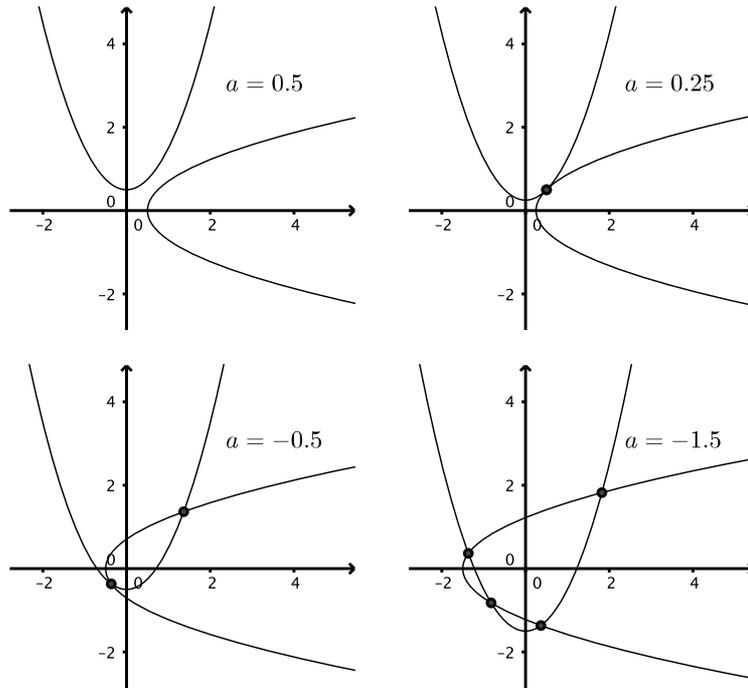
## 7.1 Pendahuluan

Pada bab sebelumnya telah dibahas sistem persamaan linear, yaitu kumpulan persamaan-persamaan linear yang harus diselesaikan secara simultan atau terpadu. Jika sebagian atau semua persamaan di dalam kumpulan ini berupa persamaan taklinear maka ia disebut sistem persamaan taklinear. Pada sekolah menengah sudah diperkenalkan sistem persamaan campuran yang terdiri atas sebuah persamaan linear dan sebuah persamaan taklinear. Umumnya persamaan ini diselesaikan dengan metode substitusi. Sebagai ilustrasi, sistem persamaan campuran berikut

$$\begin{aligned}x + y &= 1 \\x^2 - y &= 0,\end{aligned}$$

secara geometris berupa sebuah garis lurus dan sebuah parabola seperti ditunjukkan pada Gambar 7.1. Penyelesaiannya adalah titik potong kedua kurva ini. Dalam hal ini penyelesaian diperoleh dengan substitusi persamaan pertama  $y = 1 - x$  ke persamaan kedua  $x^2 - (1 - x) = 0$ . Persamaan kuadrat ini  $x^2 + x - 1 = 0$  memiliki akar  $x_1 = -\frac{1}{2} + \frac{1}{2}\sqrt{5}$  dan  $x_2 = -\frac{1}{2} - \frac{1}{2}\sqrt{5}$ . Masing-masing disubstitusikan ke  $y = 1 - x$  untuk memperoleh  $y_1 = 1 - x_1 = \frac{3}{2} - \frac{1}{2}\sqrt{5}$  dan  $y_2 = 1 - x_2 = \frac{3}{2} + \frac{1}{2}\sqrt{5}$ . Dengan demikian akar sistem persamaan ini adalah  $(x_1, y_1) = (-\frac{1}{2} + \frac{1}{2}\sqrt{5}, \frac{3}{2} - \frac{1}{2}\sqrt{5})$  dan  $(x_2, y_2) = (-\frac{1}{2} - \frac{1}{2}\sqrt{5}, \frac{3}{2} + \frac{1}{2}\sqrt{5})$ . Dalam hal ini, sistem persamaan memiliki dua penyelesaian.

Menurut Heat (1997), paling tidak ada tiga permasalahan utama dalam menyelesaikan sistem persamaan taklinear. Pertama, eksistensi dan banyak akar sebuah sistem persamaan taklinear lebih sulit dideteksi daripada sistem persamaan linear. Kedua, tidak ada cara sederhana untuk menjamin kekonvergenan barisan iterasi ke penyelesaian eksak. Ketiga, kompleksitas komputasi bertambah sangat



Gambar 7.2: Kemungkinan eksistensi dan banyak akar sistem persamaan taklinear

dengan masing-masing  $f_i : \mathbb{R}^n \rightarrow \mathbb{R}$ . Selanjutnya misalkan  $\mathbf{F} : \mathbb{R}^n \rightarrow \mathbb{R}^n$  dan  $\mathbf{x} = (x_1, x_2, \dots, x_n)$  dengan

$$\mathbf{F}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_n(\mathbf{x}))^T$$

maka sistem persamaan taklinear di atas dapat ditulis dalam bentuk:

$$\mathbf{F}(\mathbf{x}) = \mathbf{0}. \quad (7.1.1)$$

Oleh karena sulitnya pendeteksian eksistensi akar sistem persamaan taklinear maka pembahasan berikutnya fokus pada metode aproksimasi akar dengan mengasumsikan sistem persamaan linear yang dihadapi memiliki minimal satu akar. Sedangkan masalah eksistensi akar dan analisis galat aproksimasi akan dibahas setelah implementasi numerik setiap metode yang dibahas.

