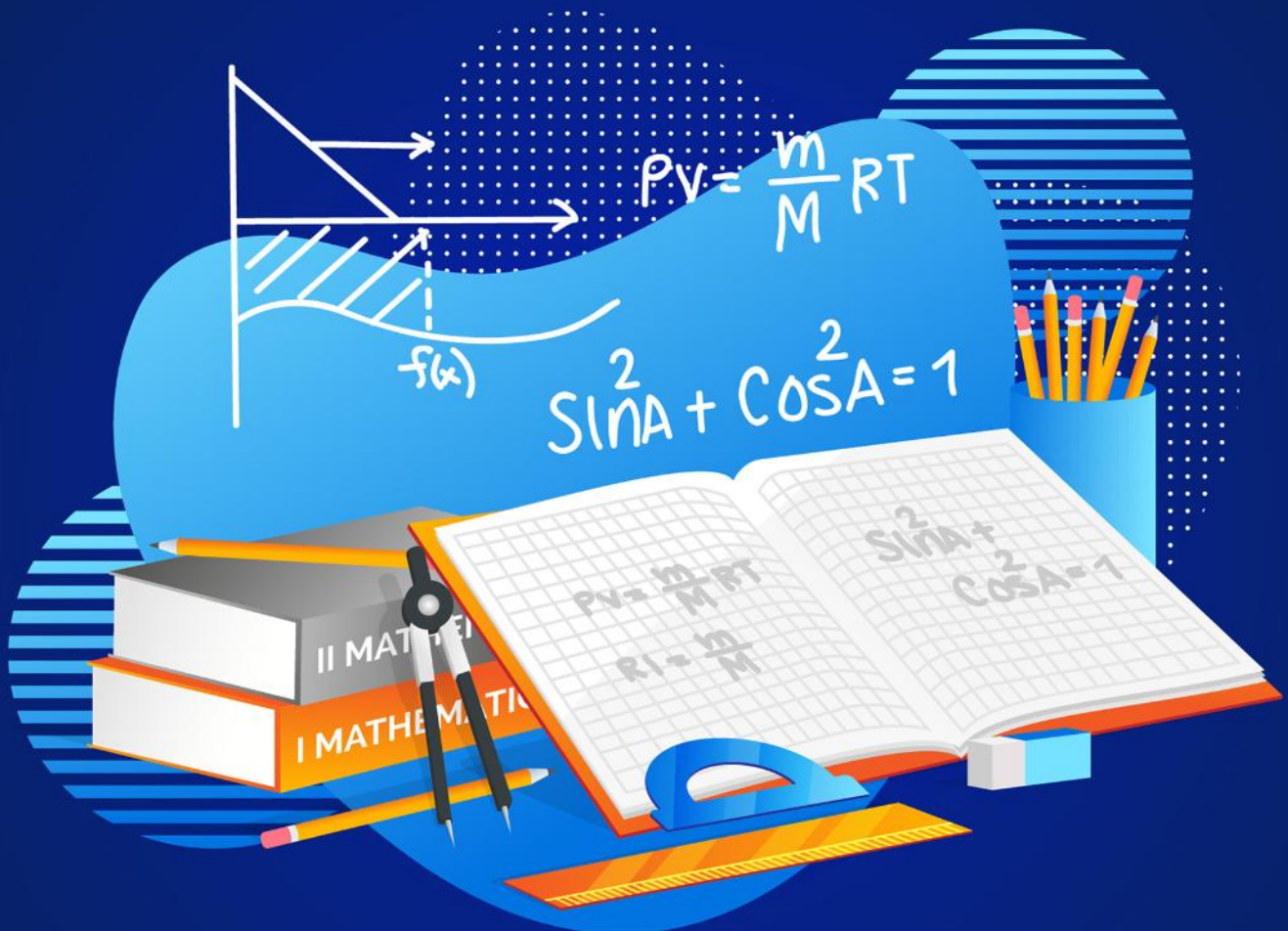




METODE NUMERIK

Drs. Ropinus Sidabutar, M.Pd.



METODE NUMERIK

Drs. Ropinus Sidabutar, M.Pd.



METODE NUMERIK

Penulis:
Drs. Ropinus Sidabutar, M.Pd.

Editor:
Maria Barus, S.Pd., M.Pd.

Layouter :
Tim Kreatif PRCI

Cover:
Rusli

Cetakan Pertama : Januari 2024

Hak Cipta 2024, pada Penulis. Diterbitkan pertama kali oleh:

Perkumpulan Rumah Cemerlang Indonesia
ANGGOTA IKAPI JAWA BARAT
Pondok Karisma Residence Jalan Rafflesia VI D.151
Panglayungan, Cipedes Tasikmalaya – 085223186009

Website : www.rcipress.rcipublisher.org
E-mail : rumahcemerlangindonesia@gmail.com

Copyright © 2024 by Perkumpulan Rumah Cemerlang Indonesia
All Right Reserved

- Cet. I - : Perkumpulan Rumah Cemerlang Indonesia, 2024
; 18 x 25cm
ISBN : 978-623-448-805-0

Hak cipta dilindungi undang-undang
Dilarang memperbanyak buku ini dalam bentuk dan dengan
cara apapun tanpa izin tertulis dari penulis dan penerbit

Undang-undang No.19 Tahun 2002 Tentang
Hak Cipta Pasal 72

Undang-undang No.19 Tahun 2002 Tentang Hak Cipta Pasal 72

Barang siapa dengan sengaja melanggar dan tanpa hak melakukan perbuatan sebagaimana dimaksud dalam pasal ayat (1) atau pasal 49 ayat (1) dan ayat (2) dipidana dengan pidana penjara masing-masing paling sedikit 1 (satu) bulan dan/atau denda paling sedikit Rp.1.000.000,00 (satu juta rupiah), atau pidana penjara paling lama 7 (tujuh) tahun dan/atau denda paling banyak Rp.5.000.000.000,00 (lima miliar rupiah).

Barang siapa dengan sengaja menyiarkan, memamerkan, mengedarkan, atau menjual kepada umum suatu ciptaan atau barang hasil pelanggaran hak cipta terkait sebagai dimaksud pada ayat (1) dipidana dengan pidana penjara paling lama 5 (lima) tahun dan/atau denda paling banyak Rp.500.000.000,00 (lima ratus juta rupiah)

KATA PENGANTAR

Selamat datang dalam lembar-lembar petualangan ilmiah ini! Buku "Panduan Praktis Metode Numerik" hadir untuk membimbing pembaca melalui dunia fasciasi dan keahlian dalam pemecahan masalah teknik menggunakan metode numerik. Sebagai sebuah karya yang disusun dengan penuh dedikasi, buku ini bertujuan untuk menyuguhkan pemahaman yang mendalam tentang konsep-konsep metode numerik dan menerapkannya secara praktis dalam pemecahan masalah dunia nyata.

