Continuum mechanics

Lecture 10

Tomáš Profant

Brno University of Technology
Faculty of Mechanical Engineering

Direct Variational Methods

In the principle of virtual displacements, the Euler equations are the equilibrium equations, whereas in the principle of virtual forces, they are the compatibility equations. The Euler equations are in the form of differential equations that are not always tractable by exact methods of solution. The direct variational methods usually bypass the derivation of the Euler equations and go directly from a variational statement of the problem to the solution of the Euler equations. One of these methods is *the Ritz method* and *the Galerkin method*.

Ritz Method

In the Ritz method, the displacements u_i for $i = 1, 2, 3$ are approximated by a finite combination of the form

$$u_i(x_k) \approx U_i(x_k) = \sum_{j=1}^{n} c_j^i \phi_j^i(x_k) + \phi_0^i(x_k) \quad (i, k = 1, 2, 3),$$

where the parameters c_j^i must be determined by requiring that the principle of virtual displacements hold for arbitrary variations of these parameters.

Ritz Method

The functions $\phi_0^i(x_k)$ and $\phi_j^i(x_k)$ must satisfy the following requirements:

1. $\phi_0^i(x_k)$ should satisfy the specified essential boundary conditions associated with $u_i(x_k)$,
2. $\phi_j^i(x_k)$ for $j = 1, 2, \ldots, n$ must be continuous as required by the variational principle being used,
3. $\phi_j^i(x_k)$ for $j = 1, 2, \ldots, n$ satisfy the homogeneous form of the specified essential boundary conditions,
4. $\phi_m^i(x_k)$ and $\phi_n^i(x_k)$ must be linearly independent for $m \neq n$ and $m, n = 1, 2, \ldots,$
5. $\phi_j^i(x_k)$ must be complete for $j = 1, 2, \ldots$.

Ritz Method

The functions $\phi_j^i(x_k)$ for $j = 0, 1, \ldots$ form an infinite basis of the particular function space (a vector space in which vectors are replaced by functions) on some finite domain $x_k \in \Omega$. The above mentioned combination of the requirements that these functions $\phi_j^i(x_k)$ must be as complete and sufficiently smooth as the variational principle requires is generally very problematic from the mathematical point of view. It is a great achievement of modern mathematics that functional spaces satisfying this requirements was found for the wide range of partial differential equations used in mathematical physics.

Ritz Method

Since the natural boundary conditions of the problem are included in the variational statement, we require the assumed displacements $U_i(x_k)$ to satisfy only the essential boundary conditions. For convenience, we select $\phi_j^i(x_k)$ for $j = 1, 2, \ldots$ to satisfy the homogeneous form and $\phi_0^i(x_k)$ to satisfy the actual form of the essential boundary conditions. The general rule is that the coordinate functions $\phi_j^i(x_k)$ should be selected from an admissible set, from the lowest order to a desirable order, without missing any intermediate terms. The j-order components of the polynomials $\phi_j^i(x_k) = x^j$ can be chosen in the case of the one-dimensional problems of the elasticity.

Ritz Method

The equation

$$U_i(x_k) = \sum_{j=1}^{n} c_j^i \phi_j^i(x_k) + \phi_0^i(x_k) \quad (i, k = 1, 2, 3)$$

is used to compute approximate strains e_{ij} which are then used to compute the virtual work of the system in equilibrium given by the expression

$$\delta\Pi = \delta(V + U) = 0.$$

We obtain

$$\delta\Pi(u_1, u_2, u_3) = \sum_{j=1}^{n} \left(\frac{\partial\Pi}{\partial c_j^1} \delta c_j^1 + \frac{\partial\Pi}{\partial c_j^2} \delta c_j^2 + \frac{\partial\Pi}{\partial c_j^3} \delta c_j^3 \right) = 0,$$

where the variation $\delta u_i(x_k)$ were replaced by

$$\delta u_i(x_k) = \sum_{j=1}^{n} \delta c_j^i \phi_j^i(x_k) \quad (i, k = 1, 2, 3).$$

Ritz Method

Since δc_j^i for $j = 1, 2, \ldots, n$ are arbitrary and independent, it follows that

$$\frac{\partial \Pi}{\partial c_j^i} = 0, \quad (i = 1, 2, 3,\ j = 1, 2, \ldots, n).$$

Thus, we obtain $3n$ linearly independent simultaneous equations for the $3n$ unknowns. Once c_j^i are determined from the equations above, the approximate displacements of the problem are given by

$$U_i(x_k) = \sum_{j=1}^{n} c_j^i \phi_j^i(x_k) + \phi_0^i(x_k) \quad (i, k = 1, 2, 3).$$

These displacements can be used to evaluate strains and stresses.

Ritz Method

Some general features of the Ritz approximations based on the principle of virtual displacements are as follows

1. for increasing values of n, the previously computed coefficients of the above mentioned algebraic equations remain unchanged (provided the previously selected coordinate functions are not changed), and one must add newly computed coefficients to the system of equations,
2. if $\delta\Pi$ is nonlinear in u_i, the resulting algebraic equations will also be nonlinear in the parameters c_j^i. To solve such nonlinear equations, a variety of numerical methods are available, such as the Newton-Raphson method, generally, there is more than one solution to the equations,

Ritz Method

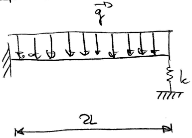
3. since the strains are computed from the approximate displacements, the strains and stresses are generally less accurate than the displacements,
4. the equilibrium equations of the problem are satisfied only in the energy sense $\delta\Pi = 0$, not in the differential equation sense, therefore the displacements obtained from the approximation generally do not satisfy the equations of equilibrium point-wise,
5. since a continuous system is approximate by a finite number coordinates (or degree of freedom), the approximate system is less flexible than the actual system, consequently, the displacements obtained by the Ritz method converge to the exact displacements from bellow, the displacements obtained from the Ritz approximations based on the complementary energy principle provide an upper bound for the exact solution.

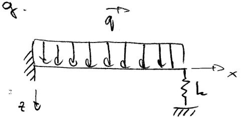
Ritz Method

The Ritz method is not applicable for the problems with complex geometry, external loadings and boundary conditions because it is not possible to establish the adequate function space, in which the solution should be found. On the other hand, the decomposition of the problem to be solved into the set of the simplified ones (elements) with respect to the geometry, loadings and boundary conditions allows the Ritz method to be used very effectively. This way of applying the Ritz method is known as *the finite element method*.

Direct Variational Methods: Examples

Example 11: Consider a uniform cross-section beam in the figure. We wish to find the transverse deflection of the beam under uniformly distributed transverse loading q.





