

Rayleigh Quotient

Giacomo Boffi

1 Rayleigh Quotient

1.1 Preliminars are important

We are going to do some symbolic computations, to do so we have to

1. load a number of names from the `sympy` module, a library dealing with creation and manipulation of symbolic expressions (and initialize its printing system),
2. associate symbols with names.

Having defined names for the symbols, we can start to write symbolic expressions, and give names to these expression too.

```
from sympy import init_printing, integrate, latex, plot, solve, symbols
init_printing(use_latex=1)

EJ, H, m, Z = symbols('EJ_H_m_zeta')
s, w, z, xi = symbols('s_omega_z_xi')
Z0 = Z*H # $Z_0 = \zeta \times H$
w2_0 = EJ/m/H**4 # reference frequency
```

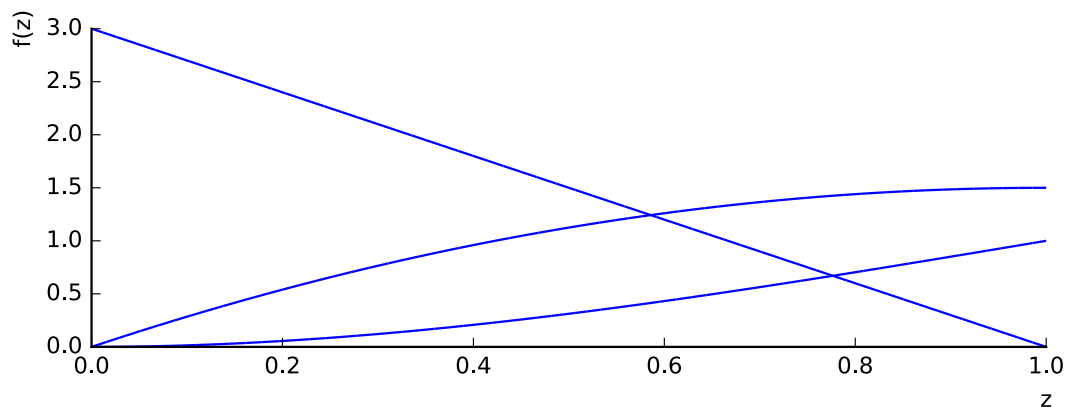
We will repeatedly compute ω^2 by equating $V_{\max} = T_{\max}$, hence we define a function to avoid repeating ourselves, we will repeatedly generate and display formatted representations of mathematical expressions, hence we define a function to do so.

```
def omega_2(T, V):
    roots = solve(T-V, w)
    omega = roots[-1].expand()
    return omega**2
from IPython.display import display, Latex
def displatex(*strings):
    display(Latex('$$' + '_' .join(strings) + '$$'))
```

1.2 Choosing a good shape function

A good shape function can be obtained starting with a point load applied to the top, to represent the inertial force of the supported mass, much larger than the distributed inertial forces, computing the bending moment and finally have, by double integration, adimensionalization and normalization, a shape function that respects not only ALL the boundary conditions but also the equation of indefinite equilibrium, if we neglect the small (relative to mass'inertial force) distributed inertial forces.

```
phi2 = (H-z)*3/H**3
phi1 = (6*H-3*z)*z/2/H**3
phi0 = (3*H-z)*z*z/2/H**3
plot(phi0.subs(H,1), phi1.subs(H,1), phi2.subs(H,1), (z, 0, 1));
```



1.2.1 Choosing a shape function, this time for real...

But, as requested by the text of the problem, we choose a different, simpler shape function

$$\varphi_0, \varphi_1, \varphi_2 = (z/H)^2, 2z/H, 2/H$$

With this shape function we have

$$v(z,t) = Z_0 \phi(z) \sin \omega t = Z_0 \frac{z^2}{H^2} \sin \omega t, \quad (1)$$

$$\dot{v}_{\max}(z) = \omega Z_0 \phi(z) = \omega Z_0 \frac{z^2}{H^2}, \quad (2)$$

$$v''_{\max}(z) = Z_0 \phi''(z) = Z_0 \frac{2}{H^2}. \quad (3)$$

2 Computing R_{00}

2.1 Maximum kinetic energy

Let's write an expression to represent the maximum velocity and immediately compute the maximum value of the kinetic energy — note that there is a (LARGE) contribution from the supported mass...

```
dot_v0 = Z0 * phi0 * w
T = (m*dot_v0**2).integrate((z, 0, H)) / 2
T = T + 3*m*H*(dot_v0.subs(z,H))**2 / 2
displatex('\dot{v}_{\max} =', latex(dot_v0), '\\cos\\omega t')
displatex('T_{\max} =',
           latex((m*dot_v0**2).integrate((z, 0, H))/2), '+',
           latex(3*m*H*(dot_v0.subs(z,H))**2/2), '=', latex(T))
```

$$\dot{v} = \frac{\omega \zeta}{H} z^2 \cos \omega t$$

$$T_{\max} = \frac{H^3 m}{10} \omega^2 \zeta^2 + \frac{3m}{2} H^3 \omega^2 \zeta^2 = \frac{8m}{5} H^3 \omega^2 \zeta^2$$

