7 Persamaan Diferensial Biasa

Banyak masalah di dalam ilmu pengetahuan dan teknik menyangkut pengkajian suatu sistem selama periode waktu tertentu. Kebanyakan masalah ini dimodelkan dengan menggunakan suatu sistem persamaan diferensial, dengan waktu sebagai variabel bebas. Bidang kajian persamaan diferensial (PD) tidak hanya merupakan salah satu bagian tercantik dalam matematika, namun ia juga merupakan alat yang penting di dalam memodelkan berbagai fenomena dan masalah dalam bidang ilmu-ilmu fisika, kimia, biologi, ekonomi, dan teknik. Sebagai contoh, masalah-masalah sistem massa pegas, rangkaian induktansi resistor – kapasitor, pemuaian lempeng logam, reaksi kimia, ayunan pendulum, gerak putar massa mengitari benda lain, dan lain-lain dapat dimodelkan dalam bentuk persamaan-persamaan diferensial. Masalah predator – mangsa merupakan suatu contoh klasik masalah persamaan diferensial.

Pemunculan persamaan-persamaan diferensial di dalam matematika terapan tidak lain karena kebanyakan hukum-hukum dalam kajian ilmiah dapat dinyatakan dalam laju perubahan. Sebagai contoh,

$$\frac{du}{dt} = -0.27(u-60)^{5/4}$$

adalah sebuah persamaan yang menjelaskan (secara pendekatan) laju perubahan suhu (u) badan yang kehilangan panas karena pengaruh suhu di sekitar ([3] p. 393).

Penyelesaian suatu persamaan diferensial secara eksak adalah fungsi yang memenuhi PD tersebut dan juga memenuhi beberapa syarat nilai awal fungsi tersebut. Penyelesaian suatu PD dengan metode numerik menghasilkan tabel nilai-nilai fungsi pada beberapa nilai variabel bebasnya, namun tidak dinyatakan secara eksplisit dalam bentuk rumus fungsi.

Pada bab ini kita membahas beberapa metode numerik untuk menyelesaikan persamaan-persamaan diferensial biasa, yakni persamaan-persamaan diferensial yang hanya memuat satu variabel bebas.

Persamaan-persamaan diferensial biasa yang menjadi pusat perhatian kita adalah PD yang berbentuk

$$y'(t) = f(t, y(t)), \qquad t \geq t_0, \tag{7.1}$$

dengan $y(t)$ adalah fungsi tak diketahui yang hendak dicari dan f adalah fungsi dua variabel (dalam t dan y) yang mendefinisikan PD tersebut. Persamaan (7.1) disebut **persamaan diferensial tingkat satu**, karena ia memuat turunan tingkat satu (paling tinggi) fungsi yang hendak dicari. Suatu persamaan diferensial tingkat yang lebih tinggi dapat dirumuskan sebagai sistem persamaan diferensial tingkat satu dan selanjutnya dapat diterapkan metode numerik pada sistem PD tingkat satu tersebut.

Suatu sistem persamaan differensial tingkat satu dapat disajikan dalam bentuk

$$\begin{aligned}
\tfrac{dy_1}{dt} &= f_1(t, y_1, y_2, ..., y_n) \\
\tfrac{dy_2}{dt} &= f_2(t, y_1, y_2, ..., y_n) \\
\tfrac{dy_3}{dt} &= f_3(t, y_1, y_2, ..., y_n) \\
&\vdots \\
\tfrac{dy_n}{dt} &= f_n(t, y_1, y_2, ..., y_n)
\end{aligned} \tag{7.2}$$

untuk $t \in [a,\ b]$. Pada (7.2), f_1, f_2, ..., f_n adalah fungsi-fungsi yang diketahui dalam variabel-variabel t, y_1, ..., y_n. Masing-maing y_i ($i = 1$, 2, ..., n) adalah fungsi dalam t, yang merupakan variabel bebas.

Untuk kenyamanan notasi, sistem (7.2) sering dituliskan dalam bentuk vektor. Jika dituliskan

$$\mathbf{y} = [y_1\ y_2\ y_3 \ldots\ y_n]^T, \qquad \mathbf{f} = [f_1\ f_2\ f_3 \ldots\ f_n]^T,$$

dengan $\mathbf{y}$ dan $\mathbf{f}$ adalah vektor-vektor fungsi, maka (7.2) dapat ditulis sebagai

$$\frac{d\mathbf{y}}{dt} = \mathbf{f}(t, \mathbf{y}). \tag{7.3}$$

Dalam ekspresi terakhir ini kita seolah-olah hanya memiliki sebuah persamaan diferensial dengan satu variabel tak bebas.

mempunyai penyelesaian 'trivial' $y(t) = 0$ *dan penyelesaian umum*

$$y(t) = \frac{1}{1 + ce^{-t}}$$

untuk sebarang konstanta c. Cara lain memandang PD di atas adalah dengan memisahkan variabel-variabelnya, yakni

$$\frac{dy}{y^2 - y} = -dt \qquad \text{atau} \qquad \frac{dy}{y-1} - \frac{dy}{y} = -dt,$$

kemudian mengintegralkan kedua ruas guna memperoleh

$$\ln \frac{y-1}{y} = -t + C \quad \text{atau} \quad \frac{1}{y} = 1 - e^C e^{-t} = 1 + ce^t \quad \text{atau} \quad y = \frac{1}{1 + ce^{-t}}.$$

Untuk menentukan penyelesaian khusus yang memenuhi $y(0) = 4$, *misalnya, kita masukkan nilai ini ke penyelesaian umum*

$$4 = \frac{1}{1+c}, \text{ atau } c = 0.25 - 1 = -0.75,$$

sehingga diperoleh penyelesaian

$$y(t) = \frac{1}{1 - 0.75e^{-t}}, \ t \geq 0.$$

Untuk syarat nilai awal $y(0) = y_0 \neq 0$, *nilai konstanta* $c = 1/y_0 - 1$. *Jika* $y_0 > 0$, *maka* $c > -1$ *dan penyelesaian* $y(t)$ *ada untuk* $0 \leq t < \infty$. *Akan tetapi, jika* $y_0 < 0$, *penyelesaian* $y(t)$ *ada hanya pada interval terbatas* $[0, \ \ln(1 - 1/y_0)]$.

2. *Sekarang perhatikan PD*

$$y'(t) = -y(t)^2$$

yang mempunyai penyelesaian trivial $y(t) = 0$ *dan penyelesaian umum*

$$y(t) = \frac{1}{t + c}$$

dengan c suatu konstanta sebarang. Penyelesaian yang memenuhi syarat nilai awal $y(0) = y_0$, *kita perlu membedakan beberapa kasus.*

(a) *Jika* $y_0 = 0$, *maka penyelesaiannya adalah* $y(t) = 0$ *untuk* $t \geq 0$.

Ini artinya, perubahan kecil pada syarat nilai awal y_0 hanya akan sedikit mengubah penyelesaian $y(t)$ untuk masalah nilai awal semula. Sifat ini sangat diperlukan dalam praktek penyelesaian masalah-masalah nilai awal.

CONTOH 7.5.
Masalah nilai awal

$$y'(t) = -y(t) + 1, \quad 0 \le t \le b, \quad y(0) = 1$$

mempunyai penyelesaian $y(t) = 1$. *Apabila syarat nilai awal diubah sedikit, masalah nilai awalnya menjadi*

$$y'_\epsilon(t) = -y_\epsilon(t) + 1, \quad 0 \le t \le b, \quad y_\epsilon(0) = 1 + \epsilon$$

dan mempunyai penyelesaian $y_\epsilon(t) = 1 - \epsilon e^{-t}$, $t \ge 0$. *Jadi,*

$$|y(t) - y_\epsilon(t)| = |\epsilon e^{-t}| \le |\epsilon|, \quad t \ge 0.$$

Dengan demikian masalah nilai awal di atas bersifat stabil. □

Apabila galat maksimum pada (7.17) jauh lebih besar daripada ϵ (artinya nilai minimum c yang mungkin jauh lebih besar), maka masalah nilai awal (7.15) dikatakan **berkondisi sakit** (*ill-conditioned*). Penyelesaian numerik masalah-masalah nilai awal yang sakit akan menghasilkan galat yang besar pada hasil-hasil perhitungannya.

CONTOH 7.6.
Masalah nilai awal

$$y'(t) = \lambda[y(t) - 1], \quad 0 \le t \le b, \quad y(0) = 1$$

mempunyai penyelesaian $y(t) = 1$. *Apabila syarat nilai awal diubah sedikit, masalah nilai awalnya menjadi*

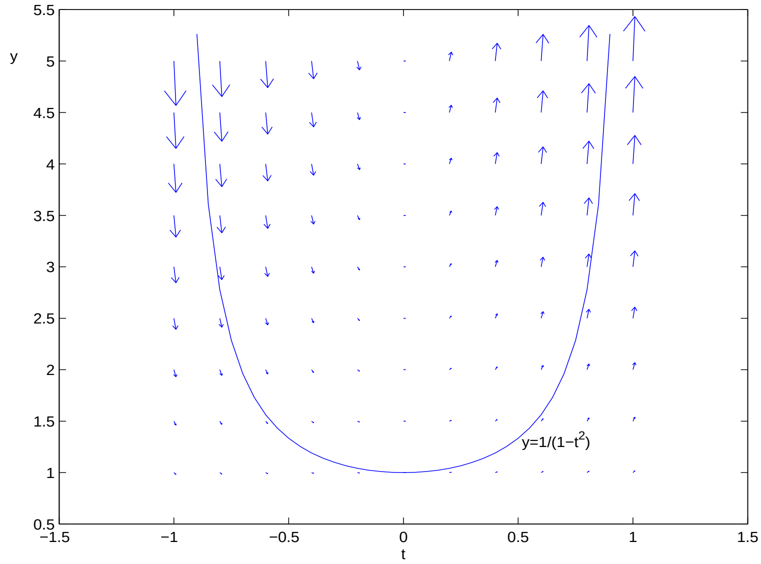
$$y'_\epsilon(t) = \lambda[y_\epsilon(t) - 1], \quad 0 \le t \le b, \quad y_\epsilon(0) = 1 + \epsilon$$

dan mempunyai penyelesaian $y_\epsilon(t) = 1 + \epsilon e^{\lambda t}$, $t \ge 0$. *Jadi,*

$$|y(t) - y_\epsilon(t)| = |\epsilon e^{\lambda t}| \le \begin{cases} |\epsilon|, & \text{jika} \quad \lambda \le 0 \\ |\epsilon| e^{\lambda b} & \text{jika} \quad \lambda \ge 0. \end{cases}$$

Apabila $\lambda < 0$, *selisih* $y(t) - y_\epsilon(t)$ *semakin menurun bersamaan dengan kenaikan* alat yang bermanfaat untuk mengamati perilaku penyelesaian suatu persamaan diferensial.

