Gumatika
Kabinet Niscala Aletta

Catatan Kuliah

Metode Numerik

MAT1233

Daniel Juniro Aslacris Situmorang
Keilmuan dan Pengembangan Akademik

Revisi 4 Maret 2025

Bab 1

Pengenalan Metode Numerik

1.1 Apa itu *scientific computing*

Scientific Computing, atau yang lazim disebut Analisis Numerik adalah ilmu mendesain dan menganalisas algoritma untuk menyelesaikan masalah matematis secara numerikal. Biasanya *scientific computing* atau analisis numerik dipakai dalam simulasi fenomena alam atau untuk *prototyping* dalam membuat model teknik
Strategi umum dalam menyelesaikan masalah matematis yang rumit adalah sebagai berikut

- Tak hingga $\rightarrow$ berhingga
- Diferensial $\rightarrow$ aljabar
- Non-linier $\rightarrow$ linier
- Rumit $\rightarrow$ simpel

Perlu diingat bahwa dalam metode numerik/analisis numerik, kita mencari *approximation* atau **hampirannya**

1.2 Approximation

Salah satu contoh yang akan digunakan adalah ketika kita menghampiri luas permukaan bumi dengan rumus luas permukaan bola $A = 4\pi r$. Penggunaan rumus ini menggunakan asumsi bahwa:

- Bumi dianggap sebuah bola sempurna, yang pada realitanya tidak
- Nilai jari-jari tidak bisa ditentukan secara pasti, hanya mengandalkan pengukuran yang sudah ada
- Komputer tidak dapat memuat nilai π secara benar, maka hanya dapat ditampilkan hasil dari pemotongan nilai tak hingga π
- Nilai dari data input dan output algoritma dibulatkan oleh proses komputer

1.3 Galat absolut dan Galat Relative

Rumus dalam mencari nilai galat absolut:

$$\text{Galat absolut} = \text{nilai hampiran} - \text{nilai eksak}$$

Sedangkan, rumus dalam mencari nilai galat relatif:

$$\text{Galat relatif} = \frac{\text{galat absolut}}{\text{nilai eksak}}$$

dari dua persamaan di atas didapat Nilai hampiran = (nilai eksak) $\cdot$ (1 + galat relatif). Biasanya, nilai eksak tidak diketahui, sehingga kita memperkirakan galat atau membatasi errornya. Hal ini juga membuat nilai galat relatif biasanya diambil terhadap nilai hampiran, dikarenakan nilai eksaknya yang tidak diketahui

1.4 Galat data dan Galat Komputasi

Contoh masalah : carilah nilai dari fungsi $f : \mathbb{R} \to \mathbb{R}$ dengan argumen/data sebagai berikut:

- x = nilai eksak input
- $f(x)$ = nilai yang diinginkan
- $\hat{x}$ = nilai hampiran
- $\hat{f}$ = fungsi hampiran yang dijalankan komputer

Galat total : $\hat{f}(\hat{x}) - f(x) =$

$$\underbrace{\hat{f}(\hat{x}) - f(\hat{x})}_{\text{Galat Komputasi}} + \underbrace{f(\hat{x}) - f(x)}_{\text{Galat input data}}$$

dari persamaan diatas, dapat disimpulkan bahwa algoritma tidak berpengaruh pada galat input data.

1.5 Galat pemotongan dan galat pembulatan

Galat Komputasi adalah gabungan dari Galat Pemotongan dan Pembulatan, namun biasanya salah satu lebih dominan

1.5.1 Galat pemotongan

Galat pemotongan adalah selisih antara nilai eksak dengan hasil yang diberikan algoritma menggunakan aritmatika eksakta. Galat ini sering terjadi ketika ketika kita menghampiri deret tak hingga atau ketika memotong barisan iteratif sebelum fungsi menjadi konvergen.

1.5.2 Galat pembulatan

Galat pembulatan adalah selisih antara nilai yang dihasilkan algoritma dengan aritmatika eksak dengan niali yang diberikan algoritma dengan aritmatika dengan tingkat presisi yang terbatas. Hal ini terjadi karena representasi bilangan real yang kurang tepat dan juga operasi matematika yang digunakan.

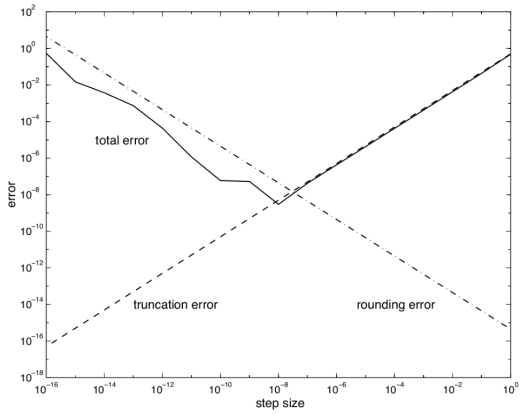
1.5.3 Contoh kasus : Hampiran turunan berhingga

Salah satu contoh yang memperlihatkan cara kerja galat pemotongan dan galat pemotongan adalah galat pada hampiran turunan berhingga.

$$f'(x) \approx \frac{f(x+h) - f(x)}{h}$$

- Galat pemotongan dibatasi oleh $\frac{Mh}{2}$, dimana M dibatasi $|f"(t)|$ untuk t di sekitar x
- Galat pembulatan dibatasi oleh $2\epsilon/h$, dimana galat nilai fungsi dibatasi oleh ϵ
- Galat total diminimalkan ketika $h \approx 2\sqrt{\epsilon/M}$

Pada contoh kita ini, untuk nilai h yang kecil, maka nilai galat pembulatan bertambah. Sedangkan, jika nilai h besar, maka nilai pemotongan akan bertambah.



