LAPORAN TUGAS KECIL

IF5162/Metode Numerik Lanjut

Dipersiapkan oleh:

Gugy Lucky Khamdani / 23517041

Sekolah Teknik Elektro dan Informatika - Institut Teknologi Bandung

Jl. Ganesha 10, Bandung 40132

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Sekolah Teknik Elektro dan Informatika ITB** | Nomor Dokumen | | Halaman |
| *IF5162/23517041* | | *24* |
| *Revisi* | *1* | *18 April 2018* |

1. Secara umum, digunakan metode numerik untuk menghampiri solusi dari . Dengan menggunakan metode Euler, metode Runge-Kutta orde 2, metode Runge-Kutta orde 4, dan metode banyak-langkah Adams-Bashforth-Moulton untuk menghampiri persamaan diferensial

dengan kondisi awal

persamaan diferensial tersebut diselesaikan dengan waktu mencapai 2.0, dengan jumlah titik .

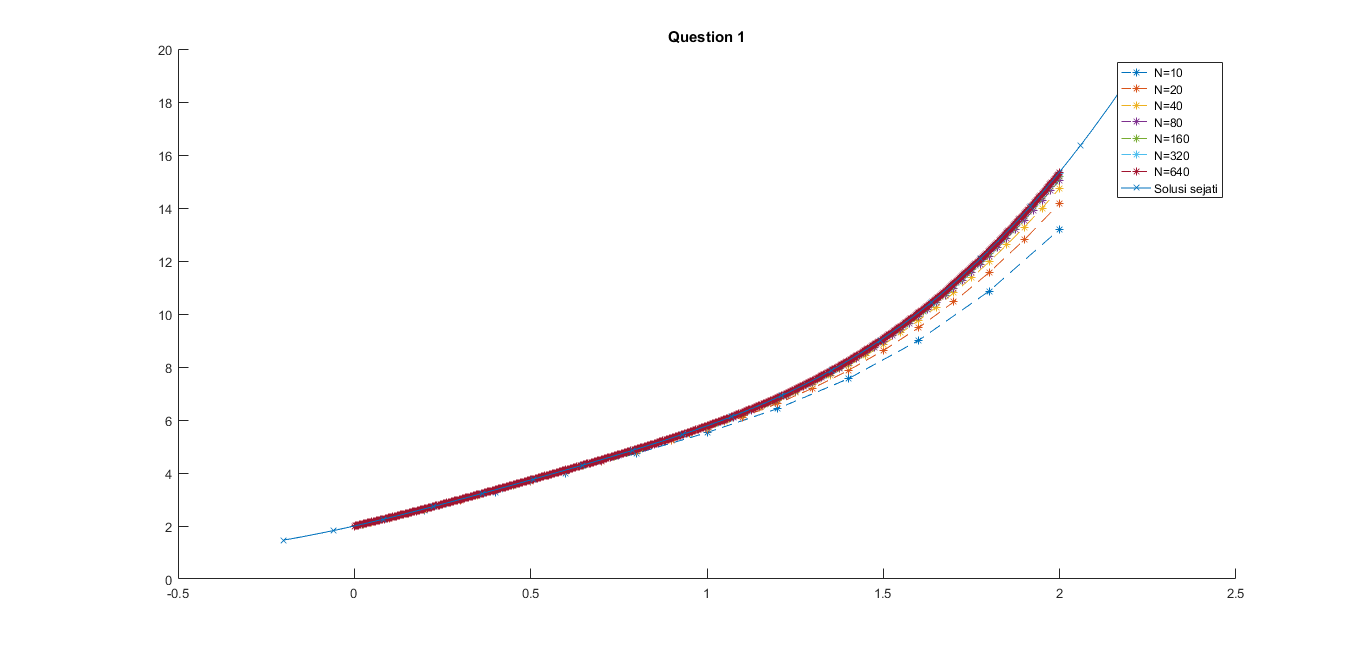
Solusi sejati dari persamaan diferensial tersebut adalah

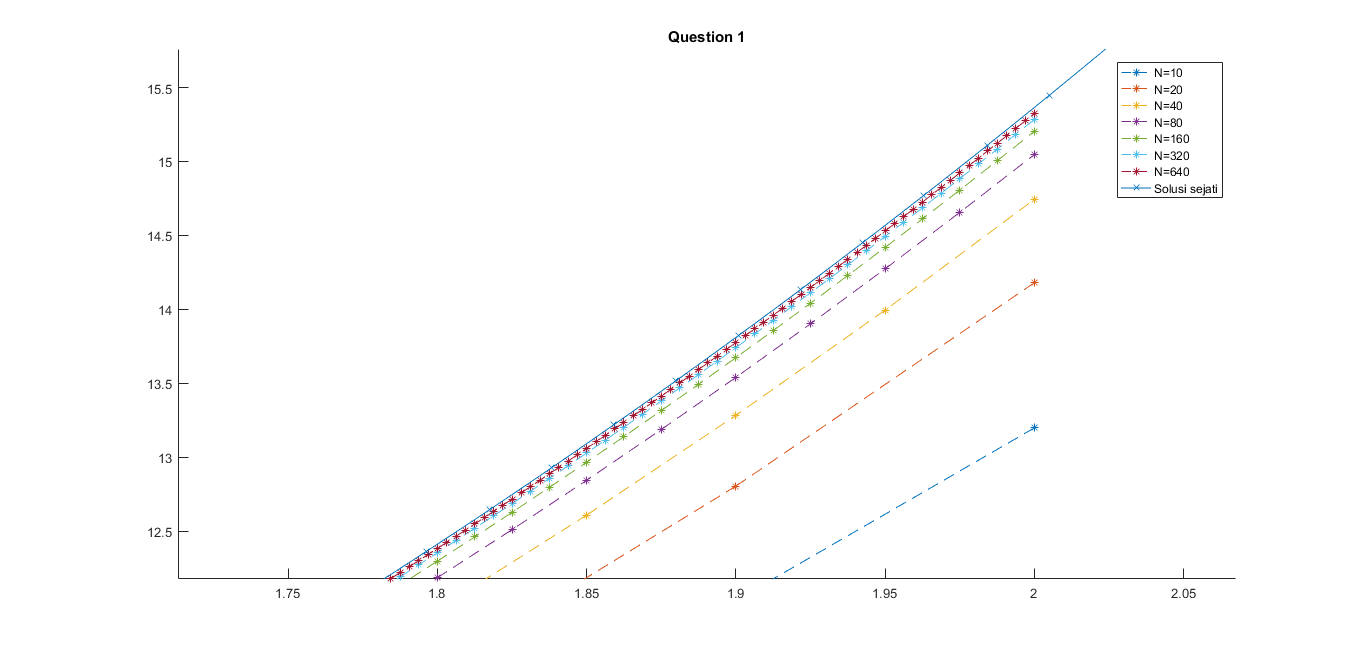
Table 1.1 galat dari hasil pendekatan menggunakan metode Euler, Runge-Kutta Orde 2, Runge-Kutta 4, dan Adams-Bashforth-Moulton

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **N** | **Euler** | **Runge-Kutta orde 2** | **Runge-Kutta orde 4** | **Adams-Bashforth-Moulton** |
| 10  20  40  80  160  320  640 | 2.16780485772888  1.18596688019685  0.622253520635988  0.318997148031420  0.161541515460526  0.0812913388496440  0.0407770734612680 | 0.168116170199903  0.0452148761098119  0.0117093652187119  0.00297833426775362  0.000750969069558849  0.000188540944016324  4.72350938469646e-05 | 0.000414507642171103  2.73632829763670e-05  1.75969036142476e-06  1.11591541340772e-07  7.02583058398432e-09  4.40696368286808e-10  2.75175437991493e-11 | 0.00338930652274172  0.000217154875313597  7.26498145198207e-06  1.21881623016407e-07  -5.04256902900124e-09  -7.47194306427446e-10  -6.07940364716342e-11 |

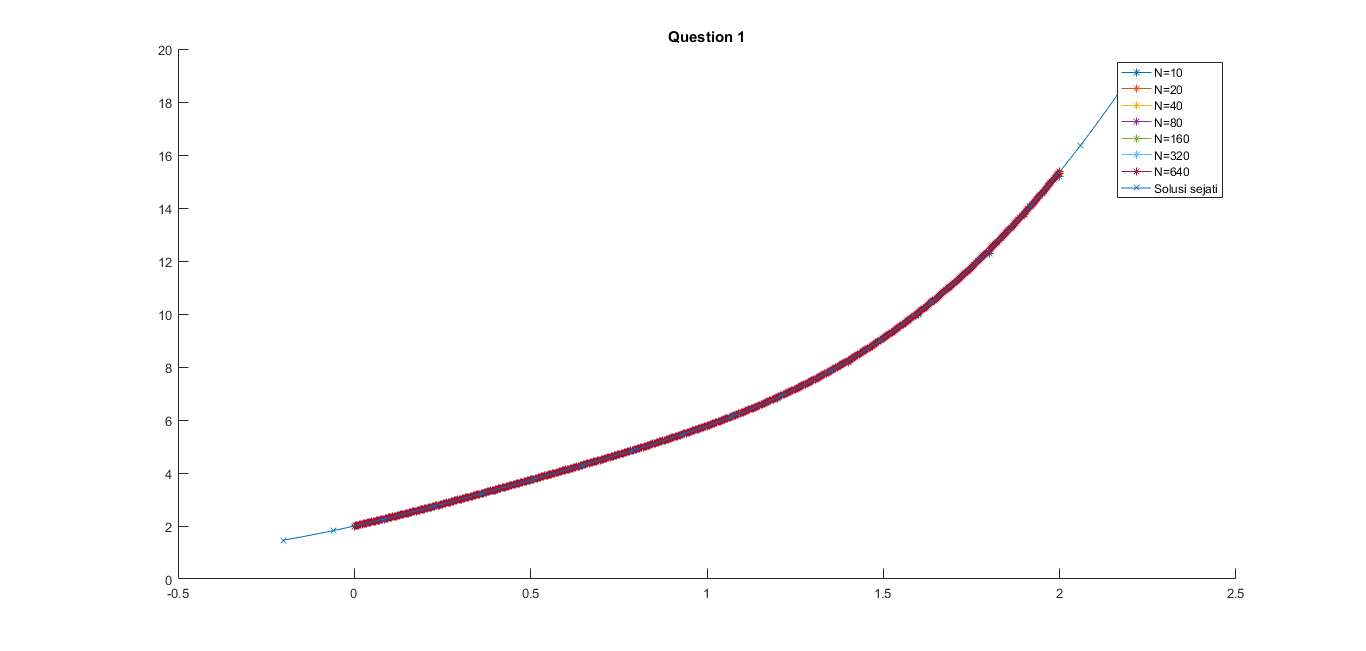
Grafik solusi numerik:

