METODE NUMERIK

Edisi Jurusan T. Informatika - ITS

Irfan Subakti - 司馬伊凡

IRFAN SUBAKTI, M.SC.ENG.
司馬伊凡，碩士學位

Jurusan Teknik Informatika
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember
Surabaya
2006

Irfan Subakti - 司馬伊凡

Aku persembahkan buku ini buat:

Ibu, Bapak dan Adik-adikku tercinta

Irfan Subakti - 司馬伊凡

Irfan Subakti - 司馬伊凡

KATA PENGANTAR

Salam sejahtera ^__^

Modul Metode Numerik ini dibuat dalam rangka membantu para mahasiswa memahami mata kuliah Metode Numerik (CI1414) di Jurusan Teknik Informatika, Fakultas Teknologi Informasi, Institut Teknologi Sepuluh Nopember.

Kuliah Metode Numerik ini diberikan sebagai salah satu Mata Kuliah Wajib yang memiliki bobot 3 SKS (Satuan Kredit Semester).

Tujuan yang ingin didapat dari Mata Kuliah ini adalah untuk memahami konsep dasar metode numerik, kelebihan dan kekurangan masing-masing metode numerik dibandingkan dengan metode lainnya, serta ketepatan hasil dan penerapannya.

Materi yang dibahas adalah: sistem bilangan dan kesalahan numerik, eliminasi Gauss, metode Gauss-Jordan, matriks inversi dan Gauss Seidel, metode grafik, metode biseksi, metode Regula Falsi, metode iterasi, metode Newton-Raphson, Secant dan akar ganda, regresi linier, regresi polinomial dan berganda, interpolasi linier dan kuadratik, interpolasi Newton, interpolasi Lagrange, interpolasi Spline, integrasi Newton-Cotes, solusi diferensial Euler, serta solusi diferensial Range-Kutta.

Semoga apa yang ada di buku ini bermanfaat bagi pembaca semua. Tentu saja, kritik dan saran dipersilakan. Pembaca dapat menghubungi penulis di: is@its-sby.edu

Surabaya, Juli 2006

Penulis

Irfan Subakti - 印章

Irfan Subakti - 司馬伊凡

DAFTAR ISI

Halaman

Irfan Subakti 司馬伊凡

Irfan Subakti 司馬伊凡

DAFTAR TABEL

Halaman

Irfan Subakti - 司馬伊凡

Irfan Subakti - 司馬伊凡

DAFTAR GAMBAR

Halaman

Irfan Subakti 司馬伊凡

司馬伊凡
Irfan Subakti

Bab 1 Pendahuluan Kuliah

1.1 Gambaran Umum

Metode Numerik merupakan Mata Kuliah Wajib yang diajarkan kepada mahasiswa pada Jurusan Teknik Informatika, Fakultas Teknologi Informasi, Institut Teknologi Sepuluh Nopember.

Kredit untuk mata kuliah ini adalah 3 SKS (Satuan Kredit Semester), yang artinya adalah 50 menit × 3 = 150 menit tatap muka.

Prasyarat untuk mengikuti mata kuliah ini adalah mata kuliah Kalkulus 1 (UM1201), Kalkulus 2 (UM1202) dan Aljabar Linier (CI1402).

Tujuan yang ingin didapat dari Mata Kuliah ini adalah untuk memahami konsep dasar metode numerik, kelebihan dan kekurangan masing-masing metode numerik dibandingkan dengan metode lainnya, serta ketepatan hasil dan penerapannya.

Irfan Subakti - 司馬仲元

1.2 Tujuan

Tujuan yang ingin dicapai setelah mengikuti mata kuliah Metode Numerik ini adalah:

- Mahasiswa memahami pengertian dasar metode numerik
- Mahasiswa memahami kelebihan dan kekurangan masing-masing metode numerik
- Mahasiswa dapat mencari akar-akar persamaan
- Mahasiswa dapat menyelesaikan persoalan persamaan linier
- Mahasiswa dapat membuat formula dari data-data yang ada

1.3 Topik yang Dibahas

Topik-topik yang dibahas dalam mata kuliah Metode Numerik ini adalah sebagai berikut:

- Peraturan kuliah, silabus/materi, aturan penilaian, dll.
- Pendekatan dan kesalahan
- Angka signifikan

- Akurasi dan presisi
- Definisi kesalahan
- Akar-akar persamaan → metode Akolade (Bracketing method)
- Metode Grafik
- Metode Bagidua
- Metode Posisi Salah
- Akar-akar persamaan → metode Terbuka
- Iterasi satu titik sederhana
- Metode Newton-Raphson
- Metode Secant
- Metode Newton-Raphson yang dimodifikasi
- Sistem persamaan aljabar linier
- Eliminasi Gauss
- Eliminasi Gauss Jordan
- Gauss Jordan
- Gauss Seidell
- Determinan
- Invers
- Pencocokan kurva: regresi kuadrat terkecil
- Regresi linier
- Regresi polinomial
- Regresi linier berganda
- Regresi non linier
- Pencocokan kurva: Interpolasi
- Polinomial interpolasi diferensia terbagi Newton
- Polinomial interpolasi Lagrange

Irfan Subakti - 司馬伊凡

- Interpolasi Spline
- Integrasi: formulasi integrasi Newton-Cotes
- Aturan Trapesium
- Aturan Simpson
- Integrasi dengan segmen tidak sama
- Integrasi terbuka
- Multiple integrasi
- Integrasi: integrasi Romberg dan kuadratur Gauss
- Integrasi Romberg
- Kuadratur Gauss
- Improper integral
- Persamaan diferensia biasa: Runge-Kutta
- Metode Euler
- Modifikasi dan perbaikan metode Euler
- Metode Runge-Kutta
- Sistem persamaan
- Nilai batasan dan permasalahan nilai Eigen
- Metode umum untuk permasalahan nilai batasan
- Permasalahan nilai Eigen
- Diferensia terbatas: persamaan Elliptic
- Persamaan Laplace
- Teknik-teknik solusi
- Kondisi-kondisi batasan
- Pendekatan control-volume

Irfan Subakti - 司馬伊凡

1.4 Prasyarat

Prasyarat untuk mengikuti mata kuliah Metode Numerik ini adalah sebagai berikut:

- Kalkulus 1 (UM1201)
- Kalkulus 2 (UM1202)
- Aljabar Linier (CI1402)

1.5 Pustaka Acuan

Pustaka yang dipakai sebagai acuan dalam mata kuliah Metode Numerik ini dapat dilihat pada Daftar Pustaka di bagian terakhir modul ini.

Irfan Subakti - 司馬伊凡

PROLOG

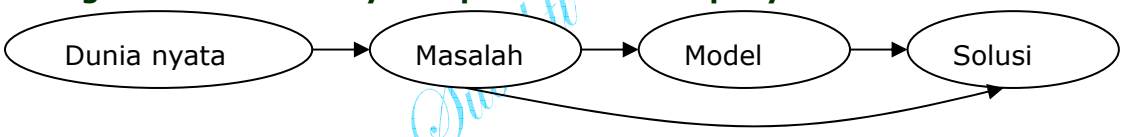
Metode Numerik adalah: Teknik dimana masalah matematika diformulasikan sedemikian rupa sehingga dapat diselesaikan oleh pengoperasian Aritmetika.

Dengan adanya perkembangan komputer sekarang ini, maka kalkulasi Aritmetika yang banyak dan menjenuhkan bila dikerjakan secara manual akan menjadi lebih mudah dan menyenangkan. ☺

Metode Prakomputer. 3 pendekatan penyelesaian masalah teknik:

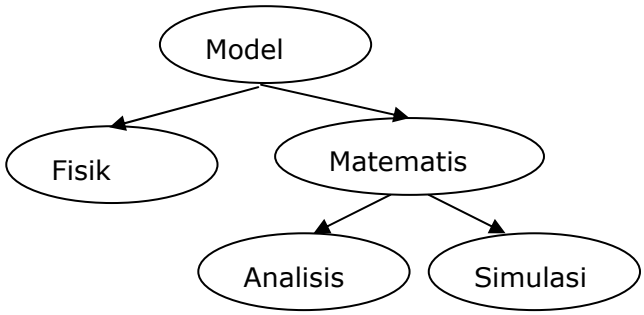
1. Menggunakan cara analitis atau eksak. Hanya bisa untuk masalah yang sederhana.
2. Menggunakan grafik. Terbatas untuk 2, 3 dimensi saja.
3. Kalkulator. Kesulitan dalam mengatasi kekeliruan pemakai.

Hubungan antara dunia nyata - permasalahan – penyelesaian



Irfan Subakti

Dunia nyata → Model (untuk menghemat waktu, biaya, dan mengurangi resiko)



Sehingga MetNum itu sendiri sebenarnya adalah cara penyelesaian Matematis, yang dikembangkan dari cara analisis, dan memasuki wilayah simulasi. Simulasi dilangsungkan dengan menggunakan media komputer.

Irfan Subakti - 司馬伊凡

BAB 1 MODEL MATEMATIKA

Seperti telah dibahas dalam prolog sebelumnya, model dibuat untuk memudahkan orang dalam menganalisis suatu permasalahan, disamping untuk menghemat waktu, biaya, dan juga mengurangi resiko. Dengan adanya sistem komputer yang demikian canggih saat ini, maka pemodelan ini menjadi lebih mudah dan nyaman dilakukan. Dari sini lahirlah simulasi yang menggunakan komputer untuk menirukan hal-hal yang ada di dunia nyata, yang dapat dianalisis, dievaluasi dan didapatkan hasilnya, serta dapat diulangi kapanpun dengan hasil yang sama.

1.1 MODEL MATEMATIKA

Adalah persamaan yang mengungkapkan segi utama sistem/proses fisika dalam istilah matematika.

Contoh kita disini adalah hukum Newton II: Laju waktu perubahan momentum sebuah benda sama dengan gaya resultan yang bekerja padanya.

$$F = ma \qquad [1.1]$$

Dimana F: gaya total yang bekerja pada benda (dyne atau gr-cm/detik)

m: masa benda (gr)

a: percepatan (cm/detik2)

Arti persamaan 1.1:

- Menjelaskan suatu proses/sistem alami dalam istilah matematika
- Menunjukkan penyederhanaan dari kenyataan
- Memberikan hasil-hasil yang dapat ditiru sehingga dapat dipakai untuk tujuan perkiraan

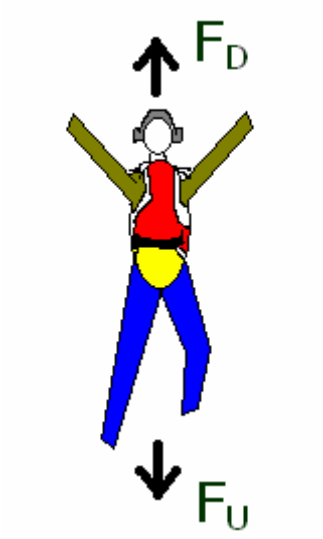
$$a = \frac{F}{m} \qquad [1.2]$$

Percepatan sendiri adalah: laju waktu dari perubahan kecepatan (dv/dt), sehingga:

$$m\frac{dv}{dt} = F \qquad [1.3]$$

Dimana v: kecepatan (cm/detik)

Bila gaya (F) positif, benda dipercepat, kalau negatif, maka benda diperlambat, jika 0 maka kecepatan benda akan konstan.



Gambar 1.1 Diagram skema gaya-gaya yang bekerja pada penerjun payung

Pada gambar 1.1 di atas, F_D adalah gaya ke bawah disebabkan tarikan gravitasi. F_U adalah gaya ke atas disebabkan gesekan udara.

Jika benda jatuh di sekitar permukaan bumi, gaya netto terdiri dari dua gaya yang berlawanan, yaitu F_D dan F_U.

$$F = F_D + F_U \qquad [1.4]$$

$F_D = mg$ g adalah konstanta gravitasi = 980 cm/det2 [1.5]

$F_U = -cv$ c adalah konstanta perbandingan (koefisien tahanan, gram/detik) [1.6]

$$m\frac{dv}{dt} = mg - cv \qquad [1.7]$$

Jika tiap suku di bagi m, maka

$$\frac{dv}{dt} = g - \frac{c}{m}v$$ yang merupakan persamaan diferensial (adanya dv/dt) [1.8]

Solusi eksak dari persamaan penerjun bebas ini (penerjun pada mulanya diam v=0, t=0)

adalah:

$$v(t) = \frac{gm}{c}\left[1 - e^{-(c/m)\,t}\right] \qquad [1.9]$$

Contoh solusi analitis/eksak

Penerjun dengan massa 68.100 gr meloncat dari sebuah pesawat terbang. Dengan persamaan eksak di atas, hitung kecepatan sebelum penerjun membuka payungnya. Koefisien tahanan/geser *c* kira-kira besarnya 12.500 gr/det.

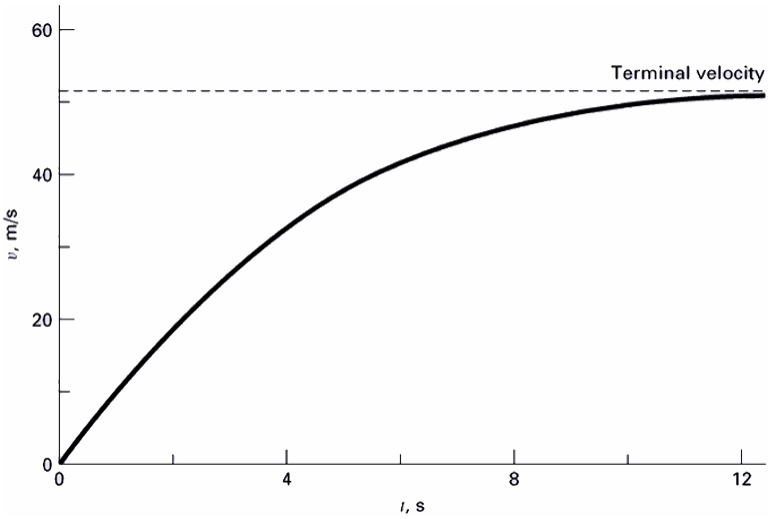
$$V(t) = \frac{980(68.100)}{12.500}\left[1 - e^{-(12.500/68.100)\,t}\right]$$

$$= 5.339{,}0\,(1 - e^{-0{,}18355t})$$ yang dapat digunakan untuk menghitung:

Tabel 1.1 Waktu dan kecepatan penerjuan payung untuk solusi analitis

t (det)	v (cm/det)
0	0,00
2	1.640,5
4	2.776,9
6	3.564,2
8	4.109,5
10	4.487,3
12	4.749,0
∞	5.339,0

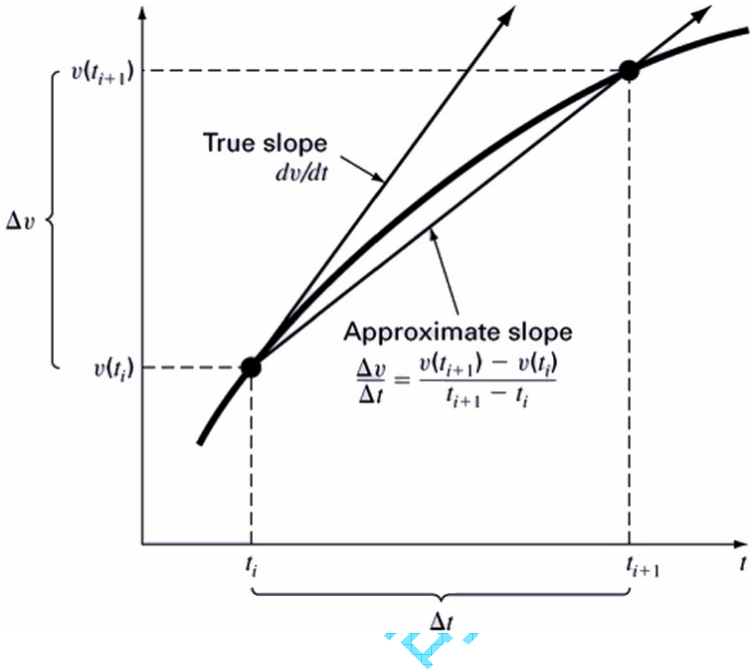
Dimana tabel 1.1 di atas dalam bentuk grafik dapat digambarkan seperti gambar 1.2 di bawah ini.



Gambar 1.2 Grafik dari hasil perhitungan analitis

Sayangnya dalam kenyataan sehari-hari, amat sulit mendapatkan solusi eksak, alternatifnya adalah solusi numerik yang mendekati solusi eksak.

Dalam kasus penerjun bebas ini, disadari bahwa laju perubahan kecepatan terhadap waktu dapat didekati dengan gambar 1.3 berikut ini.



Gambar 1.3. Grafik penggunaan diferensiasi hingga untuk mengaproksimasi turunan pertama v thd t

$$\frac{dv}{dt} \approx \frac{\Delta v}{\Delta t} = \frac{v(t_{i+1}) - v(t_i)}{t_{i+1} - t_i}$$ yang disebut diferensiasi terbagi hingga [1.10]

Dimana Δv = beda kecepatan

Δt = beda waktu

$v(t_i)$ = kecepatan pada awal t_i

$v(t_{i+1})$ = kecepatan beberapa saat t_{i+1} berikutnya

$$\frac{v(t_{i+1}) - v(t_i)}{t_{i+1} - t_i} = g - \frac{c}{m} v(t_i)$$ yang dapat disusun kembali menjadi: [1.11]

$$v(t_{i+1}) = v(t_i) + \left[g - \frac{c}{m} v(t_i) \right] (t_{i+1} - t_i)$$ [1.12]

Yaitu: (harga baru v) = (harga lama v)+ (nilai taksiran slope) × (interval waktu) [1.13]

Contoh solusi numerik

Kasusnya sama dengan untuk solusi eksak di atas, penerjun dengan massa 68.100 gr meloncat dari sebuah pesawat terbang. Koefisien tahanan/geser c kira-kira besarnya 12.500 gr/det. Interval waktunya = 2 detik. Hitung v(t) nya.

Pada saat mulai perhitungan ($t_i = 0$), kecepatan penerjun $v(t_i) = 0$. Pada $t_{i+1} = 2$ detik:

$$v(2) = 0 + \left[980 - \frac{12.500}{68.100}(0)\right]2$$

$$= 1.960 \text{ cm/det.}$$

Untuk interval berikutnya (dari t = 2 ke 4 detik), perhitungan diulang lagi dengan hasil:

$$v(4) = 1.960 + \left[980 - \frac{12.500}{68.100}(1.960)\right]2$$

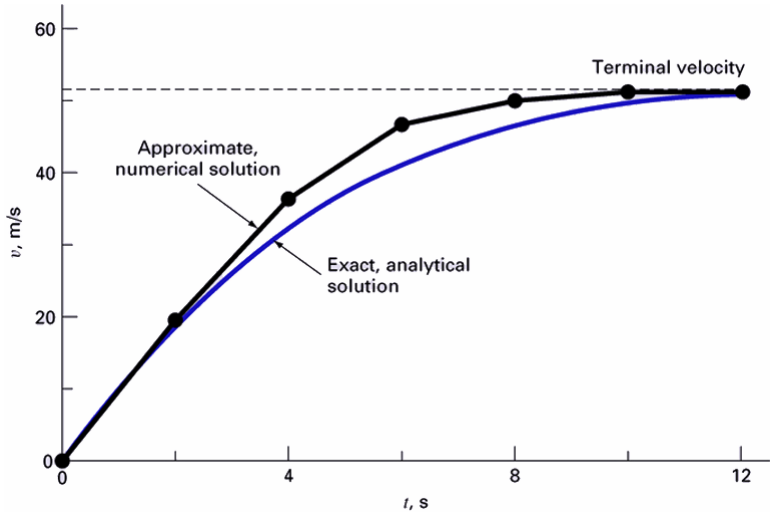
= 3.200,5 cm/det. Dan hal ini diteruskan, sehingga didapat:

Tabel 1.2 Waktu dan kecepatan penerjuan payung untuk solusi numerik

t (det)	v (cm/det)
0	0,00
2	1.960,0
4	3.200,5
6	3.985,6
8	4.482,5
10	4.796,9
12	4.995,9
∞	5.339,0

Irfan Subakti

Dimana tabel 1.2 di atas dalam bentuk grafik dapat digambarkan seperti gambar 1.4 di bawah ini.



Gambar 1.4. Grafik perbandingan solusi numerik dengan solusi analitis

Dengan komputer, hasil perhitungan dengan solusi numerik dapat ditingkatkan ketepatannya dengan cara memperkecil interval waktunya. Setiap pembagian setengah interval, agar mencapai ketelitian yang lebih baik, mengakibatkan terjadinya kelipatan jumlah perhitungan. Jadi memang ada trade-off (kompromi) antara kecepatan dan upaya perhitungan.

Tugas 1

Buatlah makalah berdasarkan dua contoh di atas (solusi analitis/eksak dan solusi numerik/pendekatan), gunakan interval waktu 2, 1, dan 0.5 detik untuk solusi numerik. Makalah harus lengkap, mulai cover, masalah, algoritma, program komputer, evaluasi dan hasilnya.

Irfan Subakti - 司馬伊凡

BAB 2 KOMPUTER DAN PERANGKAT LUNAK

Segala hal mengenai komputer dan software, sebenarnya telah dibahas dalam mata kuliah PTI (Pengantar Teknologi Informasi), Bahasa Pemrograman, dan lain-lain mata kuliah, seperti:

- Sejarah komputer, sejak jaman mainframe sampai microcomputer dewasa ini.
- Pengembangan software, meliputi gaya pemrograman (desain algoritma, komposisi program, debugging dan testing, dokumentasi, storage dan maintenance), desain yang bersifat modular, Top-Down Design, pemrograman terstruktur.

Namun demikian agar memberikan pemahaman menyeluruh kepada kita semua, materi mengenai komputer dan perangkat lunak ini tetap akan kita bahas disini sekilas saja.

