TTK4130 MODELING AND SIMULATION | ASSIGNMENT 7

ODIN A. SEVERINSEN | MTTK

1) We have $e_k = x_{k+1} - x(t_{k+1}|k)$. Taylor expanding $x(t_{k+1}|k)$:

$$x(t_{k+1}|k) = x_k + \Delta t\, f(x_k,u_k) + \frac{\Delta t^2}{2}\underbrace{\frac{\partial f}{\partial x}\cdot f(x_k,u_k)}_{=\dot f} + \mathcal{O}(\Delta t^3) \quad (1)$$

($\ddot{x}(t_{k+1})$)

ERK2 for x_{k+1} gives

$$x_{k+1} = x_k + \Delta t \sum_{i=1}^{2} b_i k_i$$

$$= x_k + \Delta t\, b_1 \cdot k_1 + \Delta t\, b_2 k_2$$

$$= x_k + \Delta t\, b_1 \cdot f(x_k,u_k) + \Delta t\, b_2\, f(x_k + a\Delta t \cdot f(x_k,u_k), u_k) \quad (2)$$

$u(t_k + c\Delta t) = u_k$, $0<c<1$

Taylor expanding the last term gives:

$$\Delta t\, b_2\, f(x_k + a\Delta t \cdot f(x_k,u_k), u_k) = \Delta t \cdot b_2\left(f(x_k,u_k) + a\Delta t \cdot f(x_k,u_k)\frac{\partial f}{\partial x} + \mathcal{O}(\Delta t^2)\right)$$

$$= \Delta t \cdot b_2 \cdot f(x_k,u_k) + a b_2 \Delta t^2 f(x_k,u_k)\frac{\partial f}{\partial x} + \mathcal{O}(\Delta t^3) \quad (3)$$

Inserting (3) into (2) and taking the difference with (1) gives

$$e_k = x_k + \Delta t f(x_k, u_k) + \frac{\Delta t^2}{2} \cdot \frac{\partial f}{\partial x} f(x_k, u_k) + \mathcal{O}(\Delta t^3)$$
$$- \left[x_k + \Delta t b_1 f(x_k, u_k) + \Delta t b_2 f(x_k, u_k) + a b_2 \Delta t^2 f(x_k, u_k) \cdot \frac{\partial f}{\partial x} + \mathcal{O}(\Delta t^3) \right]$$
$$= \Delta t f(x_k, u_k) \cdot (1 - b_1 - b_2) + \Delta t^2 \frac{\partial f}{\partial x} \cdot f(x_k, u_k) \left(\frac{1}{2} - a b_2 \right) + \mathcal{O}(\Delta t^3)$$

For the error to be $\mathcal{O}(\Delta t^3)$, the two first terms must be 0.

$\Rightarrow$ $1 - b_1 - b_2 = 0$ and $\frac{1}{2} - a b_2 = 0$

$\Rightarrow$ $b_1 + b_2 = 1$ and $a b_2 = \frac{1}{2}$

As long as $0 < c < 1$, it can be whatever as u_k is constant.

b) The global or total error is the sum of errors over all steps, so

$$e = \| x_N - x(T) \| = N \cdot \| x_{k+1} - x(t_{k+1}) \|, \quad N = \frac{T}{\Delta t},$$

so $$e = \frac{T}{\Delta t} \| x_{k+1} - x(t_{k+1}) \| \leq \frac{T}{\Delta t} \cdot \mathcal{O}(\Delta t^3) = \underline{\underline{\mathcal{O}(\Delta t^2)}}$$

$\boxed{2}$

a) ERK1 and ERK2 ~~has~~ have approximately the same accuracy, but are on the lower or upper side of the true trajectory. ERK4 is the most accurate.

b) See added plots. RK1 : order 1 : global error $\mathcal{O}(\Delta t)$

RK2 : order 2 : global error $\mathcal{O}(\Delta t^2)$

(RK3: order 3 : global error $\mathcal{O}(\Delta t^3)$)

