Deret Taylor dan Analisis Galat

Kuliah ke-2 IF4058 Topik Khusus Informatika I

Oleh; Rinaldi Munir (IF-STEI ITB)
Deret Taylor

- **Kakas** (*tools*) yang sangat penting dalam metode numerik adalah **deret Taylor**.

- Deret Taylor adalah kakas yang utama untuk menurunkan suatu metode numerik.

- Dere Taylor berguna untuk menghampiri fungsi ke dalam bentuk polinom

- Fungsi yang rumit menjadi sederhana dengan deret Taylor
Definisi Deret Taylor

Andaikan $f$ dan semua turunannya, $f', f'', f''', ...$, menerus di dalam selang $[a, b]$. Misalkan $x_0 \in [a, b]$, maka untuk nilai-nilai $x$ di sekitar $x_0$ (Gambar 2.1) dan $x \in [a, b]$, $f(x)$ dapat diperluas (diekspansi) ke dalam deret Taylor:

$$f(x) = f(x_0) + \frac{(x-x_0)}{1!} f'(x_0) + \frac{(x-x_0)^2}{2!} f''(x_0) + \frac{(x-x_0)^3}{3!} f'''(x_0) + ... \frac{(x-x_0)^m}{m!} f^{(m)}(x_0) + ...$$

Misalkan $x - x_0 = h$, maka:

$$f(x) = f(x_0) + \frac{h}{1!} f'(x_0) + \frac{h^2}{2!} f''(x_0) + \frac{h^3}{3!} f'''(x_0) + ... \frac{h}{m!} f^{(m)}(x_0) + ...$$
**Contoh 1:** Hampir fungsi $f(x) = \sin(x)$ ke dalam deret Taylor di sekitar $x_0 = 1$.

Penyelesaian:

\[
f(x) = \sin(x), \quad f'(x) = \cos(x), \quad f''(x) = -\sin(x),
\]
\[
f'''(x) = -\cos(x), \quad f^{(4)}(x) = \sin(x), \ldots
\]

\[
\sin(x) = \sin(1) + \frac{(x - 1)}{1!} \cos(1) + \frac{(x - 1)^2}{2!} (-\sin(1)) + \frac{(x - 1)^3}{3!} (-\cos(1)) + \frac{(x - 1)^4}{4!} \sin(1) + \ldots
\]

Bila dimisalkan $x - 1 = h$, maka,

\[
\sin(x) = \sin(1) + h \cos(1) + \frac{h^2}{2} (-\sin(1)) + \frac{h^3}{6} (-\cos(1)) + \frac{h^4}{24} \sin(1) + \ldots
\]

\[
= 0.8415 + 0.5403h - 0.4208h^2 - 0.0901h^3 + 0.0351h^4 + \ldots
\]
• Kasus khusus: jika $x_0 = 0$, maka deretnya dinamakan deret **Maclaurin**, yang merupakan deret Taylor baku.

• **Contoh 2**: $\sin(x)$, $e^x$, $\cos(x)$ dan $\ln(x + 1)$ masing-masing dalam deret Maclaurin

$$\sin(x) = \sin(0) + \frac{(x-0)}{1!} \cos(0) + \frac{(x-0)^2}{2!} (-\sin(0)) + \frac{(x-0)^3}{3!} (-\cos(0)) + \frac{(x-0)^4}{4!} \sin(0) + ...$$

$$= x - \frac{x^3}{3!} + \frac{x^5}{5!} - ...$$

$$e^x = e^{(0)} + \frac{(x-0)}{1!} e^{(0)} + \frac{(x-0)^2}{2!} e^{(0)} + \frac{(x-0)^3}{3!} e^{(0)} + \frac{(x-0)^4}{4!} e^{(0)} + ...$$

$$= 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!} + ...$$
\[
\cos(x) = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \ldots
\]

\[
\ln(x + 1) = \ln(0 + 1) + \frac{(x - 0)}{1!}(0 + 1)^{-1} + \frac{(x - 0)^2}{2!}(-(0 + 1)^{-2}) + \frac{(x - 0)^3}{3!}2(0 + 1)^{-3} + \frac{(x - 0)^4}{4!}(-6(0 + 1)^{-4}) + \ldots
\]

\[
= x - \frac{x^2}{2} + \frac{x^3}{3} - \frac{x^4}{4} + \ldots
\]
• Karena suku-suku deret Taylor tidak berhingga banyaknya, maka -untuk alasan praktis- deret Taylor dipotong sampai suku orde tertentu.