Metode numerik menjadi landasan utama dalam menyelesaikan berbagai permasalahan matematika yang melibatkan perhitungan angka secara aproksimatif. Dalam buku ini, pembaca akan dibawa melangkah dari dasar-dasar konsep metode numerik hingga penerapannya dalam berbagai kasus nyata di bidang teknik. Setiap materi disajikan dengan bahasa yang jelas dan ringan, dilengkapi dengan contoh-contoh aplikatif yang memperkaya pemahaman.

Penggunaan metode numerik dalam pemecahan masalah teknik tidak hanya sekadar suatu keterampilan, tetapi juga suatu kebutuhan yang mendesak dalam era teknologi modern. Pengetahuan yang diperoleh dari buku ini diharapkan dapat membekali pembaca dengan kemampuan untuk menghadapi tantangan kompleks di dunia teknik, mulai dari simulasi permodelan, analisis struktural, hingga optimisasi desain.

Saya berharap buku ini menjadi teman setia bagi pembaca dalam memahami dan menguasai metode numerik, membuka jalan menuju penguasaan konsep-konsep yang mendasar serta memberikan wawasan yang luas tentang penerapannya dalam penyelesaian masalah teknik. Semoga buku ini memberikan kontribusi positif dalam pengembangan pengetahuan dan keahlian pembaca di dunia metode numerik. Selamat membaca!

Januari 2024, Penulis

DAFTAR ISI

KATA PENGANTAR	I
DAFTAR ISI	II
BAB I PENGANTAR METODE NUMERIK	1
A. Defenisi Metode Numerik	1
B. Sejarah dan Perkembangan Metode Numerik	2
C. Peran Metode Numerik dan Ilmu Komputasi	2
D. Aplikasi Metode Numerik dalam Berbagai Disiplin Ilmu	3
 BAB II REPRESENTASI ANGKA DAN KESALAHAN NUMERIK	 7
A. Sistem Bilangan dan Representasi Komputer	7
B. Kesalahan Pembulatan dan Representasi bilangan <i>Floating-Point</i>	9
C. Analisis Kesalahan dalam Perhitungan Numerik	12
 BAB III SOLUSI DAN PERSAMAAN NONLINEAR	 18
A. Metode Pencarian Akar	18
B. Metode Bagi Dua (<i>Bisection</i>)	21
C. Metode Newton-Raphson	24
D. Analisis Konvergensi dan Ketidakpastian	27
 BAB IV INTERPOLASI DAN APROKSIMASI	 31
A. Metode Interpolasi <i>Lagrange</i>	31
B. Metode Interpolasi Newton	33
C. Aproksimasi dengan Metode Kuadrat Terkecil (<i>Least Squares</i>)	35
 BAB V PENDAHULUAN PERSAMAAN DIFERENSIAL	 37
A. Pengertian dan Jenis Persamaan Diferensial	37
B. Kondisi Awal dan Batas dalam Penyelesaian Persamaan Diferensial	39
 BAB VI METODE EULER DAN METODE RUNGE-KUTTA	 42
A. Metode Euler untuk Persamaan Diferensial Biasa(ODE)	42
B. Metode Runge-Kutta Orde Tinggi untuk ODE	43
C. Implementasi Komputasional dan Stabilitas	45

BAB VII PERSAMAAN DAN DIFERENSIAL PARSIAL	49
A. Pengantar Persamaan Diferensial Parsial PDE	49
B. Metode Beda Hingga untuk PDE Eliptik, Parabolik, dan Hiperbolik	51
C. Metode Godunov: Efektif untuk persamaan hiperbolik nonlinier.	53
D. Iterasi Sederhana dan Konvergensi	56
E. Metode Jacobi -Seidel untuk Sistem Persamaan Linear	58
F. Metode Relaksasi dan Aplikasinya	61
 BAB IX PEMODELAN NUMERIK DAN SIMULASI	 64
A. Pemahaman konsep Pemodelan Numerik	64
B. Implementasi Metode Numerik dalam Pemodelan Matematika	67
C. Simulasi dan Studi kasus Nyata	70
 BAB X TANTANGAN DAN ARAH PENGEMBANGAN	 76
A. Tantangan Terkini dalam Pengembangan Metode Numerik	76
B. Integrasi Metode Numerik dengan Teknologi Baru	82
C. Proyeksi Masa Depan dan Kontribusi Potensial	85
LAMPIRAN	89
DAFTAR PUSTAKA	90

BAB I

PENGANTAR METODE NUMERIK

A. Defenisi Metode Numerik

Metode numerik merupakan satu metode yang digunakan dalam menyelesaikan masalah-masalah matematika yang terjadi dalam keseharian dengan menggunakan sekumpulan aritmatika sederhana dan metode logika pada bilangan atau data numerik yang diberikan. Metode Numerik juga metode yang mengacu pada konsep dan teknik-teknik yang digunakan dalam bidang matematika untuk menyelesaikan permasalahan secara numerik atau menggunakan angka-angka. Metode Numerik melibatkan pendekatan komputasional dalam menangani masalah-masalah matematis yang sulit atau kompleks dan solusi analitik tidak selalu memungkinkan atau efisien. Metode komputasi sering juga disebut dengan algoritma. Proses penyelesaian dilakukan dengan puluhan bahkan sampai jutaan operasi tergantung pada kompleksitas masalah yang harus diselesaikan. Penyelesaian ini tidak keluar dari dasar pemikiran analitis namun juga adanya pertimbangan dalam pemakaian metode numerik. Atas dasar algoritma yang dikembangkan dalam metode numerik maka perhitungan dilakukan secara berulang-ulang untuk mendapatkan hasil yang semakin mendekati nilai yang sebenarnya.

Diharapkan dengan metode pendekatan ini, setiap nilai hasil perhitungan akan mempunyai galat (*error*) atau nilai kesalahan. Kesalahan ini penting karena kesalahan dalam pemakaian algoritma menyebabkan nilai kesalahan yang besar tidak diharapkan. Pendekatan metode numerik membahas Tingkat kesalahan dan Tingkat kecepatan proses yang terjadi. Masalah matematik yang terjadi sejatinya dengan menggunakan metode analitik yang memberikan Solusi sejati karena

memiliki galat yang bernilai nol. Metode analitik dibatasi pada permasalahan tertentu saja. Apabila metode analitik tidak dapat diterapkan lagi maka solusinya dapat dicari dengan menggunakan metode numerik. Pada metode numerik solusinya merupakan pendekatan terhadap solusi sejati.

B. Sejarah dan Perkembangan Metode Numerik

Perkembangan Metode Numerik melibatkan evolusi dan progres dalam penerapan konsep dan teknik matematika untuk menyelesaikan permasalahan secara numerik. Sejarahinya mencakup berbagai fase di mana metode-metode ini berkembang dan disesuaikan dengan kebutuhan pemecahan masalah yang semakin kompleks. Dari perkembangan awal hingga zaman modern, Sejarah Metode Numerik mencerminkan inovasi dan penemuan yang terus-menerus dalam upaya meningkatkan efisiensi dan akurasi penyelesaian masalah matematis dengan pendekatan komputasional.

