Bab 8

Solusi Persamaan Diferensial Biasa

Penalaran adalah metode yang lambat dan berliku-liku dengan mana mereka
yang tidak mengetahui kebenaran menemukannya. Hati mempunyai penalaran sendiri
sedangkan penalaran itu tidak mengetahuinya.
(Blaise Pascal)

Persamaan diferensial adalah gabungan antara fungsi yang tidak diketahui secara eksplisit dan turunan (diferensial)-nya. Dalam kuliah Fisika anda tentu masih ingat persamaan gerak sistem pegas.

$$m\frac{d^2x}{dt^2} + c\frac{dx}{dt} + kx = 0 \qquad \text{(P.8.1)}$$

dengan m adalah massa pegas, k tetapan pegas, c koefisien redaman, dan x posisi sebuah titik pada pegas. Karena x adalah fungsi dari t, maka persamaan (P.8.1) ditulis juga sebagai

$$m\,x''(t) + cx'(t) + kx(t) = 0$$

atau dalam bentuk yang lebih ringkas,

$$mx'' + cx' + kx = 0.$$

Persamaan (P.8.1) mengandung fungsi $x(t)$ yang tidak diketahui rumus eksplisitnya, turunan pertamanya $x'(t)$, dan turunan kedua $x''(t)$.

Arti fisis diferensial adalah laju *perubahan* sebuah peubah terhadap peubah lain. Pada persaman (P.8.1), $x'(t)$ menyatakan laju perubahan posisi pegas x terhadap waktu t.

8.1 Kelompok Persamaan Diferensial

Persamaan diferensial dapat dibagi menjadi dua kelompok besar, yaitu persamaan diferensial biasa dan persamaan diferensial parsial.

1. Persamaan diferensial biasa (PDB) - *Ordinary Differential Equations* (ODE).

PDB adalah persamaan diferensial yang hanya mempunyai satu peubah bebas. Peubah bebas biasanya disimbolkan dengan x.

Contoh 8.1

Contoh-contoh persamaan berikut adalah persamaan diferensial biasa (PDB):

(i) $\dfrac{dy}{dx} = x + y$

(ii) $y' = x^2 + y^2$

(iii) $2\, dy/dx + x^2 y - y = 0$

(iv) $y'' + y' \cos x - 3y = \sin 2x$

(v) $2y''' - 23y' = 1 - y''$

Peubah bebas untuk contoh (i) sampai (v) adalah x, sedangkan peubah terikatnya adalah y, yang merupakan fungsi dari x, atau ditulis sebagai $y = g(x)$. ■

Berdasarkan turunan tertinggi yang terdapat di dalam persamaannya, PDB dapat lagi dikelompokkan menurut ordenya, yaitu:

a. PDB orde 1, yaitu PDB yang turunan tertingginya adalah turunan pertama. Contoh (i), (ii), dan (iii) di atas adalah PDB orde 1.

b. PDB orde 2, yaitu PDB yang turunan tertingginya adalah turunan kedua. Contoh (iv) adalah PDB orde dua.

c. PDB orde 3, yaitu PDB yang turunan tertingginya adalah turunan ketiga Contoh (v) di atas adalah PDB orde tiga.

d. dan seterusnya untuk PDB dengan orde yang lebih tinggi. PDB orde 2 ke atas dinamakan juga PDB orde lanjut.

2. Persamaan Diferensial Parsial (PDP) - *Partial Differential Equations* (PDE).

PDP adalah persamaan diferensial yang mempunyai lebih dari satu peubah bebas. Turunan fungsi terhadap setiap peubah bebas dilakukan secara parsial.

Contoh 8.2

Contoh-contoh persamaan berikut adalah persamaan diferensial parsial (PDP):

(i) $$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 6xye^{x+y}$$ (yang dalam hal ini, $u = g(x,y)$)

(ii) $$\frac{\partial u}{\partial t} = 3\sin(x+t) + \frac{\partial^2 u}{\partial x^2} + (1+x^2)\frac{\partial^2 u}{\partial y^2}$$ (yang dalam hal ini, $u = g(x, y, t)$)

■

Peubah bebas untuk contoh (i) adalah x dan y, sedangkan peubah terikatnya adalah u, yang merupakan fungsi dari x dan y, atau ditulis sebagai $u = g(x,y)$. Sedangkan peubah bebas untuk contoh (ii) adalah x, y, dan t, sedangkan peubah terikatnya adalah u, yang merupakan fungsi dari x, y, dan t, atau ditulis sebagai $u = g(x, y, t)$.

Buku ini hanya membahas metode-metode numerik untuk menyelesaikan PDB, khususnya PDB orde satu. Pada bagian akhir bab akan ditunjukkan bahwa PDB orde lanjut dapat dikembalikan bentuknya menjadi sistem PDB orde satu.

8.2 Terapan Persamaan Diferensial

Persamaan diferensial berperanan penting di alam, sebab kebanyakan fenomena alam dirumuskan dalam bentuk diferensial. Persamaan diferensial sering digunakan sebagai model matematika dalam bidang sains maupun dalam bidang rekayasa. Hukum-hukum dasar fisika, mekanika, listrik, dan termodinamika biasanya didasarkan pada perubahan sifat fisik dan keadaan sistem. Daripada menjelaskan keadaan sistem fisik secara langsung, hukum-hukum tersebut biasanya dinyatakan dalam perubahan *spasial* (koordinat) dan *temporal* (waktu) [CHA91]. Misalnya hukum Newton II menyatakan percepatan sebagai laju perubahan kecepatan setiap waktu, atau $a = dv/dt$, hukum termodinamika (Fluks panas = $-k\ \partial T/\partial x$, dengan k = konduktivitas panas, dan T = suhu), hukum Faraday (Beda tegangan = $L\ di/dt$, dengan L = induktansi, dan i = arus). Dengan mengintegralkan persamaan diferensial, dihasilkan fungsi matematika yang menjelaskan keadaan spasial dan temporal sebuah sistem, dinyatakan dalam percepatan, energi, massa, atau tegangan.

Persamaan (P.8.1) adalah terapan PDB dalam bidang fisika. Dalam bidang teknologi pangan, biologi, farmasi, dan teknik kimia, dikenal persamaan yang menyatakan bahwa laju pertumbuhan bakteri pada waktu t sebanding dengan jumlah bakteri (p) pada saat itu,

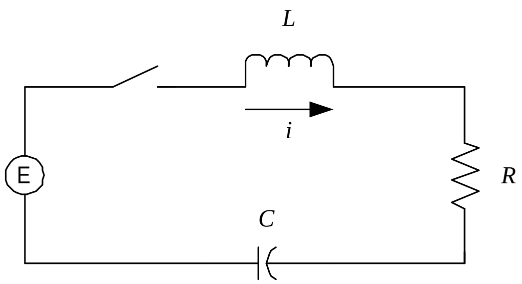
$$\frac{dp}{dt} = kp \tag{P.8.2}$$

dengan k adalah *tetapan kesebandingan*. Dalam bidang kelistrikan (elektro), para rekayasawan-nya tentu mengetahui benar hukum *Kirchoff* untuk sebuah rangkaian listrik sederhana *RLC* seperti pada Gambar 8.1.

Hukum tegangan Kirchoff menyatakan bahwa jumlah aljabar dari perubahan tegangan di sekeliling rangkaian tertutup adalah nol,

$$L\frac{di}{dt} + Ri = \frac{q}{C} - E(t) = 0 \tag{P.8.3}$$

dengan $L(di/dt)$ adalah perubahan tegangan antara induktor, L adalah induktansi kumparan (dalam *henry*), R adalah tahanan (dalam *ohm*), q adalah muatan pada kapasitor (dalam *coulomb*), C adalah kapasitansi (dalam *farad*), dan $E(t)$ adalah tegangan yang berubah terhadap waktu.



Gambar 8.1 Rangkaian listrik RLC

.

Beberapa persamaan diferensial dapat dicari solusinya secara analitis dengan teknik integral. Persamaan (P.8.2) misalnya, solusinya dapat ditemukan sebagai berikut:

$$dp/dt = kp$$

$$\Leftrightarrow \quad dp/p = k\,dt$$

$$\Leftrightarrow \quad \int dp/p = \int k\,dt$$

$$\Leftrightarrow \quad \ln(p) + C_1 = kt + C_2$$

$\Leftrightarrow \quad \ln(p) = kt + (C_2 - C_1) = kt + C \quad , \quad$ dengan $C = C_2 - C_1$

$\Leftrightarrow \quad p = e^{kt + C} = e^{kt} e^C = p_0 e^{kt} \quad , \quad$ dengan $p_0 = e^C$

Jadi, solusi analitiknya adalah

$$p(t) = p_0 e^{kt}$$

dengan p_0 adalah jumlah bakteri pada waktu $t = 0$. Bila $p_0 = p(0)$ diketahui, maka solusi yang unik dapat diperoleh. Dengan cara yang sama, solusi unik persamaan (P.8.3) juga dapat dihitung secara analitik bila diketahui besar arus pada $t = 0$ adalah $i(0) = 0$, yaitu

$$i(t) = (E/R)(1 - e^{-Rt/L})$$

Setelah persamaan $i(t)$ diperoleh, besar arus pada sembarang waktu t dapat dihitung.

Metode numerik untuk persamaan diferensial memainkan peranan sangat penting bagi rekayasawan, karena dalam prakteknya sebagian besar persamaan diferensial tidak dapat diselesaikan secara analitik. Metode numerik dipakai para rekayasawan untuk memperoleh solusi persaman diferensial. Bila metode analitik memberikan solusi persamaan diferensial dalam bentuk fungsi menerus, maka metode numerik memberikan solusi persamaan diferensial dalam bentuk farik. Upabab berikut ini membahas berbagai metode numerik untuk menghitung solusi PDB orde satu.

8.3 PDB Orde Satu

Bentuk baku PDB orde satu dengan nilai awal ditulis sebagai

$$y' = f(x, y)$$
$$\text{dengan nilai awal } y(x_0) = y \qquad \text{(P.8.4)}$$

Catatan: Kadang-kadang y' ditulis sebagai dy/dx. Jadi, $y' = dy/dx$.

PDB orde satu yang tidak mengikuti bentuk baku tersebut harus ditulis ulang menjadi bentuk persamaan (P.8.4), agar ia dapat diselesaikan secara numerik.

Contoh 8.3

Contoh-contoh persamaan berikut adalah persamaan diferensial biasa dan transformasinya ke dalam bentuk baku PDB orde 1:

(i) $2y' + xy = 100$; $y(0) = 1$

Bentuk baku: $y' = (100 - xy)/2$; $y(0) = 1$

(ii) $-xy' + 2y/x = y' - y$; $y(1) = -1$

Bentuk baku: $y' = \dfrac{2y/x + y}{1 + x}$; $y(1) = -1$ ■

Penyelesaian PDB secara numerik berarti menghitung nilai fungsi di $x_{r+1} = x_r + h$, dengan h adalah *ukuran langkah* (*step*) setiap lelaran. Pada metode analitik, nilai awal berfungsi untuk memperoleh solusi yang unik, sedangkan pada metode numerik nilai awal (*initial value*) pada persamaan (P.8.4) berfungsi untuk memulai lelaran. Terdapat beberapa metode numerik yang sering digunakan untuk menghitung solusi PDB, mulai dari metode yang paling dasar sampai dengan metode yang lebih teliti, yaitu

1. Metode Euler
2. Metode Heun
3. Metode Deret Taylor
4. Metode Runge-Kutta
5. Metode *predictor-corrector*.

8.4 Metode Euler

Diberikan PDB orde satu,

$$y' = dy/dx = f(x, y) \text{ dan nilai awal } y(x_0) = y_0$$

Misalkan

$$y_r = y(x_r)$$

adalah hampiran nilai y di x_r yang dihitung dengan metode Euler. Dalam hal ini

$$x_r = x_0 + rh, \qquad r = 0,1,2,\ldots\, n.$$

Metoda Euler diturunkan dengan cara menguraikan $y(x_{r+1})$ di sekitar x_r ke dalam deret Taylor:

$$y(x_{r+1}) = y(x_r) + \frac{(x_{r+1} - x_r)}{1!} y'(x_r) + \frac{(x_{r+1} - x_r)^2}{2!} y''(x_r) + \ ... \qquad \text{(P.8.5)}$$

Bila persamaan (P.8.5) dipotong sampai suku orde tiga, diperoleh

$$y(x_{r+1}) \approx y(x_r) + \frac{(x_{r+1} - x_r)}{1!} y'(x_r) + \frac{(x_{r+1} - x_r)^2}{2!} y''(t) \ , x_r < t < x_{r+1}$$

(P.8.6)

Berdasarkan persamaan (P.8.4),

$$y'(x_r) = f(x_r, y_r)$$

dan

$$x_{r+1} - x_r = h$$

maka persamaan (P.8.6) dapat ditulis menjadi

$$y(x_{r+1}) \approx y(x_r) + hf(x_r, y_r) + \frac{h^2}{2} y''(t) \qquad \text{(P.8.7)}$$

Dua suku pertama persamaan (P.8.7), yaitu

$$y(x_{r+1}) = y(x_r) + hf(x_r, y_r) \quad ; \qquad r = 0, 1, 2, ..., n \qquad \text{(P.8.8)}$$

menyatakan **metode Euler** atau **metode Euler-Cauchy**. Metode Euler disebut juga **metode orde-pertama**, karena pada persamaan (P.8.7) kita hanya mengambil sampai suku orde pertama saja. Untuk menyederhanakan penulisan, persamaan (P.8.8) dapat juga ditulis lebih singkat sebagai

$$y_{r+1} = y_r + hf_r$$

Selain dengan bantuan deret Taylor, metode Euler juga dapat diturunkan dengan cara yang berbeda. Sebagai contoh, misalkan kita menggunakan aturan segiempat untuk mengintegrasi-kan $f(x,y)$ pada persamaan diferensial

$$y' = f(x, y) \ ; \quad y(x_0) = y_0$$

Integrasikan kedua ruas dalam selang $[x_r, x_{r+1}]$:

$$\int_{x_r}^{x_{r+1}} y'(x)dx = \int_{x_r}^{x_{r+1}} f(x, y(x))dx$$

Gunakan aturan segiempat untuk mengintegrasikan ruas kanan, menghasilkan:

$$y(x_{r+1}) - y(x_r) = hf(x_r , y(x_r))$$

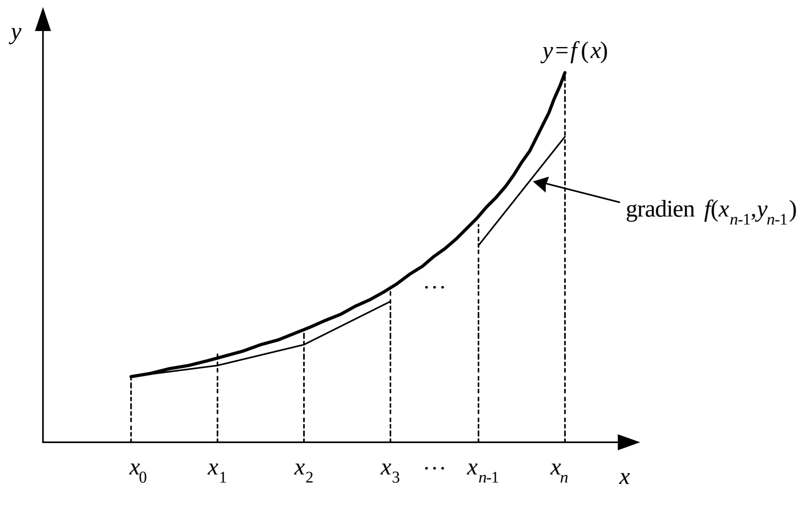
atau

$$y(x_{r+1}) = y(x_r) + hf(x_r, y_r)$$

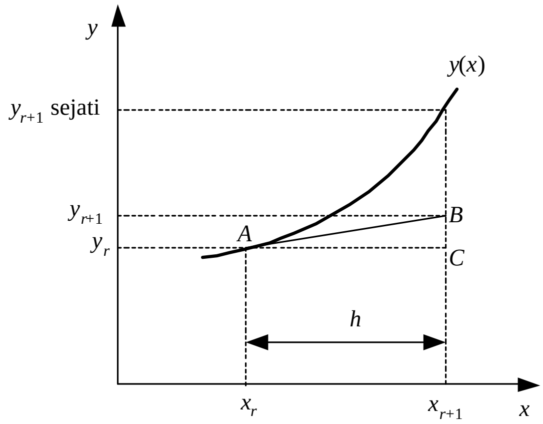
yang merupakan metode Euler.

8.4.1 Tafsiran Geometri Metode PDB

Pikirkanlah kembali bahwa $f(x,y)$ dalam persamaan diferensial menyatakan gradien garis singgung kurva di titik (x,y). Kita mulai menarik garis singgung dari titik (x_0, y_0) dengan gradien $f(x_0, y_0)$ dan berhenti di titik (x_1, y_1), dengan y_1 dihitung dari persamaan (P.8.8). Selanjutnya, dari titik (x_1, y_1) ditarik lagi garis dengan gradien $f(x_1, y_1)$ dan berhenti di titik (x_2, y_2), dengan y_2 dihitung dari persamaan (P.8.8). Proses ini kita ulang beberapa kali, misalnya sampai lelaran ke-n, sehingga hasilnya adalah garis patah-patah seperti yang ditunjukkan pada Gambar 8.2.