RK4: order 4 : global error $\mathcal{O}(\Delta t^4)$

c) RK1 and RK2 becomes unstable at ~~about~~ $\approx \Delta t = 1{,}2$ with $\lambda = -2$ $\Rightarrow$

RK4 $\approx \Delta t = 2{,}5$. (more like $\approx \Delta t = 2{,}12$)

Table of Contents

Setup sim 1
SIM: ERK1 1
SIM2: ERK2 1
SIM3: ERK4 1
Plot results 2

Setup sim

```
clear; close all;
lambda = -2;
f = @(t, x) lambda*x;
dt = 0.4;
T0 = 0;
Tf = 2;
T = linspace(T0, Tf, (Tf - T0) / dt);
Nt = length(T);
x0 = 1;
X.true = x0*exp(lambda*T);
```

SIM: ERK1

```
b = 1;
c = 0;
A = 0;
BT = struct('A', A, 'b', b, 'c', c);
X.erk1 = ERKTemplate(BT, f, T, dt, x0);
```

SIM2: ERK2

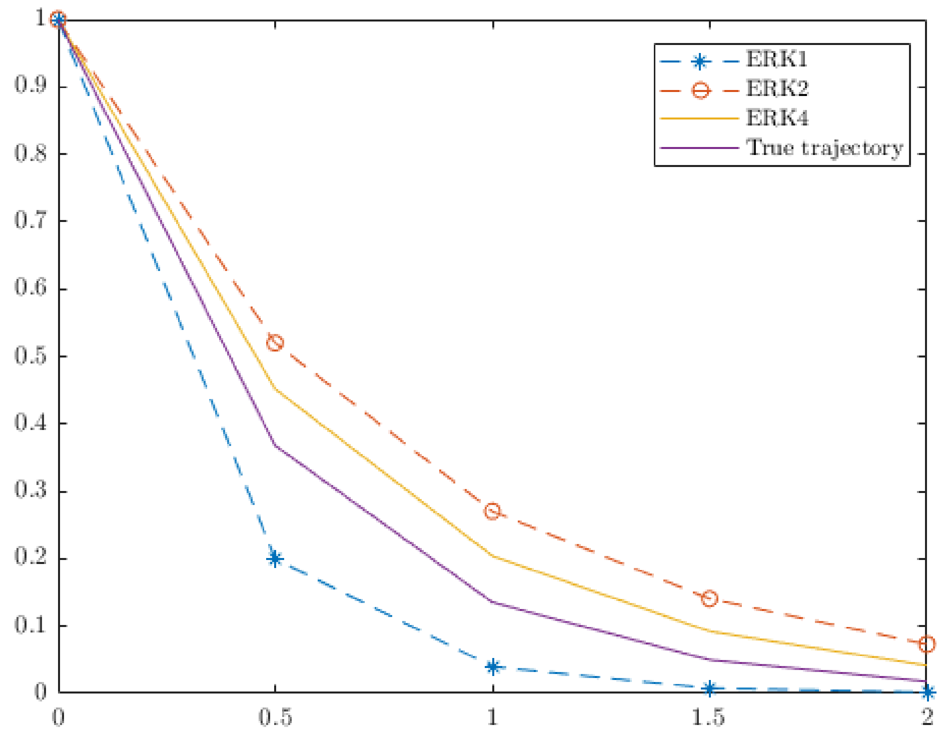
```
b = [0 1]';
c = [0 1/2]';
A = [  0, 0;
     1/2, 0];
BT = struct('A', A, 'b', b, 'c', c);
X.erk2 = ERKTemplate(BT, f, T, dt, x0);
```

SIM3: ERK4

```
b = [1/6, 1/3, 1/3, 1/6]';
c = [0 1/2 1/2 1]';
A = [  0,   0, 0, 0;
     1/2,   0, 0, 0;
       0, 1/2, 0, 0;
       0,   0, 1, 0];
BT = struct('A', A, 'b', b, 'c', c);
X.erk4 = ERKTemplate(BT, f, T, dt, x0);
```