## 7.2 Metode Newton

Untuk persamaan taklinear tunggal  $f(x) = 0$ , iterasi metode Newton adalah  $p_n = p_{n-1} - \frac{f(p_{n-1})}{f'(p_{n-1})}$  dengan  $p_0$  nilai awal yang diberikan. Iterasi ini diperoleh melalui perpotongan garis singgung kurva  $y = f(x)$  dengan sumbu  $X$ . Cara lainnya adalah dengan menggunakan aproksimasi polinomial Taylor derajat satu di sekitar  $x = p_0$  sebagai berikut:

$$f(x) \approx f(p_0) + f'(p_0)(x - p_0).$$

Dengan menetapkan  $f(x) = 0$  dan mengambil  $p_1 := x$  maka diperoleh aproksimasi  $f'(p_0)(p_1 - p_0) = -f(p_0)$ , yaitu  $p_1 = p_0 - \frac{f(p_0)}{f'(p_0)}$ . Hasil ini sama dengan perolehan dengan menggunakan garis singgung. Derivasi

Kemudian hitung  $\mathbf{F}(\mathbf{p}^{(0)})$  diperoleh

$$\mathbf{F}(\mathbf{p}^{(0)}) = \begin{bmatrix} f_1(\mathbf{p}^{(0)}) \\ f_2(\mathbf{p}^{(0)}) \end{bmatrix} = \begin{bmatrix} 0^2 - 1 + 0.25 \\ 0 + 1^2 + 0.25 \end{bmatrix} = \begin{bmatrix} -0.75 \\ 1.25 \end{bmatrix}.$$

Tentukan  $\mathbf{q}_0$  dari sistem persamaan linear  $\mathbf{J}_F(\mathbf{p}^{(0)})\mathbf{q}^{(0)} = \mathbf{F}(\mathbf{p}^{(0)})$  diperoleh

$$\mathbf{q}^{(0)} = \begin{bmatrix} -0.25 \\ 0.75 \end{bmatrix}.$$

Hitung pembaruan akar  $\mathbf{p}^{(1)}$  dengan

$$\mathbf{p}^{(1)} = \mathbf{p}^{(0)} - \mathbf{q}^{(0)} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} - \begin{bmatrix} -0.25 \\ 0.75 \end{bmatrix} = \begin{bmatrix} 0.25 \\ 0.25 \end{bmatrix}.$$

Aproksimasi 2: Berangkat dari  $\mathbf{p}^{(1)} = \begin{bmatrix} 0.25 \\ 0.25 \end{bmatrix}$  hitung  $\mathbf{J}_F(\mathbf{p}^{(1)})$ ,  $\mathbf{F}(\mathbf{p}^{(1)})$ , dan  $\mathbf{q}^{(1)}$  pada sistem persamaan linear  $\mathbf{J}_F(\mathbf{p}^{(1)})\mathbf{q}^{(1)} = \mathbf{F}(\mathbf{p}^{(1)})$  diperoleh

$$\mathbf{J}_F(\mathbf{p}^{(1)}) = \begin{bmatrix} 0.5 & -1 \\ 1 & 0.5 \end{bmatrix}, \mathbf{F}(\mathbf{p}^{(1)}) = \begin{bmatrix} 0.0625 \\ 0.5625 \end{bmatrix}, \text{ dan } \mathbf{q}^{(1)} = \begin{bmatrix} 0.475 \\ 0.175 \end{bmatrix}.$$

Perbaharui aproksimasi menjadi  $\mathbf{p}^{(2)}$  dengan

$$\mathbf{p}^{(2)} = \mathbf{p}^{(1)} - \mathbf{q}^{(1)} = \begin{bmatrix} 0.25 \\ 0.25 \end{bmatrix} - \begin{bmatrix} 0.475 \\ 0.175 \end{bmatrix} = \begin{bmatrix} -0.225 \\ 0.075 \end{bmatrix}.$$

Prosedur ini dilanjutkan sampai diperoleh 6 iterasi pertama seperti diberikan pada tabel berikut.

$k$	0	1	2	3	4	5	6
$p_1^{(k)}$	0	0.25	-0.225	-0.2941	-0.4074	-0.4525	-0.4763
$p_2^{(k)}$	1	0.25	0.075	0.3317	0.4032	0.4527	0.4763

Ternyata pada iterasi ke-enam diperoleh  $p_1^{(6)} = -0.4363$  aproksimasi untuk  $x_1 = -0.5$  dan  $p_2^{(6)} = 0.4763$  aproksimasi untuk  $x_2 = 0.5$ .

**Permasalahan kritis metode Newton** Seperti halnya metode Newton untuk persamaan tunggal, iterasi gagal ketika ditemukan Jacobian yang singular. Permasalahan berikutnya adalah kompleksitas komputasi tumbuh sangat cepat seiring bertambahnya dimensi sistem persamaan. Untuk sistem berdimensi  $n$ , perhitungan Jacobian  $\mathbf{J}_F$  membutuhkan sebanyak  $n^2$  evaluasi fungsi skalar, kemudian penyelesaian sistem persamaan linear  $\mathbf{J}_F(\mathbf{p}_{n-1})\mathbf{q}_{n-1} = \mathbf{F}(\mathbf{p}_{n-1})$  dengan eliminasi Gaussian membutuhkan  $\mathcal{O}(n^3)$  operasi aritmetika (flops). Jika Jacobian taksingular, kekonvergenan barisan iterasi biasanya kuadratik asalkan berangkat dari nilai awal yang cukup dekat dengan solusi eksak (Heat, 1997). Dalam prakteknya, pemilihan nilai awal ini menjadi permasalahan tersendiri.

metode Broyden dengan mengambil  $B_0$  matriks identitas, yaitu

$$B_0 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

Bandingkan hasilnya dengan ketika  $B_0$  diambil Jacobian eksak seperti pada contoh sebelumnya.