The principle of virtual displacements for the problem becomes

$$0 = \delta\Pi = \delta(U+V) = \delta\left(\frac{1}{2}\int_0^{2L} EI\left(\frac{d^2w}{dx^2}\right)^2 dx - \int_0^{2L} f_0 w\,dx - \frac{1}{2}kw^2\Big|_{x=2L}\right)$$

$$\Rightarrow \quad 0 = \int_0^{2L}\left(EI\,\frac{d^2w}{dx^2}\,\frac{d^2\delta w}{dx^2} - f_0\,\delta w\right)dx - (k\,w\,\delta w)\Big|_{x=2L}$$

where $w(0)=0$, $\frac{dw}{dx}(0)=0$ are essential boundary conditions and $f_0=q$ is a volume external force. We choose a four-parameter approximation.

$$w = c_1\phi_1 + c_2\phi_2 + c_3\phi_3 + c_4\phi_4 + \phi_0 \Rightarrow \delta w = \delta c_1\phi_1 + \delta c_2\phi_2 + \delta c_3\phi_3 + \delta c_4\phi_4$$

where $\phi_i = x^i$ for $i=2,3,4$, $\phi_0=0$ due to the essential condition $w(0)=0$ and $\phi_1=0$ due to the essential condition $\frac{dw(0)}{dx}=0$

$$\Rightarrow \quad 0 = \int_0^{2L}\Big[EI(c_2\phi_2'' + c_3\phi_3'' + c_4\phi_4'')(\delta c_2\phi_2'' + \delta c_3\phi_3'' + \delta c_4\phi_4'') -$$

$$- q(\delta c_2\phi_2 + \delta c_3\phi_3 + \delta c_4\phi_4)\Big]dx - k(c_2\phi_2 + c_3\phi_3 + c_4\phi_4)\Big|_{x=2L}(\delta c_2\phi_2 + \delta c_3\phi_3 + \delta c_4\phi_4)\Big|_{x=2L}$$

Direct Variational Methods: Examples

$$b_{33} = \int_0^{2L} EI\left(\frac{d^2}{dx^2}\phi_3\right)^2 dx - k(\phi_3)^2\Big|_{x=2L} = EI\int_0^{2L}\left(\frac{d^2}{dx^2}x^3\right)^2 dx - k(x^3)^2\Big|_{x=2L} =$$

$$= 36EI\left[\frac{x^3}{3}\right]_0^{2L} - k(2L)^6 = 36EI\,\frac{8L^3}{3} - k\cdot 64L^6 = 96EIL^3 - 64kL^6$$

$$b_{34} = \int_0^{2L} EI\,\frac{d^2}{dx^2}\phi_3\,\frac{d^2}{dx^2}\phi_4\,dx - k(\phi_3\phi_4)\Big|_{x=2L} = EI\int_0^{2L}\frac{d^2}{dx^2}x^3\,\frac{d^2}{dx^2}x^4\,dx - k(x^3x^4)\Big|_{x=2L} =$$

$$= 72EI\left[\frac{x^4}{4}\right]_0^{2L} - k(2L)^7 = 72EI\cdot 4L^4 - k\cdot 128L^7 = 288EIL^4 - 128kL^7$$

$$b_{42} = \int_0^{2L} EI\,\frac{d^2}{dx^2}\phi_4\,\frac{d^2}{dx^2}\phi_2\,dx - k(\phi_4\phi_2)\Big|_{x=2L} = b_{24}$$

$$b_{43} = \int_0^{2L} EI\,\frac{d^2}{dx^2}\phi_4\,\frac{d^2}{dx^2}\phi_3\,dx - k(\phi_4\phi_3)\Big|_{x=2L} = b_{34}$$

$$b_{44} = \int_0^{2L} EI\left(\frac{d^2}{dx^2}\phi_4\right)^2 dx - k(\phi_4)^2\Big|_{x=2L} = EI\int_0^{2L}\left(\frac{d^2}{dx^2}x^4\right)^2 dx - k(x^4)^2\Big|_{x=2L} =$$

$$= 144EI\left[\frac{x^5}{5}\right]_0^{2L} - k(2L)^8 = \frac{1}{5}4608EIL^5 - k\,256L^8$$

$$f_2 = \int_0^{2L} q\cdot\phi_2\,dx = q\int_0^{2L} x^2\,dx = q\left[\frac{x^3}{3}\right]_0^{2L} = q\,\frac{8}{3}L^3$$

$$f_3 = \int_0^{2L} q\,\phi_3\,dx = q\int_0^{2L} x^3\,dx = q\left[\frac{x^4}{4}\right]_0^{2L} = q\,4L^4$$

Direct Variational Methods: Examples

$$\Rightarrow \quad 0 = \sum_{i=2}^{4}\left(\sum_{j=2}^{4} b_{ij}c_j - f_i\right)\delta c_i \quad \Rightarrow$$

$$\Rightarrow \quad 0 = \sum_{j=2}^{4} b_{ij}c_j - f_i \quad \text{for } i = 2,3,4 \text{, where}$$

$$b_{ij} = \int_0^{2L}\left(EI\,\frac{d^2\phi_i}{dx^2}\cdot\frac{d^2\phi_j}{dx^2}\right)dx - (k\,\phi_i\phi_j)\Big|_{x=2L}, \quad f_i = \int_0^{2L} q\phi_i\,dx$$

$$b_{22} = \int_0^{2L}\left(EI\cdot\frac{d^2}{dx^2}\phi_2\cdot\frac{d^2}{dx^2}\phi_2\right)dx - k(\phi_2)^2\Big|_{x=2L} = EI\int_0^{2L}\left(\frac{d^2}{dx^2}\cdot x^2\right)^2 dx - k(2L)^4 =$$

$$= 4EI\cdot 2L - k16L^4 = 8EIL - 16kL^4$$

$$b_{23} = \int_0^{2L}\left(EI\,\frac{d^2}{dx^2}\phi_2\,\frac{d^2}{dx^2}\phi_3\right)dx - k(\phi_2\phi_3)\Big|_{x=2L} = EI\int_0^{2L}\frac{d^2}{dx^2}x^2\,\frac{d^2}{dx^2}x^3\,dx - k(x^2x^3)\Big|_{x=2L} =$$

$$= 12EI\left[\frac{x^2}{2}\right]_0^{2L} - k(2L)^5 = 12EI\,\frac{4L^2}{2} - 32L^5 = 24EIL^2 - 32kL^5$$

$$b_{24} = \int_0^{2L}\left(EI\,\frac{d^2}{dx^2}\phi_2\,\frac{d^2}{dx^2}\phi_4\right)dx - k(\phi_2\phi_4)\Big|_{x=2L} = EI\int_0^{2L}\frac{d^2}{dx^2}x^2\,\frac{d^2}{dx^2}x^4\,dx - k(x^2x^4)\Big|_{x=2L} =$$

$$= 24EI\left[\frac{x^3}{3}\right]_0^{2L} - k(2L)^6 = 24EI\,\frac{8L^3}{3} - 64kL^6 = 64EIL^3 - 64kL^6$$

$$b_{32} = \int_0^{2L} EI\,\frac{d^2}{dx^2}\phi_3\,\frac{d^2}{dx^2}\phi_2\,dx - k(\phi_3\phi_2)\Big|_{x=2L} = 24EIL^2 - 32kL^5 = b_{23}$$