Plot results

```
figure(1); clf;
    plot(T, X.erk1(:), '--*');
    hold on;
    plot(T, X.erk2(:), '--o');
    hold on;
    plot(T, X.erk4(:));
    hold on;
    plot(T, X.true(:));
    hold on;
    legend("ERK1", "ERK2", "ERK4", "True
 trajectory", "location", "best");
```



Published with MATLAB® R2019a

Table of Contents

Setup sim 1
SIM: ERK1 1
SIM2: ERK2 1
SIM3: ERK4 1
Get errors 2
Plot results 5

Setup sim

```
clear; close all;
lambda = -2;
f = @(t, x) lambda*x;
dt = linspace(0.4, 2.5, 12);
T0 = 0;
Tf = 10;
x0 = 1;
n_dt = length(dt);
T = cell(n_dt, 1);
for i = 1:n_dt
    T{i} = linspace(T0, Tf, (Tf - T0) / dt(i));
end
```

SIM: ERK1

```
b = 1;
c = 0;
A = 0;
BT = struct('A', A, 'b', b, 'c', c);
X.erk1 = cell(n_dt, 1);
for i = 1:n_dt
    X.erk1{i} = ERKTemplate(BT, f, T{i}, dt(i), x0);
end
```

SIM2: ERK2

```
b = [0 1]';
c = [0 1/2]';
A = [  0, 0;
     1/2, 0];
BT = struct('A', A, 'b', b, 'c', c);
X.erk2 = cell(n_dt, 1);
for i = 1:n_dt
    X.erk2{i} = ERKTemplate(BT, f, T{i}, dt(i), x0);
end
```

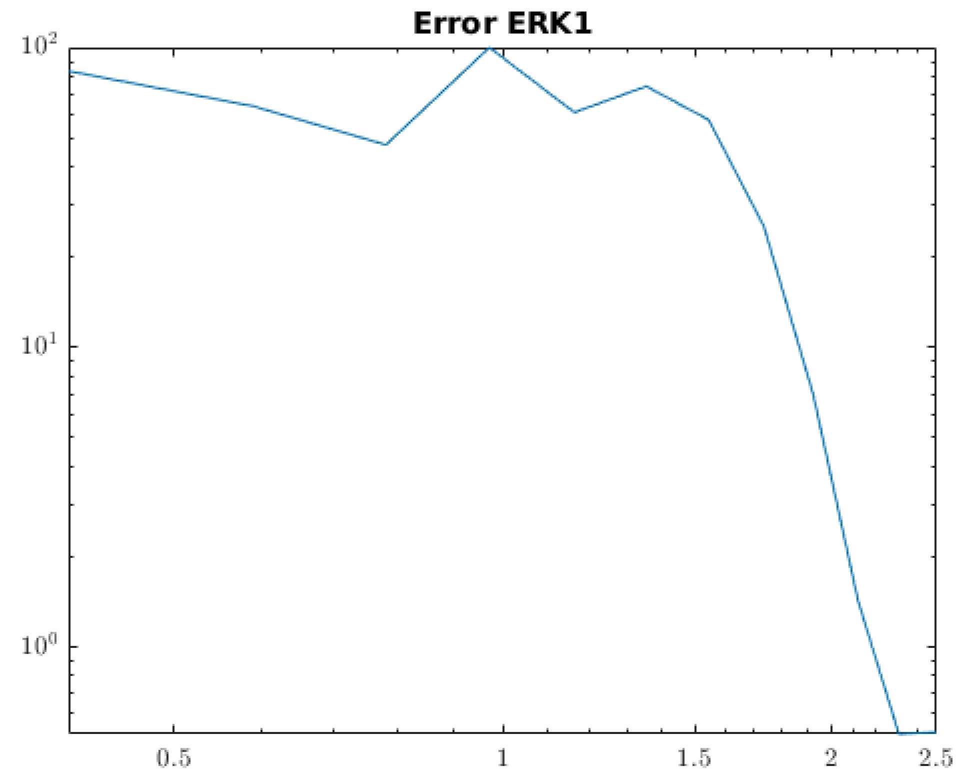
SIM3: ERK4

```
b = [1/6, 1/3, 1/3, 1/6]';
```

