Beispiel: Rayleigh-Quotient an einem Siebenmassenschwinger beim Ersatzkraftverfahren

```
from sympycalcs.helpers import Equation as Eqn
from sympycalcs import render, convert
import sympy as sp

import sympy.physics.units as unit
from sympy.abc import *
sp.init_printing(use_latex='mathjax', latex_mode='equation')
```

Aufgabenstellung

Das Beispiel ist aus @Dupraz2004 Seite 77 entnommen. Die Lastermittlung wird übernommen. Es wird vom vereinfachten Modell in Figure 1 ausgegangen. Der Fokus liegt auf der Bestimmung der Grundfrequenz.

Gesucht:

- Erstelle die Nachgiebigkeitsmatrix (Verwende das Bildungsgesetz)
- Grundschwingzeit

Gegeben:

```
H_i, m_i,  I, I_x, I_y, f= sp.symbols('H_i, m_i  I  I_x I_y f')
n = 6 # n-Massenschwinger
F = sp.MatrixSymbol('\mathbf{F}', n,1)
u = sp.MatrixSymbol('\mathbf{u}', n,1)
f_hat = sp.MatrixSymbol('\hat{\mathbf{f}}', n,n)
M = sp.MatrixSymbol('\mathbf{M}', n,1)
```

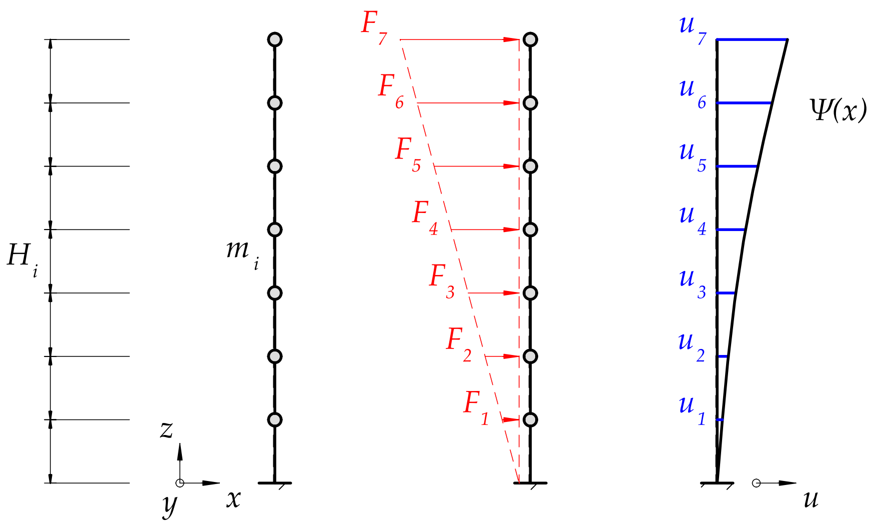

Figure 1: Vereinfachtes Modell für einen Mehrmassenschwinger

```
params={
    E: 27*10**9*unit.N/ unit.m**2,
    H_i: 3.105*unit.m,
    m_i: 1278*10**3*unit.N*unit.second**2 /unit.m,
    I_x: 14.89*unit.m**4,
    I_y: 28.27*unit.m**4,
}

params_plot = convert.param_value(params)

render.dict_to_table(params)
```

$E = \frac{27000000000\mathrm{N}}{\mathrm{m}^2}$	$H_i = 3.105\mathrm{m}$
$I_x = 14.89\mathrm{m}^4$	$I_y = 28.27\mathrm{m}^4$
$m_i = \frac{1278000\mathrm{Ns}^2}{\mathrm{m}}$	