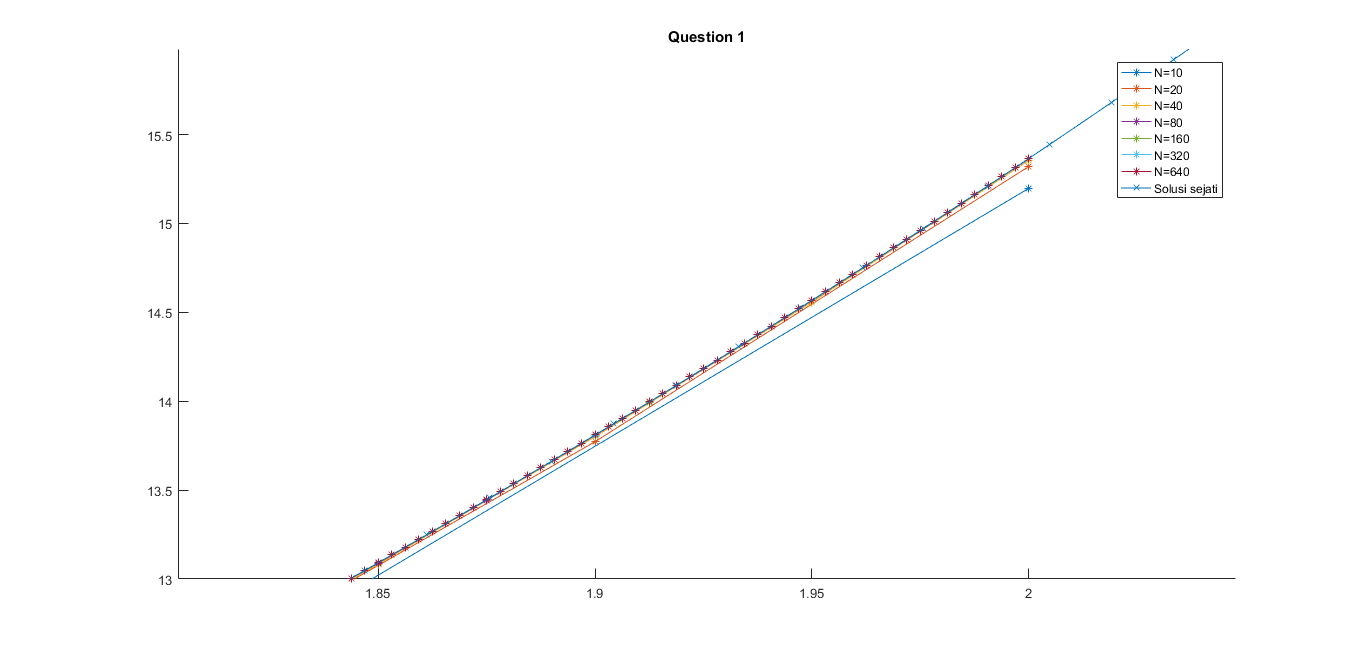
* Metode Euler



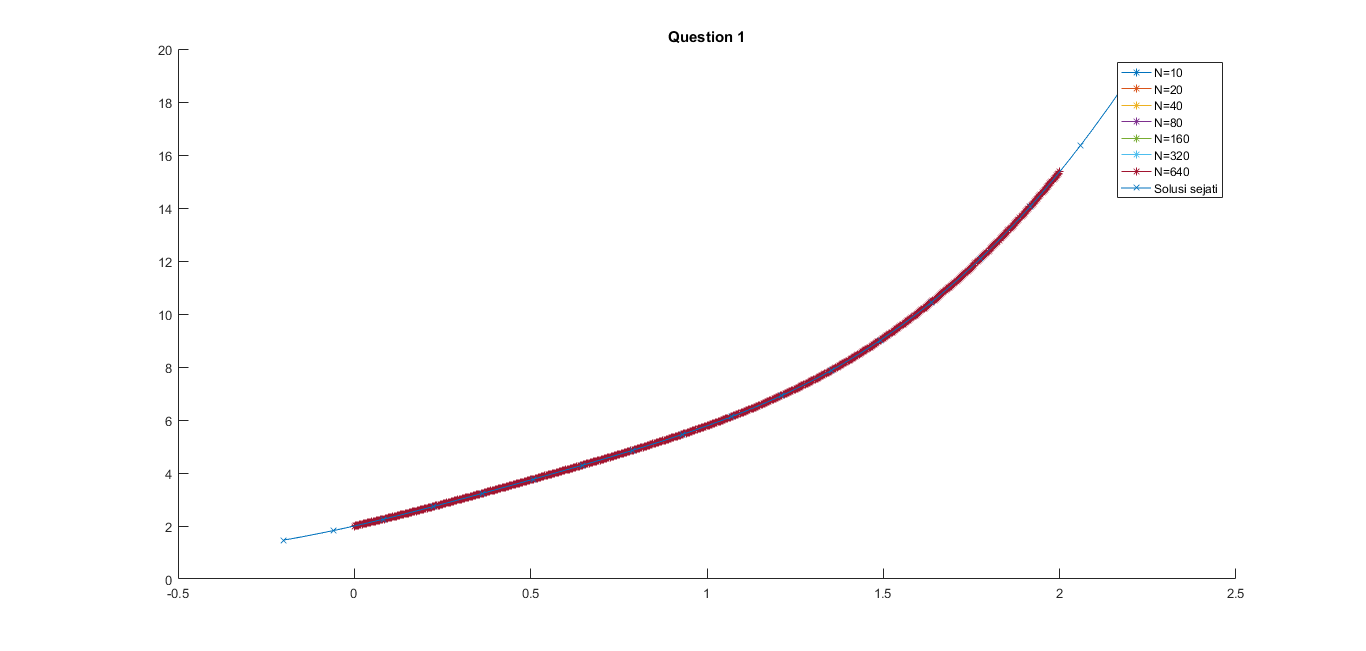


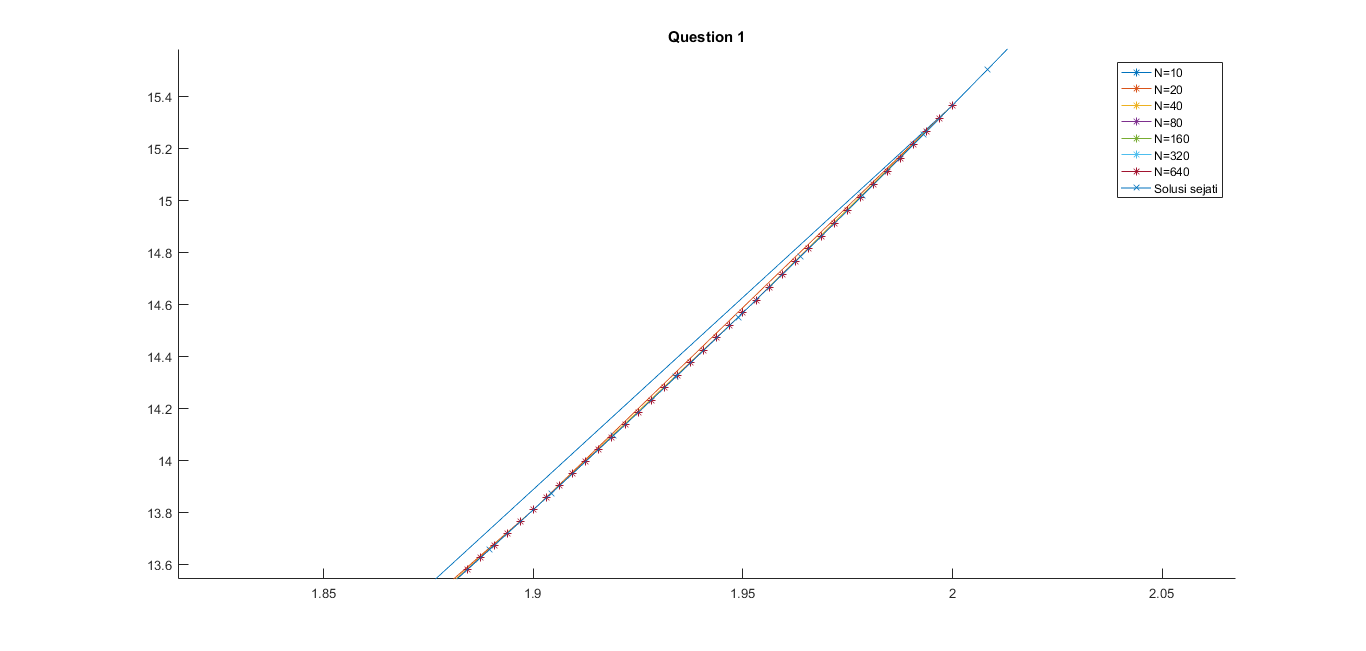
* Metode Runge-Kutta Order 2



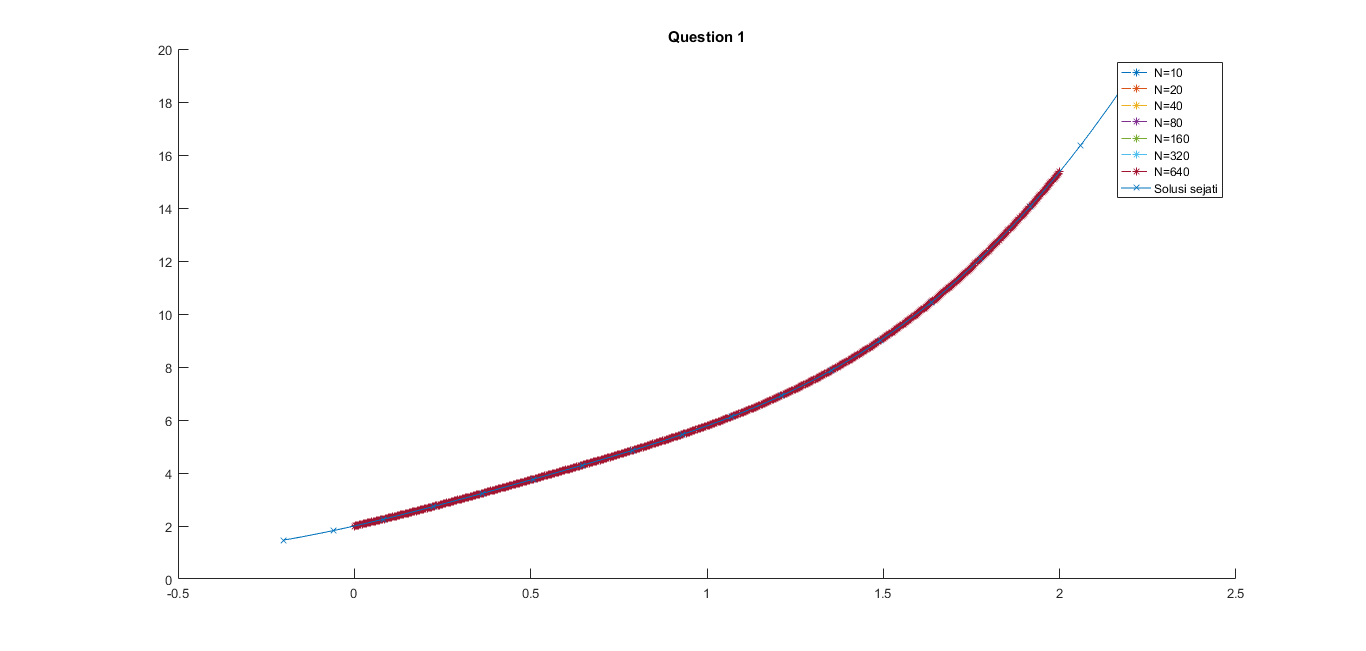


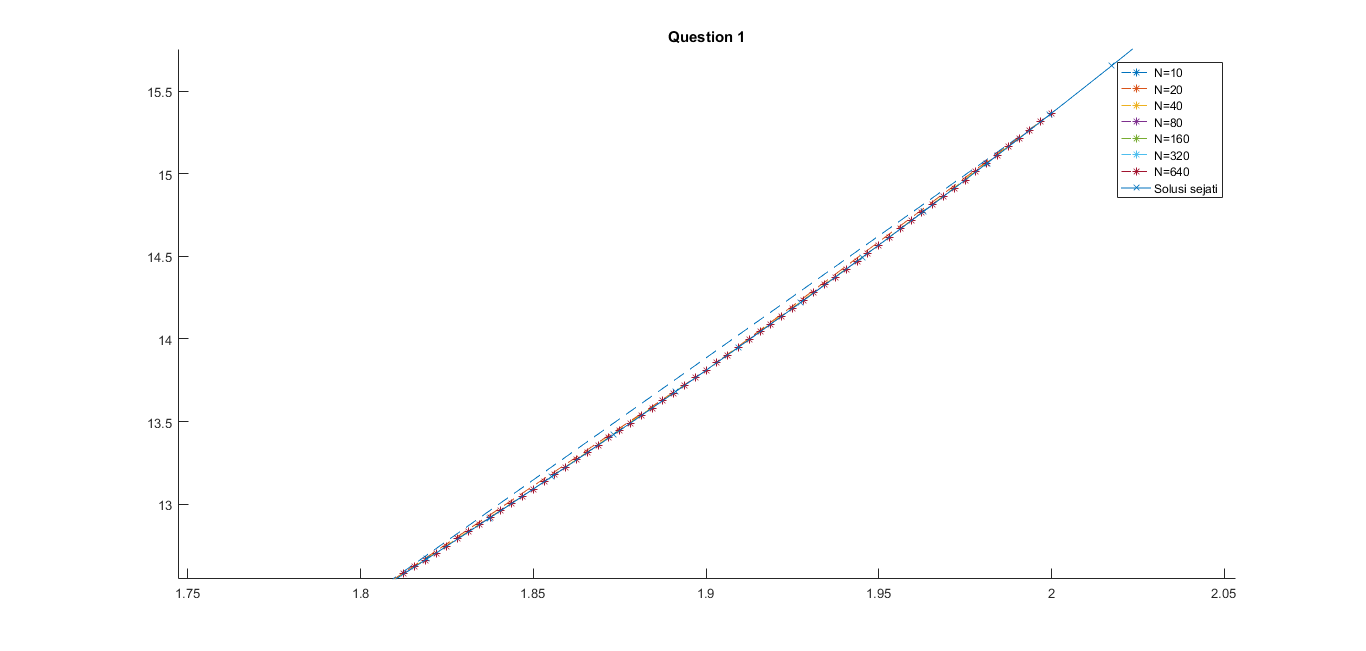
* Metode Runge-Kutta Orde 4





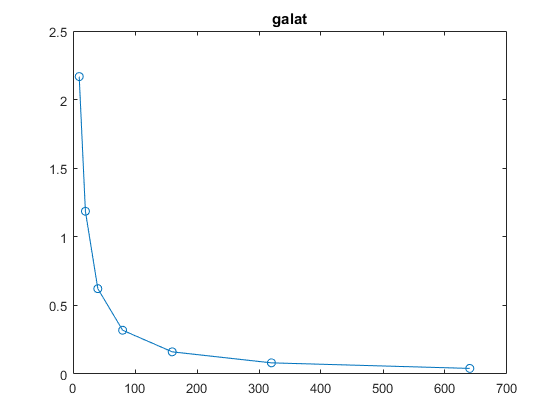
* Metode Adams-Bashforth-Moulton



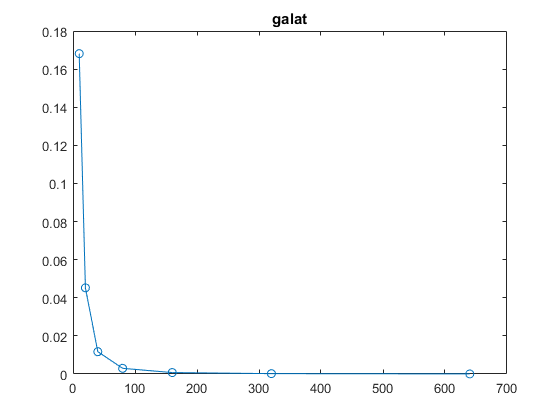


Grafik galat:

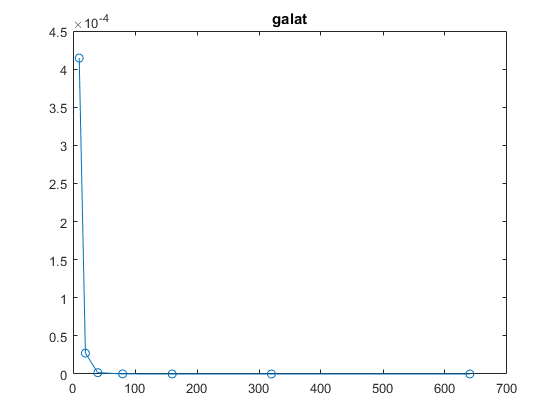
* Metode Euler



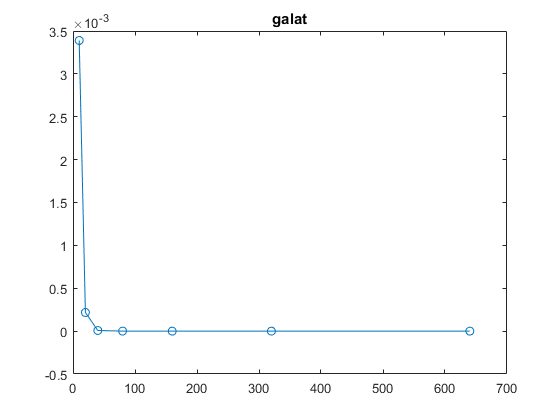
* Metode Runge-Kutta Orde 2



* Metode Runge-Kutta Orde 4



* Metode Adams-Bashforth-Moulton



Berdasarkan hasil perhitungan hampiran menggunakan berbagai jenis metode di atas, dapat dilihat pada tabel 1.1 bahwa metode banyak-langkah Runge-Kutta orde 4 memiliki galat yang paling kecil untuk persamaan diferensial nomor 1 di antara keempat metode yang digunakan, lalu yang terkecil berikutnya adalah metode Adams-Bashforth-Moulton yang memiliki orde galat yang hampir sama dengan metode Runge-Kutta orde 4. Akan tetapi metode Adams-Bashforth-Moulton membutuhkan komputasi yang lebih besar daripada metode Runge-Kutta orde 4 karena masih perlu menggunakan metode Runge-Kutta orde 3 untuk menentukan 3 titik awalnya.

Selain itu, semakin besar jumlah iterasi (N) yang digunakan akan semakin kecil galat yang dihasilkan. Metode yang paling cepat mendekati nilai sejatinya adalah Metode Runge-Kutta orde 2 hanya dengan 10 iterasi sudah menghasilkan nilai galat yang cukup layak (<0.0001).

1. Dengan menggunakan:
2. Metode Euler
3. Metode Euler Implisit
4. Metode Runge-Kutta orde 2
5. Metode Runge-Kutta orde 4
6. Metode banyak-langkah Adam-Bashforth-Moulton
7. Metode Runge-Kutta implisit

untuk menghampiri solusi dari persamaan diferensial

dengan kondisi awal

Persamaan diferensial di atas diselesaikan dengan mencapai 1.0 atau selama hasilnya masih layak, dengan jumlah titik .