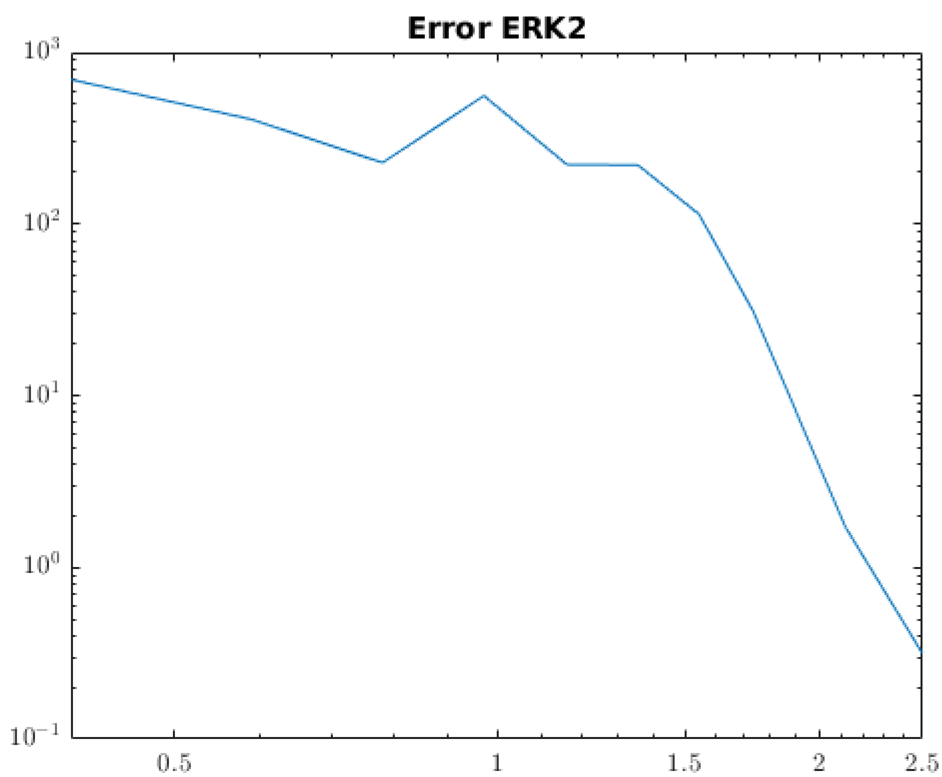
```
c = [0 1/2 1/2 1]';
A = [  0,    0, 0, 0;
     1/2,    0, 0, 0;
       0, 1/2, 0, 0;
       0,    0, 1, 0];
BT = struct('A', A, 'b', b, 'c', c);
X.erk4 = cell(n_dt, 1);
for i = 1:n_dt
    X.erk4{i} = ERKTemplate(BT, f, T{i}, dt(i), x0);
end
```