2.2 Maximum strain energy

Let's write an expression to represent the maximum curvature and the the maximum strain energy

```
v2 = Z0 * phi2
V = (EJ*v2**2).integrate((z, 0, H))/2
displatex("v''_{\max} =", latex(v2))
displatex("V_{\max} =", latex(V))
```

$$v''_{\max} = \frac{2\zeta}{H}$$

$$V_{\max} = \frac{2EJ}{H} \zeta^2$$

2.3 R_{00}

Finally, from equating the maximum energies and solving for ω^2 we have the Rayleigh quotient approximation,

```
R00 = omega_2(T, V)
displatex('R_{00} =', latex(R00), '=', latex(R00/w2_0), '\\,\\omega_0^2 =',
           latex((R00/w2_0).evalf(n=6)), '\\,\\omega_0^2')
```

$$R_{00} = \frac{5EJ}{4H^4 m} = \frac{5}{4} \omega_0^2 = 1.25 \omega_0^2$$

3 Computing R_{01} and R_{11}

To compute the first refinement we need

1. the inertial forces, $f_i(z,t)$ and
2. the displacements produced by the inertial forces, $v_i(z,t)$.

3.1 The inertial forces

The *distributed* inertial forces are obtained from the accelerations,

```

ddot_v = w*w*Z0*phi0
fI = m*ddot_v

displatex('\ddot{v}(z,t)=-', latex(ddot_v), r'\sin\omega t')
displatex(r'f_{\text{I}}(z,t)=-m\ddot{v}_m\sin\omega t', latex(ddot_v), '\sin\omega t')

```

$$\ddot{v}(z,t) = -\frac{\omega^2 \zeta}{H} z^2 \sin \omega t$$

$$f_I(z,t) = -m\ddot{v} = m \frac{\omega^2 \zeta}{H} z^2 \sin \omega t$$

and there is also an inertial force associated with the supported mass.

3.1.1 Internal forces, shear V and bending moment M

We want the internal forces, the shear $V(z,t)$ and the bending moment $M(z,t)$; the shear, writing the horizontal equilibrium for the part of the tower above z is

$$V(z,t) = \int_z^H f_I(s,t) ds - (3\bar{m}H)\ddot{v}(H,t)$$

while we have. from $V = M'$ and $M(H,t) = 0$,

$$M(z,t) = \int_z^H V(s,t) ds,$$

```

Vz = (fI.subs(z, s)).integrate((s, z, H))
Vz = Vz + (3*m*H)*ddot_v.subs(z,H)
Mz = (Vz.subs(z, s)).integrate((s, z, H))
displatex('V(z,t)=(', latex(Vz), ')\sin\omega t')
displatex('M(z,t)=(', latex(Mz), ')\sin\omega t')

```

$$V(z,t) = \left(\frac{10m}{3} H^2 \omega^2 \zeta - \frac{m\omega^2 z^3 \zeta}{3H} \right) \sin \omega t,$$

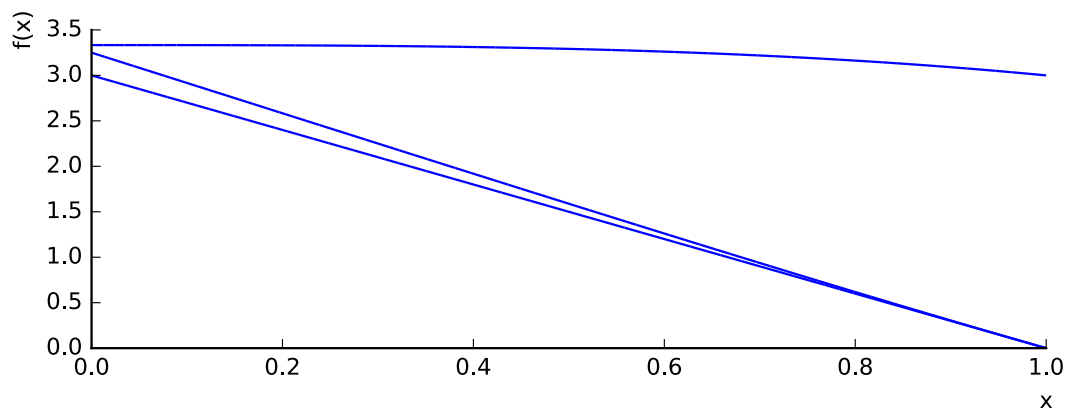
$$M(z,t) = \left(\frac{13m}{4} H^3 \omega^2 \zeta - \frac{10m}{3} H^2 \omega^2 z \zeta + \frac{m\omega^2 z^4 \zeta}{12H} \right) \sin \omega t.$$