2.1 PENDAHULUAN

Komputer adalah partisipan dalam interaksi yang menjalankan program

- Merupakan frasa umum, mencakup banyak peralatan interaktif – saklar cahaya, mobil, dan lain-lain.
- Kita harus lebih memperhatikan komputer elektronik

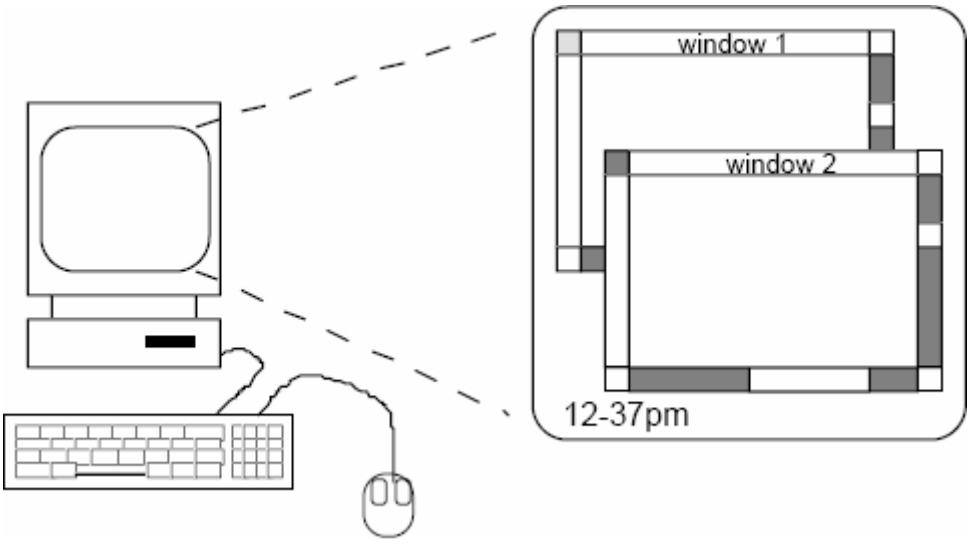
Ada 2 bentuk perbedaan utama interaksi

- *Batch* – biasanya jika sejumlah besar data harus dibaca/diproses dalam mesin; membutuhkan hanya sedikit intervensi/campur tangan pengguna
- Interaktif – saat pengguna mengontrol sesuatu di sepanjang waktu

Konsentrasi pada penggunaan interaktif

Irfan Subakti - 司馬伊凡

2.2 Sistem Komputer Secara Umum



Gambar 2.1 Sistem Komputer Secara Umum

Seperti terlihat pada gambar 2.1 di atas, maka sistem komputer memiliki beberapa peralatan seperti:

- Layar (*screen*), atau monitor, dimana disitu terdapat beberapa
  - *Windows* – area-area berbeda yang berjalan secara mandiri satu dengan lainnya
- *Keyboard*
- *Mouse*

Peralatan-peralatan ini menentukan gaya interaksi yang didukung oleh sistem

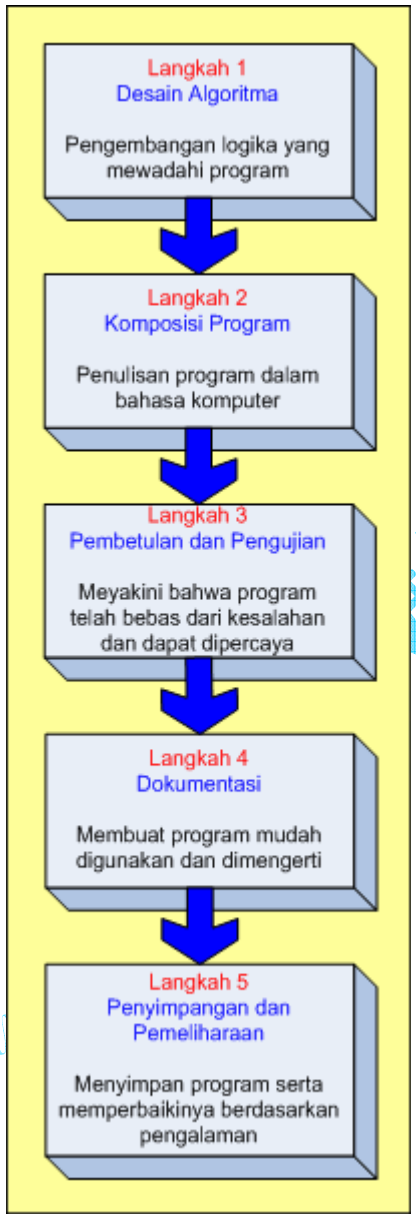
Jika kita menggunakan peralatan yang berbeda, maka antarmuka akan mendukung gaya interaksi yang berbeda

2.3 Pengembangan Perangkat Lunak

Dalam bab ini materi disusun dalam lima langkah, seperti terlihat pada gambar 2.2 di bawah ini. Hal ini dibutuhkan untuk membuat dan membina perangkat lunak berkualitas tinggi.

Bab ini memuat memuat penjelasan dari setiap langkah-langkah tersebut. Materi ini disusun oleh sebuah studi kasus di mana langkah-langkah tersebut diterapkan pada pengembangan perangkat lunak untuk masalah penerjun bebas.

Setelah memahami materi ini, kita sebaiknya mempersiapkan diri lebih baik guna mengembangkan perangkat lunak berkualitas tinggi bagi metode selanjutnya.



Gambar 2.2 Lima langkah yang diperlukan untuk membuat dan memelihara perangkat lunak

2.3.1 Desain Algoritma

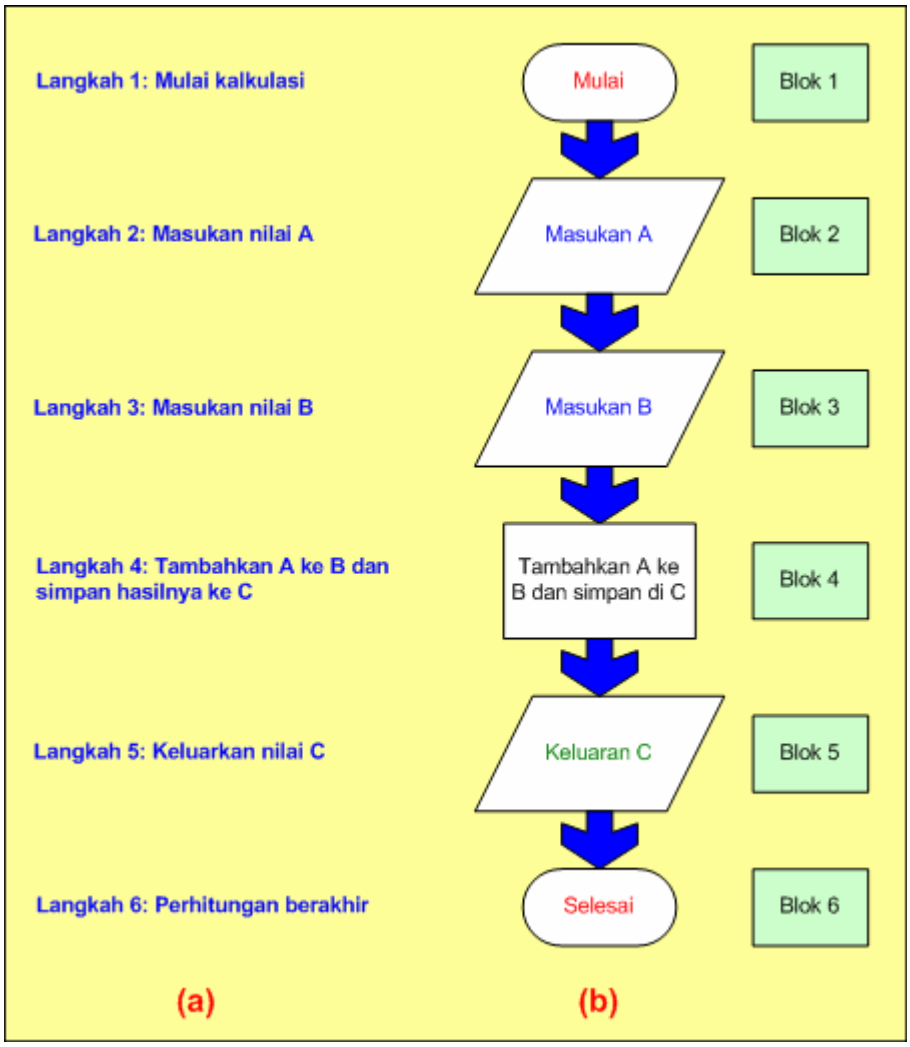
Proses pengembangan program komputer dapat kita mulai pada tahap ini. Sebuah program adalah sekumpulan instruksi belaka bagi komputer. Semua program yang dibutuhkan untuk dijalankan pada sebuah komputer secara bersama-sama disebut dengan perangkat lunak.

Sebuah algoritma adalah urutan langkah-langkah logika yang dibutuhkan guna melakukan suatu tugas terperinci/tertentu seperti halnya penyelesaian suatu masalah. Algoritma yang baik memiliki sejumlah sifat. Algoritma harus selalu berakhir setelah sejumlah langkah

berhingga dan seharusnya cukup umum dalam menghadapi segala jenis kemungkinan.

Algoritma yang baik harus dapat ditentukan/deterministik. Yaitu tidak satu pun dari kesempatan yang ada sampai ada yang tertinggal. Hasil akhir tidak tergantung pada siapa yang mengikuti algoritma tersebut. Dalam pengertian ini, suatu algoritma dianalogikan dengan sebuah resep. Dua koki bekerja sendiri-sendiri memakai suatu resep yang baik, dan akhirnya masing-masing dari mereka dapat menyajikan masakan yang identik.

Gambar 2.3 berikut ini memperlihatkan suatu algoritma untuk menyelesaikan masalah sederhana dari penambahan dua bilangan. Dua orang pemrogram yang bekerja dengan algoritma ini mungkin saja mengembangkan program dengan gaya yang berlainan, tetapi dari data yang sama, program harus memberikan hasil yang sama.



Gambar 2.3 (a) algoritma dan (b) diagram alir untuk pemecahan masalah penambahan sederhana

Suatu alternatif untuk menyatakan sebuah algoritma adalah dengan sebuah diagram alir (flowchart). Ini merupakan pernyataan algoritma secara visual atau grafis dengan menggunakan sederetan blok dan tanda panah. Setiap blok menunjukkan suatu

pengoperasian tertentu atau langkah dalam algoritma. Tanda panah menunjukkan arah urutan operasi yang akan dilaksanakan. Contoh adalah gambar 2.3 (b) di atas.

2.3.2 Komposisi Program

Setelah algoritma telah kita susun, maka selanjutnya adalah menyatakan algoritma sebagai suatu urutan pernyataan program yang disebut kode.

Menahan keinginan untuk menulis kode penting kita lakukan sebelum keseluruhan ruang lingkup masalah terdefinisi secara jelas dan teknik pemecahan dan algoritma dirancang secara hati-hati.

Masalah yang paling umum dihadapi oleh pemrograman yang belum berpengalaman biasanya suatu kode yang persiapannya ditemukan belum matang yaitu tidak mencakup suatu strategi atau rancangan menyeluruh.

Setelah suatu algoritma yang baik dirancang, kode ditulis dalam suatu bahasa pemrograman tingkat tinggi. Ratusan bahasa tingkat tinggi telah dikembangkan sejak dimulainya abad komputer. Contoh adalah: bahasa C, Pascal, Java, C# dan lain-lain.

6 elemen utama pemrograman yang langsung berkaitan dengan metode numerik adalah sebagai berikut:

- Konstanta dan variabel
- Masukan-keluaran (input-output)
- Komputasi
- Kontrol
- Subprogram
- Dokumentasi

Irfan Subakti - 司馬特凡

2.3.3 Debugging dan Testing

Setelah menulis kode program, kita harus menguji program tersebut terhadap kesalahan-kesalahan yang disebut dengan bug.

Proses mencari kesalahan dan membetulkannya dinamakan dengan debugging.

Beberapa jenis kesalahan dapat terjadi saat pemrograman dalam sembarang bahasa. Kesalahan sintaks yang menyalahi aturan bahasa, seperti pernyataan, formasi bilangan,

baris bilangan dan konvensi lainnya. Dan kesalahan ini spesifik terjadinya untuk tiap bahasa pemrograman. Seringkali kesalahan ini adalah akibat dari salah ketik.

Kesalahan yang sukar dideteksi adalah kesalahan yang berhubungan dengan konstruksi dan logika program (disebut dengan kesalahan semantik) yang dapat terjadi tanpa adanya kesalahan sintaks.

Pembetulan dan pengujian program dimungkinkan dengan penggunaan gaya pengkodean yang baik. Hal ini mencakup desain program sedemikian rupa sehingga program terdiri dari beberapa bagian kecil. Jenis gaya pemrograman ini disebut dengan pemrograman modular.

2.3.4 Dokumentasi

Setelah program bersih dari kesalahan dan telah diuji, program harus didokumentasikan.

Dokumentasi adalah penambahan komentar yang memberi petunjuk bagi pemakai untuk menjalankan program secara lebih mudah.

Terdapat 2 jenis aspek dokumentasi, yaitu:

- Aspek internal. Dokumentasi internal terdiri dari diskusi atau penjelasan yang dimasukkan selama pengkodean program guna menjelaskan bagaimana tahapan-tahapan setiap program bekerja. Nama mnemonic untuk variabel bisa dipakai untuk lebih memberikan kejelasan dokumentasi program dan program itu sendiri.
- Aspek eksternal. Ini berhubungan dengan instruksi-instruksi dalam bentuk message (pesan) dan bahan pendukung tercetak yang didesain untuk membantu pemakai sewaktu menjalankan perangkat lunak kita.

Irfan Subakti - 司馬伊凡

2.3.5 Penyimpanan dan Pemeliharaan

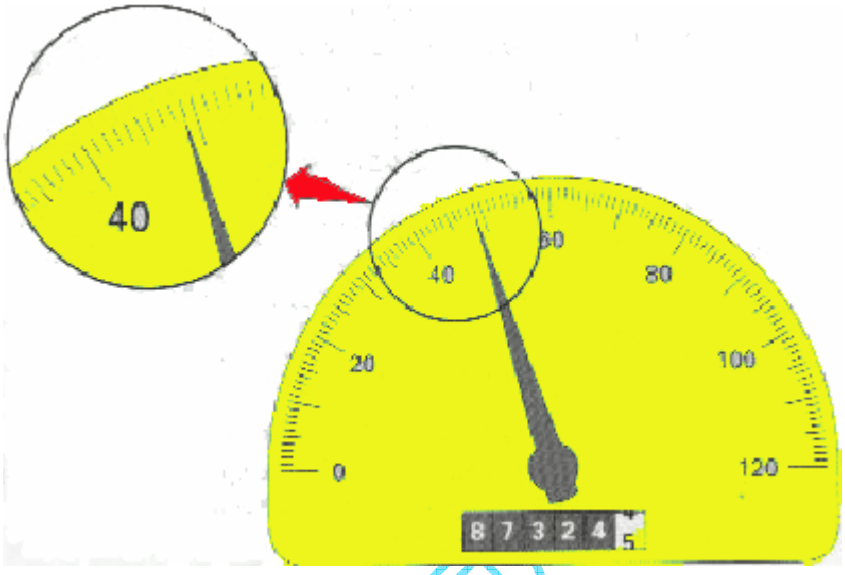
Langkah akhir dalam pengembangan program adalah penyimpanan dan pemeliharaan.

Pemeliharaan meliputi perbaikan atau perubahan yang dibuat dalam program yang mencakup penerapan terhadap masalah nyata. Dalam waktu yang lama, perubahan demikian dapat membuat program lebih mudah dipakai atau lebih dapat diterapkan untuk jenis permasalahan yang lebih besar. Pemeliharaan dimungkinkan dengan adanya dokumentasi yang baik.

Penyimpanan berhubungan dengan cara dimana perangkat lunak disimpan untuk penggunaan lebih lanjut.

BAB 3 APROKSIMASI DAN ROUND-OFF ERROR

3.1 ANGKA SIGNIFIKAN



Gambar 3.1 Speedometer dan Odometer

Dari gambar 3.1 diatas, bila dilihat maka dari speedometer menunjukkan mobil berjalan dengan kecepatan 48 atau 49 km/jam. Mungkin lebih tepatnya sekita 49 km/jam. Jika diinginkan 1 angka dibelakang koma, maka kita masih bisa memperkirakan kira-kira nilainya 48,7 atau mungkin 48,8 km/jam. Adanya keterbatasan speedometer tadi menyebabkan kita tak dapat memastikan, berarti menduga saja, untuk digit ketiga (digit kedua di belakang koma). Jadi menggelikan sekali kalo dapat diperkirakan bahwa kecepatan mobil itu 48,8642138 km/jam.

Angka signifikan atau digit menyatakan suatu keandalan sebuah nilai numerik. Banyaknya angka signifikan adalah banyaknya digit tertentu yang dapat meyakinkan kita. Untuk speedometer di atas, maka mengandung taksiran 3 angka signifikan. Odometer memiliki taksiran 7 angka signifikan.

Beberapa angka 0 tak selamanya angka signifikan, karena mereka diperlakukan sekedar menempatkan sebuah titik desimal. Jadi bilangan-bilangan 0,00001845 lalu 0,0001845 lalu 0,001845 semuanya memiliki 4 angka signifikan.

Jika beberapa angka 0 dipakai di bagian ekor suatu bilangan, tak jelas berapa banyaknya 0 itu yang signifikan. Misal: 45,300 dapat memiliki 3, 4, atau 5 buah digit signifikan tergantung apakah harga 0 itu telah diketahui dengan pasti. Ketidakpastian itu dapat

diselesaikan dengan memakai notasi ilmiah dimana $4{,}53 \times 10^4$ atau $4{,}530 \times 10^4$ dan $4{,}5300 \times 10^4$ menandakan bahwa angka-angka tersebut memiliki 3, 4, dan 5 angka signifikan.

Implikasi dari angka signifikan:

- MetNum mengandung hasil pendekatan. Keyakinannya ditentukan oleh angka signifikan.
- Pernyataan secara eksak besaran-besaran yang signifikan seperti π, dibatasi oleh tipe data yang dapat disimpan oleh komputer sampai sejumlah digit tertentu, selebihnya diabaikan. Pengabaian ini dinamakan dengan kesalahan pembulatan (round-off error).

3.2 Akurasi dan Presisi

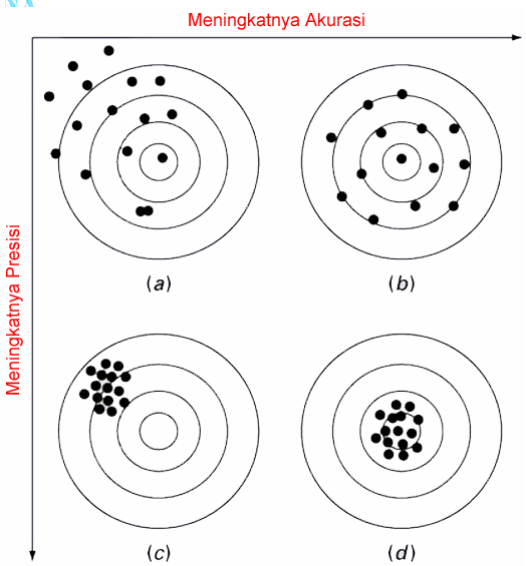
Akurasi mengacu pada seberapa dekat angka pendekatan/pengukuran terhadap harga sebenarnya.

Presisi mengacu pada:

- Jumlah angka signifikan yang menyatakan suatu besaran
- Penyebaran dari nilai-nilai yang terbaca dari suatu alat ukur

Gambar 3.2 di bawah ini disajikan untuk lebih memperjelas penggambaran dari pengertian akurasi dan presisi di atas.

Irfan Subakti 司馬伊凡



Gambar 3.2 Gambaran presisi dan akurasi

3.3 Definisi Error (Kesalahan)

Timbul dari penggunaan aproksimasi. Meliputi 2 hal, yaitu:

- **Kesalahan pemotongan** (truncation error), dihasilkan sewaktu aproksimasi digunakan untuk menyatakan suatu prosedur matematika eksak.
- **Kesalahan pembulatan** (round-off error), dihasilkan bila angka-angka aproksimasi dipakai untuk menyatakan angka-angka eksak.

Adfadsfa alkdj faljf lads

Harga sebenarnya = aproksimasi + error [3.1]

E_t = harga sebenarnya – aproksimasi [3.2]

Dimana E_t = harga pasti error, dengan t berarti true.

Bila besaran diperhitungkan dengan menormalisasikan error terhadap harga sebenarnya:

$$\text{Kesalahan relatif fraksional} = \frac{kesalahan}{harga\ sebenarnya}$$

Bila dinyatakan dalam persentase:

$$\epsilon_t = \frac{error\ sebenarnya}{harga\ sebenarnya} 100\% \quad [3.3]$$

Dimana ϵ_t = error relatif persen sebenarnya

Irfan Subakti - 司馬伊凡

Contoh perhitungan error

Terdapat tugas untuk mengukur panjang sebuah jembatan dan sebuah paku keling. Didapat harga 9.999 dan 9 cm. Kalau harga sebenarnya adalah 10.000 dan 10 cm, maka hitunglah (a) error (b) error relatif persen, untuk setiap kasus.

(a) Untuk jembatan $E_t = 10.000 - 9.999 = 1$ cm

Untuk paku keling $E_t = 10 - 9 = 1$ cm

(b) Untuk jembatan $\epsilon_t = \frac{1}{10.000} 100\% = 0{,}01\%$

Untuk paku keling $\epsilon_t = \frac{1}{10} 100\% = 10\%$

Jadi walau sama-sama error 1 cm, tapi pengukuran dikatakan lebih baik untuk jembatan.

Terdapat notasi E_t dan $\in_t$ (dimana t berarti true, menandakan bahwa error diaproksimasi terhadap harga sebenarnya), tapi dalam kenyataannya, amat jarang kita bisa mengetahui harga sesungguhnya ini (bisa juga, kalau digunakan fungsi yang dapat diselesaikan secara analitis). Apalagi dalam kenyataan sehari-hari, sulit sekali mengetahui nilai eksak dalam pelbagai kasus. Maka dari itu terdapat alternatif untuk menormalisasi error dengan menggunakan taksiran terbaik dari harga sebenarnya, yaitu:

$$\in_a = \frac{error\ aproksimasi}{aproksimasi} 100\% \qquad [3.4]$$

dengan a menandakan aproksimasi.

Taksiran kesalahan ditentukan tanpa pengetahuan mengenai harga sebenarnya. Misalnya MetNum tertentu memakai pendekatan iterasi untuk menghitung jawaban. Maka aproksimasinya dibuat berdasarkan suatu aproksimasi sebelumnya, dan proses ini dilakukan secara berulang.

$$\in_a = \frac{aproksimasi\ sekarang - aproksimasi\ sebelumnya}{aproksimasi\ sekarang} 100\% \qquad [3.5]$$

Proses iterasi/perulangan akan berakhir pada suatu nilai $\in_s$, yaitu persentase toleransi praspesifikasi.

$$|\in_a| < \in_s \qquad [3.6]$$

Irfan Subakti - 司馬特凡

Hubungan error dengan jumlah angka signifikan

Jika kriteria berikut dipenuhi, dapat dijamin bahwa hasilnya adalah betul hingga sekurang-kurangnya n angka signifikan [Scarborough, 1966]:

$$\in_s = (0{,}5) \times 10^{2-n}) \% \qquad [3.7]$$

Contoh taksiran error untuk metode iterasi

Fungsi eksponensial dapat dihitung dengan:

$$e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!} + \ldots \text{ (contoh perluasan deret MacLaurin)} \qquad [C3.2.1]$$

Semakin banyak suku yang ditambahkan dalam deret, aproksimasi akan lebih baik.

Harga sebenarnya dari $e^{0,5}$ = 1,648721271.

Suku pertama: e^x = 1, lalu dilanjutkan dengan suku-suku berikutnya.

Hasil diatur agar sekurang-kurangnya memiliki 3 angka signifikan:

$$\epsilon_s = (0{,}5) \times 10^{2-3}) \% = 0{,}5\%$$

Dimulai dengan $e^x \approx 1 + x$,

untuk $x = 0{,}5 \rightarrow e^{0{,}5} \approx 1 + 0{,}5 = 1{,}5$

$$\epsilon_t = \frac{1{,}648721271 - 1{,}5}{1{,}648721271} \ 100\% = 9{,}02\%$$

$$\epsilon_a = \frac{1{,}5 - 1}{1{,}5} 100\% = 33{,}3\%$$

Ternyata $\epsilon_a \geq \epsilon_s$ maka komputasi dilanjutkan dengan menambahkan suku lainnya → $x^2/2!$, dan akan terus dilanjutkan sampai $\epsilon_a < \epsilon_s$.

Tabel 3.1 Hasil perhitungan ϵ_t dan ϵ_a

Suku	Hasil	ϵ_t %	ϵ_a %
1	1	39,3	
2	1,5	9,02	33,3
3	1,625	1,44	7,69
4	1,645833333	0,175	1,27
5	1,648437500	0,0172	0,158
6	1,648697917	0,00142	0,0158

Irfan Subakti 司馬伊凡

Ternyata cuma butuh 6 suku saja, sehingga kesalahan taksiran kurang dari $\epsilon_s = 0{,}5\%$. Dan juga didapatkan bahwa hasilnya akurat sampai 5 angka signifikan (tidak hanya 3).

3.4 Round-Off Error (Kesalahan Pembulatan)

Komputer hanya dapat menyimpan sejumlah tertentu angka signifikan selama kalkulasi.

Contoh $\pi = 3{,}141592$ dengan mengabaikan suku-suku yang lainnya → $E_t = 0{,}00000065\ldots$

Nama teknik penyimpanan ini adalah *chopping*, jadi tergantung pada tipe data yang digunakan. Cara yang paling gampang adalah cukup mengambil digit bilangan sesuai dengan maksimal tipe datanya, dan ini sering dilakukan. Cara lain adalah dengan memperhitungkan juga pada digit selanjutnya setelah dipotong. Apakah perlu dibulatkan ataukah tidak, tapi cara ini memperlama waktu komputasi. Maka cara pertama yang biasanya diambil, yaitu *chopping* sederhana.