Kompleksitas komputasi Jacobian berkurang dari sebelumnya  $n^2$  pada metode Newton menjadi  $n$  pada metode Broyden ini, terlihat ketika pemutakhiran  $\mathbf{B}_k$  seperti pada langkah 4. Namun, order konvergensinya hanya superlinear, bukan kuadratik seperti pada metode Newton (Faires dan Burden, 2013). Akibatnya,

$$\lim_{k \rightarrow \infty} \frac{\|\mathbf{p}^{(k+1)} - \mathbf{p}\|}{\|\mathbf{p}^{(k)} - \mathbf{p}\|} = 0,$$

di mana  $\mathbf{p}$  akar eksak sistem persamaan taklinear  $\mathbf{F}(\mathbf{x}) = \mathbf{0}$ ,  $\mathbf{p}^{(k)}$  dan  $\mathbf{p}^{(k+1)}$  adalah dua aproksimasi berurutan. Dengan kata lain, galat  $E_{k+1} := \|\mathbf{p}^{(k+1)} - \mathbf{p}\|$  semakin lama semakin lebih kecil daripada galat  $E_k := \|\mathbf{p}^{(k)} - \mathbf{p}\|$ .

## 7.5 Metode Penurunan Tercuram

Kelebihan metode Newton dan metode quasi-Newton khususnya metode Broyden yang baru kita bahas terletak pada kecepatan konvergensinya, namun kelemahannya adalah kepastian pemilihan nilai awal yang menjamin barisan iterasinya konvergen. Kali ini dibahas metode penurunan tercuram (*steepest descent method*) yang hanya konvergen order linear namun secara umum pasti konvergen untuk hampir setiap nilai awal yang diberikan.

Ide metode ini dapat dijelaskan sebagai berikut: misalkan sistem persamaan taklinear disajikan dalam bentuk

$$\mathbf{F}(\mathbf{x}) = \mathbf{0} \tag{7.5.1}$$

dengan  $\mathbf{F} = [f_1, f_2, \dots, f_n]^T$  di mana  $f_i : \mathbb{R}^n \rightarrow \mathbb{R}$ ,  $i = 1, 2, \dots, n$  biasanya disebut fungsi-fungsi koordinat  $\mathbf{F}$ . Definisikan fungsi  $g : \mathbb{R}^n \rightarrow \mathbb{R}$  dengan

$$g(\mathbf{x}) := \sum_{i=1}^n [f_i(\mathbf{x})]^2. \tag{7.5.2}$$

Persamaan (7.5.1) mempunyai penyelesaian jika dan hanya jika fungsi pada (7.5.2) mempunyai nilai minimum nol. Selanjutnya  $\mathbf{p} := \arg \min g(\mathbf{x})$  adalah akar dari persamaan  $\mathbf{F}(\mathbf{x}) = \mathbf{0}$ . Metode penurunan tercuram adalah metode untuk menemukan minimum lokal fungsi  $g : \mathbb{R}^n \rightarrow \mathbb{R}$  secara intuitif sebagai berikut:

1. Hitung nilai  $g$  di titik awal yang diberikan  $\mathbf{p}^{(0)} = (p_1^{(0)}, p_2^{(0)}, \dots, p_n^{(0)})^T$ .
2. Tentukan arah dari  $\mathbf{p}^{(0)}$  yang menghasilkan penurunan nilai  $g$ .
3. Geserkan sedikit vektor  $\mathbf{p}^{(0)}$  dalam arah ini untuk memperoleh pemutakhiran  $\mathbf{p}^{(1)}$ .
4. Ulangi langkah 1-3 dengan mengganti  $\mathbf{p}^{(0)}$  dengan  $\mathbf{p}^{(1)}$ .

Pada analisis elementer dibahas konsep fungsi kontraksi, nilai mutlak  $|\cdot|$  pada  $\mathbb{R}$ , fungsi Lipschitz, dan berbagai sifat fungsi dengan sifat-sifat tersebut maka di sini konsep fungsi diperumum menjadi operator yang bekerja pada ruang bernorma  $(X, \|\cdot\|)$ . Pengertian norma, khususnya norma vektor dan matriks telah dibahas pada Bab 6. Pengertian kontraksi dan Lipschitz pada operator mirip dengan pengertian yang sama pada fungsi real dengan nilai mutlak diganti norma secara umum. Berikut ini diberikan beberapa pengertian dan teorema pendukung yang dibutuhkan. Istilah-istilah tertentu yang belum dipahami dapat dibaca pada berbagai buku analisis fungsional elementer.

**Definisi 7.2.** Misalkan  $U$  himpunan bagian dari sebuah ruang bernorma  $(X, \|\cdot\|)$ . Sebuah operator  $A : U \rightarrow X$  dikatakan kontraksi jika terdapat sebuah konstanta  $0 \leq q < 1$  sehingga

$$\|Ax - Ay\| \leq q \|x - y\|$$

untuk setiap  $x, y \in U$ . Sebuah operator  $A : U \rightarrow X$  dikatakan kontinu Lipschitz jika terdapat konstanta positif  $L$  sehingga

$$\|Ax - Ay\| \leq L \|x - y\|$$

untuk setiap  $x, y \in U$ . Jadi operator kontraksi adalah kontinu Lipschitz dengan konstanta kurang dari 1.