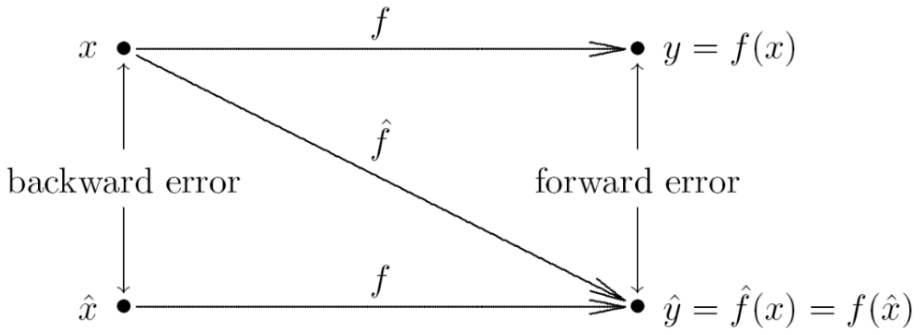
Gambar 1.1: Ilustrasi Galat Pemotongan dan Galat Pembulatan

1.6 Galat maju dan galat mundur

Misalkan kita ingin mencari nilai $y = f(x)$, where $f : \mathbb{R} \to \mathbb{R}$, dan kita mendapatkan nilai $\hat{y}$, maka didapat :

- Galat maju/*forward error*: $\Delta y = \hat{y} - y$
- Galat mundur/*backward error*: $\Delta x = \hat{x} - x$

Untuk ilustrasi galat maju dan mundur dapat dilihat dibawah ini:



Contoh 1.1 Misal $y = \sqrt{2}$ dan $\hat{y} = 1.4$. Galat maju absolutnya adalah?

Solusi.

$$|\Delta y| = |\hat{y} - y| = |1.4 - 1.41421...| \approx 0.0142$$

Dari persamaan di atas, didapat juga galat relatifnya sebesar 1 persen (gunakan rumus galat relatif pada sub-bagian sebelumnya yaa :D). ✠

Contoh 1.2 Misal $\sqrt{1.96} = 1.4$. Tentukan nilai galat mundur absolut!

Solusi.

$$|\Delta x| = |\hat{x} - x| = |1.96 - 2| \approx 0.04$$

Dari persamaan di atas, didapat juga bahwa galat mundur relatifnya 2 persen(*Once again*, jangan lupa cek rumusnya di sub-bagian diatas hehe :D). ✠

1.7 Sensitivitas dan conditioning

Sebuah permasalahan dianggap tidak sensitif, atau *well-conditioned*, jika perubahan relatif pada input juga menghasilkan perubahan relatif yang sama pada output. Di lain sisi, permasalahan dianggap sensitif atau *ill-conditioned*, jika perubahan relatif input dapat mengubah drastis outputnya.

1.7.1 Angka Kondisi

$$\text{cond} = \left| \frac{\text{Perubahan relatif output}}{\text{Perubahan relatif data input}} \right| = \left| \frac{\Delta y / y}{\Delta x / x} \right|$$

Jika nilai dari cond » 1, maka permasalahan dianggap sensitif. Angka kondisi adalah faktor amplifikasi yang menghubungkan galat maju relatif dengan galat mundur relatif.

$$|\text{galat maju relatif}| = \text{cond} \times |\text{galat mundur relatif}|$$

Angka kondisi biasanya tidak diketahui secara pasti dan dapat bervariasi tergantung pada input, sehingga digunakan perkiraan kasar atau batas atas untuk cond, yang menghasilkan:

$$|\text{galat maju relatif}| \approx \text{cond} \times |\text{galat mundur relatif}|$$

1.7.2 Stabilitas

- Suatu algoritma dapat dikatakan **stabil** jika hasil yang diberikan tidak sensitif secara relatif pada gangguan yang terjadi saat komputasi
- Stabilitas algoritma sebanding denga pengkondisian masalah
- Dari sudut pandang analisis galat mundur, algoritma dikatakan stabil jika output yang dihasilkan adalah solusi eksak dari masalah serupa.
- Untuk algoritma yang stabil, efek dari galat komputasional tidak akan pernah lebih besar dari efek galat data kecil pada input

1.7.3 Akurasi

akurasi adalah seberapa deketnya hasil komputasi dengan hasil aslinya. Stabilitas sendiri tidak menjamin akurasi, dikarenakan akurasi tergantung pada pengondisian masalah dan juga stabilitas algoritmanya. Ketidakakurasian dapat terjadi karena kita menggunakan algoritma stabil pada masalah yang tidak dikondisikan dengan baik(*Ill-conditioned*), atau sebaliknya masalah yang dikondisikan dengan baik (*well conditioned*)yang diselesaikan dengan algoritma yang tidak stabil.
Diperlukan masalah yang dikondisikan dengan baik dengan algoritma yang stabil untuk mendapatkan akurasi yang baik juga