Musterlösung

Rayleigh-Quotient

```
def nachgiebigkeitsmatrix_nach_bildungsgesetz(n):
    """
    Erstellt die Nachgiebigkeitsmatrix nach dem Bildungsgesetz. Dieses ist nur zulässig fü
    Dies muss abschliessend mit h^3/(6*E*I) mutlipliziert werden.
    """
    from sympy import symbols, zeros


    def bildungsgesetz(i,j):
        return j**2*(3*i-j)

    matrix = sp.zeros(n, n)
    matrix_symbols =sp.zeros(n,n)
    matrix_unsymm = sp.zeros(n, n)

    for i in range(1,n+1):
        for j in range(1,n+1):
            if i >=j:
                matrix[i-1,j-1] = bildungsgesetz(i,j)
                matrix_symbols[i-1,j-1] = sp.Symbol(f'f_{i,j}')
                matrix_unsymm[i-1,j-1] = bildungsgesetz(i,j)

            if j>i:
                matrix[i-1, j-1] = bildungsgesetz(j,i)
                matrix_symbols[i-1,j-1] = sp.Symbol(f'f_{i,j}')


    return matrix, matrix_symbols, matrix_unsymm
```

```
eq_f_hat = Eqn(f_hat, sp.UnevaluatedExpr(H_i**3/(6*E*I))*nachgiebigkeitsmatrix_nach_bildun

eq_F = Eqn(F,sp.Matrix(list(range(1, n+1))))
eq_u = Eqn(u,f_hat * F)
```

```
eq_M = Eqn(M,sp.ones(n,1)*m_i)
```

```
eq_f = Eqn(f,sp.Mul(sp.UnevaluatedExpr(1 / (2*sp.pi)),sp.sqrt(sp.Sum(sp.HadamardProduct(F,

eq_T = Eqn(T,1 / f)
```

Mittels dem Rayleigh-Quotient für das vereinfachte Modell lässt sich die Grundfrequenz direkt bestimmen.

```
display(eq_f, eq_F, eq_M, eq_T)
```

$$f = \frac{1}{2\pi}\sqrt{\frac{\sum_{i=0}^{5} \mathbf{F}_{i,0}\mathbf{u}_{i,0}}{\sum_{i=0}^{5} \mathbf{M}_{i,0}\mathbf{u}_{i,0}^{2}}} \tag{1}$$

$$\mathbf{F} = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{bmatrix} \tag{2}$$

$$\mathbf{M} = \begin{bmatrix} m_i \\ m_i \\ m_i \\ m_i \\ m_i \\ m_i \end{bmatrix} \tag{3}$$

$$T = \frac{1}{f} \tag{4}$$

Dabei entspricht $\mathbf{u}$ dem Verschiebungsvektor infolge des Kraftvektors $\mathbf{F}$. Der Verschiebungsvektor kann mittels Nachgiebigkeitsmatrix bestimmt werden.

Nachgiebigkeitsmatrix

Für gleichbleibende Geschosshöhen und Geschosssteifigkeiten lässt sich die Nachgiebigkeitsmatrix leicht mittels dem Bildungsgesetz in Equation 5 ermitteln.

$$\hat{f}_{i,j} = \frac{H^3}{6EI} \cdot j^2(3i - j) \text{ für } i \geq j \tag{5}$$

$\hat{\mathbf{f}}$ entspricht der Nachgiebigkeitsmatrix mit den Einträgen $\hat{f}_{i,j}$.

Beachte dabei dass die Equation 5 nur für $i \geq j$ gilt. Die Einträge entsprechen folgendem Schema:

```
display(nachgiebigkeitsmatrix_nach_bildungsgesetz(n)[1])
```

$$\left[\begin{matrix}f_{(1,1)} & f_{(1,2)} & f_{(1,3)} & f_{(1,4)} & f_{(1,5)} & f_{(1,6)}\\f_{(2,1)} & f_{(2,2)} & f_{(2,3)} & f_{(2,4)} & f_{(2,5)} & f_{(2,6)}\\f_{(3,1)} & f_{(3,2)} & f_{(3,3)} & f_{(3,4)} & f_{(3,5)} & f_{(3,6)}\\f_{(4,1)} & f_{(4,2)} & f_{(4,3)} & f_{(4,4)} & f_{(4,5)} & f_{(4,6)}\\f_{(5,1)} & f_{(5,2)} & f_{(5,3)} & f_{(5,4)} & f_{(5,5)} & f_{(5,6)}\\f_{(6,1)} & f_{(6,2)} & f_{(6,3)} & f_{(6,4)} & f_{(6,5)} & f_{(6,6)}\end{matrix}\right] \tag{6}$$