Direct Variational Methods: Examples

$$f_4 = \int_0^{2L} q \cdot \phi_4 \, dx = q \int_0^{2L} x^4 \, dx = q \left[\frac{x^5}{5} \right]_0^{2L} = q \cdot \frac{32}{5} L^5$$

We have

$$\begin{bmatrix} 8EIL - 16kL^4 & 24EIL^2 - 32kL^5 & 64EIL^3 - 64kL^6 \\ 0 & 96EIL^3 - 64kL^6 & 288EIL^4 - 128kL^7 \\ 0 & 0 & \frac{1}{5}\left(4608\, EIL^5 - k256L^8\right) \end{bmatrix} \begin{bmatrix} c_2 \\ c_3 \\ c_4 \end{bmatrix} = 4qL^3 \begin{bmatrix} \frac{2}{3} \\ L \\ \frac{8}{5}L^2 \end{bmatrix}$$

and the solution of this algebraic system of equations is

$$c_2 = \frac{L^2 q\,(3EI + 2L^3 k)}{EI\,(3EI + 8L^3 k)}, \quad c_3 = \frac{-2q\,(3EI + 5L^3 k)}{3EI\,(3EI + 8L^3 k)}, \quad c_4 = \frac{q}{24EI}$$

and after its substitution into w we can get

$$w(2L) = \frac{6L^4 q}{3EI + 8L^3 k}$$

Direct Variational Methods: Examples

```
#-----[            ]-----
#-----[ example 11 ]-----
#-----[            ]-----

#-----[ derivation of basis of function space ]-----

#-----[ import of sympy, numpy and matplotlib libraries ]--

import sympy as sp
import numpy as np
import matplotlib.pyplot as plt

#-----[ initiation of quality printing ]-----
sp.init_printing()
```

Direct Variational Methods: Examples

```
#-----[ definition of used symbols ]-----

c0,c1,c2,c3,c4,c5=sp.symbols('c0 c1 c2 c3 c4 c5')
x=sp.symbols('x')
a=sp.symbols('a')
```

Direct Variational Methods: Examples

```
#-----[ general form of basis functions ]-----

f=c0+c1*x+c2*x**2+c3*x**3+c4*x**4+c5*x**5

#-----[ essential boundary conditions ]-----
#-----[       of solved problem        ]-----

eqn1=f.subs(x,0)
eqn2=(f.diff(x)).subs(x,0)

#-----[ solution of essential boundary ]-----
#-----[      conditions for c0,c1      ]-----

sol=sp.solve([eqn1,eqn2],[c0,c1])
```

Direct Variational Methods: Examples

```
#-----[ choosen basis of functional space ]-----

phi2=f.subs({c0:sol[c0],c1:sol[c1], \
             c2:1,c3:0,c4:0,c5:0})
phi3=f.subs({c0:sol[c0],c1:sol[c1], \
             c2:0,c3:1,c4:0,c5:0})
phi4=f.subs({c0:sol[c0],c1:sol[c1], \
             c2:0,c3:0,c4:1,c5:0})
phi5=f.subs({c0:sol[c0],c1:sol[c1], \
             c2:0,c3:0,c4:0,c5:1})
```

Direct Variational Methods: Examples

```
print("\nphi2=")
sp.pprint(phi2)
print("\nphi3=")
sp.pprint(phi3)
print("\nphi4=")
sp.pprint(phi4)
print("\nphi5=")
sp.pprint(phi5)

#-----[ plot of first four basis functions ]-----

x_plot=np.linspace(0,3,100)
y2_plot=[]
y3_plot=[]
y4_plot=[]
y5_plot=[]
```

Direct Variational Methods: Examples

```
for ii in x_plot:
  y2_plot.append(phi2.subs({a:1,x:ii}))
  y3_plot.append(phi3.subs({a:1,x:ii}))
  y4_plot.append(phi4.subs({a:1,x:ii}))
  y5_plot.append(phi5.subs({a:1,x:ii}))
```

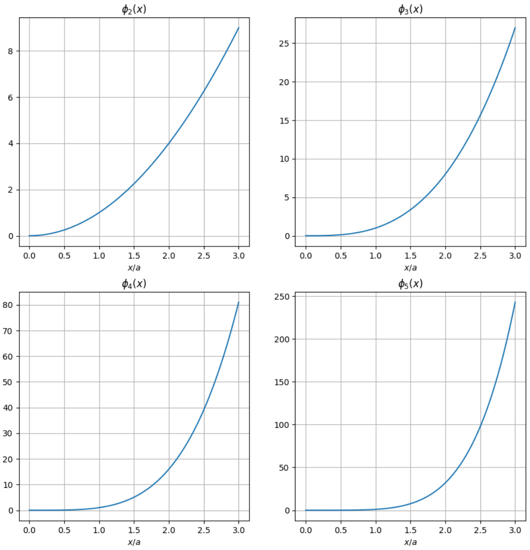
Direct Variational Methods: Examples

```
fig,axs=plt.subplots(2,2,figsize=(11,11))
axs[0,0].plot(x_plot,y2_plot)
axs[0,0].set_title(r'$\phi_2(x)$')
axs[0,0].set_xlabel(r'$x/a$')
axs[0,0].grid(True)
axs[0,1].plot(x_plot,y3_plot)
axs[0,1].set_title(r'$\phi_3(x)$')
axs[0,1].set_xlabel(r'$x/a$')
axs[0,1].grid(True)
```

Direct Variational Methods: Examples

```
axs[1,0].plot(x_plot,y4_plot)
axs[1,0].set_title(r'$\phi_4(x)$')
axs[1,0].set_xlabel(r'$x/a$')
axs[1,0].grid(True)
axs[1,1].plot(x_plot,y5_plot)
axs[1,1].set_title(r'$\phi_5(x)$')
axs[1,1].set_xlabel(r'$x/a$')
axs[1,1].grid(True)
plt.savefig('plot_example11-1')
```

