

# Task 1

## Defines the file path

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

## Task a

### Task description

Implementer eksponentfunksjonen i likning (21). Input skal være en matrise på formen (18). Funksjonen bør ta inn både  $h$  og  $\Omega$  som input. Lag et testprogram som sjekker om output  $X$  er en matrise som oppfyller kravet

$$X^T X = I$$

### Parameter initialization

```
% Omega matrix
o = [ 0  -3   2
      3   0  -1
     -2   1   0 ];

h = 0.1;           % Step size
X = Exp.e(h, o) % X-Matrix
```

```
X = 3x3
    0.9358   -0.2832    0.2102
    0.3029    0.9506   -0.0680
   -0.1805    0.1273    0.9753
```

### Tests the Exp-function

```
TOL = 1e-12;
Exp.test(X, TOL)
```

```
ans = logical
     1
```

Answer is 1, which is equal to true. In other words, the matrix  $X$  fulfills the requirement of  $X^T X = I$  within a tolerance of  $1e-12$ .

## Task b

### Task description

Implementer en funksjon som regner ut energien til et roterende legeme som har treghetsmoment/og rotasjonsvektor  $\vec{\omega}$ . Denne vil dere trenge til å sjekke om simuleringene bevarer energien til T-nøkkelen.

## Parameter initialization

```
w = [1, 0, 0]'; % Rotation vector
L = [1, 0, 0]'; % Torque vector
```

## Calculates the energy

```
K = Energy.calculate(L, w)
```

```
K = 0.5000
```

Total energy for this system is 0.5.

## Task c

### Task description

Regn ut treghetsmomentet til T-nøkkelen ved å bruke formelen i forrige avsnitt.

## Parameter initialization

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

## Calculates the moment of inertia

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

```
I = 3x3
103 ×
    0.9823         0         0
         0    0.7227         0
         0         0    1.5787
```

The moment of inertia for the T-Handle is given above.