3.4.1 Cara Meminimalkan Round-Off Error

Efek round-off error dapat diminimalkan dengan mengubah algoritma komputasional, walaupun ini juga harus melihat kasus demi kasus. Beberapa strategi yang berguna adalah:

1. Membuat tipe datanya menjadi double precision [McCracken]
2. Grouping
3. Perluasan deret Taylor
4. Mengubah definisi variabel
5. Menuliskan kembali persamaan yang dapat mencegahnya dari operasi pengurangan

Contoh kasus

- Kalikan 0,00001 sebanyak 10000 kali dan tambahkan ke bilangan 1.0.

Di bawah ini diperlihatkan kode asal program dalam bahasa C:

```
/*Summation by Single Precision sum_singl.c*/
#include <stdio.h>
main ()   {
float sum = 1.0;
int i;
     for (i=1; i<=10000; i++) {
          sum = sum + 0.00001;
     }
printf("\nSum = %f\n", sum);
}
```

Irfan Subakti - 司馬伊凡

Membuat tipe datanya menjadi double precision

```
/*Summation by Double Precision sum_dbl.c*/
#include <stdio.h>
main ()   {
double sum = 1.0;
int i;
     for (i=1; i<=10000; i++) {
          sum = sum + 0.00001;
     }
printf("\nSum = %f\n", sum);
}
```

Grouping

```
/*Summation by Grouping sum_gr.c*/
#include <stdio.h>
main ()   {
float gr_total, sum = 1;
int i, k;
     for (i=1; i<=100; i++)   {
          gr_total = 0;
     for (k=1; k<=100; k++)   {
          gr_total = gr_total + 0.00001;
     }
     }
sum = sum + gr_total;
printf("\nSum = %13.8f\n", sum);
}
```

Perluasan deret Taylor

Contoh. Jika θ mendekati 0, maka akurasi dari evaluasi numerik di bawah ini menjadi jelek hasilnya disebabkan adanya round-off error.

$$d \cong \frac{\sin(1+\theta) - \sin(1)}{\theta}$$

Dengan menggunakan perluasan Deret Taylor, kita dapat menuliskan kembali persamaan di atas sehingga akurasi θ dapat ditingkatkan.

Jika dicoba menghitung perbedaan kecil dari 2 nilai fungsi, perluasan Deret Taylor merupakan metode yang sangat berguna.

Perluasan Deret Taylor dari sin (1 + θ) adalah

$$\sin(1+\theta) = \sin(1) + \theta\cos(1) - 0{,}5\,\theta^2 \sin(1) \ldots$$

Jika kita dekati sin (1 + θ) dengan 3 suku pertama, d menjadi:

$$d \approx \cos(1) - 0{,}5\theta \sin(1)$$

Hasil perhitungannya:

Irfan Subakti 司馬伊凡

Tabel 3.1 Hasil perhitungan sin (1 + θ)

θ	d
0,1	0,49822
0,01	0,53609
0,001	0,53988
0,0001	0,54026
0,00001	0,54030
0,000001	0,54030
0,0000001	0,54030

Nilai sesungguhnya = 0,54030

Akurasi dari pendekatan/aproksimasi meningkat di saat θ mendekati 0.

Mengubah definisi variabel.

Membuat variabel yang dipakai didefinisikan untuk tingkat presisi yang tinggi. Ini berhubungan dengan membuat tipe datanya menjadi double precision. Dalam program, kasus seperti ini dapat dilakukan dengan *casting* tipe data.

Menuliskan kembali persamaan yang dapat mencegahnya dari operasi pengurangan

Jika persamaan di bawah ini langsung dievaluasi dalam program, maka round-off error akan terjadi jika x mendekati +∞ dan -∞

$$y = \frac{1}{(a-z)(b-z)} \qquad [A]$$

dimana

$$z = \frac{a+b+(b-a)\tanh(x)}{2} \qquad [B]$$

Tuliskan kembali persamaan di atas, sehingga tak terjadi round-off error yang serius.

Disebabkan $-1 < \tanh(x) < 1$, domain dari z adalah $a < z < b$. Saat x mendekati ∞, z mendekati b; dan saat x mendekati -∞, z mendekati b. 2 sumber round-off error ini harus dipertimbangkan.

Jika b = 0, persamaan [B] menjadi a[1 – tanh(x)]. Maka, beberapa round-off error terjadi disaat tanh(x) mendekati 1. Operasi pengurangan dari bilangan yang serupa dalam

persamaan [B] dapat dicegah dengan mencari hubungan:

$$\tanh(x) = \frac{\exp(x) - \exp(-x)}{\exp(x) + \exp(-x)} \qquad [C]$$

Gunakan persamaan [C] ke dalam persamaan [B] dan tuliskan kembali persamaannya, menghasilkan:

$$z = \frac{b\exp(x) + a\exp(-x)}{\exp(x) + \exp(-x)}$$

Disaat z mendekati a atau b, round-off error terjadi dalam persamaan [A]. Untuk mencegah hal ini, komputasi dari persamaan [A] dapat dibagi menjadi 2 kasus.

Kasus 1 $a < z < (a + b)/2$

Kasus 2 $(a + b)/2 \le z < b$

Untuk kasus 1, a – z dalam persamaan [A] ditulis menjadi:

$$a - z = -\frac{b\exp(x) + a\exp(-x)}{\exp(x) + \exp(-x)}$$

$$= \frac{(a-b)\exp(x)}{\exp(x) + \exp(-x)}$$

Untuk kasus 2, b – z ditulis menjadi:

$$b - z = b - \frac{b\exp(x) + a\exp(-x)}{\exp(x) + \exp(-x)} = \frac{(b-a)\exp(-x)}{\exp(x) + \exp(-x)}$$

Irfan Subakti - 司馬伊凡

3.5 Truncation Error (Kesalahan Pemotongan)

Dihasilkan dari pengunaan aproksimasi pengganti matematika eksak. Contoh adalah kasus penerjun jatuh yang telah dibahas di depan.

$$\frac{dv}{dt} \approx \frac{\Delta v}{\Delta t} = \frac{v(t_{i+1}) - v(t_i)}{t_{i+1} - t_i} \qquad [3.8]$$

Truncation error terjadi karena yang diambil adalah sebagian suku saja.

3.5.1 Deret Taylor

Aproksimasi orde ke-0 dilakukan dengan mengambil 1 suku pertama, yaitu nilai sebelumnya.

$$f(x_{i+1}) \approx f(x_i) \qquad [3.9]$$

Aproksimasi orde ke-1 dilakukan dengan menambahi suku lainnya.

$$f(x_{i+1}) \approx f(x_i) + f'(x_i)(x_{i+1}-x_i) \qquad \rightarrow f'(xi) \text{ adalah kemiringan/slope} \qquad [3.10]$$

Aproksimasi orde ke-2 dilakukan dengan menambahi lagi suku lainnya.

$$f(x_{i+1}) \approx f(x_i) + f'(x_i)(x_{i+1}-x_i) + \frac{f''(x_i)}{2!}(x_{i+1}-x_i)^2 \qquad [3.11]$$

Sehingga deret Taylor selengkapnya adalah sbb:

$$f(x_{i+1}) = f(x_i) + f'(x_i)(x_{i+1}-x_i) + \frac{f''(x_i)}{2!}(x_{i+1}-x_i)^2 + \frac{f'''(x_i)}{3!}(x_{i+1}-x_i)^3 + \ldots + (x_{i+1}-x_i)^n + R_n \qquad [3.12]$$

$R_n = \frac{f^{(n+1)}(\xi)}{(n+1)!}(x_{i+1}-x_i)^{n+1}$ dimana ξ adalah suatu harga x pada interval = $h = x_{i+1}-x_i$.

[3.13]

Seringkali ada baiknya untuk memudahkan Deret Taylor dengan mendefinisikan suatu ukuran langkah $h = x_{i+1} - x_i$ dan menyatakan persamaan [3.12] sebagai:

$$f(x_{i+1}) = f(x_i) + f'(x_i)\,h + \frac{f''(x_i)}{2!}h^2 + \frac{f'''(x_i)}{3!}h^3 + \ldots + \frac{f^{(n)}(x_i)}{n!}h^n + R_n \qquad [3.14]$$

Dimana suku sisa sekarang adalah:

$$R_n = \frac{f^{(n+1)}(\xi)}{(n+1)!}h^{n+1} \qquad [3.15]$$

Contoh

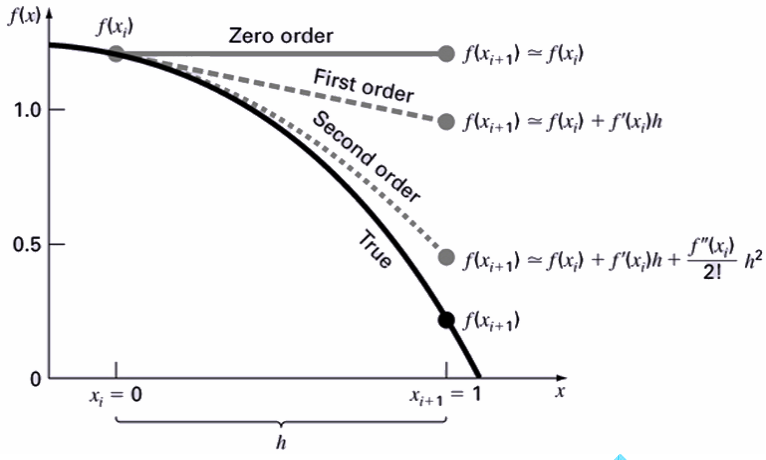
$$f(x) = -0{,}1x^4 - 0{,}15x^3 - 0{,}5x^2 - 0{,}25x + 1{,}2$$

Dimana $x_i = 0$ dengan $h = 1 \rightarrow x_{i+1} = 1$

- Harga sebenarnya adalah 0,2 karena f(1) = 0,2.
- Aproksimasi orde ke-0 → n=0 → ambil dari soal, konstantanya yaitu 1,2. Shg $f(x_{i+1}) \approx 1{,}2$
- Aproksimasi orde ke-1 → n=1 → ambil dari soal, yang variabelnya memiliki pangkat 1 yaitu -0,25x dengan tetap memperhitungkan orde sebelumnya (orde 0). Sebelumnya turunan pertama pd $x_i = 0 \rightarrow f'(0) = -0{,}4(0)^3 - 0{,}45(0)^2 - 1{,}0(0) - 0{,}25 = -0{,}25$.

Irfan Subakti - 包馬伊凡

- Sehingga pendekatan orde ke-1 adalah $f(x_{i+1}) \approx 1,2 - 0,25h \rightarrow f(1) \approx 0,95$. Akibatnya, aproksimasi dimulai guna mengikuti lintasan ke arah bawah dari fungsi dalam bentuk sebuah kemiringan garis lurus, seperti gambar 3.3 di bawah ini:



Gambar 3.3 Aproksimasi dari $f(x) = -0,1x^4 - 0,15x^3 - 0,5x^2 - 0,25x + 1,2$ pada $x = 1$ dengan orde ke nol, orde pertama, dan orde kedua dari perluasan Deret Taylor

- n=2, turunan kedua pd $x_i = 0 \rightarrow f''(0) = -1,2(0)^2 - 0,9(0) - 1 = -1,0$.
- Sehingga pendekatan orde ke-2 adalah $f(x_{i+1}) \approx 1,2 - 0,25h - 0,5h^2 \rightarrow f(1) \approx 0,45$
- Untuk orde ke-4, karena turunan kelima adalah $0 \rightarrow R_4 = 0$, sehingga perluasan Deret Taylor pada taksiran $x_{i+1} = 1$ adalah $f(1) = 1,2 - 0,25(1) - 0,5(1)^2 - 0,15(1)^3 - 0,1(1)^4 = 0,2$.
- $R_4 = \frac{f^{(5)}(\xi)}{5!} h^5$
- Umumnya perluasan Deret Taylor orde ke-n hasilnya sesuai dengan nilai sesungguhnya pada polinomial orde ke-n.
- Untuk fungsi lain seperti eksponensial dan sinusoidal memberikan hasil yang kurang baik. Penambahan orde-orde berikutnya memberikan sedikit kedekatan nilai pada nilai sesungguhnya.
- $R_n = O(h^{n+1}) \rightarrow O(h^{n+1})$ berarti kesalahan pemotongan berorde h^{n+1}. Artinya error yang terjadi sebanding terhadap interval h berpangkat n + 1.
- Misalnya error adalah $O(h)$, membagi 2 interval berarti membagi 2 error. Jika error adalah $O(h^2)$, membagi 2 interval berarti membagi 4 error.

Irfan Subakti 河馬伊乙

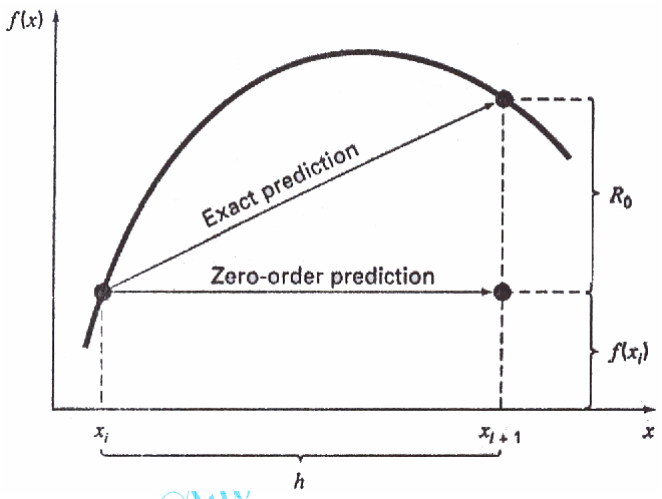
3.5.2 Suku Sisa Perluasan Deret Taylor

$$R_0 = f'(x_i)\, h + \frac{f''(x_i)}{2!} h^2 + \frac{f'''(x_i)}{3!} h^3 + \ldots$$

Untuk mudahnya ambil saja/potong saja sisa itu sehingga menjadi:

$$R_0 \approx f'(x_i)\, h \qquad [3.16]$$

Walaupun turunan orde yang lebih rendah biasanya memiliki andil yang lebih besar terhadap sisa dari suku-suku berorde lebih tinggi, hasil ini masih belum pasti, karena diabaikannya suku-suku orde kedua dan orde yang lebih tinggi. Ketidakpastian ini dinyatakan oleh simbol kesamaan aproksimasi (≈) yang dipakai pada persamaan [3.16].



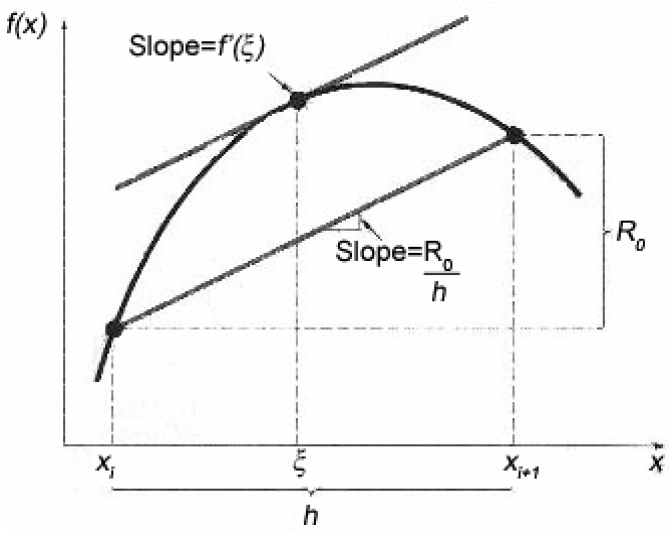
Gambar 3.4 Penjelasan grafik dari sebuah perkiraan dan sisa Deret Taylor orde ke nol

Irfan Subakti

Dari gambar 3.4 di atas kesalahan R_0 dapat ditentukan jika kita mengetahui letak harga pasti. Jelas harga ini tak diketahui, karena kalau demikian tidak diperlukan adanya suatu perluasan Deret Taylor.

Teori harga rata-rata

Teori harga rata-rata menyatakan bahwa jika sebuah fungsi f(x) dan turunan pertamanya kontinu di sepanjang suatu interval x_i hingga x_{i+1}, maka akan ada sekurang-kurangnya sebuah titik pada fungsi yang mempunyai kemiringan, dinyatakan oleh $f'(\xi)$ yang sejajar dengan jarak yang menghubungkan $f(x_i)$ dan $f(x_{i+1})$. Parameter ξ menandai harga x dimana kemiringan itu terjadi, seperti gambar 3.5 di bawah ini:



Gambar 3.5 Grafik teori harga rata-rata

Suatu ilustrasi fisis teori ini diperlihatkan oleh kenyataan bahwa bila anda berjalan antara dua titik dengan suatu kecepatan rata-rata, maka sekurang-kurangnya ada satu saat selama perjalanan itu dimana anda bergerak dengan suatu kecepatan rata-rata.

Dengan memakai teori ini, seperti terlihat pada gambar di atas, mudah disadari bahwa kemiringan f' (ξ) sama dengan kenaikan R_0 dibagi oleh jarak h, atau:

$$f'(\xi) = \frac{R_0}{h}$$

Yang dapat diatur kembali menjadi:

$$R_0 = f'(\xi)\, h \qquad [3.17]$$

Jadi kita telah menurunkan versi orde ke nol dari persamaan [3.15]. Versi orde lebih tinggi ternyata suatu perluasan logika belaka dari alasan yang dipergunakan untuk menurunkan persamaan [3.17], berdasarkan pada bentuk umum teori harga rata-rata (Thomas dan Finney, 1979). Jadi versi orde pertama adalah:

$$R_1 = \frac{f''(\xi)}{2!} h^2 \qquad [3.18]$$

Dalam hal ini, harga ξ memastikan harga x yang sesuai dengan turunan kedua yang menjadikan persamaan [3.18] pasti. Versi berorde lebih tinggi yang serupa dapat dikembangkan dari persamaan [3.15].

Irfan Subakti - 司馬用人

3.5.3 Penggunaan Deret Taylor untuk Memperkirakan Kesalahan Pemotongan.

Walaupun Deret Taylor sangat berguna dalam menaksir kesalahan pemotongan, tapi mungkin kurang jelas bagaimana sesungguhnya perluasan tersebut dapat diterapkan untuk metode numerik. Realitasnya, kita telah menggunakannya dalam contoh penerjun bebas. Ingatlah kembali bahwa tujuan kedua contoh dalam kasus penerjun bebas tersebut ialah untuk meramalkan kecepatan sebagai fungsi waktu. Jadi kita tertarik dalam menentukan v(t). Seperti telah ditentukan dalam persamaan [3.12], v(t) dapat diperluas dalam sebuah Deret Taylor:

$$v(t_{i+1}) = v(t_i) + v'(t_i)(t_{i+1} - t_i) + \frac{v''(t_i)}{2!}(t_{i+1} - t_i)^2 + \ldots + R_n \qquad [3.19]$$

Mari kita potong deret tersebut setelah suku turunan pertama:

$$v(t_{i+1}) = v(t_i) + v'(t_i)(t_{i+1} - t_i) + R_1 \qquad [3.20]$$

Persamaan [3.20] dapat diselesaikan untuk:

$$v'(t_i) = \underbrace{\frac{v(t_{i+1}) - v(t_i)}{t_{i+1} - t_i}}_{\text{Aproksimasi orde pertama}} - \underbrace{\frac{R_1}{t_{i+1} - t_i}}_{\text{Kesalahan pemotongan}} \qquad [3.21]$$

Karena aproksimasinya dengan Deret Taylor, kita telah memperoleh pula suatu taksiran kesalahan pemotongan sehubungan dengan pendekatan ini terhadap turunan. Dengan menggunakan persamaan [3.13] dan [3.21] menjadi:

$$\frac{R_1}{t_{i+1} - t_i} = \frac{v''(\xi)}{2!}(t_{i+1} - t_i) \qquad [3.22]$$

atau

$$\frac{R_1}{t_{i+1} - t_i} = O(ti+1 - ti) \qquad [3.23]$$

Jadi taksiran dari turunan persamaan [1.10] atau bagian pertama dari persamaan [3.21] mempunyai suatu kesalahan pemotongan berorde $t_{i+1} - t_i$. Dengan kata lain, kesalahan dari aproksimasi turunan harus sebanding dengan ukuran langkah. Akibatnya jika kita membagi dua ukuran langkah, kita dapat mengharapkan kesalahan yang terjadi akan turun setengahnya.

Irfan Subakti - 司馬伊凡

3.5.4 Diferensiasi Numerik

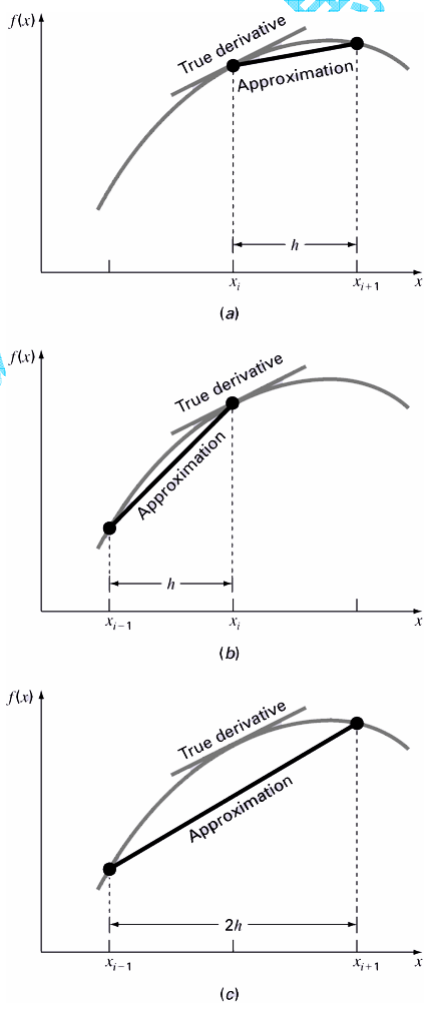
Persamaan [3.21] diberi label resmi dalam metode numerik, disebut suatu **diferensi terbagi hingga**, yang secara umum dapat dinyatakan dengan:

$$f'(x_i) = \frac{f(x_{i+1}) - f(x_i)}{x_{i+1} - x_i} + O(x_{i+1} - x_i) \qquad [3.24]$$

atau

$$f'(x_i) = \frac{\Delta f_i}{h} + O(h) \qquad [3.25]$$

dimana Δf_i diacu sebagai **diferensi ke depan pertama** dan h dinamakan **ukuran langkah**, yakni panjang interval dimana pendekatan dilakukan. Ia disebut diferensi "ke depan" karena memanfaatkan data pada i dan i + 1 untuk menaksir turunan, seperti terlihat pada gambar di bawah ini, di bagian (a). Keseluruhan suku $\Delta f_i/h$ diacu sebagai suatu **diferensi terbagi hingga pertama**. Penggambaran dari penjelasan ini dapat dilihat pada gambar 3.6 di bawah ini.



Gambar 3.6 Penjelasan grafik dari turunan pertama pendekatan diferensi terbagi hingga (a) ke depan, (b) ke belakang, dan (c) terpusat

Diferensi terbagi ke depan ini merupakan satu dari sekian banyak Deret Taylor yang dapat dikembangkan untuk mencari aproksimasi turunan secara numerik. Misalnya pendekatan **diferensi terpusat** dan **ke belakang** turunan pertama yang dapat dikembangkan dalam model yang mirip dengan turunan persamaan [3.24]. Yang pertama memakai data pada x_{i+1} (gambar 3.6 bagian (a)), sedangkan yang terakhir memakai informasi yang berspasi sama di sekitar titik dimana turunan tersebut ditaksir (gambar 3.6 bagian (c)). Aproksimasi turunan pertama yang lebih akurat dapat dikembangkan dengan memasukkan suku-suku berorde lebih tinggi dari Deret Taylor. Akhirnya semua versi di atas dapat juga dikembangkan dari turunan kedua, ketiga, dan yang lebih tinggi.

Pendekatan diferensi ke belakang dari turunan pertama

Deret Taylor dapat diperluas ke belakang untuk menghitung suatu harga sebelumnya berdasarkan harga sekarang, seperti pada:

$$f(x_{i-1}) = f(x_i) - f'(x_i)h + \frac{f''(x_i)}{2}h^2 - \ldots \qquad [3.26]$$

Potonglah persamaan ini setelah turunan pertama, lalu aturlah kembali agar memenuhi:

$$f'(x_i) = \frac{f(x_i) - f(x_{i-1})}{h} = \frac{\nabla f_i}{h} \qquad [3.27]$$

dimana kesalahannya adalah O(h) dan ∇f_i, diacu sebagai **diferensi ke belakang pertama**. Lihat gambar 3.6 bagian (b) untuk gambaran grafiknya.