Gambar 8.2 Tafsiran geometri metode PDB



Gambar 8.3 Tafsiran geometri untuk penurunan metode Euler

Berdasarkan tafsiran geometri pada Gambar 8.2, kita juga dapat menurunkan metode Euler. Tinjau Gambar 8.3. Gradien (m) garis singgung di x_r adalah

$$m = y'(x_r) = f(x_r, y_r) = \frac{\Delta y}{\Delta x} = \frac{BC}{AB} = \frac{y_{r+1} - y_r}{h}$$

$$\Leftrightarrow \; y_{r+1} = y_r + hf(x_r, y_r)$$

yang tidak lain adalah persamaan metode Euler.

8.4.2 Analisis Galat Metode Euler

Meskipun metode Euler sederhana, tetapi ia mengandung dua macam galat, *yaitu* **galat pemotongan** (*truncation error*) dan **galat longgokan** (*cumulative error*). Galat pemotongan dapat langsung ditentukan dari persamaan (P.8.7), yaitu

$$E_p \approx \frac{1}{2} h^2 y''(t) = O(h^2) \qquad \text{(P.8.9)}$$

Galat pemotongan ini sebanding dengan kuadrat ukuran langkah h sehingga disebut juga **galat per langkah** (*error per step*) atau **galat lokal**. Semakin kecil nilai h (yang berarti semakin banyak langkah perhitungan), semakin kecil pula galat hasil perhitungannya. Perhatikan bahwa nilai pada setiap langkah (y_r) dipakai lagi pada langkah berikutnya (y_{r+1}). Galat solusi pada langkah ke-r adalah tumpukan galat dari langkah-langkah sebelumnya. Galat yang terkumpul pada akhir langkah ke-r ini disebut **galat longgokan** (*cumulative error*). Jika langkah dimulai dari $x_0 = a$

dan berakhir di $x_n = b$ maka total galat yang terkumpul pada solusi akhir (y_n) adalah

$$E_{total} = \sum_{r=1}^{n} (1/2)h^2 y''(t) = n\frac{h^2}{2} y''(t) = \frac{(b-a)}{2h} h^2 y''(t) = \frac{(b-a)}{2} y''(t)h \quad \text{(P.8.10)}$$

Galat longgokan total ini sebenarnya adalah

$$E_{total} = y(b)_{\text{sejati}} - y(x_n)_{\text{Euler}}$$

Persamaan (P.8.10) menyatakan bahwa galat longgokan sebanding dengan h. Ini berarti metode Euler memberikan hampiran solusi yang buruk, sehingga dalam praktek metode ini kurang disukai, namun metode ini membantu untuk memahami gagasan dasar metode penyelesaian PDB dengan orde yang lebih tinggi. Pengurangan h dapat meningkatkan ketelitian hasil, namun pengurangan h tanpa penggunaan bilangan berketelitian ganda tidaklah menguntungkan karena galat numerik meningkat disebabkan oleh galat pembulatan [NAK93].

Selain galat pemotongan, solusi PDB juga mengandung galat pembulatan, yang mempengaruhi ketelitian nilai y_1, y_2, …, semakin lama semakin buruk dengan meningkatnya n (baca kembali Bab 2 Deret Taylor dan Analisis Galat).

Program 8.1 Metode Euler

```
function y_Euler(x0, y0, b, h:real):real;
 {menghitung nilai y(b) pada PDB
       y'=f(x,y);   y(x0)=y0
  dengan metode Euler
 }
 var
    r, n: integer;
     x, y: real;
 begin
     n:=(b-x0)/h;      {jumlah langkah}
     y:=y0;            {nilai awal}
     x:=x0;
     for r:=1 to n do
    begin
        y:=y + h*f(x,y);       { hitung solusi y[xr] }
        x:=x + h;              { hitung titik berikutnya }
      end; {for}
     y_Euler:=y;       {y(b)}
 end;
```

Contoh 8.4

Diketahui PDB

$$dy/dx = x + y \text{ dan } y(0) = 1$$

Gunakan metode Euler untuk menghitung $y(0{,}10)$ dengan ukuran langkah $h = 0.05$ dan $h = 0.02$. Jumlah angka bena = 5. Diketahui solusi sejati PDB tersebut adalah $y(x) = e^x - x - 1$.

Penyelesaian:

(i) Diketahui

$a = x_0 = 0$
$b = 0.10$
$h = 0.05$

Dalam hal ini, $f(x, y) = x + y$, dan penerapan metode Euler pada PDB tersebut menjadi

$$y_{r+1} = y_r + 0.02(x_r + y_r)$$

Langkah-langkah:

$x_0 = 0 \quad \rightarrow \; y_0 = 1$
$x_1 = 0.05 \; \rightarrow \; y_1 = y_0 + 0.05(x_0 + y_0) = 1 + (0.05)(0 + 1) = 1.0050$
$x_2 = 0.10 \; \rightarrow \; y_2 = y_1 + 0.05(x_1 + y_1) = 1.0050 + (0.05)(0.05 + 1.0050) = 1.05775$

Jadi, $y(0.10) \approx 1.05775$.

(Bandingkan dengan nilai solusi sejatinya,

$$y(0.10) = e^{0.10} - 0.01 - 1 = 1.1103$$

sehingga galatnya adalah

galat = 1.1103 - 1.05775 = 0.05255)

(ii) Diketahui

$a = x_0 = 0$
$b = 0.10$
$h = 0.02$

Dalam hal ini, $f(x, y) = x + y$, dan penerapan metode Euler pada PDB tersebut menjadi

$$y_{r+1} = y_r + 0.02(x_r + y_r)$$

Langkah-langkah:

$x_0 = 0 \quad \rightarrow y_0 = 1$
$x_1 = 0.02 \rightarrow y_1 = y_0 + 0.02(x_0 + y_0) = 1 + (0.02)(0 + 1) = 1.0200$
$x_2 = 0.04 \quad \rightarrow y_2 = y_1 + 0.02(x_1 + y_1) = 1.0200 + (0.02)(0.02 + 1.0200) = 1.0408$
$x_3 = 0.06 \rightarrow y_3 = 1.0624$
$x_4 = 0.08 \rightarrow y_4 = 1.0848$
$x_5 = 0.10 \rightarrow y_5 = 1.1081$

Jadi, y (0,10) ≈ 1.1081

(Bandingkan dengan solusi sejatinya, y (0.10) = 1.1103, sehingga galatnya adalah

galat = 1.1103 - 1.1081 = 1.1081) ■

Contoh 8.4 memperlihatkan bahwa kita dapat mengurangi galat dengan memperbanyak langkah (memperkecil h).

8.5 Metode Heun (Perbaikan Metoda Euler)

Metode Euler mempunyai ketelitian yang rendah karena galatnya besar (sebanding dengan h). Buruknya galat ini dapat dikurangi dengan menggunakan metode Heun, yang merupakan perbaikan metode Euler (*modified Euler's method*). Pada metode Heun, solusi dari metode Euler dijadikan sebagai solusi perkiraan awal (*predictor*). Selanjutnya, solusi perkiraan awal ini diperbaiki dengan metode Heun (*corrector*).

Metode Heun diturunkan sebagai berikut: Pandang PDB orde satu

$$y'(x) = f(x, y(x))$$

Integrasikan kedua ruas persamaan dari x_r sampai x_{r+1}:

$$\begin{aligned} \int_{x_r}^{x_{r+1}} f(x, y(x))dx &= \int_{x_r}^{x_{r+1}} y'(x)dx \\ &= y(x_{r+1}) - y(x_r) \\ &= y_{r+1} - y_r \end{aligned}$$

Nyatakan y_{r+1} di ruas kiri dan suku-suku lainnya di ruas kanan:

$$y_{r+1} = y_r + \int_{x_r}^{x_{r+1}} f(x, y(x))dx \qquad \text{(P.8.11)}$$

Suku yang mengandung integral di ruas kanan, $\int_{x_r}^{x_{r+1}} f(x, y(x))dx$, dapat diselesaikan dengan kaidah trapesium menjadi

$$\int_{x_r}^{x_{r+1}} f(x, y(x))\, dx \approx \frac{h}{2}[f(x_r, y_r) + f(x_{r+1}, y_{r+1})] \qquad \text{(P.8.12)}$$

Sulihkan persamaan (P.8.12) ke dalam persamaan (P.8.11), menghasilkan persamaan

$$y_{r+1} = y_r + {}^h/_2\,[f(x_r, y_r) + f(x_{r+1}, y_{r+1})] \qquad \text{(P.8.13)}$$

yang merupakan **metode Heun**, atau **metode Euler-Cauchy yang diperbaiki**. Dalam persaman (P.8.13), suku ruas kanan mengandung y_{r+1}. Nilai y_{r+1} ini adalah solusi perkiraan awal (*predictor*) yang dihitung dengan metode Euler. Karena itu, persamaan (P.8.13) dapat ditulis sebagai

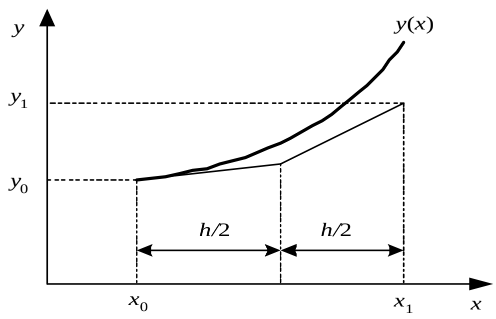
$$\begin{aligned} &\textit{Predictor}: \; y^{(0)}_{r+1} = y_r + hf(x_r, y_r) \\ &\textit{Corrector}: \; y_{r+1} = y_r + {}^h/_2\,[f(x_r, y_r) + f(x_{r+1}, y^{(0)}_{r+1})] \end{aligned} \qquad \text{(P.8.14)}$$

atau ditulis dalam satu kesatuan,

$$y_{r+1} = y_r + {}^h/_2[f(x_r,y_r) + f(x_{r+1}, y_r + hf(x_r, y_r)] \qquad \text{(P.8.15)}$$

8.5.1 Tafsiran Geometri Metode Heun

Metode ini mempunyai tafsiran geometri yang sederhana. Perhatikanlah bahwa dalam selang x_r sampai $x_r + ½\, h$ kita menghampiri solusi y dengan garis singgung melalui titik (x_r, y_r) dengan gradien $f(x_r, y_r)$, dan kemudian meneruskan garis singgung dengan gradien $f(x_{r+1}, y^{(0)}_{r+1})$ sampai x mencapai x_{r+1} [KRE88] (lihat Gambar 8.4 dengan r = 0).



Gambar 8.4 Tafsiran geometri metode Heun

8.5.2 Galat Metode Heun

Dari persamaan (P.8.14), suku $^{h}/_{2}$ $[f(x_r,y_r) + f(x_{r+1}, y^{(0)}_{r+1})]$ bersesuaian dengan aturan trapesium pada integrasi numerik. Dapat dibuktikan bahwa galat per langkah metode Heun sama dengan galat kaidah trapesium, yaitu

$$E_p \approx -\frac{h^3}{12}\, y''(t) \qquad , \; x_r < t < x_{r+1} \tag{P.8.15}$$

$$= O(h^3)$$

Bukti:

Misalkan,

Y_{r+1} adalah nilai y sejati di x_{r+1}
y_{r+1} adalah hampiran nilai y di x_{r+1}

Uraikan Y_{r+1} di sekitar x_r:

$$Y(x_{r+1}) = y(x_r) + \frac{(x_{r+1} - x_r)^2}{1!} y'(x_r) + \frac{(x_{r+1} - x_r)^2}{2!} y''(x_r) + \frac{(x_{r+1} - x_r)^3}{3!} y'''(x_r) + \dots$$

$$= y_r + hy'_r + \frac{h^2}{2} y_r'' + \frac{h^3}{6} y_r''' + \dots$$

Dengan menyatakan $y_r' = f(x_r, y_r) = f_r$, maka

$$Y_{r+1} = y_r + hf_r + \frac{h^2}{2} f_r' + \frac{h^3}{6} f_r''' + \dots \tag{P.8.16}$$

Dari persamaan (P.8.14),

$$y_{r+1} = y_r + {}^{h}/_{2}[f(x_r,y_r) + f(x_{r+1}, y^{(0)}_{r+1})]$$

uraikan $f(x_{r+1}, y^{(0)}_{r+1})$ dengan menggunakan deret Taylor di sekitar x_r:

$$f(x_{r+1}, y^{(0)}_{r+1}) = f(x_{r+1}, y_{r+1})$$

$$= f(x_r, y_r) + \frac{(x_{r+1} - x_r)^2}{2!} f'(x_r, y_r) + \frac{(x_{r+1} - x_r)^2}{2!} f''(x_r, y_r)$$

$$+ \ldots$$

$$= f_r + hf_r' + \frac{h^2}{2} f_r'' + \ldots$$

sehingga persamaan (P.8.14) dapat ditulis menjadi:

$$y_{r+1} = y_r + {}^h/_2 \, [f(x_r,y_r) + f(x_{r+1}, y^{(0)}{}_{r+1})]$$

$$= y_r + {}^h/_2 \, [f_r + f_r + hf_r' + ½ \, h^2 f_r'' + \ldots]$$

$$= y_r + hf_r + \frac{h^2}{2} f_r' + \frac{h^3}{4} f_r'' + \ldots \qquad \text{(P.8.17)}$$

Galat per langkah = nilai sejati - nilai hampiran

$$= Y_{r+1} - y_{r+1}$$

$$= \left(y_r + hf_r + \frac{h^2}{2} f_r' + \frac{h^3}{6} f_r''' + \ldots \right) - \left(y_r + hf_r + \frac{h^2}{2} f_r' + \frac{h^3}{4} f_r'' + \ldots \right)$$

$$= \frac{h^3}{6} f_r''' - \frac{h^3}{4} f_r''' + \ldots$$

$$= - \frac{h^3}{12} f_r''' + \ldots$$

$$= - \frac{h^3}{12} f_r'''(\mathrm{t}) \ , \quad x_r < t < x_{r+1}$$

$$= O(h^3)$$

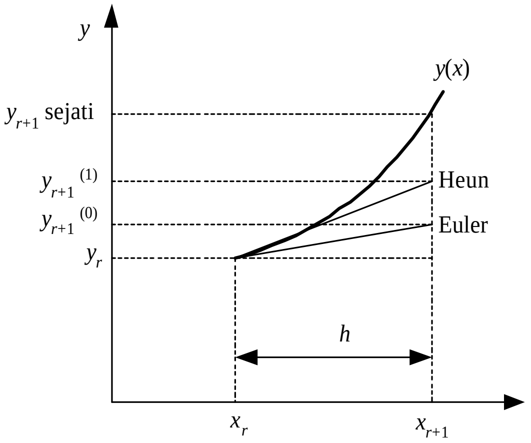
Galat longgokannya adalah,

$$E_L = \sum_{r=1}^{n} -\frac{1}{12} h^3 y''(t)$$

$$= - \frac{(b-a)}{12} h^2 y''(t) \qquad \text{(P.8.18)}$$

$$= O(h^2)$$

Jadi, galat longgokan metode Heun sebanding dengan h^2. Ini berarti solusi PDB dari metode Heun lebih baik daripada solusi dari metode Euler, namun jumlah komputasinya menjadi lebih banyak dibandingkan dengan metode Euler. Perbandingan metode Heun dengan metode Euler dilukiskan pada Gambar 8.5.



Gambar 8.5 Perbandingan metode Euler dengan metode Heun

Program 8.2 Metode Heun

```
function y_Heun(x0, y0, b, h:real):real;
{menghitung y(b) dengan metode Heun pada PDB
               y'=f(x,y);   y(x0)=y0
}
var
    r, n: integer;
    x, y, y_s : real;
begin
   n:=(b-x0)/h;        {jumlah langkah}
   y:=y0;              {nilai awal}
   x:=x0;
   for r:=1 to n do
    begin
      y_s:=y;                                 { y dari langkah r-1 }
      y:=y + h*f(x,y);                        { y(xr) dengan Euler }
      y:=y_s + h/2 * ((f(x,y_s) + f(x+h,y));  { y(xr) dengan Heun }
      x:=x+1;                                 { titik berikutnya}
    end;
  y_Heun:=y;
 end;
```

Contoh 8.5

Diketahui PDB

$$dy/dx = x + y \quad ; \; y(0) = 1$$

Hitung y (0.10) dengan metode Heun (h = 0.02)

Penyelesaian:

Diketahui

$$f(x, y) = x + y$$
$$a = x_0 = 0$$
$$b = 0.10$$
$$h = 0.02$$

maka n = (0.10 - 0)/0.02 = 5 (jumlah langkah)

Langkah-langkah:

$$x_1 = 0.02 \rightarrow y^{(0)}_1 = y_0 + hf(x_0, y_0)$$
$$= 1 + 0.02(0 + 1)$$
$$= 1.0200$$

$$y^{(1)}_1 = y_0 + (h/2)\,[f(x_0,y_0) + f(x_1,y^{(0)}_1)]$$
$$= 1 + (0.02/2)\,(0 + 1 + 0.02 + 1.0200)$$
$$= 1.0204$$

$$x_2 = 0.04 \rightarrow y^{(0)}_2 = y_1 + hf(x_1, y_1)$$
$$= 1.0204 + 0.02\,(0.02 + 1.0204)$$
$$= 1.0412$$

$$y^{(1)}_2 = y_1 + (h/2)\,[f(x_1,y_1) + f(x_2, y^{(0)}_2)]$$
$$= 1.0204 + (0.02/2)\,[0.02 + 1.0204 + 0.04 + 1.0412]$$
$$= 1.0416$$

…

$$x_5 = 0.10 \rightarrow y^{(0)}_5 = y_4 + hf(x_4, y_4)$$
$$y^{(1)}_5 = y_4 + (h/2)\,[f(x_4,y_4) + f(x_5,y^{(0)}_5)]$$
$$= 1.1104$$

Jadi, y (0.10) ≈ 1.1104.