• Deret Taylor yang dipotong sampai suku orde ke-\(n\) dinamakan deret Taylor terpotong dan dinyatakan oleh:

\[
f(x) \approx f(x_0) + \frac{(x-x_0)}{1!} f'(x_0) + \frac{(x-x_0)^2}{2!} f''(x_0) + \ldots + \frac{(x-x_0)^n}{n!} f^{(n)}(x_0) + R_n(x)
\]

\[
R_n(x) = \frac{(x-x_0)^{(n+1)}}{(n+1)!} f^{(n+1)}(c), \quad x_0 < c < x
\]

Galat/residu/sisa
• Deret Taylor terpotong di sekitar \( x_0 = 0 \) disebut **deret Maclaurin terpotong**.

**Contoh 3:**

\[
\sin(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} + R_5(x); \quad R_5(x) = \frac{x^6}{6!} \cos(c)
\]

\[
e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!} + R_4(x); \quad R_4(x) = \frac{x^5}{5!} e^c
\]

\[
\cos(x) = 1 - \frac{x^2}{4!} + \frac{x^4}{6!} - \frac{x^6}{6!} + R_6(x); \quad R_6(x) = \frac{x^7}{7!} \cos(c)
\]

\[
\ln(x+1) = x - \frac{x^2}{2} + \frac{x^3}{3} - \frac{x^4}{4}; + R_4(x); \quad R_4(x) = \frac{x^5}{5!} (c + 1)^{-5}
\]

yang dalam hal ini, \( 0 < c < x \).
• **Contoh 4**: Hitung hampiran nilai \( \cos(0.2) = \ldots \)

Jawab: \( \cos(0.2) \approx 1 - 0.2^2/2 + 0.2^4/24 - 0.2^6/720 \)

\[= 0.9800667\]
Analisis Galat

• Solusi dengan metode numerik adalah solusi hampiran (aproksimasi)
• Hampiran terhadap solusi eksak
• Oleh karena itu, solusi numerik mengandung galat.
• **Galat** ($\varepsilon$): perbedaan antara solusi hampiran dengan solusi eksak.

\[
\varepsilon = a - \hat{a}
\]

• Galat mutlak: $|\varepsilon| = |a - \hat{a}|$
• Galat relatif: \( \varepsilon_R = \frac{\varepsilon}{a} \) atau \( \varepsilon_R = \frac{\varepsilon}{a} \times 100\% \)

• Galat relatif hampiran: \( \varepsilon_{RA} = \frac{\varepsilon}{\hat{a}} \)

**Contoh 5:** 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
Sumber utama galat:

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

- **Galat pemotongan**: galat yang ditimbulkan akibat penggunaan hampiran sebagai pengganti formula eksak

Contoh: hampiran \( \cos(x) \) dengan deret McLaurin:

\[
\cos(x) \approx 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} \underbrace{+ \frac{x^8}{8!} - \frac{x^{10}}{10!}}_{\text{nilai hampiran}} + \underbrace{\ldots}_{\text{galat pemotongan}}
\]
• **Galat pembulatan**: galat yang timbul akibat keterbatasan komputer dalam merepresentasikan bilangan riil.

• **Contoh 6**: \( \frac{1}{6} = 0.1666666666... \), dalam mesin dengan 6-digit direpresentasikan sebagai 0.166667. Galat pembulatan = \( \frac{1}{6} - 0.166667 = -0.000000333 \).

• Contoh dalam sistem biner misalnya \( \frac{1}{10} = 0.0001100110011001100110011..._2 \) direpresentasikan di dalam komputer dalam jumlah bit yang terbatas.
Representasi bilangan riil di dalam komputer:

1. Bilangan titik-tetap (*fixed-point*)

   Setiap bilangan riil disajikan dengan jumlah tempat desimal yang tetap

   Contoh: 62.358, 0.013, 1.000.

2. Bilangan titik-kambang (*floating-point*)

   Setiap bilangan riil disajikan dengan jumlah digit *berarti* yang sudah tetap

   Contoh: $0.6238 \times 10^3$, $0.1714 \times 10^{-13}$
Angka Bena (signifikan)

- Angka bena adalah angka bermakna, angka penting, atau angka yang dapat digunakan dengan pasti
- Contoh:
  - 43.123 memiliki 5 angka bena (yaitu 4, 3, 1, 2, 3)
  - 0.1764 memiliki 4 angka bena (yaitu 1, 7, 6, 4)
  - 0.0000012 memiliki 2 angka bena (yaitu 1, 2)
  - 278.300 memiliki 6 angka bena (yaitu 2, 7, 8, 3, 0, 0)
  - 270.0090 memiliki 7 angka bena (yaitu 2, 7, 0, 0, 0, 9, 0)
  - 0.0090 memiliki 2 angka bena (yaitu 9, 0)
  - 1360, 1.360, 0.001360 semuanya memiliki 4 angka bena
• Komputer hanya menyimpan sejumlah tertentu angka bena.