Solusi sejati dari persamaan diferensial tersebut adalah

Table 2.1 galat dari hasil pendekatan menggunakan metode Euler, Euler Implisit, dan Runge-Kutta orde 2

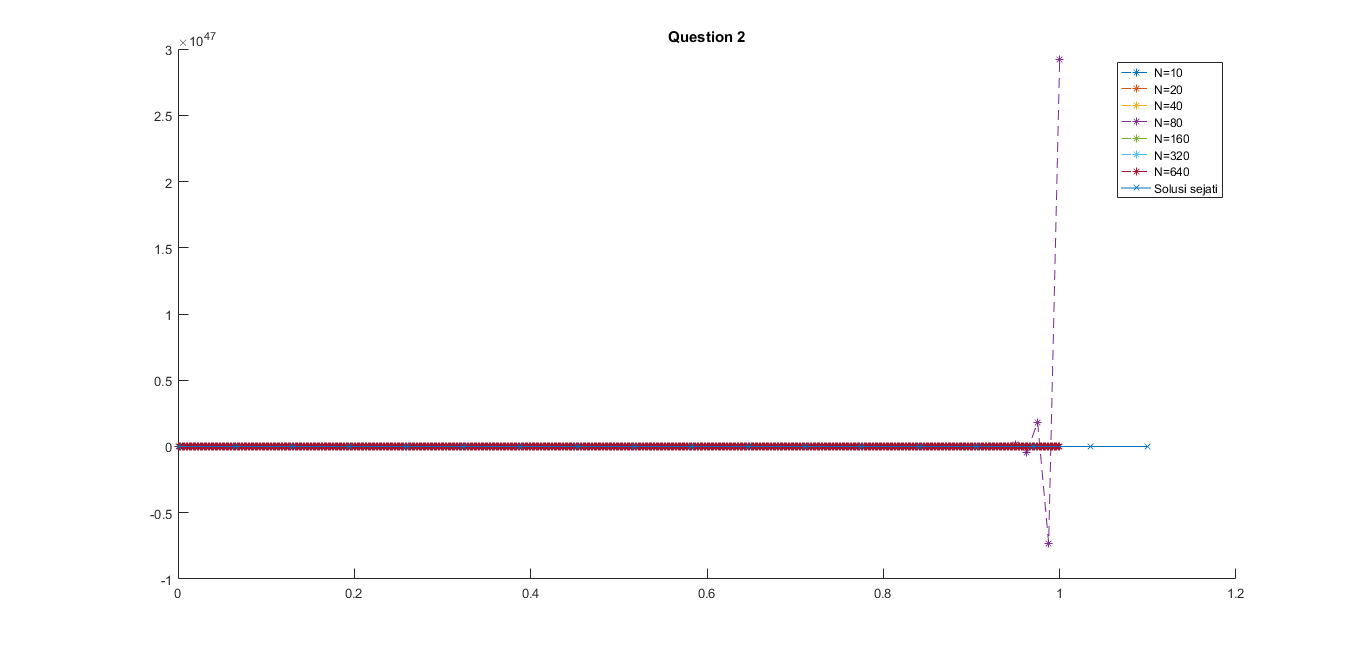
|  |  |  |  |
| --- | --- | --- | --- |
| **N** | **Euler** | **Euler**  **Implisit** | **Runge-Kutta**  **Orde 2** |
| 10  20  40  80  160  320  640 | -1.62808121703832e+15  -7.51799469150920e+24  -2.95617658828692e+37  -2.92300327466181e+47  -2.98972638754465e+27  3.83033919342801e-175  3.83033919342801e-175 | -1.49001770348688e-17  -7.18859639879228e-28  -4.41898563043600e-43  -1.11925737566628e-63  -1.77886462876446e-88  -4.00519993115218e-114  -2.26399880942307e-136 | -1.30279644066346e+28  -2.84840338395411e+44  -6.49208600308643e+63  -4.51381809085076e+73  -1.09035632210524e+33  -2.49282490174640e-89  -1.63410477153124e-157 |

|  |  |  |  |
| --- | --- | --- | --- |
| **N** | **Runge-Kutta**  **Orde 4** | **Adam-Bashforth-Moulton** | **Runge-Kutta**  **Implisit** |
| 10  20  40  80  160  320  640 | -1.43890948708335e+49  -1.35173106955599e+74  -7.19034212466235e+97  -1.81887349651419e+90  -1.58463641634757e-31  -2.45963284082546e-165  -5.21116050021688e-175 | 1.41906576551356e+33  1.20712767733995e+51  6.10303317650836e+74  6.57446145688126e+93  2.09607697627784e+25  1.32006683854052e-05  3.81321473477289e-175 | -97.6074840801362  -11175.0200641218  -2211466.46418801  -8717.37550607887  -1.33791507365501e-29  3.83033919342801e-175  3.83033919342801e-175 |

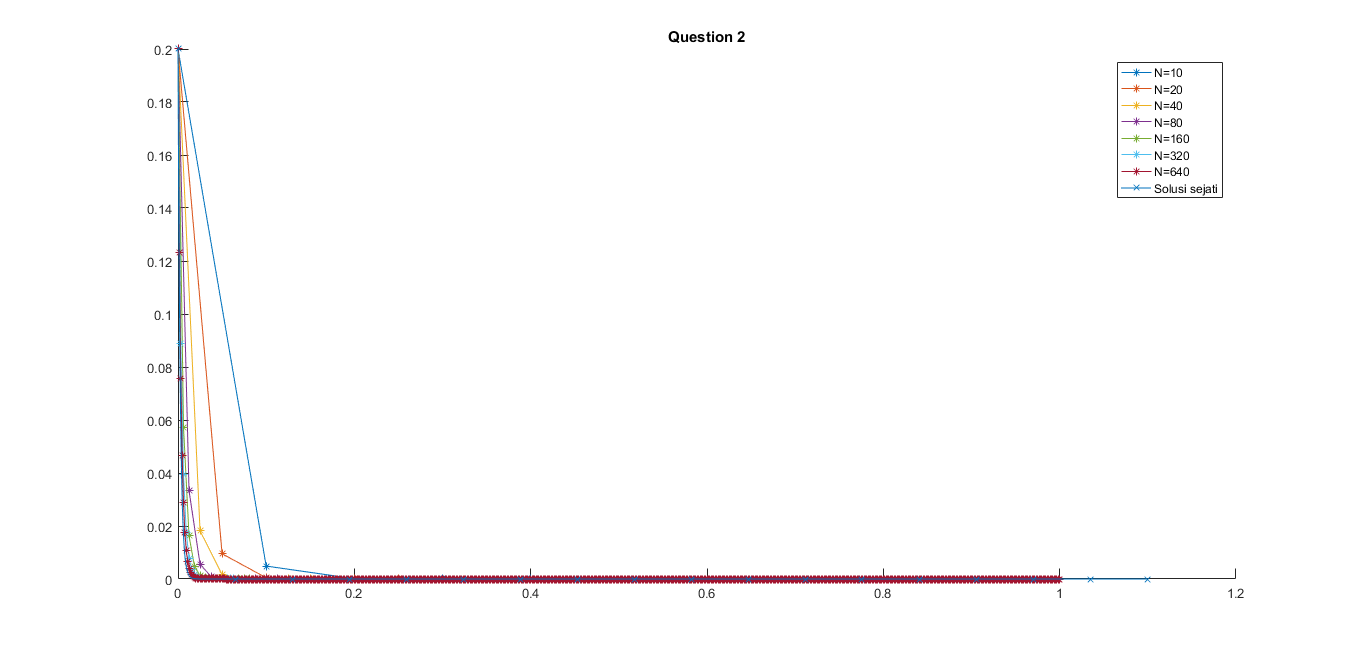
Tabel 2. 2 galat dari hasil pendekatan menggunakan metode Runge-Kutta orde 4, Adam-Bashforth-Moulton, dan Runge-Kutta implisit

Grafik solusi numerik:

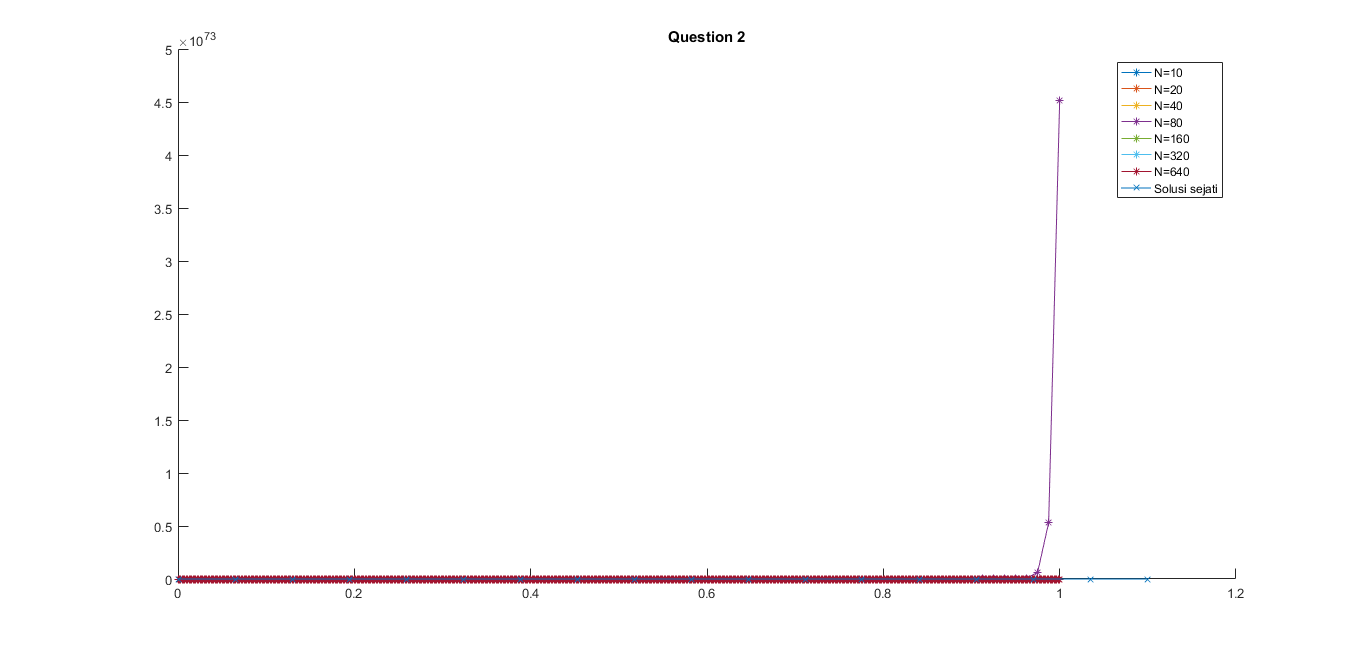
* Metode Euler



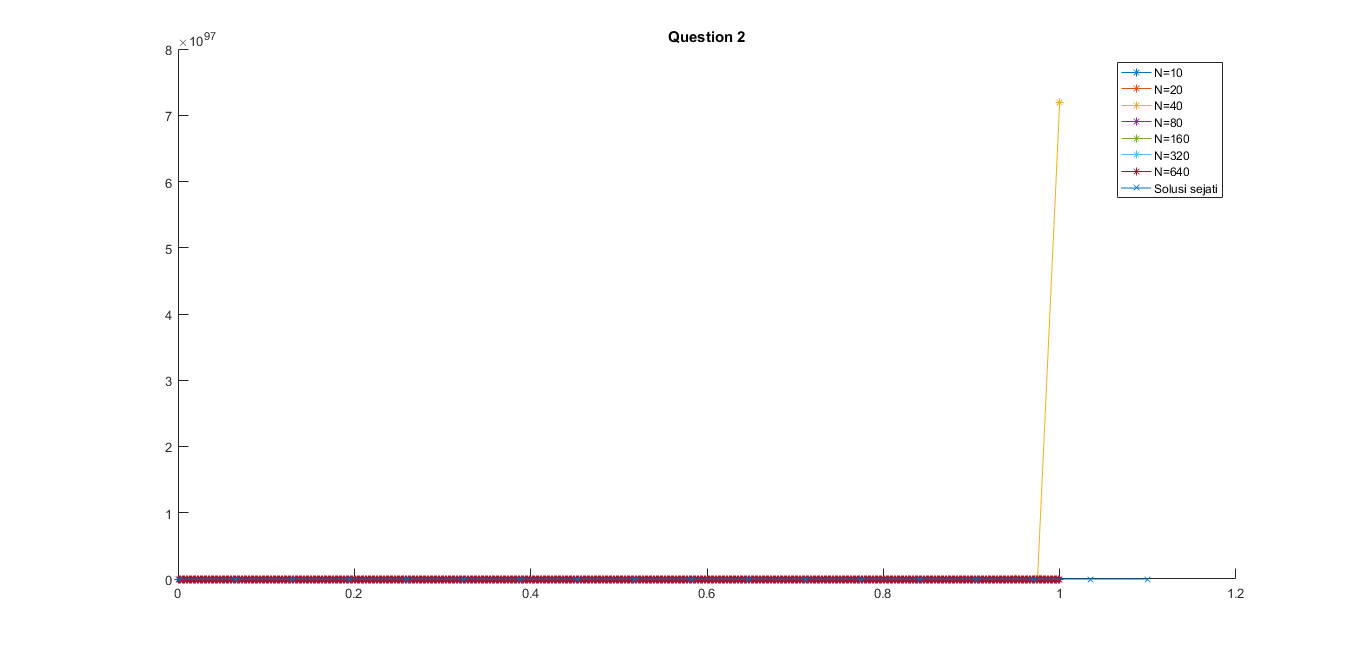
* Metode Euler implisit



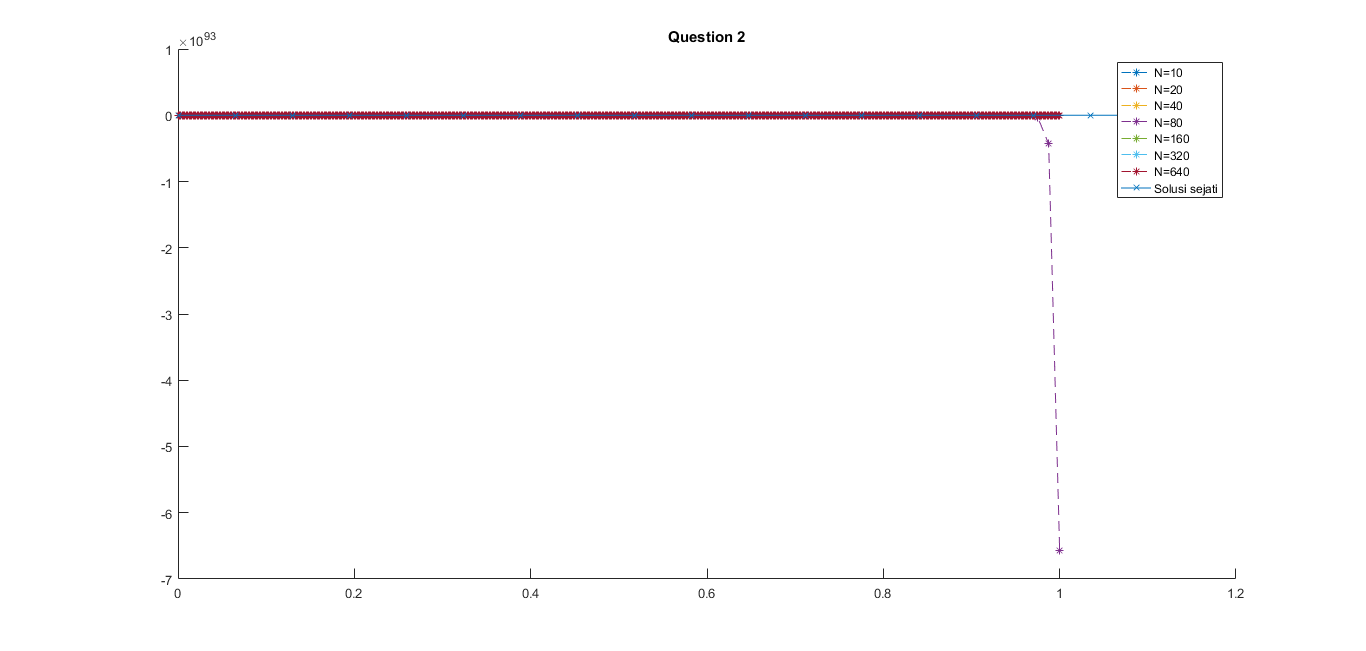
* Metode Runge-Kutta orde 2



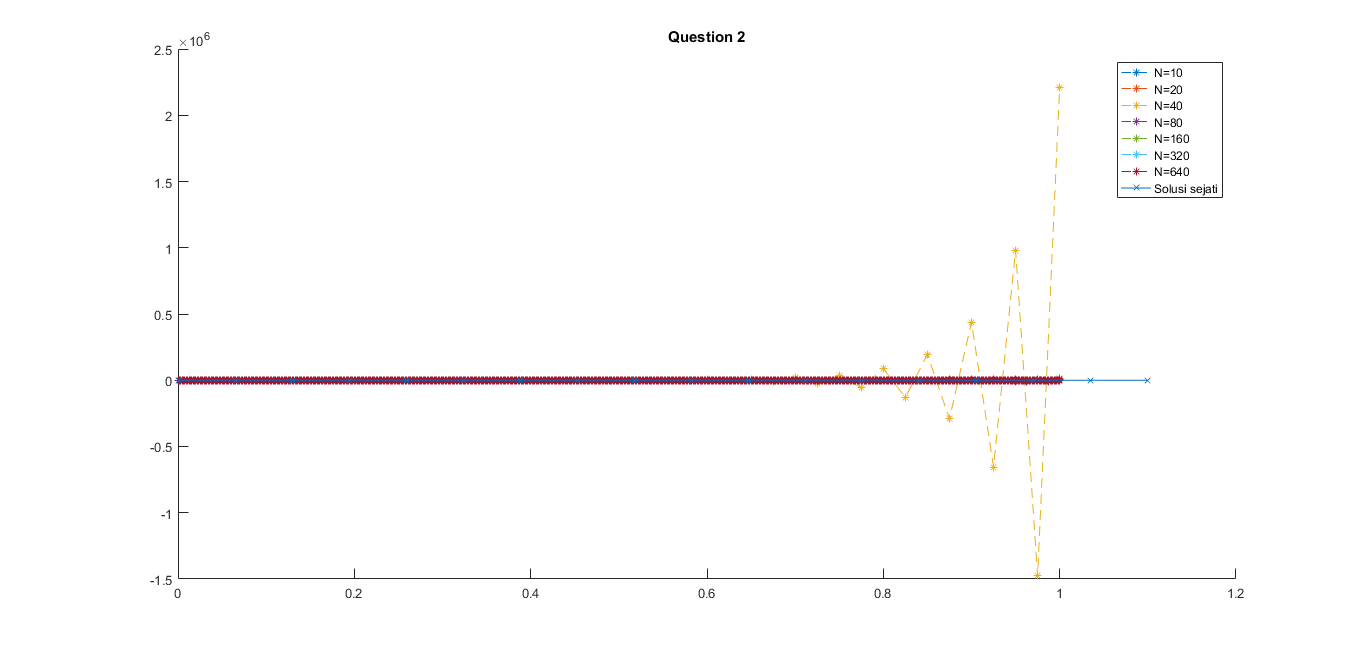
* Metode Runge-Kutta orde 4



* Metode Adams-Bashforth-Moulton

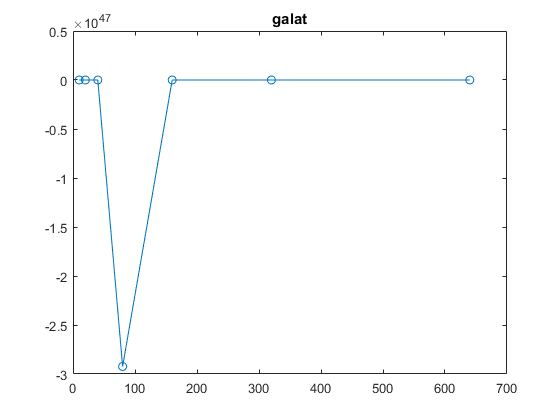


* Metode Runge-Kutta implisit

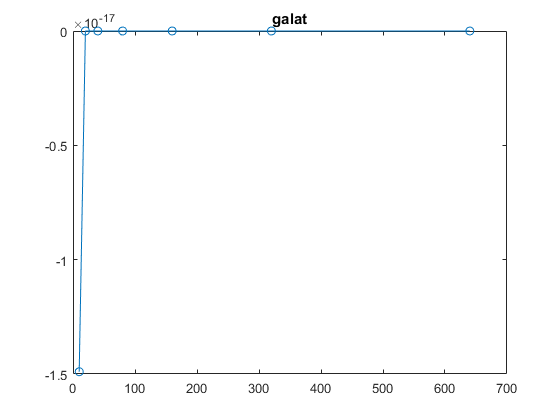


Grafik galat:

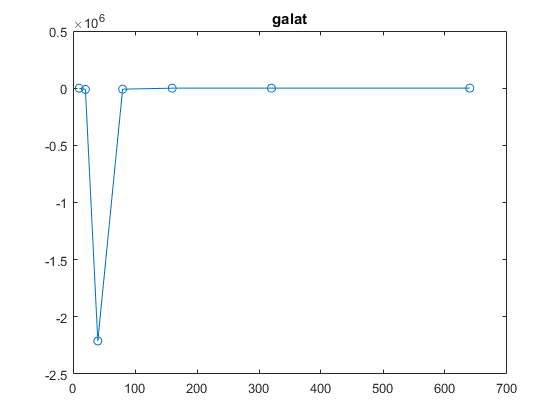
* Metode Euler



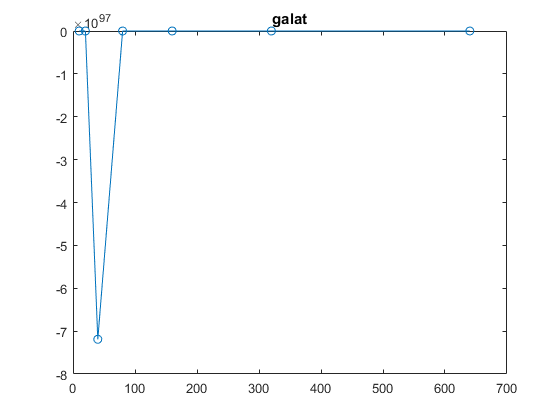
* Metode Euler implisit



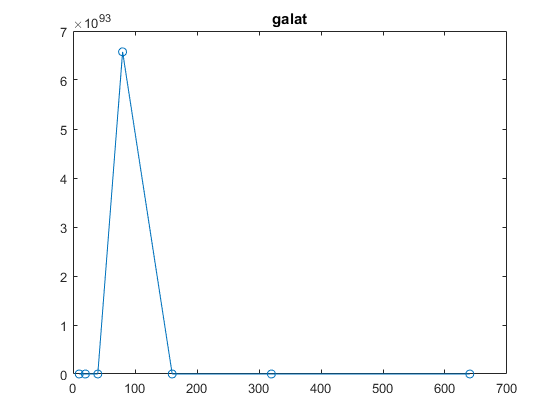
* Metode Runge-Kutta orde 2



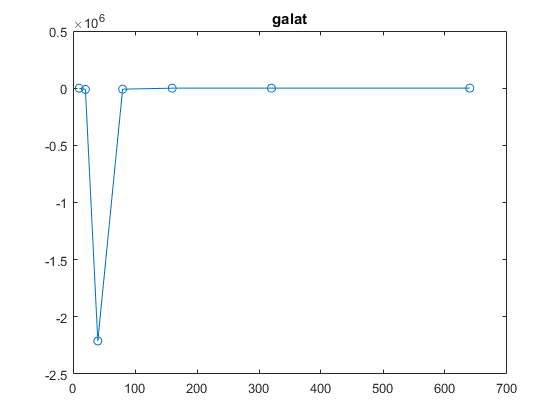
* Metode Runge-Kutta orde 4



* Metode Adams-Bashforth-Moulton



* Metode Runge-Kutta implisit



Jika dilihat dari tabel 2.1, dapat disimpulkan bahwa metode yang memiliki galat paling kecil dengan jumlah iterasi 640 untuk persamaan diferensial nomor 2 adalah metode banyak-langkah Adams-Bashforth-Moulton, tetapi memiliki orde galat yang mirip dengan metode Euler, Runge-Kutta orde 4, dan Runge-Kutta implisit. Pada semua metode yang digunakan kecuali metode Euler implisit, terjadi osilasi pada jumlah iterasi antara 40 dan 80, dengan metode Runge-Kutta implisit yang mengalami osilasi paling besar sehingga memiliki nilai galat yang paling besar untuk jumlah iterasi tersebut.

1. Dengan menggunakan metode Euler, Runge-Kutta orde 2, dan Runge-Kutta orde 4 untuk menghampiri solusi dari persamaan diferensial

dengan kondisi awal

Persamaan diferensial tersebut bisa dinyatakan sebagai

persamaan diferensial tersebut diselesaikan dengan waktu mencapai 2.0, dengan jumlah titik .

Solusi sejati yang didapatkan dari persamaan diferensial tersebut adalah

Table 3.1 galat dari hasil pendekatan fungsi x menggunakan metode Euler, Runge-Kutta orde 2, dan Runge-Kutta orde 4

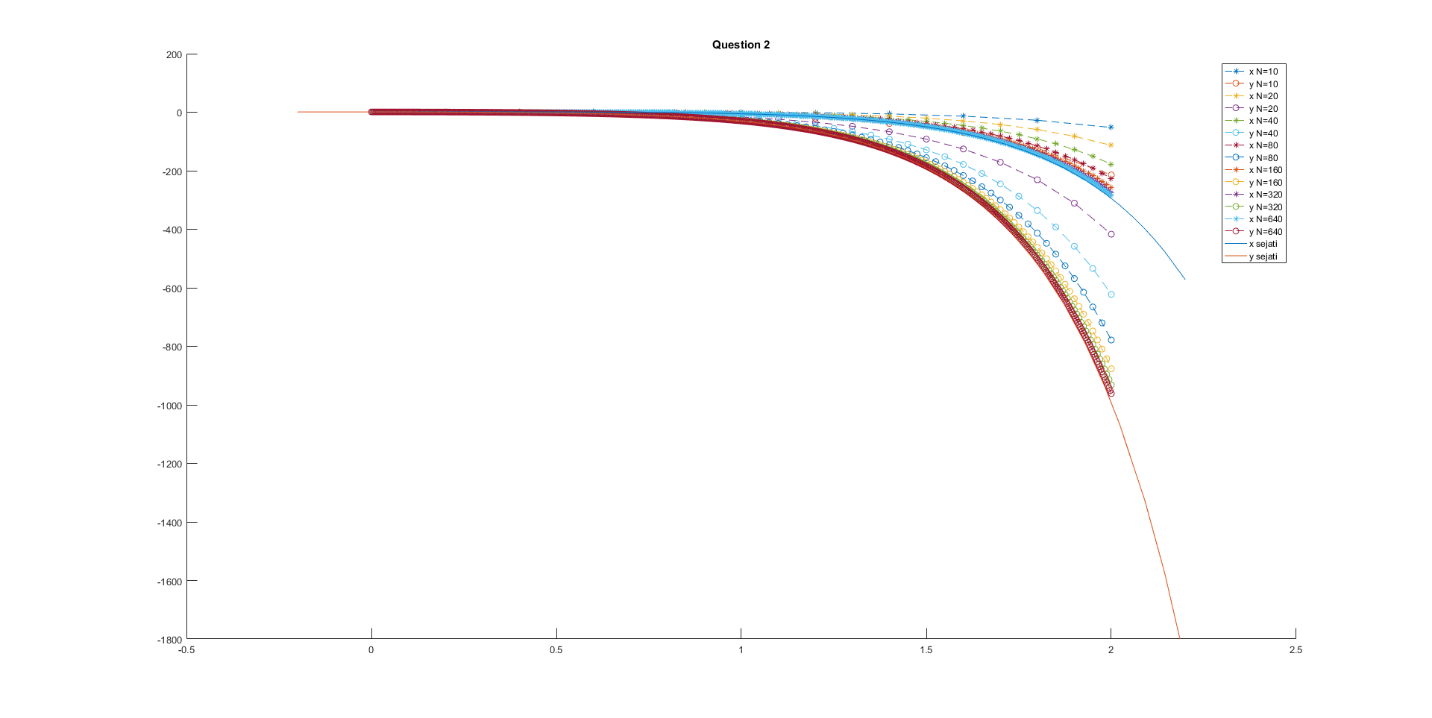
|  |  |  |  |
| --- | --- | --- | --- |
| **N** | **Euler** | **Runge-Kutta**  **Orde 2** | **Runge-Kutta**  **Orde 4** |
| 10  20  40  80  160  320  640 | -242.132261644047  -180.858055526588  -116.887458328095  -67.7608155607873  -36.6977521147114  -19.1278288180220  -9.76902006470715 | -254.249537399542  -203.537928617781  -141.542274443988  -86.7374445975576  -48.6308764873269  -25.8448905862213  -13.3361922465001 | -252.999760179929  -202.735346333538  -141.168906522777  -86.6022691524673  -48.5894104992180  -25.8333430876077  -13.3331407782976 |

Table 3.2 galat dari hasil pendekatan fungsi y menggunakan metode Euler, Runge-Kutta orde 2, dan Runge-Kutta orde 4