Direct Variational Methods: Examples



Direct Variational Methods: Examples

```
#-----[             ]-----
#-----[ example 11 ]-----
#-----[             ]-----

#-----[ solution found using Ritz method ]-----

#-----[  import of sympy, numpy  ]-----
#-----[ and matplotlib libraries ]-----

import sympy as sp
import numpy as np
import matplotlib.pyplot as plt

#-----[ initiation of quality printing ]-----

sp.init_printing()
```

Direct Variational Methods: Examples

```
#-----[ symbol definition ]-----

q,L,k=sp.symbols('q L k')
E,I=sp.symbols('E I')
x=sp.symbols('x')
c2,c3,c4=sp.symbols('c2 c3 c4')

#-----[ basis of function space of solution ]-----

phi2=x**2
phi3=x**3
phi4=x**4
```

Direct Variational Methods: Examples

```
#-----[    system of algebraic equations    ]-----
#-----[ resulting from Hamilton's principle ]-----

b22=sp.integrate(E*I*phi2.diff(x,2)*phi2.diff(x,2), \
                 (x,0,2*L)) \
    +k*phi2.subs(x,2*L)*phi2.subs(x,2*L)
b23=sp.integrate(E*I*phi2.diff(x,2)*phi3.diff(x,2), \
                 (x,0,2*L)) \
    +k*phi2.subs(x,2*L)*phi3.subs(x,2*L)
b24=sp.integrate(E*I*phi2.diff(x,2)*phi4.diff(x,2), \
                 (x,0,2*L)) \
    +k*phi2.subs(x,2*L)*phi4.subs(x,2*L)
```

Direct Variational Methods: Examples

```
b32=b23
b33=sp.integrate(E*I*phi3.diff(x,2)*phi3.diff(x,2), \
                 (x,0,2*L)) \
    +k*phi3.subs(x,2*L)*phi3.subs(x,2*L)
b34=sp.integrate(E*I*phi3.diff(x,2)*phi4.diff(x,2), \
                 (x,0,2*L)) \
    +k*phi3.subs(x,2*L)*phi4.subs(x,2*L)
b42=b24
b43=b34
b44=sp.integrate(E*I*phi4.diff(x,2)*phi4.diff(x,2), \
                 (x,0,2*L)) \
    +k*phi4.subs(x,2*L)*phi4.subs(x,2*L)
```

Direct Variational Methods: Examples

```
f2=sp.integrate(q*phi2,(x,0,2*L))
f3=sp.integrate(q*phi3,(x,0,2*L))
f4=sp.integrate(q*phi4,(x,0,2*L))
#-----[ matrix of algebraic eqautions ]-----

A=sp.Matrix([[b22,b23,b24],[b32,b33,b34],[b42,b43,b44]])

#-----[     vector of right-hand    ]-----
#-----[ side of algebraic equations ]-----

b=sp.Matrix(3,1,[f2,f3,f4])
```

Direct Variational Methods: Examples

```
#-----[            solution using LU             ]-----
#-----[        docomposition of matrix A         ]-----
#-----[    (this method of symbolic solution     ]-----
#-----[ search is not efficient and difficult ]-----
#-----[          to handle in this case)         ]-----

sol=A.LUsolve(b)
```

Direct Variational Methods: Examples

```
#-----[     more efficient is    ]-----
#-----[ using of standard solver ]-----

eqn1=b22*c2+b23*c3+b24*c4-f2
eqn2=b32*c2+b33*c3+b34*c4-f3
eqn3=b42*c2+b43*c3+b44*c4-f4

sol=sp.solve([eqn1,eqn2,eqn3],[c2,c3,c4])

#-----[ resulting form of solution ]-----

w=sol[c2]*phi2+sol[c3]*phi3+sol[c4]*phi4

print("\nw(x)=")
sp.pprint(sp.collect(sp.expand(w),x))
```

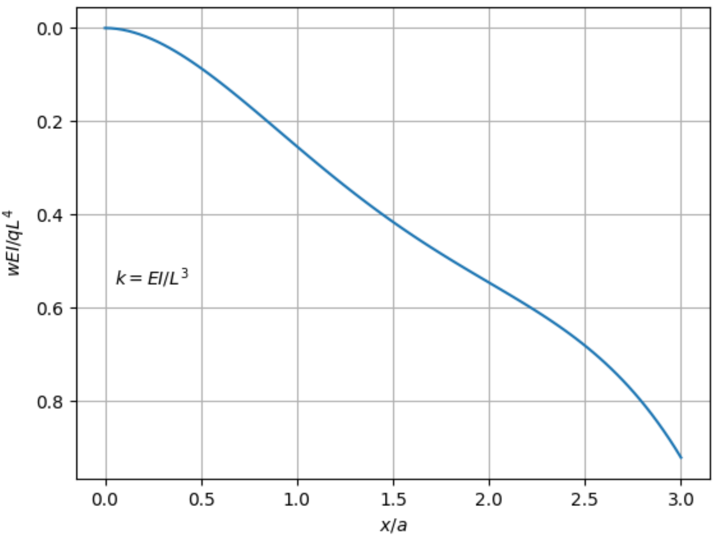
Direct Variational Methods: Examples

```
#-----[ beam deflection in x=2*L ]-----

print("\nw(2*L)=")
sp.pprint(sp.simplify(w.subs(x,2*L)))

#-----[ plot of deflextion w ]-----

x_plot=np.linspace(0,3,100)
y_plot=[]

for ii in x_plot:
  y_plot.append(w.subs(k,E*I/L**3) \
        .subs({L:1,q:1,E:1,I:1,x:ii}))
```

Direct Variational Methods: Examples

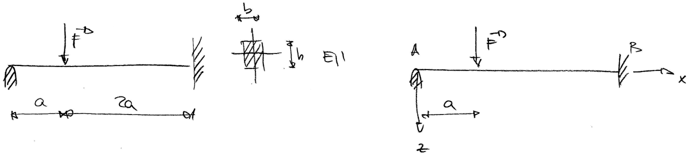
```
fig,axs=plt.subplots()
axs.plot(x_plot,y_plot)
axs.invert_yaxis()
axs.set_xlabel(r'$x/a$')
axs.set_ylabel(r'$wEI/qL^4$')
axs.text(0.05,0.55,r'$k=EI/L^3$')
axs.grid(True)
plt.savefig('plot_example11-2')
```

Direct Variational Methods: Examples



Direct Variational Methods: Examples

<u>Example 14</u>: Consider a beam, one end fixed, the second one simply supported, under the external point force loading F. Derive the transverse deflection using the Ritz method.