C. Peran Metode Numerik dan Ilmu Komputasi

Metode numerik memiliki peran krusial dalam ilmu komputasi dengan menyediakan pendekatan komputasional untuk menangani masalah matematis yang kompleks. Beberapa peran utamanya melibatkan:

1. Simulasi dan Pemodelan: Metode numerik digunakan untuk mensimulasikan fenomena alam atau perilaku sistem yang sulit dijelaskan secara analitis. Ini dapat mencakup simulasi dinamika fluida, model ekosistem, atau perubahan iklim.
2. Pemecahan Persamaan Diferensial: Metode numerik memungkinkan pendekatan komputasional untuk memecahkan

persamaan diferensial, yang memiliki banyak aplikasi dalam ilmu fisika, teknik, biologi, dan lainnya. Ini termasuk solusi numerik untuk persamaan diferensial parsial yang kompleks.

3. Analisis Data Numerik: Metode numerik digunakan untuk menganalisis data yang diperoleh dari eksperimen atau pengukuran, termasuk interpolasi data, regresi numerik, dan metode statistik yang melibatkan perhitungan matematis.
4. Optimisasi Numerik: Dalam ilmu komputasi, metode numerik digunakan untuk menemukan solusi optimal dalam berbagai konteks, seperti optimisasi fungsi dalam pembelajaran mesin, perencanaan rantai pasokan, atau desain struktur.
5. Pemrosesan Gambar dan Pengolahan Sinyal: Metode numerik digunakan untuk mengolah gambar dan sinyal dalam bidang seperti penglihatan komputer, pengenalan pola, dan komunikasi data.

Secara keseluruhan, metode numerik memungkinkan komputasi untuk memecahkan masalah matematis yang mendasar, memungkinkan pemodelan dan analisis yang lebih baik dalam berbagai disiplin ilmu komputasi.

D. Aplikasi Metode Numerik dalam Berbagai Disiplin Ilmu

Aplikasi Metode Numerik dalam Berbagai disiplin Ilmu: Metode numerik digunakan luas dalam berbagai disiplin ilmu, seperti:

1. Fisika: Simulasi dinamika fluida, analisis tegangan-strain, dan penyelesaian persamaan diferensial parsial.
2. Matematika: Penyelesaian persamaan nonlinier, interpolasi, dan integrasi numerik.

3. Rekayasa: Analisis elemen hingga, optimasi desain, dan penyelesaian persamaan diferensial.
4. Ekonomi dan Keuangan: Evaluasi opsi finansial, analisis risiko, dan penyelesaian model matematis terkait keuangan.
5. Sains Komputer: Metode numerik digunakan dalam algoritma optimisasi, analisis kompleksitas algoritma, dan simulasi sistem.
6. Biologi: Analisis populasi, model ekologi, dan pemodelan reaksi biokimia menggunakan metode numerik.
7. Kimia Simulasi reaksi kimia, penyelesaian persamaan diferensial kimia, dan analisis struktur molekuler.
8. Ilmu Geofisika: Pemodelan pergerakan lempeng tektonik, analisis data seismik, dan perhitungan medan gravitasi.
9. Kesehatan Analisis data medis, pemodelan penyebaran penyakit, dan simulasi proses biomedis.
10. Ilmu Sosial: Pemodelan dinamika populasi, analisis data sosial, dan prediksi tren.

Penerapan metode numerik dalam berbagai disiplin ilmu melibatkan penggunaan teknik-teknik matematika dan komputasi untuk memecahkan masalah dunia nyata yang seringkali kompleks dan sulit atau bahkan tidak dapat dipecahkan secara analitis. Berikut adalah penjelasan lebih luas tentang aplikasi metode numerik dalam beberapa disiplin ilmu:

1. Fisika dan Rekayasa:
 - a. Simulasi Dinamika Fluida: Metode numerik seperti Finite Element Method (FEM) atau Finite Volume Method (FVM) digunakan untuk mensimulasikan aliran fluida di sekitar objek atau dalam suatu sistem.

- b. **Analisis Struktur:** Metode elemen hingga digunakan untuk menganalisis distribusi tegangan dan deformasi pada struktur rekayasa, seperti jembatan, bangunan, atau kendaraan.
 - c. **Proses Pengolahan Sinyal:** Metode Fourier dan Transformasi Laplace digunakan dalam pemrosesan sinyal dan analisis sistem dinamis.
2. **Matematika dan Statistika:**
- a. **Penyelesaian Persamaan Diferensial:** Metode numerik banyak digunakan untuk menemukan solusi numerik dari persamaan diferensial parsial dan biasa, yang muncul dalam model matematika untuk berbagai fenomena alam.
 - b. **Analisis Data:** Metode statistik numerik seperti regresi linier, clustering, dan analisis komponen utama digunakan untuk menganalisis dan merumuskan pola dari kumpulan data.
3. **Keuangan:**
- a. **Penilaian Opsi dan Derivatif:** Metode Monte Carlo dan metode Black-Scholes digunakan dalam penilaian opsi dan manajemen risiko keuangan.
 - b. **Model Matematika Keuangan:** Persamaan diferensial parsial dan metode numerik lainnya digunakan untuk memodelkan perubahan harga aset keuangan dan perilaku pasar.
4. **Biologi dan Kedokteran:**
- a. **Simulasi Model Biologis:** Metode numerik digunakan untuk mensimulasikan model-model matematika yang mewakili dinamika populasi, penyebaran penyakit, dan evolusi genetika.
 - b. **Analisis Citra Medis:** Metode pemrosesan citra numerik digunakan untuk menganalisis dan memproses citra medis, seperti MRI, CT scan, dan gambar mikroskopis.

5. Ilmu Komputer:

- a. **Algoritma dan Kecerdasan Buatan:** Metode numerik digunakan dalam pengembangan algoritma, pembelajaran mesin, dan kecerdasan buatan untuk menyelesaikan berbagai masalah komputasi.
- b. **Analisis Kinerja Jaringan:** Metode numerik digunakan untuk menganalisis kinerja jaringan komputer, routing, dan optimisasi lalu lintas data.

6. Ilmu Pengetahuan Alam:

- a. **Simulasi Fenomena Alam:** Metode numerik digunakan untuk mensimulasikan dan memahami fenomena fisika seperti perubahan iklim, gempa bumi, dan dinamika sistem planet.
- b. **Analisis Data Eksperimental:** Metode statistik numerik digunakan untuk menganalisis dan menafsirkan data eksperimental dalam fisika dan kimia.

7. Teknologi Informasi:

- a. **Pemecahan Masalah Keamanan:** Metode kriptografi numerik digunakan dalam pemecahan masalah keamanan informasi dan pengamanan data.
- b. **Analisis Kinerja Sistem:** Metode numerik digunakan untuk menganalisis kinerja sistem komputer, database, dan jaringan.