Get errors

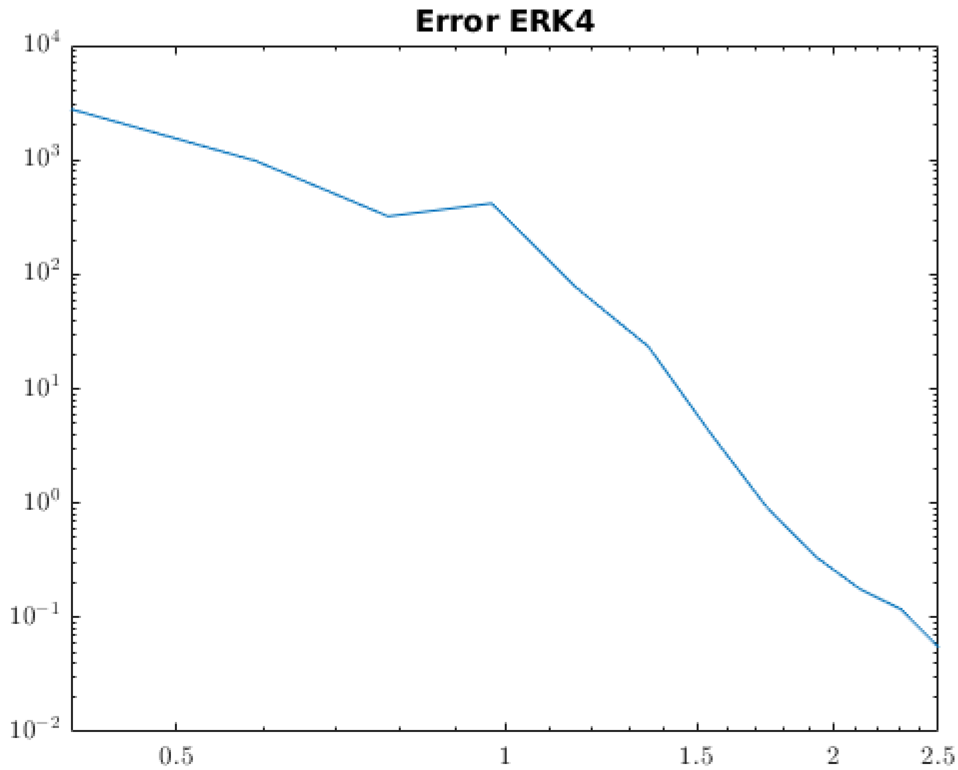
```
e.erk1 = zeros(size(X.erk1));
for i = 1:size(X.erk1, 1)
   e.erk1(i) = sum(abs(X.erk1{i} - x0*exp(lambda*T{i})));
end
figure(4); clf;
    loglog(dt(end:-1:1), e.erk1);
    title("Error ERK1");


e.erk2 = zeros(size(X.erk2));
for i = 1:size(X.erk2, 1)
   e.erk2(i) = sum(abs(X.erk2{i} - x0*exp(lambda*T{i})));
end
figure(5); clf;
    loglog(dt(end:-1:1), e.erk2);
    title("Error ERK2");

e.erk4 = zeros(size(X.erk4));
for i = 1:size(X.erk4, 1)
   e.erk4(i) = sum(abs(X.erk4{i} - x0*exp(lambda*T{i})));
end
figure(6); clf;
    loglog(dt(end:-1:1), e.erk4);
    title("Error ERK4");


figure(7); clf;
    loglog(dt(end:-1:1), e.erk1); hold on;
    loglog(dt(end:-1:1), e.erk2); hold on;
    loglog(dt(end:-1:1), e.erk4);
    title("Errors");
    legend("ERK1", "ERK2", "ERK4")
```