**Definisi 7.3.** Sebuah elemen  $x$  pada sebuah ruang bernorma  $X$  dikatakan titik tetap (*fixed point*) dari operator  $A : U \subset X \rightarrow X$  jika berlaku  $Ax = x$ .

Pada fungsi real biasa, titik tetap diinterpretasikan sebagai absis titik potong kurva  $y = f(x)$  dan garis  $y = x$ , namun untuk fungsi pada ruang bernorma secara umum, titik tetap ini sangat sulit diilustrasikan secara visual. Untuk ruang bernorma berdimensi  $n$ , titik tetap diperoleh melalui perpotongan antara luasan (bidang lengkung) dengan sebuah bidang berdimensi  $n$  (*hyperplane*).

Berikut ini diberikan teorema titik tetap Banach yang memberikan syarat cukup sebuah operator memiliki titik tetap tunggal.

**Teorema 7.1.** [TEOREMA TITIK TETAP BANACH] *Misalkan  $U$  himpunan bagian lengkap dari sebuah ruang bernorma  $X$  dan  $A : U \rightarrow U$  sebuah operator kontraksi. Maka  $A$  memiliki titik tetap tunggal.*

BUKTI. Kita ingatkan dulu bahwa di dalam sebuah himpunan lengkap setiap barisan Cauchy konvergen. Proses pembuktian mulai dari pengambilan sebarang elemen  $x_0 \in U$ , kemudian secara rekursif didefinisikan barisan  $(x_n)$  sebagai berikut:

$$x_{n+1} := Ax_n, n = 0, 1, 2, \dots$$

Mengingat  $A$  kontraksi maka diperoleh

$$\begin{aligned} \|x_{n+1} - x_n\| &= \|Ax_n - Ax_{n-1}\| \leq q \|x_n - x_{n-1}\| \\ &\leq q^2 \|x_{n-1} - x_{n-2}\| \leq \dots \\ &\leq q^n \|x_1 - x_0\|, n = 1, 2, \dots \end{aligned}$$

Untuk  $m > n$  kita gunakan ketaksamaan segitiga untuk menghasilkan ekspansi berikut.

$$\begin{aligned} \|x_n - x_m\| &\leq \|x_n - x_{n+1}\| + \|x_{n+1} - x_{n+2}\| + \cdots + \|x_{m-1} - x_m\| \\ &\leq (q^n + q^{n+1} + \cdots + q^{m-1}) \|x_1 - x_0\| \\ &\leq (q + q^2 + \cdots + q^{m-1} + \cdots + q^n) \|x_1 - x_0\| = \frac{q^n}{1-q} \|x_1 - x_0\|. \end{aligned}$$

Oleh karena  $0 \leq q < 1$  maka  $q^n \rightarrow 0$  sehingga  $(x_n)$  merupakan barisan Cauchy pada himpunan lengkap  $U$ . Jadi,  $(x_n)$  konvergen yaitu ada  $x \in U$  sehingga  $x_n \rightarrow x$ . Fakta lain sebuah operator kontraksi adalah kontinu. Dengan sifat ini diperoleh:

$$x = \lim(x_{n+1}) = \lim(Ax_n) = A(\lim(x_n)) = Ax;$$

yakni,  $x$  titik tetap dari  $A$ . Selanjutnya kita buktikan titik tetap ini tunggal dengan metode kontradiksi. Andai ada titik tetap  $y$  dengan  $x \neq y$  maka berlaku

$$0 \neq \|x - y\| = \|Ax - Ay\| \leq q \|x - y\| \rightarrow q \geq 1,$$

padahal  $q < 1$  (kontradiksi). □

Kelebihan teorema titik tetap Banach adalah pembuktiannya secara konstruktif. Di samping menunjukkan eksistensi titik tetap, ia juga memberikan sebuah algoritma untuk menentukan aproksimasi penyelesaian secara numerik. Bentuk algoritmanya sangat sederhana sehingga dapat diaplikasi pada program komputer dengan sangat mudah. Berikut kita berikan bentuk estimasi galatnya.

**Teorema 7.2.** *Misalkan  $A$  sebuah operator kontraksi dengan konstanta  $0 \leq q < 1$  yang memetakan subhimpunan lengkap  $U$  pada ruang bernorma  $X$  kepada dirinya sendiri. Jika didefinisikan iterasi*

$$x_{n+1} := Ax_n, n = 0, 1, 2, \dots,$$

*dengan sebarang  $x_0 \in U$  maka  $(x_n)$  konvergen ke titik tetap  $x$  dari  $A$ . Lebih lanjut estimasi galat awal (priori error estimate) diberikan oleh*

$$\|x_n - x\| \leq \frac{q^n}{1-q} \|x_1 - x_0\|$$

*dan estimasi galat akhir (posteriori error estimate) diberikan oleh*

$$\|x_n - x\| \leq \frac{q}{1-q} \|x_n - x_{n-1}\|.$$

**BUKTI.** Sebelumnya telah diperoleh bahwa  $\|x_n - x_m\| \leq \frac{q^n}{1-q} \|x_1 - x_0\|$ . Dengan mengambil  $m \rightarrow \infty$  maka diperoleh estimasi galat awal. Jika titik awal pada estimasi galat awal  $x_0$  diganti  $x_{n-1}$  maka terjadi pergeseran  $n-1$  langkah ke belakang yaitu dari  $x_1 - x_0$  menjadi  $x_n - x_{n-1}$ . Akibatnya berlaku pergeseran pangkat dari  $q^n$  menjadi  $q^{n-(n-1)} = q$ . Dengan argumen ini diperoleh estimasi galat akhir. □