Bandingkan:

Nilai sejati	: y(0.10) = 1.1103
Euler (Contoh 8.4)	: y(0.10) = 1.1081
Heun (Contoh 8.5)	: y(0.10) = 1.1104 → lebih baik dari Euler ■

8.5.3 Perluasan Metode Heun

Metode Heun dapat diperluas dengan meneruskan lelarannya sebagai berikut:

$$y^{(0)}_{r+1} = y_r + hf(x_r , y_r)$$

$$y^{(1)}_{r+1} = y_r + {}^{h}/_{2}\ [f(x_r , y_r) + f(x_{r+1}, y^{(0)}_{r+1})]$$

$$y^{(2)}_{r+1} = y_r + {}^{h}/_{2}\ [f(\mathrm{x}_{\mathrm{r}} , y_r) + f(x_{r+1}, y^{(1)}_{r+1})]$$

$$y^{(3)}_{r+1} = y_r + {}^{h}/_{2}\ [f(\mathrm{x}_{\mathrm{r}} , y_r) + f(x_{r+1}, y^{(2)}_{r+1})]$$

....

$$y^{(k+1)}_{r+1} = y_r + {}^{\mathrm{h}}/_{2}\ [f(x_r , y_r) + f(x_{r+1}, y^{(k)}_{r+1})]$$

Kondisi berhenti adalah bila

$$y^{(k)}_{r+1} - y^{(k\text{-}1)}_{r+1} < e$$

dengan ε adalah batas galat yang diinginkan. Jika lelarannya dilakukan satu kali (sampai dengan $y^{(1)}_{r+1}$ saja), maka lelarannya dinamakan **lelaran satu lemparan** (*one shot iteration*). Metoda Heun adalah lelaran satu lemparan.

Program 8.3: Perluasan Metode Heun

```
function y_Heun2(x0, y0, b, h:real):real;
{menghitung y(b) dengan perluasan metode Heun pada PDB
               y'=f(x,y);   y(x0)=y0
}
 const
     epsilon = 0.0000001;
 var
     r, n: integer;
     x, y, y_s, tampung : real;
 begin
    n:=(b-x0)/h;       {jumlah langkah}
    y:=y0;             {nilai awal}
    x:=x0;
    for r:=1 to n do
     begin
       y_s:=y;                {  y dari langkah r-1 }
       y:=y + h*f(x,y);       { y(xr) dengan Euler }
       repeat
          tampung:=y;
          y:=y_s + h/2 *((f(x, y_s)+ f(x+h, y));   {y(xr) dengan Heun}
       until ABS(y-tampung) < epsilon;
       x:=x+h;                { hitung titik berikutnya }
     end;
    y_Heun2:=y;
 end;
```

8.6 Metode Deret Taylor

Kita sudah melihat bahwa metode Euler diturunkan dengan menggunakan deret Taylor. Deret Taylor pada penurunan metode Euler dipotong sampai suku orde pertama sehingga solusinya kurang teliti. Kita dapat meningkatkan ketelitian dengan memotong deret sampaisuku yang lebih tinggi lagi. Metode deret Taylor adalah metode yang umum untuk menurunkan rumus-rumus solusi PDB. Metode Euler merupakan metode deret Taylor yang paling sederhana.

Diberikan PDB

$$y'(x) = f(x,y) \quad \text{dengan kondisi awal} \quad y(x_0) = y_0$$

Misalkan

$$y_{r+1} = y(x_{r+1}), \qquad r = 0,1,\ldots,n$$

adalah hampiran nilai y di x_{r+1}. Hampiran ini diperoleh dengan menguraikan y_{r+1} di sekitar x_r sebagai berikut:

$$y(x_{r+1}) = y(x_r) + \frac{(x_{r+1} - x_r)}{1!} y'(x_r) + \frac{(x_{r+1} - x_r)^2}{2!} y''(x_r) + \frac{(x_{r+1} - x_r)^3}{3!}$$
$$y'''(x_r) + \ldots \qquad + \frac{(x_{r+1} - x_r)^n}{n!} y^{(n)}(x_r)$$

atau

$$y(x_{r+1}) = y(x_r) + hy'(x_r) + \frac{h^2}{2} y''(x_r) + \frac{h^3}{6} y'''(x_r) + \ldots + \frac{h^{(n)} y^{(n)}}{n!} x_r \tag{P.8.19}$$

Persamaan (P.8.19) menyiratkan bahwa untuk menghitung hampiran nilai y_{r+1}, kita perlu menghitung $y'(x_r)$, $y''(x_r)$,…, $y^{(n)}(x_r)$, yang dapat dikerjakan dengan rumus

$$y^{(k)}(x) = P^{(k-1)} f(x, y) \tag{P.8.20}$$

yang dalam hal ini, P adalah operator turunan,

$$P = \left(\frac{\partial}{\partial x} + f \frac{\partial}{\partial y} \right) \tag{P.8.21}$$

Contoh 8.3

Diketahui PDB

$dy/dx = ½ x - ½ y$; $y(0) = 1$

Tentukan $y(0.50)$ dengan metode deret Taylor ($h = 0.25$).

Penyelesaian:

$x_0 = 0 \quad \rightarrow \quad y_0 = 1$

$x_1 = 0.25 \quad \rightarrow \quad y_1 = ?$

$$y(x_1) = y(x_0) + hy'(x_0) + \frac{h^2}{2} y''(x_0) + \frac{h^3}{6} y'''(x_0) + \ldots + \frac{h^{(n)}}{n!} y^{(n)} x_0 + \ldots$$

Misal kita hanya menghitung $y(x_1)$ sampai suku orde ke-4 saja.

$$\begin{aligned}
y'(x) &= ½ x - ½ y \\
y''(x) &= \frac{d}{dx} (½ x - ½ y) \\
&= ½ + f . (-1/2) \\
&= ½ - (½ x - ½ y) . ½ \\
&= ½ - ¼ x + ¼ y
\end{aligned}$$

$$\begin{aligned}
y'''(x) &= \frac{d}{dx} (½ - ¼ x + ¼ y) \\
&= -1/4 + f \; . 1/4 \\
&= -1/4 + (½ x - ½ y) . ¼ \\
&= -1/4 + x/8 - y/8
\end{aligned}$$

$$\begin{aligned}
y^{(4)}(x) &= \frac{d}{dx} (1/4 + 1/8 x - 1/8 y) \\
&= 1/8 + f . (-1/8) \\
&= 1/8 - (x/2 - y/2) . 1/8 \\
&= 1/8 - x/16 + y/16
\end{aligned}$$

Diperoleh:

$$\begin{aligned}
y(x_0) &= y(0) = 1 \\
y'(x_0) &= y'(0) = ½ \times 0 - ½ \times 1 = -1/2 \\
y''(x_0) &= y''(0) = ½ - ¼ \times 0 + ¼ \times 1 = 3/4 \\
y'''(x_0) &= y'''(0) = -1/4 + 1/8 \times 0 - 1/8 \times 1 = - 3/8 \\
y^{(4)}(x_0) &= y^{(4)}(0) = 1/8 - 1/16 \times 0 + 1/16 \times 1 = 3/16
\end{aligned}$$

sehingga

$$y(x_1) = 1 + 0.25\,(-1/2) + ((0.25)^2/2)\,(3/4) + ((0.25)^3/6)\,(-3/8) + ((0.25)^4/24)\,(3/16)$$
$$= 0.8974915$$

$x_2 = 0.50 \rightarrow y_2 = ?$

$$y(x_2) = y(x_1) + hy'(x_1) + \frac{h^2}{2}\, y''(x_1) + \frac{h^3}{6}\, y'''(x_1) + \ldots + \frac{h^{(n)}}{n!}\, y^{(n)}\, x_1 = \ldots$$

Diperoleh:

$$\begin{aligned}
y(x_1) &= 0.8974915 \\
y'(x_1) &= (1/2)(0.25) - (1/2)(0.8974915) = -0.3237458 \\
y''(x_1) &= ½ - (¼)\,(0.25) + (1/4)(0.8974915) = 0.6618729 \\
y'''(x_1) &= -1/4 + (1/8)(0.25) - (1/8)(0.8974915) = -0.3309634 \\
y^{(4)}(x_1) &= 1/8 - (1/16)(0.25) + (1/16)(.8974915) = 0.1654682
\end{aligned}$$

Sehingga,

$$\begin{aligned}
y_2 &= 0.8974915 + 0.25\,(-0.3237458) + (0.25^2/2)(0.6618729) \\
&\quad + (0.25^3/6)(-0.3309634) + (0.25^4/24)(0.1654682) \\
&= 0.8364037
\end{aligned}$$

Jadi, $y(0.50) \approx 0.8364037$
(Bandingkan dengan solusi sejati, $y(0.50) = 0.8364023$) ■

Galat Metode Deret Taylor

Galat perlangkah metode deret Taylor setelah pemotongan ke-n adalah

$$E_t \approx \frac{h^{(n+1)} f^{(n+1)}}{(n+1)!}(t), \quad x_0 < t < x_{r+1} \qquad \text{(P.8.22)}$$
$$= O(h^{n+1})$$

Pada Contoh 8.1, galat per langkahnya adalah

$$E_p \approx \frac{h^5}{5!}\, f^{(5)}(t), \qquad 0 < t < 0.50$$

Karena t tidak diketahui, kita hanya dapat menghitung batas atas dan batas bawah galat E_p dalam selang-buka (0, 50).

Galat longgokan total metode deret Taylor adalah:

$$E_L = \frac{h^{(n+1)}}{(n+1)!} f^{(n+1)}(t)$$

$$= \frac{b-a}{h} \frac{h^{(n+1)}}{(n+1)!} f^{(n+1)}(t)$$

$$= (b - a) \frac{f^{(n+1)}(t)}{(n+1)!} h^n \qquad \text{(P.8.23)}$$

$$= O(h^n)$$

8.7 Orde Metode PDB

Orde metode penyelesaian PDB menyatakan ukuran ketelitian solusinya. Makin tinggi orde metode, makin teliti solusinya. Orde metode PDB dapat ditentukan dari persamaan galat per langkah atau dari galat longgokannya.

1. Jika galat longgokan suatu metode PDB berbentuk Ch^p, C tetapan, maka metode tersebut dikatakan berorde p.

   Sebagai contoh,

   metode Euler $\rightarrow$ Galat longgokan = $\frac{(b-a)}{2} y''(t)\, h = Ch$, $C = \frac{(b-a)}{2} y''(t)$

   $= O(h) \rightarrow$ orde metode Euler = 1

   metode Heun $\rightarrow$ Galat longgokan = $\frac{-(b-a)}{12} y''(t)\, h^2 = Ch^2$, $C = \frac{-(b-a)}{12} y''(t)$

   $= O(h^2) \rightarrow$ orde metode Heun = 2

2. Jika galat per langkah suatu metoda PDB berbentuk Bh^{p+1}, B konstanta, maka metode tersebut dikatakan berorder p. Dengan kata lain, jika galat per langkah = $O(h^{p+1})$ maka galat longgokan = $O(h^p)$.

   Sebagai contoh,

   metode Euler $\rightarrow$ Galat per langkah = $\frac{1}{2} y''(t)\, h^2 = Bh^2$, $B = \frac{1}{2} y''(t)$

   $= O(h^2) \rightarrow$ orde metode Euler = 2-1=1

metode Heun $\rightarrow$ Galat per langkah $= {}^{-1}/_{12}\, y''(t)\, h^3 = Bh^3$, dengan $B = {}^{-1}/_{12}\, y''(t)$
$= O(h^3) \rightarrow$ orde metode Heun =3-1 = 2

Menentukan Galat per Langkah Metode PDB

Galat per langkah metode PDB diperoleh dengan bantuan deret Taylor. Kita sudah pernah menurunkan galat per langkah metode Heun dengan bantuan deret Taylor. Sekarang, prosedur untuk menentukan galat per langkah suatu metode PDB dapat ditulis sebagai berikut:

(1) Notasi nilai y hampiran di x_{r+1} adalah y_{r+1}
(2) Notasi nilai y sejati di x_{r+1} adalah Y_{r+1}
(3) Uraikan y_{r+1} di sekitar x_r
(4) Uraikan Y_{r+1} di sekitar x_r
(5) Galat per langkah adalah = (4) - (3)

Contoh 8.4

Hitung galat per langkah metode PDB

$y_{r+1} = y_r + hf_r$ (metode Euler)

dan tentukan orde metodenya.

Penyelesaian:

Hampiran : $y_{r+1} = y_r + hf_r$
Sejati : Y_{r+1}

Uraikan y_{r+1} hampiran di sekitar x_r:

Ruas kanan persamaan y_{r+1} sudah terdefinisi dalam x_r, jadi y_{r+1} tidak perlu diuraikan lagi.

Uraikan Y_{r+1} di sekitar x_r:

$$Y_{r+1} = Y(x_{r+1}) = y(x_r) + \frac{(x_{r+1} - x_r)}{1!}\, y'(x_r) + \frac{(x_{r+1} - x_r)^2}{2!}\, y''(x_r) + \ldots$$

$$= y_r + hy_r' + \frac{h^2}{2}\, y_r'' + \ldots = y_r + hf_r + \frac{h^2}{2}\, f_r' + \ldots$$

$$\text{Galat per langkah } E_p = Y_{r+1} - y_{r+1}$$
$$= \frac{h^2}{2} f_r' + \ldots$$
$$= \frac{h^2}{2} f'(t), \quad x_r < t < x_{r+1}$$
$$= O(h^2)$$

Orde metode = 2 - 1 = 1 ■

Contoh 8.5

Hitung galat per langkah metode PDB

$$y_{r+1} = y_r + {}^h/_{12}\,(23 f_r - 16 f_{r-1} + 5 f_{r-2})$$

dan tentukan orde metodenya.

Penyelesaian:

Hampiran : $y_{r+1} = y_r + {}^h/_{12}\,(23 f_r - 16 f_{r-1} + 5 f_{r-2})$
Sejati : Y_{r+1}

Uraikan y_{r+1} di sekitar x_r :

$$\begin{aligned}
23 f_r &= 23 f_r \\
-16 f_{r-1} &= -16\,(f_r - hf_r' + ½h^2 f_r'' - h^3/6\ f_r''' + \ldots) \\
5 f_{r-2} &= 5\,(f_r - 2hf_r' + 4h^2/2\, f_r'' - 8h^3/6\, f_r''' + \ldots) + \\
\hline
23 f_r - 16 f_{r-1} + 5 f_{r-2} &= 12 f_r + 6\,hf_r' + 2h^2 f_r'' - {}^{24}/_6\, h^3 f_r''' + \ldots
\end{aligned}$$

$$\begin{aligned}
y_{r+1} &= y_r + {}^h/_{12}\,(23 f_r - 16 f_{r-1} + 5 f_{r-2}) \\
&= y_r + {}^h/_{12}\,(12 f_r + 6hf_r' + 2h^2 f_r'' - {}^{24}/_6\, h^3 f_r''' + \ldots) \\
&= y_r + hf_r + ½\, h^2 f_r' + {}^1/_6\, h^3 f_r'' - {}^1/_3\, h^4 f_r''' + \ldots
\end{aligned}$$

Uraikan Y_{r+1} di sekitar x_r :

$$\begin{aligned}
Y_{r+1} &= y_r + hy_r' + h^2/2\, y_r'' + h^3/6\, y_r''' + h^4/24\, y^{(4)}_r + \ldots \\
&= y_r + hf_r + {}^1/_2\, h^2 f_r' + {}^1/_6\, h^3 f_r'' + {}^1/_{24}\, h^4 f_r''' + \ldots
\end{aligned}$$

$$\begin{aligned}
\text{Galat per langkah } E_p = Y_{r+1} - y_{r+1} &= {}^1/_{24}\, h^4 f_r''' + {}^1/_3\, h^4 f_r''' + \ldots \\
&= {}^9/_{24}\, h^4 f_r''' + \ldots \\
&= {}^9/_{24}\, h^4 f'''(t), \quad x_{r-2} < t < x_{r+1}
\end{aligned}$$

Orde metode = 4 - 1 = 3 ■

8.8 Metode Runge-Kutta

Penyelesaian PDB dengan metode deret Taylor tidak praktis karena metode tersebut membutuhkan perhitungan turunan $f(x, y)$. Lagipula, tidak semua fungsi mudah dihitung turunannya, terutama bagi fungsi yang bentuknya rumit. Semakin tinggi orde metode deret Taylor, semakin tinggi turunan fungsi yang harus dihitung. Karena pertimbangan ini, metode deret Taylor yang berorde tinggi pun tidak dapat dapat diterima dalam masalah praktek.