Dengan MATLAB medan arah dapat digambar menggunakan perintah `quiver`. Daerah R dapat didefinisikan dengan menggunakan perintah MATLAB `meshgrid`.



Gambar 7.2: Medan arah $y'(t) = 2ty^2$ dan sebuah kurva penyelesaian $y = \frac{1}{1-t^2}$

CONTOH 7.7.
Berikut adalah perintah-perintah MATLAB untuk menggambar medan arah PD $y'(t) = 1 - e^{-t}$ dan dua kurva penyelesaian $y_1 = t + e^{-t}$ dan $y_1 = t + e^{-t} + 1$. Hasilnya ditunjukkan pada Gambar 7.1. □

```
>> % medan gradien solusi PD y'=1-e^{-t}, y=t+e^{-t}+c
>> y1=inline('t+exp(-t)'); %dua penyelesaian khusus
>> y2=inline('t+exp(-t)+1');
>> fplot(y1,[-2 5]);
>> fplot(y2,[-2 5]);
>> [t,y]=meshgrid(-2:.4:5,1:.3:7);%daerah medan gradien
```

(b) $y'(t) = \lambda y(t) + t, \qquad y(0) = 3.$

4. Perhatikan persamaan diferensial

$$y'(t) = f_1(t) f_2(y(t))$$

dengan fungsi-fungsi f_1 dan f_2 diketahui. Inilah bentuk umum persamaan diferensial terpisahkan, yang dapat diselesaikan dengan mengelompokkan variabel-variabel yang sama dan mengintegralkannya. PD di atas dapat ditulis ulang menjadi

$$\frac{y'(t)}{f_2(y(t))} = f_1(t)$$

dan inetgralkan kedua ruas

$$\int \frac{y'(t)}{f_2(y(t))}\, dt = \int f_1(t)\, dt.$$

Pada ruas kiri, ganti variabel integrasinya menjadi $z = y(t)$, sehingga

$$\int \frac{dz}{f_2(z)}\, dt = \int f_1(t)\, dt.$$

Setelah pengintegralan, ganti z dengan $y(t)$, selanjutnya selesaikan $y(t)$, apabila mungkin. Jika integral ini dapat dilakukan, maka PD di atas dapat diselesaikan. Lakukan langkah-langkah di atas untuk PD-PD di bawah ini, kemudian carilah penyelesaian umum dan penyelesaian khusus yang memenuhi syarat nilai awal yang diberikan.

(a) $y'(t) = t/y(t), \qquad y(0) = 2$

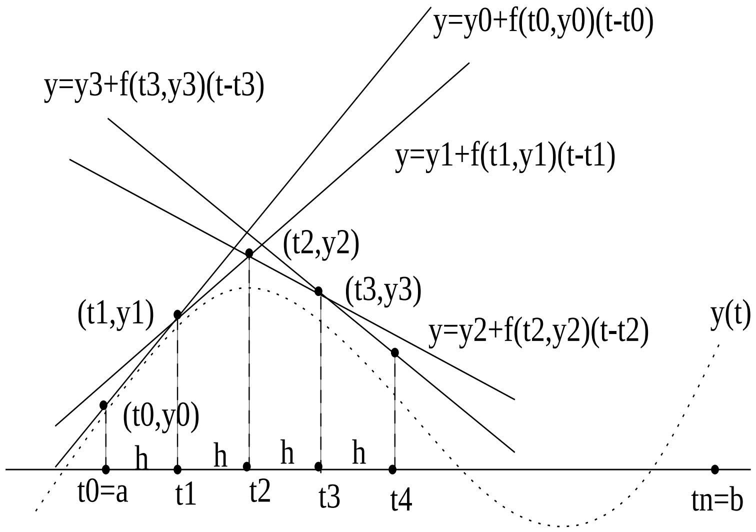
(b) $y'(t) = y(t) + x, \qquad y(1) = 0.$

(c) $y'(t) = y(t)(a - y(t)), \qquad y(0) = a/2, \qquad a > 0$

5. Periksalah kestabilan setiap masalah nilai awal pada soal nomor 1, 3, dan 5 di atas. Periksa pula apakah setiap PD tersebut memenuhi syarat Lipschitz.

6. Perhatikan masalah nilai awal

$$y' = (1 - y^2)^{1/2}, \qquad y(0) = 0.$$



Gambar 7.3: Proses iterasi pada Metode Euler: $y_k = y_{k-1} + hf(t_{k-1}, y_{k-1})$

ALGORITMA 7.1 (METODE EULER).
Menghitung hampiran penyelesaian masalah nilai awal $y' = f(t, y)$ *dengan* $y(t_0) = y_0$ *pada* $[t_0,\ b]$.

INPUT: n, t_0, b, y_0, *dan fungsi* f

OUTPUT: $(t_k,\ y_k)$, $k = 1,\ 2,\ ...,\ n$

LANGKAH-LANGKAH:

1. *Hitung* $h = (b - t_0)/n$
2. *FOR* $k = 1,\ 2,\ 3,\ \ldots,\ n$

   Hitung $t_k = t_{k-1} + h$, $\quad y_k = y_{k-1} + h * f(t_{k-1}, y_{k-1})$.
3. *SELESAI*

Selanjutnya, kita hitung $f(t_1, y_1)$. Nilai ini merupakan hampiran gradien garis singgung kurva tersebut di $t = t_1$. Tarik garis melalui (t_1, y_1) degalat yang terakumulasi menjadi

$$\sum_{k=1}^{n} y''(\xi_k)\frac{h^2}{2} \leq ny''(\xi)\frac{h^2}{2} = \frac{(b-a)}{h}y''(\xi)\frac{h^2}{2} = \frac{(b-a)}{2}y''(c)h = O(h),$$

dengan ξ adalah suatu bilangan pada interval $[a,\ b]$.

Jadi kita melihat bahwa galat pada setiap langkah dalam metode Euler adalah $O(h^2)$, sedangkan galat hampiran pada akhir iterasi dalam metode Euler adalah $O(h)$.

CONTOH 7.9.
Selesaikan $\frac{dy}{dt} = y,\ y(0) = 1$ dengan metode Euler dengan menggunakan beberapa lebar langkah h. Hitung pula galat hampiran yang diperoleh.

Penyelesaian:

Di sini kita mempunyai $f(t_k, y_k) = y_k$, sehingga $y_{k+1} = y_k + hy_k = (1+h)y_k$ dengan $y_0 = 1$. Misalkan pertama-tama kita pilih lebar langkah $h = 0.2$ dan tabulasikan nilai-nilai y untuk $t = 0,\ 0.2,\ 0.4,\ 0.6$, kemudian bandingkan dengan nilai-nilai yang sesungguhnya. Kita tahu bahwa solusi eksak PD di atas adalah $y = e^t$.

t_k	y_k	*Nilai eksak*	*Galat*
0	1	1	0
0.2	1.2	1.22140275816	0.0214027581602
0.4	1.44	1.491824697641	0.0518246976413
0.6	1.728	1.822118800391	0.0941188003905

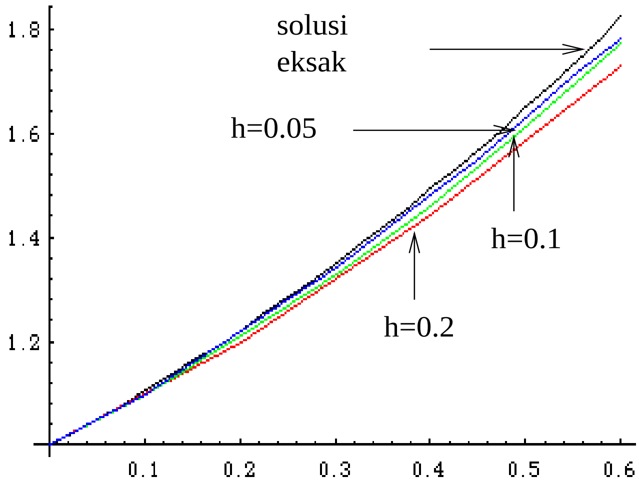
Misalkan sekarang kita pilih lebar langkah $h = 0.1$ dan tabulasikan nilai-nilai y untuk $t = 0,\ 0.1,\ 0.2,\ 0.3, 0.4,\ 0.5,\ 0.6$, kemudian bandingkan dengan nilai-nilai yang sesungguhnya.