Perlu dijelaskan bahwa estimasi galat awal (*prior*) adalah perbandingan galat pada iterasi ke- $n$ ,  $e_n := \|x_n - x\|$  terhadap selisih dua iterasi pertama  $\|x_1 - x_0\|$  sedangkan estimasi akhir (*posterior*) adalah membandingkan galat  $e_n$  dengan selisih dua iterasi terakhir  $\|x_n - x_{n-1}\|$ . Mengingat barisan  $(x_n)$

Pada bab ini telah dibahas beberapa metode untuk menyelesaikan sistem persamaan taklinear. Eksistensi dan ketunggalan penyelesaian sulit dideteksi apalagi dimensinya cukup tinggi. Metode Newton dapat dikatakan paling populer, formulanya merupakan perumuman dari kasus persamaan taklinear tunggal di mana istilah derivatif fungsi diperumum menjadi matriks Jacobian. Dalam implementasinya, metode Newton membutuhkan nilai awal yang cocok agar barisan iterasinya konvergen cepat ke penyelesaian eksak. Secara teori, nilai awal yang baik harus cukup dekat terhadap akar eksak. Mengingat akar eksak tidak diketahui maka pemilihan ini tidak dapat dilakukan dan biasanya dipilih secara intuitif atau coba-coba. Untuk bentuk modifikasi metode Newton dapat dibaca dalam (McDougall, dkk., 2014).

Dalam banyak kasus metode Newton sangat cepat kekonvergenannya namun ia berbiaya tinggi dalam menghitung Jacobian pada setiap iterasinya. Melalui metode Chord dan modifikasinya, biaya komputasi ini dapat dikurangi namun rate konvergensinya menjadi lambat. Metode Broyden hampir sama dengan metode Chord. Kalau pada metode Chord, Jacobian diaproksimasi oleh sebuah matriks tetap namun pada metode Broyden matriks Jacobian diaproksimasi pada setiap iterasi tanpa melakukan evaluasi fungsi seperti menghitung Jacobian secara eksak. Ada cara bagus menghitung Jacobian yaitu menggunakan metode *automatic differentiation* (AD) yang didasarkan aturan rantai namun tidak dibahas di sini. Pembahasan lebih lanjut tentang AD dapat dibaca pada (Hernadi, 2015; Neidinger, 2010).

Metode yang lebih umum adalah metode penurunan tercuram yang biasa digunakan untuk menentukan minimum sebuah fungsi multivaribel. Fungsi multivaribel di sini diperoleh dengan cara menjumlahkan kuadrat fungsi-fungsi koordinat yang ada pada sistem persamaan taklinear. Pada metode ini, pemutakhiran aproksimasi dilakukan dengan cara menggeser vektor sebelumnya dalam arah negatif gradien. Permasalahan krusial di sini adalah menentukan skala geser pada setiap iterasi karena skala ini berpengaruh pada kecepatan konvergensi. Selama ini skala geser ini diambil pada interval  $(0, 1)$  dan diupdate pada setiap iterasi dengan menggunakan pendekatan polinomial interpolasi kuadratik.

Semua metode yang dibahas telah disusun m-file Matlab dan telah dilakukan eksperimen numerik. Untuk menjangkau semua kemungkinan akar dilakukan dengan cara memainkan nilai awal. Tidak semua nilai awal menghasilkan barisan yang konvergen oleh karena itu kriteria stopping algoritma dijadikan sebagai indikator akurasi.

Teori yang mendasari sistem persamaan taklinear ini ternyata berasal dari analisis fungsional. Eksistensi titik tetap Banach menjadi pijakan bagi metode-metode yang ada khususnya metode Newton. Hal ini disebabkan bentuk ekuivalensi antara metode Newton dan metode iterasi titik tetap. Syarat cukup bagi metode Newton mempunyai penyelesaian, jaminan kekonvergenan untuk setiap pemilihan titik awal, dan order konvergensinya sudah diberikan dengan jelas pada teorema-teorema sebelumnya namun masih sulit diimplementasikan secara numerik.

## 7.8 Soal-soal Latihan Bab 7

1. Susunlah formula iterasi Newton untuk menyelesaikan sistem persamaan taklinear berikut:

(a)  $x_1^2 + x_2^2 = 1$  dan  $x_1^2 - x_2 = 0$ .

(b)  $2 \sin x_1 + \cos x_2 - 5x_1 = 0$  dan  $4 \cos x_1 + 2 \sin x_2 - 5x_2 = 0$ .

2. Perhatikan sistem persamaan taklinear  $x_1 - 1 = 0$  dan  $x_1 x_2 - 1 = 0$ . Untuk nilai awal apa saja, metode Newton untuk menyelesaikan masalah ini akan gagal?