Irfan Subakti - 司馬伊凡

Pendekatan diferensi terpusat dari turunan pertama

Cara ketiga untuk aproksimasi turunan pertama ialah mengurangkan persamaan [3.26] dari perluasan Deret Taylor ke depan:

$$f(x_{i+1}) = f(x_i) - f'(x_i)h + \frac{f''(x_i)}{2}h^2 + \ldots \qquad [3.28]$$

sehingga

$$f(x_{i+1}) - f(x_{i-1}) = 2f'(x_i)h + h^3 + \ldots$$

yang dapat diselesaikan sehingga:

$$f'(x_i) = \frac{f(x_{i+1}) - f(x_{i-1})}{2h} - \frac{f'''(x_i)}{6}h^2 + \ldots \qquad \text{atau}$$

$$f'(x_i) = \frac{f(x_{i+1}) - f(x_{i-1})}{2h} + O(h^2) \qquad [3.29]$$

Persamaan [3.29] merupakan sebuah pernyataan turunan pertama **diferensi terpusat (sentral)**. Perhatikan bahwa kesalahan pemotongan adalah berorde h^2 yang bertentangan dengan aproksimasi ke depan dan ke belakang yang berorde h. Akibatnya, analisis Deret Taylor mengandung informasi praktis bahwa diferensi terpusat merupakan pernyataan turunan yang lebih akurat (gambar 3.6 bagian (c). Misalkan jika kita menempatkan ukuran langkah menggunakan sebuh diferensi ke depan atau ke belakang, secara aproksimasi kita melakukan kesalahan pemotongan, sedangkan untuk diferensi terpusat, kesalahan akan menjadi seperempatnya.

Pendekatan diferensi hingga dari turunan lebih tinggi

Di samping turunan pertama, perluasan Deret Taylor dapat digunakan untuk menurunkan taksiran numerik dari turunan yang lebih tinggi. Untuk melakukan ini, kita tulis sebuah perluasan Deret Taylor ke depan dari $f(x_{i+2})$ dinyatakan dengan $f(x_i)$:

$$f(x_{i+2}) = f(x_i) + f'(x_i)(2h) + \frac{f''(x_i)}{2}(2h)^2 + \ldots \qquad [3.30]$$

Persamaan [3.28] dapat dikalikan 2 dan dikurangkan dari persamaan [3.30]:

$$f(x_{i+2}) - 2f(x_{i+1}) = -f(x_i) + f''(x_i)h^2 + \ldots$$

yang dapat diselesaikan sehingga:

$$f''(x_i) = \frac{f(x_{i+2}) - 2f(x_{i+1}) + f(x_i)}{h^2} + O(h) \qquad [3.31]$$

Hubungan ini dinamakan **diferensi terbagi hingga ke depan kedua**. Manipulasi yang serupa dapat dipakai untuk menurunkan versi ke belakang dan terpusat. Pendekatan ke belakang, ke depan dan terpusat dari turunan orde ketiga dan lebih tinggi dapat juga dikembangkan (gambar 3.7 sampai dengan gambar 3.9). Pada semua kasus, diferensi terpusat mengandung taksiran yang lebih akurat.

Pada gambar 3.7 disajikan penggambaran dari rumus diferensi terbagi hingga ke belakang. Disitu juga terlihat dua versi diberikan untuk setiap turunan. Yang terakhir menyatakan suku-suku lebih banyak dari perluasan Deret Taylor dan akibatnya lebih akurat.

Irfan Subakti - 同舟共济

Turunan pertama	**Kesalahan**
$f'(x_i) = \dfrac{f(x_i) - f(x_{i-1})}{h}$	O(h)
$f'(x_i) = \dfrac{3f(x_i) - 4f(x_{i-1}) + f(x_{i-2})}{2h}$	$O(h^2)$
Turunan kedua	
$f''(x_i) = \dfrac{f(x_i) - 2f(x_{i-1}) + f(x_{i-2})}{h^2}$	O(h)
$f''(x_i) = \dfrac{2f(x_i) - 5f(x_{i-1}) + 4f(x_{i-2}) - f(x_{i-3})}{h^2}$	$O(h^2)$
Turunan ketiga	
$f'''(x_i) = \dfrac{f(x_i) - 3f(x_{i-1}) + 3f(x_{i-2}) - f(x_{i-3})}{h^3}$	O(h)
$f'''(x_i) = \dfrac{5f(x_i) - 18f(x_{i-1}) + 24f(x_{i-2}) - 14f(x_{i-3}) + 3f(x_{i-4})}{2h^3}$	$O(h^2)$
Turunan keempat	
$f''''(x_i) = \dfrac{f(x_i) - 4f(x_{i-1}) + 6f(x_{i-2}) - 4f(x_{i-3}) + f(x_{i-4})}{h^4}$	O(h)
$f''''(x_i) = \dfrac{3f(x_i) - 14f(x_{i-1}) + 26f(x_{i-2}) - 24f(x_{i-3}) + 11f(x_{i-4}) - 2f(x_{i-5})}{h^4}$	$O(h^2)$

Gambar 3.7 Rumus diferensi terbagi hingga ke belakang: dua versi diberikan untuk setiap turunan

Sedangkan pada gambar 3.8 disajikan penggambaran dari rumus diferensi terbagi hingga ke depan. Yang terakhir menyatakan suku-suku lebih banyak dari perluasan Deret Taylor dan akibatnya lebih akurat.

Turunan pertama	**Kesalahan**
$f'(x_i) = \frac{f(x_{i+1}) - f(x_i)}{h}$	O(h)
$f'(x_i) = \frac{-f(x_{i+2}) + 4f(x_{i+1}) - 3f(x_i)}{2h}$	$O(h^2)$
Turunan kedua	
$f''(x_i) = \frac{f(x_{i+2}) - 2f(x_{i+1}) + f(x_i)}{h^2}$	O(h)
$f''(x_i) = \frac{-f(x_{i+3}) + 4f(x_{i+2}) - 5f(x_{i+1}) + 2f(x_i)}{h^2}$	$O(h^2)$
Turunan ketiga	
$f'''(x_i) = \frac{f(x_{i+3}) - 3f(x_{i+2}) + 3f(x_{i+1}) - f(x_i)}{h^3}$	O(h)
$f'''(x_i) = \frac{-3f(x_{i+4}) + 14f(x_{i+3}) - 24f(x_{i+2}) + 18f(x_{i+1}) - 5f(x_i)}{2h^3}$	$O(h^2)$
Turunan keempat	
$f''''(x_i) = \frac{f(x_{i+4}) - 4f(x_{i+3}) + 6f(x_{i+2}) - 4f(x_{i+1}) + f(x_i)}{h^4}$	O(h)
$f''''(x_i) = \frac{-2f(x_{i+5}) + 11f(x_{i+4}) - 24f(x_{i+3}) + 26f(x_{i+2}) - 14f(x_{i+1}) + 3f(x_i)}{h^4}$	$O(h^2)$

Gambar 3.8 Rumus diferensi terbagi hingga ke depan: dua versi diberikan untuk setiap turunan

Irfan Subakti 同意伊凡

Pada gambar 3.9 disajikan penggambaran dari rumus diferensi terbagi hingga terpusat. Yang terakhir menyatakan suku-suku lebih banyak dari perluasan Deret Taylor dan akibatnya lebih akurat.

Turunan pertama	Kesalahan
$f'(x_i) = \frac{f(x_{i+1}) - f(x_{i-1})}{2h}$	$O(h^2)$
$f'(x_i) = \frac{-f(x_{i+2}) + 8f(x_{i+1}) - 8f(x_{i-1}) + f(x_{i-2})}{12h}$	$O(h^4)$
Turunan kedua	
$f''(x_i) = \frac{f(x_{i+1}) - 2f(x_i) + f(x_{i-1})}{h^2}$	$O(h^2)$
$f''(x_i) = \frac{-f(x_{i+2}) + 16f(x_{i+1}) - 30f(x_i) + 16f(x_{i-1}) - f(x_{i-2})}{12h^2}$	$O(h^4)$
Turunan ketiga	
$f'''(x_i) = \frac{f(x_{i+2}) - 2f(x_{i+1}) + 2f(x_{i-1}) - f(x_{i-2})}{2h^3}$	$O(h^2)$
$f'''(x_i) = \frac{-f(x_{i+3}) + 8f(x_{i+2}) - 13f(x_{i+1}) + 13f(x_{i-1}) - 8f(x_{i-2}) + f(x_{i-3})}{8h^3}$	$O(h^4)$
Turunan keempat	
$f''''(x_i) = \frac{f(x_{i+2}) - 4f(x_{i+1}) + 6f(x_i) - 4f(x_{i-1}) + f(x_{i-2})}{h^4}$	$O(h^2)$
$f''''(x_i) = \frac{-f(x_{i+3}) + 12f(x_{i+2}) - 39f(x_{i+1}) + 56f(x_i) - 39f(x_{i-1}) + 12f(x_{i-2}) - f(x_{i-3})}{6h^4}$	$O(h^4)$

Gambar 3.9 Rumus diferensi terbagi hingga terpusat: dua versi diberikan untuk setiap turunan

Irfan Subakti 司馬伊凡

Formula diferensi yang lebih akurat

Semua taksiran di atas memotong taksiran Deret Taylor hanya setelah beberapa suku. Rumus dengan ketepatan lebih tinggi dapat dikembangkan dengan memasukkan suku-suku tambahan. Misalnya perluasan persamaan [3.28] dapat diselesaikan untuk:

$$f'(x_i) = \frac{f(x_{i+1}) - f(x_i)}{h} - \frac{f''(x_i)}{2}h + O(h^2) \qquad [3.32]$$

Bertentangan dengan persamaan [3.24], kita dapat menahan suku turunan kedua dengan memasukkan persamaan [3.31] ke dalam persamaan [3.32] untuk memenuhi:

$$f'(x_i) = \frac{f(x_{i+1}) - f(x_i)}{h} - \frac{f(x_{i+2}) - 2f(x_{i+1}) + f(x_i)}{2h^2}h + O(h^2)$$

atau dengan mengumpulkan suku-suku

$$f'(xi) = \frac{-f(x_{i+2}) + 4f(x_{i+1}) - 3f(xi)}{2h} + O(h^2)$$

Perhatikan bahwa suku turunan kedua telah memperbaiki ketepatan menjadi $O(h^2)$. Versi perbaikan yang serupa dapat dikembangkan untuk rumus ke belakang dan terpusat, untuk aproksimasi turunan lebih tinggi. Rumus diringkaskan dalam gambar 3.7 s/d gambar 3.9. Contoh yang berikut memperlihatkan manfaat dari penaksiran turunan.

Contoh: Aproksimasi diferensi terbagi hingga dari turunan

Pernyataan masalah: gunakan aproksimasi terbagi hingga ke depan dan ke belakang dari O(h) dan aproksimasi diferensi terpusat dari $O(h^2)$ untuk menaksir turunan pertama dari:

$$f(x) = -0{,}1x^4 - 0{,}15x^3 - 0{,}5x^2 - 0{,}25x + 1{,}2$$

pada x = 0,5 menggunakan ukuran langkah h = 0,5. Ulangi perhitungan dengan memakai h = 0,25. Perhatikan bahwa turunan dapat dihitung secara langsung sebagai:

$$f'(x) = -0{,}4x^3 - 0{,}45x^2 - 1{,}0x - 0{,}25$$

dan dapat digunakan untuk menghitung harga sebenarnya, karena:

$$f'(0{,}5) = -0{,}9125$$

Solusi

Untuk h = 0,5; fungsi dapat dipakai untuk menentukan:

$x_{i-1} = 0$ $\quad f(x_{i-1}) = 1{,}2$

$x_i = 0{,}5$ $\quad f(x_i) = 0{,}925$

$x_{i+1} = 1{,}0$ $\quad f(x_{i+1}) = 0{,}2$

Data ini dapat digunakan untuk menghitung diferensi terbagi hingga ke depan (persamaan [3.24]):

$$f'(0{,}5) = \frac{0{,}2 - 0{,}925}{0{,}5} = -1{,}45 \qquad \epsilon_t = -58{,}9\%$$

diferensi terbagi hingga ke belakang (persamaan [3.27]):

$$f'(0{,}5) = \frac{0{,}925 - 1{,}2}{0{,}5} = -0{,}55 \qquad \epsilon_t = 39{,}7\%$$

diferensi terbagi hingga terpusat (persamaan [3.29]):

$$f'(0{,}5) = \frac{0{,}2 - 1{,}2}{1{,}0} = -1{,}0 \qquad \epsilon_t = -9{,}6\%$$

Irfan Subakti - 司馬伊凡

Untuk h = 0,25; data adalah:

$x_{i-1} = 0,25 \rightarrow f(x_{i-1}) = 1,10351563$

$x_i = 0,5\ \ 0 \rightarrow f(x_i) = 0,925$

$x_{i+1} = 0,75 \rightarrow f(x_{i+1}) = 0,63632813$

yang dapat digunakan untuk menghitung diferensi terbagi hingga ke depan:

$$f'(0,5) = \frac{0,63632813 - 0,925}{0,25} = -1,155 \qquad \epsilon_t = -26,5\%$$

diferensi terbagi hingga ke belakang:

$$f'(0,5) = \frac{0,925 - 1,10351563}{0,25} = -0,714 \qquad \epsilon_t = 21,7\%$$

diferensi terbagi hingga terpusat:

$$f'(0,5) = \frac{0,63632813 - 1,10351563}{0,5} = -0,934 \qquad \epsilon_t = -2,4\%$$

Untuk kedua ukuran langkah, aproksimasi diferensi terpusat lebih akurat daripada diferensi ke depan dan ke belakang. Juga seperti diperkirakan oleh analisis Deret Taylor, membagi 2 ukuran langkah berarti membagi 2 kesalahan diferensi ke depan dan ke belakang serta membagi 4 kesalahan diferensi terpusat.

Irfan Subakti - 司馬伊凡

3.6 Kesalahan Numerik Total

Kesalahan numerik total adalah penjumlahan dari kesalahan pemotongan dan kesalahan pembulatan.

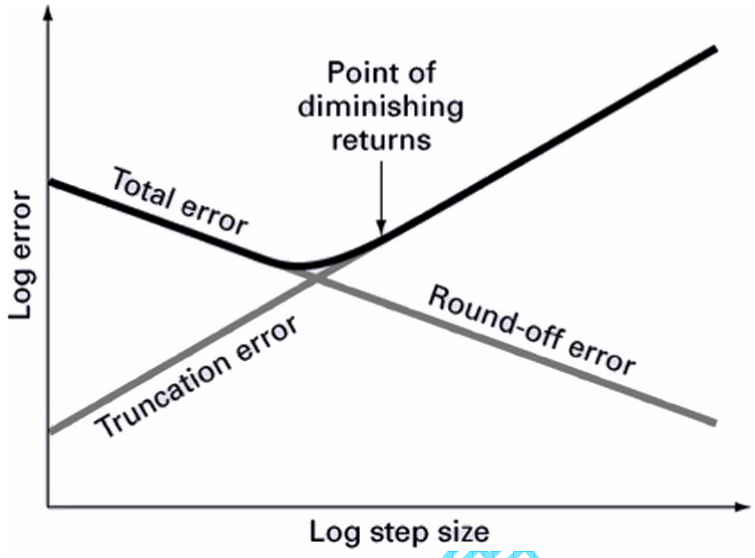
Pada kasus penerjun payung, kita telah menemukan bahwa cara mengurangi kesalahan pembulatan hanya dengan meningkatkan angka signifikan komputer. Juga didapat: kesalahan pembulatan akan bertambah kalau jumlah komputasi dalam analisis dinaikkan. Cara meminimalkan round-off error selengkapnya dapat dibaca kembali di sub bab 3.4.1.

Sebaliknya taksiran turunan dapat diperbaiki dengan mengurangi ukuran langkap (step). Karena suatu pengurangan ukuran langkah membawa akibat kenaikan komputasi, kesalahan pemotongan dikurangi jika jumlah komputasi bertambah.

Terdapat dilema: strategi untuk mengurangi satu komponen dari kesalahan total akan menaikkan komponen lainnya. Dalam suatu komputasi, secara konsepsional kita dapat mengurangi ukuran langkah guna meminimalkan kesalahan pemotongan, hanya untuk

menemukan bahwa dalam melakukan hal ini, kesalahan pembulatan mulai mendominasi solusi dan kesalahan total bertambah.

Pemecahannya adalah seperti gambar 3.10 di bawah ini. Terlihat titik balik pengurangan dimana kesalahan pembulatan mulai meniadakan manfaat pengurangan ukuran langkah.



Gambar 3.10 Kompromi antara kesalahan pembulatan dan kesalahan pemotongan yang sering muncul berkenaan dengan metode numerik

Irfan Subakti - Blog

Tantangannya adalah: menentukan suatu ukuran langkah yang layak bagi suatu perhitungan tertentu.

Diinginkan memilih suatu ukuran langkah yang besar agar mengurangi jumlah kalkulasi dan kesalahan pembulatan tanpa dirugikan suatu kesalahan pemotongan yang besar.

Jika kesalahan total, seperti terlihat pada gambar 3.10, tantangannya adalah untuk menemukan titik balik pengurangan (**diminishing return**) dimana kesalahan pembulatan mulai meniadakan keuntungan dari pengurangan ukuran langkah.

Untungnya dalam realitas yang ada, sekarang ini komputer dapat menangani angka signifikan yang cukup besar dimana kesalahan pembulatan tak lebih dominan.

Pelbagai kekurangan di atas, menyebabkan kita harus mencoba-coba (trial and error) dalam memperkirakan kesalahan, juga melibatkan intuisi dan pengalaman.

3.7 Kekeliruan, Kesalahan Perumusan dan Ketidakpastian Data

Walau sumber kesalahan di bawah ini secara langsung tak dihubungkan dalam metode numerik, dampak dari kesalahan ini cukup besar.

Kekeliruan

Kesalahan bruto/kekeliruan.

Tahun awal penggunaan komputer, komputer sering kali gagal pakai (**malfunction**).

Sekarang kekeliruan ini dihubungkan dengan ketidaksempurnaan manusianya.

Kekeliruan dapat terjadi pada sembarang langkah proses pemodelan matematika dan dapat mengambil bagian terhadap semua komponen kesalahan lainnya. Ia hanya dapat dicegah oleh pengetahuan yang baik tentang prinsip dasar dan berhati-hatilah dalam melakukan pendekatan dan mendesain solusi untuk masalah anda.

Biasanya tak dianggap dalam pembahasan metode numerik.

Ini terjadi, karena kesalahan bruto sampai taraf tertentu tak dapat dihindari. Tapi tentu saja pasti ada cara untuk memperbaiki keadaan ini.

Misalnya: kebiasaan pemrograman yang baik, seperti yang dibahas dalam bab 2, sangat berguna untuk mengurangi kekeliruan pemrograman. Sebagai tambahan, terdapat juga cara-cara sederhana untuk memeriksa apakah suatu metode numerik tertentu bekerja secara sempurna.

Irfan Subakti - 司馬伊凡

Kesalahan Perumusan

Kesalahan perumusan model dihubungkan dengan penyimpangan yang dapat dianggap berasal dari model matematika yang tak sempurna.

Contoh: fakta bahwa hukum Newton kedua tak menghitung efek relativistik. Ini tak mengurangi kelayakan solusi pada contoh sebelumnya, karena kesalahan-kesalahan ini adalah minimal pada skala waktu dan ruang dari seorang penerjun payung.

Anggap bahwa tahanan udara bukan proporsi linier terhadap kecepatan jatuh seperti dalam persamaan [1.6], tetapi merupakan sebuah fungsi kuadrat kecepatan. Kalau hal ini benar, baik kedua solusi analitis maupun numerik yang diperoleh dalam bab 1 hasilnya menjadi salah karena kesalahan perumusan.

Ketidakpastian Data

Kesalahan-kesalahan seringkali masuk ke dalam suatu analisis karena ketidakpastian data fisika yang mendasari suatu model.

Misalnya kita ingin menguji model penerjun payung dengan loncatan-loncatan berulang

yang dibuatnya, mengukur kecepatan orang tersebut setelah interval waktu tertentu.

Ketidakpastian yang menyertai pengukuran-pengukuran ini tak diragukan, karena penerjun akan jatuh lebih cepat selama beberapa loncatan daripada loncatan lainnya. Kesalahan-kesalahan ini dapat memunculkan ketidak akuratan dan ketidak presisian.

Jika instrumen kita menaksir terlalu rendah atau terlalu tinggi terhadap kecepatan, kita menghadapi suatu alat yang tak akurat atau menyimpang.

Pada keadaan lainnya, jika pengukuran tinggi dan rendah secara acak, kita akan berhadapan dengan sebuah pertanyaan mengenai kepresisian.

Kesalahan-kesalahan pengukuran dapat dikuantifikasikan dengan meringkaskan data dengan satu atau lebih statistik yang dipilih yang membawa sebanyak mungkin informasi mengenai sifat-sifat data tertentu.

Statistik yang deskriptif ini kebanyakan sering dipilih untuk menyatakan (1) letak pusat distribusi data, dan (2) tingkat penyebaran data. Hal demikian memberikan suatu ukuran penyimpangan dan ketidakpresisian.

Irfan Subakti - 司馬伊凡

Bab 4 Metode Akolade (Bracketing Method)

Sebuah fungsi berdasarkan jenisnya akan berubah tanda di sekitar suatu harga akar.

Teknik ini dinamakan metode akoladi (bracketing method), karena dibutuhkan 2 tebakan awal untuk akar.

Sesuai namanya, tebakan tersebut harus "dalam kurung" atau berada pada kedua sisi nilai akar.

4.1 Metode Grafik

Untuk memperoleh taksiran akar persamaan f(x) = 0 ialah dengan membuat grafik fungsi itu dan mengamati dimana ia memotong sumbu x.

Titik ini, yang menyatakan harga x untuk f(x) = 0, memberikan suatu pendekatan kasar dari akar tersebut.

Irfan Subakti - 司馬伊凡

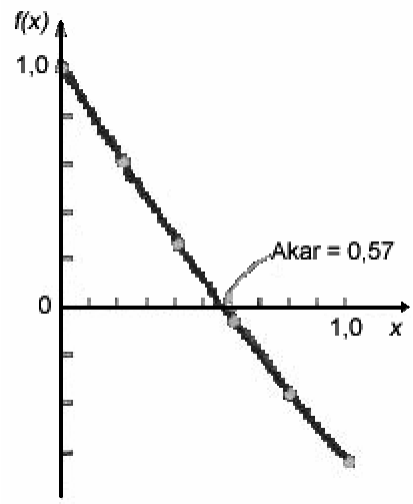
Contoh 4.1 Pendekatan Grafik

Gunakan pendekatan grafik untuk memperoleh suatu akar persamaan dari $f(x) = e^{-x} - x$.

Solusinya yang kita peroleh dapat disajikan dalam tabel 4.1 dan gambar 4.1 seperti berikut ini. Pada gambar 4.1 terlihat grafik f(x) = e-x – x terhadap x. Akar sesuai dengan harga x dimana f(x) = 0, yaitu titik dimana fungsi memotong sumbu x. Pemeriksaan secara visual mengenai plot memberikan taksiran kasar 0,57.

Tabel 4.1 Solusi pendekatan grafik

x	f(x)
0,0	1,000
0,2	0,619
0,4	0,270
0,6	-0,051
0,8	-0,351
1,0	-0,632



Gambar 4.1 Ilustrasi pendekatan grafik untuk memecahkan persamaan aljabar dan transendental

Harga sebenarnya adalah 0,56714329…

Kecocokan taksiran visual dapat dicek dengan memasukkan harga itu ke dalam persamaan awal agar memenuhi:

$f(0,57) = e^{-0,57} - 0,57 = -0,0045$ yang mendekati nol.

Teknik grafik praktis digunakan, dan dapat memberikan taksiran akar secara kasar, tapi tidak presisi.

Ia dapat digunakan sebagai tebakan awal dalam metode numerik.