Unter strikter Anwendung von Equation 5 folgt daraus:

```
display(nachgiebigkeitsmatrix_nach_bildungsgesetz(n)[2])
```

$$\left[\begin{matrix}2 & 0 & 0 & 0 & 0 & 0\\5 & 16 & 0 & 0 & 0 & 0\\8 & 28 & 54 & 0 & 0 & 0\\11 & 40 & 81 & 128 & 0 & 0\\14 & 52 & 108 & 176 & 250 & 0\\17 & 64 & 135 & 224 & 325 & 432\end{matrix}\right] \tag{7}$$

Aufgrund von Symmetrie kann diese abschliessend über die Diagonale gespiegelt werden:

```
display(eq_f_hat/sp.UnevaluatedExpr(H_i**3 /(6*E*I)))
```

$$\frac{6EI}{H_i^3}\hat{\mathbf{f}} = \left[\begin{matrix}2 & 5 & 8 & 11 & 14 & 17\\5 & 16 & 28 & 40 & 52 & 64\\8 & 28 & 54 & 81 & 108 & 135\\11 & 40 & 81 & 128 & 176 & 224\\14 & 52 & 108 & 176 & 250 & 325\\17 & 64 & 135 & 224 & 325 & 432\end{matrix}\right] \tag{8}$$

Durch Multiplikation der *Nachgiebigkeit* mit der *Einwirkung* resultiert die Deformation.

```
display(eq_u, eq_u.subs(eq_f_hat, eq_F),eq_u.subs(eq_f_hat, eq_F).doit())
```

$$\mathbf{u} = \hat{\mathbf{f}}\mathbf{F} \tag{9}$$

$$\mathbf{u} = \begin{bmatrix} 2\frac{H_i^3}{6EI} & 5\frac{H_i^3}{6EI} & 8\frac{H_i^3}{6EI} & 11\frac{H_i^3}{6EI} & 14\frac{H_i^3}{6EI} & 17\frac{H_i^3}{6EI} \\ 5\frac{H_i^3}{6EI} & 16\frac{H_i^3}{6EI} & 28\frac{H_i^3}{6EI} & 40\frac{H_i^3}{6EI} & 52\frac{H_i^3}{6EI} & 64\frac{H_i^3}{6EI} \\ 8\frac{H_i^3}{6EI} & 28\frac{H_i^3}{6EI} & 54\frac{H_i^3}{6EI} & 81\frac{H_i^3}{6EI} & 108\frac{H_i^3}{6EI} & 135\frac{H_i^3}{6EI} \\ 11\frac{H_i^3}{6EI} & 40\frac{H_i^3}{6EI} & 81\frac{H_i^3}{6EI} & 128\frac{H_i^3}{6EI} & 176\frac{H_i^3}{6EI} & 224\frac{H_i^3}{6EI} \\ 14\frac{H_i^3}{6EI} & 52\frac{H_i^3}{6EI} & 108\frac{H_i^3}{6EI} & 176\frac{H_i^3}{6EI} & 250\frac{H_i^3}{6EI} & 325\frac{H_i^3}{6EI} \\ 17\frac{H_i^3}{6EI} & 64\frac{H_i^3}{6EI} & 135\frac{H_i^3}{6EI} & 224\frac{H_i^3}{6EI} & 325\frac{H_i^3}{6EI} & 432\frac{H_i^3}{6EI} \end{bmatrix} \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{bmatrix} \tag{10}$$

$$\mathbf{u} = \begin{bmatrix} \frac{42H_i^3}{EI} \\ \frac{925H_i^3}{6EI} \\ \frac{950H_i^3}{3EI} \\ \frac{1535H_i^3}{3EI} \\ \frac{2173H_i^3}{3EI} \\ \frac{5663H_i^3}{6EI} \end{bmatrix} \tag{11}$$

