2 DOF System

```python
from numpy import array, cos, diag, exp, linspace, pi, poly1d, sin, sqrt
from scipy.linalg import inv, det, eigh
import matplotlib.pyplot as plt
from IPython.display import Latex
t = linspace(0, 20, 501)
```

First Part, Structural Matrices & Eigenvalues

Let's start with a few helper functions

```
def p(*l): return poly1d(l)
def i(p1, p2):
    p = (p1*p2).integ()
    return(p(1)-p(0))
def pmat(mat, fmt='%.3f'):
    inside =  r'\\'.join('&'.join(fmt%x for x in row) for row in mat)
    return r'\begin{bmatrix}'+inside+r'\end{bmatrix}'
def dl(s): display(Latex('$$'+s+'$$'))
```

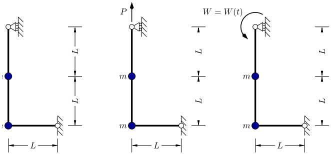
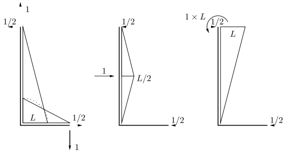


Figure 1*a*. Figure 1*b*. Figure 1*c*.

The beam is split in three intervals of definition for the bending moments, all intervals of length L: ❶ from the top to the middle of the vertical part, ❷ from the corner to the middle of the vertical part, ❸ from the hinge to the corner.

The bending moments, for a 1 vertical load, for a 1 horizontal load at midspan and for a $1 \times L$ couple applied to the upper support are

$$\begin{array}{llll} \text{vertical:} & +\frac{1}{2}x+0, & -\frac{1}{2}x+L, & +1x+0; \\ \text{horizontal:} & +\frac{1}{2}x+0, & +\frac{1}{2}x+0, & +0x+0; \\ \text{couple} & -\frac{1}{2}x+L, & +\frac{1}{2}x+0, & +0x+0. \end{array}$$



The bending moments are here expressed as polynomials (`p(...)`) and collected in a list for each load conditions, those lists collected in an outer list `m`.

```
m = [
        [p( 0.5, 0), p(-0.5, 1), p( 1, 0)],
        [p( 0.5, 0), p( 0.5, 0), p( 0, 0)],
        [p(-0.5, 1), p( 0.5, 0), p( 0, 0)],
    ]
```

The bending moments due to unit forces are used to compute the 3×3 flexibility matrix, `c` stands for curvatures and `bm` for bending moments.

```
F33 = array([[sum(i(ci, bmi) for ci, bmi in zip(c, bm)) for c in m] for bm in m])
```

Also needed are the 2×2 flexibility, the 3×3 stiffness matrix and the $2x2$ one (note that, having the flexibility, we used a procedure different from the static condensation); the 3×3 matrix is needed to compute the equivalent load on the dynamic degrees of freedom below.

```
F22 = F33[:-1,:-1]
K33 = inv(F33)
K22 = inv(F22)
```

```
dl(r'F33=\frac{L^3}{12EJ}'+pmat(F33*12)+',\qquad F22= \\frac{L^3}{12EJ}'+pmat(F22*1
2))
dl('K33=\\frac{3EJ}{13L^3}'+pmat(K33*13/3, fmt='%.3f')+
   ',\qquad K22=\\frac{4EJ}{5L^3}'+pmat(K22*5/4, fmt='%.3f'))
```

$$F33=\frac{L^3}{12EJ}\begin{bmatrix}12.000 & 3.000 & 4.000\\ 3.000 & 2.000 & 3.000\\ 4.000 & 3.000 & 8.000\end{bmatrix},$$

$$F22=\frac{L^3}{12EJ}\begin{bmatrix}12.000 & 3.000\\ 3.000 & 2.000\end{bmatrix}$$

$$K33=\frac{3EJ}{13L^3}\begin{bmatrix}7.000 & -12.000 & 1.000\\ -12.000 & 80.000 & -24.000\\ 1.000 & -24.000 & 15.000\end{bmatrix},$$

$$K22=\frac{4EJ}{5L^3}\begin{bmatrix}2.000 & -3.000\\ -3.000 & 12.000\end{bmatrix}$$