3. Pada awal bab ini kita berikan ilustrasi sistem persamaan taklinear berikut ini:  $x^2 - y + a = 0$  dan  $x + y^2 + a = 0$  dengan  $a$  sebuah konstanta. Eksistensi akar sistem persamaan ini bergantung pada nilai konstanta  $a$ , yaitu tidak memiliki akar jika  $a > 0.25$ , memiliki akar tunggal jika  $a = 0.25$ , memiliki 2 akar jika  $a \in [-0.75, 0.25)$ , dan memiliki 4 akar jika  $a < -0.75$ . Lakukan eksperimen numerik dengan mengambil masing-masing kasus yang mewakili. yaitu  $a = 0.5$ ,  $a = 0.25$ ,  $a = -0.5$ , dan  $a = -1$ . Pilihlah nilai awal secara intuitif atau coba-coba.
- Untuk  $a = 0.5$ , kemana kecenderungan barisan iterasi yang diperoleh konvergen jika digunakan metode Newton.
  - Untuk  $a = 0.25$ , ujlilah ke-empat metode yang sudah dibahas yaitu Newton, chord, modifikasi chord, Broyden, dan penurunan tercuram. Bagaimana kekonvergenan masing-masing metode.
  - Untuk  $a = -0.5$ , pilih dua nilai awal pada metode Newton sehingga masing-masing konvergen ke akar-akar yang berbeda. Apa yang Anda peroleh jika nilai awal yang sama diterapkan pada metode Broyden.
  - Untuk  $a = -1$ , temukan 4 akar yang ada dengan menggunakan Newton dan penurunan tercuram. Bandingkan kecepatan konvergensi kedua metode ini.
4. Ketika mempelajari integral Gauss pada Bab 4, kita perlu menentukan absis  $x_i$  dan bobot  $w_i$ ,  $i = 1, 2, \dots, n$  dengan cara menyelesaikan sistem persamaan taklinear. Untuk Gauss order  $n = 2$  diperoleh sistem persamaan taklinear sebagai berikut:

$$\begin{aligned}w_1 + w_2 &= 2, \\w_1 x_1 + w_2 x_2 &= 0, \\w_1 x_1^2 + w_2 x_2^2 &= \frac{2}{3}, \\w_1 x_1^3 + w_2 x_2^3 &= 0.\end{aligned}$$

Secara analitik penyelesaian eksak persamaan ini adalah  $w_1 = w_2 = 1$  dan  $x_1 = -\frac{1}{\sqrt{3}}$ ,  $x_2 = \frac{1}{\sqrt{3}}$ . Gunakan metode Newton untuk menyelesaikan sistem persamaan taklinear ini. Pilihlah nilai awal sebarang dan berhentilah ketika  $\|\mathbf{x}^{(k)} - \mathbf{x}^{(k-1)}\| < 10^{-6}$ . Dalam hal ini  $\mathbf{x} = (x_1, w_1, x_2, w_2)^T$ .

- Temukan nilai absis dan bobot integral Gauss order  $n = 3, 4, 5$  seperti tertera pada Tabel 4.2 dengan terlebih dahulu membangun sistem persamaan taklinear yang bersesuaian.
- Diberikan sistem persamaan taklinear sebagai berikut

$$\begin{aligned}x_1^3 + x_1^2 x_2 - x_1 x_3 + 6 &= 0 \\e^{x_1} + e^{x_2} - x_3 &= 0 \\x_2^2 - 2x_1 x_3 &= 4.\end{aligned}$$

Gunakan nilai awal  $\mathbf{x}^{(0)} = (-1, -2, 1)^T$ , kemudian lakukan iterasi untuk menemukan aproksimasi  $\mathbf{x}^{(n)}$  dengan  $\|\mathbf{x}^{(n)} - \mathbf{x}^{(n-1)}\| < 10^{-6}$ . Gunakan metode Newton, chord modifikasi dengan periode 3, metode Broyden, dan metode penurunan tercuram. Metode mana yang memberikan kecepatan paling tinggi?

(b)  $|g(x_1, x_2, x_3)|$

Apakah nilai minimum yang diperoleh sudah dekat dengan nol?

11. Perhatikan sistem persamaan taklinear berikut:

$$\begin{aligned} 3x_1 - \cos(x_2x_3) - 0.5 &= 0, \\ x_1^2 - 625x_2^2 - 0.25 &= 0, \\ e^{-x_1x_2} + 20x_3 + \frac{1}{3}(10\pi - 3) &= 0. \end{aligned}$$

- (a) Tunjukkan  $x_1 = 0.5, x_2 = 0$  dan  $x_3 = -\frac{\pi}{6}$  adalah penyelesaian eksak sistem persamaan ini.
- (b) Tunjukkan bahwa Jacobian tidak terdefinisi pada penyelesaian ini.
- (c) Terapkan metode Newton dan metode Broyden dengan mencoba beberapa nilai awal. Bagaimana kekonvergenannya? Hati-hati pengambilan nilai awal pada metode Newton agar menghindari  $x_2^{(0)} = 0$  karena mengakibatkan Jacobian tidak terdefinisi.

12. Persamaan lingkaran pusat  $(a, b)$  dan jari-jari  $r$  diberikan oleh

$$(x - a)^2 + (y - b)^2 = r^2.$$

Jika tiga titik berikut  $(0.62, 2.52), (-2.92, 5.22)$ , dan  $(-1.04, -1.25)$  terletak pada lingkaran ini, tentukan nilai  $a, b$ , dan  $r$ .