t_k	y_k	*Nilai eksak*	*Galat*
0	1	1	0
0.1	1.1	1.105170918076	0.00517091807565
0.2	1.21	1.22140275816	0.0114027581602
0.3	1.331	1.349858807576	0.018858807576
0.4	1.4641	1.491824697641	0.0277246976413
0.5	1.61051	1.6487212707	0.0382112707001
0.6	1.771561	1.822118800391	0.0505578003905

menjadi 0.0505578003905 dengan memperkecil lebar langkah dari 0.2 ke 0.1. Untuk menelusuri fakta ini lebih jauh, misalkan kita pilih lebar langkah $h = 0.05$*. Tabulasikan nilai-nilai* y *untuk* t *=0, 0.05, 0.1, 0.15, 0.2, 0.25, 0.3, 0.35, 0.4, 0.45, 0.5, 0.55, 0.6, kemudian bandingkan dengan nilai-nilai yang sesungguhnya.*

Perhatikan sekarang galat hampiran di $t = 0.6$ *menjadi 0.0262624743684 dari hampiran sebelumnya, sebesar 0.0505578003905.* □

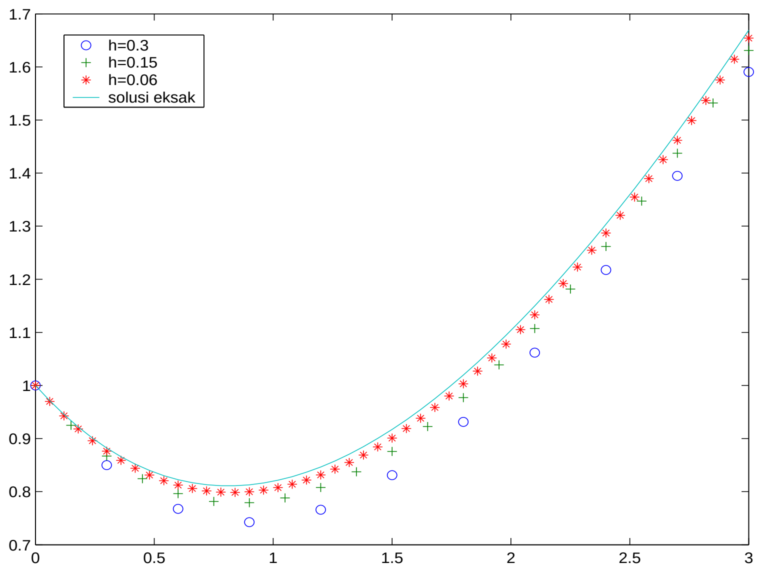
Kelemahan metode ini adalah bahwa lebar langkah harus amat kecil agar mencapai tingkat keakuratan yang dapat diterima. Gambar 7.4 dan 7.5 menunjukkan hampiran penyelesaian persamaan diferensial di atas dengan lebar langkah berlainan.



Gambar 7.5: Solusi eksak dan hampiran penyelesaian PD $y' = y, \quad y(0) = 1$ dengan lebar langkah 0.05, 0.1, dan 0.2

Gambar 7.6 menyajikan fungsi MATLAB `euler4pdb` yang mengimplementasikan metode Euler. Dengan menggunakan fungsi tersebut kita dapat menghitung titik-titik hampiran penyelesaian suatu masalah nilai awal. Kita cukup memasukkan nama fungsi, cacah titik, batas kiri interval, batas kanan interval, dan nilai awal. Titik-titik yang diperoleh dapat

Hasil plot ditunjukkan pada Gambar 7.7 (setelah diberi legenda melalui editor gambar di MATLAB).



Gambar 7.7: Solusi eksak dan tiga hampiran penyelesaian PD $y' = (t - y)/2$, $y(0) = 1$ *dengan tiga langkah berbeda*

Berikut adalah perintah untuk menampilkan tabel perbandingan ketiga hampiran. Kolom pertama adalah nilai t_k, *kolom kedua nilai hampiran* $y(t_k)$ *dengan lebar langkah* $h = 0.3$, *kolom ketiga nilai hampiran* $y(t_k)$ *dengan lebar langkah* $h = 0.15$, *kolom keempat nilai hampiran* $y(t_k)$ *dengan lebar langkah* $h = 0.06$, *dan kolom terakhir adalah nilai eksaknya.* □

```
>> tabel=[t1 y1 y2(1:2:21) y3(1:5:51) ye3(1:5:51)]
tabel =
   0            1.00000000   1.00000000   1.00000000   1.00000000
   0.30000000   0.85000000   0.86687500   0.87620208   0.88212393
   0.60000000   0.76750000   0.79628242   0.81227238   0.82245466
   0.90000000   0.74237500   0.77919415   0.79975357   0.81288445
   1.20000000   0.76601875   0.80788549   0.83138303   0.84643491
   1.50000000   0.83111594   0.87574702   0.90092412   0.91709966
   1.80000000   0.93144855   0.97712355   1.00302121   1.01970898
   2.10000000   1.06173127   1.10717634   1.13307524   1.14981325
   2.40000000   1.21747158   1.26176525   1.28713686   1.30358264
   2.70000000   1.39485084   1.43734789   1.46181461   1.47772078
   3.00000000   1.59062321   1.63089329   1.65419613   1.66939048
```

(d) $y'(t) = -y(t)^2,\ [1,\ 10],\ y(1) = 1,\ y(t) = \frac{1}{t}$

5. Tunjukkan bahwa jika metode Euler digunakan untuk menyelesaikan masalah nilai awal

$$y' = f(t), \qquad y(a) = y_0 = 0$$

pada $[a,\ b]$ dengan lebar langkah $h = (b-a)/N$, maka akan diperoleh

$$y(b) \approx \sum_{k=0}^{N-1} hf(t_k),$$

yang merupakan jumlah Riemann sebagai hampiran $\int_a^b f(x)\ dx$.

7.3 Metode Runge–Kutta

Pada metode Euler nilai y_{k+1} dihitung dengan menggunakan y_k memakai rumus

$$y_{k+1} = y_k + hf(t_k, y_k).$$

Metode ini disebut metode satu-langkah, karena informasi dari satu langkah sebelumnya digunakan untuk menghitung hampiran sekarang. Oleh karena galat di dalam metode Euler adalah $O(h)$, kita perlu memilih lebar langkah yang sangat kecil untuk mendapatkan keakuratan yang diinginkan. Jadi metode ini kurang disenangi daripada metode Runge–Kutta, karena galat di dalam metode Runge–Kutta jauh lebih kecil daripada metode Euler.

Ide di belakang metode Runge–Kutta adalah menghitung nilai $f(t, y)$ pada beberapa titik di dekat kurva penyelesaian yang dipilih dengan metode tertentu di dalam interval $(t_k,\ t_k + h)$ dan mengkombinasikan nilai-nilai ini sedemikian hingga diperoleh keakuratan yang baik pada hampiran berikutnya, y_{k+1}.

Jika titik $(t_k,\ y_k)$ diketahui, maka titik solusi $(t_{k+1},\ y_{k+1})$ dapat diperoleh dengan mengintegralkan $y'(t)$ pada $[t_k,\ t_{k+1}]$,

$$\int_{t_k}^{t_{k+1}} f(t, y(t))\ dt = \int_{t_k}^{t_{k+1}} y'(t)\ dt = y(t_{k+1}) - y(t_k),$$

atau

$$y(t_{k+1}) = y(t_k) + \int_{t_k}^{t_{k+1}} y'(t)\ dt.$$

Dengan menggunakan aturan trapesium dengan lebar langkah $h = t_{k+1} - t_k$ untuk menghitung hampiran suku integral diperoleh hampiran

$$y(t_{k+1}) \approx y(t_k) + \frac{h}{2}[f(t_k, y_k) + f(t_{k+1}, y_{k+1})].$$

Suku kedua pada ruas kanan memuat nilai ruas kiri, y_{k+1}. Kita dapat menggunakan metode Euler untuk menaksir y_{k+1} pada ruas kanan, sehingga diperoleh metode Heun:

$$y(t_{k+1}) = y(t_k) + \frac{h}{2}[f(t_k, y_k) + f(t_{k+1}, y_k + hf(t_k, y_k))].$$

Galat pada setiap langkah dalam metode Heun (atau metode RK2) sama dengan galat aturan trapesium, yakni (dapatkah Anda jelaskan?)

$$y''(\xi_k)\frac{h^3}{12}.$$

Oleh karena itu, galat setelah n langkah pada metode Heun menjadi

$$\sum_{k=1}^{n} y''(\xi_k)\frac{h^3}{12} \approx \frac{t_n - t_0}{12h} y''(\xi)h^3 = O(h^2).$$

Dengan demikian kita peroleh bahwa galat pada setiap langkah (iterasi) dalam metode Heun (RK2) adalah $O(h^3)$, sedangkan galat yang terakumulasi pada akhir setiap langkah adalah $O(h^2)$.