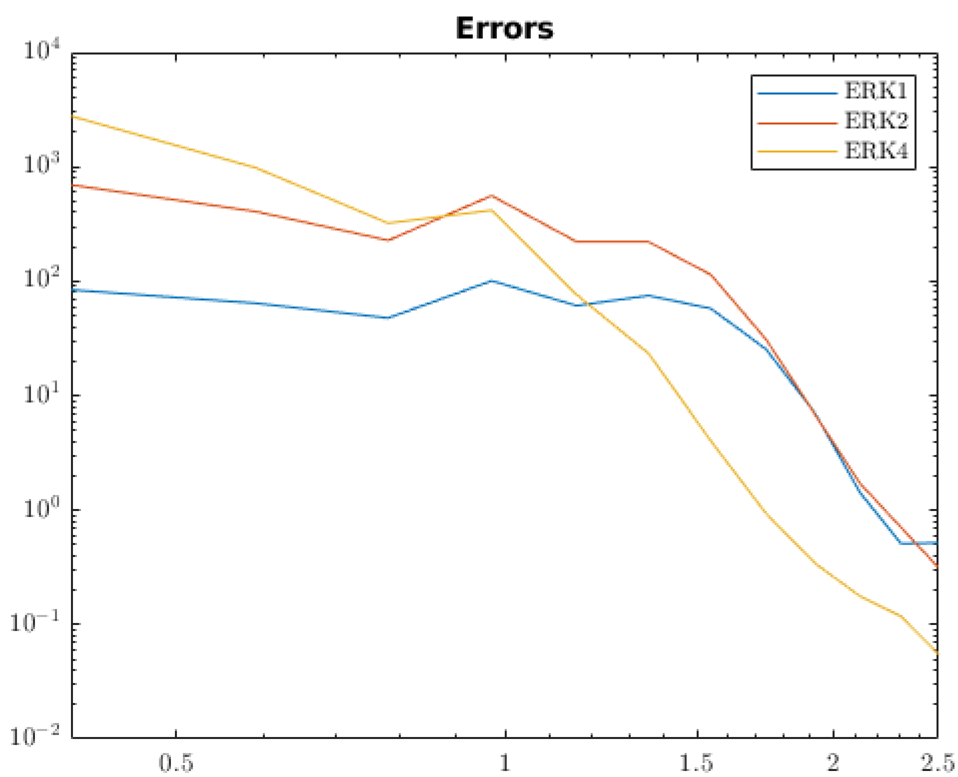


Error ERK1
10^2
10^1
10^0
0.5
1
1.5
2
2.5



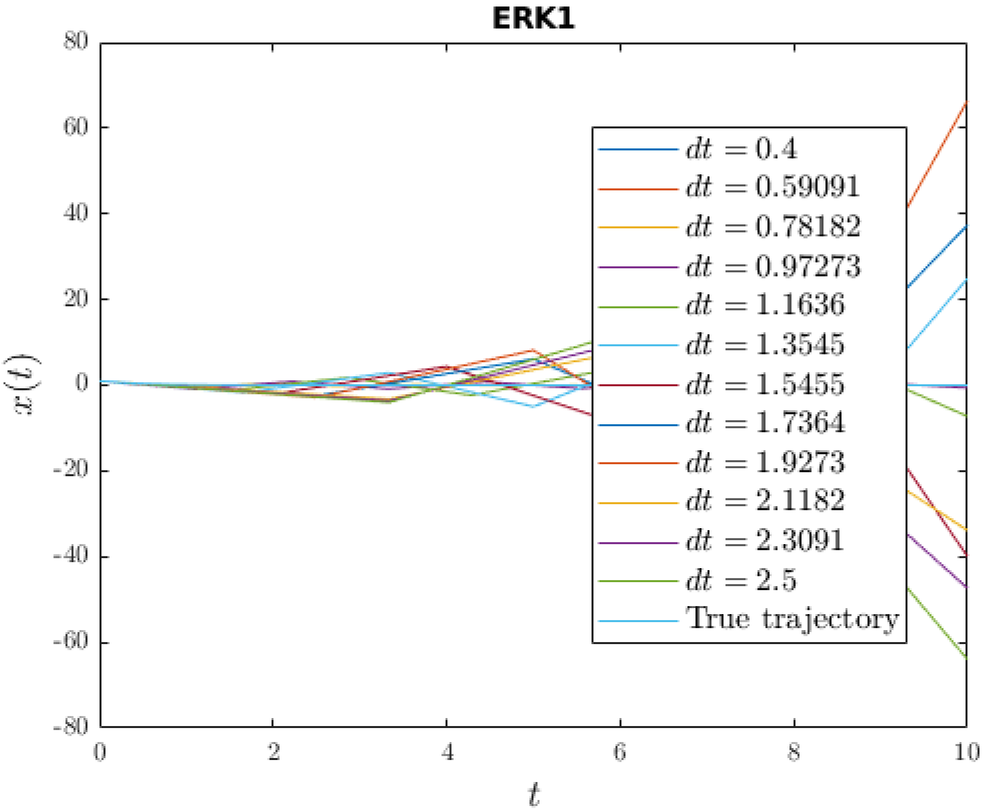
Error ERK2
10^3
10^2
10^1
10^0
10^{-1}
0.5
1
1.5
2
2.5