Penerapan metode numerik membantu ilmuwan, insinyur, dan profesional dalam memahami, merancang, dan memecahkan masalah yang kompleks di berbagai bidang ilmu. Dengan menggunakan pendekatan ini, mereka dapat menghasilkan solusi numerik yang mendekati solusi analitis dan memberikan wawasan yang lebih baik dalam pemahaman fenomena di dunia nyata.

BAB II

REPRESENTASI ANGKA DAN KESALAHAN NUMERIK

A. Sistem Bilangan dan Representasi Komputer

Sistem bilangan adalah cara mengungkapkan nilai numerik, seperti desimal, biner, oktal, dan heksadesimal. Representasi komputer menggunakan biner untuk menyimpan dan memproses informasi. Misalnya, desimal 10 dapat direpresentasikan sebagai biner 1010 dalam konteks komputer.

Sistem bilangan adalah suatu cara atau aturan untuk mewakili nilai-nilai dalam bentuk angka. Dalam konteks komputer, dua sistem bilangan yang paling umum digunakan adalah sistem bilangan biner (base-2) dan sistem bilangan desimal (base-10). Berikut adalah penjelasan lebih luas tentang beberapa sistem bilangan:

1. **Bilangan Desimal (Base-10):**
 - a. **Representasi Angka:** Sistem bilangan desimal menggunakan angka 0 hingga 9. Setiap digit mewakili nilai dari 0 hingga 9, dan setiap posisi digit memiliki bobot yang merupakan pangkat dari 10.
 - b. **Contoh:** Angka 365 di desimal berarti $(3 * 10^2) + (6 * 10^1) + (5 * 10^0)$.
2. **Bilangan Biner (Base-2):**
 - a. **Representasi Angka:** Sistem bilangan biner hanya menggunakan dua digit, yaitu 0 dan 1. Setiap digit mewakili pangkat dari 2.
 - b. **Contoh:** Angka 101 dalam biner berarti $(1 * 2^2) + (0 * 2^1) + (1 * 2^0) = 5$ dalam desimal.

3. Bilangan Heksadesimal (Base-16):

- a. **Representasi Angka:** Sistem bilangan heksadesimal menggunakan angka 0-9 dan huruf A-F untuk mewakili nilai dari 0 hingga 15. Setiap digit mewakili pangkat dari 16.
- b. **Contoh:** Angka 1A dalam heksadesimal berarti $(1 * 16^1) + (10 * 16^0) = 26$ dalam desimal.

4. Bilangan Oktal (Base-8):

- a. **Representasi Angka:** Sistem bilangan oktal menggunakan angka 0-7. Setiap digit mewakili pangkat dari 8.
- b. **Contoh:** Angka 54 dalam oktal berarti $(5 * 8^1) + (4 * 8^0) = 44$ dalam desimal.

Representasi Komputer:

Representasi komputer adalah cara angka dan karakter direpresentasikan dalam bentuk biner, yang merupakan dasar bagi semua operasi komputasi dalam sistem komputer. Beberapa konsep utama dalam representasi komputer melibatkan pengkodean bilangan, karakter, dan instruksi.

1. Representasi Bilangan dalam Komputer:

- a. **Bit dan Byte:** Bit (Binary Digit) adalah unit terkecil dalam representasi komputer yang dapat bernilai 0 atau 1. Delapan bit membentuk satu byte. Sebagai contoh, bilangan 10110101 dalam biner direpresentasikan sebagai satu byte.
- b. **Representasi Angka Negatif:** Komputer menggunakan metode seperti Two's complement untuk merepresentasikan angka negatif.

2. Representasi Karakter dalam Komputer:

- a. **ASCII (American Standard Code for Information Interchange):** ASCII adalah suatu sistem pengkodean

karakter yang umum digunakan dalam komputer. Setiap karakter diberi kode numerik dalam rentang 0 hingga 127 (untuk ASCII 7-bit) atau 0 hingga 255 (untuk ASCII 8-bit/Extended ASCII).

3. Representasi Instruksi dan Data:

- a. Kode Mesin: Instruksi dan data dalam komputer direpresentasikan dalam bentuk kode mesin, yaitu serangkaian angka biner yang diinterpretasikan oleh unit pemrosesan komputer.
- b. Instruksi Assembly: Bahasa assembly memberikan representasi teks yang lebih mudah dimengerti oleh manusia, tetapi setiap instruksi masih diubah menjadi kode mesin saat eksekusi.

4. Floating-Point Representation:

- a. Representasi Titik Mengambang: Untuk merepresentasikan angka desimal yang sangat besar atau sangat kecil, komputer menggunakan sistem titik mengambang (floating-point). Ini terdiri dari mantissa (nilai sebenarnya) dan eksponen (pangkat).

Pemahaman yang baik tentang sistem bilangan dan representasi komputer sangat penting dalam pengembangan perangkat lunak, pemrograman, dan pemecahan masalah komputasi. Selain itu, memahami cara komputer merepresentasikan data membantu dalam mengoptimalkan efisiensi dan kinerja program.

B. Kesalahan Pembulatan dan Representasi bilangan *Floating-Point*

Kesalahan pembulatan dan representasi bilangan floating-point muncul karena keterbatasan dalam merepresentasikan nilai desimal

secara presisi pada komputer. Sistem floating-point menggunakan biner untuk merepresentasikan nilai desimal, dan beberapa nilai desimal tidak dapat diwakili secara akurat, menghasilkan kesalahan pembulatan.

Kesalahan Pembulatan dalam Representasi Bilangan Floating-

Point:

Sistem komputer menggunakan representasi titik mengambang untuk merepresentasikan angka real dalam bentuk biner. Walaupun metode ini memungkinkan representasi luas rentang nilai, namun seringkali menyebabkan kesalahan pembulatan. Kesalahan pembulatan adalah perbedaan antara nilai sebenarnya dan nilai yang dapat direpresentasikan oleh sistem. Berikut adalah beberapa jenis kesalahan pembulatan yang umum terjadi dalam representasi bilangan floating-point:

1. Kesalahan Pembulatan Absolut:

Kesalahan ini merupakan selisih absolut antara nilai sebenarnya dan nilai yang direpresentasikan dalam format floating-point. Dapat terjadi ketika suatu angka real tidak dapat direpresentasikan secara tepat dalam format biner.

2. Kesalahan Pembulatan Relatif:

Kesalahan ini mengukur sejauh mana kesalahan pembulatan relatif terhadap besaran nilai yang direpresentasikan. Kesalahan ini dapat diukur sebagai perbandingan antara kesalahan pembulatan dan nilai sebenarnya.

3. Kesalahan Pembulatan Dinamis:

Kesalahan ini dapat terjadi selama operasi aritmatika pada bilangan floating-point. Setiap operasi aritmatika dapat menambahkan kesalahan pembulatan ke nilai hasilnya.

4. Underflow dan Overflow:

Kesalahan ini terjadi ketika hasil operasi menghasilkan nilai yang lebih kecil dari nilai terkecil atau lebih besar dari nilai terbesar yang dapat direpresentasikan dalam format floating-point.