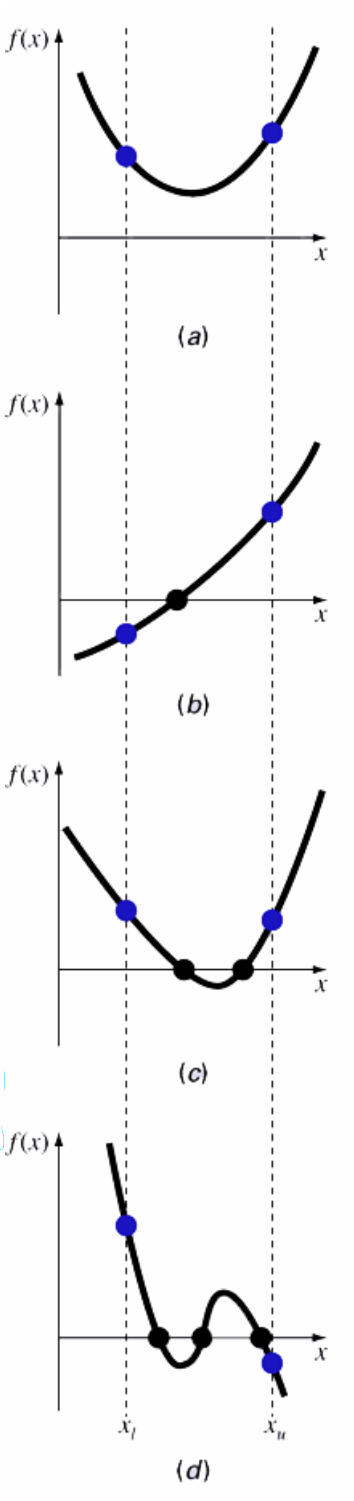
Interpretasi grafik penting untuk memahami sifat-sifat fungsi dan dapat memperkirakan jebakan pada metode numerik, seperti terlihat pada gambar 4.2 di bawah ini.

Pada gambar 4.2 di bawah ini terlihat bagian (a) dan (c) menunjukkan bahwa bila $f(x_l)$ dan $f(x_u)$ mempunyai tanda yang sama, tidak akan ada akar-akar atau akar dalam jumlah genap pada interval. Bagian (b) dan (d) menunjukkan bahwa bila fungsi mempunyai tanda yang berbeda pada kedua titik ujung, akan terdapat akar dalam jumlah ganjil pada interval.

Gambar 4.2 memperlihatkan sejumlah cara dimana akar bisa berada dalam interval yang dijelaskan oleh suatu batas bawah x_l dan batas atas x_u.

Gambar 4.2b memperlihatkan kasus dmana sebuah akar tunggal dikurung oleh harga-harga positif dan negatif dari f(x).

Irfan Subakti - 司馬伊凡



Gambar 4.2 Ilustrasi sejumlah cara yang umum bahwa sebuah akar bisa terjadi dalam sebuah interval yang dijelaskan oleh batas bawah x_l dan batas atas x_u

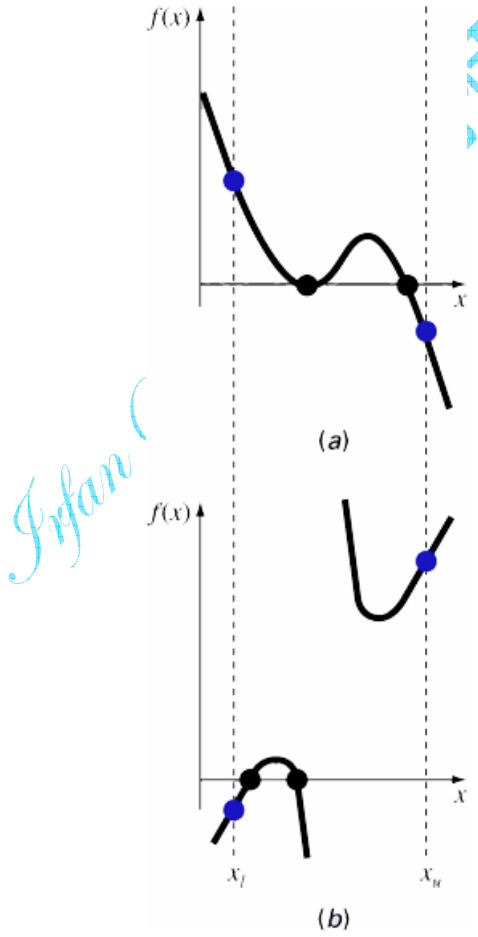
Tetapi gambar 4.2d, dimana $f(x_l)$ dan $f(x_u)$ berlawanan tanda terhadap sumbu x, memperlihatkan 3 akar yang berada di dalam interval. Umumnya jika $f(x_l)$ dan $f(x_u)$ mempunyai tanda yang berbeda akan terdapat akar yang jumlahnya ganjil dalam interval.

Seperti ditunjukkan oleh gambar 4.2a dan c, jika $f(x_l)$ dan $f(x_u)$ mempunyai tanda yang sama, tidak terdapat akar-akar atau akar yang jumlahnya genap berada diantara harga-harga itu.

Meskipun generalisasi ini biasanya benar, namun terdapat kasus-kasus dimana hal itu tak dapat dipegang.

Misalnya akar ganda. Yakni fungsi yang menyinggung sumbu x (gambar 4.3a) dan fungsi-fungsi diskontinu (gambar 4.3b) bisa menyalahi prinsip ini.

Pada gambar 4.3 di bawah ini, terlihat (a) Akar ganda yang terjadi sewaktu fungsi menyinggung sumbu x. Dalam hal ini, walaupun titik-titik ujungnya berlawanan tanda, terdapat akar-akar dalam jumlah genap untuk interval tersebut. (b) Fungsi diskontinu dimana titik-titik ujung tanda yang berlawanan juga mengurung akar-akar dalam jumlah genap. Strategi khusus dibutuhkan untuk penentuan akar-akar dalam kasus ini.



Gambar 4.3 Ilustrasi beberapa perkecualian terhadap kasus-kasus umum yang ditunjukkan dalam gambar 4.2

Sebagai contoh fungsi yang mempunyai akar ganda adalah persamaan kubik $f(x) = (x - 2)(x - 2)(x - 4)$. Perhatikan bahwa $x = 2$ membuat kedua suku polinomial itu sama dengan

0. Jadi x = 2 disebut sebuah akar ganda.

Adanya kasus-kasus seperti yang dinyatakan dalam gambar 4.3 di atas mempersulit pengembangan algoritma komputer secara umum yang menjamin penempatan semua akar dalam suatu interval. Tetapi kalau digunakan bersama-sama dengan pendekatan grafik, akan banyak memberikan nilai guna.

4.2 Metode Bagidua (Biseksi)

Pada teknik grafik sebelumnya, terlihat bahwa f(x) berganti tanda pada kedua sisi yang berlawanan dari kedudukan akar. Pada umumnya, kalau f(x) nyata (real) dan kontinu dalam interval dari x_l hingga x_u, serta $f(x_l)$ dan $f(x_u)$ berlainan tanda, yakni:

$$f(x_l)\, f(x_u) < 0 \qquad [4.1]$$

Maka terdapat sekurang-kurangnya 1 akar nyata diantara x_l dan x_u.

Incremental search method (metode pencarian inkremental) mengawali pengamatan ini dengan penempatan sebuah interval dimana fungsi tersebut bertukar tanda.

Lalu penempatan perubahan tanda (tentunya harga agar) ditandai lebih teliti dengan cara membagi interval tersebut menjadi sejumlah subinterval. Setiap subinterval itu dicari untuk menempatkan perubahan tanda. Proses tersebut diulangi dan perkiraan akar diperhalus dengan membagi subinterval menjadi lebih halus lagi.

Metode Bagidua (**biseksi**), disebut juga pemotongan biner (**binary chopping**), pembagian 2 (**interval halving**) atau metode **Bolzano**.

Yaitu suatu jenis pencarian inkremental dimana interval senantiasa dibagi separuhnya. Kalau suatu fungsi berubah tanda sepanjang suatu interval, harga fungsi di tengahnya dievaluasi. Letak akarnya kemudian ditentukan ada di tengah-tengah subinterval dimana perubahan tanda terjadi. Proses ini diulangi untuk memperoleh taksiran yang diperhalus.

Irfan Subakti - 司馬伊凡

Step 1: Pilih taksiran terendah x_l dan tertinggi x_u untuk akar agar fungsi berubah tanda sepanjang interval. Ini dapat diperiksa dengan: $f(x_l)\ f(x_u) < 0$.

Step 2: Taksiran pertama akar x_r ditentukan oleh:

$$x_r = \frac{x_l + x_u}{2}$$

Step 3: Buat evaluasi yang berikut untuk menentukan subinterval, di dalam mana akar terletak:

a. Jika $f(x_l)\ f(x_r) < 0$, akar terletak pada subinterval pertama, maka $x_u = x_r$, dan lanjutkan ke step 2.

b. Jika $f(x_l)\ f(x_r) > 0$, akar terletak pada subinterval kedua, maka $x_l = x_r$, dan lanjutkan ke step 2.

c. $f(x_l)\ f(x_r) = 0$, akar = x_r, komputasi selesai.

Gambar 4.4 Algoritma Biseksi

Irfan Subakti - 司馬伊凡

Contoh 4.3 Bagidua

Gunakan Bagidua untuk menentukan akar dari $f(x) = e^{-x} - x$.

Dari grafik fungsi tersebut (gambar 4.1) terlihat bahwa harga akar terletak diantara 0 dan 1. Karenanya interval awal dapat dipilih dari $x_l = 0$ hingga $x_u = 1$. Dengan sendirinya, taksiran awal akar terletak di tengah interval tersebut:

$$x_r = \frac{0+1}{2} = 0{,}5$$

Taksiran ini menunjukkan kesalahan dari (harga sebenarnya adalah 0,56714329…):

$$E_t = 0{,}56714329 - 0{,}5 = 0{,}06714329$$

atau dalam bentuk relatif:

$$|\epsilon_t| = \left|\frac{0{,}06714329}{0{,}56714329}\right| 100\% = 11{,}8\%$$

dimana indeks t menunjukkan bahwa kesalahan diacu terhadap harga sebenarnya. Lalu:

$$f(0)\ f(0{,}5) = (1)\ (0{,}10653) = 0{,}10653$$

yang lebih besar dari nol, dengan sendirinya tak ada perubahan tanda terjadi antara x_l dan x_r.

Karena itu, akar terletak pada interval antara x = 0,5 dan 1,0. Batas bawah didefinisikan lagi sebagai x_l = 0,5 dan taksiran akar untuk iterasi kedua dihitung sebagai:

$$x_r = \frac{0,5+1,0}{2} = 0,75 \qquad |\epsilon_t| = 32,2\%$$

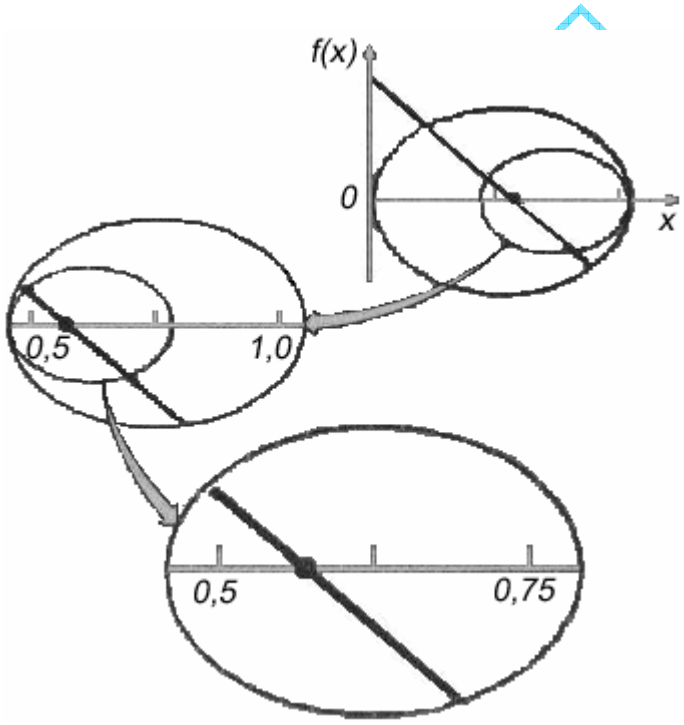
Proses dapat dilanjutkan lagi untuk memperhalus taksiran, misal untuk iterasi ketiga:

$$f(0,5)\, f(0,75) = -0,030 < 0$$

Karenanya akar terletak diantara 0,5 dan 0,75:

$$x_u = 0,75$$

$$x_r = \frac{0,5+0,75}{2} = 0,625 \qquad |\epsilon_t| = 10,2\%$$



Gambar 4.5 Grafik metode Bagidua. Gambar ini sesuai dengan 3 iterasi pertama dari contoh 4.3 di atas.

Dan iterasi keempatnya

$$f(0,5)\, f(0,625) = -0,010 < 0$$

Karenanya akar terletak diantara 0,5 dan 0,625:

$$x_u = 0,625$$

$$x_r = \frac{0,5+0,625}{2} = 0,5625 \qquad |\epsilon_t| = 0,819\%$$

Metode ini dapat diulangi lagi untuk memperoleh taksiran yang lebih halus.

4.2.1 Kriteria Berhenti dan Taksiran Kesalahan

Dari contoh sebelumnya, dicoba menentukan saat berhentinya komputasi yang dilakukan.

Diasumsikan bahwa kita akan berhenti jika angka kesalahan turun di bawah nilai tertentu, misal 0,1%. Strategi ditempuh karena taksiran kesalahan dalam contoh dihitung setelah sebelumnya kita sudah mengetahui akar sesungguhnya dari fungsi itu. Ini bukanlah merupakan realitas yang gampang ditemui, karenanya tak ada gunanya pemakaian metode ini jika akarnya telah kita ketahui.

Maka diperlukan suatu taksiran kesalahan dimana kita tak tahu mengenai akar sebelumnya. Sehingga kesalahan relatif aproksimasinya adalah:

$$|\epsilon_a| = \left| \frac{x_r^{baru} - x_r^{lama}}{x_r^{baru}} \right| 100\% \qquad [4.2]$$

Dimana x_r^{baru} adalah akar dari iterasi sekarang dan x_r^{lama} adalah akar dari iterasi sebelumnya.

Harga absolut dipakai karena kita biasanya cenderung memakai besarnya ϵ_a ketimbang tandanya.

Bila $|\epsilon_a|$ lebih kecil daripada suatu angka tertentu (ϵ_s), maka komputasi dihentikan.

Pada contoh 4.3 sebelumnya, dengan menggunakan rumus [4.2], didapat hasil-hasil seperti tabel 4.2 di bawah ini.

Irfan Subakti - 司馬伊凡

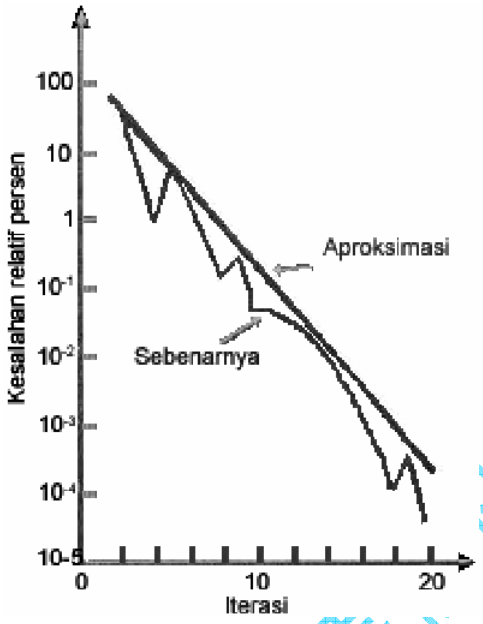
Tabel 4.2 Hasil Perhitungan contoh 4.3 dengan menggunakan persamaan [4.2]

Iterasi	x_r	$\lvert\epsilon_t\rvert$%	$\lvert\epsilon_a\rvert$%
1	0,5	11,8	
2	0,75	32,2	33,3
3	0,625	10,2	20,0
4	0,5625	0,819	11,1
5	0,59375	4,69	5,3

ϵ_a tak memberikan taksiran pasti dari kesalahan sebenarnya, gambar di atas menyarankan secara umum bahwa ϵ_a cenderung ke arah bawah ϵ_t.

Juga ϵ_a selalu lebih besar dari ϵ_t.

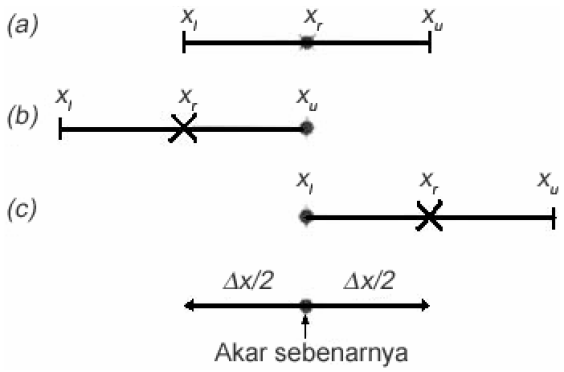
Jadi bila $\in_a$ jatuh di bawah $\in_s$, komputasi dihentikan dengan keyakinan bahwa akar telah diketahui sekurang-kurangnya sama telitinya dengan tingkat penentuan awal yang dapat diterima.



Gambar 4.7 Kesalahan untuk metode Bagidua. $\in_t$ dan taksiran digambar terhadap jumlah iterasi.

Dari gambar 4.7 di atas juga terlihat bahwa bentuk kurvanya tidak rata pada $\in_t$. Ini disebabkan dalam Bagidua, akar sebenarnya dapat terletak dimana saja dalam interval Akolade.

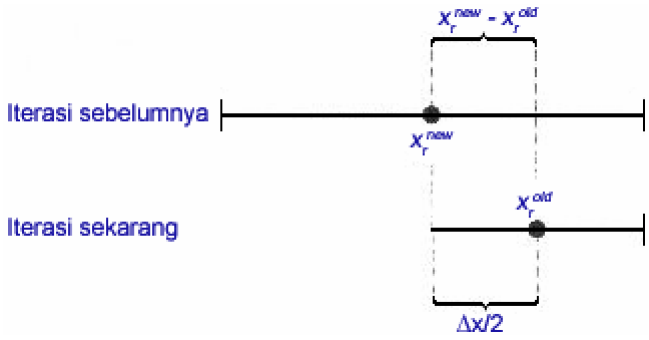
$\in_t$ dan aproksimasi akan berdekatan jika interval terpusat pada akar sebenarnya, dan jauh terpisah jika akar sebenarnya jatuh di salah satu ujung interval.



Gambar 4.8 Tiga cara dimana interval dapat mengurung akar

Pada gambar 4.8 diatas terlihat: (a) harga sebenarnya terletak di pusat interval. (b) dan (c) harga sebenarnya terletak di dekat ekstrim. Perhatikan bahwa ketidakcocokan diantara

harga sebenarnya dan titik tengah interval tak pernah melebihi separuh panjang interval atau $\Delta x/2$.



Gambar 4.9. Penjelasan grafik mengapa taksiran kesalahan untuk Bagidua

Pada gambar 4.9 di atas terlihat grafik yang menjelaskan mengapa Bagidua ($\Delta x/2$) ekuivalen terhadap taksiran akar untuk iterasi sekarang (x_r^{new}) dikurangi taksiran akar untuk iterasi sebelumnya (x_r^{old}).

Terdapat kesimpulan umum sementara, bahwa $\in_a$ selalu lebih besar dari $\in_t$.

Setiap kali suatu akar aproksimasi dapat ditempatkan dengan menggunakan Bagidua $x_r = (x_l + x_u) / 2$, diketahui bahwa akar sebenarnya terletak pada suatu tempat di dalam interval $(x_u - x_l) / 2 = \Delta x/2$.

Karenanya akar harus terletak di dalam $\pm \Delta x/2$ dari taksiran kita.

$\Delta x/2 = x_r^{new} - x_r^{old}$

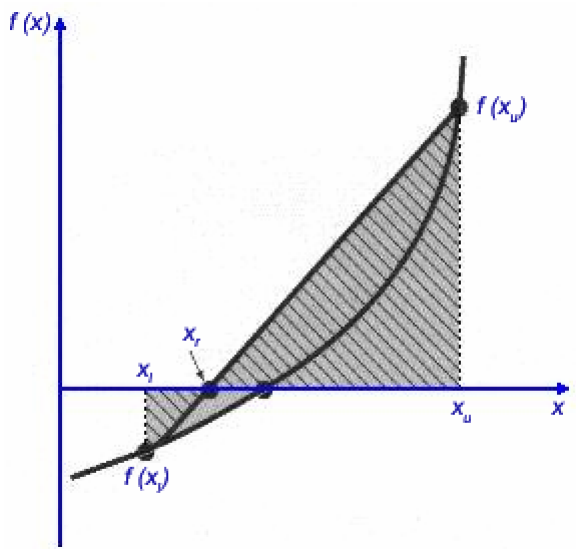
Irfan Subakti - 同馬伊凡

4.3 Metode Regula Falsi (False Position)

Disebut juga metode interpolasi linier.

$$x_r = x_u - \frac{f(x_u)(x_l - x_u)}{f(x_l) - f(x_u)}$$

Penjelasan grafiknya dapat dilihat pada gambar 4.10 di bawah ini. Segitiga serupa yang digunakan untuk menurunkan rumus buat metode tersebut adalah yang diarsir.



Gambar 4.10 Penjelasan grafik dari metode Regula Falsi

Contoh 4.4 Regula Falsi

Gunakan Regula Falsi untuk menentukan akar dari $f(x) = e^{-x} - x$. Akar sesungguhnya 0,56714329.

$x_l = 0$ dan $x_u = 1$.

Iterasi pertama

$x_l = 0$ $\quad f(x_l) = 1$

$x_u = 1$ $\quad f(x_u) = -0{,}63212$

$$x_r = 1 - \frac{-0{,}63212(0-1)}{1-(-0{,}63212)} = 0{,}6127$$

$$|\epsilon_t| = \left|\frac{0{,}56714329 - 0{,}6127}{0{,}56714329}\right| 100\% = 8\%$$

Iterasi kedua

$f(x_l)\, f(x_r) = -0{,}0708$ → akar pada subinterval I. x_r ada di batas atas berikutnya

$x_l = 0$ → $f(x_l) = 1$

$x_u = 0{,}6127$ → $f(x_u) = -0{,}0708$

Irfan Subakti - 司馬伊凡

$$x_r = 0{,}6127 - \frac{-0{,}0708(0 - 0{,}6127)}{1 - (-0{,}0708)} = 0{,}57219$$

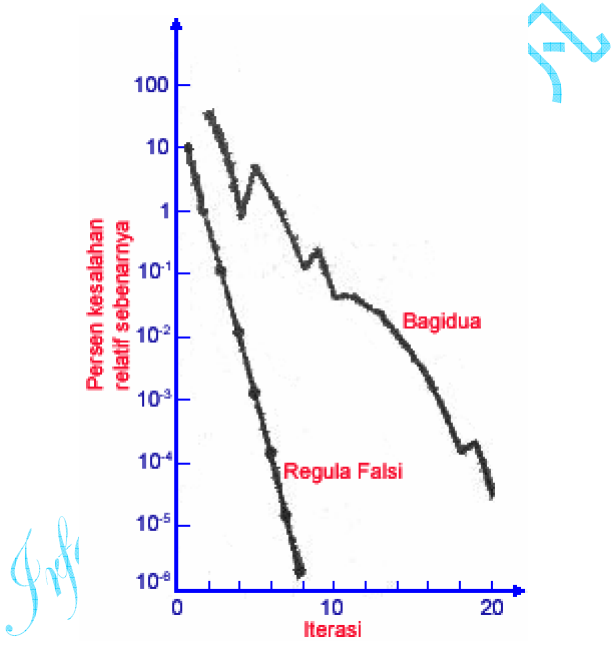
$$|\epsilon_t| = 0{,}89\%$$

$$|\epsilon_a| = \left|\frac{0{,}57219 - 0{,}6127}{0{,}57219}\right| 100\% = 7{,}08\%$$

Dst.

