METODE NUMERIK

Pendahuluan

Persoalan yang melibatkan model matematika banyak muncul dalam berbagai disiplin ilmu pengetahuan, seperti dalam bidang fisika, kimia, ekonomi, atau pada persoalan rekayasa (*engineering*), seperti Teknik Sipil, Teknik Mesin, Elektro, dan sebagainya.

Seringkali model matematika tersebut muncul dalam bentuk yang tidak ideal alias rumit. Model matematika yang rumit ini adakalanya tidak dapat diselesaikan dengan metode analitik yang sudah umum untuk mendapatkan **solusi sejatinya (*exact solution*)**. Yang dimaksud dengan metode analitik adalah metode penyelesaian model matematika dengan rumus-rumus aljabar yang sudah baku (lazim).

Pendahuluan

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

i. Tentukan akar-akar persamaan polinom:

$$23{,}4x^7 - 1.25x^6 + 120x^4 + 15x^3 - 120x^2 - x + 100 = 0$$

ii. Selesaikan sistem persamaan lanjar (*linear*):

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

Pendahuluan

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

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

Hitung taksiran nilai y untuk $x = 3.8$!

BAGAIMANA CARA ANDA MENYELESAIKAN PERSOALAN-PERSOALAN TERSEBUT???

Pendahuluan

Metode Numerik adalah teknik untuk menyelesaikan masalah-masalah yang diformulasikan secara matematis dengan cara operasi hitungan (aritmatika).

Metode numerik juga adalah metode yang menyempurnakan metode analisis, jika metode analisis tidak memecahkan suatu masalah maka solusinya adalah dengan metode numerik.

Metode numerik disimulasikan dengan menggunakan program komputerisasi, sehingga perhitungannya bisa cepat dan mudah.

Pendahuluan

Prinsip-prinsip Metode Numerik

- **Jika metode analitik** tidak dapat digunakan lagi, maka **metode numerik** digunakan.
- **Metode Numerik** merupakan pendekatan untuk mendapatkan pemecahan masalah yang dapat dipertanggung jawabkan secara **analitik.**
- Pendekatannya merupakan **analisis matematis.**
- **Metode Numerik** terdiri atas **algoritma-algoritma** yang dapat dihitung secara cepat dan mudah
- **Metode Numerik** ini akan memakai **iterasi (pengulangan),** karena menggunakan **algoritma pendekatan.**
- Nilai kesalahan merupakan hal paling utama untuk mengetahui seberapa baik **metode** yang digunakan.

Pendahuluan

METODE ANALITIK

Dikenal sebagai **metode sejati, karena menghasilkan solusi sejati/eksak (sebenarnya).** Solusi dapat berupa fungsi matematik yang dapat dievaluasi kembali untuk mendapatkan nilai berbentuk angka.

- **Kelemahan :**

Digunakan hanya untuk penyelesaian persoalan matematika tertentu saja yaitu persoalan yang memiliki tafsiran geometri sederhana serta berdimensi rendah.

METODE NUMERIK

- Metode : cara, sedangkan numerik : angka**, sehingga dapat dikatakan bahwa: Metode numerik adalah cara berhitung menggunakan angka-angka.**
- **solusi berupa solusi hampiran atau bukan solusi eksak.**

Solusi **HAMPIRAN** tidak **TEPAT SAMA** dengan solusi **EKSAK.** Terdapat **SELISIH (GALAT)** antara **KEDUANYA.**

Pendahuluan

ANALISIS NUMERIK

Terapan matematika untuk menganalisis metode. Misal analisis galat dan kecepatan konvergensi sebuah metode.

METODE NUMERIK

Algoritma, menyangkut langkah-langkah penyelesaian persoalan secara numerik.

Pendahuluan

Kenapa Harus Belajar Metode Numerik?