CONTOH 7.11.
Gunakan metode Heun (RK2) untuk menyelesaikan masalah nilai awal

$$y' = (t - y)/2 \quad \textit{pada} \quad [0,\ 3] \quad \textit{dengan} \quad y(0) = 1,$$


```
>>h1=1;h2=1/2;h3=1/4;h4=1/8;tn=3;
>>t0=0;y0=1;yh1=[t0 y0];yh2=yh1;yh3=yh1;yh4=yh1;
>>for k=1:tn/h1, %perhitungan untuk h1
>>t=t0+h1; y=y0+h1/2*((t0-y0)/2+(t-y0-h1*(t0-y0)/2)/2);
>>t0=t; yh1=[yh1;t y]; y0=y;
>>end
>>t0=0;y0=1; for k=1:tn/h2,  %perhitungan untuk h2
>>t=t0+h2; y=y0+h2/2*((t0-y0)/2+(t-y0-h2*(t0-y0)/2)/2);
>>t0=t; yh2=[yh2;t y]; y0=y;
>>end
>>t0=0;y0=1; for k=1:tn/h3, %perhitungan untuk h3
>>t=t0+h3; y=y0+h3/2*((t0-y0)/2+(t-y0-h3*(t0-y0)/2)/2);
>>t0=t; yh3=[yh3;t y]; y0=y;
>>end
>>t0=0;y0=1; for k=1:tn/h4, %perhitungan untuk h4
>>t=t0+h4; y=y0+h4/2*((t0-y0)/2+(t-y0-h4*(t0-y0)/2)/2);
>>t0=t; yh4=[yh4;t y]; y0=y;
>>end
```

Ringkasan hasil perhitungan untuk lebar-lebar langkah ditentukan tersebut disajikan pada Tabel 7.1. □

7.3.2 Metode Runge–Kutta Orde 4 (RK4)

Dalam pembahasan metode Heun (RK2) kita menggunakan aturan trapesium untuk menaksir nilai integral $\int_{t_k}^{t_{k+1}} y'(t)\ dt$. Metode RK4 diturunkan secara serupa, namun dengan menggunakan aturan Simpson dengan lebar langkah $h/2 = (t_{k+1} - t_k)/2$ untuk mendapatkan hampiran nilai integral tersebut, sehingga diperoleh hampiran (Coba Anda turunkan!)

$$y(t_{k+1}) = y(t_k) + \frac{h}{6}\left[f(t_k, y_k) + 4\ f\left(\frac{t_k + t_{k+1}}{2}, y(\frac{t_k + t_{k+1}}{2})\right)\right.$$
$$\left. + f(t_{k+1}, y_k + hf(t_k, y_k))\right].$$

Suku galat di dalam aturan Simpson dengan lebar langkah $h/2$ adalah (Jelaskan!)

$$y^{(4)}(\xi_k)\frac{h^5}{2880},$$

sehingga galat yang terakumulasi setelah n langkah pada metode RK4 adalah

$$\sum_{k=1}^{n} y^{(4)}(\xi_k)\frac{h^5}{2880} \approx ny^{(4)}(\xi)\frac{h^5}{2880} = \frac{(t_n - t_0)}{2h}y^{(4)}(\xi)\frac{h^5}{2880} = O(h^4).$$

CONTOH 7.12.
Gunakan metode Runge–Kutta orde 4 (RK4) untuk menyelesaikan masalah nilai awal

$$y' = (t-y)/2 \quad \textit{pada} \quad [0,\ 3] \quad \textit{dengan} \quad y(0) = 1,$$

dengan menggunakan lebar langkah $h = 1,\ \frac{1}{2},\ \frac{1}{4}$, *dan* $\frac{1}{8}$.
Penyelesaian:
Rumus-rumus gradien S_i *yang perlu dihitung untuk PD di atas adalah, untuk* $k = 0,\ 1,\ 2, ...,\ n(= 3/h)$,

$$\begin{aligned}
S_1 &= \frac{t_k - y_k}{2} \\
S_2 &= \frac{t_k + h/2 - y_k - hS_1/2}{2} \\
S_3 &= \frac{t_k + h/2 - y_k - hS_2/2}{2} \\
S_4 &= \frac{t_{k+1} - y_k - hS_3}{2}.
\end{aligned}$$

Dengan rumus-rumus di atas kita dapat menghitung hampiran-hampiran pada setiap iterasi, untuk $k = 0,\ 1,\ 2, ...,\ n(= 3/h)$,

$$y(t_{k+1}) = y(t_k) + \frac{h}{6}[S_1 + 2(S_2 + S_3) + S_4],$$

dengan $t_0 = 0$ *dan* $y_0 = 1$.

Kode MATLAB berikut ini akan menghasilkan nilai-nilai hampiran tersebut untuk lebar-lebar langkah yang ditentukan.

```
>>tn=3;h1=1;h2=1/2;h3=1/4;h4=1/8;
>>t0=0;y0=1;yh1=[t0 y0];yh2=yh1;yh3=yh1;yh4=yh1;
```

```
>>end
>>t0=0;y0=1; for k=1:tn/h3, %perhitungan untuk h3
>>t=t0+h3;
>>s1=(t0-y0)/2;s2=(t0+h3/2-y0-h3*s1/2)/2;
>>s3=(t0+h3/2-y0-h3*s2/2)/2; s4=(t-y0-h3*s3)/2;
>>y=y0+h3*(s1+2*(s2+s3)+s4)/6;yh3=[yh3;t y];t0=t;y0=y;
>>end
>>t0=0;y0=1; for k=1:tn/h4, %perhitungan untuk h4
>>t=t0+h4;
>>s1=(t0-y0)/2;s2=(t0+h4/2-y0-h4*s1/2)/2;
>>s3=(t0+h4/2-y0-h4*s2/2)/2; s4=(t-y0-h4*s3)/2;
>>y=y0+h4*(s1+2*(s2+s3)+s4)/6;yh4=[yh4;t y];t0=t;y0=y;
>>end
```

Tabel 7.2 menyajikan ringkasan hasil perhitungan dengan kode di atas. Bandingkan hasil perhitungan dengan metode RK4 dan hasil perhitugan dengan metode RK2 sebelumnya! □

LATIHAN 7.3

1. Tulis program MATLAB yang mengimplementasikan metode Heun.
2. Gunakan metode Heun (RK2) untuk mendapatkan hampiran solusi masalah-masalah nilai di bawah ini pada interval yang diberikan dan dengan lebar langkah yang ditentukan. Bandingkan nilai-nilai hampiran solusi dengan nilai-nilai fungsi solusi eksak yang diberikan. Untuk keperluan perhitungan gunakan program MATLAB yang Anda tulis.
   (a) $y' = t^2 - y, \quad y(0) = 1$ pada (i) $[0,\ 0.2]$ dengan $h = 0.2,\ 0.1,\ 0.05$ dan (ii) $[0,\ 2]$ dengan $h = 0.2,\ 0.1,\ 0.05$. Solusi eksak $y(t) = -e^{-t} + t^2 - 2t + 2$.
   (b) $y' = 3t+3y, \quad y(0) = 1$ pada (i) $[0,\ 0.2]$ dengan $h = 0.2,\ 0.1,\ 0.05$ dan (ii) $[0,\ 2]$ dengan $h = 0.2,\ 0.1,\ 0.05$. Solusi eksak $y(t) = \frac{4}{3}e^{3t} - t - \frac{1}{3}$.
   (c) $y' = -ty, \quad y(0) = 1$ pada (i) $[0,\ 0.2]$ dengan $h = 0.2,\ 0.1,\ 0.05$ dan (ii) $[0,\ 2]$ dengan $h = 0.2,\ 0.1,\ 0.05$. Solusi eksak $y(t) = e^{-t^2/2}$.

7.4 Metode Prediktor-Korektor

Metode-metode yang sudah dibahas pada bagian-bagian sebelumnya (metode Euler dan Runge–Kutta) merupakan metode-metode satu langkah untuk menyelesaiakan persamaan diferensial biasa. Sekarang kita akan membahas metode multilangkah, yang menghitung y_k dengan menggunakan gradien-gradien f_j, dengan $j < k$, yang sudah diperoleh sebelumnya. Metode ini tidak dapat mulai dengan sendirinya, karena tergantung pada metode-metode satu langkah seperti metode Euler untuk memperoleh beberapa gradien awal.

Metode prediktor-korektor terdiri atas dua bagian: (1) bagian prediktor, yang memprediksi y_k dengan menggunakan gradien-gradien f_j ($j < k$), dan (2) bagian korekor, yang menggunakan suatu rumus integrasi untuk memperbaiki hampiran.

7.4.1 Metode Trapesium–Euler

Metode trapesium–Euler menggunakan metode Euler sebagai algoritma prediktor dan aturan trapesium sebagai algoritma korektor. Misalkan kita gunakan indeks pertama untuk menunjukkan interval (langkah) dan indeks kedua untuk menunjukkan urutan hampiran. Maka rumus Euler dapat ditulis sebagai

$$y_{k+1,0} = y_{k,*} + hf_{k,*} \tag{7.27}$$

dengan tanda '0' dan '*' beturut-turut menunjukkan hampiran awal dan akhir. Pada rumus (7.27), $y_{k,*} = y_k \approx y(t_k)$, dan $f_{k,*} = f(t_k, y_k)$.