Kesalahan untuk Regula Falsi berkurang lebih cepat daripada Bagidua disebabkan rancangan yang lebih efisien untuk penempatan akar dalam Regula Falsi.

Pada gambar 4. 11 di bawah ini disajikan perbandingan ϵ_t pada metode Bagidua dan Regula Falsi untuk $f(x) = e^{-x} - x$.



Gambar 4.11 Perbandingan ϵ_t pada metode Bagidua dan Regula Falsi untuk $f(x) = e^{-x} - x$

Pada Bagidua, interval antara x_l dan x_u muncul semakin kecil selama komputasi. Interval, $\Delta x/2 = |x_u - x_l| / 2$, merupakan ukuran error untuk pendekatan ini.

Pada Bagidua, hal di atas tak terjadi, karena salah satu tebakan awal kondisinya tetap selama komputasi, sedangkan tebakan lainnya konvergen terhadap akar.

Pada contoh 4.4 di atas, x_l tetap pada 0, sedangkan x_u konvergen terhadap akar. Didapat, interval tak mengkerut, tapi agak mendekati suatu harga konstan.

4.3.1 Jebakan pada Metode Regula Falsi

Contoh 4.5 Bagidua lebih baik dari Regula Falsi

Gunakan Bagidua dan Regula Falsi untuk menempatkan akar di antara x = 0 dan 1,3 untuk: $f(x) = x^{10} - 1$.

Dengan Bagidua, didapatkan hasil seperti terlihat pada tabel 4.3 di bawah ini.

Tabel 4.3 Hasil Perhitungan dengan metode Bagidua

Iterasi	x_l	x_u	x_r	$\lvert\epsilon_t\rvert$%	$\lvert\epsilon_a\rvert$%
1	0	1,3	0,65	35	
2	0,65	1,3	0,975	2,5	33,3
3	0,975	1,3	1,1375	13,8	14,3
4	0,975	1,1375	1,05625	5,6	7,7
5	0,975	1,05625	1,015625	1,6	4,0

Setelah 5 iterasi, $\epsilon_t < 2\%$.

Kemudian dengan Regula Falsi, didapatkan hasil seperti terlihat pada tabel 4.4 di bawahi ini.

Tabel 4.4 Hasil Perhitungan dengan metode Regula Falsi

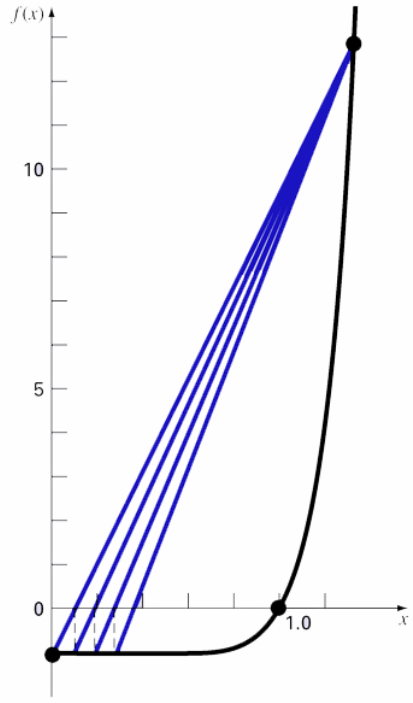
Iterasi	x_l	x_u	x_r	$\lvert\epsilon_t\rvert$%	$\lvert\epsilon_a\rvert$%
1	0	1,3	0,09430	90,6	
2	0,09430	1,3	0,18176	81,8	48,1
3	0,18176	1,3	0,26287	73,7	30,9
4	0,26287	1,3	0,33811	66,2	22,3
5	0,33811	1,3	0,40788	59,2	17,1

Setelah 5 iterasi, $\epsilon_t < 60\%$.

Juga $\lvert\epsilon_a\rvert < \lvert\epsilon_t\rvert$

Ternyata dengan Regula Falsi, ϵ_a ternyata meleset. Lebih jelas terlihat dalam grafik:

Irfan Subakti - 司馬伊凡



Gambar 4.12 Grafik dari f(x) = x^{10} – 1, menunjukkan konvergensi metode Regula Falsi yang lambat.

Terlihat, kurva menyalahi perjanjian yang mendasar Regula Falsi, yakni jika $f(x_l)$ lebih mendekati 0 dibanding $f(x_u)$, sehingga akan lebih dekat ke x_l daripada ke x_u (lihat gambar 4.10).

Karena bentuk fungsi yang sekarang, kebalikannya tentu juga benar.

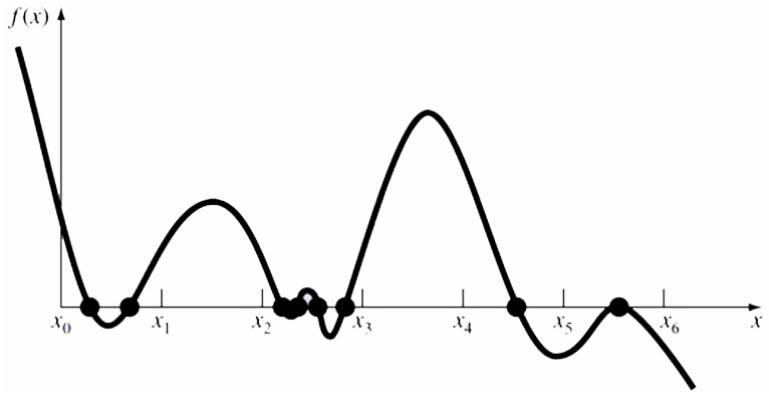
Yang harus dilakukan adalah memasukkan taksiran akar ke dalam persamaan semula dan ditentukan apakah hasil itu mendekati nol. Pengecekan semacam ini juga harus dilakukan pada program komputer untuk penempatan akar.

4.4 Incremental Searches dan Penentuan Tebakan Awal

Semua kemungkinan penempatan akar harus diperiksa.

Incremental search (pencarian inkremental) diikutsertakan pada saat memulai pemrograman komputer.

Dimulai pada ujung daerah yang diinginkan, lalu membuat evaluasi fungsi dengan kenaikan (inkremen) kecil di sepanjang daerah tersebut. Jika tanda fungsi berubah, harus dianggap bahwa suatu akar ada dalam kenaikan itu. Harga x pada saat permulaan dan akhir dari inkremen dapat memberikan tebakan awal bagi salah satu teknik Akolade yang sudah dibahas.



Gambar 4.13 Kasus-kasus dimana akar-akar dapat hilang karena panjang inkremen dari prosedur pencarian terlalu besar

Pada gambar 4.13 juga dapat kita amati bahwa akar terakhir adalah ganda dan akan dihilangkan tanpa memandang panjang inkremen.

Masalah yang ada yaitu panjang inkremen.

Jika terlalu kecil, pencarian akan menghabiskan waktu, jika terlalu besar mungkin saja akar-akar yang terpisah secara berdekatan akan hilang.

Terdapat juga masalah yaitu adanya kemungkinan akar ganda.

Untuk akar ganda, sedikit bantuan yang bisa diterapkan adalah dengan mencari turunan pertama fungsi (f'(x)), pada awal dan akhir setiap interval. Jika turunan ini berubah tanda, maka suatu minimal atau maksimal telah terjadi, dan interval harus diperiksa lebih dekat untuk mendapatkan kemungkinan adanya akar disitu.

Irfan Subakti 司馬伊凡

BAB 5 METODE TERBUKA

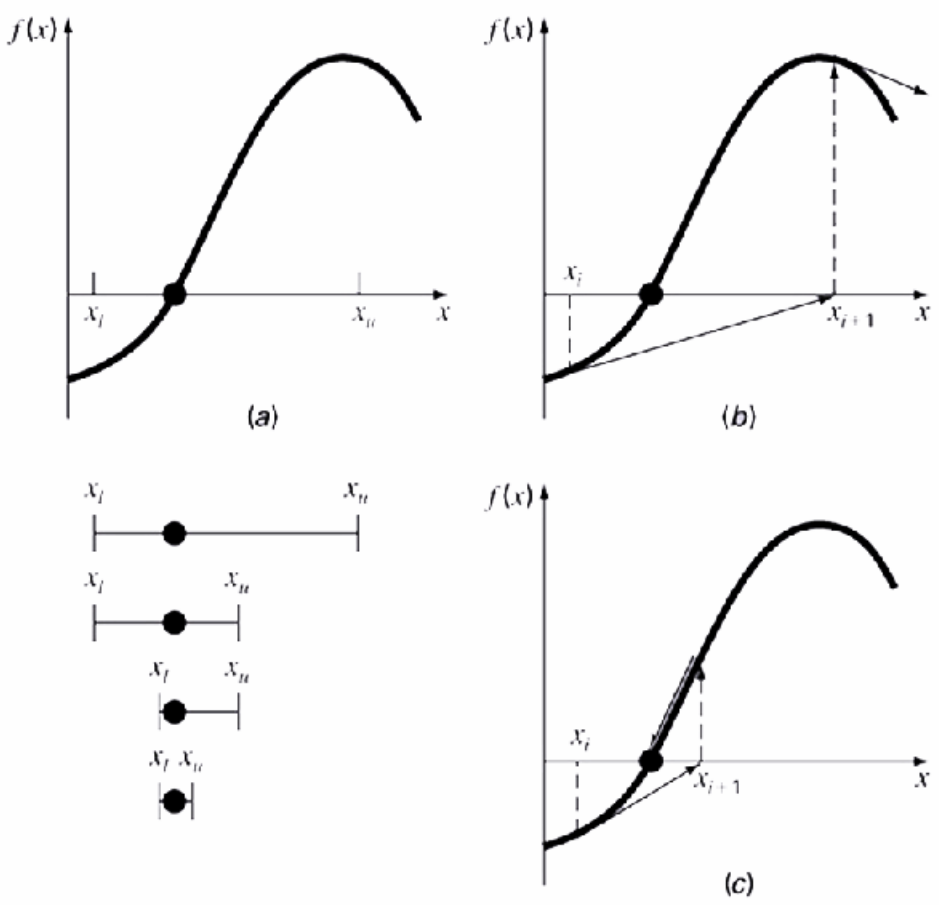
Akar dalam metode Akolade selalu dikurungi oleh batas bawah dan batas atas.

Metode Akolade selalu konvergen, yaitu makin lama makin mendekati nilai sesungguhnya.

Dalam metode Terbuka, pencarian dimulai dari harga tunggal x, atau 2 harga yang tak perlu mengurungi akar.

Metode Terbuka bisa divergen (semakin lama menjauhi nilai sebenarnya) atau konvergen, tapi kalau konvergen biasanya lebih cepat kelajuannya bila dibandingkan dengan metode Akolade.

Pada gambar 5.1 di bawah ini terlihat bahwa pada (a) yaitu metode Bagidua, akar dibatasi dalam interval yang ditentukan oleh x_l dan x_u. Pada metode Terbuka (b) dan (c), suatu formula dipakai untuk memproyeksikan x_i ke x_{i+1} dalam bentuk iteratif. Jadi metode dapat divergen (b) atau konvergen (c) secara cepat, tergantung pada harga tebakan awal.



Gambar 5.1. Penjelasan grafik mengenai perbedaan fundamental antara metode Akolade (a), dan metode Terbuka (b) dan (c) untuk menempatkan akar.

5.1 Iterasi Satu Titik Sederhana

Mengatur kembali fungsi f(x) = 0 sedemikian sehingga x berada pada ruas kiri persamaan:

$$x = g(x) \qquad [5.1]$$

Transformasi ini dapat dilakukan dengan manipulasi aljabar atau penambahan sederhana x ke kedua ruas persamaan. Misal:

$x^2 - 2x + 3 = 0 \quad \rightarrow \quad 2x = x^2 + 3$

$$x = \frac{x^2 + 3}{2} \text{ (dengan manipulasi aljabar)}$$

sin x = 0 akan dimasukkan dalam bentuk persamaan [5.1] dengan menambahkan x pada kedua ruas:

$x = \sin x + x$ (dengan penambahan sederhana x)

Persamaan [5.1] dapat memperkirakan sebuah harga x, sebagai fungsi dari x. Jadi dengan adanya tebakan awal x_i, dapat dihitung suatu taksiran baru x_{i+1} yang dapat dinyatakan:

$$x_{i+1} = g(x_i) \qquad [5.2]$$

Seperti rumus iterasi lain, maka kesalahan aproksimasinya:

$$|\epsilon_a| = \left| \frac{x_{i+1} - x_i}{x_{i+1}} \right| 100\%$$

Irfan Subakti - 司馬伊凡

Contoh 5.1 Iterasi satu titik sederhana

Gunakan iterasi satu titik sederhana untuk menempatkan akar dari $f(x) = e^{-x} - x$.

Solusi. Fungsi dapat dipisahkan secara langsung dan dinyatakan dalam bentuk persamaan [5.2] sebagai $x_{i+1} = e^{-x_i}$. Dimulai dari tebakan awal $x_0 = 0$. Maka didapat hasil seperti diperlihat pada tabel 5.1 di bawah ini.

Tabel 5.1 Hasil Perhitungan dengan metode Iterasi satu titik sederhana

Iterasi	x_i	$\|\epsilon_t\|$%	$\|\epsilon_a\|$%
0	0	100	
1	1,000000	76,3	100,0
2	0,367879	35,1	171,8
3	0,692201	22,1	46,9
4	0,500473	11,8	38,3
5	0,606244	6,89	17,4
6	0,545396	3,83	11,2
7	0,579612	2,20	5,90
8	0,560115	1,24	3,48
9	0,571143	0,705	1,93
10	0,564879	0,399	1,11

5.1.1 Konvergensi

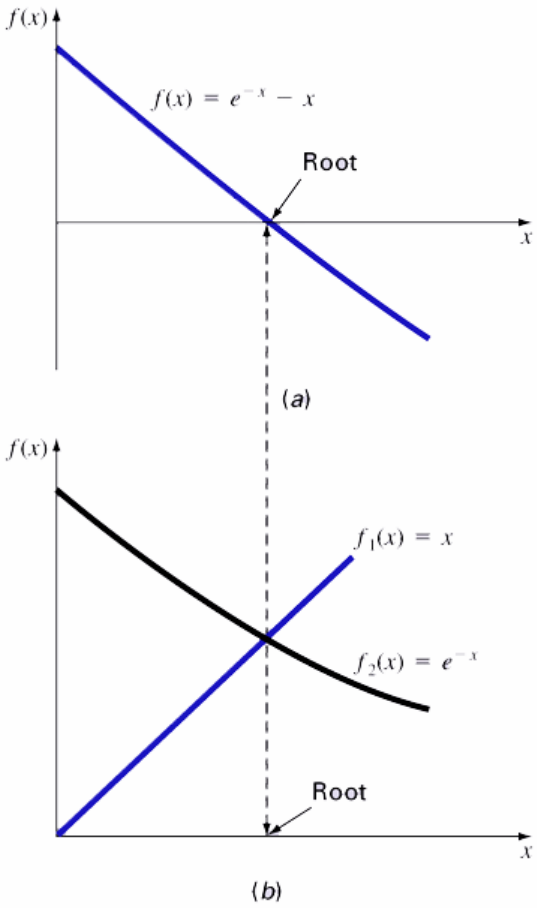
Dari tabel di atas, terlihat bahwa ϵ_a untuk setiap iterasi dari contoh 5.1 secara kasar sebanding (oleh suatu faktor kira-kira 0,5 hingga 0,6) dengan kesalahan dari iterasi sebelumnya.

Hal itu disebut dengan konvergensi linier, merupakan sifat iterasi satu titik.

Kelajuan konvergensi diatas, membuat kita ingin mengetahui juga "kemungkinan" konvergensi.

Fungsi ini digambarkan lagi seperti gambar 5.2a di bawah ini.

Irfan Subakti - 同焦伊尤



Gambar 5.2 Dua metode grafik alternatif untuk menentukan akar dari $f(x) = e^{-x} - x$

Irfan Subakti

Pada gambar 5.2 di atas terlihat bahwa (a) Akar berada pada titik dimana ia memotong sumbu x; (b) akar pada perpotongan dari fungsi-fungsi komponen.

Suatu pendekatan grafik alternatif ialah memisahkan persamaan f(x) = 0 menjadi 2 bagian komponen seperti dalam:

$$f_1(x) = f_2(x)$$

Kemudian kedua persamaan:

$$y_1 = f_1(x) \qquad [5.3]$$

dan

$$y_2 = f_2(x) \qquad [5.4]$$

Dapat digambarkan secara terpisah (gambar 5.2b). Harga-harga x yang bersesuaian dengan perpotongan kedua fungsi dinyatakan oleh akar f(x) = 0.

5.1.2 Metode Grafik 2 Kurva

Contoh 5.2 Metode grafik 2 kurva

Pernyataan masalah: pisahkan persamaan $e^{-x} - x = 0$ menjadi 2 bagian dan tentukan akarnya secara grafik.

Solusi: tuliskan kembali persamaan sebagai $y_1 = x$ dan $y_2 = e^{-x}$. Didapat perhitungan yang disajikan pada tabel 5.2 di bawah ini.

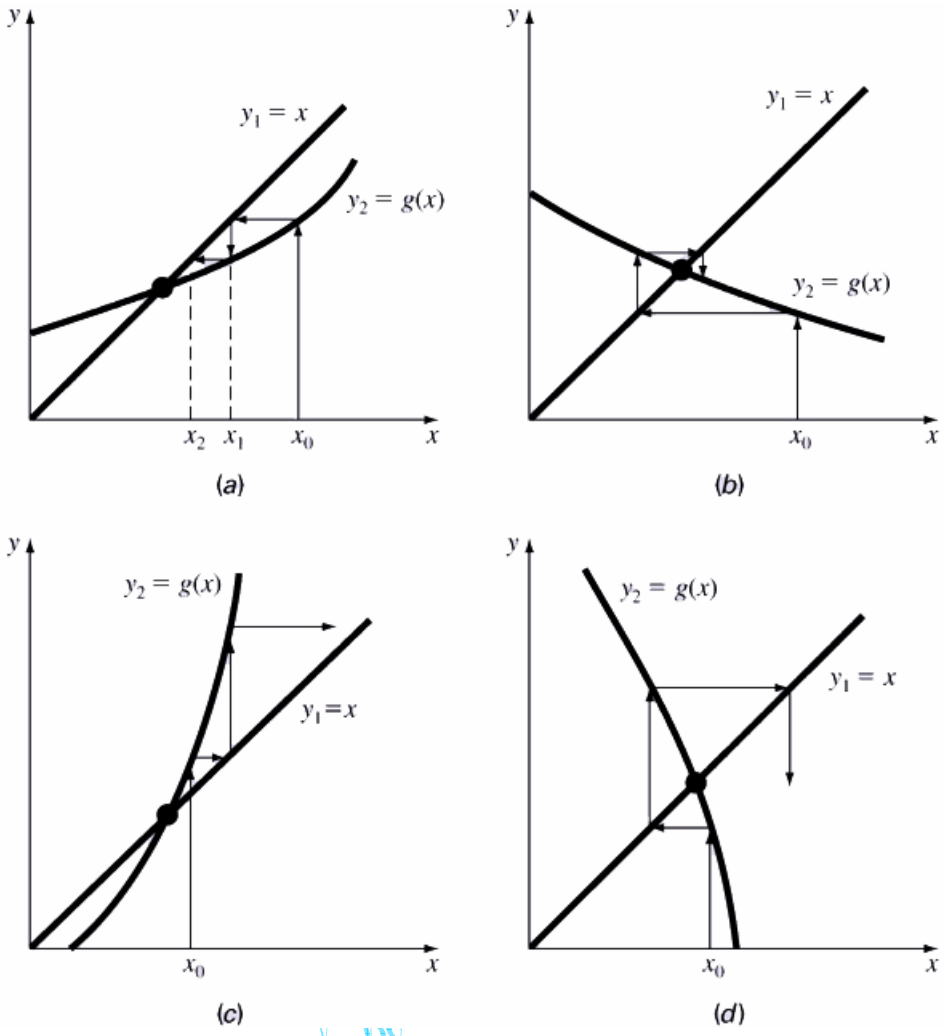
Tabel 5.2 Hasil Perhitungan dengan metode grafik 2 kurva

x	y_1	y_2
0,0	0,0	1,000
0,2	0,2	0,819
0,4	0,4	0,670
0,6	0,6	0,549
0,8	0,8	0,449
1,0	1,0	0,368

Titik-titik ini telah digambarkan pada gambar 5.2b. Perpotongan dari kedua kurva tampak pada taksiran akar x = 0,57 yang bersesuaian dengan titik dimana kurva tunggal dalam gambar 5.2a memotong sumbu x.

Metode 2 kurva sekarang dapat digunakan untuk melukiskan konsep konvergensi dan divergensi dari metode iterasi 1 titik, seperti gambar 5.3 di bawah ini.

Irfan Subakti - 司馬伋凡



Gambar 5.3 Penjelasan grafik mengenai konvergensi (a) dan (b), serta divergensi (c) dan (d) dari iterasi 1 titik sederhana.

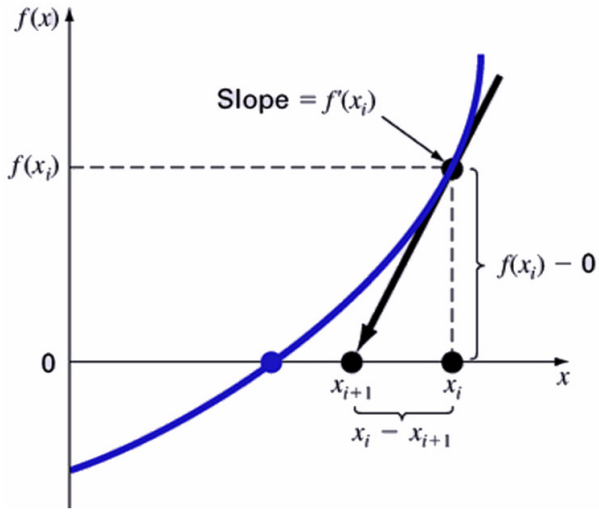
Irfan Subakti

Pada gambar 5.3 di atas grafik (a) dan (c) dinamakan pola monoton, sedangkan (b) dan (d) dinamakan osilasi atau pola spiral. Perhatikan bahwa konvergensi terjadi bila $|g'(x)| < 1$.

5.2 Metode Newton-Raphson

$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)} \qquad [5.5]$$

Pada gambar 5.4 di bawah ini akan digambarkan penjelasan grafik mengenai metode Newton-Raphson. Pada gambar tersebut garis singgung terhadap fungsi pada x_i [yakni $f'(x_i)$] diekstrapolasikan ke bawah terhadap sumbu x untuk memberikan sebuah taksiran akar pada x_{i+1}.



Gambar 5.4 Penjelasan grafik dari metode Newton-Raphson

Contoh 5.3. Metode Newton-Raphson (NR)

Pernyataan masalah: gunakan NR untuk menaksir akar dari $e^{-x} - x$ dengan $x_0 = 0$.

Solusi

Turunan pertama dari fungsi adalah:

$$f'(x) = -e^{-x}-1$$

Yang dapat disubstitusikan ke dalam persamaan [5.5]:

$$x_{i+1} = x_i - \frac{e^{-x_i} - x_i}{-e^{-x_i} - 1}$$

Dimulai dari tebakan awal $x_0 = 0$, didapatkan hasil seperti disajikan pada tabel 5.3 di bawah ini.

Tabel 5.3 Hasil Perhitungan dengan metode Newton-Raphson

Iterasi, i	x_i	$\|\epsilon_t\|$%
0	0	100
1	0,500000000	11,8
2	0,566311003	0,147
3	0,567143165	0,0000220
4	0,567143290	$<10^{-8}$

5.2.1 Kriteria Berhenti dan Taksiran Kesalahan

Laju konvergensi dinyatakan oleh $E_{i+1} = O(E_i^2)$

Jadi secara kasar, kesalahan sebanding terhadap pangkat 2 dari kesalahan sebelumnya.

Dengan kata lain, jumlah angka signifikan dari ketelitian (akurasi) berlipat 2 pada setiap kali iterasi.