Terdapat beberapa alasan mengapa harus belajar metode numerik :

1. Metode numerik merupakan alat bantu pemecahan masalah matematika yang sangat ampuh. *Metode numerik mampu menangani sistem persamaan besar, ketidaklinearan, dan geometri yang rumit yang seringkali tidak mungkin dipecahkan secara analitik.*

2. Dapat mengetahui alur penyelesaian soal matematika yang terdapat pada program aplikasi numerik komersial.

3. Dapat membuat sendiri program komputer.

Pendahuluan

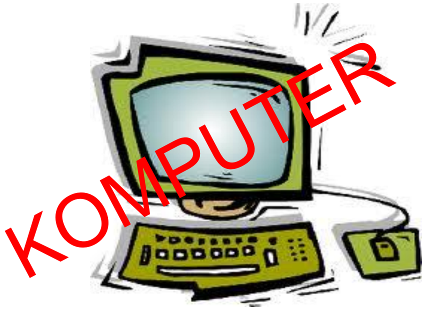
Real World Problem

Pemodelan Matematika

Formulasi Numerik

Pemograman

Komputer
&
Metode Numerik



Dalam proses perhitungannya (algoritma) dilakukan dengan iterasi dalam jumlah yang sangat banyak dan berulang-ulang

Analisis Galat

Analisis Galat

Menganalisis galat sangat penting di dalam perhitungan yang menggunakan metode numerik. Galat berasosiasi dengan seberapa dekat solusi hampiran terhadap solusi sejatinya. Semakin kecil galatnya, semakin teliti solusi numerik yang didapatkan. Perlu dipahami dua hal terkait analisis galat: yaitu

(a) bagaimana menghitung galat, dan

(b) bagaimana galat timbul.

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

$$\varepsilon = a - \hat{a}$$

Disebut **galat**.

Analisis Galat

Sebagai contoh, jika $\hat{a}$= 10.5 adalah nilai hampiran dari *a* = 10.45, maka galatnya adalah ε = -0.01.

Jika tanda galat (positif atai negatif) tidak dipertimbangkan, maka **galat mutlak** dapat didefinisikan sebagai

$$|\varepsilon| = |a - \hat{a}|$$

Sayangnya, ukuran galat ε kurang bermakna sebab tidak menceritakan seberapa besar galat tersebut dibandingkan dengan nilai sejatinya.

Analisis Galat

Sebagai contoh, seorang anak melaporkan panjang sebatang kawat 99 cm, padahal panjang sebenarnya 100 cm. Galatnya adalah 100 - 99 = 1 cm. Anak yang lain melaporkan panjang sebatang pensil 9 cm, padahal panjang sebenarnya 10 cm, sehingga galatnya juga 1 cm. Kedua galat pengukuran sama-sama bernilai 1 cm, namun galat 1 cm pada pengukuran panjang pensil lebih berarti daripada galat 1 cm pada pengukuran panjang kawat.

Jika tidak ada informasi mengenai panjang sesungguhnya, dimungkinkan akan menganggap kedua galat tersebut sama saja. Untuk mengatasi interpretasi nilai galat ini, maka galat harus *dinormalkan* terhadap nilai sejatinya. Gagasan ini melahirkan apa yang dinamakan **galat relatif.**

Analisis Galat

Galat relatif didefinisikan sebagai:

$$\boldsymbol{\varepsilon_R = \frac{|\varepsilon|}{a}}$$

atau dalam **presentase**

$$\boldsymbol{\varepsilon_R = \frac{|\varepsilon|}{a} \times 100\%}$$

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

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

$$\boldsymbol{\varepsilon_{RA} = \frac{|\varepsilon|}{\hat{a}}}$$

Analisis Galat

Ilustrasi

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

Penyelesaian:
galat = 10/3 – 3.333 = 10/3 – 3333/1000 = 1/3000 = 0.000333…
galat mutlak = | 0.000333…| = 0.000333…
galat relatif = (1/3000)/(10/3) = 1/1000 = 0.0001
galat relatif hampiran = (1/3000)/3.333 = 1/9999