The principle of virtual displacement for the problem becomes

$$0 = \delta \Pi = \delta(U+V) = \delta\left(\frac{1}{2}\int_0^{3a} EI\left(\frac{d^2w}{dx^2}\right)^2 dx - \int_0^{3a} f_0\, w\, dx\right)$$

$$\Rightarrow \quad 0 = \int_0^{3a}\left(EI\,\frac{d^2w}{dx^2}\cdot\frac{d^2}{dx^2}\delta w - f_0\,\delta w\right)dx$$

where $w(0)=0$, $w(3a)=0$ and $\frac{d}{dx}w(3a)=0$ are essential boundary conditions and for the volume external force we have

$$f_0 = F\,\delta(x-a), \quad \text{where } \delta(x-a) \text{ is Dirac delta function}$$

$$\delta(x-a) = \begin{cases} 0, & x \neq a \\ \infty, & x = a \end{cases} \quad \text{and} \quad \int_{-\infty}^{\infty} \delta(x-a)\,\delta w\, dx = \delta w(a)$$

Direct Variational Methods: Examples

So we get

$$0 = \int_0^{3a} EI \frac{d^2w}{dx^2} \frac{d^2\delta w}{dx^2} dx - F \cdot \delta w \Big|_{x=a}$$

We choose a three-parameter approximation

$$w = c_1\phi_1 + c_2\phi_2 + c_3\phi_3 + \phi_0$$

, where $\phi_0 = 0$ due to the essential boundary conditions $w(0) = w(3a) = 0$,

and

$$\phi_1 = x(x-3a)^2$$

$$\phi_2 = x(x-3a)^2(x+6a)$$

$$\phi_3 = x(x-3a)^2(x^2+6ax+24a^2)$$

Basis functions fulfills homogenous essential boundary cond. $w(0)=w(3a)=0$ and $\frac{d}{dx}w(3a)=0$. So we continue

$$0 = \int_0^{3a} EI(c_1\phi_1'' + c_2\phi_2'' + c_3\phi_3'')(\delta c_1\phi_1'' + \delta c_2\phi_2'' + \delta c_3\phi_3'')\,dx -$$
$$- F(\delta c_1\phi_1(a) + \delta c_2\phi_2(a) + \delta c_3\phi_3(a))$$

Direct Variational Methods: Examples

δc_i are arbitrary, so

$$\sum_{i=1}^{3}\left(\sum_{j=1}^{3} b_{ij} c_j\right) = F \cdot \phi_i(a), \quad \text{where}$$

$$b_{ij} = \int_0^{3a} EI\, \phi_j'' \phi_i''\, dx,$$

$$b_{11} = \int_0^{3a} EI\, [\phi_1'']^2 dx = EI \int_0^{3a} \left\{\frac{d}{dx^2}\left[x(x-3a)^2\right]\right\}^2 dx = 108\, EI a^3$$

$$b_{12} = \int_0^{3a} EI\, \phi_1'' \phi_2''\, dx = EI \int_0^{3a} \frac{d}{dx^2}\left[x(x-3a)^2\right] \cdot \frac{d}{dx^2}\left[x(x-3a)^2 \cdot (x+6a)\right] dx =$$

$$= 648\, EI a^4$$

$$b_{13} = \int_0^{3a} EI\, \phi_1'' \phi_3''\, dx = EI \int_0^{3a} \frac{d}{dx^2}\left[x(x-3a)^2\right] \cdot \frac{d}{dx^2}\left[x(x-3a)^2 (x^2+2ax+24a^2)\right] dx =$$

$$= 2916\, EI a^5$$

$$b_{21} = b_{12}$$

$$b_{22} = \int_0^{3a} EI [\phi_2'']^2 dx = EI \int_0^{3a} \left\{\frac{d}{dx^2}\left[x(x-3a)^2(x+6a)\right]\right\}^2 dx = \frac{20412}{5} EI a^5$$

$$b_{23} = \int_0^{3a} EI\, \phi_2'' \cdot \phi_3''\, dx = EI \int_0^{3a} \frac{d}{dx^2}\left[x(x-3a)^2(x+6a)\right] \frac{d}{dx^2}\left[x(x-3a)^2(x^2+2ax+24a^2)\right] dx$$

Direct Variational Methods: Examples

$= 18954\, EIa^6$

$b_{31} = b_{13}$

$b_{32} = b_{23}$

$$b_{33} = \int_0^{3a} EI\,[\phi_3'']^2 dx = EI \int_0^{3a} \left\{ \frac{d}{dx^2}\left[x(x-3a)^2(x^2+2ax+24a^2) \right] \right\}^2 dx = \frac{629856}{4} EIa^4$$

We have to solve system of algebraic equations

$$EIa^3 \begin{bmatrix} 108 & 648a & 2916a^2 \\ 648a & \frac{20412}{5}a^2 & 18954a^3 \\ 2916a^2 & 18954a^3 & \frac{629856}{4}a^4 \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix} = Fa^3 \begin{bmatrix} 4 \\ 28a \\ 136a^2 \end{bmatrix}$$

from which follows

$$c_1 = -\frac{49F}{243EI}, \quad c_2 = \frac{50F}{729EIa}, \quad c_3 = -\frac{14F}{2187EIa^2}$$

and

$$w(a) = w_F = 0.2432 \frac{Fa^3}{EI} \quad \left(\text{exact solution Example 15 is } 0.2469 \frac{Fa^3}{EI}\right)$$

Direct Variational Methods: Examples

```
#-----[            ]-----
#-----[ example 14 ]-----
#-----[            ]-----

#-----[ derivation of basis ]-----
#-----[  of function space  ]-----

#-----[  import of sympy, numpy  ]-----
#-----[ and matplotlib libraries ]-----

import sympy as sp
import numpy as np
import matplotlib.pyplot as plt
```

Direct Variational Methods: Examples

```
#-----[ initiation of quality printing ]-----

sp.init_printing()

#-----[ definition of used symbols ]-----

c0,c1,c2,c3,c4,c5,c6=sp.symbols('c0 c1 c2 c3 c4 c5 c6')
x=sp.symbols('x')
a=sp.symbols('a')
```

Direct Variational Methods: Examples

```
#-----[ general form of basis functions ]-----

f=c0+c1*x+c2*x**2+c3*x**3 \
  +c4*x**4+c5*x**5+c6*x**6

#-----[ essential boundary conditions ]-----
#-----[      of solved problem        ]-----

eqn1=f.subs(x,0)
eqn2=f.subs(x,3*a)
eqn3=(f.diff(x)).subs(x,3*a)
```