• Bilangan riil yang jumlah angka benanya melebihi jumlah angka bena komputer akan disimpan dalam sejumlah angka bena komputer itu.

• Pengabaian angka bena sisanya itulah yang menimbulkan galat pembulatan.
• **Galat total**: adalah galat akhir pada solusi numerik merupakan jumlah galat pemotongan dan galat pembulatan.

**Contoh 7:**

\[
\cos(0.2) \approx 1 - \frac{0.2^2}{2} + \frac{0.2^4}{24} \approx 0.9800667
\]

\[
\uparrow \quad \uparrow
\]

galat pemotongan  
galat pembulatan

Pada contoh di atas, galat pemotongan timbul karena kita menghampiri \( \cos(0.2) \) sampai suku orde empat, sedangkan galat pembulatan timbul karena kita membulatkan nilai hampiran ke dalam 7 digit benua.
Bilangan Titik-Kambang

- Bilangan riil di dalam komputer umumnya disajikan dalam format *bilangan titik-kambang*
- Bilangan titik-kambang *a* ditulis sebagai

\[ a = \pm m \times B^p = \pm 0.d_1d_2d_3d_4d_5d_6 \ldots d_n \times B^p \]

- *m* = mantisa (riil), *d_1d_2d_3d_4d_5d_6 \ldots d_n* adalah digit mantisa.
- *B* = basis sistem bilangan yang dipakai (2, 8, 10, 16, dsb)
- *p* = pangkat (berupa bilangan bulat), dari \(-P_{\text{min}}\) sampai \(+P_{\text{maks}}\)

- Contoh: 245.7654 \(\rightarrow\) 0.2457654 \(\times\) 10^3
Bilangan Titik-Kambang Ternormalisasi

• Syarat: digit mantis yang pertama tidak boleh 0

\[ a = \pm m \times B^p = \pm 0.d_1d_2d_3d_4d_5d_6 \ldots d_n \times B^p \]

\[ 1 \leq d_1 \leq b - 1 \text{ dan } 0 \leq d_k \leq B-1 \text{ untuk } k > 1. \]

• Pada sistem desimal, \( 1 \leq d_1 \leq 9 \) dan \( 0 \leq d_k \leq 9 \), sedangkan pada sistem biner, \( d_1 = 1 \) dan \( 0 \leq d_k \leq 1 \).

• Contoh 8: \( 0.0563 \times 10^{-3} \Rightarrow 0.563 \times 10^{-4} \),

\[ 0.00023270 \times 10^6 \Rightarrow 0.23270 \times 10^3. \]
Pembulatan pada Bilangan Titik-Kambang

• Bilangan riil di dalam komputer mempunyai rentang nilai yang terbatas.

• Bilangan titik-kambang yang tidak dapat mencocok satu dari nilai-nilai di dalam rentang nilai yang tersedia, dibulatkan ke salah satu nilai di dalam rentang.

• Galat yang timbul akibat penghampiran tersebut diacu sebagai galat pembulatan.

• Ada dua teknik pembulatan yang lazim dipakai oleh komputer, yaitu pemenggalan (chopping) dan pembulatan ke digit terdekat (in-rounding).
Pemenggalan \textit{(chopping)}

Misalkan \( a = \pm 0.d_1d_2d_3 \ldots d_n d_{n+1} \ldots \times 10^p \)

\[ fl_{\text{chop}}(a) = \pm 0.d_1d_2d_3 \ldots d_{n-1}d_n \times 10^p \]

Contoh: \( \pi = 0.31459265358 \ldots \times 10^0 \)

\[ fl_{\text{chop}}(\pi) = 0.3141592 \times 10^0 \) (6 digit mantis)

Galat = 0.00000065\ldots
Pembulatan ke digit terdekat (**in-rounding**)  

Misalkan \( a = \pm 0.d_1d_2d_3 ... d_n d_{n+1} ... \times 10^p \)  

\[
fl_{\text{round}}(a) = \pm 0.d_1d_2d_3...\hat{d}_n \times 10^p
\]

\[
\hat{d}_n = \begin{cases} 
  d_n & \text{, jika } d_{n+1} < 5 \\
  d_n + 1 & \text{, jika } d_{n+1} > 5 \\
  d_n & \text{, jika } d_{n+1} = 5 \text{ dan } n \text{ genap} \\
  d_n + 1 & \text{, jika } d_{n+1} = 5 \text{ dan } n \text{ ganjil}
\end{cases}
\]
• **Contoh 9:** \( a = 0.5682785715287 \times 10^{-4} : \)