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

$$\boldsymbol{\varepsilon_{RA} = \left|\frac{a_{r+1} - a_r}{a_{r+1}}\right|}$$

Analisis Galat

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

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

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

Analisis Galat

Ilustrasi

Misalkan ada prosedur lelaran sebagai berikut
$x_{r+1} = (-x_r^3 + 3)/6$, r = 0, 1, 2, 3, ...
Lelaran dihentikan bila kondisi $|\varepsilon_{RA}| < \varepsilon_S$, dalam hal ini ε_S adalah toleransi galat yang diinginkan. Misalkan dengan memberikan x_0 = 0.5, dan ε_S = 0.00001 diperoleh runtunan:

$$x_0 = 0.5$$

$$x_1 = 0.4791667 ; \left|\varepsilon_{RA} = \frac{x_1 - x_0}{x_1}\right| = 0.043478 > \varepsilon_S$$

$$x_2 = 0.4816638; \left|\varepsilon_{RA} = \frac{x_2 - x_1}{x_2}\right| = 0.0051843 > \varepsilon_S$$

$$x_3 = 0.4813757; \left|\varepsilon_{RA} = \frac{x_3 - x_2}{x_3}\right| = 0.0005984 > \varepsilon_S$$

$$x_4 = 0.4814091; \left|\varepsilon_{RA} = \frac{x_4 - x_3}{x_4}\right| = 0.0000693 > \varepsilon_S$$

$$x_5 = 0.4814052; \left|\varepsilon_{RA} = \frac{x_5 - x_4}{x_5}\right| = 0.0000081 < \varepsilon_S$$, **berhenti!**

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

Analisis Galat

Sumber Galat

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

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

Selain kedua galat ini, masih ada sumber galat lain, antara lain:

a. **Galat eksperimental**, yaitu galat yang timbul dari data yang diberikan, misalnya karena kesalahan pengukuran, ketidaktelitian alat ukur, dan sebagainya.

b. **Galat pemrograman**. Galat yang terdapat di dalam program sering dinamakan dengan kutu (*bug*), dan proses penghilangan galat ini dinamakan penirkutuan (*debugging*).

Analisis Galat

Galat Pemotongan

Galat pemotongan mengacu pada galat yang ditimbulkan akibat penggunaan hampiran sebagai pengganti formula eksak. Maksudnya, ekspresi matematik yang lebih kompleks "diganti" dengan formula yang lebih sederhana. Tipe galat pemotongan bergantung pada metode komputasi yang digunakan untuk penghampiran sehingga kadang-kadaang ia disebut juga **galat metode**.

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

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

$$\cos(x) \approx \underbrace{\underbrace{1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!}}_{nilai\ hampiran} \Bigg| \underbrace{+ \frac{x^8}{8!} - \frac{x^{10}}{10!} + \cdots}_{galat\ pemotongan}}_{pemotongan}$$

Analisis Galat

Galat Pembulatan

Perhitungan dengan metode numerik hampir selalu menggunakan bilangan riil. Masalah timbul bila komputasi numerik dikerjakan oleh mesin (dalam hal ini komputer) karena semua bilangan riil tidak dapat disajikan secara tepat di dalam komputer. Keterbatasan komputer dalam menyajikan bilangan riil menghasilkan galat yang disebut **galat pembulatan**.

Sebagai contoh 1/6 = 0.166666666… tidak dapat dinyatakan secara tepat oleh komputer karena digit 6 panjangnya tidak terbatas. Komputer hanya mampu merepresentasikan sejumlah digit (atau bit dalam sistem biner) saja. Bilangan riil yang panjangnya melebihi jumlah digit (bit) yang dapat direpresentasikan oleh komputer dibulatkan ke bilangan terdekat. Misalnya sebuah komputer hanya dapat merepresentasikan bilangan riil dalam 6 digit angka berarti, maka representasi bilangan 1/6 = 0.1666666666… di dalam komputer 6-digit tersebut adalah 0.166667. Galat pembulatannya adalah 1/6 – 0.166667 = -0.000000333.

TERIMA KASIH