13. Sebuah satelit yang mengorbit bumi memiliki jarak terhadap bumi  $R$  mengikuti formula

$$R = \frac{C}{1 + e \sin(\theta + \alpha)}$$

di mana  $(R, \theta)$  adalah koordinat polar dari lintasan satelit tersebut terhadap pusat bumi. Konstanta  $C, e$ , dan  $\alpha$  perlu diketahui. Dalam hal ini konstanta  $e$  disebut eksentrisitas orbit. Jika diperoleh 3 data hasil observasi berikut, tentukan nilai konstanta  $C, e$ , dan  $\alpha$ .

$\theta$	$-30^\circ$	$0^\circ$	$30^\circ$
R(km)	6870	6728	6615

Kemudian gambarkan bentuk orbit satelit ini dan tentukan jarak terdekat dan terjauhnya terhadap bumi.

# Daftar Pustaka

- [1] R. G. Bartle and D. R. Sherbert. *Introduction to Real Analysis*. John Wiley & Sons, Inc, New York, 1994.
- [2] M. Benzi. Preconditioning techniques for large linear systems: a survey. *Journal of Computational Physics*, 182(2):418–477, November 2002.
- [3] G. Birkhoff and G. C. Rotta. *Ordinary Differential Equations, 4th Edition*. John Wiley and Sons, 1989.
- [4] P. Bogacki and L. F. Shampine. A pair of runge-kutta formulas. *Applied Mathematics Letters*, 2(4)., 1989.
- [5] J. L. Buchanan and P.R. Turner. *Numerical Method and Analysis*. McGraw-Hill., 1992.
- [6] J. C. Burkill. *A First Course in Mathematical Analysis*. Cambridge University Press, 1962.
- [7] J. C. Butcher. *Numerical Methods for Ordinary Differential Equations. 2nd Edition*. John Wiley & Sons, Ltd., Chichester, 2008.
- [8] K. Chen. *Matrix Preconditioning Techniques and Applications*. Cambridge University Press, UK, 2005.
- [9] Carl Christian. Machine epsilon value for ieee double precision standard alternative proof using relative error. Mathematics Stack Exchange, October 2020. URL:<https://math.stackexchange.com/q/3862085> (version: 2020-10-12),.
- [10] E. Coddington and N. Levinson. *Theory of Ordinary Differential Equations. McGraw-Hill, New York*. McGraw-Hill, New York, 1955.
- [11] J. F. Epperson. *Introduction to Numerical Methods and Analysis*. John Willey & Sons, Inc., New Jersey., 2013.
- [12] D. J. Faires and R. Burden. *Numerical Methods*. Thomson Brooks-Cole, California., 2003.
- [13] G. H. Golub and J. M. Ortega. *Scientific Computing and Differential Equations*. Academic Press, Inc, Cambridge, Massachusetts, 1992.
- [14] G. H. Golub and C. F. Van Loan. *Matrix Computations. 4th Edition*. Johns Hopkins University Press, Baltimore., 2013.
- [15] M. T. Heat. *Scientific Computing, An Introductory Survey*. McGraw-Hill Companies, 1997.

- [16] J. Hernadi. *The Wavelet Projection Method for Solving Some Operator Equations*. PhD thesis, Gadjah Mada University, 2004.
- [17] J. Hernadi. *Materi Pelatihan MATLAB*. Jurusan Matematika FMIPA UAD, 2008.
- [18] J. Hernadi. *Teori dan Komputasi Numerik Diferensial dan Integral*. Graha Ilmu, Yogyakarta, 2015.
- [19] J. Hernadi. *Analisis Real Elementer dengan Ilustrasi Numeris dan Grafis*. Erlangga, Jakarta, 2016.
- [20] N. J. Higham. *Accuracy and Stability of Numerical Algorithms*. SIAM, Philadelphia, 2002.
- [21] D. Kincaid and W. Cheney. *Numerical Analysis Mathematics of Scientific Computing*. Brooks/Cole Publishing Company, California, 1991.
- [22] R. Kress. *Numerical Analysis*. Springer-Verlag, New York, 1998.
- [23] McDougall, Trevor & Wotherspoon, and Simon. A simple modification of newton's method to achieve convergence of order 1+2. *Applied Mathematics Letters*., 29:20–25, March 2014.
- [24] R. D. Neidinger. Introduction to automatic differentiation and matlab object-oriented programming. *SIAM Review*, 52(3):545–563, 2010.
- [25] J. Penny and G. Lindfield. *Numerical Methods Using MATLAB*. Prentice Hall, New Jersey, 2000.
- [26] D. L. Powers. *Boundary Value Problems, Fifth Edition*. Academic Press, 2006.
- [27] J. N. Reddy. *Applied Functional Analysis and Variational Methods in Engineering*. McGraw-Hill College, 1989.
- [28] L. F. Shampine, I. Gladwell, and S. Thomson. *Solving ODEs with MATLAB 1st*. Cambridge University Press., 2003.
- [29] L. F. Shampine and M. W. Reichelt. The matlab ode suite. *SIAM Journal on Scientific Computing*, Vol. 18, No. 1., 1997.
- [30] D. S. Watkins. *Fundamental of Matrix Computations*. John Willey & Sons, Inc., New York, 2003.

