where takes, an implemental state and majorital and activation and the controlling of the relation of the following the colority of the color		npute the 3x3 flexibility, next the 3x3 stiffness and the 2x2 stiffness plus the effects of the external load on librium related to the dynamic degrees of freedom.
The control of the		$\begin{array}{c c} & & & \\ \hline & &$
$\begin{aligned} & = \begin{cases} 1.3.3 \\ one of the set of the se$	ntegrals to have Ms = [[p(1.0, [p(0.5,	e the 3x3 flexibility, then it's just following the book 0), p(1.0, 0), p(0.0, 0), p(0.0, 0)], 0), p(0.5, 0), p(0.5, 0), p(0.5, 0)],
The proposal proposal $(a_1, a_2, a_3, a_4) = (a_1, a_2, a_4) = (a_1, a_4, a_4) = (a_1, a_4, a_4, a_4) = (a_1, a_4, a_4, a_4) = (a_1, a_4, a_4, a_4, a_4) = (a_1, a_4, a_4, a_4, a_4, a_4, a_4, a_4, a_4$	Ls = [1, 1, 1, F = np.array([[sum(i(*mcl) for mcl in zip(M, C, Ls)) for M in Ms] for C in Ms])
The control of the c	K33 = np.linal	lg.inv(F)
Consideration 1. The control of the control benefits	Ksd, Kss = K33 display(HTML() dl(r'\boldsymb K = np.linalg display(HTML() dl(r'\boldsymb	<pre>3[2:,:2], K33[2:,2:] '<h3>The 3x3 stiffness</h3>')) pol K_{(3,3)} = \frac{3}{14}\frac{EJ}{L^3}'+pmat(14*K33/3)) .inv(F[:2,:2]) '<h3>The 2x2 structural matrices</h3>')) pol K = \frac{EJ}{L^3}'+pmat(K))</pre>
The 2.02 structural matrices $N_{\rm min} = \frac{N_{\rm min}}{N_{\rm min}} \left[\frac{1.000 - 0.000}{0.000} \right]$ at $N_{\rm min} = \frac{1.000 - 0.000}{0.000} \left[\frac{1.000 - 0.000}{0.000} \right]$. The credit bank in some of the consent load $N_{\rm min} = \frac{1.000 - 0.000}{0.000} \left[\frac{1.000 - 0.000}{0.000} \right]$ (10.02). The credit bank in some of the consent load $N_{\rm min} = \frac{1.000}{0.000} \left[\frac{1.000}{0.000} \right]$ (10.02). The credit bank is some of the consent load of the object individual and its symmetry is structured as a structure of the consent load of the consen	dl(r'\boldsymb display(HTML(nodal_forces = dl(r'\boldsymb	<pre>cool M = m'+pmat(M)) '<h3>The nodal loads in terms of the external load</h3>')) = -Kds@np.linalg.inv(Kss) cool p(t) ='+pmat(nodal_forces)+'W(t)/L')</pre>
The model location in terms of the external local Size of Code (1972) The model location in terms of the external local Size of Code (1972) The model location in expension model (no model) is the related to the relation of the symmetric action or endough and the symmetric action of the symmetric action of the symmetric action or endough and the symmetric action of the s		
## 19	$oldsymbol{M} = m \; egin{bmatrix} 2.00 \ 0.00 \end{bmatrix}$	$egin{pmatrix} 0 & 0.000 \ 0 & 3.000 \end{bmatrix}$
where the state $S_{\rm const}(1)$ and $S_{\rm const}($	$oldsymbol{p}(t) = \left[egin{array}{c} -0.25 \ 1.500 \end{array} ight]$	$\left[egin{array}{c} 0 \ 0 \end{array} ight] W(t)/L$
The contract of Spreams ($(x_1, x_2, x_3, x_4) = (x_1, x_2, x_4) = (x_1, x_4) = (x$	<pre>dl(r'\omega^2 evecs[:,0] /= evecs[:,1] /= evecs *= 2</pre>	<pre>= \omega_0^2'+pmat(evals[:,None])) evecs[1,0] evecs[0,1]</pre>