Direct Variational Methods: Examples

```
#-----[ solution of essential boundary ]-----
#-----[    conditions for c0,c1,c2     ]-----

sol=sp.solve([eqn1,eqn2,eqn3],[c0,c1,c2])

#-----[ choosen basis of functional space ]-----

phi1=f.subs({c0:sol[c0],c1:sol[c1],c2:sol[c2], \
            c3:1,c4:0,c5:0,c6:0})
phi2=f.subs({c0:sol[c0],c1:sol[c1],c2:sol[c2], \
            c3:0,c4:1,c5:0,c6:0})
phi3=f.subs({c0:sol[c0],c1:sol[c1],c2:sol[c2], \
            c3:0,c4:0,c5:1,c6:0})
phi4=f.subs({c0:sol[c0],c1:sol[c1],c2:sol[c2], \
            c3:0,c4:0,c5:0,c6:1})
```

Direct Variational Methods: Examples

```
print("\nphi1=")
sp.pprint(phi1)
print("\nphi2=")
sp.pprint(phi2)
print("\nphi3=")
sp.pprint(phi3)
print("\nphi4=")
sp.pprint(phi4)

#-----[ plot of first four basis functions ]-----

x_plot=np.linspace(0,3,100)
y1_plot=[]
y2_plot=[]
y3_plot=[]
y4_plot=[]
```

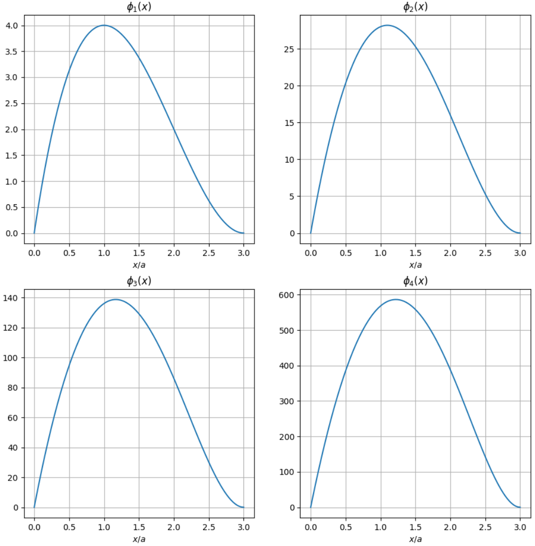
Direct Variational Methods: Examples

```
for ii in x_plot:
  y1_plot.append(phi1.subs({a:1,x:ii}))
  y2_plot.append(phi2.subs({a:1,x:ii}))
  y3_plot.append(phi3.subs({a:1,x:ii}))
  y4_plot.append(phi4.subs({a:1,x:ii}))

fig,axs=plt.subplots(2,2,figsize=(11,11))
axs[0,0].plot(x_plot,y1_plot)
axs[0,0].set_title(r'$\phi_1(x)$')
axs[0,0].set_xlabel(r'$x/a$')
axs[0,0].grid(True)
```

Direct Variational Methods: Examples

```
axs[0,1].plot(x_plot,y2_plot)
axs[0,1].set_title(r'$\phi_2(x)$')
axs[0,1].set_xlabel(r'$x/a$')
axs[0,1].grid(True)
axs[1,0].plot(x_plot,y3_plot)
axs[1,0].set_title(r'$\phi_3(x)$')
axs[1,0].set_xlabel(r'$x/a$')
axs[1,0].grid(True)
axs[1,1].plot(x_plot,y4_plot)
axs[1,1].set_title(r'$\phi_4(x)$')
axs[1,1].set_xlabel(r'$x/a$')
axs[1,1].grid(True)
plt.savefig('plot_example14-1')
```

Direct Variational Methods: Examples



Direct Variational Methods: Examples

```
#-----[            ]-----
#-----[ example 14 ]-----
#-----[            ]-----

#-----[ solution found using Ritz method ]-----

#-----[  import of sympy, numpy  ]-----
#-----[ and matplotlib libraries ]-----

import sympy as sp
import numpy as np
import matplotlib.pyplot as plt

#-----[ initialization of quality printing ]-----

sp.init_printing()
```

Direct Variational Methods: Examples

```
#-----[ symbol definition ]-----

a,E,I=sp.symbols('a E I')
x=sp.symbols("x")

#-----[ basis of function space of solution ]-----

phi1=x*(x-3*a)**2
phi2=x*(x-3*a)**2*(x+6*a)
phi3=x*(x-3*a)**2*(x**2+6*a*x+27*a**2)
```

Direct Variational Methods: Examples

```
#-----[    system of algebraic equations     ]-----
#-----[ resulting from Hamilton's principle ]-----

b11=sp.integrate(E*I*phi1.diff(x,2)**2,(x,0,3*a))
b12=sp.integrate(E*I*phi1.diff(x,2)*phi2.diff(x,2), \
                 (x,0,3*a))
b13=sp.integrate(E*I*phi1.diff(x,2)*phi3.diff(x,2), \
                 (x,0,3*a))
b21=b12
b22=sp.integrate(E*I*phi2.diff(x,2)*phi2.diff(x,2), \
                 (x,0,3*a))
b23=sp.integrate(E*I*phi2.diff(x,2)*phi3.diff(x,2), \
                 (x,0,3*a))
b31=b13
b32=b23
b33=sp.integrate(E*I*phi3.diff(x,2)*phi3.diff(x,2), \
                 (x,0,3*a))
```

Direct Variational Methods: Examples

```
b1=F*phi1.subs(x,a)
b2=F*phi2.subs(x,a)
b3=F*phi3.subs(x,a)

#-----[ matrix of algebraic eqautions ]-----

A=sp.Matrix([[b11,b12,b13],[b21,b22,b23],[b31,b32,b33]])

#-----[ vector of right-hand side ]-----
#-----[   of algebraic equations  ]-----

b=sp.Matrix(3,1,[b1,b2,b3])
```