  – di dalam komputer 7 digit dibulatkan menjadi
  \[ fl_{\text{round}}(a) = 0.5682786 \times 10^{-4} \]
  – di dalam komputer 8 digit dibulatkan menjadi
  \[ fl_{\text{round}}(a) = 0.56827857 \times 10^{-4} \]
  – di dalam komputer 6 digit dibulatkan menjadi
  \[ fl_{\text{round}}(a) = 0.568278 \times 10^{-4} \]
  – di dalam komputer 9 digit dibulatkan menjadi
  \[ fl_{\text{round}}(a) = 0.568278572 \times 10^{-4} \]
Aritmetika Bilangan Titik-Kambang

• **Kasus 1:** Penjumlahan (termasuk pengurangan) bilangan yang sangat kecil ke (atau dari) bilangan yang lebih besar menyebabkan timbulnya galat pembulatan.

**Contoh 10:** Misalkan digunakan komputer dengan mantis 4 digit (basis 10). Hitunglah

$$1.557 + 0.04381 = 0.1557 \times 10^1 + 0.4381 \times 10^{-1}$$
Penyelesaian:

\[ 0.1557 \times 10^1 = 0.1557 \times 10^1 \]
\[ 0.4381 \times 10^{-1} = 0.004381 \times 10^1 + \]

\[ = 0.160081 \times 10^1 \]

\textit{in-rounding} \rightarrow 0.1601 \times 10^1

\textit{chopping} \rightarrow 0.1600 \times 10^1

- Galat pembulatan = \[ |(0.160081 \times 10^1) - (0.1601 \times 10^1)| \]
  = 0.000019

- Galat pemenggalan = \[ |(0.160081 \times 10^1) - (0.1600 \times 10^1)| \]
  = 0.000081
• Tips: untuk menjumlahkan deret bilangan, selalu jumlahkan dari yang kecil-kecil lebih dahulu, baru menjumlahkan dengan bulangan yang lebih besar.

• Contoh: \( x = 1.0 + \sum_{i=1}^{10000} 0.00001 = 1.0 + \frac{0.00001 + 0.00001 + \ldots + 0.00001}{10000} \) 

Jumlahkan dulu 0.00001 sebanyak 1000 kali, baru jumlahkan dengan 1.0.
• **Kasus 2**: Pengurangan dua buah bilangan yang hampir sama besar (*nearly equal*).

• Bila dua bilangan titik-kambang dikurangkan, hasilnya mungkin mengandung nol pada posisi digit mantis yang paling berarti (posisi digit paling kiri).

• Keadaan ini dinamakan **kehilangan angka bena** (*loss of significance*). Baik pemenggalan maupun pembulatan ke digit terdekat menghasilkan jawaban yang sama.
• **Contoh 11**: Kurangi $0.56780 \times 10^5$ dengan $0.56430 \times 10^5$ (5 angka bena)

**Penyelesaian:**

\[
\begin{align*}
0.56780 \times 10^5 \\
0.56430 \times 10^5 - \\
0.00350 \times 10^5 & \rightarrow \text{normalisasi: } 0.350 \times 10^3 \\
\end{align*}
\]

(3 angka bena)

*in-rounding* → $0.350 \times 10^3$

*chopping* → $0.350 \times 10^3$

Hasil yang diperoleh hanya mempunyai 3 angka bena. Jadi kita kehilangan 2 buah angka bena
• **Contoh 12**: Kurangi 3.1415926536 dengan 3.1415957341 (11 angka bena).

**Penyelesaian:**

\[
3.1415926536 = 0.31415926536 \times 10^1 \\
3.1415957341 = 0.31415957341 \times 10^1 - \\
-0.30805 \times 10^{-5} \quad (5 \text{ angka bena})
\]

*in-rounding* → -0.30805 \times 10^{-5}  \\
*chopping* → -0.30805 \times 10^{-5}

Jadi, kita kehilangan 6 angka bena!.
• **Contoh 13.** Diberikan \( f(x) = x(\sqrt{x + 1} - \sqrt{x}) \) Hitunglah \( f(500) \) dengan menggunakan 6 angka bena dan pembulatan ke digit terdekat.

Penyelesaian:

\[
f(500) = 500(\sqrt{501} - \sqrt{500})
\]

\[
= 500(22.3830 - 22.3607)
\]

\[
= 500(0.0223)
\]

\[
= 11.15 \text{ (empat angka bena)}
\]

(solusi eksaknya adalah 11.174755300747198..)