Sebagai persamaan korektor, aturan trapesium dinyatakan sebagai

$$y_{k+1,j} = y_{k,*} + \frac{h}{2}(f_{k,*} + f_{k+1,j-1}) \tag{7.28}$$

dengan j adalah penghitung iterasi proses koreksi dan $f_{k+1,j-1} = f(t_{k+1}, y_{k+1,j-1})$. Persamaan korektor digunakan sebanyak yang diperlukan untuk mendapatkan keakuratan yang diinginkan. Perhatikan bahwa, dengan menggunakan (7.27) sebagai nilai awal, $y_{k+1,j}$ dapat dihitung untuk $j = 1,\ 2,\ 3,\ \ldots$ dengan rumus (7.28). Proses koreksi dapat dihentikan setelah iterasi ke–n (ditentukan) atau setelah $|y_{k+1,j+1} - y_{k+1,j}| < \epsilon$, untuk suatu nilai ϵ yang ditentukan.

penyelesaian PD di atas dengan metode Euler.

```
>>a=1;b=2;h=0.1;y0=1;
>>xy=[a y0];
>>for t=a+h:h:b,y=y0+h*t*sqrt(y0);
  xy=[xy;t y];y0=y;end
>>xy
 xy  =
   1.     1.
   1.1    1.11
   1.2    1.2364278
   1.3    1.3809811
   1.4    1.5455023
   1.5    1.7319796
   1.6    1.9425471
   1.7    2.1794851
   1.8    2.4452205
   1.9    2.7423274
   2.     3.0735268
```

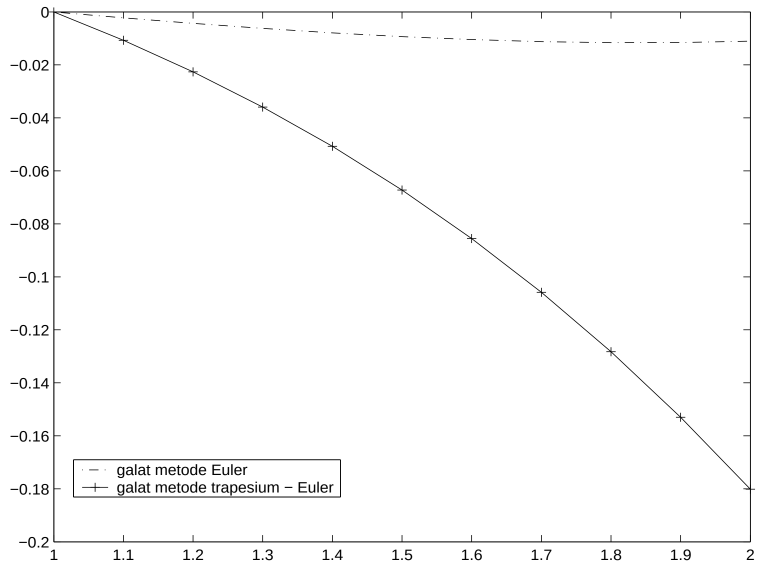
Bandingkan nilai-nilai tersebut dengan nilai-nilai penyelesaian eksak y_1:

x	$y1 = (\frac{x^2+3}{4})^2$
1.	1.
1.1	1.1077562
1.2	1.2321
1.3	1.3747563
1.4	1.5376
1.5	1.7226562
1.6	1.9321
1.7	2.1682562
1.8	2.4336
1.9	2.7307563
2.	3.0625

Gambar 7.9 menyajikan grafik hampiran penyelesaian dengan metode Euler dan kurva penyelesaian eksak y_1.

```
1.1    1.118454
1.2    1.2547172
1.3    1.4106661
1.4    1.5883275
1.5    1.7898779
1.6    2.0176438
1.7    2.2741018
1.8    2.5618783
1.9    2.8837496
2.     3.2426423
```

Perbandingan hasil metode trapesium–Euler dan penyelesaian eksak dapat dilihat pada Gambar 7.10.



Gambar 7.11: Perbandingan galat metode Euler dan galat metode trapesium–Euler terhadap solusi eksak PD $\frac{dy}{dx} = x\sqrt{y}$ *dengan syarat awal* $y(1) = 1$.

Dengan membandingkan Gambar 7.9 dan Gambar 7.10 kita melihat bahwa titik-titik hampiran dengan metode trapesium – Euler lebih jauh daripada titik-titik hampiran dengan metode Euler ke kurva solusi eksak. Hal ini menunjukkan

9. Berdasarkan uraian pada subbab 7.4, turunkan rumus Adams–Bashforth dua–langkah, tiga–langkah, dan rumus Adams–Moulton dua–langkah.

7.5 Metode Galerkin

7.5.1 Metode Kuadrat Terkecil

Misalkan persamaan diferensial yang hendak diselesaikan dinyatakan sebagai

$$\frac{dy}{dx} = f(x, y)$$

dengan syarat awal $y(a) = c$.

Metode kuadrat terkecil dapat dijelaskan sebagai berikut.

1. Anggap penyelesaian PD di atas merupakan suatu fungsi diferensiabel, misalnya, berbentuk polinomial berderajad n:

   $$y^* = b_0 + b_1 x + b_2 x^2 + \ \dots\ + b_n x^n$$

   Kita ingin menentukan koefisien-koefisien $b_0,\ b_1,\ b_2,\ \dots,\ b_n$.

2. Gunakan syarat awal untuk menghitung salah satu koefisien atau untuk memperoleh suatu syarat untuk mendapatkan koefisien-koefisien tersebut.

3. Misalkan e adalah galat di dalam asumsi ini, yakni selisih antara turunan-turunan tersebut:

   $$e = \frac{dy^*}{dx} - \frac{dy}{dx}$$

   Definisikan fungsi kuadrat terkecil g sebagai integral kuadrat galat tersebut, yakni

   $$g = \int_x e^2\, dx$$

4. Masalah kita adalah mencari nilai-nilai $b_0,\ b_1,\ b_2,\ \dots,\ b_n$ yang meminimumkan g, yakni sedemikian hingga

   $$\frac{\partial g}{\partial b_i} = \int_x 2e \frac{\partial e}{\partial b_i}\, dx = 0, \quad \text{untuk} \quad i = 1,\ 2,\ \dots,\ n.$$

Tabel 7.3: Penyelesaian eksak dan metode kuadrat terkecil PD $y' = xy,\ y(0) = 1$ pada $[0,\ 1]$.

x	y^*	y	$\|y - y^*\|$	$\frac{\|y-y^*\|}{y}100\%$
0	1	1	0	0
0.1	1.04688	1.00501	0.0418625	4.16537
0.2	1.09375	1.0202	0.0735487	7.20923
0.3	1.14063	1.04603	0.0945971	9.04346
0.4	1.1875	1.08329	0.104213	9.62007
0.5	1.23438	1.13315	0.101227	8.93321
0.6	1.28125	1.19722	0.0840326	7.019
0.7	1.32813	1.27762	0.0505037	3.95295
0.8	1.375	1.37713	0.00212776	0.154507
0.9	1.42187	1.4993	0.0774275	5.16423
1.	1.46875	1.64872	0.179971	10.9158

berikan suatu penyelesaian.

7.5.2 Metode Galerkin

Metode ini sangat mirip dengan metode kuadrat terkecil. Perbedaannya pada persamaan normal, yang diberikan oleh

$$\int_x ew_i\,dx = 0 \qquad \text{untuk} \qquad i = 1,\ 2,\ 3,\ ...,\ n.$$

Khususnya, untuk hampiran dengan polinomial berderajad n kita mempunyai persamaan normal

$$\int_x ex^i\,dx = 0 \qquad \text{untuk} \qquad i = 1,\ 2,\ 3,\ ...,\ n.$$

CONTOH 7.15.
Carilah penyelesaian persamaan diferensial

$$\frac{dy}{dx} = xy, \qquad \textit{dengan syarat awal} \qquad y(0) = 1$$

dengan menggunakan metode Galerkin.

Tabel 7.5: Penyelesaian eksak, metode kuadrat terkecil, dan metode Galerkin PD $y' = xy,\ y(0) = 1$ pada $[0,\ 1]$.

x	y^*	y	$\lvert y - y^*\rvert$	$\frac{\lvert y-y^*\rvert}{y}100\%$
0	1	1	0	0
0.1	0.982237	1.00501	0.0227757	2.26621
0.2	0.981579	1.0202	0.0386224	3.78576
0.3	0.998026	1.04603	0.0480015	4.58894
0.4	1.03158	1.08329	0.0517081	4.77326
0.5	1.08224	1.13315	0.0509116	4.49293
0.6	1.15	1.19722	0.0472174	3.94393
0.7	1.23487	1.27762	0.0427529	3.34629
0.8	1.33684	1.37713	0.0402857	2.92534
0.9	1.45592	1.4993	0.0433814	2.89344
1.	1.59211	1.64872	0.056616	3.43393

LATIHAN 7.5

1. Tulis algoritma kuadrat terkecil untuk menyelesaikan masalah nilai awal
$$y'(t) = f(t, y(t)), \qquad y(t_0) = y_0$$
dengan menganggap solusi PD di atas merupakan polinomial berderajad n. Implementasikan algoritma tersebut dengan program MATLAB.

2. Gunakan metode kuadrat terkecil untuk menyelesaikan masalah-masalah nilai awal berikut ini dengan menggukana hampiran fungsi solusi berupa polinomial berderajad n, untuk nilai n yang diberikan.