Eventually, the mass matrix (note that a vertical motion involves *both* the supported masses).

```
M22 = array(((2.0,0.0),(0.0,1.0)))
dl('M22 = m' + pmat(M22, fmt='%.1f'))
```

$$M22 = m \begin{bmatrix} 2.0 & 0.0 \\ 0.0 & 1.0 \end{bmatrix}$$

Let's compute the eigenvalues `evals`, the eigenvectors `evecs` (aliased to `Psi`) and the circular frequencies. While at it, we *need* the inverse of the starred mass matrix, don't we?

```
evals, evecs = eigh(K22, M22)
Psi = evecs
cfreqs = sqrt(evals)
Mstarinv = array(((1.0,0),(0,1.0)))
dl(r'\boldsymbol\Lambda^2=\omega_0^2'+pmat(diag(evals)))
dl(r'\boldsymbol\Lambda  =\omega_0'+pmat(diag(sqrt(evals))))
dl('\\boldsymbol\\Psi='+pmat(evecs))
```

$$\boldsymbol\Lambda^2=\omega_0^2\begin{bmatrix}0.484 & 0.000\\0.000 & 9.916\end{bmatrix}$$

$$\boldsymbol\Lambda=\omega_0\begin{bmatrix}0.696 & 0.000\\0.000 & 3.149\end{bmatrix}$$

$$\boldsymbol\Psi=\begin{bmatrix}-0.695 & -0.129\\-0.183 & 0.983\end{bmatrix}$$

Second Part, Free Vibrations

The system, after the load removal, is freely vibrating so that the modal responses are simply cosines, whose amplitude is given by the components of the q_0 vector.

The initial displacements can be computed from the static load vector, next we convert to modal coordinates and check our result

```
x0 = F22@array((1, 0)) # L**3/EJ * F * {1,0} EJ/300/L**2 = L/300 * F *{1,0}
dl('\\frac{300\\,x_0}L = '+pmat(x0[:,None]))
q0 = Mstarinv@evecs.T@M22@x0
dl('\\frac{300\\,q_0}L='+pmat(q0[:,None]))
dl(r'\Psi\, q_0 = \frac{L}{300}\,'+pmat((evecs@q0)[:,None], fmt="%g"))
```

$$\frac{300\,x_0}{L} = \begin{bmatrix}1.000\\0.250\end{bmatrix}$$

$$\frac{300\,q_0}{L} = \begin{bmatrix}-1.436\\-0.013\end{bmatrix}$$

$$\Psi\, q_0 = \frac{L}{300}\,\begin{bmatrix}1\\0.25\end{bmatrix}$$

Analytical expressions of modal responses

```
dl(r'\frac{300\,q_1(t)}L = %+f \cos %.3f\,\omega_0t'%(q0[0],cfreqs[0]))
dl(r'\frac{300\,q_2(t)}L = %+f \cos %.3f\,\omega_0t'%(q0[1],cfreqs[1]))
```

$$\frac{300\,q_1(t)}{L} = -1.436081 \cos 0.696\,\omega_0 t$$

$$\frac{300\,q_2(t)}{L} = -0.013051 \cos 3.149\,\omega_0 t$$

Analytical expressions of displacements

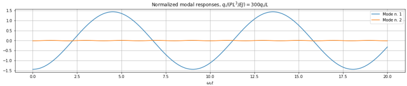
```
dl(r'\frac{300\,x_1(t)}L = %+f \cos %.3f\,\omega_0t %+f \cos%.3f\,\omega_0t'
   %(Psi[0,0]*q0[0], cfreqs[0], Psi[0,1]*q0[1], cfreqs[1]))
dl(r'\frac{300\,x_2(t)}L = %+f \cos %.3f\,\omega_0t %+f \cos%.3f\,\omega_0t'
   %(Psi[1,0]*q0[0], cfreqs[0], Psi[1,1]*q0[1], cfreqs[1]))
```

$$\frac{300\,x_1(t)}{L} = +0.998311 \cos 0.696\,\omega_0 t + 0.001689 \cos 3.149\,\omega_0 t$$

$$\frac{300\,x_2(t)}{L} = +0.262831 \cos 0.696\,\omega_0 t - 0.012831 \cos 3.149\,\omega_0 t$$

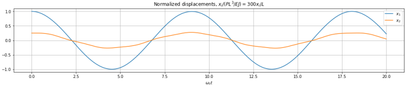
Plot of normalized modal responses

```
plt.rcParams["figure.figsize"] = (18,3)
lines = plt.plot(t, q0*cos(t[:,None]*cfreqs))
for n, line in enumerate(lines, 1): line.set_label('Mode n. %d'%n)
plt.legend(loc='best') ; plt.xlabel('$\\omega_0t$') ; plt.grid()
plt.title('Normalized modal responses, $q_i/(PL^3/EJ)=300q_i/L$');
```



Plot of normalized displacements

```
lines = plt.plot(t, (q0*cos(t[:,None]*cfreqs))@evecs.T)
for n, line in enumerate(lines): line.set_label('$x_%d$'%(n+1))
plt.legend(loc='best') ; plt.xlabel('$\\omega_0t$') ; plt.grid()
plt.title('Normalized displacements, $x_i/(PL^3/EJ)=300x_i/L$');
```