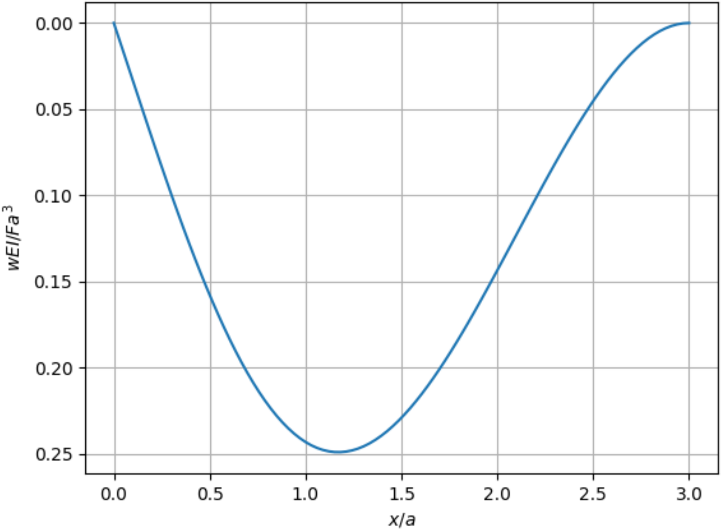
Direct Variational Methods: Examples

```
#-----[ solution using LU docomposition ]-----
#-----[           of matrix A           ]-----

sol=A.LUsolve(b)

#-----[ resulting form of solution ]-----

w=sol[0]*phi1+sol[1]*phi2+sol[2]*phi3

print("\nw(x)=")
sp.pprint(sp.collect(sp.expand(w),x))

#-----[ beam deflection in x=a ]-----

print("\nw(a)=")
sp.pprint(sp.collect(sp.expand(w),x).subs(x,a))
```

Direct Variational Methods: Examples

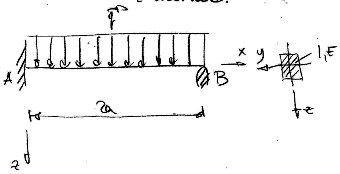
```
#-----[ plot of deflextion w ]-----

x_plot=np.linspace(0,3,100)
y_plot=[]

for ii in x_plot:
  y_plot.append(w.subs({a:1,F:1,E:1,I:1,x:ii}))

fig,axs=plt.subplots()
axs.plot(x_plot,y_plot)
axs.invert_yaxis()
axs.set_xlabel(r'$x/a$')
axs.set_ylabel(r'$wEI/Fa^3$')
axs.grid(True)
plt.savefig('plot_example14-2')
```

Direct Variational Methods: Examples



Direct Variational Methods: Examples

Example 21: Derive the transverse deflection of the beam in the figure using the Ritz method.



The principle of virtual displacement for the problem becomes

$$0 = \delta \Pi = \delta(U+V) = \delta\left(\frac{1}{2}\int_0^{2a} EI \left(\frac{d^2w}{dx^2}\right)^2 dx - \int_0^{2a} q w \, dx \right)$$

$$\Rightarrow \quad 0 = \int_0^{2a} EI \frac{d^2w}{dx^2} \delta\left(\frac{d^2w}{dx^2}\right) dx - q \int_0^{2a} \delta w \, dx$$

where $w(0)=0$, $\frac{d}{dx}w(0)=0$ and $w(2a)=0$ are essential boundary conditions.

We choose a two-parameter approximation, see example21-1.py,

$$w = c_3\phi_3 + c_4\phi_4$$

where

$$\phi_3 = -2ax^2 + x^3 \quad \text{and} \quad \phi_4 = -4a^2x^2 + x^4,$$

Direct Variational Methods: Examples

Both functions, ϕ_3 and ϕ_4, fulfill essentialy boundary conditions

$$\phi_3(0) = \phi_3'(0) = \phi_3(2a) = 0,$$

$$\phi_4(0) = \phi_4'(0) = \phi_4(2a) = 0.$$

Substituting ϕ_3, ϕ_4 into the principle of virtual displacements

$$0 = \int_0^{2a} EI\left(c_3\phi_3'' + c_4\phi_4''\right)\left(\delta c_3 \phi_3'' + \delta c_4 \phi_4''\right)dx - q\int_0^{2a}\left(\delta c_3\phi_3 + \delta c_4 \phi_4\right)dx$$

$\delta c_3, \delta c_4$ are arbitrary, hence

$$c_3\int_0^{2a} EI(\phi_3'')^2 dx + c_4 \int_0^{2a} EI\phi_3''\phi_4'' dx - q\int_0^{2a}\phi_3 dx = 0,$$

$$c_3\int_0^{2a} EI\phi_3''\phi_4'' dx + c_4\int_0^{2a} EI(\phi_4'')^2 dx - q\int_0^{2a}\phi_4 dx = 0.$$

By the integration we get a system of algebraic eqations with solution

$$EI\begin{bmatrix} 32a^3 & 128a^4 \\ 128a^4 & \frac{2688a^5}{5}\end{bmatrix}\begin{bmatrix} c_3 \\ c_4\end{bmatrix} = qa^4\begin{bmatrix} -\frac{4}{3} \\ -\frac{64}{15}a\end{bmatrix} \Rightarrow \begin{cases} c_4 = -\frac{5}{24}\frac{qa}{EI} \\ c_3 = \frac{1}{24}\frac{q}{EI}\end{cases}$$

Direct Variational Methods: Examples

The final solution is

$$w = \frac{q}{4EI}\left[a^2x^2 - \frac{5}{6}ax^3 + \frac{1}{6}x^4\right]$$ (it is an exact solution of the problem)

and the deflection at the center of the cantilever

$$w(a) = \frac{qa^4}{12EI}$$

Direct Variational Methods: Examples

```
#-----[                                   ]-----
#-----[             example 21            ]-----
#-----[                                   ]-----
#-----[ derivation of basis of function space ]-----

#-----[  import of sympy, numpy  ]-----
#-----[ and matplotlib libraries ]-----

import sympy as sp
import numpy as np
import matplotlib.pyplot as plt

#-----[ initiation of quality printing ]-----

sp.init_printing()
```

Direct Variational Methods: Examples

```
#-----[ definition of used symbols ]-----

c0,c1,c2,c3,c4,c5,c6=sp.symbols('c0 c1 c2 c3 \
                                  c4 c5 c6')
x=sp.symbols('x')
a=sp.symbols('a')

#-----[ general form of basis functions ]-----

f=c0+c1*x+c2*x**2+c3*x**3+c4*x**4+c5*x**5+c6*x**6

#-----[ essential boundary conditions of ]-----
#-----[          solved problem          ]-----

eqn1=f.subs(x,0)
eqn2=(f.diff(x)).subs(x,0)
eqn3=f.subs(x,2*a)
```

Direct Variational Methods: Examples

```
#-----[ solution of essential boundary ]-----
#-----[     conditions for c0,c1,c2i   ]------

sol=sp.solve([eqn1,eqn2,eqn3],[c0,c1,c2])

#-----[ choosen basis of functional space ]-----

phi3=f.subs({c0:sol[c0],c1:sol[c1],c2:sol[c2], \
             c3:1,c4:0,c5:0,c6:0})
phi4=f.subs({c0:sol[c0],c1:sol[c1],c2:sol[c2], \
             c3:0,c4:1,c5:0,c6:0})
phi5=f.subs({c0:sol[c0],c1:sol[c1],c2:sol[c2], \
             c3:0,c4:0,c5:1,c6:0})
phi6=f.subs({c0:sol[c0],c1:sol[c1],c2:sol[c2], \
             c3:0,c4:0,c5:0,c6:1})
```