Metode Runge-Kutta adalah alternatif lain dari metode deret Taylor yang tidak membutuhkan perhitungan turunan. Metode ini berusaha mendapatkan derajat ketelitian yang lebih tinggi, dan sekaligus menghindarkan keperluan mencari turunan yang lebih tinggi dengan jalan mengevaluasi fungsi $f(x, y)$ pada titik terpilih dalam setiap selang langkah [CON80]. Metode Runge-Kutta adalah metode PDB yang paling popuper karena banyak dipakai dalam praktek.

Bentuk umum metoda Range-Kutta orde-n ialah:

$$y_{r+1} = y_r + a_1 k_1 + a_2 k_2 + ... + a_n k_n \qquad \text{(P.8.24)}$$

dengan a_1, a_2, ..., a_n adalah tetapan, dan

$$\begin{aligned}
k_1 &= hf\,(x_r\,, y_r) \\
k_2 &= hf\,(x_r + p_1 h, y_r + q_{11} k_1) \\
k_3 &= hf\,(x_r + p_2 h, y_r + q_{21} k_1 + q_{22} k_2) \\
&\ \ ... \\
k_n &= hf\,(x_r + p_{n-1} h, y_r + q_{n-1,1}\, k_1 + q_{n-1,2}\, k_2 + ... + q_{n-1,\,n-1}\, k_{n-1})
\end{aligned}$$

Nilai a_i, p_i, q_{ij} dipilih sedemikian rupa sehingga meminimumkan galat per langkah, dan persamaan (P.8.24) akan sama dengan metode deret Taylor dari orde setinggi mungkin..

Galat per langkah metode Runge-Kutta orde-n : $O(h^{n+1})$
Galat longgokan metode Runge-Kutta orde-n : $O(h^n)$
Orde metode = n

8.8.1 Metode Runge-Kutta Orde Satu

Metode Runge-Kutta orde satu berbentuk

$$k_1 = hf(x_r, y_r)$$
$$y_{r+1} = y_r + (a_1 k_1) \qquad \text{(P.8.25)}$$

Galat per langkah metode R-K orde satu adalah $O(h^2)$.
Galat longgokan metode R-K orde satu adalah $O(h)$.

Yang termasuk ke dalam metode Runge-Kutta orde satu ialah metode Euler:

$$k_1 = hf(x_r, y_r)$$
$$y_{r+1} = y_r + k_1 \qquad \text{(dalam hal ini } a_1 = 1\text{)}$$

8.8.2 Metode Runge-Kutta Orde Dua

Metode Runge-Kutta orde dua berbentuk

$$k_1 = hf(x_r, y_r)$$
$$k_2 = hf(x_r + p_1 h,\ y_r + q_{11} k_1)$$
$$y_{r+1} = y_r + (a_1 k_1 + a_2 k_2) \qquad \text{(P.8.26)}$$

Galat per langkah metode Runge-Kutta orde dua adalah $O(h^3)$.
Galat longgokan metode Runge-Kutta orde dua adalah $O(h^2)$.

Nilai a_1, a_2, p_1, dan q_{11} ditentukan sebagai berikut:

Misalkan

$$f_r = f(x_r, y_r)$$
$$f_x = \frac{\partial f(x_r, y_r)}{\partial x}, \text{ dan } f_y = \frac{\partial f(x_r, y_r)}{\partial y}$$

Uraikan k_2 ke dalam deret Taylor di sekitar (x_r, y_r) sampai suku orde satu saja:

$$\begin{aligned} k_2 &= hf(x_r + p_1 h, y_r + q_{11} k_1) \\ &= h(f + p_1 h f_x + q_{11} k_1 f_y) \\ &= h(f + p_1 h f_x + q_{11} hf f_y) \\ &= h(f + h(p_1 f_x + q_{11} ff_y)) \end{aligned}$$

Sedangkan k_1 tidak perlu diuraikan karena sudah berada dalam bentuk (x_r, y_r).

Jadi,

$$
\begin{aligned}
y_{r+1} &= y_r + a_1k_1 + a_2k_2 \\
&= y_r + a_1hf_r + a_2hf + a_2h^2(p_1f_x + q_{11}\, ff_y) \\
&= y_r + (a_1 + a_2)\, hf + a_2h^2(p_1f_x + q_{11}\, ff_y) \qquad \text{(P.8.27)}
\end{aligned}
$$

Uraikan Y_{+1} sejati di sekitar x_r sampai suku orde dua saja:

$$Y_{r+1} = y_r + hy_r' + {}^1/_2\, h^2y_r'' \qquad \text{(P.8.28)}$$

Mengingat

$$y_r' = f(x_r,y_r) = f_r$$

dan

$$
\begin{aligned}
y_r'' &= f\,'(x_r, y_r) = \\
&= \frac{df(x_r\,, y_r)}{dx} \\
&= \frac{\partial f}{\partial x}\ \frac{dx}{dx} + \frac{\partial f}{\partial y}\ \frac{dy}{dx} \\
&= f_x + f_y\, f_r \\
&= f_x + ff_y
\end{aligned}
$$

maka persamaan (P.8.28) menjadi

$$y_{r+1} = y_r + hf + {}^1/_2\, h^2(f_x + f_r\, f_y) \qquad \text{(P.8.29)}$$

Galat per langkah metode adalah

$$
\begin{aligned}
E_p &= \text{(P.8.29) - (P.8.27)}: \\
&= \{ y_r + hf + {}^1/_2\, h^2(f_x + f_r\, f_y) \} - \{ y_r + (a_1 + a_2)\, hf_r + a_2h^2(p_1\, f_x + q_{11}\, f_r\, f_y) \} \\
&= \{ hf + {}^1/_2\, h^2(f_x + ff_y) \} - \{ (a_1 + a_2)\, hf + a_2h^2(p_1\, f_x + q_{11}\, ff_y) \}
\end{aligned}
$$

Dengan membuat galat per langkah $E_p = 0$,

$$0 = \{ hf_r + {}^1/_2\, h^2(f_x + f_r\, f_y) \} - \{ (a_1 + a_2)\, hf_r + a_2h^2(p_1\, f_x + q_{11}\, f_r\, f_y) \}$$

atau

$$hf_r + {}^1/_2\, h^2(f_x + f_r\, f_y) = (a_1 + a_2)\, hf_r + a_2h^2(p_1 f_x + q_{11} f_r\, f_y) \qquad \text{(P.8.30)}$$

Agar ruas kiri dan ruas kanannya sama, haruslah

$$a_1 + a_2 = 1$$
$$a_2\, p_1 = 1/2$$
$$a_2\, q_{11} = 1/2$$

Karena sistem persamaan di atas terdiri atas tiga persamaan dengan empat peubah yang tidak diketahui, maka solusinya tidak unik, dengan kata lain, solusinya banyak. Solusi yang unik hanya dapat diperoleh dengan memberikan sebuah peubah dengan sebuah harga. Misalkan ditentukan nilai $a_2 = t$, $t \in R$, maka

$$a_1 = 1 - a_2 = 1 - t$$

$$p_1 = \frac{1}{2a_2} = \frac{1}{2t}$$

$$q_{11} = \frac{1}{2a_2} = \frac{1}{2t}$$

Karena kita dapat memberikan sembarang nilai t, berarti metode Runge-Kutta Orde dua tidak terhingga banyaknya.

Contoh metode Runge-Kutta orde dua adalah metode Heun, yang dalam hal ini

$$a_2 = 1/2,$$
$$a_1 = 1/2,$$
$$p_1 = q_{11} = 1$$

Dalam bentuk Runge-Kutta orde 2, metode Heun dapat ditulis sebagai

$$k_1 = hf(x_r, y_r)$$

$$k_2 = hf(x_r + h, y_r + k_1)$$

$$y_{r+1} = y_r + {}^1/_2\,(k_1 + k_2)$$

Program Heun sudah pernah kita tulis (Program 8.2). Sekarang program tersebut kita tulis lagi dalam bentuk Runge-Kutta orde 2 menjadi Program 8.4 berikut ini.

Program 8.4 Metode Heun

```
function y_Heun(x0, y0, b, h:real):real;
{menghitung y(b) dengan metode Heun pada PDB
               y'=f(x,y);   y(x0)=y0
}
var
   r, n: integer;
   x, y, y_s, x_s : real;
begin
   n:=(b - x0)/h; {jumlah langkah}
   y:=y0;         {nilai awal}
   x:=x0;
  for r:=1 to n do
   begin
      k1:=h*f(x,y);
      k2:=h*f(x+h, y+k1);
      y:=y + (k1 + k2)/2;
      x:=x+1;
   end;
  y_Heun:=y;
 end;
```

Contoh metode Runge-Kutta orde dua lainnya ialah **metode Ralston**, yang dalam hal ini

$$a_2 = 2/3$$
$$a_1 = 1/3,$$
$$p_1 = q_{11} = 3/4$$

sehingga metode Ralston dapat ditulis dalam bentuk Runge-Kutta orde dua sebagai

$$k_1 = hf(x_r, y_r)$$
$$k_2 = hf(x_r + {}^3/_4\, h, y_r + {}^3/_4\, k_1)$$
$$y_{r+1} = y_r + ({}^1/_3\, k_1 + {}^2/_3\, k_2) \qquad \text{(P.8.31)}$$

Sepintas, metode Runge-Kutta tampaknya rumit, tapi sebenarnya metode Runge-Kutta mudah diprogram. Dengan perhitungan tangan, seringnya menghitung $f(x, y)$ merupakan pekerjaan yang melelahkan. Tetapi dengan komputer, hal ini tidak menjadi masalah.

8.8.3 Metode Runge-Kutta Orde Tiga

Metode Runge-Kutta yang terkenal dan banyak dipakai dalam praktek adalah **metode Runge-Kutta orde tiga** dan **metode Runge-Kutta orde empat**. Kedua metode tersebut terkenal karena tingkat ketelitian solusinya tinggi (dibandingkan metode Runge-Kutta orde sebelumnya, mudah diprogram, dan stabil (akan dijelaskan kemudian).

Metode Runge-Kutta orde tiga berbentuk:

$$\begin{aligned}
k_1 &= hf(x_r, y_r) \\
k_2 &= hf(x_r + {}^1/_2\, h, y_r + {}^1/_2\, k_1) \\
k_3 &= hf(x_r + h, y_r - k_1 + 2k_2) \\
y_{r+1} &= y_r + {}^1/_6\,(k_1 + 4k_2 + k_3)
\end{aligned} \tag{P.8.32}$$

Galat per langkah metode R-K orde tiga adalah $O(h^4)$.
Galat longgokan metode R-K orde tiga adalah $O(h^3)$.

Program 8.5 Metode Runge-Kutta Orde 3

```
function y_RK3(x0, y0, b, h:real):real;
{menghitung y(b) dengan metode Runge-Kutta orde tiga pada PDB
       y'=f(x,y);   y(x0)=y0
}
 var
    r, n: integer;
    x, y, k1, k2, k3: real;
 begin
    n:=(b - x0)/h;  {jumlah langkah}
    y:=y0;          {nilai awal}
    x:=x0;
    for r:=1 to n do

     begin
       k1:=h*f(x, y);
       k2:=h*f(x + h/2, y + k1/2);
       k3:=h*f(x + h, y - k1 + 2*k2);
       y:=y + (k1 + 4*k2 + k3)/6            { nilai y(xr) }
       x:=x+h;                              { titik berikutnya}
     end;
   y_RK3:=y;
 end;
```

8.8.4 Metode Runge-Kutta Orde Empat

Metode Runge-Kutta orde empat adalah

$$
\begin{aligned}
k_1 &= hf\,(x_r, y_r) \\
k_2 &= hf\,(x_r + {}^1/_2\, h,\, y_r + {}^1/_2\, k_1) \\
k_3 &= hf\,(x_r + {}^1/_2\, h,\, y_r + {}^1/_2\, k_2) \\
k_4 &= hf\,(x_r + h,\, y_r + k_3) \\
y_{r+1} &= y_r + {}^1/_6\,(k_1 + 2k_2 + 2k_3 + k_4)
\end{aligned} \tag{P.8.33}
$$

Galat per langkah metode Runge-Kutta orde empat adalah $O(h^3)$.
Galat longgokan metode Runge-Kutta orde empat adalah $O(h^2)$.

Program 8.6 Metode Runge-Kutta Orde 4

```
function y_RK4(x0, y0, b, h:real):real;
 {menghitung y(b) dengan metode Runge-Kutta orde empat pada PDB
        y'=f(x,y);   y(x0)=y0
 }
 var
    r, n: integer;
    x, y, k1, k2, k3, k4: real;
 begin
    n:=(b - x0)/h;       {jumlah langkah}
    y:=y0;               {nilai awal}
    x:=x0;
    for r:=1 to n do
     begin
       k1:=h*f(x, y);
       k2:=h*f(x + h/2, y + k1/2);
       k3:=h*f(x + h/2, y + k2/2);
       k4:=h*f(x + h, y + k3);
       y:=y + (k1 + 2*k2 + 2*k3 + k4)/6      { nilai y(xr) }
       x:=x+h;                                { titik berikutnya}
     end;
    y_RK4:=y;
 end;
```

Contoh 8.6

Diketahui PDB

$$\frac{dy}{dx} = 1 + y^2 \quad ; \; y(0) = 0$$

Tentukan $y(0.20)$ dengan metode Runge-Kutta orde tiga. Gunakan ukuran langkah $h = 0.10$.

Penyelesaian:

Diketahui

$$\begin{aligned} a &= x_0 = 0 \\ b &= 0.20 \\ h &= 0.10 \end{aligned}$$

maka $n = (0.20 - 0)/0.10 = 2$ (jumlah langkah)

Langkah:

$$\begin{aligned} x_0 &= 0 \rightarrow y_0 = 0 \\ x_1 &= 0.10 \rightarrow y_1 = ? \end{aligned}$$

$$\begin{aligned} k_1 &= hf(x_0, y_0) = (0.10)\,(1 + 0^2) = 0.10 \\ k_2 &= hf(x_0 + {}^1\!/_2\, h,\; y_0 + {}^1\!/_2\, k_1) = (0.10)\,(1 + 0.05^2) = 0.10025 \\ k_3 &= hf(x_0 + h,\; y_0 - k_1 + 2k_2) = (0.10)\,(1 + 0.1005^2) = 0.10101 \\ y_1 &= y_0 + {}^1\!/_6\,(k_1 + 4k_2 + k_3) \\ &= 0 + {}^1\!/_6\,(0.10 + 4 \times 0.10025 + 0.10101) = 0.10034 \end{aligned}$$

$$\begin{aligned} x_2 &= 0.20 \rightarrow y_2 = ? \\ k_1 &= hf(x_1, y_1) = (0.10)(1 + 0.10034^2) = 0.10101 \\ k_2 &= hf(x_1 + {}^1\!/_2\, h, y_1 + {}^1\!/_2\, k_1) = (0.10)(1 + 0.150845^2) = 0.10228 \\ k_3 &= hf(x_1 + h, y_1 - k_1 + 2k_2) = (0.10)\,(1 + 0.20389^2) = 0.10416 \\ y_2 &= y_1 + {}^1\!/_6\,(k_1 + 4k_2 + k_3) \\ &= 0.10034 + {}^1\!/_6\,(0.10101 + 4 \times 0.10228 + 0.10416) \\ &= 0.20272 \end{aligned}$$

Jadi, $y(0.20) \approx 0.20272$.
Nilai sejati $\rightarrow y(0.20) = 0.20271$. ■

Metode Runge-Kutta orde yang lebih tinggi tentu memberikan solusi yang semakin teliti. Tetapi ketelitian ini harus dibayar dengan jumlah komputasi yang semakin banyak. Jadi ada timbal-balik (*trade-off*) dalam memilih suatu metode Runge-Kutta.

8.9 Ekstrapolasi Richardson

Ekstrapolasi Richardson dapat diterapkan untuk memperbaiki solusi PDB dan memperkirakan galatnya, asal kita mengetahui orde metode PDB. Mula-mula solusi PDB dihitung dengan ukuran langkah h. Kemudian solusinya dihitung lagi tetapi dengan ukuran langkah $2h$. Maka, solusi yang lebih baik adalah

$$y(x) = y(x; h) + \frac{1}{2^p - 1} [\, y(x; h) - y(x; 2h)] \tag{P.8.34}$$

yang dalam hal ini,

$y(x; h)$ = solusi PDB di x dengan ukuran langkah h
$y(x; 2h)$ = solusi PDB di x dengan ukuran langkah $2h$
$y(x)$ = solusi PDB yang lebih baik.
p = orde metode PDB yang digunakan

taksiran galatnya adalah

$$e = \frac{1}{2^p - 1} [\, y(x; h) - y(x; 2h)] \tag{P.8.35}$$

Bila kita tidak mengetahui p, maka nilai perkiraan ketiga, $y(x; 4h)$ memungkinkan kita menggunakan ekstrapolasi Aitken sebagai pengganti ekstrapolasi Richardson. Lihat kembali Bab Integrasi Numerik.