Third Part, Forced Response

The equivalent load

```
Kaa, Kab = K33[:2,:2], K33[:2,2:]
Kba, Kbb = K33[2:,:2], K33[2:,2:]

W = 1.0 # EJ/(400L)
p = -Kab@inv(Kbb)*W # Kab multiplies by L, inv(Kbb) divides by L**2,
                    # the result is the prod of adimensional matrix * W/L
p = p.flatten()     # p.flatten remedies an implementation detail
dl(r'p = \frac{EJ}{400\,L^2}'+pmat(p[:,None]))
pstar = evecs.T@p
dl(r'p^\star = \frac{EJ}{400\,L^2}'+pmat(pstar[:,None]))
```

$$p = \frac{EJ}{400\,L^2}\begin{bmatrix}-0.067\\1.600\end{bmatrix}$$

$$p^\star = \frac{EJ}{400\,L^2}\begin{bmatrix}-0.246\\1.582\end{bmatrix}$$

The frequency of the excitation is ω_0, we write $p_i^\star = \pi_i\,\frac{EJ}{400\,L^2}$

$$m\ddot{q}_i + m\omega_i^2 q_i = \frac{EJ}{400\,L^2}\pi_i \sin\omega_0 t = \frac{EJ}{400\,L^2}\frac{mL^3}{EJ}\omega_0^2\pi_i \sin\omega_0 t$$

Dividing by m and simplifying, the EOM can be written as

$$\ddot{q}_i + \omega_i^2 q_i = \frac{L}{400}\omega_0^2\pi_i \sin\omega_0 t = \delta\,\omega_0^2\pi_i \sin\omega_0 t$$

The particular integral is $\xi_i = C_i \sin\omega_0 t$, substituting in the EOM

$$C_i(\omega_i^2 - \omega_0^2) = \delta\,\omega_0^2\pi_i \quad \rightarrow \quad C_i = \frac{\omega_0^2}{\omega_i^2 - \omega_0^2}\delta\,\pi_i = \frac{\pi_i}{\lambda_i^2 - 1}\delta$$

```
C = pstar/(evals-1)
```

Analytical expressions of modal responses

$$q_i = \delta\, C_i\, (\sin\omega_0 t - \beta_i \sin\omega_i t)$$

```
dl(r'\frac{400\,q_1}L=%+f(\sin\omega_0t%+f\sin%.3f\,\omega_0t)'%(C[0],-1/cfreqs[0],
cfreqs[0]))
dl(r'\frac{400\,q_2}L=%+f(\sin\omega_0t%+f\sin%.3f\,\omega_0t)'%(C[1],-1/cfreqs[1],
cfreqs[1]))
```

$$\frac{400\,q_1}{L} = +0.477752(\sin\omega_0 t - 1.437296\sin 0.696\,\omega_0 t)$$

$$\frac{400\,q_2}{L} = +0.177391(\sin\omega_0 t - 0.317565\sin 3.149\,\omega_0 t)$$

Analytical expressions of displacements

$$x_i = \sum_j \psi_{ij} q_j$$

```
for i in (0, 1):
    dl(r'x_%d = %+f\sin\omega_0t %+f\sin%.3f\,\omega_0t %+f\sin%.3f\,\omega_0t'%(
        i+1, Psi[i,0]*C[0]+Psi[i,1]*C[1],
        -Psi[i,0]*C[0]/cfreqs[0], cfreqs[0],
        -Psi[i,1]*C[1]/cfreqs[1], cfreqs[1]))
```

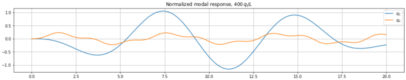
$$x_1 = -0.355072\sin\omega_0 t + 0.477348\sin 0.696\,\omega_0 t \\ + 0.007290\sin 3.149\,\omega_0 t$$

$$x_2 = +0.086957\sin\omega_0 t + 0.125674\sin 0.696\,\omega_0 t \\ - 0.055382\sin 3.149\,\omega_0 t$$

Plot of normalized modal responses

```
q1 = C[0]*(sin(t)-sin(cfreqs[0]*t)/cfreqs[0])
q2 = C[1]*(sin(t)-sin(cfreqs[1]*t)/cfreqs[1])

plt.plot(t, q1, label='$q_1$')
plt.plot(t, q2, label='$q_2$')
plt.title(r'Normalized modal response, $400\,q_i/L$')
plt.legend(loc='best') ; plt.grid() ;
```



Plot of normalized displacements

```
x1 = Psi[0,0]*q1 + Psi[0,1]*q2
x2 = Psi[1,0]*q1 + Psi[1,1]*q2

plt.plot(t, x1, label='$x_1$')
plt.plot(t, x2, label='$x_2$')
plt.title(r'Normalized displacements, $400\,x_i/L$')
plt.legend(loc='best') ; plt.grid() ;
```