Direct Variational Methods: Examples

```
print("\nphi3=")
sp.pprint(phi3)
print("\nphi4=")
sp.pprint(phi4)
print("\nphi5=")
sp.pprint(phi5)
print("\nphi6=")
sp.pprint(phi6)
```

Direct Variational Methods: Examples

```
#-----[ plot of first two basis functions ]-----

x_plot=np.linspace(0,2,100)
y3_plot=[]
y4_plot=[]
y5_plot=[]
y6_plot=[]

for ii in x_plot:
  y3_plot.append(phi3.subs({a:1,x:ii}))
  y4_plot.append(phi4.subs({a:1,x:ii}))
  y5_plot.append(phi5.subs({a:1,x:ii}))
  y6_plot.append(phi6.subs({a:1,x:ii}))
```

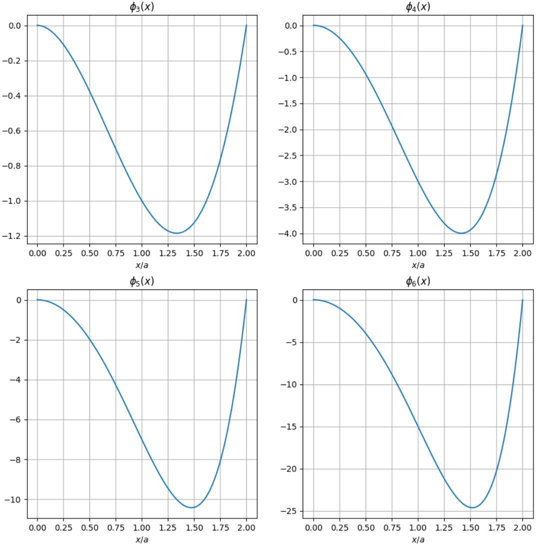
Direct Variational Methods: Examples

```
fig,axs=plt.subplots(2,2,figsize=(11,11))
axs[0,0].plot(x_plot,y3_plot)
axs[0,0].set_title(r'$\phi_3(x)$')
axs[0,0].set_xlabel(r'$x/a$')
axs[0,0].grid(True)
axs[0,1].plot(x_plot,y4_plot)
axs[0,1].set_title(r'$\phi_4(x)$')
axs[0,1].set_xlabel(r'$x/a$')
axs[0,1].grid(True)
```

Direct Variational Methods: Examples

```
axs[1,0].plot(x_plot,y5_plot)
axs[1,0].set_title(r'$\phi_5(x)$')
axs[1,0].set_xlabel(r'$x/a$')
axs[1,0].grid(True)
axs[1,1].plot(x_plot,y6_plot)
axs[1,1].set_title(r'$\phi_6(x)$')
axs[1,1].set_xlabel(r'$x/a$')
axs[1,1].grid(True)
plt.savefig('plot_example21-1')
```

Direct Variational Methods: Examples



Direct Variational Methods: Examples

```
#-----[                                  ]-----
#-----[            example 21            ]-----
#-----[                                  ]-----
#-----[ solution found using Ritz method ]-----

#-----[  import of sympy, numpy  ]-----
#-----[ and matplotlib libraries ]-----

import sympy as sp
import numpy as np
import matplotlib.pyplot as plt

#-----[ initialization of quality printing ]-----

sp.init_printing()
```

Direct Variational Methods: Examples

```
#-----[ symbol definition ]-----

a,E,I,q=sp.symbols('a E I q')
x=sp.symbols("x")

#-----[ basis of function space of solution ]-----

phi3=-2*a*x**2+x**3
phi4=-4*a**2*x**2+x**4
```

Direct Variational Methods: Examples

```
#-----[    system of algebraic equations    ]-----
#-----[ resulting from Hamilton's principle ]-----

a11=sp.integrate(E*I*phi3.diff(x,2)**2,(x,0,2*a))
a12=sp.integrate(E*I*phi3.diff(x,2)*phi4.diff(x,2),
                 (x,0,2*a))
a21=a12
a22=sp.integrate(E*I*phi4.diff(x,2)**2,(x,0,2*a))

b1=q*sp.integrate(phi3,(x,0,2*a))
b2=q*sp.integrate(phi4,(x,0,2*a))
```

Direct Variational Methods: Examples

```
#-----[ matrix of algebraic eqautions ]-----

A=sp.Matrix([[a11,a12],[a21,a22]])

#-----[ vector of right-hand side ]-----
#-----[  of algebraic equations   ]-----

b=sp.Matrix(2,1,[b1,b2])

#-----[        solution using        ]-----
#-----[ LU docomposition of matrix A ]-----

sol=A.LUsolve(b)
```

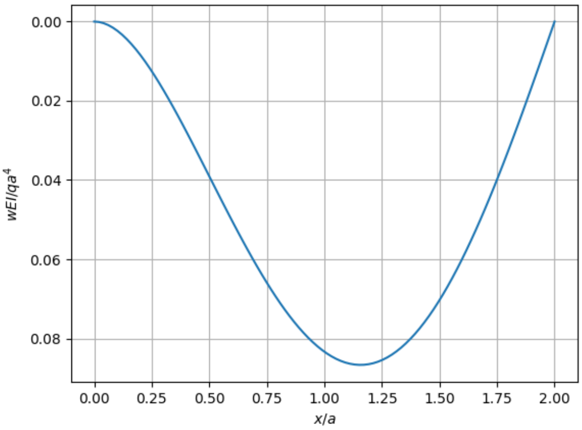
Direct Variational Methods: Examples

```
#-----[ resulting form of solution ]-----

w=sol[0]*phi3+sol[1]*phi4

print("\nw(x)=")
sp.pprint(sp.collect(sp.expand(w),x))

#-----[ beam deflection in x=a ]-----

print("\nw(a)=")
sp.pprint(sp.collect(sp.expand(w),x).subs(x,a))
```

Direct Variational Methods: Examples

```
#-----[ plot of deflextion w ]-----

x_plot=np.linspace(0,2,100)
y_plot=[]

for ii in x_plot:
  y_plot.append(w.subs({a:1,q:1,E:1,I:1,x:ii}))

fig,axs=plt.subplots()
axs.plot(x_plot,y_plot)
axs.invert_yaxis()
axs.set_xlabel(r'$x/a$')
axs.set_ylabel(r'$wEI/qa^4$')
axs.grid(True)
plt.savefig('plot_example21-2')
```

Direct Variational Methods: Examples



Thank you!