# Task 5-6

## Defines file path

```
addpath('classes');
```

## Task description

Bruk implementeringen av RK4 eller RKF45 til å løse systemet. Benytt treghetsmomentent til T-nøkkelen som ble utregnet i tidligere oppgave. Bruk $X(0) = I$ (3 × 3 identitetsmatrisen). Beregn $L$ slik at $\omega(0)$ har lengde ca lik 1 ved tiden 0. Kjør eksperimentet med:

- a) $\omega(0) = [1, 0.05, 0]^T$
- b) $\omega(0) = [0,\ 1, 0.05]^T$
- c) $\omega(0) = [0.05,\ 0,\ 1]^T$

Steglengden lar dere være så liten at energen ikke endrer seg nevneverdig

Tegn opp komponentene til løsningene X fra oppgave 5 som ni funksjoner av tiden. Gi en tolkning av disse grafene. Alternativt kan dere lage en 3-D animasjon av T-nøkkelen som roterer

## Parameter initialization

```
R1 = 1; R2 = 1; % Radius
L1 = 8; L2 = 4; % Length
p = 6.7;         % Mass density
M = 12*pi * p;   % Mass

X0 = eye(3); % X-matrix

h = 0.01;     % Initial step length
n = 10000;    % Number of iterations
TOL = 1e-30; % Tolerance
```

## Calculates moment of inertia

```
tHandle = THandle(R1, R2, L1, L2, M, p);
I = tHandle.calculateMomentOfInertia();
```

# Task a

## Parameter initialization

```
w0 = [1, 0.05, 0]'; % Rotation vector
L = X0 * I * w0;     % Torque vector
```

## RKF45

### Approximates solution

```
rkf45 = RKF45(h, n, TOL);
[t, W, E, h] = rkf45.solve(X0, I, L);
fprintf('Max Error: %d', max(E));
```

```
Max Error: 9.795761e-31
```

### Calculates energy

```
energy = zeros(length(W), 1);

for i = 1:length(W)
    w = (W{i} * I)^(-1) * L;
    energy(i) = Energy.calculate(L, W{i} * w);
end

fprintf('Initial energy: %i \nFinal energy:   %i \nDifference:     %i', energy(1), ener
```

```
Initial energy: 4.920390e+02
Final energy:   4.920390e+02
Difference:     1.149231e-05
```

### Visualization

```
v = Visualization(tHandle, '6a/RKF45');
v.plotX(t, W);
v.plotH(t, h);
v.plotHX(t, h, W, 1);
v.plotError(t, E);
v.plotErrorX(t, W, 1, E);
v.plotEnergy(t, energy);
v.plotErrorEnergy(t, energy, E, 8.0*10^(25));
v.plotAccumulativeError(t, E);
v.plotAccumulativeErrorX(t, W, 1, E);
v.plotAccumulativeErrorEnergy(t, energy, E, 9*10^(20));
v.animateTHandle(W, t(end) / length(W));
```

## RK4, h = 0.1

## Approximates solution

```
h = 0.1; n = 1000;
rk4 = RK4(h, n);
[t, W, E] = rk4.solve(X0, I, L);
fprintf('Max Error: %d', max(E));
```

```
Max Error: 5.597257e-15
```

## Calculates energy

```
energy = zeros(n, 1);

for i = 1:length(W)
    w = (W{i} * I)^(-1) * L;
    energy(i) = Energy.calculate(L, W{i} * w);
end

fprintf('Initial energy: %i \nFinal energy:   %i \nDifference:     %i', energy(1), ener
```

```
Initial energy: 4.920390e+02
Final energy:   4.921088e+02
Difference:     6.982597e-02
```

## Visualization

```
v = Visualization(tHandle, '6a/RK4/0.1');
v.plotX(t, W);
v.plotError(t, E);
v.plotErrorX(t, W, 1, E);
v.plotEnergy(t, energy);
v.plotErrorEnergy(t, energy, E, 1.15*10^(13));
v.plotAccumulativeError(t, E);
v.plotAccumulativeErrorX(t, W, 1, E);
v.plotAccumulativeErrorEnergy(t, energy, E, 5.7*10^(10));
```

## RK4, h = 0.01

## Approximates solution

```
h = 0.01; n = 10000;
rk4 = RK4(h, n);
[t, W, E] = rk4.solve(X0, I, L);
fprintf('Max Error: %d', max(E));
```

```
Max Error: 5.567912e-25
```

## Calculates energy

```
energy = zeros(n, 1);

for i = 1:length(W)
    w = (W{i} * I)^(-1) * L;
    energy(i) = Energy.calculate(L, W{i} * w);
end

fprintf('Initial energy: %i \nFinal energy:   %i \nDifference:     %i', energy(1), ener
```

```
Initial energy: 4.920390e+02
Final energy:   4.920391e+02
Difference:     8.296730e-05
```

## Visualization

```
v = Visualization(tHandle, '6a/RK4/0.01');
v.plotX(t, W);
v.plotError(t, E);
v.plotErrorX(t, W, 1, E);
v.plotEnergy(t, energy);
v.plotErrorEnergy(t, energy, E, 1.0*10^(21));
v.plotAccumulativeError(t, E);
v.plotAccumulativeErrorX(t, W, 1, E);
v.plotAccumulativeErrorEnergy(t, energy, E, 5.5*10^(16));
```