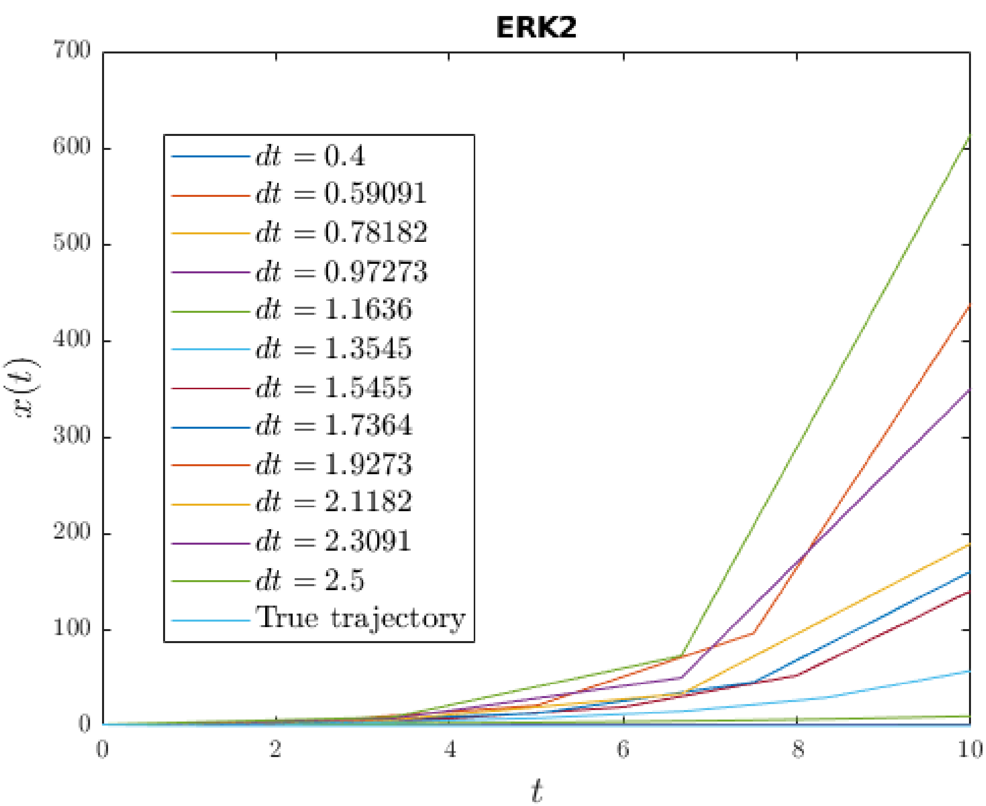

Error ERK4
10^4
10^3
10^2
10^1
10^0
10^{-1}
10^{-2}
0.5
1
1.5
2
2.5

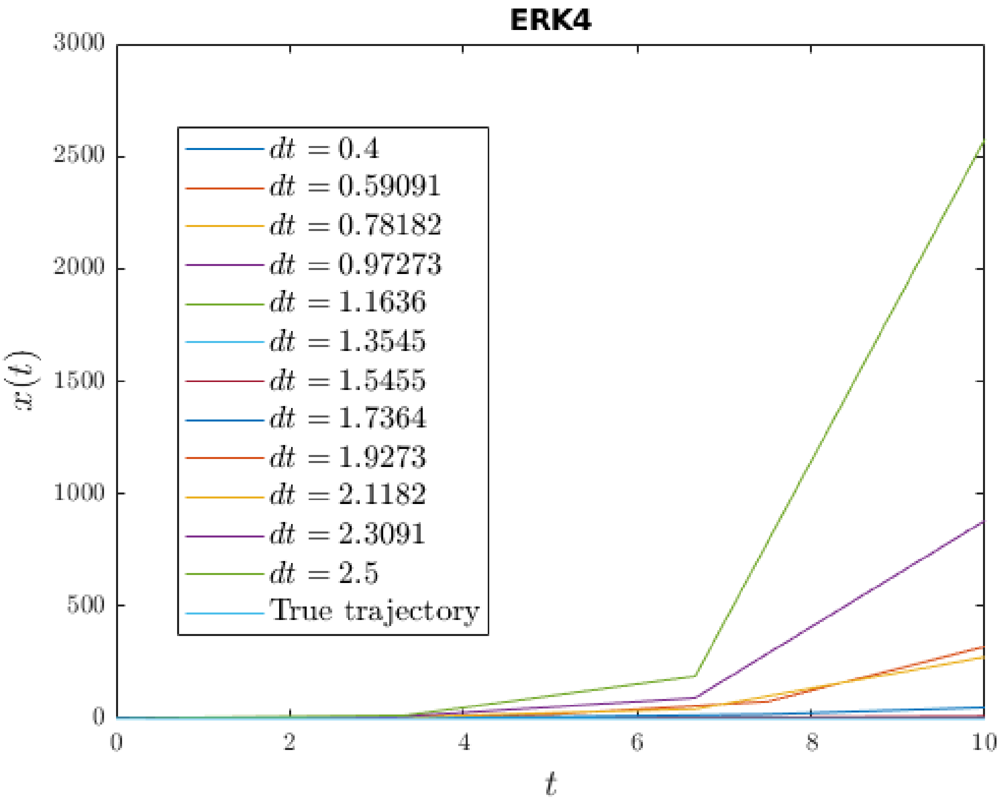

Errors
ERK1
ERK2
ERK4
10^4
10^3
10^2
10^1
10^0
10^{-1}
10^{-2}
0.5
1
1.5
2
2.5

Plot results

```
set(0, 'defaultAxesTickLabelInterpreter','latex');
 set(0,'defaultLegendInterpreter','latex');

figure(1); clf;
    for i = 1:n_dt
       plot(T{i}, X.erk1{i}, 'DisplayName', "$dt="
 +string(dt(i))+"$");
       hold on;
    end
       plot(T{1}, x0*exp(lambda*T{1}), 'DisplayName', 'True
 trajectory');
       title('ERK1');
       xlabel("$t$", 'Interpreter', 'latex', 'fontsize', 14);
       ylabel("$x(t)$", 'Interpreter', 'latex', 'fontsize', 14);
       legend('fontsize', 12, 'location', 'best');


figure(2); clf;
    for i = 1:n_dt
       plot(T{i}, X.erk2{i}, 'DisplayName', "$dt="
 +string(dt(i))+"$");
       hold on;
    end
       plot(T{1}, x0*exp(lambda*T{1}), 'DisplayName', 'True
 trajectory');
       title('ERK2');
       xlabel("$t$", 'Interpreter', 'latex', 'fontsize', 14);