Representasi Bilangan Floating-Point:

Representasi bilangan floating-point menggunakan format yang terdiri dari tiga komponen utama: bit tanda, eksponen, dan mantissa. Kesalahan pembulatan dapat terjadi karena beberapa alasan:

1. Pembatasan Panjang Mantissa:

Representasi floating-point memiliki panjang mantissa terbatas. Oleh karena itu, angka dengan digit desimal yang lebih panjang dari yang dapat direpresentasikan akan di-truncate atau pembulatan, mengakibatkan kesalahan.

2. Pembatasan Panjang Eksponen:

Panjang eksponen dalam format floating-point juga dibatasi. Jika nilai eksponen terlalu besar atau terlalu kecil, kesalahan pembulatan dapat terjadi, dan nilai yang direpresentasikan mungkin menjadi tak akurat.

3. Pembulatan Hasil Operasi:

Setiap operasi aritmatika pada bilangan floating-point dapat memperkenalkan kesalahan pembulatan. Hasil operasi harus dibulatkan kembali ke format floating-point yang sesuai, dan ini dapat menghasilkan kesalahan.

4. Penanganan Nol dan Infinity:

Representasi floating-point memiliki konsep nol positif, nol negatif, dan infinity. Kesalahan pembulatan dapat terjadi ketika melakukan operasi yang melibatkan nol atau hasil yang sangat besar yang mendekati infinity.

Strategi Mengatasi Kesalahan Pembulatan:

1. Pemahaman Batas Kesalahan:

Pemahaman yang baik tentang batas kesalahan pembulatan dan representasi bilangan floating-point membantu pengembang mengantisipasi dan mengelola kesalahan dengan lebih baik.

2. Pemilihan Format Floating-Point yang Tepat:

Pemilihan format floating-point yang sesuai dengan kebutuhan aplikasi dapat membantu mengurangi dampak kesalahan pembulatan.

3. Penggunaan Algoritma Numerik yang Stabil:

Algoritma numerik yang dirancang dengan baik dapat membantu mengurangi kesalahan pembulatan selama operasi aritmatika.

4. Penggunaan Presisi Lebih Tinggi:

Penggunaan presisi floating-point yang lebih tinggi (misalnya, double precision) dapat membantu mengurangi kesalahan pembulatan, namun ini juga meningkatkan penggunaan memori dan waktu komputasi.

5. Pengaturan Standar dan Mode Pembulatan:

Pengaturan standar dan mode pembulatan pada lingkungan komputasi dapat mempengaruhi perilaku kesalahan pembulatan. Pengaturan ini harus dipahami dan disesuaikan sesuai kebutuhan.

Penting untuk diingat bahwa kesalahan pembulatan adalah suatu keniscayaan dalam representasi bilangan floating-point, dan pengembang perlu memahami dampaknya pada keakuratan hasil perhitungan.

C. Analisis Kesalahan dalam Perhitungan Numerik

Ada beberapa kesalahan umum dalam perhitungan numerik, termasuk:

1. Kesalahan Pembulatan (*Round-off Error*): Terjadi ketika angka desimal dibulatkan, menyebabkan ketidakakuratan dalam hasil perhitungan.
2. Kesalahan Truncation (*Truncation Error*): Muncul saat sebagian dari proses matematika diabaikan atau dipotong, menghasilkan ketidakakuratan.
3. Kesalahan Metode (*Method Error*): Terjadi karena pemilihan metode numerik yang tidak sesuai atau tidak optimal untuk permasalahan tertentu.
4. Kesalahan Aproksimasi (*Approximation Error*): Disebabkan oleh penggunaan nilai pendekatan atau aproksimasi, yang dapat mempengaruhi ketepatan hasil.
5. Kesalahan Iterasi (*Iterative Error*): Muncul dalam metode iteratif seperti metode Newton-Raphson, di mana kesalahan pada iterasi sebelumnya dapat berkumpul.
6. Kesalahan Model (*Modeling Error*): Terjadi jika model matematika yang digunakan untuk memodelkan fenomena nyata tidak sepenuhnya merepresentasikan kenyataan.
7. Kesalahan Input (*Input Error*): Dapat disebabkan oleh kesalahan manusia dalam memasukkan data atau parameter yang digunakan dalam perhitungan.
8. Kesalahan Pengambilan Sampel (*Sampling Error*): Terkait dengan kesalahan yang timbul dari proses pengambilan sampel atau diskritisasi data kontinu.
9. Kesalahan Komputasi (*Computational Error*): Terjadi akibat pembulatan, representasi terbatas, dan karakteristik lain dari komputer yang digunakan.

Analisis kesalahan dalam perhitungan numerik adalah proses mengevaluasi dan memahami kesalahan yang mungkin terjadi selama eksekusi algoritma atau perhitungan matematis pada komputer. Kesalahan ini dapat berasal dari berbagai sumber, termasuk pembulatan bilangan, representasi bilangan dalam komputer, dan metode numerik yang digunakan. Berikut adalah penjelasan lebih luas tentang analisis kesalahan dalam perhitungan numerik:

1. Kesalahan Aritmatika:

- a. Kesalahan Pembulatan: Terjadi karena keterbatasan dalam representasi bilangan dalam sistem komputer. Setiap operasi aritmatika, terutama pada bilangan floating-point, dapat menyebabkan kesalahan pembulatan.
- b. Kesalahan Truncation: Terjadi ketika digit yang tidak dapat direpresentasikan dalam representasi bilangan diabaikan atau dipotong.

2. Kesalahan Model Matematika:

- a. Kesalahan Pembulatan dalam Representasi Model: Model matematika yang digunakan untuk merepresentasikan suatu fenomena dalam bentuk algoritma komputasi juga dapat menyebabkan kesalahan pembulatan jika formulanya tidak dapat dipecahkan secara tepat.
- b. Pendekatan Deret Tak Hingga: Beberapa algoritma numerik melibatkan pendekatan deret tak hingga yang harus dipotong setelah beberapa suku. Kesalahan ini disebut kesalahan potong deret.

3. Kesalahan Metode Numerik:

- a. Kesalahan Aproksimasi: Metode numerik seringkali melibatkan aproksimasi nilai yang sebenarnya. Kesalahan ini

dapat disebabkan oleh penggunaan aproksimasi yang kasar atau langkah diskritisasi yang besar.