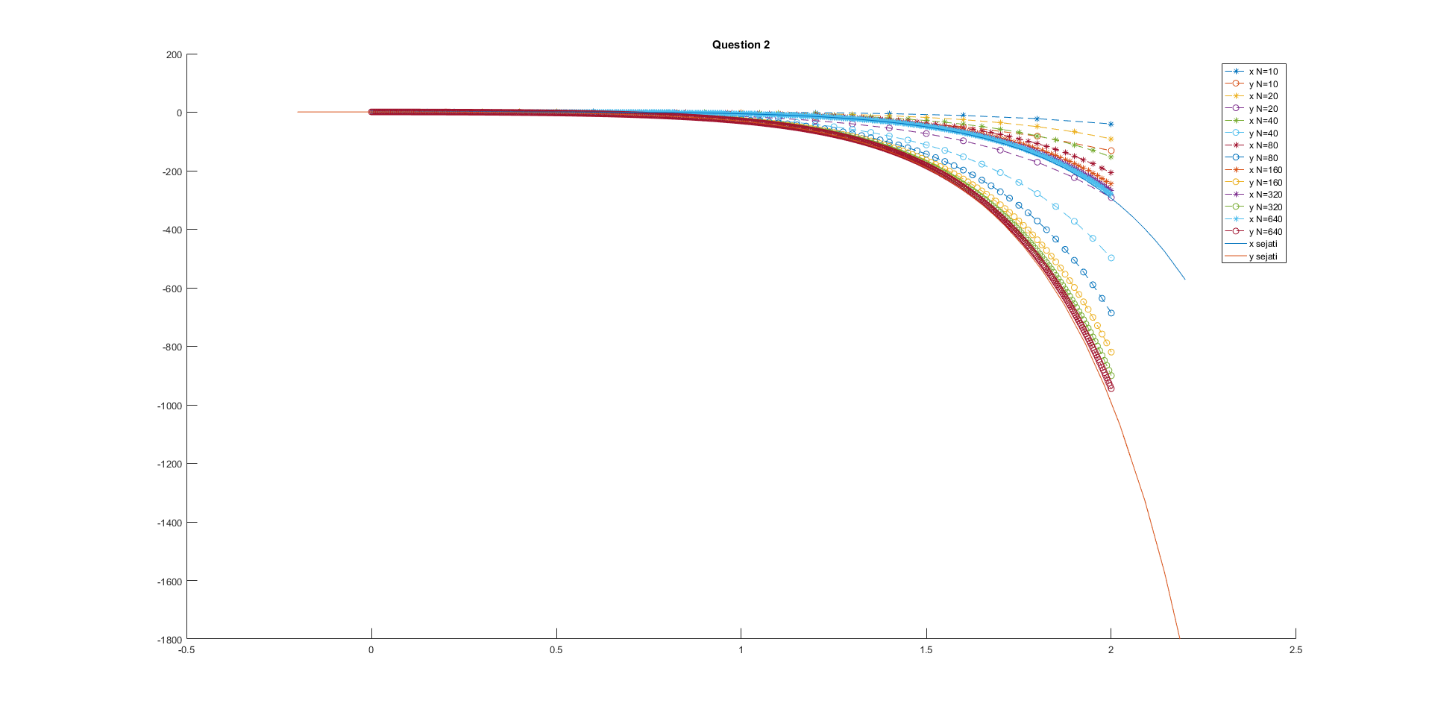
|  |  |  |  |
| --- | --- | --- | --- |
| **N** | **Euler** | **Runge-Kutta**  **Orde 2** | **Runge-Kutta**  **Orde 4** |
| 10  20  40  80  160  320  640 | -777.742154003228  -575.095266797103  -369.340163914221  -213.355864614965  -115.333820216798  -60.0575688411446  -30.6579278126663 | -860.489212140834  -700.394904848451  -493.952313182216  -305.553264941951  -172.254232998984  -91.8158597195903  -47.4507698303976 | -856.916635966859  -698.035887215980  -492.813348587361  -305.130495905735  -172.122708110418  -91.7789594238840  -47.4409815626869 |

Grafik solusi numerk:

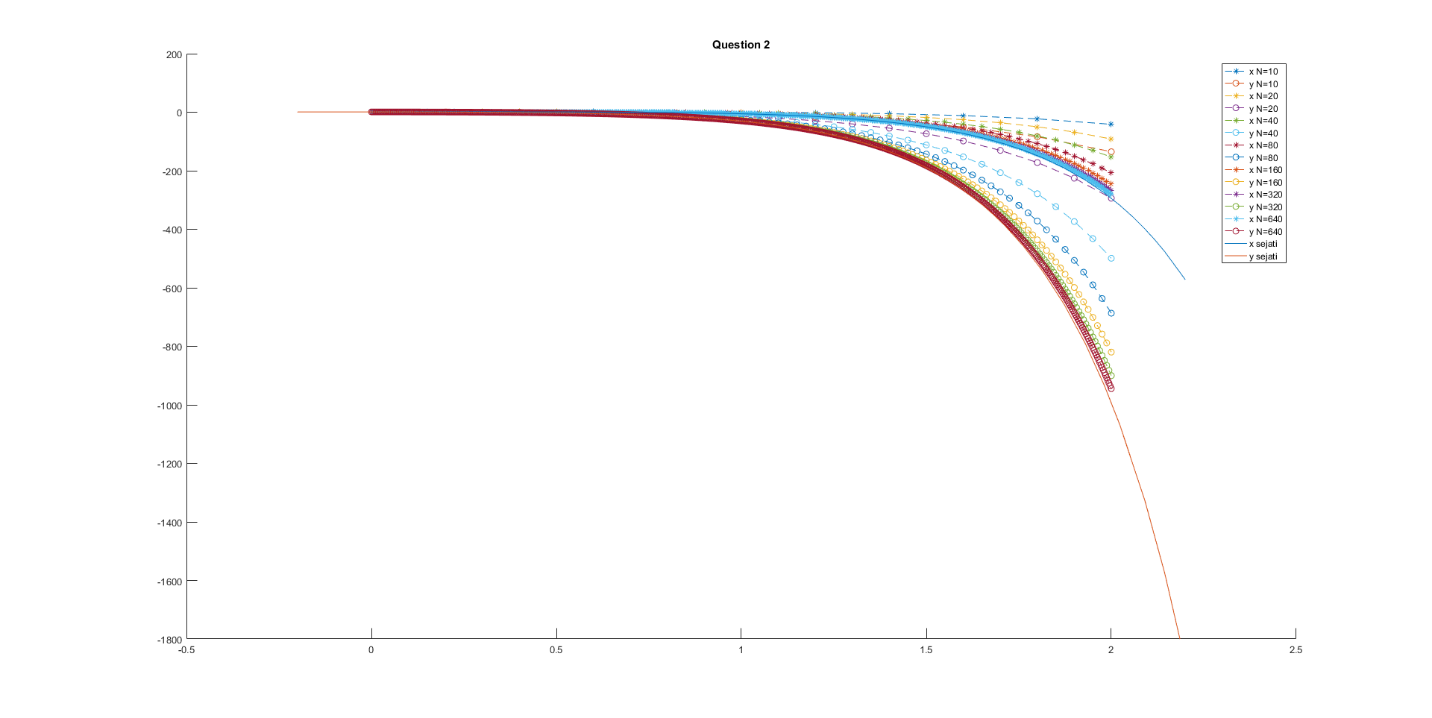
* Metode Euler



* Metode Runge-Kutta orde 2

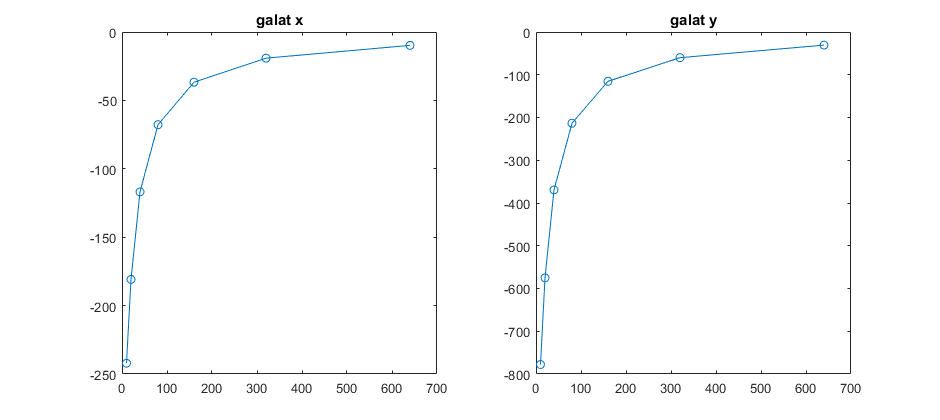


* Metode Runge-Kutta orde 4

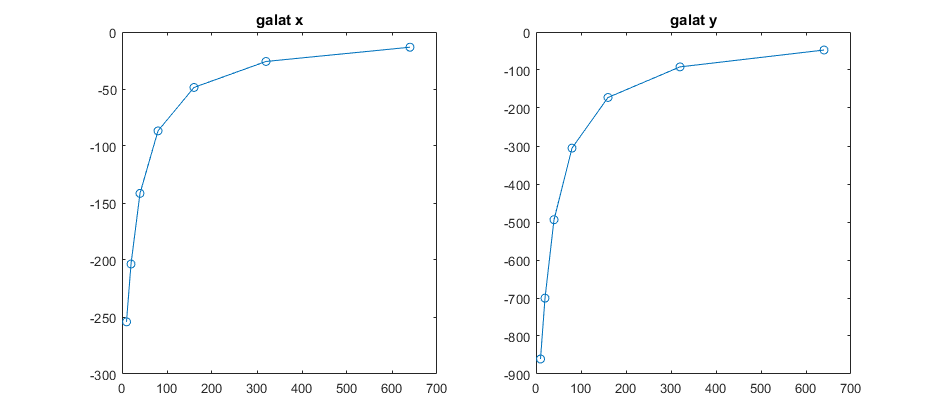


Grafik galat:

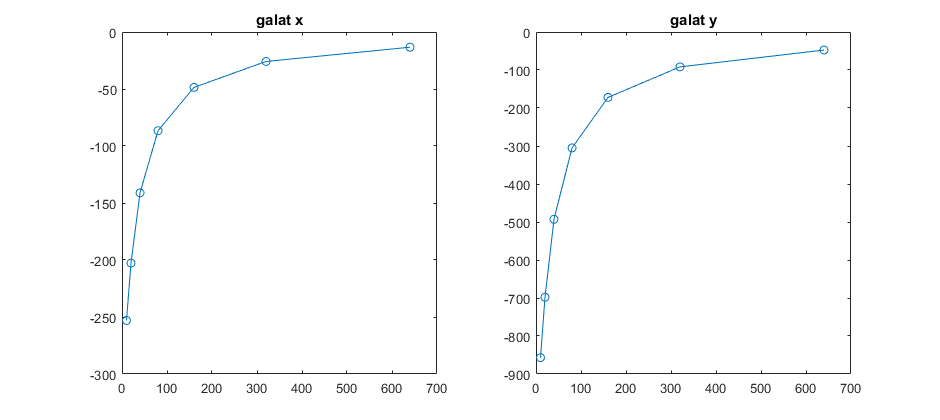
* Metode Euler



* Metode Runge-Kutta orde 2



* Metode Runge-Kutta orde 4



Untuk permasalahan dengan persamaan diferensial di atas dengan dua persamaan, metode yang memiliki nilai galat paling rendah adalah metode Euler, berbeda dengan permasalahan sebelumnya yang hanya menggunakan satu persamaan saja. Sepertinya mungkin ada kesalahan pada implementasi karena mungkin ada salah dari pengertian saya dalam memahami penjelasan pada buku Metode Numerik. Namun jika jumlah iterasinya ditambahkan menjadi 81.920 maka hasil hampirannya baru mendekati dengan nilai sejatinya dengan nilai galat yang cukup kecil (< 0.01).