## RK4, h = 0.001

## Approximates solution

```
h = 0.001; n = 100000;
rk4 = RK4(h, n);
[t, W, E] = rk4.solve(X0, I, L);
fprintf('Max Error: %d', max(E));
```

```
Max Error: 4.949640e-32
```

## Calculates energy

```
energy = zeros(n, 1);

for i = 1:length(W)
    w = (W{i} * I)^(-1) * L;
    energy(i) = Energy.calculate(L, W{i} * w);
end

fprintf('Initial energy: %i \nFinal energy:   %i \nDifference:     %i', energy(1), ener
```

```
Initial energy: 4.920390e+02
Final energy:   4.920390e+02
```

```
Difference:     2.603802e-07
```

### Visualization

```
v = Visualization(tHandle, '6a/RK4/0.001');
v.plotX(t, W);
v.plotError(t, E);
v.plotErrorX(t, W, 1, E);
v.plotEnergy(t, energy);
v.plotErrorEnergy(t, energy, E, 5.0*10^(25));
v.plotAccumulativeError(t, E);
v.plotAccumulativeErrorX(t, W, 1, E);
v.plotAccumulativeErrorEnergy(t, energy, E, 1.1*10^(21));
```

## Task b

### Parameter initialization

```
h = 0.01; n = 10000;
w0 = [0, 1, 0.05]'; % Rotation vector
L = X0 * I * w0;    % Torque vector
```

### RKF45

### Approximates solution

```
rkf45 = RKF45(h, n, TOL);
[t, W, E, h] = rkf45.solve(X0, I, L);
fprintf('Max Error: %d', max(E));
```

```
Max Error: 8.369184e-31
```

### Calculates energy

```
energy = zeros(length(W), 1);

for i = 1:length(W)
    w = (W{i} * I)^(-1) * L;
    energy(i) = Energy.calculate(L, W{i} * w);
end

fprintf('Initial energy: %i \nFinal energy:   %i \nDifference:     %i', energy(1), ener
```

```
Initial energy: 3.633088e+02
Final energy:   3.633088e+02
Difference:     8.599500e-07
```

## Visualization

```
v = Visualization(tHandle, '6b/RKF45');
v.plotX(t, W);
v.plotH(t, h);
v.plotError(t, E);
v.plotEnergy(t, energy);
v.plotAccumulativeError(t, E);
v.animateTHandle(W, t(end) / length(W));
```

## RK4

## Approximates solution

```
h = 0.01;
rk4 = RK4(h, n);
[t, W, E] = rk4.solve(X0, I, L);
fprintf('Max Error: %d', max(E));
```

```
Max Error: 3.716739e-28
```

## Calculates energy

```
energy = zeros(n, 1);

for i = 1:length(W)
    w = (W{i} * I)^(-1) * L;
    energy(i) = Energy.calculate(L, W{i} * w);
end

fprintf('Initial energy: %i \nFinal energy:   %i \nDifference:     %i', energy(1), ener
```

```
Initial energy: 3.633088e+02
Final energy:   3.633088e+02
Difference:     1.697041e-06
```

## Visualization

```
v = Visualization(tHandle, '6b/RK4');
v.plotX(t, W);
v.plotError(t, E);
v.plotEnergy(t, energy);
v.plotAccumulativeError(t, E);
```

## Task c

## Parameter initialization

```
w0 = [0.05, 0, 1]'; % Rotation vector
L = X0 * I * w0;    % Torque vector
```

# RKF45

## Approximates solution

```
rkf45 = RKF45(h, n, TOL);
[t, W, E, h] = rkf45.solve(X0, I, L);
fprintf('Max Error: %d', max(E));
```

```
Max Error: 5.828470e-31
```

## Calculates energy

```
energy = zeros(length(W), 1);

for i = 1:length(W)
    w = (W{i} * I)^(-1) * L;
    energy(i) = Energy.calculate(L, W{i} * w);
end

fprintf('Initial energy: %i \nFinal energy:   %i \nDifference:     %i', energy(1), ener
```

```
Initial energy: 7.905530e+02
Final energy:   7.905530e+02
Difference:     3.418527e-07
```

## Visualization

```
v = Visualization(tHandle, '6c/RKF45');
v.plotX(t, W);
v.plotH(t, h);
v.plotError(t, E);
v.plotEnergy(t, energy);
v.plotAccumulativeError(t, E);
v.animateTHandle(W, t(end) / length(W));
```

# RK4

## Approximates solution

```
h = 0.01;
```

```
rk4 = RK4(h, n);
[t, W, E] = rk4.solve(X0, I, L);
fprintf('Max Error: %d', max(E));
```

```
Max Error: 3.018502e-25
```

## Calculates energy

```
energy = zeros(n, 1);

for i = 1:length(W)
    w = (W{i} * I)^(-1) * L;
    energy(i) = Energy.calculate(L, W{i} * w);
end

fprintf('Initial energy: %i \nFinal energy:   %i \nDifference:     %i', energy(1), ener
```

```
Initial energy: 7.905530e+02
Final energy:   7.905530e+02
Difference:     5.764557e-06
```

## Visualization

```
v = Visualization(tHandle, '6c/RK4');
v.plotX(t, W);
v.plotError(t, E);
v.plotEnergy(t, energy);
v.plotAccumulativeError(t, E);
```