- b. **Kesalahan Iterasi:** Metode iteratif, seperti metode Newton-Raphson, dapat menghasilkan kesalahan iteratif yang akumulatif jika iterasi dihentikan sebelum konvergensi yang memadai.
- 4. **Kesalahan Pembulatan pada Sistem Komputer:**
 - a. **Overhead Kesalahan Mesin:** Kesalahan ini dapat muncul karena batasan representasi bilangan pada tingkat mesin dan cara instruksi dieksekusi oleh CPU.
- 5. **Kesalahan Input dan Output:**
 - a. **Kesalahan Pengukuran:** Kesalahan ini terkait dengan ketidakpastian dalam pengukuran dan input data ke dalam sistem. Penggunaan data yang tidak tepat atau terkontaminasi dapat menyebabkan kesalahan signifikan.
 - b. **Kesalahan Output:** Kesalahan ini terjadi saat hasil perhitungan atau output algoritma dikeluarkan dari sistem. Ini dapat mencakup kesalahan pembulatan pada nilai hasil atau format output yang tidak sesuai.
- 6. **Kesalahan Pemrograman:**
 - a. **Kesalahan Logika:** Kesalahan dalam merancang atau mengimplementasikan algoritma dapat menghasilkan kesalahan perhitungan. Misalnya, kesalahan logika pada langkah-langkah algoritma.
 - b. **Kesalahan Variabel dan Tipe Data:** Kesalahan dalam penggunaan variabel atau tipe data yang tidak tepat dapat menyebabkan kesalahan perhitungan.

7. Kesalahan Kesalahan Kondisi Awal:

- a. **Sensitivitas terhadap Kondisi Awal:** Beberapa algoritma numerik sangat sensitif terhadap kondisi awal atau parameter awal. Kesalahan kecil dalam nilai awal dapat menghasilkan perbedaan signifikan dalam hasil akhir.

Strategi untuk Mengatasi dan Mengurangi Kesalahan:

1. **Pemilihan Metode Numerik yang Sesuai:** Pilih metode numerik yang tepat untuk jenis masalah yang dihadapi dan pahami keterbatasan serta ketepatan metode tersebut.
2. **Pemahaman Terhadap Karakteristik Kesalahan:** Pahami sumber dan karakteristik kesalahan yang mungkin muncul selama perhitungan dan atasi secara tepat.
3. **Validasi dan Verifikasi:**
 - **Validasi Input:** Pastikan bahwa data yang diinput ke dalam sistem adalah valid dan akurat.
 - **Verifikasi Algoritma:** Uji algoritma secara cermat menggunakan kasus uji yang diketahui untuk memastikan hasil yang benar.
4. **Penanganan Kesalahan Pada Tingkat Kode:**
 - **Penanganan Kesalahan pada Kode:** Implementasikan pengkodean yang memperhitungkan dan menangani kesalahan yang mungkin terjadi selama eksekusi.
5. **Analisis Sensitivitas dan Stabilitas:** Lakukan analisis sensitivitas untuk mengevaluasi sejauh mana hasil perhitungan sensitif terhadap variasi input atau kondisi awal. Pastikan kestabilan algoritma terhadap kondisi awal dan parameter.

6. **Penggunaan Tipe Data yang Tepat:** Gunakan tipe data yang sesuai dengan kebutuhan dan meminimalkan potensi kesalahan pembulatan.
7. **Pelaporan dan Dokumentasi Kesalahan:** Dokumentasikan kesalahan yang muncul selama proses perhitungan dan hasil akhir. Ini membantu dalam debug dan perbaikan.

Analisis kesalahan dalam perhitungan numerik adalah langkah penting dalam memastikan keandalan dan keakuratan hasil perhitungan. Pemahaman yang baik tentang sumber kesalahan dan penerapan strategi untuk mengatasi atau meminimalkan kesalahan tersebut membantu meningkatkan kualitas perhitungan numerik.

BAB III

SOLUSI DAN PERSAMAAN NONLINEAR

A. Metode Pencarian Akar

Beberapa metode pencarian akar yang umum digunakan melibatkan metode numerik, seperti metode biseksi, metode Newton-Raphson, dan metode iterasi. Pilihan metode tergantung pada sifat persamaan dan kebutuhan komputasi.

Metode pencarian akar adalah serangkaian algoritma matematika yang dirancang untuk menemukan solusi atau akar dari suatu persamaan matematis, di mana akar ini adalah nilai dari variabel yang memenuhi persamaan tersebut. Pencarian akar atau solusi dari suatu persamaan menjadi relevan dalam berbagai disiplin ilmu, seperti matematika, fisika, ekonomi, dan teknik. Berikut adalah penjelasan lebih luas tentang metode pencarian akar:

1. Metode Bagi Dua (Bisection Method):

- a. **Konsep Dasar:** Metode ini memanfaatkan sifat kontinuitas fungsi. Jika fungsi memiliki nilai yang berbeda pada dua titik akhir interval dan hasil kali fungsi tersebut di dua titik tersebut berbeda tanda, maka di antara titik-titik tersebut terdapat akar persamaan.
- b. **Proses:** Inti dari metode ini adalah membagi interval pencarian menjadi dua dan memeriksa di mana fungsi berubah tanda. Proses ini diulangi secara iteratif hingga ditemukan solusi dengan tingkat ketepatan yang diinginkan.

2. Metode Newton-Raphson:

- a. **Konsep Dasar:** Metode ini memanfaatkan pendekatan turunan fungsi untuk mendekati lokasi akar. Dengan memulai dari suatu

tebakan awal, metode ini konvergen dengan cepat ke arah akar persamaan.

- b. **Proses:** Iterasi dilakukan dengan mengganti nilai tebakan dengan nilai yang lebih mendekati akar setiap kali. Perubahan nilai ini dihitung berdasarkan nilai fungsi dan turunannya pada tebakan tersebut.

3. Metode Titik Tetap (Fixed-Point Iteration):

- a. **Konsep Dasar:** Metode ini mengubah persamaan pencarian akar menjadi bentuk persamaan $f(x) = x$, di mana akar adalah titik tetap dari fungsi tersebut. Proses iteratif dilakukan untuk mendekati titik tetap tersebut.
- b. **Proses:** Pemilihan suatu fungsi iterasi ($g(x)$) yang memenuhi syarat konvergensi. Selanjutnya, iterasi dilakukan dengan formula $x[n+1] = g(x[n])$.

4. Metode Secant:

- a. **Konsep Dasar:** Metode ini merupakan perluasan dari metode bagi dua dan Newton-Raphson. Dengan menggantikan turunan dalam pendekatan Newton-Raphson dengan sebuah perbedaan yang terbatas, metode ini menemukan pendekatan yang cepat dengan dua titik awal.
- b. **Proses:** Dua titik awal dibutuhkan, dan proses iteratif dilakukan dengan menghitung titik-titik baru berdasarkan perbedaan terbatas dan nilai-nilai fungsi pada dua titik awal.

5. Metode Brent:

- a. **Konsep Dasar:** Metode Brent, juga dikenal sebagai algoritma Brent atau algoritma Van Wijngaarden-Dekker-Brent, adalah metode pencarian akar yang mengkombinasikan kestabilan metode bagi dua dengan kecepatan konvergensi metode inversi kuadrat.

- b. **Proses:** Metode ini menggunakan tiga titik (a, b, c) yang membentuk sebuah parabola yang diaproksimasi untuk mendekati akar persamaan. Metode ini bergantung pada interpolasi parabola dan metode bagi dua untuk mencapai kecepatan dan kestabilan.