# Apendiks A

## Pengenalan Matlab

Matlab adalah bahasa pemrograman yang menggabungkan tiga unsur, yaitu komputasi, visualisasi dan pemrograman ke dalam satu wadah. Matlab merupakan singkatan dari *matrix laboratory*. Matlab merupakan sistem interaktif dengan elemen basis data berupa *array*. Artinya setiap data yang dimasukkan pada Matlab diinterpretasikan sebagai *array*. *Array* berdimensi satu adalah vektor dan *array* berdimensi dua adalah matriks. Versi Matlab terus diperbaharui dan sampai saat ini sudah diluncurkan versi R213a sejak awal 2013. Namun versi-versi sebelumnya seperti versi 7 atau versi 6.5 masih tetap sempurna untuk mengerjakan komputasi numerik pada buku ini. Oleh karena itu manual berikut didasarkan pada Matlab versi 6.5. Untuk versi yang lebih baru tidak banyak perbedaan kecuali pada toolbox tambahan.

### Desktop Matlab

Ketika pertama kali kita memulai Matlab, akan muncul desktop Matlab yang memuat panel-panel untuk pengolahan file, variabel dan aplikasi apa pun yang berkaitan dengan Matlab.

Tampilan ini dapat diubah sesuai selera, yaitu dengan cara membuka, menutup, memperbesar atau memperkecil ukuran window yang tersedia. Unsur-unsur desktop Matlab dan fungsinya diberikan sebagai berikut.

- **Command Window** adalah tempat untuk memasukkan variabel dan mengeksekusi program Matlab atau M-files.
- **Command History** berisi rekaman perintah yang pernah ditulis pada *command window*.
- **Launch Pad** berfungsi untuk memudahkan dalam mengakses fasilitas, demonstrasi dan dokumentasi pada Matlab.
- **Help browser** digunakan untuk mencari dan melihat dokumen pada Matlab yang diperlukan untuk bantuan. Untuk mengaktifkan sarana ini klik tombol tandatanya pada toolbar, atau ketik helpbrowser pada *command window*.
- **Current Directory** adalah direktori yang sedang aktif dan dijadikan acuan. Untuk melihat, membuka atau mengedit file-file Matlab kita dapat menggunakan Current Directory browser, atau dengan menggunakan perintah `dir`, `cd`, dan `delete`.

# Indeks

- absis, 31, 33, 196
- akar, 31
- akar ganda, 81
- akar sederhana, 81
- aksesori grafik, 358
- akurasi, 6
- algoritma stabil, 6
- analisis numerik, 4
- angka kelahiran, 35
- approximation solution*, 1
- aproksimasi, 6
- aproksimasi Jacobian, 302
- array*, 10, 337
- array* kosong, 355
- aturan kuadratur, 196
- aturan L'Hospital, 25
- automatic differential*, 184
- automatic differentiation*, 308
  
- B-spline, 141
- B-spline derajat  $k$ , 142
- bagidua, 36
- barisan Cauchy, 41, 321, 328
- basic quadrature*, 230
- basis interior, 158
- basis perbatasan, 160
- bebas linear, 235
- bentuk definit positif, 238
- bentuk diagonal dominan, 238
- bentuk jarang (*sparse*), 238
- bentuk pita, 238
- bentuk simetris, 238
  
- berkondisi buruk, 286
- bersifat interpolasi, 88
- besar vektor (norm), 344
- biaya komputasi, 217, 255
- big-Oh, 16
- bilangan irasional, 9
- bilangan kondisi, 8, 285
- bilangan natural, 22, 25
- bilangan random, 353
- bilangan titik mengambang, 13
- bisection*, 36
- bobot, 196
- built-in*, 342
  
- ceiling*, 39
- chopping*, 13
- composite quadrature*, 230
- computational cost*, 217, 255
- computational error*, 5
- computational science*, 4
- condition number*, 8, 285
  
- decay*, 177
- definit positif, 287
- derajat akurasi, 217
- deret geometri, 327
- deret Taylor, 23
- derivatif, 174
- derivatif berarah, 306
- desimal digit terbatas, 9
- diagonal dominan, 139
- digit signifikan, 10
- dimensi, 345
  
- diskretisasi, 6
- divided-differences*, 97
  
- editor grafik, 358
- eksak, 6
- eksistensi penyelesaian, 235
- eksponen, 10, 11
- ekstrapolasi Richardson, 230
- elemen per elemen, 344
- epsilon mesin, 15
- error*, 7
- estimasi galat, 175
- estimasi galat akhir, 322
- estimasi galat awal, 322
- estimasi parameter, 234
- exact solution*, 1
  
- faktorial, 20
- faktorisasi Cholesky, 287
- farmakokinetika, 4
- fixed point*, 65
- fixed-point iteration*, 36
- floating*, 12
- floating-point number system*, 10
- flops*, 255
- format long**, 78
- formula eksplisit, 173
- formula kuadratur, 196
- formula selisih bagi Newton, 99
- formula Stirling, 20
- fungsi basis, 148
- fungsi iterasi, 65

- tali busur, 41, 174  
teorema konvergensi jepit, 62  
teorema lokasi akar, 34  
teorema nilai antara, 179  
teorema nilai rerata, 60  
teorema Taylor order satu,  
61  
teorema titik tetap Banach,  
321, 328  
**terdiferensial**, 175
- titik tengah interval, 36  
titik tetap, 65, 321  
TNR, 60  
TNR FUNGSI VEKTOR, 323  
transformasi variabel, 94, 219  
translasi dan kompresi, 148  
transpose, 350  
trapesium, 199  
trapesium bersusun, 210  
ukuran langkah, 181
- underflow*, 12  
*unit roundoff*, 15  
van der Monde, 125  
variabel keputusan, 123  
vektor, 344  
vektor nol, 235  
vektor penyelesaian, 234
- well-conditioned*, 6  
*well-posedness*, 1