   (a) $y' = xy^2, \quad y(0) = 1, \quad n = 1.$

   (b) $y' = xy^2, \quad y(0) = 1, \quad n = 2.$

   (c) $y' = x + y^2, \quad y(0) = 1, \quad n = 3.$

   (d) $y' = x - y, \quad y(0) = 1, \quad n = 1.$

   (e) $y' = x + xy, \quad y(0) = 1, \quad n = 2.$

CONTOH 7.16.
Selesaikan persamaan diferensial

$$y'' - 2y' + 2y = e^{2t}\sin t \quad 0 \le t \le 1; \quad y(0) = -0.4; \quad y'(0) = -0.6.$$

Penyelesaian:

Dengan $u_1(t) = y(t)$ *dan* $u_2(t) = y'(t)$, *PD di atas dapat diubah ke dalam sistem PD*

$$\frac{du_1(t)}{dt} = u_2(t) = f_1(t, u_1, u_2) \tag{D1}$$

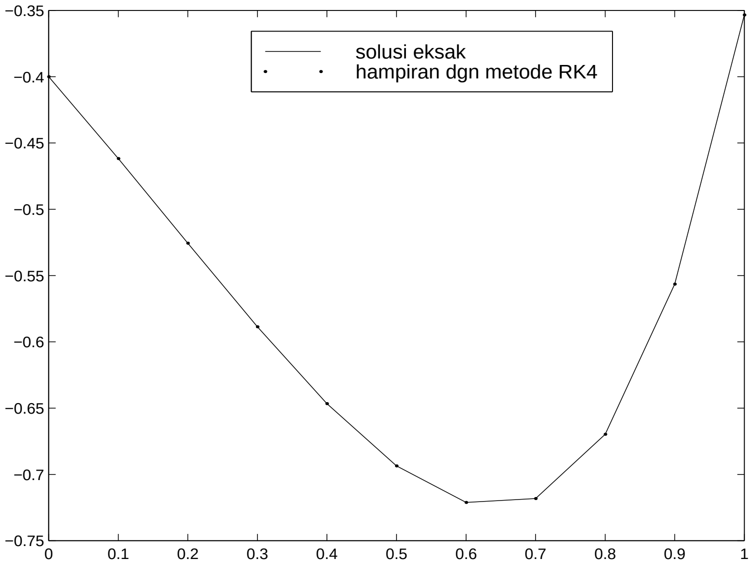
$$\frac{du_2(t)}{dt} = e^{2t}\sin t - 2u_1(t) + 2u_2(t) = f_2(t, u_1, u_2) \tag{D2}$$

dengan syarat awal $u_1(0) = -0.4$, $u_2(0) = -0.6$.

Kita coba selesaikan sistem PD di atas dengan menggunakan metode Runge–Kutta orde empat dengan lebar subinterval $h = 0.1$. *Misalkan* $S_{11},\ S_{21},\ S_{31},\ S_{41}$ *adalah gradien-gradien yang perlu dihitung untuk PD (D1) pada iterasi pertama. Misalkan* $S_{12},\ S_{22},\ S_{32},\ S_{42}$ *adalah gradien-gradien yang perlu dihitung untuk PD (D2) pada iterasi pertama. Misalkan juga* y_{1i} *dan* y_{2i} *berturut-turut menyatakan nilai-nilai* y *dan* y' *pada iterasi ke-*i*. Maka kita punya syarat awal* $y_{10} = -0.4$ *dan* $y_{20} = -0.6$*. Pada iterasi pertama kita hitung,*

$$\begin{aligned}
S_{11} &= hf_1(t_0, y_{10}, y_{20}) = hy_{20} = (0.1)(-0.6) = -0.06\\
S_{12} &= hf_2(t_0, y_{10}, y_{20}) = h[e^{2t_0}\sin t_0 - 2y_{10}(t) + 2y_{20}(t)] = -0.04\\
S_{21} &= hf_1(t_0 + h/2, y_{10} + S_{11}/2, y_{20} + S_{12}/2)\\
&= h[y_{20} + S_{12}/2] = (0.1)(-0.6) = -0.062\\
S_{22} &= hf_2(t_0 + h/2, y_{10} + S_{11}/2, y_{20} + S_{12}/2)\\
&= h[e^{2(t_0+0.05)}\sin(t_0 + 0.05) - 2(y_{10} + S_{11}/2) + 2(y_{20} + S_{12}/2)]\\
&= -0.0324764\\
S_{31} &= hf_1(t_0 + h/2, y_{10} + S_{21}/2, y_{20} + S_{22}/2)\\
&= h[y_{20} + S_{22}/2] = -0.06162382238\\
S_{32} &= hf_2(t_0 + h/2, y_{10} + S_{21}/2, y_{20} + S_{22}/2)\\
&= h[e^{2(t_0+0.05)}\sin(t_0 + 0.05) - 2(y_{10} + S_{21}/2) + 2(y_{20} + S_{22}/2)]\\
&= -0.0315241
\end{aligned}$$

```
>>for t=a:h:b,
>>S11=h*y2;S12= h*(exp(2*t)*sin(t)-2*y1+2*y2);
>>S21=h*(y2+S12/2);
>>S22= h*(exp(2*(t+0.05))*sin(t+0.05)-2*(y1+S11/2)...
  +2*(y2+S12/2));
>>S31=h*(y2+S22/2);
>>S32= h*(exp(2*(t+0.05))*sin(t+0.05)-2*(y1+S21/2)...
  +2*(y2+S22/2));
>>S41=h*(y2+S32/2);
>>S42= h*(exp(2*(t+0.1))*sin(t+0.1)-2*(y1+S31)...
  +2*(y2+S32));
>>y1i = y1 + (S11+2*S21+2*S31+S*41)*h/6;
>>y2i = y2 + (S12+2*S22+2*S32+S*42)*h/6;
>>ty1y2=[ty1y2;t y1i y2i];
>>y1=y1i;y2=y2i;
>>end
```



Gambar 7.12: Kurva penyelesaian eksak dan hampiran solusi PD $y'' - 2y' + 2y = e^{2t} \sin t \quad 0 \leq t \leq 1; \quad y(0) = -0.4; \quad y'(0) = -0.6.$

3. Ubahlah masalah-masalah nilai awal di bawah ini menjadi sistem persamaan diferensial orde satu, serta berikan nilai-nilai awalnya.

   (a) Persamaan Bessel,

   $$xy'' + y' + kxy = 0, \quad y(0) = 1,\ y'(0) = 0;$$

   (b) Ayunan teredam,

   $$mx'' + cx' + kx = F_0 \sin(\omega t), \quad x(0) = x_0,\ x'(0) = 0;$$

   (c) Persamaan Duffing,

   $$x'' + cx' + \alpha x + \beta x^3 = F_0 \cos(\omega t), \quad x(0) = x_0,\ x'(0) = 0;$$

4. Selesaikan persamaan van der Pol di bawah ini dengan menggunakan metode RK4. Gunakan lebar langkah $h = 0.1$.

   $$y'' - (.1)(1 - y^2)y' + y = 0, \quad y(0) = 1,\ y'(0) = 0$$

5. Ubahlah sistem persamaan diferensial

   $$y' = p, \quad p' = -100y - 101p, \quad y(0) = 1, \quad p(0) = -1,$$

   menjadi sebuah persamaan diferensial (masalah nilai awal) berorde dua. Selesaikan sistem PD tersebut dengan metode RK4 dengan lebar langkah $h = 0.01$ Bandingkan hasil perhitungan tersebut dengan solusi eksak $y = e^{-x}$.

6. Seekor anjing, keluar dari rumahnya, karena melihat tuannya berjalan sepanjang lorong ia mengejarnya. Dengan asumsi anjing tersebut selalu lari ke arah tuannya dan lorong tersebut lurus, persamaan lintasan anjing dapat diturunkan dari PD

   $$xy'' = c\sqrt{1 + (y')^2}$$

   dengan c adalah rasio kecepatan tuan dan anjingnya. Di sini x menyatakan jarak anjing ke tuannya dan y adalah jarak yang ditempuh oleh tuan anjing. Pertanyaannya adalah setelah tuan berjalan seberapa jauh, anjing tersebut mencapai tuannya? Penyelesaian eksak PD masalah-masalah persamaan diferensial biasa. Uraian ini didasarkan pada panduan online (*help*) MATLAB versi 6.1.

Tabel 7.7 menyajikan daftar solver (fungsi) MATLAB untuk menyelesaikan masalah-masalah nilai awal, jenis (kategori masalah) dan metode yang dipakai.

Tabel 7.7: Fungsi-fungsi MATLAB untuk menyelesaikan masalah-masalah PD

Solver	**Jenis masalah yang dapat diselesaikan**	**Metode**
`ode45`	PD non-stiff	Runge – Kutta
`ode23`	PD non-stiff	Runge – Kutta
`ode113`	PD non-stiff	Adam
`ode15s`	PD stiff dan PD aljabar (DAE)	NDF (BDF)
`ode23s`	PD stiff	Rosenbrock
`ode23t`	PD stiff moderat dan DAE	Aturan Trapesium
`ode23tb`	PD stiff	TR-BDF2

Tentang metode penyelesaian

Fungsi `ode45` didasarkan pada rumus eksplisit Runge – Kutta (4,5), pasangan Dormand – Prince, dan merupakan metode satu-langkah – untuk menghitung penyelesaian $y(t_n)$ digunakan satu penyelesaian pada titik waktu sebelumnya, $y(t_{n-1})$. Secara umm, `ode45` adalah pemecah PD terbaik untuk "pertama kali dicoba" digunakan pada kebanyakan PD biasa.

Fungsi `ode23` merupakan implementasi rumus eksplisit Runge-Kutta (2,3) pasangan Bogacki dan Shampine. Metode ini mungkin lebih efisien daripada `ode45` pada toleransi kasar dan pada masalah yang agak stiff. Seperti `ode45`, `ode23` adalah implementasi metode satu-langkah.

Fungsi `ode113` adalah implementasi metode Adams – Bashforth – Moulton PECE. Metode ini mempunyai orde variabel dan mungkin lebih efisien daripada `ode45` dengan toleransi yang ketat dan apabila fungsi PD-nya susah dihitung.

Fungsi `ode113` menggunakan metode multilangkah – biasanya untuk menghitung suatu penyelesaian diperlukan beberapa penyelesaian sebelumnya.

Kahaner, D. , C. Moler, and S. Nash, *Numerical Methods and Software*, Prentice-Hall, New Jersey, 1989.

Shampine, L. F. , *Numerical Solution of Ordinary Differential Equations*, Chapman & Hall, New York, 1994.

Shampine, L. F. and M. K. Gordon, *Computer Solution of Ordinary Differential Equations: the Initial Value Problem*, W. H. Freeman, SanăFrancisco, 1975.