6. Metode Regula Falsi:

- a. **Konsep Dasar:** Metode ini adalah variasi dari metode bagi dua, di mana titik tengah interval pencarian dihubungkan oleh garis lurus. Akar baru dihitung dengan menemukan perpotongan garis tersebut dengan sumbu x.
- b. **Proses:** Jika hasil kali fungsi pada dua ujung interval berbeda tanda, akar terdapat di antara dua ujung tersebut. Titik tengah baru dihitung sebagai perpotongan garis lurus antara dua ujung.

7. Metode Dekomposisi Kuadrat (Quadratic Decomposition Method):

- a. **Konsep Dasar:** Metode ini memecah persamaan kuadrat menjadi dua persamaan linear. Dengan cara ini, pencarian akar persamaan kuadrat menjadi lebih sederhana.
- b. **Proses:** Persamaan kuadrat $ax^2+bx+c=0$ dipecah menjadi dua persamaan linear, yaitu $x=x_1$ dan $x=x_2$. Kemudian, akar-akar persamaan kuadrat dapat ditemukan dari kedua persamaan linear tersebut.

Strategi dalam Memilih Metode:

- a. **Tipe Fungsi:** Pemilihan metode sebaiknya sesuai dengan sifat fungsi yang dicari akarnya.
- b. **Kemampuan Konvergensi:** Beberapa metode memiliki tingkat konvergensi yang lebih cepat daripada yang lain. Pilih metode yang cocok dengan tingkat ketelitian yang diinginkan.
- c. **Kondisi Awal:** Pilih metode yang dapat memberikan solusi yang baik berdasarkan kondisi awal atau tebakan yang diberikan.

Catatan Penting:

- a. Beberapa metode memerlukan syarat-syarat tertentu untuk konvergensi, dan pemilihan metode harus disesuaikan dengan karakteristik masalah yang dihadapi.
- b. Metode-metode ini bisa digunakan dalam kombinasi atau sebagai langkah-langkah dalam suatu iterasi untuk meningkatkan kecepatan dan kestabilan pencarian akar.

Dalam praktiknya, pemilihan metode pencarian akar bergantung pada kompleksitas persamaan dan kebutuhan spesifik dari suatu masalah matematika atau ilmiah. Analisis kesalahan dan kestabilan konvergensi juga menjadi faktor penting dalam menentukan metode yang paling sesuai untuk diterapkan.

B. Metode Bagi Dua (*Bisection*)

Metode bagi dua, atau bisection, adalah suatu metode numerik untuk mencari akar dari suatu fungsi matematis dengan membagi interval pencarian menjadi dua bagian, lalu menentukan di mana akar berada.

Metode bagi dua, juga dikenal sebagai metode pencarian akar interval atau metode pengurangan interval, adalah salah satu metode pencarian akar paling sederhana dan umum digunakan. Metode ini efektif untuk mencari akar dari suatu fungsi yang kontinu pada suatu interval tertutup. Berikut adalah penjelasan lebih luas tentang metode bagi dua:

Langkah-langkah Metode Bagi Dua:

1. **Penentuan Interval Awal:**
 - a. Pilih interval awal $[a, b]$ yang mengandung akar persamaan. Pastikan bahwa nilai fungsi pada ujung-ujung interval memiliki tanda yang berbeda, yaitu $f(a) \cdot f(b) < 0$. Ini menjamin bahwa akar terletak di dalam interval tersebut.

2. Iterasi:

- a. Hitung titik tengah c dari interval $[a, b]$: $c = \frac{a+b}{2}$
- b. Evaluasi nilai fungsi pada titik tengah: $f(c)$.
- c. Tentukan tanda dari $f(c)$.
- d. Jika $f(c)=0$, maka c adalah akar persamaan.
- e. Jika $f(c)$ memiliki tanda yang sama dengan $f(a)$, maka akar berada di interval $[c, b]$. Gantilah a dengan c .
- f. Jika $f(c)$ memiliki tanda yang sama dengan $f(b)$, maka akar berada di interval $[a, c]$. Gantilah b dengan c .
- g. Ulangi langkah ini hingga kondisi berhenti terpenuhi (misalnya, toleransi tertentu tercapai atau jumlah iterasi maksimum tercapai).

Kelebihan Metode Bagi Dua:

1. Konvergensi Terjamin:

- a. Metode ini dijamin konvergen untuk fungsi kontinu di interval tertutup dengan tanda yang berbeda pada ujung-ujungnya.

2. Kemudahan Penerapan:

- a. Metode ini relatif sederhana dan mudah dipahami. Algoritma yang jelas dan langkah-langkah iteratif yang sederhana membuatnya cocok untuk implementasi komputasi.

2. Stabilitas:

- a. Stabil pada fungsi yang kontinu dan memiliki akar di dalam interval awal yang dipilih.

Keterbatasan Metode Bagi Dua:

1. **Konvergensi yang Lambat:**
 - Metode ini cenderung konvergen dengan kecepatan yang lebih lambat jika dibandingkan dengan beberapa metode pencarian akar yang lebih canggih.
2. **Hanya Berlaku untuk Interval Awal yang Diketahui:**
 - Diperlukan pengetahuan awal tentang interval yang mengandung akar. Jika interval awal yang dipilih tidak memenuhi persyaratan tanda yang berbeda, metode ini tidak dapat diaplikasikan.
3. **Tidak Efisien untuk Fungsi yang Kompleks:**
 - Untuk fungsi yang kompleks atau memiliki sifat-sifat yang sulit, metode ini mungkin memerlukan jumlah iterasi yang besar untuk mencapai konvergensi.

Strategi dan Tips:

1. **Pemilihan Interval Awal yang Bijak:**
 - Pemilihan interval awal yang baik sangat penting. Interval tersebut harus memuat akar dan memiliki tanda yang berbeda pada ujungnya.
2. **Konvergensi Terpenuhi:**
 - Tentukan kriteria konvergensi yang memadai, seperti toleransi tertentu pada nilai fungsi atau selisih antara iterasi berturut-turut.
3. **Pemantauan Iterasi:**
 - Lakukan pemantauan iterasi untuk memastikan bahwa metode konvergen dan menghindari divergensi. Ini juga membantu dalam memahami tingkat konvergensi.

4. Implementasi Komputasi:

- Pastikan implementasi komputasi memperhatikan representasi bilangan dalam komputer dan mengelola kesalahan pembulatan dengan bijak.

Metode bagi dua adalah pilihan yang baik untuk pencarian akar fungsi sederhana di interval yang diketahui. Meskipun metode ini mungkin tidak secepat metode-metode iteratif yang lebih kompleks, keandalannya dan kemudahannya membuatnya tetap relevan dalam berbagai konteks aplikasi.

C. Metode Newton-Raphson

Metode Newton-Raphson adalah teknik iteratif dalam kalkulus numerik untuk mencari akar persamaan matematika. Ia menggunakan pendekatan iteratif untuk mendekati nilai akar dengan menggunakan turunan pertama fungsi yang dicari akarnya.