8.10 Metode Banyak-Langkah

Sampai sejauh ini kita telah mengenal metode Euler, metode Heun, metode deret Taylor, dan metode Runge-Kutta. Semua metode tersebut dikelompokkan ke dalam **metode satu-langkah** (*one-step*), sebab untuk menaksir nilai $y(x_{r+1})$ dibutuhkan satu buah taksiran nilai sebelumnya, $y(x_r)$.

Kelompok metode PDB yang lain ialah **metode banyak-langkah** (*multi-step*). Pada metode banyak-langkah, perkiraan nilai $y(x_{r+1})$ membutuhkan beberapa taksiran nilai sebelumnya, $y(x_r)$, $y(x_{r-1})$, $y(x_{r-2})$, ... Yang termasuk ke dalam metode banyak-langkah adalah metode *predictor-corrector*. Metode Heun adalah metode *predictor-corrector*, namun metode Heun bukanlah metode banyak-langkah, sebab taksiran nilai $y(x_{r+1})$ hanya didasarkan pada taksiran $y(x_r)$.

Tujuan utama metode banyak-langkah adalah menggunakan informasi dari beberapa titik sebelumnya, y_r, y_{r-1}, y_{r-2} , ..., untuk menghitung taksiran nilai y_{r+1} yang lebih baik.

Beberapa metode *predictor-corrector* (P-C) yang termasuk ke dalam metode banyak-langkah. Pada metode) P-C, kita menaksir nilai y_{r+1} dari y_r, y_{r-1}, y_{r-2} , ..., dengan persamaan *predictor*, dan kemudian menggunakan persamaan *corrector* untuk menghitung nilai y_{r+1} yang lebih baik (*improve*).

predictor	:	Menaksir y_{r+1} dari y_r, y_{r-1}, y_{r-2},...
corrector	:	Memperbaiki nilai y_{r+1} dari *predictor*

Metode P-C yang banyak ditulis dalam literatur dan kita bahas di sini adalah:

1. Metode Adams-Bashforth-Moulton.
2. Metode Milne-Simpson
3. Metode Hamming

8.10.1 Metode Adams-Bashforth-Moulton

Tinjau PDB orde satu

$$y'(x) = f(x, y(x))$$

Integrasikan kedua ruas persamaan dari x_r sampai x_{r+1}:

$$\begin{aligned}\int_{x_r}^{x_{r+1}} f(x, y(x))dx &= \int_{x_r}^{x_{r+1}} y'(x)dx \\ &= y(x)\Big|_{xr}^{xr+1} \\ &= y(x_{r+1}) - y(x_r) \\ &= y_{r+1} - y_r\end{aligned}$$

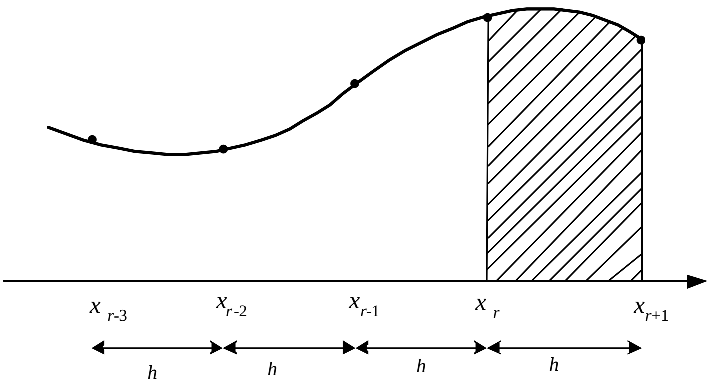
Nyatakan y_{r+1} di ruas kiri persamaan dan suku lainnya di ruas kanan:

$$y_{r+1} = y_r + \int_{x_r}^{x_{r+1}} f(x, y(x))dx \qquad \text{(P.8.34)}$$

Persaman (P.8.34) ini adalah teorema dasar kalkulus (lihat Bab 2, Integral), yang merupakan dasar penurunan persamaan *predictor* dan persamaan *corrector*.

Persamaan *Predictor* [MAT93]

Persamaan *predictor* diperoleh dengan menghampiri fungsi $f(x, y(x))$ ke dalam polinom interpolasi derajat tiga. Untuk itu, diperlukan empat buah titik yang berjarak sama, yaitu: (x_{r-3}, f_{r-3}) , (x_{r-2}, f_{r-2}) , (x_{r-1}, f_{r-1}), (x_r, f_r). Perhatikan Gambar 8.6.



Gambar 8.6 Pembentukan persaman *predictor*

Dari empat buah titik tersebut, bentuklah polinom interpolasi Lagrange derajat tiga:

$$
\begin{aligned}
f(x, y(x)) \approx\; & \frac{(x-x_{r-2})(x-x_{r-1})(x-x_r)\; f_{r-3}}{(x_{r-3}-x_{r-2})(x_{r-3}-x_{r-1})(x_{r-3}-x_r)} + \\
& \frac{(x-x_{r-3})(x-x_{r-1})(x-x_r)\; f_{r-2}}{(x_{r-2}-x_{r-3})(x_{r-2}-x_{r-1})(x_{r-2}-x_r)} + \\
& \frac{(x-x_{r-3})(x-x_{r-2})(x-x_r)\; f_{r-1}}{(x_{r-1}-x_{r-3})(x_{r-1}-x_{r-2})(x_{r-1}-x_r)} + \\
& \frac{(x-x_{r-3})(x-x_{r-2})(x-x_{r-1})\; f_r}{(x_r-x_{r-3})(x_r-x_{r-2})(x_r-x_{r-1})}
\end{aligned}
$$

$$\approx \frac{1}{-6h^3}(x - x_{r-2})(x - x_{r-1})(x - x_r)\, f_{r-3} + \frac{1}{2h^3}(x - x_{r-3})(x - x_{r-1})(x - x_r)\, f_{r-2}$$

$$- \frac{1}{2h^3}(x - x_{r-3})(x - x_{r-2})(x - x_r)\, f_{r-1} + \frac{1}{2h^3}(x - x_{r-3})(x - x_{r-2})(x - x_{r-1})\, f_r$$

(P.8.35)

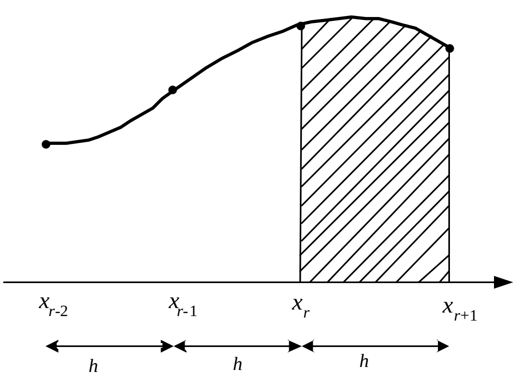
Sulihkan (P.8.35) ke dalam persamaan (P.8.34). Hasil integrasi persamaan (P.8.34) memberikan:

$$y^*_{r+1} = y_r + \frac{h}{24}(-9f_{r-3} + 37\, f_{r-2} - 59\, f_{r-1} + 55\, f_r) \qquad \text{(P.8.36)}$$

yang merupakan persamaan *predictor*.

Persamaan *Corrector* [MAT93]

Persamaan *corrector* dibentuk dengan cara yang sama seperti pada persamaan *predictor*. Tetapi, titik-titik yang diperlukan untuk pembentukan polinom interpolasi (Gambar 8.7) ialah (x_{r-2}, f_{r-2}), (x_{r-1}, f_{r-1}), (x_r, f_r), dan titik baru $(x_{r+1}, f^*_{r+1}) = (x_{r+1}, f(x_{r+1}, y^*_{r+1}))$



Gambar 8.7 Pembentukan persaman *corrector*

Dari empat buah titik tersebut, bentuklah polinom interpolasi Lagrange derajat tiga. Kemudian, integrasikan polinom interpolasi tersebut dalam selang $[x_r\ ,\ x_{r+1}]$, untuk memberikan

$$y_{r+1} = y_r + \frac{h}{24} (f_{r-2} - 5 f_{r-1} + 19 f_r + 9f^*_{r+1}) \qquad \text{(P.8.37)}$$

yang merupakan persamaan *corrector*.

Jadi, metode Adams-Bashforth-Moulton dapat diringkas sebagai berikut:

$$\textit{predictor} : y^*_{r+1} = y_r + {}^h/_{24} (-9f_{r-3} + 37 f_{r-2} -59 f_{r-1} + 55 f_r) \qquad \text{(P.8.38)}$$

$$\textit{corrector} : y_{r+1} = y_r + {}^h/_{24} (f_{r-2} - 5 f_{r-1} + 19 f_r + 9f^*_{r+1}) \qquad \text{(P.8.39)}$$

Galat per langkah metode Adams-Bashforth-Moulton adalah dalam orde $O(h^5)$, yaitu:

$$\textit{predictor} : \quad E_p = Y_{r+1} - y^*_{r+1} \approx {}^{251}/_{720}\ h^5 y^{(5)}(t)\ ,\ x_{r-3} < t < x_{r+1}$$

$$\textit{corrector} : \quad E_p = Y_{r+1} - y_{r+1} \approx {}^{-19}/_{720}\ h^5 y^{(5)}(t)\ ,\ x_{r-3} < t < x_{r+1}$$

dan galat longgokan adalah dalam orde $O(h^4)$. Karena itu, metode Adams-Bashforth-Moulton di atas dinamakan juga **metode Adams-Bashforth-Moulton orde-4**.

Metode yang lebih rendah adalah **metode Adams-Bashforth-Moulton orde-3**:

$$\textit{predictor} : y^*_{r+1} = y_r + {}^h/_{12} (23 f_r - 16 f_{r-1} + 5f_{r-2})$$

$$\textit{corrector} : y_{r+1} = y_r + {}^h/_{12}) (5 f^*_{r+1} + 18 f_r - f_{r-1})$$

Pada waktu penurunan persamaan *predictor* Adams-Bashforth-Mouton orde-3 ini, polinom interpolasinya memerlukan tiga buah titik, yaitu (x_{r-2}, f_{r-2}), (x_{r-1}, f_{r-1}), (x_r, f_r), sedangkan pada waktu penurunan persamaan *predictor*, polinom interpolasinya memerlukan titik-titik (x_{r-1}, f_{r-1}), (x_r, f_r), (x_{r+1}, f^*_{r+1}).

Galat per langkahnya adalah dalam orde $O(h^4)$, yaitu:

$$\textit{predictor} : E_p = Y_{r+1} - y^*_{r+1} \approx {}^9/_{24}\ h^4 y''(t) \quad ,\ x_{r-2} < t < x_{r+1}$$

$$\textit{corrector} : E_p = Y_{r+1} - y_{r+1} \approx {}^{-1}/_{24}\ h^4 y''(t) \quad ,\ x_{r-2} < t < x_{r+1}$$

dan galat longgokan adalah dalam orde $O(h^3)$.

Cara menurunkan persamaan galat metode *predictor-corrector* sama seperti cara yang sudah dijelaskan sebelumnya. Misalnya kita akan menurunkan persamaan galat metode Adams-Bashforth-Moulton orde-3 sebagai berikut:

Uraikan persamaan predictor, *corrector* dan y_{r+1} sejati di sekitar x_r.

Predictor

Hampiran :

$$
\begin{aligned}
y^*_{r+1} &= y_r + {}^h/_{12}\,(23\,f_r - 16\,f_{r-1} + 5\,f_{r-2}) \\
&= y_r + {}^h/_{12}\,[23\,f_r - 16(\,f_r - hf_r' + {}^1/_2\,h^2 f_r'' - {}^1/_6\,h^3 f_r''' + \ldots) \\
&\quad + 5(f_r - 2hf_r' + 2h^2 f_r'' - {}^8/_6\,h^3 f_r''' + \ldots)] \\
&= y_r + {}^h/_{12}\,[12 f_r + 6hf_r' + 2h^2 f_r'' - 4h^3 f_r''' + \ldots] \\
&= y_r + hf_\mathrm{r} + {}^1/_2\,h^2 f_r' + {}^1/_6\,h^3 f_r'' - {}^1/_3\,h^4 f_r''' + \ldots
\end{aligned}
$$

Sejati:

$$
\begin{aligned}
Y_{r+1} &= y_r + hy'_r + {}^1/_2\,h^2 y_r'' + {}^1/_6\,h^3 y_r''' + {}^1/_{24}\,h^4 y_r^{(4)} + \ldots \\
&= y_r + hf_r + {}^1/_2\,h^2 f_r' + {}^1/_6\,h^3 f_r'' + {}^1/_{24}\,h^4 f_r''' + \ldots
\end{aligned}
$$

Galat per langkah *predictor*:

$$
\begin{aligned}
E_p &= \text{sejati - hampiran} \\
&= {}^1/_{24}\,h^4 f_r''' + {}^1/_3\,h^4 f_r''' + \ldots \\
&= {}^9/_{24}\,h^4 f_r''' = {}^9/_{24}\,h^4 y''(t) \qquad , \; x_{r-2} < t < x_{r+1} \\
&= O(h^4)
\end{aligned}
$$

Corrector

Hampiran :

$$
\begin{aligned}
y_{r+1} &= y_r + {}^h/_{12}\,(5f^*_{r+1} + 8f_r - f_{r-1}) \\
&= y_r + {}^h/_{12}\,[5(\,f_r + hf_r' + {}^1/_2\,h^2 f_r'' + {}^1/_6\,h^3 f_r''' + \ldots) + 8\,f_r \\
&\quad (\,f_r - hf_r' + {}^1/_2\,h^2 f_r'' - {}^1/_6\,h^3 f_r''' + \ldots)] \\
&= y_r + {}^h/_{12}\,(12 f_r + 6hf_r' + 2h^2 f_r'' + h^3 f_r''' + \ldots) \\
&= y_r + hf_r + {}^1/_2\,h^2 f_r' + {}^1/_6\,h^3 f_r'' + {}^1/_{12}\,h^4 f_r''' + \ldots
\end{aligned}
$$

Galat per langkah *corrector*:

$$
\begin{aligned}
E_p &= \text{sejati - hampiran} \\
&= {}^1/_{24}\,h^4 f_r''' - {}^1/_{12}\,h^4 f_r''' + \ldots
\end{aligned}
$$

$= \; ^{-1}/_{24}\, h^4 f_r''' \; = \; ^{-1}/_{24}\, h^4 y''(t) \qquad , \; x_{r-2} \; < \; t \; < \; x_{r+1}$

$= \; O(h^4)$

Orde metode = 4 - 1 = 3.

8.10.2 Metode Milne-Simpson

Metode Milne-Simpson didasarkan pada integrasi $f(x, y(x))$ pada selang $[x_{r-3}, x_{r+1}]$:

$$y(x_{r+1}) = y(x_{r-3}) + \int_{x_{r-3}}^{x_{r+1}} f(x, y(x))dx \qquad \text{(P.8.40)}$$

Persamaan *predictor* dan *corrector* metode Milne-Simpson adalah

$$\textit{predictor} : \; y^*_{r+1} = y_{r-3} + {}^{4h}/_3\,(2f_{r-2} - f_{r-1} + 2f_r) \qquad \text{(P.8.41)}$$

$$\textit{corrector} : \; y_{r+1} = y_{r-1} + {}^{h}/_3\,(f_{r-1} + 4\,f_r + f_{r+1}) \qquad \text{(P.8.42)}$$

dan galat per langkahnya adalah dalam orde $O(h^5)$, yaitu:

$$\textit{predictor} : \; E_p = Y_{r+1} - y^*_{r+1} \approx \frac{28h^5}{90}\, y^{(5)}(t)$$

$$\textit{corrector} : \; E_p = Y_{r+1} - y_{r+1} \approx \frac{-1h^5}{90}\, y^{(5)}(t)$$

untuk $x_{r-3} < t < x_{r+1}$.

8.10.3 Metode Hamming

Persamaan *predictor* dan *corrector* metode Hamming adalah

$$\textit{predictor} : \; y^*_{r+1} = y_{r-3} + {}^{4h}/_3\,(2\,f_{r-2} - f_{r-1} + 2f_r) \qquad \text{(P.8.43)}$$

$$\textit{corrector} : \; y_{r+1} = \frac{-y_{r-2}}{8} + \frac{9y_r}{8} + \frac{3h}{8}(-f_{r-1} + 2\,f_r + f_{r+1}) \qquad \text{(P.8.44)}$$

8.10.4 Prosedur Pendahuluan

PDB hanya mempunyai satu nilai awal, yaitu $y_0 = y(x_0)$. Dengan demikian, metode banyak-langkah tidak swa-mulai (*self-start*), sehingga tidak dapat diterapkan langsung, sebab metode tersebut memerlukan beberapa buah nilai awal. Inilah kelemahan metode banyak-langkah.