Shampine, L. F. and M. E. Hosea, "Analysis and Implementation of TR-BDF2," *Applied Numerical Mathematics* 20, 1996.

Shampine, L. F. and M. W. Reichelt, "The MATLAB ODE Suite," *SIAM Journal on Scientific Computing*, Vol. 18, 1997, pp 1-22.

Shampine, L. F., M. W. Reichelt, and J.A. Kierzenka, "Solving Index-1 DAEs in MATLAB and Simulink," *SIAM Review*, Vol. 41, 1999, pp 538-552.

Sintaks pemakaian:

```
[T,Y] = solver(fungsiPD,tspan,y0)
[T,Y] = solver(fungsiPD,tspan,y0,opsi)
[T,Y] = solver(fungsiPD,tspan,y0,opsi,p1,p2...)
[T,Y,TE,YE,IE] = solver(fungsiPD,tspan,y0,opsi)
sol = solver(fungsiPD,[t0 tf],y0...)
```

dengan `solver` adalah salah satu di antara `ode45, ode23, ode113, ode15s, ode23s, ode23t`, atau `ode23tb`.
Masukan:

`fungsiPD` : Fungsi MATLAB yang mendefinisikan PD. Semua solver MATLAB dapat menyelesaikan sistem PD dalam bentuk $y' = f(t,y)$ atau masalah PD yang melibatkan matriks massa, $M(t,y)y' = f(t,y)$. Solver `ode23s` hanya dapat menyelesaikan PD dengan matriks massa konstan, sedangkan solver `ode15s` dan `ode23t` dapat menyelesaikan masalah PD dengan matriks massa singular, misalnya PD aljabar (DAE).

`tspan` : Suatu vektor yang menyatakan interval integrasi. Solver akan menyimpan syarat awal pada `tspan(1)` dan mengintegralkan dari

Tabel 7.8: Parameter Opsional pada Solver PD MATLAB

Parameter	ode45	ode23	ode113	ode15s	ode23s	ode23t	ode23tb
RelTol AbsTol NormControl	v	v	v	v	v	v	v
OutputFcn OutputSel Refine Stats	v	v	v	v	v	v	v
Events	v	v	v	v	v	v	v
MaxStep InitialStep	v	v	v	v	v	v	v
Jacobian JPattern Vectorized	-	-	-	v	v	v	v
Mass	v	v	v	v	v	v	v
MStateDependence	v	v	v	v	-	v	v
MvPattern	-	-	-	v	-	v	v
MassSingular	-	-	-	v	-	v	-
InitialSlope	-	-	-	v	-	v	-
MaxOrder, BDF	-	-	-	v	-	-	-

tingkat satu yang ekivalen dengan PD semula, dalam bentuk $y' = f(t, y)$. Setiap PD biasa berbentuk $y^{(n)} = f(t, y, y', y'', ..., y^{(n-1)})$ dapat dinyatakan sebagai sebuah sistem PD tingkat satu dengan menggunakan substitusi

$$y_1 = y, \quad y_2 = y', ..., \quad y_n = y^{(n-1)}.$$

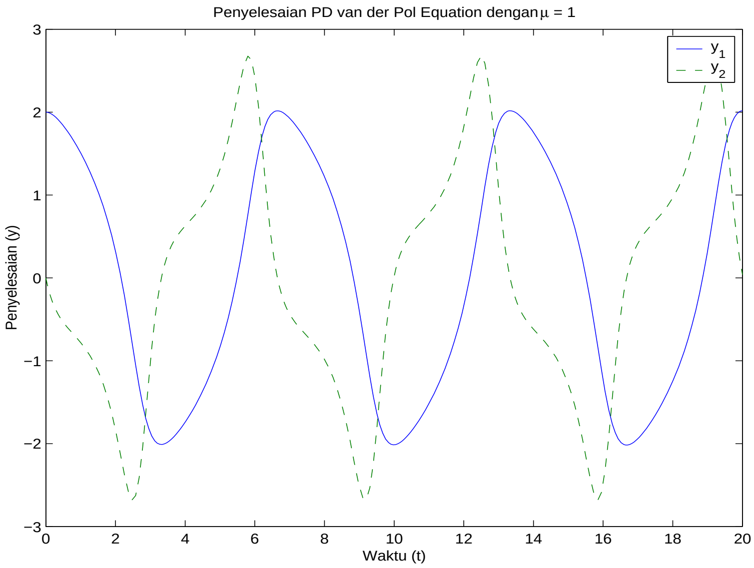
Hasilnya adalah sistem n PD tingkat satu:

$$\begin{aligned} y_1' &= y_2 \\ y_2' &= y_3 \\ &\vdots \\ y_n' &= f(t, y_1, y_2, ..., y_n). \end{aligned}$$

Sebagai contoh, PD *van der Pol* (PD tingkat dua) $y_1'' - \mu(1 - y_1^2)y_1' + y_1 = 0$ dengan μ adalah suatu parameter skalar, dengan substitusi $y_1' = y_2$, dapat diubah ke sistem 2 PD tingkat satu:

$$\begin{aligned} y_1' &= y_2 \\ y_2' &= \mu(1 - y_1^2)y_2 - y_1. \end{aligned}$$

```
>>plot(t,y(:,1),'-',t,y(:,2),'--')
>>title('Penyelesaian PD van der Pol Equation
  dengan \mu = 1');
>>xlabel('Waktu (t)');
>>ylabel('Penyelesaian (y)');
>>legend('y_1','y_2')
```



Gambar 7.13: Penyelesaian sistem $y_1' = y_2,\ y_2' = (1 - y_1^2)y_2 - y_1,\ y_1(0) = 2,\ y_2(0) = 0$

Menambahkan Parameter Tambahan ke Fungsi PD

Solver MATLAB akan menggunakan parameter-parameter tambahan setelah argumen `opsi` ke fungsi PD dan fungsi lain yang dinyatakan dalam `opsi`. Sebagai contoh:

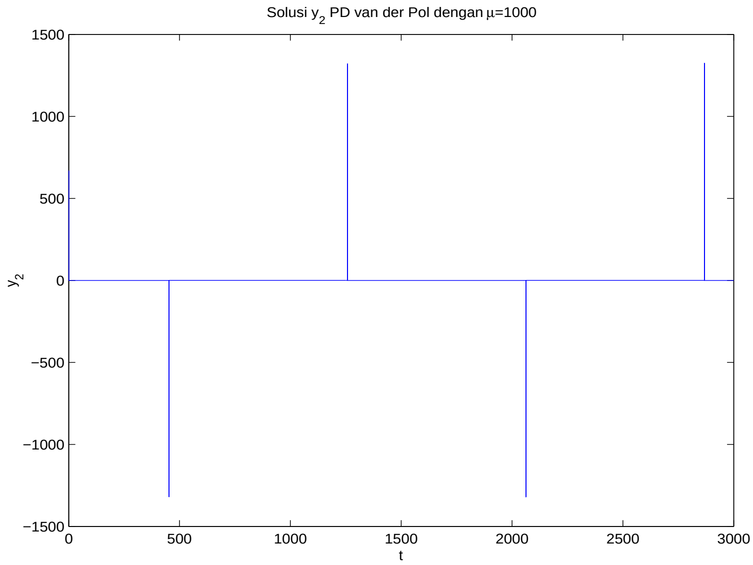
1. Perumum fungsi PD *van der Pol* dengan menambahkan parameter konstanta `mu`, sebagai ganti menuliskan nilainya pada definisi fungsi.

Sistem PD tersebut dalam MATLAB dapat didefinisikan dengan fungsi `vdp1000` *dalam berkas M-file* `vdp1000.m` *yang isinya sebagai berikut.*

```
function dy = vdp1000(t,y)
dy = zeros(2,1);    % vektor kolom
dy(1) = y(2);
dy(2) = 1000*(1 - y(1)^2)*y(2) - y(1);
```

Dalam contoh ini akan digunakan nilai-nilai batas toleransi galat relatif dan galat mutlak aslinya (yakni `1e-3` *dan* `1e-6`*) dan menyelesaikannya pada interval waktu* `[0 3000]` *dengan vektor syarat awal* `[0 1]` *pada waktu* `0`*. Solver yang digunakan adalah* `ode15s`*, karena masalahnya bersifat stiff.*

```
>>[T,Y] = ode15s(@vdp1000,[0 3000],[0 1]);
```



Gambar 7.16: Solusi y_2 *sistem* $y_1' = y_2,\ y_2' = 1000(1 - y_1^2)y_2 - y_1,\ y_1(0) = 0,\ y_2(0) = 1$

Selanjutnya, penyelesaian yang didapat diplot, yakni kolom-kolom matriks `Y` *terhadap vektor kolom* `T`*. Oleh karena nilai-nilai* y_2 *dan* y_1 *mempunyai perilaku yang berlainan sekali, maka keduanya diplot terpisah. Hasilnya ditunjukkan pada Gambar 7.15 dan 7.16.*