       ylabel("$x(t)$", 'Interpreter', 'latex', 'fontsize', 14);
       legend('fontsize', 12, 'location', 'best');

figure(3); clf;
    for i = 1:n_dt
       plot(T{i}, X.erk4{i}, 'DisplayName', "$dt="
 +string(dt(i))+"$");
       hold on;
    end
       plot(T{1}, x0*exp(lambda*T{1}), 'DisplayName', 'True
 trajectory');
       title('ERK4');
       xlabel("$t$", 'Interpreter', 'latex', 'fontsize', 14);
       ylabel("$x(t)$", 'Interpreter', 'latex', 'fontsize', 14);
       legend('fontsize', 12, 'location', 'best');



function e = computeError(x_sim, x_true)

    e = abs(x_sim - x_true);

end
```


ERK1
80
60
40
20
0
-20
-40
-60
-80
0
2
4
6
8
10
t
$x(t)$
dt = 0.4
dt = 0.59091
dt = 0.78182
dt = 0.97273
dt = 1.1636
dt = 1.3545
dt = 1.5455
dt = 1.7364
dt = 1.9273
dt = 2.1182
dt = 2.3091
dt = 2.5
True trajectory


ERK2
700
600
500
400
300
200
100
0
0
2
4
6
8
10
t
$x(t)$
dt = 0.4
dt = 0.59091
dt = 0.78182
dt = 0.97273
dt = 1.1636
dt = 1.3545
dt = 1.5455
dt = 1.7364
dt = 1.9273
dt = 2.1182
dt = 2.3091
dt = 2.5
True trajectory



Published with MATLAB® R2019a