Misalkan *predictor* mempunyai persamaan

$$y^*_{r+1} = y_r + {}^h/_{12}\ (23f_r - 16f_{r-1} + 5f_{r-2})$$

Untuk menghitung y^*_3, kita harus mempunyai nilai y_0, y_1, dan y_2 agar nilai

$$f_0 = f(x_0, y_0)\ ,\ f_1 = f(x_1, y_1)\ ,\quad f_2 = f(x_2, y_2)$$

dapat ditentukan. Untuk mendapatkan beberapa nilai awal yang lain, kita harus melakukan prosedur pendahuluan (*starting procedure*) dengan metode PDB yang bebas. Metode PDB yang sering dijadikan sebagai prosedur pendahuluan adalah:

- metode Euler
- metode Runge-Kutta
- metode deret Taylor

Jadi, untuk contoh *predictor* di atas, y_1 dan y_2 dihitung terlebih dahulu dengan salah satu prosedur pendahuluan. Selanjutnya, metode P-C dapat dipakai untuk menghitung y_3, y_4, ..., y_n.

Program 8.7 Metode Adams-Bashforth-Moulton

```
function y_Adams_Bashforth_Moulton(x0, y0, b, h:real):real;
 {menghitung y(b) dengan metode Adams_Bashforth_moulton pada PDB
   y'=f(x,y);   y(x0)=y0 }
var
   r, n: integer;
   x, y, y0, y1, y2, y3 : real;
begin
   n:=(b-x0)/h; {jumlah langkah}
   y0:=y0;      {nilai awal dari PDB}

   {Prosedur pendahuluan untuk menghitung nilai awal lain, y1, y2, y3}
   y1:=y_RK3(x0, y0, x0+h, h);        {y(x1)}
   y2:=y_RK3(x0, y0, x0+2*h, h);      {y(x2)}
   y3:=y_RK3(x0, y0, x0+3*h, h);      {y(x3)}
   x:=x0 + 3*h;                       { x3  }
   for r:=4 to n do
     begin
        y:=y3 + h/24*(-9*f(x-3*h, y0) + 37*f(x-2*h, y1) - 59*f(x-h, y2)
              + 55f(x, y3);
        y:=y3 + h/24*(f(x-2*h, y1)-5*f(x-h, y2) + 19*f(x,y3)
              + 9*f(x+h,y);
        y0:=y1;
        y1:=y2;
```

```
        y2:=y3;
        y3:=y;
         x:=x+h;        ( titik berikutnya }
     end;
    y_Adams_Bashforth_Moulton:=y;
 end;
```

8.10.5 Keidealan Metode *Predictor-Corrector*

Metode *predictor-corrector* dikatakan ideal jika galat per langkah *predictor* mempunyai orde yang sama dengan galat per langkah *corrector*:

$$\text{galat per langkah } \textit{predictor} : \; Y_{r+1} - y^*_{r+1} \approx A_r h^p$$

$$\text{galat per langkah } \textit{corrector} : \; Y_{r+1} - y_{r+1} \approx aA_r h^p$$

dengan a adalah tetapan yang diketahui. Metode Adams-Bashforth-Moulton, metode Milne-Simpson, dan metode Hamming adalah metode P-C yang ideal. Metode Heun adalah metode P-C yang tidak ideal, karena

$$\text{galat per langkah } \textit{predictor} : \; E_p = Y_{r+1} - y_{r+1} \approx \frac{1}{2} y''(t)h^2 \approx Ah^2$$

$$\text{galat per langkah } \textit{corrector} : \; E_p = Y_{r+1} - y_{r+1} \approx -\frac{1}{12} y'''(t)h^3 \approx Bh^3$$

Jika sebuah metode P-C ideal, kita dapat memperoleh nilai y_{r+1} yang lebih baik (*improve*) sebagai berikut:

$$\bar{y}_{r+1} - y^*_{r+1} = A_r h^p \qquad \text{(P.8.45)}$$

$$\bar{y}_{r+1} - y_{r+1} = \alpha A_r h^p \qquad \text{(P.8.46)}$$

dengan $\bar{y}_{r+1}$ adalah taksiran yang lebih baik dari pada y_{r+1}.

Rumus $\bar{y}_{r+1}$ dapat diperoleh dengan membagi persamaan (P.8.45) dengan persamaan (P.8.46):

$$\frac{\bar{y}_{r+1} - y^*_{r+1}}{\bar{y}_{r+1} - y_{r+1}} = \frac{A_r h^p}{aA_r h^p} = \frac{1}{a}$$

$$\Leftrightarrow \quad \bar{y}_{r+1} - Y_{r+1} = a\, y_{r+1} - a\, y^*_{r+1}$$

$$\Leftrightarrow \quad \bar{y}_{r+1}\ (1 - \alpha) = y_{r+1} - \ a\ y*_{r+1}$$

$$\Leftrightarrow \quad \bar{y}_{r+1} = \frac{y_{r+1} - a\ \ y*_{r+1}}{1-a}$$

$$\Leftrightarrow \quad \bar{y}_{r+1} = \frac{y_{r+1}}{1-a} - \frac{a\ \ y*_{r+1}}{1-a}$$

$$\Leftrightarrow \quad \bar{y}_{r+1} = \frac{(1-a)y_{r+1} + a\ \ y^{*}{}_{r+1}}{1-a} - \frac{a\ \ y*_{r+1}}{1-a}$$

$$\Leftrightarrow \quad \bar{y}_{r+1} = \frac{(1-a)y_{r+1}}{1-a} + \frac{y_{r+1}}{1-a} - \frac{a\ \ y*_{r+1}}{1-a}$$

$$\Leftrightarrow \quad \bar{y}_{r+1} = y_{r+1} + \frac{a}{1-a}\ (y_{r+1} - y*_{r+1}) \qquad \text{(P.8.47)}$$

Suku $^{\alpha}/_{(1-\alpha)}\ (y_{r+1} - y*_{r+1})$ pada persamaan (P.8.47) merupakan taksiran galat per langkah untuk menghitung $\bar{y}_{r+1}$, dan menyatakan faktor koreksi terhadap nilai y_{r+1}. Jadi, untuk mendapatkan taksiran nilai y_{r+1} yang lebih baik, tambahkan y_{r+1} dengan faktor koreksi tersebut.

Contoh 8.7

Tentukan perkiraan galat per langkah untuk nilai y_{r+1} yang lebih baik dengan metode Adams-Bashforth-Moulton.

Penyelesaian:

galat per langkah *predictor* : $E_p = y_{r+1} - y^{*}{}_{r+1} \approx {}^{251}/_{720}\ y^{(5)}(t)h^5$
galat per langkah *corrector* : $E_p = y_{r+1} - y_{r+1} \approx {}^{-19}/_{720}\ y^{(5)}(t)h^5$

Dari persamaan galat di atas, diperoleh

A_r = 251/720 dan αA_r = -19/720

Nilai α ditentukan sebagai berikut:

$\Leftrightarrow\ \alpha A_r$ = -19/720
$\Leftrightarrow\ \alpha$(251/720) = (-19/720)
$\Leftrightarrow\ \alpha$ = -19/251

Sehingga,

$$\bar{y}_{r+1} = y_{r+1} + \frac{(-19/251)}{1+19/251}\,(y_{r+1} - y^*_{r+1})$$

$$= y_{r+1} - \frac{19}{270}\,(y_{r+1} - y^*_{r+1})$$

Jadi, taksiran galat per langkah untuk nilai y_{r+1} adalah

$$E_p \approx {}^{-19}/_{270}\,(y_{r+1} - y^*_{r+1})$$ ■

8.11 Pemilihan Ukuran Langkah yang Optimal

Ukuran langkah h adalah persoalan yang penting pada metode PDB yang berdasarkan langkah per langkah ini. Jika h terlalu kecil, jumlah langkahnya semakin banyak dan galat pembulatannya semakin besar. Sebaliknya, jika h terlalu besar, galat pemotongannya juga bertambah besar karena galat pemotongan sebanding dengan h. Timbul pertanyaan: berapakah ukuran langkah yang optimal agar galat per langkah metode PDB dapat dipertahankan kurang dari e?

Misalkan kita menghitung solusi PDB dengan metode Runge-Kutta orde-4. Kita ingin galat per langkahnya kurang dari e. Galat per langkah metode Runge-Kutta orde-4 berbentuk

$$E_p(h) = Bh^5 \qquad \text{(P.8.48)}$$

dengan B adalah konstanta yang bergantung pada soal yang diberikan. Agar E_t (h) kurang dari e,

$$Bh^5 < e$$

maka ukuran langkah h haruslah

$$h < (e/B)^{1/5} \qquad \text{(P.8.49)}$$

Konstanta B ditentukan dengan cara percobaan sebagai berikut:

1. Hitung $y(x_1)$ dengan ukuran langkah h (disimbolkan dengan $y(x_1;h)$. Galat per langkahnya dinyatakan oleh persamaan (P.8.48).

2. Hitung kembali $y(x_1)$ dengan ukuran langkah $h/2$ (disimbolkan dengan $y(x_2;h/2)$. Jadi, perlu dua langkah untuk menghitung $y(x_1)$ dengan galat tiap langkah per langkah seluruhnya adalah:

$$\Leftrightarrow \quad E_p(h/2) + E_p(h/2) = B(h/2)^5 + B(h/2)^5$$

$$\Leftrightarrow \quad 2\,E_p(h/2) = 2B(h/2)^5 = \frac{Bh^5}{16} \qquad \text{(P.8.50)}$$

3. Kurangi (P.8.48) dengan (P.8.50):

$$E_p(h) - 2\,E_p(h/2) = B(h)^5 - {}^1/_{16}\,Bh^5 = {}^{15}/_{16}\,Bh^5 \qquad \text{(P.8.51)}$$

4. Ruas kiri persamaan (P.8.51) dihitung sebagai

$$E_p(h) - 2\,E_p(h/2) = y(x_1;h) - y(x_2;h/2) \qquad \text{(P.8.52)}$$

5. Samakan persamaan (P.8.51) dengan persamaan (P.8.52):

$$ {}^{15}/_{16}\,Bh^5 = y(x_1;h) - y(x_2;h/2)$$

sehingga diperoleh

$$B = \frac{16}{15}\;\frac{y(x_1;h) - y(x_2;h/2)}{h^5} \qquad \text{(P.8.53)}$$

6. Sulihkan nilai B dalam ketidaksamaan (P.8.49) sehingga diperoleh batas maksimum nilai ukuran langkah h.

Contoh 8.8

Diberikan PDB

$$y' = y/(1 + x^2) \quad , y(0)=1$$

Tentukan ukuran langkah h agar galat per langkah kurang dari 0.00001.

Penyelesaian:

Diketahui:

$x_0 = 0 \rightarrow y_0 = 1$
$\varepsilon = 0.00001$

Dengan ukuran langkah h = 1 dan h = 0.5, metode Runge-Kutta orde-4 menghasilkan

$y_1 = y\,(1; 1) = 0.4566667$
$y_1 = y\,(1; 0.5) = 0.4559973$

Nilai B dihitung dengan persamaan (P.8.53):

$$B = \frac{16}{15}\,\frac{(0.4566667 - 0.4559973)}{1^5} = 0.00063$$

Jadi, ukuran langkah yang optimal agar galat per langkah metode Runge-Kutta orde-4 kurang dari e ialah

$h < (0.00001/0.00063)^{1/5} = 0.44$ ■

8.12 Sistem Persamaan Diferensial

Dalam bidang sains dan rekayasa, persamaan diferensial banyak muncul dalam bentuk simultan, yang dinamakan **sistem persamaan diferensial**, sebagai berikut:

$$\begin{aligned}
y'_1 &= \frac{dy_1}{dx} = f_1(x, y_1, y_2, \ldots, y_n) \qquad , y_1(x_0) = y_{10} \\
y'_2 &= \frac{dy_2}{dx} = f_2(x, y_1, y_2, \ldots, y_n) \qquad , y_2(x_0) = y_{20} \\
&\vdots \\
y'_n &= \frac{dy_n}{dx} = f_n(x, y_1, y_2, \ldots, y_n) \qquad , y_n(x_0) = y_{n0}
\end{aligned} \qquad \text{(P.8.54)}$$

Sistem persamaan diferensial tersebut dapat ditulis dalam notasi vektor sebagai berikut:

$$\mathbf{y}' = \mathbf{f}\,(x, \mathbf{y}) \quad , \quad \mathbf{y}(\mathbf{x_0}) = \mathbf{y_0} \qquad \text{(P.8.55)}$$

yang dalam hal ini,

$$\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ . \\ . \\ . \\ y_n \end{bmatrix} \quad , \quad \mathbf{y}' = \begin{bmatrix} y'_1 \\ y'_2 \\ . \\ . \\ . \\ y'_n \end{bmatrix} \quad , \quad \mathbf{f} = \begin{bmatrix} f_1 \\ f_2 \\ . \\ . \\ . \\ f_n \end{bmatrix} \quad , \quad \mathbf{y_0} = \begin{bmatrix} y_{1_0} \\ y_{2_0} \\ . \\ . \\ . \\ y_{n_0} \end{bmatrix}$$

Semua metode yang telah dijelaskan untuk persamaan tunggal (Euler, Runge-Kutta, dll.) dapat diterapkan pada sistem persamaan di atas.

Contoh 8.9

Diketahui sistem PDB orde-1

$$\frac{dy}{dt} = -0.5\,y \qquad , \quad y(0) = 4$$

$$\frac{dz}{dt} = 4 - 0.3z - 0.1\,y \qquad , \quad z(0) = 6$$

Hitung $y(0.5)$ dan $z(0.5)$ dengan (a) metode Euler, dan (b) metode Runge-Kutta orde 3. Ambil $h = 0.5$.

Penyelesaian:

$$\mathbf{y} = \begin{bmatrix} y \\ z \end{bmatrix}, \quad \mathbf{y}' = \begin{bmatrix} y' \\ z' \end{bmatrix}, \quad \mathbf{f} = \begin{bmatrix} f_1 \\ f_2 \end{bmatrix} = \begin{bmatrix} -0.5y \\ 4 - 0.3z - 0.1y \end{bmatrix} \quad \mathbf{y_0} = \begin{bmatrix} 4 \\ 6 \end{bmatrix}$$

Sistem PDB di atas dapat ditulis menjadi $\mathbf{y}' = \mathbf{f}\,(t, \mathbf{y})$, $\mathbf{y}(\mathbf{t_0}) = \mathbf{y_0}$

(a) Dengan metode Euler $y_{r+1} = y_r + hf(\,t_r\,, y_r)$:

$y_{r+1} = y_r + hf_1\,(t_r, y_r, z_r)$

$z_{r+1} = z_r + hf_2\,(t_r, y_r, z_r)$

$t_0 = 0 \;\rightarrow\; y_0 = 4$ dan $z_0 = 6$

$t_0 = 0.5 \rightarrow y_1 = y(0.5) = y_0 + hf_1(t_0, y_0, z_0) = 4 + (0.5)\{(-0.5)(4)\} = 3$

$z_1 = z(0.5) = z_0 + hf_2(t_0, y_0, z_0)$

$= 6 + (0.5)\{4 - (0.3)(6) - (0.1)(4)\} = 6.9$

(b) Dengan metode Runge-Kutta orde-3,

$$k_1 = hf(t_r, y_r),$$
$$k_2 = hf(t_r + h/2, y_r + k_1/2)$$
$$k_2 = hf(t_r + h, y_r - k_1 + 2k_2)$$
$$y_{r+1} = y_r + (1/6)(k_1 + 4k_2 + k_3)$$

$t_0 = 0 \rightarrow y_0 = 4$
$t_1 = 0.5 \rightarrow y_1 = ?$

$$\begin{aligned} k_1 &= hf_1(t_0, y_0, z_0) \\ &= 0.5\,\{(-0.5)(4)\} = -1 \end{aligned}$$

$$\begin{aligned} k_2 &= hf_1(t_0 + h/2, y_0 + k_1/2, z_0 + k_1/2) \\ &= (0.5)\mathrm{f}_1(0.25, 3.5, 5.5) \\ &= (0.5)\{(-0.5)(3.5)\} \\ &= -0.875 \end{aligned}$$

$$\begin{aligned} k_3 &= hf_1(t_0 + h, y_0 - k_1 + 2k_2,\ z_0 - k_1 + 2k_2) \\ &= 0.5\ \mathrm{f}_1(0.5, 3.25, 6.815) \\ &= 0.5\{(-0.5)(3.25)\} \\ &= -0.8125 \end{aligned}$$

sehingga

$$\begin{aligned} y_1 &= y(0.5) = y_0 + {}^1/_6\,(k_1 + 4k_2 + k_3) \\ &= 4 + {}^{1/}{}_6\,\{-1 + 4(-0.875) + (-0.8125)\} \\ &= 3.114583 \end{aligned}$$

$t_0 = 0 \rightarrow z_0 = 6$
$t_1 = 0.5 \rightarrow z_1 = ?$

$$\begin{aligned} k_1 &= hf_2(t_0, y_0, z_0) \\ &= 0.5\,\{4 - (0.3)(6) - (0.1)(4)\} = 0.9 \end{aligned}$$

$$\begin{aligned} k_2 &= hf_2(t_0 + h/2, y_0 + k_1/2, z_0 + k_1/2) \\ &= (0.5)\ \mathrm{f}_2(0.25, 4.45, 6.45) \\ &= (0.5)\{4 - (0.3)(6.45) - (0.1)(4.45)\} \\ &= 0.81 \end{aligned}$$

$$\begin{aligned} k_3 &= hf_2(t_0 + h, y_0 - k_1 + 2k_2,\ z_0 - k_1 + 2k_2) \\ &= 0.5\ \mathrm{f}_2(0.5, 4.72, 6.72) \\ &= 0.5\{4 - (0.3)(6.72) - (0.1)(4.72)\} \\ &= 0.756 \end{aligned}$$

sehingga

$$\begin{aligned} z_1 &= z(0.5) = z_0 + (1/6)(k_1 + 4k_2 + k_3) \\ &= 6 + (1/6)\,\{0.9 + 4(0.81) + 0.756\} \\ &= 6.816 \end{aligned}$$

■

8.13 Persamaan Diferensial Orde Lanjut

Persamaan differensial orde lanjut adalah persaman diferensial dengan orde yang lebih besar dari satu. Persamaan diferensial ini dapat ditulis kembali sebagai sistem persamaan diferensial orde-1.