Contoh 5.4 Analisis kesalahan Metode NR

Pernyataan masalah: Metode NR adalah konvergen secara kuadratik. Artinya secara kasar, kesalahan itu sebanding dengan pangkat 2 dari kesalahan sebelumnya, seperti pada:

$$E_{t,i+1} \approx - \frac{f''(x_r)}{2f'(x_r)} E_{t,i}^2$$

Solusi: turunan pertama dari $f(x) = e^{-x} - x$ adalah:

$$f'(x) = -e^{-x} - 1$$

yang dapat dievaluasikan pada $x_r = 0{,}56714329$ sebagai

$$f'(0{,}56714329) = -1{,}56714329$$

Turunan kedua adalah:

$$f''(x) = e^{-x}$$

yang dapat dievaluasikan sebagai:

$$f''(0{,}56714329) = 0{,}56714329$$

Hasil-hasil diatas dimasukkan dalam persamaan:

$$E_{t,i+1} \approx - E_{t,i}^2$$

atau

$$E_{t,i+1} \approx 0{,}18095\ E_{t,i}^2$$

Dari contoh 5.3, kesalahan awal adalah $E_{t,0} = 0{,}56714329$ yang dapat dimasukkan ke dalam persamaan kesalahan untuk menaksir:

$$E_{t,i} \approx 0{,}18095\ (0{,}56714329)2 = 0{,}0582$$

yang mendekati kesalahan sebenarnya = 0,06714329.

Iterasi berikutnya:

$$E_{t,2} \approx 0{,}18095\ (0{,}06714329)2 = 0{,}0008158$$

yang setara dibandingkan dengan kesalahan sebenarnya = 0,0008323.

Iterasi ketiga:

Irfan Subakti - 司馬伊凡

$E_{t,3} \approx 0{,}18095\ (0{,}0008323)2 = 0{,}000000125$

yang secara tepat adalah kesalahan yang diperoleh dalam contoh 5.3.

Taksiran kesalahan memperbaiki cara ini karena semakin kita lebih mendekati akar, x_i dan ξ diaproksimasikan lebih baik dengan x_r.

Akhirnya:

$E_{t,4} \approx 0{,}18095\ (0{,}000000125)2 = 2{,}83 \times 10^{-15}$

Jadi, kesalahan metode NR dalam hal ini, secara kasar, ternyata sebanding (dengan faktor 0,18095) dengan pangkat 2 dari kesalahan iterasi sebelumnya.

5.2.2 Jebakan pada Metode Newton-Raphson

Contoh 5.5 Fungsi yang konvergen secara perlahan menggunakan Metode NR

Pernyataan masalah: Tentukan akar positif dari $f(x) = x^{10} - 1$ menggunakan metode NR dan tebakan awal x = 0,5.

Solusi

$x_{i+1} = x_i - \dfrac{x_i^{10} - 1}{10 x_i^{9}}$, didapatkan hasil seperti disajikan pada tabel 5.4 di bawah ini.

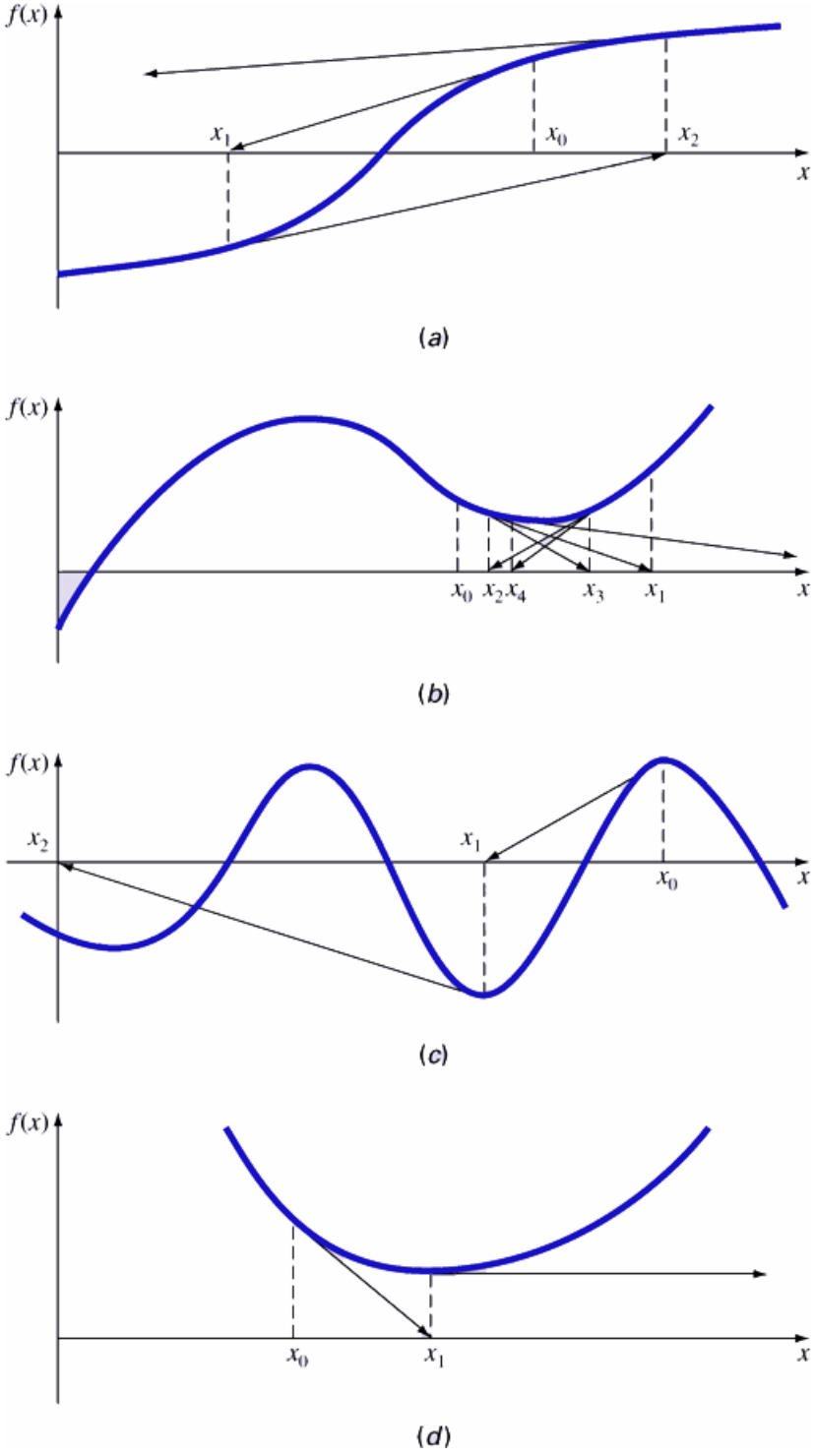
Irfan Subakti - 司馬伊凡

Tabel 5.4 Hasil Perhitungan untuk jebakan pada metode Newton-Raphson

Iterasi	x_i
0	0,5
1	51,65
2	46,485
3	41,8365
4	37,65285
5	33,887565
...	...
∞	1,00000000

Setelah taksiran pertama, ia akan konvergen pada akar sebenarnya, yakni 1, tapi dengan kelajuan yang sangat perlahan.

Di samping konvergensi yang perlahan yang disebabkan sifat bawaan fungsi, kesukaran lain timbul, seperti diperlihatkan pada gambar 5.5 di bawah ini.



Gambar 5.5. Empat kasus dimana metode Newton-Raphson memperlihatkan konvergensi yang kurang baik

Gambar 5.5a memperlihatkan kasus dimana sebuah titik belok, yakni f''(x) = 0, terjadi dalam kekosongan suatu akar. Perhatikan bahwa iterasi dimulai pada x_0 dan berlanjut dari akar secara divergen.

Gambar 5.5b memperlihatkan kecenderungan dari teknik NR untuk berosilasi di sekitar harga maksimal dan minimal setempat. Osilasi yang demikain bisa tertahan atau, seperti

dalam gambar 5.5b, sebuah kemiringan yang mendekati nol dicapai dimana solusi ditempatkan jauh dari daerah yang diinginkan.

Gambar 5.5c memperlihatkan bagaimana sebuah tebakan awal yang mendekati suatu akar dapat meloncat ke suatu lokasi jauh dari beberapa akar. Kecenderungan menjauh dari daerah yang diinginkan disebabkan ditemukannya nilai kemiringan yang mendekati nol. Tentunya harga kemiringan nol, yakni f'(x) = 0, tak dibolehkan dalam metode NR.

Secara grafik (gambar 5.5d) berarti solusi bergerak secara horizontal dan tak pernah memotong sumbu x. Jadi kita harus mempunyai tebakan awal yang mendekati akar.

5.3 Metode Secant

Masalah yang didapat dalam metode Newton-Raphson adalah terkadang sulit mendapatkan turunan pertama, yakni f'(x). Sehingga dengan jalan pendekatan

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

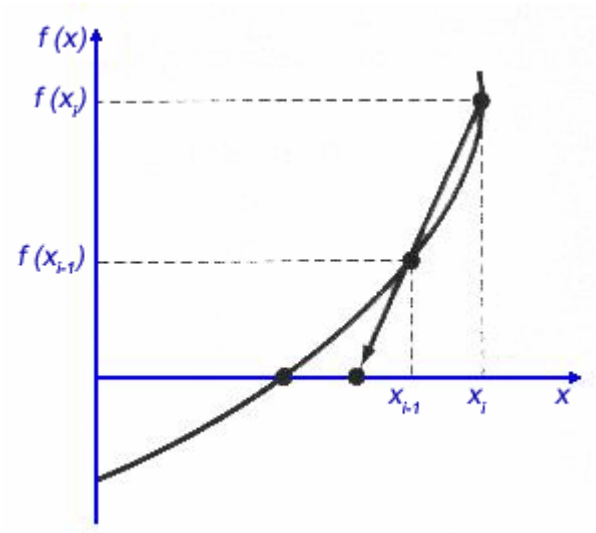
Didapat:

$$x_{i+1} = x_i - \frac{f(x_i)(x_{i-1} - x_i)}{f(x_{i-1}) - f(x_i)} \qquad [5.6]$$

Persamaan di atas memang memerlukan 2 taksiran awal x, tetapi karena f(x) tidak membutuhkan perubahan tanda diantara taksiran → maka Secant bukan metode Akolade.

Penjelasan grafik mengenai metode Secan dapat dilihat pada gambar 5.6 di bawah ini. Teknik ini serupa dengan teknik Newton-Raphson (gambar 5.4) dalam arti bahwa suatu taksiran akar diramalkan oleh ekstrapolasi sebuah garis singgung dari fungsi terhadap sumbu x. Tetapi metode Secant lebih menggunakan diferensi daripada turunan untuk memperkirakan kemiringan/slope.

Irfan Subakti - 司馬伊凡



Gambar 5.6 Penjelasan grafik mengenai metode Secant

Contoh 5.6 Metode Secant

Pernyataan masalah: taksir akar dari $f(x) = e^{-x} - x$ menggunakan metode Secant dan taksiran awal $x_{-1} = 0$ dan $x_0 = 1{,}0$.

Solusi

Ingat bahwa akar sesungguhnya adalah 0,56714329…

Iterasi 1:

$x^{-1} = 0 \quad \rightarrow \quad f(x_{-1}) = 1{,}00000$

$x_0 = 0 \quad \rightarrow \quad f(x_0) = -0{,}63212$

$$x_1 = 1 - \frac{-0{,}63212(0-1)}{1-(-0{,}63212)} = 0{,}61270 \qquad |\epsilon_t| = 8{,}0\%$$

Iterasi 2:

$x_0 = 1 \quad \rightarrow \quad f(x_0) = -0{,}63212$

$x_1 = 0{,}61270 \quad \rightarrow \quad f(x_1) = -0{,}07081$

Kedua taksiran sekarang pada ruas akar yang sama.

$$x_2 = 0{,}61270 - \frac{-0{,}07081(1-0{,}61270)}{-0{,}63212-(-0{,}07081)} = 0{,}56384 \qquad |\epsilon_t| = 0{,}58\%$$

Irfan Subakti - 司馬伊凡

Iterasi 3:

$x_1 = 0{,}61270 \rightarrow \quad f(x_1) = -0{,}07081$

$x_2 = 0{,}56384 \rightarrow \quad f(x_2) = 0{,}00518$

$$x_3 = 0{,}56384 - \frac{-0{,}00518(0{,}62170 - 0{,}56384)}{-0{,}07081 - (-0{,}00518)} = 0{,}56717 \quad |\in_t| = 0{,}0048\%$$

5.3.1 Perbedaan Metode Secant dan Regula Falsi

Persamaan di metode Secant maupun Regula Falsi identik suku demi suku.

Keduanya menggunakan 2 taksiran awal untuk menghitung aproksimasi slope fungsi yang digunakan untuk berproyek terhadap sumbu x untuk taksiran baru akar.

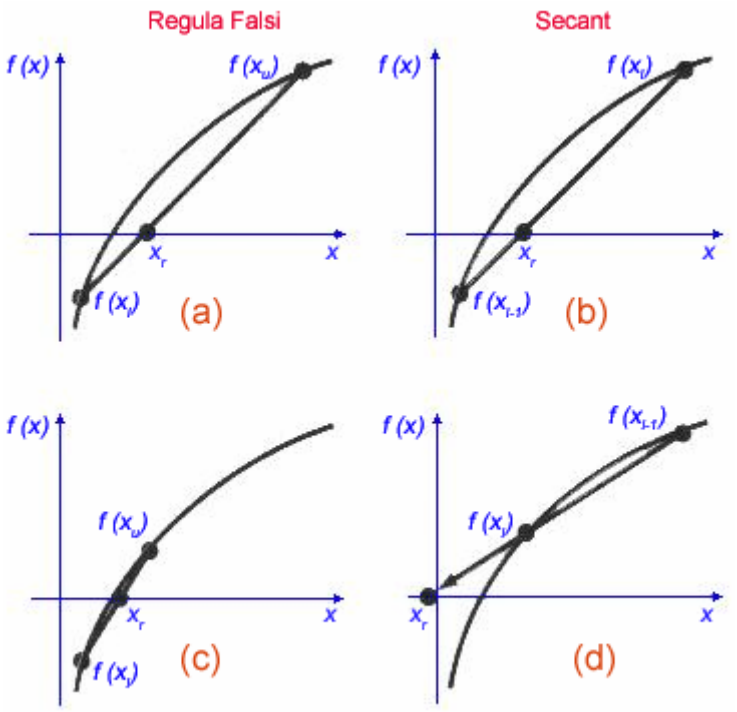
Perbedaannya pada harga awal yang digantikan oleh taksiran baru.

Dalam Regula Falsi, taksiran terakhir akar menggantikan harga asli mana saja yang mengandung suatu harga fungsi dengan tanda yang sama seperti $f(x_r)$. Sehingga 2 taksiran senantiasa mengurung akar.

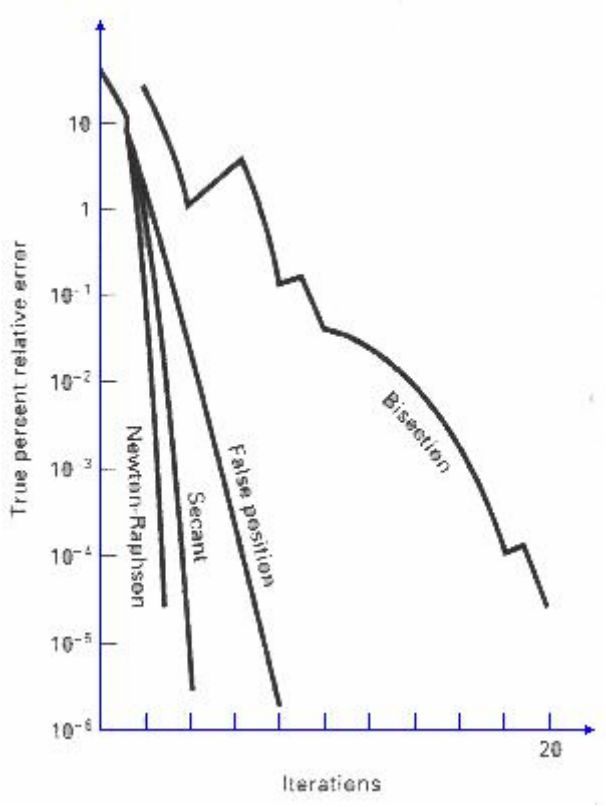
Secant mengganti harga-harga dalam deretan yang ketat, dengan harga baru x_{i+1} menggantikan x_i, dan x_i menggantikan x_{i-1}. Sehingga 2 harga terkadang dapat terletak pada ruas akar yang sama. Pada kasus tertentu ini bisa divergen.

Pada gambar 5.7 di bawah ini disajikan penggunaan metode Regula Falsi dan Secant untuk menaksir akar f(x) = ln x, dimulai dari harga $x_1 = x_{i-1} = 0{,}5$ dan $x_u = x_i = 5{,}0$. Terlihat disitu bahwa iterasi pertama (a) dan (b) untuk iterasi kedua metode adalah identik. Tetapi pada iterasi kedua (c) dan (d), titik yang dipakai berbeda. Sebagai akibatnya, metode Secant dapat divergen seperti dalam (d).

Irfan Subakti - 司馬伊凡



Gambar 5.7 Perbandingan metode Regula Falsi dan Secant



Gambar 5.8 Perbandingan ϵ_t untuk metode mencari akar f(x) = e^{-x} – x

5.4 Akar Ganda

Satu akar ganda berhubungan dengan suatu titik dimana sebuah fungsi menyinggung sumbu x.

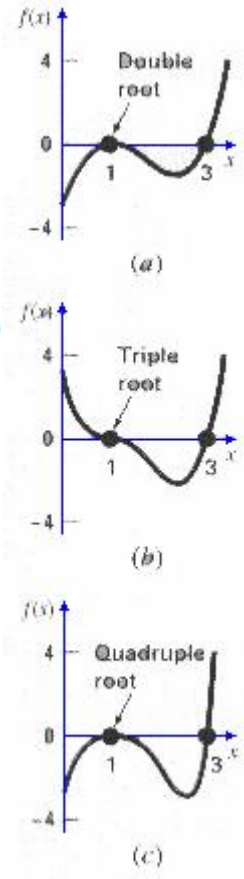
Misal akar dobel dihasilkan dari:

$$f(x) = (x - 3)(x - 1)(x - 1)$$

atau dengan pengalian suku-suku:

$$f(x) = x^3 - 5x^2 + 7x - 3$$

Persamaan diatas memiliki akar dobel, karena 1 akar x membuat kedua suku dalam persamaan itu sama dengan nol. Secara grafik, ini sesuai dengan kurva yang menyentuh sumbu x secara tangensial pada akar dobel. Ini dapat dilihat pada gambar 5.9a di bawah ini pada x = 1. Perhatikan bahwa fungsi tak memotong sumbu pada kedua sisi akar ganda genap (a) dan (c), sedangkan ia memotong sumbu untuk kasus ganjil (b).



Gambar 5.9 Contoh akar ganda yang menyinggung sumbu x

Akar tripel untuk kasus dimana satu harga x membuat 3 suku dalam suatu persamaan menjadi nol, misal:

$$f(x) = (x - 3)(x - 1)(x - 1)(x - 1)$$

atau dengan pengalian suku-suku:

$$f(x) = x^4 - 6x^3 + 12x^2 - 10x + 3$$

Kesulitan yang ditimbulkan oleh akar ganda

Hasil dari metode Akolade berkurang kepercayaannya dengan adanya kenyataan bahwa fungsi tak berubah tanda pada akar ganda genap. Pada metode Terbuka, ini bisa menyebabkan divergensi.

Tak hanya f(x) tapi juga f'(x) menuju nol pada akar.

Pada metode Newton-Raphson dan Secant, dimana keduanya mengandung turunan (atau taksiran) di bagian penyebut pada rumusnya, terjadi pembagian dengan nol jika solusi konvergen sangat mendekati akar.

Menurut Ralston dan Rabinowitz, f(x) selalu mencapai nol sebelum f'(x). Sehingga kalau pemeriksaan nol untuk f(x) disertakan dalam program, maka komputasi berhenti sebelum f'(x) mencapai nol.

Metode Newton-Raphson dan Secant konvergen secara linier (bukan kuadratik), konvergen untuk akar-akar ganda (Ralston dan Rabinowitz).

Irfan Subakti - 司馬伊凡

Selanjutnya, masih menurut Ralston dan Rabinowitz [RAL1978]:

$$u(x) = \frac{f(x)}{f'(x)} \qquad [5.7]$$

5.4.1 Metode Modifikasi Newton-Raphson

$$x_{i+1} = x_i - \frac{f(x_i)f'(x_i)}{[f'(x_i)]^2 - f(x_i)f''(x_i)} \qquad [5.8]$$

5.4.2 Metode Modifikasi Secant

$$x_{i+1} = x_i - \frac{u(x_i)(x_{i-1} - x_i)}{u(x_{i-1}) - u(x_i)} \qquad [5.9]$$

5.5 Perbandingan Pelbagai Metode

Dari pelbagai metode, baik metode Akolade maupun Terbuka, maka dapat disimpulkan seperti disajikan dalam tabel 5.5 berikut ini.

Tabel 5.5 Perbandingan pelbagai metode

Metode	**Teba- kan awal**	**Laju konversi relatif**	**Stabilitas**	**Akurasi**	**Luas aplikasi**	**Upaya pemro-graman**	**Komentar**
Langsung	-	-	-	-	Sangat terbatas	-	-
Grafik	-	-	-	Kurang	Akar sesung-guhnya	-	Memakan waktu lebih banyak daripada metode numerik
Bagidua	2	Perlahan	Selalu konverge n	Baik	Akar sesung-guhnya	Mudah	-
Regula Falsi	2	Sedang	Selalu konverge n	Baik	Akar sesung-guhnya	Mudah	-
Iterasi satu titik	1	Perlahan	Bisa divergen	Baik	Umum	Mudah	-
Newton-Raphson	1	Cepat	Bisa divergen	Baik	Umum, dibatasi jika f'(x)=0	Mudah	Memerluka n evaluasi f'(x)
Modifikasi Newton-Raphson	1	Cepat bagi akar berganda ; sedang bagi akar tunggal	Bisa divergen	Baik	Umum, didesain khusus bagi akar bergand a	Mudah	Memerluka n evaluasi f''(x) dan f'(x)
Secant	2	Sedang hingga cepat	Bisa divergen	Baik	Umum	Mudah	Tebakan awal tak harus mengurung akar
Modifikasi Secant	1	Sedang hingga cepat	Bisa divergen	Baik	Umum	Mudah	-

Irfan Subakti - 司馬伊凡

BAB 6 ELIMINASI GAUSS

6.1 MATRIKS

Sebuah matriks terdiri dari deretan elemen berbentuk persegi panjang dan dinyatakan oleh simbol tunggal.

[A] adalah notasi kependekan matriks.

a_{ij} melambangkan masing-masing elemen matriks.

Sekumpulan elemen horizontal disebut: baris.

Sekumpulan elemen vertikal disebut: kolom.

Subskrip pertama i menandai baris, di dalam mana elemen itu terletak.

Subskrip kedua j menandai kolom. Misal elemen a_{23} terletak pada baris 2 dan kolom 3.

Kolom 3

$$[A] = \begin{bmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots \\ a_{m1} & a_{m2} & a_{m3} & \dots & a_{mn} \end{bmatrix}$$

Baris 2

Matriks dengan dimensi baris, m = 1, disebut dengan vektor baris:

$$[B] = [b_1\ b_2\ \dots\ b_n]$$

Matriks dengan dimensi kolom, n = 1, disebut dengan vektor kolom:

$$[C] = \begin{bmatrix} c_1 \\ c_2 \\ \dots \\ c_m \end{bmatrix}$$

Matriks dimana m = n disebut dengan matriks bujur sangkar. Misal matriks 4 x 4 adalah:

$$[A] = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix}$$

Diagonal yang terdiri dari elemen a_{11}, a_{22}, a_{33}, dan a_{44} dinamakan diagonal utama matriks.