where the model forces and the distinct are committed with respect to the model masses, as that we can amalify the committed or the model code and authorities. The committed of the model code and authorities of the code and authorities. The code is the code and authorities of the code and authorities. The code is the code and authorities of the code and authorities. The code is the code and authorities of the code and authorities. The code is the code and authorities of the code and authorities. The code and authorities of the code and authorities. The code and authorities of the co	Mstar = evecs dl(r'\boldsymb	.T@M@evecs pol M^\star = m' +pmat(Mstar))
processors of the remoted was of excellations and all processes of the remoted was of excellations and all processes of the remoted and processes. The remoted and processes of the remoted and processes of the remoted and processes of the remoted and processes. The remoted and processes of the remoted and	$oldsymbol{M}^\star = m \; egin{bmatrix} 30. \ -0 \end{bmatrix}$	$\begin{bmatrix} .000 & -0.000 \\ .000 & 20.000 \end{bmatrix}$
and some the modal equal density (~3) state $A_{ij} = A_{ij} + A_$	<pre>uppearance of t modal_forces = dl(r'\boldsymb pmat(modal_</pre>	he modaleq.s of equilibrium = np.linalg.inv(Mstar)@evecs.T@nodal_forces pol p^\star ='+pmat(modal_forces)+r'\frac{W(t)}{mL} =' + _forces) +
$\begin{aligned} & \text{Modal equation of motion} \\ & \hat{x}_1 + 0.000 x_1^2 y_1 & = 0.075 x_1^2 x_1^2 x_1^2 x_2^2 y_1^2 x_1^2 x_1^2$	and here the modisplay(HTML()	odal eq.s of equilibrium ' <h2>Modal equation of motion</h2> '))
Fig. 1.2.000 a_1^2 b_2^2 b_3^2	dl(r' \d (i,evals	uation of motion
Modal frequencies $a_i = 0.071(10a_0), \qquad a_0 = 1.73206 (a_0).$ The excitation is diministrated by the region $a_i = 2a_0$, so it is the particular integral, $\xi_i = C_i \sin 2a_0 \xi$ Substituting in the cools equations LHS we have $\xi_i(t) = \frac{p(f_i)}{(\lambda^2 - 2^2) - \frac{1}{n^2}}$ Essats/cFm($\xi_i(t) = 2a_0 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + $	${\ddot q}_{2} + 3.000\omega_0^2{ m g}$	$q_2 = -0.175 \Delta \omega_0^2 \sin(2\omega_0 t)$ rt(evals)
isolary (IIIII.C) detained in Image also (Abs.*) ($g_i(t) = \frac{p_i^{r_i(t)}}{(\lambda_i^t - \lambda^2) \cdot a_i^t}$ isolary (IIII.C) detained in Image also (Abs.*) 1 = notice, forces, flatter($f_i(t)$ (exclusive) or f_i is $(t) = (t-1)^r$ 1 = $(t-1)^r$	Modal freq	uencies
** most Joneses, flatten() (vow)s, 4) or jal (0, 2); $i = j_1$ is j_2 ; $i = j_3$; $i $	nodal equation	s LHS we have $\xi_i(t) = rac{p_i^\star(t)}{\left(\lambda_i^2 - 2^2 ight)\omega_0^2}$
$ \begin{aligned} & \frac{1}{2}((i) = +0.178000 \Delta \sin(2\omega_0)) \\ & the modal responses are given by the superposition of the particular integral and the homogeneous solution, where the material originator are chosen to fit the initial conditions, for a sinusoidal p.i. and rest initial conditions, it is well nown that q \cdot C_*(\sin \omega t - (\sin \omega_0 t) \sin \omega_0 t) \\ & = -2\epsilon_*(r + responses) \\ & = -2\epsilon_*(r + responses) \\ & = (r \cdot (-2M(t)) - \sin(3t \cdot responses) + (h2 \cdot r))) \\ & = -2\epsilon_*(r + responses) \\ & = (r \cdot (-2M(t)) - \sin(3t \cdot responses) \\ & = (r \cdot (-1/3) - (1/3)), r + responses) \\ & = (r \cdot (-1/3) - (1/3)), r + responses) \\ & = (r \cdot (-1/3) - (1/3)), r + responses) \\ & = (r \cdot (-1/3) - (1/3)), r + responses) \\ & = (r \cdot (-1/3) - (1/3)), r + responses) \\ & = (r \cdot (-1/3) - (1/3)), r + responses) \\ & = (r \cdot (-1/3) - (1/3)), r + responses) \\ & = (r \cdot (-1/3) - (1/3)), r + responses) \\ & = (r \cdot (-1/3) - (1/3)), r + responses) \\ & = (r \cdot (-1/3) - (-1/3)), r + responses) \\ & = (r \cdot (-1/3) - (-1/3)), r + responses) \\ & = (r \cdot (-1/3) - (-1/3)), r + responses) \\ & = (r \cdot (-1/3) - (-1/3)), r + responses) \\ & = (r \cdot (-1/3) - (-1/3)), r + responses) \\ & = (r \cdot (-1/3) - (-1/3)), r + responses) \\ & = (r \cdot (-1/3) - (-1/3)), r + responses) \\ & = (r \cdot (-1/3) - (-1/3)), r + responses) \\ & = (r \cdot (-1/3) - (-1/3)), r + responses) \\ & = (r \cdot (-1/3) - (-1/3)), r + responses) \\ & = (r \cdot (-1/3) - (-1/3)), r + responses) \\ & = (r \cdot (-1/3) - (-1/3)), r + responses) \\ & = (r \cdot (-1/3) - (-1/3)), r + responses) \\ & = (r \cdot (-1/3) - (-1/3)), r + responses) \\ & = (r \cdot (-1/3) - (-1/3)), r + responses) \\ & = (r \cdot (-1/3) - (-1/3)), r + responses) \\ & = (r \cdot (-1/3) - (-1/3)), r + responses) \\ & = (r \cdot (-1/3) - (-1/3)), r + responses) \\ & = (r \cdot (-1/3) - (-1/3)), r + responses) \\ & = (r \cdot (-1/3) - (-1/3)), r + responses) \\ & = (r \cdot (-1/3) - (-1/3)), r + responses) \\ & = (r \cdot (-1/3) - (-1/3)), r + responses) \\ & = (r \cdot (-1/3) - (-1/3)), r + responses) \\ & = (r \cdot (-1/3) - (-1/3)), r + responses) \\ & = (r \cdot (-1/3) - (-1/3), r + responses) \\ & = (r \cdot (-1/3) - (-1/3), r + responses) \\ & = $	c = modal_forc for j in (0, 1 i = j+1 dl(r'\xi_9	<pre>ces.flatten()/(evals-4) 1): %d(t) = %+.6f\Delta\sin(2\omega_0t)'%(i, c[j]))</pre>
onstants of integration are chosen to fit the initial conditions, for a sinusoidal p.l. and rest initial conditions, it is well movem that $g_i = C_i \left(sinu_i + (sinu_i) \omega/\omega \right) \right)$ $\frac{1}{15} s_i \log (r^i M_i) \cdot (r^i h_i) Modal. Responses c_i h_i > r^i \right) or j in (0, 1): 1 = j+1 d(c_i - 3k(t) - b) i_i (8^i + 6^i h_i) i_i (7^i h_i) Modal. Responses g_i (t) = (-0.02149 \sin(2\omega_i t) + 0.000600 \sin(0.707107 \omega_i t)) \Delta g_i (t) = (-0.02149 \sin(2\omega_i t) + 0.000600 \sin(0.707107 \omega_i t)) \Delta is now possible to plot the stuff that we have computed, for a time interval grater than what was requested. \frac{1}{2} c_i \left(\frac{1}{2} - \frac{1}{$	$\xi_2(t) = +0.175$ The modal respo	$\sin(2\omega_0 t)$ onses are given by the superposition of the particular integral and the homogeneous solution, where the
For j in (6, 1):	constants of inte cnown that display(HTML(a = -2*c/freqs	egration are chosen to fit the initial conditions. For a sinusoidal p.i. and rest initial conditions, it is well $q_i=C_i\big(sin\omega t-(\sin\omega_i t)\omega/\omega_i)\big)$ ' <n2>Modal Responses'))</n2>
$I_{R}(t) = \left(-0.021429 \sin(2\omega_0 t) + 0.000009 \sin(0.707107\omega_0 t) \right) \Delta$ $I_{R}(t) = \left(+0.175000 \sin(2\omega_0 t) - 0.202073 \sin(1.732051\omega_0 t) \right) \Delta$ is now possible to plot the stuff that we have computed, for a time interval grater than what was requested. **Gottlements of time vector* of = np.1inspace(6, 6.2831, 201) **Modal responses 2 = $(2 9)^n$ np. $\sin(2^n w t) + \sin(p n t) + \sin(p n t) + \cos(p n t)$ **PLot the modal responses 2 = $(2 1)^n$ np. $1_n(2^n w t) + \sin(p n t) + \cos(p n t)$ **PLot the modal responses 3 = $(2^n x + 1)^n \sin(p n t) + \cos(p n t)$ **It. $1_n x + 1 \cos(p n t) + \cos(p n t)$ **It. $1_n x + 1 \cos(p n t) + \cos(p n t)$ **It. $1_n x + 1 \cos(p n t) + \cos(p n t)$ **It. $1_n x + 1 \cos(p n t) + \cos(p n t)$ **It. $1_n x + 1 \cos(p n t)$ *	for j in (0, 1 i = j+1 dl(r'q_%d((i, c[l): (t) = \big(%+.6f\sin(2\omega_0t)%+.6f\sin(%.6f\omega_0t)\big)\Delta' % j], a[j], freqs[j]))