Misalkan kepada kita diberikan PDB orde-2

$$y'' = f(x, y, y') \quad \text{dengan nilai awal } y(x_0) = y_0 \text{ dan } y'(x_0) = z_0$$

Untuk mengubah PDB orde-2 tersebut menjadi sistem PDB orde-1, misalkan

$$y' = z$$

maka

$$z' = y'' = f(x, y, y') = f(x, y, z) \quad ; y(x_0) = y_0 \text{ dan } z(x_0) = z_0$$

Dengan demikian, persamaan $y'' = f(x, y, y')$ dapat ditulis menjadi sistem persamaan diferensial biasa:

$$\frac{dy}{dx} = z \qquad , \; y(x_0) = y_0$$

$$\frac{dz}{dx} = f(x, y, y') = f(x, y, z) \qquad , \; z(x_0) = z_0$$

atau dalam notasi vektor:

$$\mathbf{y}' = \mathbf{f}(x, \mathbf{y}) \qquad ; \mathbf{y}(\mathbf{x_0}) = \mathbf{y_0}$$

yang dalam hal ini

$$\mathbf{y} = \begin{bmatrix} y \\ z \end{bmatrix} \quad , \; \mathbf{y}' = \mathbf{f} = \begin{bmatrix} z \\ f(x, y, z) \end{bmatrix} \quad , \; \mathbf{y}(\mathbf{x_0}) = \begin{bmatrix} y_0 \\ z_0 \end{bmatrix}$$

Selanjutnya sistem persamaan diferensial biasa ini sapat diselesaikan seperti pada Contoh 8.9 terdahulu.

Contoh 8.10

Nyatakan PDB orde-2 berikut:

$$y'' - 3y' - 2y = 0 \quad ; \quad y(0) = 1 \text{ dan } y'(0) = 0.5$$

ke dalam sistem persamaan diferensial biasa orde-1.

Penyelesaian:

Diketahui PDB orde-2:

$$y'' = 3y' - 2y = f(x, y, y')$$

Misalkan

$$y' = z$$

maka

$$z' = y'' = f(x, y, y') = f(x, y, z) = 3z - 2y$$

dan

$$y(0) = 1,$$
$$z(0) = 0.5;$$

sehingga diperoleh sistem PDB orde-1

$$\frac{dy}{dx} = z \qquad , \; y(0) = 1$$

$$\frac{dz}{dx} = 3z - 2y \qquad , \; z(0) = 0.5$$

atau dalam notasi vektor:

$$\mathbf{y}' = \mathbf{f}(x, \mathbf{y}) \qquad ; \mathbf{y}(0) = \mathbf{y_0}$$

yang dalam hal ini,

$$\mathbf{y} = \begin{bmatrix} y \\ z \end{bmatrix} \quad , \; \mathbf{f} = \begin{bmatrix} z \\ 3z - 2y \end{bmatrix} \quad , \; \mathbf{y_0} = \begin{bmatrix} 1 \\ 0.5 \end{bmatrix}$$

■

Contoh 8.11

Nyatakan PDB orde-3 berikut:

$$y''' - x + y2 - y' + 3y'' = 0 \quad ; \; y(0) = 0; y'(0) = 0.5, y''(0) = 1$$

ke dalam sistem persamaan diferensial biasa orde-1.

Penyelesaian:

$$y''' = x - y2 + y' - 3y'' = f(x, y, y', y'')$$

Misalkan

$$y' = z$$

dan

$$y'' = z' = t$$

maka

$$t' = y''' = f(x, y, y', y'') = f(x, y, z, t) = x - y^2 + z - 3t$$

dan

$$y(0) = 0,$$
$$z(0) = 0.5,$$
$$t(0) = 1;$$

sehingga diperoleh sistem PDB orde-1:

$$\frac{dy}{dx} = z \qquad , \; y(0) = 0$$

$$\frac{dz}{dx} = t \qquad , \; z(0) = 0.5$$

$$\frac{dt}{dx} = x - y^2 + z - 3t \qquad , \; t(0) = 1$$

atau dalam notasi vektor

$$\mathbf{y}' = \mathbf{f}(x, \mathbf{y}) \qquad , \; \mathbf{y}(0) = \boldsymbol{y_0}$$

yang dalam hal ini,

$$\mathbf{y} = \begin{bmatrix} y \\ z \\ t \end{bmatrix} \quad , \; \mathbf{f} = \begin{bmatrix} z \\ t \\ x - y^2 + z - 3t \end{bmatrix} \quad , \; \mathbf{y}(0) = \begin{bmatrix} 0 \\ 0.5 \\ 1 \end{bmatrix} \qquad \blacksquare$$

Contoh 8.12

Nyatakan PDB orde-2 berikut:

$$2x''(t) - 5x'(t) - 3x(t) = 45\, e2t\,, \quad x(0.5) = 2 \text{ dan } x'(0.5) = 1$$

ke dalam sistem PDB orde-1

Penyelesaian:

$$x''(t) = \frac{45}{2}\, e^{2t} + \frac{5}{2}\, x'(t) + \frac{3}{2}\, x(t)$$

Misalkan

$$x'(t) = z(t)$$

maka

$$z'(t) = x''(t) = f(t, x(t), z(t)) = \frac{45}{2} e^{2t} + \frac{5}{2} z(t) + \frac{3}{2} x(t)$$

dan

$$x(0) = 2,$$
$$z(0) = 1;$$

sehingga diperoleh sistem PDB orde-1:

$$\frac{dx}{dt} = z(t) \qquad , x(0.5) = 2;$$

$$\frac{dz}{dt} = \frac{45}{2} e^{2t} + \frac{5}{2} z(t) + \frac{3}{2} x(t) \qquad , z(0.5) = 1$$

atau dalam notasi vektor:

$$\mathbf{y}' = \mathbf{f}(t, \mathbf{y}) \;,\; \mathbf{y}(0.5) = \mathbf{y_0}$$

yang dalam hal ini,

$$\mathbf{y} = \begin{bmatrix} x(t) \\ z(t) \end{bmatrix} \;,\; \mathbf{f} = \begin{bmatrix} z(t) \\ \frac{45}{2} e^{2t} + \frac{5}{2} z(t) + \frac{3}{2} x(t) \end{bmatrix} \;,\; \mathbf{y}\,(0.5) = \begin{bmatrix} 2 \\ 1 \end{bmatrix} \quad \blacksquare$$

8.14 Ketidakstabilan Metode PDB

Pada bab-bab sebelumnya kita telah menyingggung ketidakstabilan pada metode numerik. Karena solusi PDB diperoleh secara lelaran, yang setiap lelaran menghasilkan galat pemotongan, maka ada kemungkinan solusi pada lelaran terakhir menyimpang cukup berarti terhadap solusi sejatinya. Untuk jelasnya perhatikan metode PDB berikut:

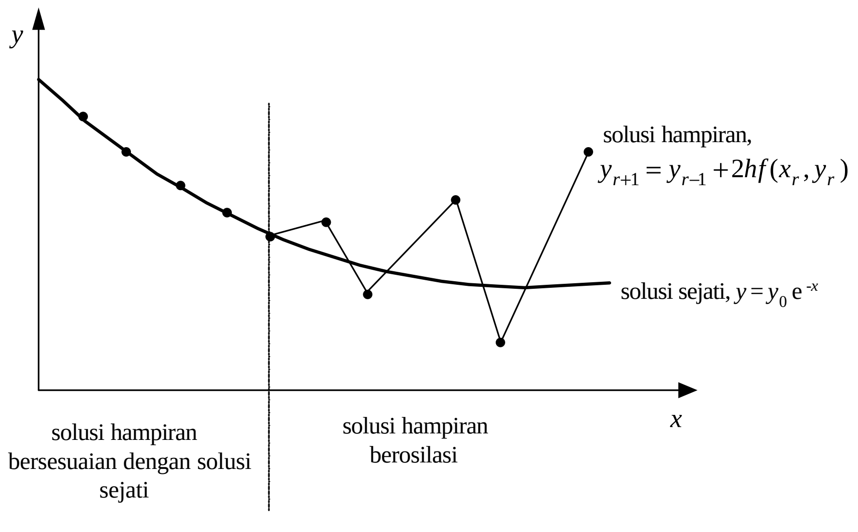
$$y_{r+1} = y_{r-1} + 2hf(x_r, y_r) \tag{P.8.56}$$

Dengan bantuan deret Taylor, galat per langkah metode ini adalah dalam orde $O(h^3)$, yang berarti metode PDB tersebut berorde dua. Kesimpulan sementara kita, metode (P.8.56) menghasilkan solusi yang lebih teliti daripada solusi dengan metoda Euler (yang berorde satu).

Bila metode (P.8.56) diterapkan pada PDB

$$y' = \frac{dy}{dx} = -ay \ , \ y(0) = y_0 > 0 \quad \text{dan } \alpha > 0$$

kita memperoleh hasil yang diperlihatkan oleh Gambar 8.8.



Gambar 8.8 Ketidakstabilan metode PDB

Perhatikan bahwa solusi analitik PDB ini adalah $y = y_0e^{-\alpha x}$, yang mendekati nol dengan peningkatan x. Tetapi solusi numeriknya menjadi tidak stabil dengan peningkatan x. Ketidakstabilan ini disebabkan oleh penumpukan galat per langkah yang "tumbuh" secara tidak terbatas dengan meningkatnya jumlah langkah. Untuk jumlah langkah yang sedikit, solusinya masih stabil. Tetapi dengan meningkatnya jumlah langkah, solusinya menjadi tidak stabil. Bahkan, untuk jumlah langkah yang tidak terhingga, solusinya tumbuh secara tidak terbatas. Jadi, ada metode PDB yang hanya baik untuk x yang kecil, tetapi buruk untuk x yang besar. Ketidakstabilan ditandai oleh solusi yang berosilasi, tetapi ini tidak selalu demikian.

Dalam praktek, hindari penggunaan metode yang tidak stabil. Contoh metode PDB yang tidak stabil untuk x yang besar adalah metode Euler dan metode Milne-Simpson. Metode Heun, metode Runge-Kutta orde-3, metode Runge-Kutta orde-4, dan metode Adams-Bashforth-Moulton adalah metode PDB yang stabil.

8.15 Contoh Soal Terapan

Pada rangkaian listrik, arus yang mengalir tidaklah tetap, tetapi berubah terhadap waktu. Tinjau kembali rangkaian RLC pada Gambar 8.1.

Hukum Kirchoff menyatakan bahwa jumlah aljabar dari perubahan tegangan di sekeliling rangkaian tertutup adalah nol. Selain dalam bentuk PDB orde-1 (P.8.3), hukum Kirchoff kadang-kadang disajikan dalam bentuk PDB orde-2:

$$L \frac{d^2q}{dt^2} + R \frac{dq}{dt} + q/C - E(t) = 0 \qquad \text{(P.8.57)}$$

yang dalam hal ini, L adalah induktansi (dalam henry), R adalah tahanan (dalam ohm), q adalah muatan pada kapasitor (dalam coulomb), C adalah kapsitasitansi (dalam farad), $E(t)$ adalah tegangan yang berubah terhadap waktu (dalam volt). Persamaan (P.8.56) adalah PDB orde-2 yang dapat dipecah menjadi sistem PDB orde-1:

$$\frac{di}{dt} = - Ri/L - q/CL + E(t)/L \qquad , i(0) = 0$$

$$\frac{dq}{dt} = i \qquad , q(0) = 0 \qquad \text{(P.8.58)}$$

Misalkan L = 1 henry, C = 0.25 coulomb, $E(t) = E_0 \sin \omega t$, E_0 = 1 volt, ω = 1.8708 detik, dan R = 0. Hitunglah muatan kapasitor setelah 10 detik dengan metode Euler dan metode Runge-Kutta orde empat (gunakan ukuran langkah h =

0.1 detik). Bandingkan jawaban anda dengan solusi analitiknya yang diturunkan sbb :

$$q(t) = \frac{-E_0}{L(p^2 - w^2)} \frac{w}{p} \sin\ pt + \frac{E_0}{L(p^2 - w^2)} \sin wt \qquad \text{(P.8.59)}$$

dengan $p = 1/\sqrt{(LC)}$. Dengan menyulihkan besaran-besaran di atas diperoleh persamaan $q(t)$, yaitu :

$$q(t) = -1.8708 \sin 2t + 2 \sin (1.8708t)$$

Penyelesaian:

Persoalan ini adalah pencarian solusi PDB

$$\frac{di}{dt} = -Ri/L - q/CL + E(t)/L \qquad , i(0) = 0$$

$$\frac{dq}{dt} = i \qquad , q(0) = 0$$

dengan L = 1 henry, C = 0.25 coulomb, $E(t) = E_0 \sin wt$, E_0 = 1 volt, w = 1.8708 detik, dan R = 0. Diminta menentukan $q(10)$, dalam coulomb, dengan metode Euler dan metode Runge-Kutta orde-4. Solusi dengan kedua metode PDB tersebut diperlihatkan oleh tabel berikut:

Metode Euler	Metode Runge-Kutta Orde-4	Nilai sejati
q[0.0000000000] = 0.0000000000	q[0.0000000000] = 0.0000000000	q[0.0000000000]=0.0000000000
q[0.1000000000] = 0.0000000000	q[0.1000000000] = 0.0003113455	q[0.1000000000]=0.0003106990
q[0.2000000000] = 0.0000000000	q[0.2000000000] = 0.0024585603	q[0.2000000000]=0.0024577123
q[0.3000000000] = 0.0018599064	q[0.3000000000] = 0.0081397788	q[0.3000000000]=0.0081396320
q[0.4000000000] = 0.0073747206	q[0.4000000000] = 0.0187855018	q[0.4000000000]=0.0187873479
q[0.5000000000] = 0.0181375023	q[0.5000000000] = 0.0354437446	q[0.5000000000]=0.0354491638
q[0.6000000000] = 0.0354093809	q[0.6000000000] = 0.0586868743	q[0.6000000000]=0.0586976091
q[0.7000000000] = 0.0600041247	q[0.7000000000] = 0.0885456437	q[0.7000000000]=0.0885634533
q[0.8000000000] = 0.0921942750	q[0.8000000000] = 0.1244745566	q[0.8000000000]=0.1245010622
q[0.9000000000] = 0.1316449743	q[0.9000000000] = 0.1653511112	q[0.9000000000]=0.1653876410
q[1.0000000000] = 0.1773804195	q[1.0000000000] = 0.2095097383	q[1.0000000000]=0.2095571811
q[1.1000000000] = 0.2277863746	q[1.1000000000] = 0.2548094562	q[1.1000000000]=0.2548681346