Irfan Subakti

Matriks bujur sangkar

Matriks Simetris → $a_{ij} = a_{ji}$ untuk semua i dan j.

$$[A] = \begin{bmatrix} 3 & 2 & 9 \\ 2 & 5 & 6 \\ 9 & 6 & 7 \end{bmatrix}$$

Matriks Diagonal → semua elemen, kecuali diagonal utama, sama dengan nol.

$$[A] = \begin{bmatrix} a_{11} & & & \\ & a_{22} & & \\ & & a_{33} & \\ & & & a_{44} \end{bmatrix}$$

Matriks Identitas → matriks diagonal dimana semua elemen pada diagonal utama = 1.

$$[A] = \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ & & & 1 \end{bmatrix}$$

Matriks Triangular Atas → semua elemen di bawah diagonal utama adalah nol.

$$[A] = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ & a_{22} & a_{23} & a_{24} \\ & & a_{33} & a_{34} \\ & & & a_{44} \end{bmatrix}$$

Matriks Triangular Bawah → semua elemen di atas diagonal utama adalah nol.

$$[A] = \begin{bmatrix} a_{11} & & & \\ a_{21} & a_{22} & & \\ a_{31} & a_{32} & a_{33} & \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix}$$

Matriks Pita → elemennya sama dengan nol, kecuali pita (band) yang dipusatkan pada diagonal utama.

$$[A] = \begin{bmatrix} a_{11} & a_{12} & & \\ a_{21} & a_{22} & a_{23} & \\ & a_{32} & a_{33} & a_{34} \\ & & a_{43} & a_{44} \end{bmatrix}$$

Matriks Pita yang mempunyai lebar pita = 3 → matriks Tridiagonal.

Irfan Subakti - 司馬伊凡

Aturan pengoperasian matriks

2 matriks m x n adalah sepadan, jika setiap elemen pada matriks pertama = setiap elemen pada matriks kedua → [A] = [B], jika $a_{ij} = b_{ij}$ untuk semua i dan j.

Penambahan 2 matriks → [C] = [A] + [B], $c_{ij} = a_{ij} + b_{ij}$; untuk semua i = 1, 2, ⋯ m dan j = 1, 2, ⋯ n.

Pengurangan 2 matriks → [D] = [E] - [F], $d_{ij} = e_{ij} - f_{ij}$; untuk semua i = 1, 2, ⋯ m dan j = 1, 2, ⋯ n.

Penambahan dan pengurangan adalah Komutatif:

$$[A] + [B] = [B] + [A] \quad \text{dan} \quad [E] - [F] = -[F] + [E]$$

Penambahan dan pengurangan adalah Asosiatif:

$$[A] + ([B] + [C]) = ([A] + [B]) + [C]$$

Perkalian matriks [A] oleh suatu skalar g didapat dengan mengalikan setiap elemen dari [A] dengan g:

$$[B] = g[A] = \begin{bmatrix} ga_{11} & ga_{12} & ga_{13} & \cdots & ga_{1n} \\ ga_{21} & ga_{22} & ga_{23} & \cdots & ga_{2n} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ ga_{m1} & ga_{m2} & ga_{m3} & \cdots & ga_{mn} \end{bmatrix}$$

Hasil kali 2 matriks dinyatakan sebagai [C] = [A][B] dimana elemen [C] didefinisikan sebagai:

$$c_{ij} = \sum_{k=1}^{n} a_{ik} b_{kj}$$

dimana n = dimensi kolom dari [A] dan dimensi baris dari [B].

Syarat perkalian 2 matriks $[A]_{m \times n}$ x $[B]_{n \times p}$ adalah n-nya harus sama, jadi jumlah kolom di matriks [A] harus sama dengan jumlah baris di matriks [B].

Contoh: $[A]_{3 \times 5}$ x $[B]_{5 \times 7}$ yang menghasilkan $[C]_{3 \times 7}$

Jika dimensi kedua matriks cocok, maka perkalian matriks adalah Asosiatif:

$$([A]\,[B])\,[C] = [A]\,([B]\,[C])$$

Juga Distributif:

$$[A]\,([B] + [C]) = [A]\,[B] + [A]\,[C]$$

Irfan Subakti 司馬伊凡

Atau

$$([A] + [B])\,[C] = [A]\,[C] + [B]\,[C]$$

Tetapi perkalian matriks tak selalu komutatif:

$$[A]\,[B] \neq [B]\,[A]$$

Yang terpenting adalah orde perkaliannya. Seperti contoh di atas: $[A]_{3 \times 5}$ x $[B]_{5 \times 7}$ yang menghasilkan $[C]_{3 \times 7}$ tapi tidak sama dengan $[B]_{5 \times 7}$ x $[A]_{3 \times 5}$ yang malahan tak bisa dilakukan operasi perkalian.

Jika [A] merupakan matriks bujur sangkar, terdapat matriks $[A]^{-1}$ yang disebut inversi [A]:

$$[A][A]^{-1} = [A]^{-1}[A] = [I]$$

Jadi perkalian suatu matriks inversi adalah analog dengan pembagian, dalam arti bahwa suatu bilangan yang dibagi dengan bilangan itu sendiri adalah 1. Artinya perkalian suatu matriks dengan inversinya menghasilkan matriks identitas.

$$[A]^{-1} = \frac{1}{a_{11}a_{22} - a_{12}a_{21}} \begin{bmatrix} a_{22} & -a_{12} \\ -a_{21} & a_{11} \end{bmatrix}$$

Atau biar lebih mudahnya andaikan:

$$[A] = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \rightarrow \text{maka:}$$

$$[A]^{-1} = \frac{1}{ad - bc} \begin{bmatrix} d & -b \\ -c & c \end{bmatrix}$$

Transposisi sebuah matriks: pemindahan baris ke dalam kolom serta kolom ke dalam baris.

$$[A] = \begin{bmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots \\ a_{m1} & a_{m2} & a_{m3} & \dots & a_{mn} \end{bmatrix}$$

Transposisi ditandai oleh $[A]^T$:

$$[A]^T = \begin{bmatrix} a_{11} & a_{21} & a_{31} & \dots & a_{m1} \\ a_{12} & a_{22} & a_{32} & \dots & a_{m2} \\ \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots \\ a_{1n} & a_{2n} & a_{3n} & \dots & a_{mn} \end{bmatrix}$$

Irfan Subakti - 司馬伊凡

Demikian juga:

$$[C] = \begin{bmatrix} c_{11} \\ c_{21} \\ c_{31} \\ c_{41} \end{bmatrix}$$

$$[C]^T = \begin{bmatrix} c_{11} & c_{21} & c_{31} & c_{41} \end{bmatrix}$$

Matriks diperluas (augmented) dengan menambahkan satu (atau beberapa) kolom pada matriks semula.

$$[A] = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$$

Diperluas menjadi berdimensi 3 x 6:

$$\left[\begin{array}{ccc|ccc} a_{11} & a_{12} & a_{13} & 1 & 0 & 0 \\ a_{21} & a_{22} & a_{23} & 0 & 1 & 0 \\ a_{31} & a_{32} & a_{33} & 0 & 0 & 1 \end{array}\right]$$

Menyatakan persamaan aljabar linier simultan dalam bentuk matriks

Persamaan aljabar linier bentuk umumnya:

$$\begin{array}{ccccccccc} a_{11}x_1 & + & a_{12}x_2 & + & \ldots & + & a_{1n}x_n & = & c_1 \\ a_{21}x_1 & + & a_{22}x_2 & + & \ldots & + & a_{2n}x_n & = & c_2 \\ . & & . & & & & . & & . \\ . & & . & & & & . & & . \\ . & & . & & & & . & & . \\ a_{n1}x_1 & + & a_{n2}x_2 & + & \ldots & + & a_{nn}x_n & = & c_n \end{array} \qquad [6.1]$$

Dimana a = koefisien konstanta, c = konstanta, dan n = jumlah persamaan.

Persamaan di atas dapat dinyatakan dalam bentuk matriks sebagai:

$$[A]\,[X] = [C]$$

Dimana [A] adalah matriks bujur sangkar n x n dari koefisien:

$$[A] = \begin{bmatrix} a_{11} & a_{12} & \ldots & a_{1n} \\ a_{21} & a_{22} & \ldots & a_{2n} \\ \ldots & \ldots & \ldots & \ldots \\ a_{n1} & a_{n2} & \ldots & a_{nn} \end{bmatrix}$$

Irfan Subakti - 司馬伊凡

[C] adalah vektor kolom n x 1 dari konstanta-konstanta:

$$[C]^T = [c_1 \quad c_2 \quad c_3 \quad \ldots \quad c_n]$$

Dan [X] adalah vektor kolom n x 1 dari yang tak diketahui:

$$[X]^T = [x_1 \quad x_2 \quad x_3 \quad \ldots \quad x_n]$$

Maka:

$$[A]^{-1}[A]\,[X] = [A]^{-1}[C]$$

Karena $[A]^{-1}[A]$ sama dengan [I] maka persamaan menjadi:

$$[X] = [A]^{-1}[C]$$

Sehingga persamaan dapat diselesaikan untuk [X]. Inilah contoh bagaimana inversi memainkan peranan yang serupa terhadap pembagian dalam aljabar matriks.

Lalu seringkali dijumpai kegunaan untuk memperluas [A] dengan [C]. Misal bila n = 3 dalam matriks berdimensi 3 x 4:

$$\left[\begin{array}{ccc|c} a_{11} & a_{21} & a_{13} & c_1 \\ a_{21} & a_{22} & a_{23} & c_2 \\ a_{31} & a_{32} & a_{33} & c_3 \end{array}\right]$$

Menyatakan persamaan dalam bentuk ini mempunyai banyak manfaat, karena beberapa teknik untuk menyelesaikan sistem persamaan linier melakukan operasi yang identik terhadap sebuah baris dari koefisien-koefisien, serta konstanta di ruas kanan yang bersesuaian.

Irfan Subakti - 司馬依元

6.2 Penyelesaian Persamaan yang Sedikit Jumlahnya

Untuk jumlah persamaan yang kecil ($n \leq 3$) cara penyelesaiannya tak memerlukan komputer:

1. Metode Grafik
2. Aturan Cramer
3. Eliminasi yang tak diketahui

6.2.1 Metode Grafik

Solusi grafik dapat diperoleh untuk 2 persamaan dengan menggambarkannya pada koordinat Cartesian (Cartesius) dengan satu sumbu x_1 (absis) dan lainnya x_2 (ordinat).

Karena yang ditangani adalah sistem linier, maka setiap persamaan merupakan garis lurus.

Ini dapat mudah digambarkan untuk persamaan umum:

$$a_{11}x_1 + a_{12}x_2 = c_1$$

$$a_{21}x_1 + a_{22}x_2 = c_2$$

Kedua persamaan dapat diselesaikan untuk x_2:

$$x_2 = -\left(\frac{a_{11}}{a_{12}}\right) x_1 + \frac{c_1}{a_{12}}$$

$$x_2 = -\left(\frac{a_{21}}{a_{22}}\right) x_1 + \frac{c_2}{a_{22}}$$

Jadi kini persamaan dalam bentuk garis lurus; artinya:

x_2 = kemiringan x_1 + perpotongan dengan absis

Harga x_1 dan x_2 pada perpotongan garis-garis menyatakan solusinya.

Irfan Subakti 同馬伊凡

Contoh 6.1 Metode Grafik untuk 2 persamaan

Masalah: gunakan metode Grafik untuk menyelesaikan:

$$3x_1 + 2x_2 = 18 \qquad [C6.1.1]$$

$$-x_1 + 2x_2 = 2 \qquad [C6.1.2]$$

Solusi: misalkan x_1 jadi absis. Selesaikan persamaan [C6.1.1] untuk x2:

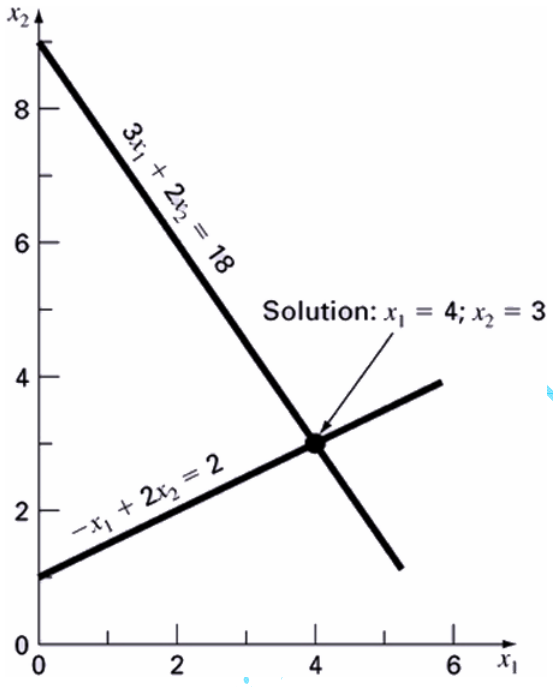
$x_2 = (-\frac{3}{2})\, x_1 + 9$ → yang kalau digambarkan merupakan garis lurus dengan perpotongan di angka 9 pada absis, dengan kemiringan = $-\frac{3}{2}$

Persamaan [C6.1.2] juga dapat diselesaikan untuk x_2:

$x_2 = (\frac{1}{2})\, x_1 + 1$ → yang kalau digambarkan merupakan garis lurus dengan

perpotongan di angka 1 pada absis, dengan kemiringan = $\frac{1}{2}$

Dari gambar grafik yang telah dibuat, solusi didapat dari perpotongan 2 garis. Yaitu pada $x_1 = 4$ dan $x_2 = 3$, seperti gambar 6.1 di bawah ini. Perpotongan garis-garis menunjukkan solusinya.



Gambar 6.1 Solusi grafik dari kumpulan 2 persamaan aljabar linier simultan

Irfan Subakti

Hasil ini dapat dicek dengan memasukkan harga-harga itu ke dalam persamaan semula:

$3(4) + 2(3) = 18$

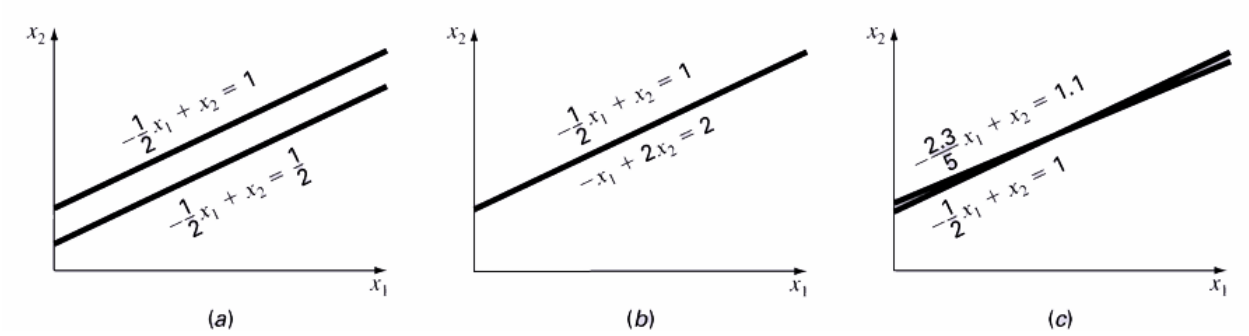
$-4 + 2(3) = 2$

Di atas adalah contoh untuk 2 persamaan → dinyatakan dalam suatu bidang dalam sistem koordinat 2 dimensi.

Sedangkan untuk 3 persamaan → dinyatakan dalam suatu ruang dalam sistem koordinat 3 dimensi.

Lebih dari 3 persamaan → tak bisa dinyatakan.

Lebih jauh, karena mengandalkan pengamatan secara visual, maka metode Grafik juga dihadapkan pada kondisi timpang (**ill-conditioned**) seperti digambarkan pada gambar 6.2 di bawah ini. Terlihat bahwa pada (a) tak ada solusi, (b) solusi tak terhingga, dan (c) kemiringan yang begitu dekat membuatnya sulit secara visual mendeteksi titik perpotongannya. (a) dan (b) disebut sistem Singular, (c) disebut sistem hampir Singular.



Gambar 6.2 Penjelasan grafik dari sistem kondisi timpang (ill-conditioned)

6.2.2 Determinan dan Aturan Cramer

Determinan

Determinan dapat dijelaskan untuk sekumpulan 3 persamaan.

$$a_{11}x_1 + a_{12}x_2 + a_{13}x_3 = c_1$$

$$a_{21}x_1 + a_{22}x_2 + a_{23}x_3 = c_2$$

$$a_{31}x_1 + a_{32}x_2 + a_{33}x_3 = c_3$$

Atau dalam bentuk matriks:

$$[A]\,[X] = [C]$$

Dimana [A] adalah matriks koefisien:

$$[A] = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$$

Determinan D dari sistem ini dibentuk dari koefisien-koefisien persamaan, seperti:

$$D = \begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix} \rightarrow \text{perhatikan tak menggunakan [} \cdots \text{] tapi | } \cdots \text{ |} \qquad [6.2]$$

Walau determinan D dan matriks koefisien [A] terdiri dari elemen-elemen yang sama, mereka adalah konsep matematika yang sangat berbeda. Maka dibedakan secara visual oleh tanda kurung yang berbeda. Berlainan dengan matriks, determinan adalah suatu bilangan tunggal.

Misal harga orde kedua determinan:

Irfan Subakti - 司馬伊凡

$$D = \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix}$$

dihitung dengan:

$$D = a_{11}a_{22} - a_{12}a_{21} \qquad [6.3]$$

Atau untuk lebih mudahnya:

$$D = \begin{vmatrix} a & b \\ c & d \end{vmatrix}$$ dihitung dengan D = ad - bc

Untuk orde ketiga (persamaan [6.2]), satu harga tunggal determinan dapat dihitung:

$$D = a_{11} \begin{vmatrix} a_{22} & a_{23} \\ a_{32} & a_{33} \end{vmatrix} - a_{12} \begin{vmatrix} a_{21} & a_{23} \\ a_{31} & a_{33} \end{vmatrix} + a_{13} \begin{vmatrix} a_{21} & a_{22} \\ a_{31} & a_{32} \end{vmatrix} \qquad [6.4]$$

atau untuk lebih mudahnya:

$$D = \begin{vmatrix} a & b & c \\ d & e & f \\ g & h & i \end{vmatrix}$$

$$D = a \begin{vmatrix} e & f \\ h & i \end{vmatrix} - b \begin{vmatrix} d & f \\ g & i \end{vmatrix} + c \begin{vmatrix} d & e \\ g & h \end{vmatrix}$$

Dimana determinan 2 x 2 disebut dengan **minor**.

Irfan Subakti - 司馬伊凡

Contoh 6.2 Determinan

Masalah: hitunglah harga-harga determinan dari sistem yang ditunjukkan oleh gambar 6.1 dan 6.2 di atas.

Solusi

Untuk gambar 6.1:

$$D = \begin{vmatrix} 3 & 2 \\ -1 & 2 \end{vmatrix} = 3(2) - 2(-1) = 8$$

Untuk gambar 6.2a:

$$D = \begin{vmatrix} -\frac{1}{2} & 1 \\ -\frac{1}{2} & 1 \end{vmatrix} = -\frac{1}{2}(1) - 1(-\frac{1}{2}) = 0$$

Untuk gambar 6.2b:

$$D = \begin{vmatrix} -\frac{1}{2} & 1 \\ -1 & 2 \end{vmatrix} = -\frac{1}{2}(2) - 1(-1) = 0$$

Untuk gambar 6.2c:

$$D = \begin{vmatrix} -\frac{1}{2} & 1 \\ -\frac{2,3}{5} & 1 \end{vmatrix} = -(1) - 1(-\frac{2,3}{5}) = -0,04$$

Dalam contoh sebelumnya, sistem **Singular** mempunyai determinan = 0, sedang sistem yang hampir Singular mempunyai determinan yang mendekati 0.

Aturan Cramer

Setiap yang tak diketahui dalam sebuah sistem persamaan aljabar linier boleh dinyatakan sebagai sebuah fraksi dari 2 determinan, penyebut D dan pembilang yang diperoleh dari D, dengan mengganti kolom dari koefisien-koefisien yang tak diketahui yang ditanyakan oleh konstanta-konstanta c_1, c_2, … , c_n.

Misalnya x_1 dapat dihitung sebagai:

$$x_1 = \frac{\begin{vmatrix} c_1 & a_{12} & a_{13} \\ c_2 & a_{22} & a_{23} \\ c_3 & a_{32} & a_{33} \end{vmatrix}}{D} \qquad [6.5]$$

Irfan Subakti - 司馬伊凡

Contoh 6.3 Aturan Cramer

Masalah: gunakan aturan Cramer untuk menyelesaikan:

$$0,3x_1 + 0,52x_2 + x_3 = 0,01$$

$$0,5x_1 + x_2 + 1,9x_3 = 0,67$$

$$0,1x_1 + 0,3x_2 + 0,5x_3 = -0,44$$

Solusi: Determinan D dapat ditulis sebagai (persamaan [6.2]):

$$D = \begin{vmatrix} 0,3 & 0,52 & 1 \\ 0,5 & 1 & 1,9 \\ 0,1 & 0,3 & 0,5 \end{vmatrix}$$

Minor-minornya adalah:

$$A_1 = \begin{vmatrix} 1 & 1{,}9 \\ 0{,}3 & 0{,}5 \end{vmatrix} = 1(0{,}5) - 1{,}9(0{,}3) = -0{,}07$$

$$A_2 = \begin{vmatrix} 0{,}5 & 1{,}9 \\ 0{,}1 & 0{,}5 \end{vmatrix} = 0{,}5(0{,}5) - 1{,}9(0{,}3) = 0{,}06$$

$$A_3 = \begin{vmatrix} 0{,}5 & 1 \\ 0{,}1 & 0{,}3 \end{vmatrix} = 0{,}5(0{,}3) - 1(0{,}1) = 0{,}05$$

Yang dapat digunakan untuk mengevaluasi determinan, seperti dalam persamaan [6.4]:

$$D = 0{,}3(-0{,}07) - 0{,}52(0{,}06) + 1(0{,}05) = -0{,}0022$$

Dengan menerapkan persamaan [6.5], solusinya adalah:

$$x_1 = \frac{\begin{vmatrix} -0{,}01 & 0{,}52 & 1 \\ 0{,}67 & 1 & 1{,}9 \\ -0{,}44 & 0{,}3 & 0{,}5 \end{vmatrix}}{-0{,}0022} = \frac{0{,}03278}{-0{,}0022} = -14{,}9$$

$$x_2 = \frac{\begin{vmatrix} 0{,}3 & -0{,}01 & 1 \\ 0{,}5 & 0{,}67 & 1{,}9 \\ 0{,}1 & -0{,}44 & 0{,}5 \end{vmatrix}}{-0{,}0022} = \frac{0{,}0649}{-0{,}0022} = -29{,}5$$

$$x_3 = \frac{\begin{vmatrix} 0{,}3 & 0{,}52 & -0{,}01 \\ 0{,}5 & 1 & 0{,}67 \\ 0{,}1 & 0{,}3 & -0{,}44 \end{vmatrix}}{-0{,}0022} = \frac{-0{,}04356}{-0{,}0022} = 19{,}8$$

Persamaan yang lebih dari 3, aturan Cramer tidak praktis diterapkan. Ini disebabkan kalau jumlah persamaan bertambah, determinan akan menghabiskan waktu jika dievaluasikan dengan tangan.

Irfan Subakti - 司馬伊凡

Irfan Subakti - 司馬伊凡

DAFTAR PUSTAKA

[Cha91] S.C. Chapra dan R.P. Canale, Metode Numerik Untuk Teknik: Dengan Penerapan pada Komputer Pribadi, penerjemah: S. Sardy dan pendamping: Lamyarni I.S., Cetakan 1, Universitas Indonesia (UI-Press), Jakarta, 1991, ISBN: 979-456-071-5.

[Con87] A. Constantinides, Applied Numerical Methods with Personal Computers, International Edition, McGraw-Hill, Inc., Singapore, 1987, ISBN: 0-07-100168-9.

[Dix03] A.J. Dix, J.E. Finlay, G.D. Abowd and R. Beale, Human-Computer Interaction, Third Edition, Prentice Hall, USA, 2003.

[Naka93] S. Nakamura, Applied Numerical Methods in C, International Edition, Prentic-Hall, Inc., USA, 1993, ISBN: 0-13-034349-8.

[Ral78] A. Ralston, and P. Rabinowitz, A First Course in Numerical Analysis, 2d ed., McGraw-Hill, New York, 1978.

Irfan Subakti - 司馬伊凡