# Addinessional time vector ####################################	$q_1(t)=ig(-0.0$ $q_2(t)=ig(+0.1$	$egin{aligned} 21429\sin(2\omega_0 t) + 0.060609\sin(0.707107\omega_0 t)ig)\Delta \ 75000\sin(2\omega_0 t) - 0.202073\sin(1.732051\omega_0 t)ig)\Delta \end{aligned}$
Plot the modal responses plt.plot(w0t/np.pi,q1,label='\$q_1\$') hit.plot(w0t/np.pi,q1,label='\$q_2\$') hit.plot(w0t/np.pi,q1,label='\$q_2\$') hit.slabel(n'\$\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4\u00e4	# adimensional w0t = np.linsp # Modal respon q1 = c[0]*np.s	L time vector pace(0, 6.2831, 201) nses sin(2*w0t)+a[0]*np.sin(freqs[0]*w0t)
# Nodal responses il = qi*evecs[0,0]+q2*evecs[0,1] il = qi*evecs[0,0]+q2*evecs[1,1] il = plot the nodal responses lit.plot(wet/np.pi,xl,label='\$x_1\$') lit.lplot(wet/np.pi,xl,label='\$x_2\$') hit.legend(loc*best') hit.splad(loc*best') hit.grid() lit.grid() lit.grid() lit.show()	# Plot the model plt.plot(w0t/roplt.plot(w0t/roplt.legend(looplt.xlabel(r's	<pre>dal responses np.pi,q1,label='\$q_1\$') np.pi,q2,label='\$q_2\$') t='best') \$\omega_0t/\pi\$')</pre>
Dit.plot(w0t/np.pi,xl,label='\$x_1\$') Dit.plot(w0t/np.pi,x2,label='\$x_2\$') Dit.legend(loc='best') Dit.xlabel(r'\$\times_omega_0t/\pi\$') Dit.ylabel(r'\$x_i/\Delta\$') Dit.grid() Dit.grid() Dit.grid() Dit.grid() Dit.show() 02 01 02 000 025 050 075 100 125 150 175 200 04 04 000 025 050 075 100 125 150 175 200	olt.show() # Nodal respon x1 = q1*evecs x2 = q1*evecs # Plot the nod	[0,0]+q2*evecs[0,1] [1,0]+q2*evecs[1,1] dal responses
0.1 -0.1 -0.2 -0.0 -0.2 -0.0 -0.2 -0.4 -0.4 -0.4 -0.4 -0.4 -0.4 -0.4 -0.4	olt.plot(w0t/r olt.plot(w0t/r olt.legend(loc olt.xlabel(r'\$	np.pi,x1,label='\$x_1\$') np.pi,x2,label='\$x_2\$') c='best') \$\omega_0t/\pi\$')
0.4 0.2 0.0 0.0 0.25 0.50 0.75 1.00 1.25 1.50 1.75 2.00 0.4 0.2 0.0 0.0 0.25 0.50 0.75 1.00 1.25 1.50 1.75 2.00 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0	0.1	
0.2 -0.2 -0.4 -0.4 -0.0 0.00 0.25 0.50 0.75 100 125 150 175 200 ω ₀ t/π	-0.2	$\omega_0 t/\pi$
0.00 0.25 0.50 0.75 100 125 150 1.75 2.00 $\omega_0 t/\pi$	0.2 4 0.0	
	0.00 0.29	$\omega_0 t/\pi$

2 DoF System

couple that excites the system.

Let's start with a few imports

 ${\tt import\ matplotlib.pyplot\ as\ plt}$

 $\textbf{from IPython.display import} \ \ \mathsf{Latex}, \ \ \mathsf{HTML}$

 $\label{from scipy.linalg import eigh} \textbf{from scipy.linalg import eigh}$

 $\mbox{\em matplotlib}$ inline

import numpy as np

Here it is the problem statement in a single figure... beams are uniform, slender and massless, the dynamic degrees of freedom are the masses' vertical displacements but we need also the upper right rotation to take into account the external

Next, we define some helper functions, p deals with creating polynomial objects, i given two polys and a length computes the definite integral of the polys' product, pmat formats a 2D array object for MathJax diplay and finally dl is an helper to