Lampiran

function [ y\_euler ] = euler( x0, y0, b, h, f )

% menghitung nilai y(b) pada PDB

% y'=f(x,y); y(x0)=y0

% dengan metode Euler

n = (b-x0)/h;

x = zeros(1, n);

y = zeros(1, n);

y(1) = y0;

x(1) = x0;

for r = 1:n

y(r+1) = y(r) + h \* f(x(r),y(r));

x(r+1) = x(r) + h;

end

y\_euler = y(n+1);

plot(x, y, '--\*');

end

function [ y\_RK2 ] = RK2( x0, y0, b, h, f )

% menghitung nilai y(b) pada PDB

% y'=f(x,y); y(x0)=y0

% dengan metode Runge-Kutta orde 2

n = (b-x0)/h;

x = zeros(1, n);

y = zeros(1, n);

x(1) = x0;

y(1) = y0;

for r = 1:n

k1 = h \* f(x(r),y(r));

k2 = h \* f(x(r) + h, y(r) + k1);

y(r+1) = y(r) + (k1+k2)/2;

x(r+1) = x(r) + h;

end

y\_RK2 = y(n+1);

plot(x, y, '-\*');

end

function [ y\_RK3 ] = RK3( x0, y0, b, h, f )

% menghitung nilai y(b) pada PDB

% y'=f(x,y); y(x0)=y0

% dengan metode Runge-Kutta orde 3

n = (b-x0)/h;

x = zeros(1, uint8(n));

y = zeros(1, uint8(n));

x(1) = x0;

y(1) = y0;

for r = 1:n

k1 = h \* f(x(r),y(r));

k2 = h \* f(x(r) + h/2, y(r) + k1/2);

k3 = h \* f(x(r) + h, y(r) - k1 + 2\*k2);

y(r+1) = y(r) + (k1+4\*k2+k3)/6;

x(r+1) = x(r) + h;

end

y\_RK3 = y(n+1);

% plot(x, y, '-\*');

end

function [ y\_RK4 ] = RK4( x0, y0, b, h, f )

% menghitung nilai y(b) pada PDB

% y'=f(x,y); y(x0)=y0

% dengan metode Runge-Kutta orde 4

n = (b-x0)/h;

x = zeros(1, n);

y = zeros(1, n);

x(1) = x0;

y(1) = y0;

for r = 1:n

k1 = h \* f(x(r),y(r));

k2 = h \* f(x(r) + h/2, y(r) + k1/2);

k3 = h \* f(x(r) + h/2, y(r) + k2/2);

k4 = h \* f(x(r) + h, y(r) + k3);

y(r+1) = y(r) + (k1 + 2\*k2 + 2\*k3 + k4) / 6;

x(r+1) = x(r) + h;

end

y\_RK4 = y(n+1);

plot(x, y, '-\*');

end

function [ y\_ABM ] = ABM( x0, y0, b, h, f )

% menghitung nilai y(b) pada PDB

% dengan menggunakan metode Adams-Bashforth-Moulton

n = (b-x0)/h;

x = zeros(1, n);

y = zeros(1, n);

x(1) = x0;

y(1) = y0;

x(2) = x0 + h;

y(2) = RK3(x0, y0, x(2), h, f);

x(3) = x0 + 2\*h;

y(3) = RK3(x0, y0, x(3), h, f);

x(4) = x0 + 3\*h;

y(4) = RK3(x0, y0, x(4), h, f);

for r = 4:n

% predictor

y(r+1) = y(r) + h/24\*(-9\*f(x(r)-3\*h, y(r-3)) +...

37\*f(x(r)-2\*h, y(r-2)) - 59\*f(x(r)-h, y(r-1)) +...

55\*f(x(r), y(r)));

% corrector

y(r+1) = y(r) + h/24\*(f(x(r)-2\*h, y(r-2)) - 5\*f(x(r)-h, y(r-1)) +...

19\*f(x(r), y(r)) + 9\*f(x(r)+h, y(r+1)));

x(r+1) = x(r) + h;

end

y\_ABM = y(n+1);

plot(x, y, '--\*');

end

function [ y\_implicit\_euler ] = implicit\_euler( x0, y0, b, h, f )

% menghitung nilai y(b) pada PDB

% y'=f(x,y); y(x0)=y0

% dengan metode backward euler

n = (b-x0)/h;

x = zeros(1, n);

y = zeros(1, n);

x(1) = x0;

y(1) = y0;

for r = 1:n

x(r+1) = x(r) + h;

% ini mah backward

% y\_temp = y(i) + h \* f(x(i), y(i)); % harusnya pake newton rhapson

% y(i+1) = y(i) + h \* f(x(i+1), y\_temp);

% ini baru implicit

y(r+1) = y(r) / (1 + (400\*h));

end

y\_implicit\_euler = y(n+1);

plot(x, y, '-\*');

end

function [ y\_IRK ] = implicit\_runge\_kutta( x0, y0, b, h, f )

% menghitung nilai y(b) pada PDB

% dengan menggunakan metode runge-kutta implicit

n = (b-x0)/h;

x = zeros(1, n);

y = zeros(1, n);

y(1) = y0;

x(1) = x0;

for r = 1:n

k1 = f(x(r), y(r));

y(r+1) = (y(r) + h\*k1) / (1 + 200\*h);

x(r+1) = x(r) + h;

end

y\_IRK = y(n+1);

plot(x, y, '--\*');

end

function [ x\_euler, y\_euler ] = two\_euler( t0, x0, y0, b, h, f1, f2 )

% menghitung nilai x(b) dan y(b) pada PDB

% x'=f(t,x,y); x(t0)=x0

% y'=f(t,x,y); y(t0)=y0

% dengan metode Euler

n = (b-t0)/h;

t = zeros(1, n);

x = zeros(1, n);

y = zeros(1, n);

t(1) = t0;

y(1) = y0;

x(1) = x0;

for r = 1:n

x(r+1) = x(r) + h \* f1(t(r), x(r), y(r));

y(r+1) = y(r) + h \* f2(t(r), x(r), y(r));

t(r+1) = t(r) + h;

end

x\_euler = x(n+1);

y\_euler = y(n+1);

plot(t, x, '--\*');

plot(t, y, '--o');

end

function [ x\_RK2, y\_RK2 ] = two\_RK2( t0, x0, y0, b, h, f1, f2 )

% menghitung nilai x(b) dan y(b) pada PDB

% x'=f(t,x,y); x(t0)=x0

% y'=f(t,x,y); y(t0)=y0

% dengan metode Runge-Kutta orde 2

n = (b-t0)/h;

t = zeros(1, n);

x = zeros(1, n);

y = zeros(1, n);

t(1) = t0;

y(1) = y0;

x(1) = x0;

for r = 1:n

k1 = h \* f1(t(r),x(r),y(r));

k2 = h \* f1(t(r) + h, x(r) + k1, y(r) + k1);

x(r+1) = x(r) + (k1+k2)/2;

k1 = h \* f2(t(r),x(r),y(r));

k2 = h \* f2(t(r) + h, x(r) + k1, y(r)+ k1);

y(r+1) = y(r) + (k1+k2)/2;

t(r+1) = t(r) + h;

end

x\_RK2 = x(n+1);

y\_RK2 = y(n+1);

plot(t, x, '--\*');

plot(t, y, '--o');

end

function [ x\_RK4, y\_RK4 ] = two\_RK4( t0, x0, y0, b, h, f1, f2 )

% menghitung nilai x(b) dan y(b) pada PDB

% x'=f(t,x,y); x(t0)=x0

% y'=f(t,x,y); y(t0)=y0

% dengan metode Runge-Kutta orde 2

n = (b-t0)/h;

t = zeros(1, n);

x = zeros(1, n);

y = zeros(1, n);

t(1) = t0;

y(1) = y0;

x(1) = x0;

for r = 1:n

k1 = h \* f1(t(r),x(r),y(r));

k2 = h \* f1(t(r) + h/2, x(r) + k1/2, y(r) + k1/2);

k3 = h \* f1(t(r) + h/2, x(r) + k2/2, y(r) + k2/2);

k4 = h \* f1(t(r) + h, x(r) + k3, y(r) + k3);

x(r+1) = x(r) + (k1 + 2\*k2 + 2\*k3 + k4) / 6;

k1 = h \* f2(t(r),x(r),y(r));

k2 = h \* f2(t(r) + h/2, x(r) + k1/2, y(r) + k1/2);

k3 = h \* f2(t(r) + h/2, x(r) + k2/2, y(r) + k2/2);

k4 = h \* f2(t(r) + h, x(r) + k3, y(r) + k3);

y(r+1) = y(r) + (k1 + 2\*k2 + 2\*k3 + k4) / 6;

t(r+1) = t(r) + h;

end

x\_RK4 = x(n+1);

y\_RK4 = y(n+1);

plot(t, x, '--\*');

plot(t, y, '--o');

end

Referensi

1. Munir, Rinaldi. 2003. Metode Numerik. Bandung: Penerbit Informatika.