The maximum values of $V(z)$ and $M(z)$ can be plotted to see how much the shear is close to a constant value and the moment to a linear dependency, due to the prevalence of the inertial force of the supported mass.

```

for_plot = {m:1, Z0:1, w:1, EJ:1, H:1, Z:1}
plot((Vz.subs(for_plot), (z, 0, 1)),
      (Mz.subs(for_plot), (z, 0, 1)),
      (3-3*z, (z, 0, 1)));

```



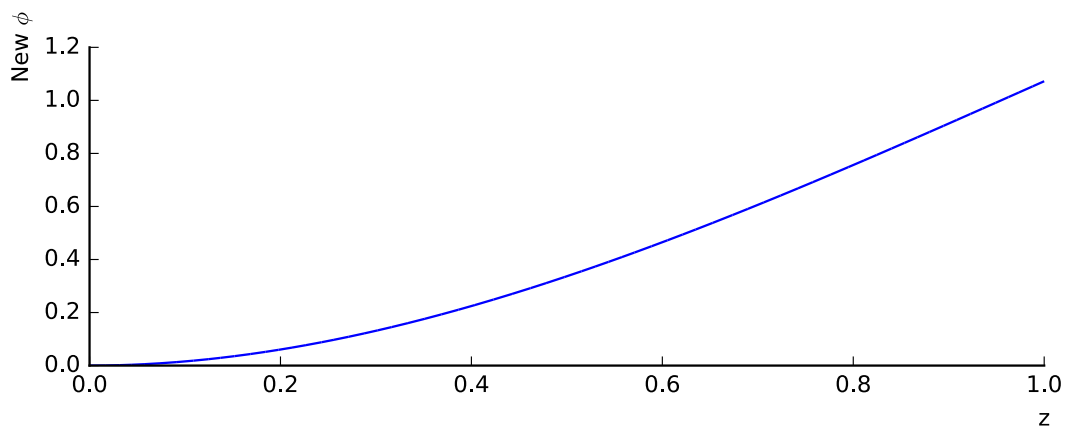
3.2 Displacements due to inertial forces

Starting from $v''(z,t) = M/EJ$, taking into account the boundary conditions at $z=0$ we have

```
v2 = Mz/EJ
v1 = integrate(v2, z)
v0 = integrate(v1, z)
displatex('v(z,t)=\\left(', latex(v0), '\\right)\\sin\\omega_t')
displatex('v(H,t)=\\_', latex(v0.subs(for_plot).subs(z,1)), '\\_,Z_0\\_,\\sin\\omega_t')
plot(v0.subs(for_plot), (z, 0, 1), ylabel='New_\\phi$');
```

$$v(z,t) = \left(\frac{13H^3 m \zeta}{8EJ} \omega^2 z^2 - \frac{5H^2 m \zeta}{9EJ} \omega^2 z^3 + \frac{m \omega^2 z^6 \zeta}{360EJH} \right) \sin \omega t$$

$$v(H,t) = \frac{193}{180} Z_0 \sin \omega t$$



3.3 New maximum strain energy

Our new estimate of the maximum strain energy is

$$V_{\max} = \frac{1}{2} \int_0^H f_{i,\max}(z) v_{\max}(z) dz + \frac{1}{2} (3\bar{m}H) \ddot{v}_{\max}(H) \times v_{\max}(H)$$

```
V = (v0*fI).integrate((z, 0, H))/2
V = V + v0.subs(z,H)*3*m*H*ddot_v.subs(z,H)/2
displatex('V_\\text{max}_=', latex(V))
```

$$V_{\max} = \frac{1397H^7 m^2 \omega^4}{810EJ} \zeta^2$$

3.4 R_{01}

Equating the new maximum strain energy to the maximum kinetic energy of the previous estimate we have

```
R01 = omega_2(T, V)
displatex('R_{01}_=', latex(R01/w2_0), '\\_,\\omega_0^2',
          '=', latex(R01.evalf(n=6)/w2_0), '\\_,\\omega_0^2')
```

$$R_{01} = \frac{1296}{1397} \omega_0^2 = 0.927702 \omega_0^2$$

3.5 New maximum kinetic energy

At this point, we can compute a better estimate of the maximum kinetic energy,

$$\dot{v}_{\max}(z) = \omega v_{\max}(z),$$

and T_{\max} can be computed exactly as before, using the new \dot{v} .

```
dot_v0 = w*v0
T = (m*dot_v0**2).integrate((z, 0, H))/2
T = T + 3*m*H*(dot_v0.subs(z,H))**2/2
displatex('T_\\text{max}_=', latex(T))
```

$$T_{\max} = \frac{109709H^{11}m^3\omega^6}{58968EJ^2}\zeta^2$$

3.6 R_{11}

Equating T_{\max} to V_{\max} eventually we have

```
R11 = omega_2(T, V)
displatex('R_{11}_=', latex(R11/w2_0), '\\,\\omega_0^2',
          '=', latex(R11.evalf(n=6)/w2_0), '\\,\\omega_0^2')
```

$$R_{11} = \frac{508508}{548545}\omega_0^2 = 0.927012\omega_0^2$$