Hasil yang tidak akurat ini disebabkan adanya operasi pengurangan dua bilangan yang hampir sama besar, yaitu 22.3830 - 22.3607.
Cara komputasi yang lebih baik:

\[ f(x) = x(\sqrt{x+1} - \sqrt{x}) \]

\[ = x(\sqrt{x+1} - \sqrt{x}) \frac{(\sqrt{x+1} + \sqrt{x})}{(\sqrt{x+1} + \sqrt{x})} \]

\[ = \frac{x[((\sqrt{x+1})^2 - (\sqrt{x})^2]}{(\sqrt{x+1} + \sqrt{x})} \]

\[ = \frac{x}{\sqrt{x+1} + \sqrt{x}} = p(x) \]

\[ p(500) = \frac{500}{\sqrt{501} + \sqrt{500}} = \frac{500}{22.3830 + 22.3607} = 11.1748 \]
• **Soal Latihan.** Carilah cara yang lebih baik untuk menghitung:

(i) \( f(x) = (x - \sin(x))/\tan(x) \) untuk \( x \) mendekati nol

(ii) \( f(x) = x - \sqrt{x^2 - a} \) untuk \( x \) yang jauh lebih besar dari \( a \)

(iii) \( f(x) = \cos^2(x) - \sin^2(x) \) untuk \( x \) di sekitar \( \pi/4 \)

(iv) \( f(x) = \log(x + 1) - \log(x) \) untuk \( x \) yang besar

(v) \( (1 + \alpha)^{1/2} - 1, \quad |\alpha| \leq 0.01 \) sampai enam angka bena

(vi) \( \sin(\alpha + x) - \sin(\alpha) \) untuk \( x \) yang kecil

(vii) \( (a + x)^n - a^n \) untuk \( x \) yang kecil

(viii) \( ((x^3 - 3x^2) + 3x) - 1 \) untuk \( x = 2.72 \)

(ix) \( \sqrt{1 + \cos x}/2 \) untuk \( x \approx \pi/4 \)
Kondisi Buruk (*Ill Conditioned*)

• Suatu persoalan dikatakan *berkondisi buruk* (*ill conditioned*) bila jawabannya sangat peka terhadap perubahan kecil data (misalnya perubahan kecil akibat pembulatan).

• Ciri-ciri: Bila kita mengubah sedikit data, maka jawabannya berubah sangat besar (drastis).

• Lawan dari berkondisi buruk adalah *berkondisi baik* (*well conditioned*).

• Suatu persoalan dikatakan berkondisi baik bila perubahan kecil data hanya mengakibatkan perubahan kecil pada jawabannya.
• Contoh: persoalan menghitung akar persamaan kuadrat $ax^2 + bx + c = 0$ dengan mengubah nilai $c$

(i) $x^2 - 4x + 3.999 = 0 \implies x_1 = 2.032$ dan $x_2 = 1.968$
(ii) $x^2 - 4x + 4.000 = 0 \implies x_1 = x_2 = 2.000$
(iii) $x^2 - 4x + 4.001 = 0 \implies$ akar-akarnya imajiner!

• Kesimpulan: persoalan akar-akar persamaan kuadrat di atas berkondisi buruk
• Kapankah akar persamaan \( f(x) = 0 \) berkondisi buruk?

• Misalkan \( f(x) \) diubah sebesar \( \varepsilon \) sehingga akarnya berubah sebesar \( h \):

\[
f(x + h) + \varepsilon = 0
\]

• Bila karena pengubahan \( \varepsilon \) yang sangat kecil mengakibatkan \( h \) menjadi besar, dikatakan persoalan mencari akar \( f(x) = 0 \) berkondisi buruk
Contoh lain: persoalan mencari mencari solusi sistem persamaan lanjar

(i) \[ x + y = 2 \]
\[ x + 0.9999y = 1.9999 \]
→ Solusi: \( x = y = 1.0000 \)

(ii) \[ x + y = 2 \]
\[ x + 0.9999y = 2.0010 \]
→ Solusi: \( x = 12, \ y = -10 \)

(iii) \[ x + y = 2 \]
\[ x + y = 1.9999 \]
→ Solusi: tidak ada

(iv) \[ x + y = 2 \]
\[ x + y = 2 \]
→ Solusi: tidak berhingga, yaitu disepanjang garis \( x + y = 2 \)
\( f(x) \)

\( f(x + h_1) + \varepsilon_1 \)

\( f(x + h_2) + \varepsilon_2 \)