Metode Newton-Raphson:

Metode Newton-Raphson adalah salah satu metode iteratif yang digunakan untuk menemukan akar dari suatu fungsi matematika. Metode ini dapat digunakan untuk fungsi-fungsi yang kontinu dan memiliki turunan yang kontinu. Berikut adalah penjelasan lebih luas tentang metode Newton-Raphson:

Langkah-langkah Metode Newton-Raphson:

1. Tebakan Awal:

- a. Pilih suatu tebakan awal x_0 yang berada cukup dekat dengan akar yang dicari.

2. Iterasi:

- a. Gunakan rumus iterasi Newton-Raphson:

- b.
$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

di mana $f(x_n)$ adalah nilai fungsi pada tebakan x_n , dan $f'(x_n)$ adalah turunan pertama fungsi pada tebakan x_n .

b. Iterasi dilakukan sampai kondisi berhenti terpenuhi, misalnya, nilai fungsi mendekati nol atau selisih antara dua iterasi berturut-turut kecil.

3. Kondisi Berhenti:

a. Beberapa kondisi berhenti yang umum digunakan melibatkan pengecekan nilai fungsi atau selisih antara iterasi berturut-turut.

1. $|f(x_n)| < \text{Toleransi}$

2. $|x_{n+1} - x_n| < \text{Toleransi}$

3. Jumlah iterasi mencapai batas maksimum yang ditentukan.

Kelebihan Metode Newton-Raphson:

1. Konvergensi Cepat:

- Metode ini dapat menghasilkan konvergensi yang cepat terutama ketika tebakan awal cukup mendekati akar sebenarnya.

2. Efisiensi Tinggi:

- Jika kondisi konvergensi terpenuhi, metode ini dapat mencapai akurasi yang tinggi dengan jumlah iterasi yang relatif sedikit.

3. Penerapan dalam Pencarian Akar Fungsi:

- Cocok digunakan untuk pencarian akar fungsi matematika, terutama jika nilai awal yang baik dapat ditemukan.

Keterbatasan Metode Newton-Raphson:

1. Sensitivitas Terhadap Tebakan Awal:

- Kesuksesan metode ini sangat tergantung pada tebakan awal yang diberikan. Jika tebakan awal jauh dari akar sebenarnya atau berada di sekitar titik singgung yang ekstrem, metode ini dapat gagal atau menghasilkan solusi yang salah.

2. Mungkin Divergen:

- Metode ini bisa divergen jika turunan fungsi mendekati nol atau fungsi memiliki titik singgung yang ekstrem di sekitar tebakan awal.

3. Perlu Turunan Fungsi:

- Membutuhkan turunan pertama fungsi yang dianalisis. Untuk fungsi yang rumit atau di mana turunan sulit dihitung, ini dapat menjadi keterbatasan.

Strategi dan Tips:

1. Pemilihan Toleransi dan Kriteria Konvergensi:

- Pemilihan toleransi dan kriteria konvergensi harus disesuaikan dengan tingkat akurasi yang diinginkan.

2. Pemilihan Tebakan Awal yang Bijak:

- Pilih tebakan awal yang mendekati akar sebenarnya dan pastikan tebakan awal tidak terlalu dekat dengan titik singgung ekstrem yang mungkin menyebabkan divergensi.

3. Pemantauan Iterasi:

- Monitor iterasi untuk memahami perilaku metode. Jika metode mulai divergen atau konvergensi terlalu lambat, langkah-langkah perbaikan mungkin diperlukan.

4. Penanganan Singularitas:

- Atasi kemungkinan kegagalan metode akibat singularitas atau divergensi dengan melakukan penanganan khusus pada titik-titik yang memungkinkan terjadinya hal tersebut.

Contoh Penggunaan:

Misalkan kita ingin mencari akar dari fungsi $f(x)=x^2-4$, yang memiliki akar pada $x=2$ dan $x=-2$. Jika kita memilih tebakan awal $x_0=1$, kita dapat menggunakan metode Newton-Raphson untuk mengiterasi dan mendekati akar sebenarnya.

Iterasi:

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)}$$

$$x_1 = 1 - \frac{(1^2-4)}{(2 \cdot 1)} = \frac{3}{2}$$

Iterasi ini dapat diulangi hingga mendekati nilai akar sebenarnya $x=2$.

Metode Newton-Raphson sangat berguna dalam konteks di mana turunan fungsi mudah dihitung dan tebakan awal yang baik dapat ditemukan. Meskipun memiliki keterbatasan, metode ini tetap menjadi pilihan yang populer dalam analisis numerik dan optimisasi.

D. Analisis Konvergensi dan Ketidakpastian

Analisis konvergensi mengacu pada evaluasi sejauh mana hasil suatu proses mendekati nilai yang seharusnya. Sementara itu, ketidakpastian adalah ukuran sejauh mana kita yakin dengan hasil atau estimasi yang diberikan.

Analisis konvergensi bertujuan untuk mengevaluasi sejauh mana suatu metode numerik atau algoritma dapat mendekati solusi yang tepat seiring dengan bertambahnya iterasi atau langkah. Pada dasarnya, ini adalah langkah kritis untuk memastikan bahwa suatu algoritma

memberikan hasil yang stabil dan dapat diandalkan. Faktor-faktor yang mempengaruhi konvergensi melibatkan tebakan awal yang bijak, sifat fungsi, dan metode numerik yang digunakan.

Contoh Analisis Konvergensi:

Misalnya, kita menggunakan metode Newton-Raphson untuk mencari akar dari fungsi kuadratik $f(x)=x^2-4$. Dengan tebakan awal $x_0=1$, kita dapat melakukan iterasi sebagai berikut:

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)}$$

$$x_1 = 1 - \frac{(1^2-4)}{(2 \cdot 1)} = \frac{3}{2}$$

$$x_2 = x_1 - \frac{f(x_1)}{f'(x_1)}$$

$$x_2 = \frac{3}{2} - \frac{\left(\left(\frac{3}{2}\right)^2-4\right)}{(2 \cdot \frac{3}{2})} = \frac{17}{12}$$

Iterasi ini dapat terus dilakukan hingga solusi konvergen. Analisis konvergensi akan melibatkan evaluasi seberapa cepat nilai x mendekati solusi sebenarnya, dan faktor-faktor seperti tebakan awal dan sifat fungsi akan memainkan peran penting dalam proses ini.

Analisis Ketidakpastian:

Analisis ketidakpastian berkaitan dengan tingkat ketidakpastian atau ketidakpastian dalam hasil perhitungan atau eksperimen. Kesalahan pembulatan, kesalahan input data, dan ketidakpastian model atau algoritma dapat menyebabkan hasil yang tidak pasti. Metode untuk mengukur dan mengelola ketidakpastian melibatkan analisis sensitivitas, interval kepercayaan, dan pemodelan ketidakpastian.