Durch einsetzen der bestimmten Verformung in die Gleichung der Eigenfrequenz folgt:

```
eq_f_subs = eq_f.subs(eq_F, eq_u, eq_M).subs(eq_f_hat, eq_F).doit()
eq_T_subs = eq_T.subs(eq_f_subs)

display(eq_f_subs, eq_T_subs)
```

$$f = \frac{\sqrt{43665341610}\sqrt{\frac{EI}{H_i^3 m_i}}}{4993178\pi} \tag{12}$$

$$T = \frac{\sqrt{43665341610}\pi}{8745\sqrt{\frac{EI}{H_i^3 m_i}}} \tag{13}$$

Grundschwingzeit

X-Richtung

Es gilt I mit I_y zu substituieren.

```
u_x = eq_u.subs(eq_f_hat, eq_F).subs(I, I_y).doit()

f_x = eq_f_subs.subs(I, I_y)
T_x = eq_T_subs.subs(I,I_y)
```

```
render.eq_display(I_x, I_x.subs(params))
display(u_x, f_x.subs(params).evalf(3), T_x.subs(params).evalf(3))
```

$$I_x = 14.89\mathrm{m}^4 \tag{14}$$

$$\mathbf{u} = \begin{bmatrix} \frac{42H_i^3}{EI_y} \\ \frac{925H_i^3}{6EI_y} \\ \frac{950H_i^3}{3EI_y} \\ \frac{1535H_i^3}{3EI_y} \\ \frac{2173H_i^3}{3EI_y} \\ \frac{5663H_i^3}{6EI_y} \end{bmatrix} \tag{15}$$

$$f = \frac{1.88}{\mathrm{s}} \tag{16}$$

$$T = 0.531\mathrm{s} \tag{17}$$

Y-Richtung

Es gilt I mit I_x zu substituieren.

```
u_y = eq_u.subs(eq_f_hat, eq_F).subs(I, I_x).doit()
f_y = eq_f_subs.subs(I, I_x)
T_y = eq_T_subs.subs(I,I_x)


render.eq_display(I_y, I_y.subs(params))
display(u_y, f_y.subs(params).evalf(3), T_y.subs(params).evalf(3))
```

$$I_y = 28.27\mathrm{m}^4 \tag{18}$$

$$\mathbf{u} = \begin{bmatrix} \frac{42H_i^3}{EI_x} \\ \frac{925H_i^3}{6EI_x} \\ \frac{950H_i^3}{3EI_x} \\ \frac{1535H_i^3}{3EI_x} \\ \frac{2173H_i^3}{3EI_x} \\ \frac{5663H_i^3}{6EI_x} \end{bmatrix} \tag{19}$$

$$f = \frac{1.37}{\mathrm{s}} \tag{20}$$

$$T = 0.732\mathrm{s} \tag{21}$$

Abminderung der Steifigkeit

Um die Rissbildung zu berücksichtigen wird die Steifigkeit auf 30 % abgemindert.

X-Richtung

```
f_x_red = f_x.subs(E,0.3*E)
T_x_red = T_x.subs(E,0.3*E)

display(f_x_red.subs(params).evalf(3), T_x_red.subs(params).evalf(3))
```

$$f = \frac{1.03}{\mathrm{s}} \tag{22}$$

$$T = 0.97\mathrm{s} \tag{23}$$

Y-Richtung

```
f_y_red = f_y.subs(E,0.3*E)
T_y_red = T_y.subs(E,0.3*E)

display(f_y_red.subs(params).evalf(3), T_y_red.subs(params).evalf(3))
```

$$f = \frac{0.748}{\mathrm{s}} \tag{24}$$

$$T = 1.34\mathrm{s} \tag{25}$$