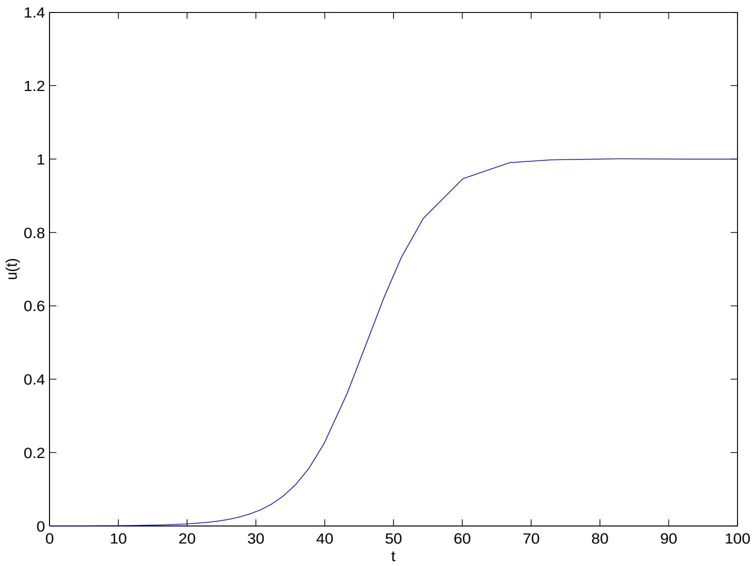
Dengan MATLAB kita dapat menghitung hampiran nilai-nilai $u(t)$ untuk $t > 0$. Dengan demikian kita dapat menggambar kurva penyelesaian PD (7.37)*. Berikut disajikan dua perintah MATLAB untuk menyelesaiakan PD tersebut, yakni perintah* `ode23` *dan* `ode45`*. Misalkan kita menginginkan penyelesaian pada interval* `[0, 100]`*.*

Mula-mula didefinisikan fungsi PD (7.37)*, yang dinyatakan dalam variabel t dan u, pada M-file* `infeksi.m`*.*

```
function du=infeksi(t,u)
k=0.2; du=k*u*(1-u);
```

(Sekalipun pada perhitungan fungsi di atas tidak memerlukan variabel `t`*, namun variabel* `t` *diperlukan oleh perintah* `ode23` *dan* `ode45`*.) Berikut adalah penyelesaiannya.*

```
>>[t,u]=ode23(@infeksi,[0,100],1/10000);
>>plot(t,u)
```



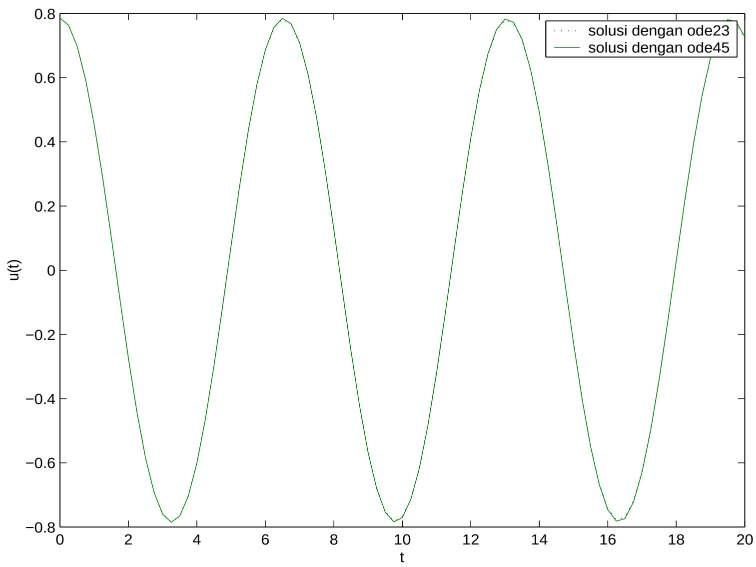
Gambar 7.18: Solusi $u'(t) = 0.2u(t)(1 - u(t)), \quad u(0) = 1/10000$ dengan `ode23`

Fungsi `ode23` *di atas menghitung nilai-nilai* `u` *(fungsi yang merupakan penyelesaian PD) pada nilai-nilai* `t` *di dalam interval* `[0, 100]` *berdasarkan nilai*

```
dY=[Y(2);-sin(Y(1))];
```

Akhirnya kita dapat menggunakan perintah `ode23` *atau* `ode45` *untuk mendapatkan penyelesainnya.*

```
>>[t,Y]=ode23(@pdtk2,[0:.25:20],[pi/4;0]);
>>[t1,Y1]=ode45(@pdtk2,[0:.25:20],[pi/4;0]);
>>u=Y(:,1);u1=Y1(:,1);plot(t,u,':',t1,u1);
```



Gambar 7.21: Penyelesaian PD $u'' + \sin(u) = 0,\ u(0) = \pi/4,\ u'(0) = 0$ *dengan* `ode23` *dan* `ode45`

Pada Gambar 7.21 tampak kurva kedua penyelesaian berimpit. Jadi kedua perintah dapat dianggap memberikan penyelesaian yang sama. □

Latihan 7.7

1. Tunjukkan bahwa (7.39) merupakan penyelesaian PD (7.37) dengan nilai awal (7.38).

(c) Apabila pendulum tersebut berayun melalui sudut-sudut kecil, persamaan (7.40) dapat dihampiri oleh

$$\frac{d^2u}{dt^2} + u(t) = 0, \qquad (7.44)$$
$$u(0) = \pi/4, \quad \frac{du}{dt}(0) = 0.$$

Selesaikan PD (7.44) dan bandingkan hasilnya dengan penyelesaian PD (7.40). (Gambarlah kurva kedua penyelesaian pada sumbu koordinat yang sama.)

7.8 Rangkuman

Sebagai akhir bab ini berikut disajikan beberapa konsep dasar tentang persamaan diferensial, masalah nilai awal, dan beberapa metode numerik untuk mendapatkan hampiran penyelesaian masalah nilai awal.

Persamaan diferensial (PD) biasa tingkat satu adalah persamaan yang berbentuk

$$y'(t) = f(t, y(t)), \qquad t \geq t_0,$$

dengan $y(t)$ adalah fungsi tak diketahui yang hendak dicari dan f adalah fungsi dua variabel (dalam t dan y). Penyelesaian eksak suatu PD adalah fungsi $y(t)$ yang memenuhi persamaan tersebut.

Masalah nilai awal adalah suatu persamaan diferensial yang dilengkapi dengan syarat nilai awal untuk fungsi yang tak diketahui,

$$y'(t) = f(t, y(t)), \quad t \geq t_0, \quad y(t_0) = y_0.$$

Penyelesaian numerik suatu masalah nilai awal adalah himpunan titik-titik $(t_k,\ y_k)$, untuk $k = 1,\ 2,\ ..., n$, dengan $y_k \approx y(t_k)$, jika $y(t)$ adalah penyelesaian eksaknya.

Syarat Lipschitz Misalkan $R = \{(t, y) | \, a \leq t \leq b, \ c \leq y \leq d\}$ sebuah persegi panjang, dan $f(t, y)$ adalah suatu fungsi yang kontinyu pada R. Fungsi f dikatakan memenuhi **syarat Lipschitz** terhadap variabe

Metode Adams – Bashforth Empat-Langkah untuk menghitung hampiran penyelesaian masalah nilai awal $y' = f(t, y)$ dengan $y(t_0) = y_0$ pada $[t_0,\ b]$:

$$\begin{aligned} t_k &= t_{k-1} + h, \\ y_{k+1} &= y_k + \frac{h}{24}[55f_k - 59f_{k-1} + 37f_{k-2} - 9f_{k-3}] \\ &\quad \text{untuk setiap} \quad k = 3,\ 4,\ 5,\ ..., \end{aligned}$$

dengan h adalah lebar langkah yang diberikan, $f_i = f(t_i, y_i)$. Nilai y_1, y_2, dan y_3 dihitung dengan metode lain, misalnya metode RK4.

Metode Adams – Moulton Dua-Langkah untuk menghitung hampiran penyelesaian masalah nilai awal $y' = f(t, y)$ dengan $y(t_0) = y_0$ pada $[t_0,\ b]$:

$$\begin{aligned} t_k &= t_{k-1} + h, \quad k = 1,\ 2,\ 3,\ ..., \\ y_{k+1,0} &= y_k + \frac{h}{12}[23f_k - 16f_{k-1} + 5f_{k-2}), \quad k = 2,\ 3,\ ... \\ y_{k+1,j} &= y_k + \frac{h}{12}[5f(t_{k+1}, y_{k+1,j-1}) + 8f_k - f_{k-1}],\ j = 1,\ 2,\ ...,\ m, \\ y_{k+1} &= y_{k+1,m} \ni |y_{k+1,m} - y_{k+1,m-1}| < \epsilon \\ &\quad \text{untuk setiap} \quad k = 2,\ 3,\ ..., \end{aligned}$$

dengan h adalah lebar langkah dan ϵ adalah batas toleransi yang diberikan, $f_i = f(t_i, y_i)$. Nilai-nilai y_1 dan y_2 dihitung dengan salah satu metode lain (misalnya RK4).

Metode Adams – Moulton Tiga-Langkah untuk menghitung hampiran penyelesaian masalah nilai awal $y' = f(t, y)$ dengan $y(t_0) = y_0$ pada $[t_0,\ b]$:

$$\begin{aligned} t_k &= t_{k-1} + h, \quad k = 1,\ 2,\ 3,\ ..., \\ y_{k+1,0} &= y_k + \frac{h}{24}[55f_k - 59f_{k-1} + 37f_{k-2} - 9f_{k-3}), \quad k = 3,\ 4,\ ... \\ y_{k+1,j} &= y_k + \frac{h}{24}[9f(t_{k+1}, y_{k+1,j-1}) + 19f_k - 5f_{k-1} + f_{k-2}], \\ &\quad j = 1,\ 2,\ ...,\ m, \quad \text{dan} \\ y_{k+1} &= y_{k+1,m} \ni |y_{k+1,m} - y_{k+1,m-1}| < \epsilon \\ &\quad \text{untuk setiap} \quad k = 3,\ 4,\ ..., \end{aligned}$$