q[1.2000000000] = 0.2806504669	q[1.2000000000] = 0.2987325039	q[1.2000000000]=0.2988020746
q[1.3000000000] = 0.3332401203	q[1.3000000000] = 0.3385095552	q[1.3000000000]=0.3385889443
q[1.4000000000] = 0.3824160480	q[1.4000000000] = 0.3712656561	q[1.4000000000]=0.3713530345
q[1.5000000000] = 0.4247773155	q[1.5000000000] = 0.3941798264	q[1.5000000000]=0.3942726302
q[1.6000000000] = 0.4568321853	q[1.6000000000] = 0.4046503946	q[1.6000000000]=0.4047453899
q[1.7000000000] = 0.4751873622	q[1.7000000000] = 0.4004576274	q[1.7000000000]=0.4005510206
q[1.8000000000] = 0.4767469470	q[1.8000000000] = 0.3799151173	q[1.8000000000]=0.3800027049
q[1.9000000000] = 0.4589114608	q[1.9000000000] = 0.3420016950	q[1.9000000000]=0.3420790492
q[2.0000000000] = 0.4197667741	q[2.0000000000] = 0.2864663543	q[2.0000000000]=0.2865290349
q[2.1000000000] = 0.3582527055	q[2.1000000000] = 0.2138997625	q[2.1000000000]=0.2139435459
q[2.2000000000] = 0.2743014806	q[2.2000000000] = 0.1257673604	q[2.2000000000]=0.1257884744
q[2.3000000000] = 0.1689371454	q[2.3000000000] = 0.0244007565	q[2.3000000000]=0.0243961093
q[2.4000000000] = 0.0443284075	q[2.4000000000] = -0.0870539817	q[2.4000000000]=-0.0870865913
q[2.5000000000] = -0.0962108212	q[2.5000000000] = -0.2047305158	q[2.5000000000]=-0.2047922177
q[2.6000000000] = -0.2482767430	q[2.6000000000] = -0.3241762399	q[2.6000000000]=-0.3242669531
q[2.7000000000] = -0.4064879707	q[2.7000000000] = -0.4405214587	q[2.7000000000]=-0.4406398018
q[2.8000000000] = -0.5646532983	q[2.8000000000] = -0.5486772875	q[2.8000000000]=-0.5488205463
q[2.9000000000] = -0.7159907450	q[2.9000000000] = -0.6435542527	q[2.9000000000]=-0.6437184101
q[3.0000000000] = -0.8533910308	q[3.0000000000] = -0.7202923292	q[3.0000000000]=-0.7204721585
q[3.1000000000] = -0.9697161675	q[3.1000000000] = -0.7744922968	q[3.1000000000]=-0.7746815187
q[3.2000000000] = -1.0581216764	q[3.2000000000] = -0.8024378810	q[3.2000000000]=-0.8026293781
q[3.3000000000] = -1.1123891644	q[3.3000000000] = -0.8012981891	q[3.3000000000]=-0.8014842725
q[3.4000000000] = -1.1272547310	q[3.4000000000] = -0.7693004777	q[3.4000000000]=-0.7694731949
q[3.5000000000] = -1.0987179945	q[3.5000000000] = -0.7058642733	q[3.5000000000]=-0.7060157456
q[3.6000000000] = -1.0243164985	q[3.6000000000] = -0.6116892854	q[3.6000000000]=-0.6118120597
q3.7000000000] = -0.9033509087	q[3.7000000000] = -0.4887913460	q[3.7000000000]=-0.4888787465
q[3.8000000000] = -0.7370477501	q[3.8000000000] = -0.3404827158	q[3.8000000000]=-0.3405291792
q[3.9000000000] = -0.5286484334	q[3.9000000000] = -0.1712954258	q[3.9000000000]=-0.1712968034
q[4.0000000000] = -0.2834159319	q[4.0000000000] = 0.0131512245	q[4.0000000000]=0.0131974120
q[4.1000000000] = -0.0085536027	q[4.1000000000] = 0.2063354087	q[4.1000000000]=0.2064297785
q[4.2000000000] = 0.2869658080	q[4.2000000000] = 0.4010637627	q[4.2000000000]=0.4012049494
q[4.3000000000] = 0.5926591052	q[4.3000000000] = 0.5897404236	q[4.3000000000]=0.5899250402
q[4.4000000000] = 0.8968737130	q[4.4000000000] = 0.7646640123	q[4.4000000000]=0.7648866983
q[4.5000000000] = 1.1872011323	q[4.5000000000] = 0.9183409247	q[4.5000000000]=0.9185944813
q[4.6000000000] = 1.4509493520	q[4.6000000000] = 1.0438022210	q[4.6000000000]=1.0440778328
q[4.7000000000] = 1.6756574551	q[4.7000000000] = 1.1349108485	q[4.7000000000]=1.1351983842
q[4.8000000000] = 1.8496328856	q[4.8000000000] = 1.1866459239	q[4.8000000000]=1.1869343067

q[4.9000000000] = 1.9624897591	q[4.9000000000] = 1.1953513582	q[4.9000000000]=1.1956289919
q[5.0000000000] = 2.0056653360	q[5.0000000000] = 1.1589372228	q[5.0000000000]=1.1591924575
q[5.1000000000] = 1.9728914217	q[5.1000000000] = 1.0770238867	q[5.1000000000]=1.0772455061
q[5.2000000000] = 1.8605980828	q[5.2000000000] = 0.9510210585	q[5.2000000000]=0.9511987667
q[5.3000000000] = 1.6682286867	q[5.3000000000] = 0.7841363494	q[5.3000000000]=0.7842612382
q[5.4000000000] = 1.3984478724	q[5.4000000000] = 0.5813107578	q[5.4000000000]=0.5813757328
q[5.5000000000] = 1.0572275755	q[5.5000000000] = 0.3490814358	q[5.5000000000]=0.3490815810
q[5.6000000000] = 0.6538005648	q[5.6000000000] = 0.0953751228	q[5.6000000000]=0.0953079874
q[5.7000000000] = 0.2004759507	q[5.7000000000] = -0.1707614026	q[5.7000000000]=-0.1708956119
q[5.8000000000] = -0.2876833746	q[5.8000000000] = -0.4394847308	q[5.8000000000]=-0.4396830618
q[5.9000000000] = -0.7933156150	q[5.9000000000] = -0.7005204937	q[5.9000000000]=-0.7007772731
q[6.0000000000] = -1.2973356744	q[6.0000000000] = -0.9435576587	q[5.9999999999]=-0.9438646315
q[6.1000000000] = -1.7796142292	q[6.1000000000] = -1.1586562582	q[6.0999999999]=-1.1590028396
q[6.2000000000] = -2.2197377833	q[6.2000000000] = -1.3366524173	q[6.1999999999]=-1.3370260507
q[6.3000000000] = -2.5978226591	q[6.3000000000] = -1.4695442264	q[6.2999999999]=-1.4699308346
q[6.4000000000] = -2.8953522153	q[6.4000000000] = -1.5508423588	q[6.3999999999]=-1.5512268737
q[6.5000000000] = -3.0960040093	q[6.5000000000] = -1.5758703518	q[6.4999999999]=-1.5762373017
q[6.6000000000] = -3.1864323062	q[6.6000000000] = -1.5420011309	q[6.5999999999]=-1.5423352630
q[6.7000000000] = -3.1569713867	q[6.7000000000] = -1.4488185980	q[6.6999999999]=-1.4491055107
q[6.8000000000] = -3.0022265956	q[6.8000000000] = -1.2981958585	q[6.7999999999]=-1.2984226178
q[6.9000000000] = -2.7215230006	q[6.9000000000] = -1.0942848179	q[6.8999999999]=-1.0944405314
q[7.0000000000] = -2.3191858552	q[7.0000000000] = -0.8434153189	q[6.9999999999]=-0.8434916419
q[7.1000000000] = -1.8046326632	q[7.1000000000] = -0.5539055852	q[7.0999999999]=-0.5538971368
q[7.2000000000] = -1.1922633539	q[7.2000000000] = -0.2357893358	q[7.1999999999]=-0.2356940044
q[7.3000000000] = -0.5011426825	q[7.3000000000] = 0.0995315992	q[7.2999999999]=0.0997124765
q[7.4000000000] = 0.2455228159	q[7.4000000000] = 0.4396971489	q[7.3999999999]=0.4399587439
q[7.5000000000] = 1.0211024606	q[7.5000000000] = 0.7718465789	q[7.4999999999]=0.7721806721
q[7.6000000000] = 1.7964342962	q[7.6000000000] = 1.0831067579	q[7.5999999999]=1.0835019807
q[7.7000000000] = 2.5408657292	q[7.7000000000] = 1.3610925048	q[7.6999999999]=1.3615347167
q[7.8000000000] = 3.2234070736	q[7.8000000000] = 1.5943995153	q[7.7999999999]=1.5948723054
q[7.9000000000] = 3.8139568316	q[7.9000000000] = 1.7730702468	q[7.8999999999]=1.7735555409
q[8.0000000000] = 4.2845525957	q[8.0000000000] = 1.8890137961	q[7.9999999999]=1.8894925477
q[8.1000000000] = 4.6105981677	q[8.1000000000] = 1.9363622321	q[8.0999999999]=1.9368151703
q[8.2000000000] = 4.7720160509	q[8.2000000000] = 1.9117479983	q[8.1999999999]=1.9121564038
q[8.3000000000] = 4.7542750480	q[8.3000000000] = 1.8144898094	q[8.2999999999]=1.8148362870
q[8.4000000000] = 4.5492453352	q[8.4000000000] = 1.6466778292	q[8.3999999999]=1.6469470453
q[8.5000000000] = 4.1558380966	q[8.5000000000] = 1.4131527092	q[8.4999999999]=1.4133320623

q[8.6000000000] = 3.5803934777	q[8.6000000000] = 1.1213771469	q[8.5999999999]=1.1214573435
q[8.7000000000] = 2.8367890832	q[8.7000000000] = 0.7812028329	q[8.7000000000]=0.7811783415
q[8.8000000000] = 1.9462512330	q[8.8000000000] = 0.4045398276	q[8.8000000000]=0.4044091900
q[8.9000000000] = 0.9368623754	q[8.9000000000] = 0.0049393809	q[8.9000000000]=0.0047053655
q[9.0000000000] = -0.1572299770	q[9.0000000000] = -0.4028951852	q[9.0000000000]=-0.4032255991
q[9.1000000000] = -1.2968851068	q[9.1000000000] = -0.8036518155	q[9.1000000000]=-0.8040676248
q[9.2000000000] = -2.4392919003	q[9.2000000000] = -1.1819871163	q[9.2000000000]=-1.1824736482
q[9.3000000000] = -3.5395012337	q[9.3000000000] = -1.5231159385	q[9.3000000000]=-1.5236553627
q[9.4000000000] = -4.5521161863	q[9.4000000000] = -1.8133904192	q[9.4000000000]=-1.8139623973
q[9.5000000000] = -5.4330795579	q[9.5000000000] = -2.0408457557	q[9.5000000000]=-2.0414282084
q[9.6000000000] = -6.1414914505	q[9.6000000000] = -2.1956908998	q[9.6000000000]=-2.1962608590
q[9.7000000000] = -6.6413853504	q[9.7000000000] = -2.2707241281	q[9.7000000000]=-2.2712586439
q[9.8000000000] = -6.9033895287	q[9.8000000000] = -2.2616560433	q[9.8000000000]=-2.2621331086
q[9.9000000000] = -6.9062018290	q[9.9000000000] = -2.1673258748	q[9.9000000000]=-2.1677253308
q[10.000000000] = -6.6378101261	q[10.000000000] = -1.9898008772	q[10.000000000]=-1.9901052620

Perbandingan solusi:

	Euler	Runge-Kutta Orde-4	Sejati
$q(10)$	-6.6378101261	-1.9898008772	-1.9901052620

Untuk menghitung $q(10)$ dengan $h = 0.1$ diperlukan sejumlah

$$n = (10 - 0)/0.1 = 100$$

langkah. Karena itu, dapatlah dimengerti mengapa metode PDB yang berorde rendah seperti metode Euler memperlihatkan hasil yang sangat menyimpang (divergen) dengan solusi sejatinya ketika jumlah langkahnya membesar, sedangkan solusi dengan metode Runge-Kutta memperlihatkan kestabilannya pada setiap langkah (bandingkan dengan solusi sejati pada setiap langkah). Ini disebabkan galat per langkah pada metode Euler semakin menumpuk dengan bertambahnya jumlah langkah. Jadi, metode dengan orde tinggi seperti metode Runge-Kutta orde-4 lebih disukai untuk masalah ini.

Tidak ada alasan bagi kita meremehkan hal-hal kecil, karena bukankah sutera itu berasal dari ulat?
(Anonim)

Soal Latihan

1. Diberikan persamaan diferensial berikut : $dy/dx = -2xy^2$, $y(0) = 1$. Lakukan perhitungan numerik untuk menaksir nilai y pada nilai-nilai x dalam selang [0,5] (ambil ukuran langkah $h = 0.2$) :

   (a) metode Euler

   (b) metode Heun

   (c) metode deret Taylor

   (d) metode Runge-Kutta orde-3

   (e) metode Runge-Kutta orde-4

   (f) metode Adams-Bashforth-Moulton

2. Diberikan persamaan diferensial berikut : $dy/dx = x^2y^2$, $y(1) = 0$. Tentukan nilai (1.4) dengan metode-metode (ambil ukuran langkah $h = 0.2$) :

   (a) metode Euler

   (b) metode Heun

   (c) metode deret Taylor

   (d) metode Runge-Kutta orde 3

   (e) metode predictor-corrector Milne

3. Nyatakan dalam sistem persamaan diferensial biasa orde satu :

   (a) $4y''' + 3xy'' + 5y' + xy = 10 - y$; $y(1) = 0$, $y'(1) = y''(1) = 1$

   (b) $Ak\, d^2T/dx^2 + P\sigma(T^4 - 273^4) = Q$; $T'(1) = 0$, $T(1)=0$

   (c) model matematika rangkaian listrik : $0.5\, d^2Q/dt^2 + 6dQ/dt + 50Q = 24 \sin(10t)$ dengan $Q = 0$ dan $i=dQ/dt = 0$ pada $t=0$.

   (d) $x''(t) - x(t) = 6 \cos(t)$; $x(0)$ 2 dan $x''(0) = 3$

   (e) $2y'' + (y')^2 + y = 0$, $y(0)=0$, $y'(0) = 1$

   (f) $y''' = -y + y'$, $y(0) = 1$, $y'(0) = y''(0) = 0$

4. Perlihatkan bahwa metode Runge-Kutta berikut ini :

$$k_1 = hf(x_r , y_r)$$
$$k_2 = hf(x_r + \alpha h, y_r + \alpha k_1)$$

$$y_{r+1} = y_r + [(1 - \frac{1}{2a}) k_1 + \frac{1}{2a} k_2)]$$

adalah berorde dua untuk sembarang tetapan α ($\alpha \neq 0$).

5. *Ekstrapolasi Richardson* yang telah anda kenal di integrasi numerik dapat juga diterapkan pada solusi PDB, yang bertujuan untuk memperbaiki hasil metode Runge-Kutta orde-4. Jika metode Runge-Kutta orde-4 digunakan dengan ukuran langkah h, maka nilai hampiran $y(x)$ adalah:

   $$y(x; h) = y_h + Ch^4 \qquad (1)$$

   dan jika digunakan ukuran step $2h$, maka nilai hampiran $y(x)$ adalah :

   $$y(x) = y(x;2h) + 16Ch^4 \qquad (2)$$

   Perlihatkanlah bahwa hampiran $y(x)$ yang lebih baik *(improve)* adalah :

   $$y(x) = {}^1/_{15}\, [16y(x;h) - y(x;2h)] \qquad (3)$$

   Kemudian hitunglah y (1.4) menggunakan persamaan (3) di atas bila PDB yang digunakan adalah seperti soal nomor 1 di atas.

6. Masih berkaitan dengan e*kstrapolasi Richardson*. Jika metode Heun digunakan dengan ukuran langkah h, maka nilai hampiran $y(x)$ adalah:

   $$y(x) = y_h + Ch^2 \qquad (1)$$

   dan jika digunakan ukuran step $2h$, maka nilai aproksimasi $y(x)$ adalah :

   $$y(x) = y_{2h} + 4Ch^2 \qquad (2)$$

   Perlihatkanlah bahwa hampiran $y(x)$ yang lebih baik *(improve)* adalah :

   $$y(x) = 1/3\ (4y_h - y_{2h}) \qquad (3)$$

   Kemudian hitunglah y (1.4) menggunakan persamaan (3) di atas bila PDB yang digunakan adalah seperti soal nomor 1 di atas.

7. Dengan menggunakan PDB orde 2 pada soal nomor 3(c) di atas, tentukan muatan listrik Q dan arus I pada saat t = 0.2. Metode yang digunakan : Runge-Kutta orde 3 dan ukuran langkah h = 0.1.

8. (a) Perlihatkan galat per langkah metode *predictor-corrector* Milne adalah :
galat per langkah *predictor* :

$$y_{r+1} - y^*_{r+1} \approx \frac{28}{90} h^5 y^{(5)}(t)$$

galat per langkah *corrector* :

$$y_{r+1} - y_{r+1} \approx \frac{-1}{90} h^5 y^{(5)}(t)$$

(b) Tentukan orde metode Milne

(c) Tentukan taksiran y_{r+1} (yaitu nilai yang lebih baik daripada y_{r+1}

(d) Tentukan galat per langkah untuk y_{r+1}

7. Dengan mengingat defenisi kalukulus untuk turunan adalah

$$y'(x) = \lim_{h \to 0} \frac{y(x+h) - y(x)}{h}$$

(a) Turunkan metode Euler dari defenisi turunan tersebut
(b) Bila nilai y dihitung pada $x+h$ dan $x-h$, turunkan metode baru, yaitu metode titik-tengah

10. Turunkanlah persamaan *predictor* pada metode P-C Adams-Bashforth-Moulton bila titik-titik datanya diinterpolasi dengan polinom Newton-Gregory mundur

11. Bila PDB-nya adalah seperti pada soal nomor 1, tentukan ukuran langkah yang optimal agar galat per langkah pada solusi PDB dengan metode Runge-Kutta orde-4 kurang dari 0.000001.

12. Diberikan PDB $y' = -y$, $y(0) = 1$. Dengan mengambil ukuran langkah $h = 0.1$, periksa kestabilan metode Euler, metode Runge-Kutta orde-3, metode titik-tengah (lihat jawaban soal 7b), dan metode Milne pada penaksiran nilai $y(10)$