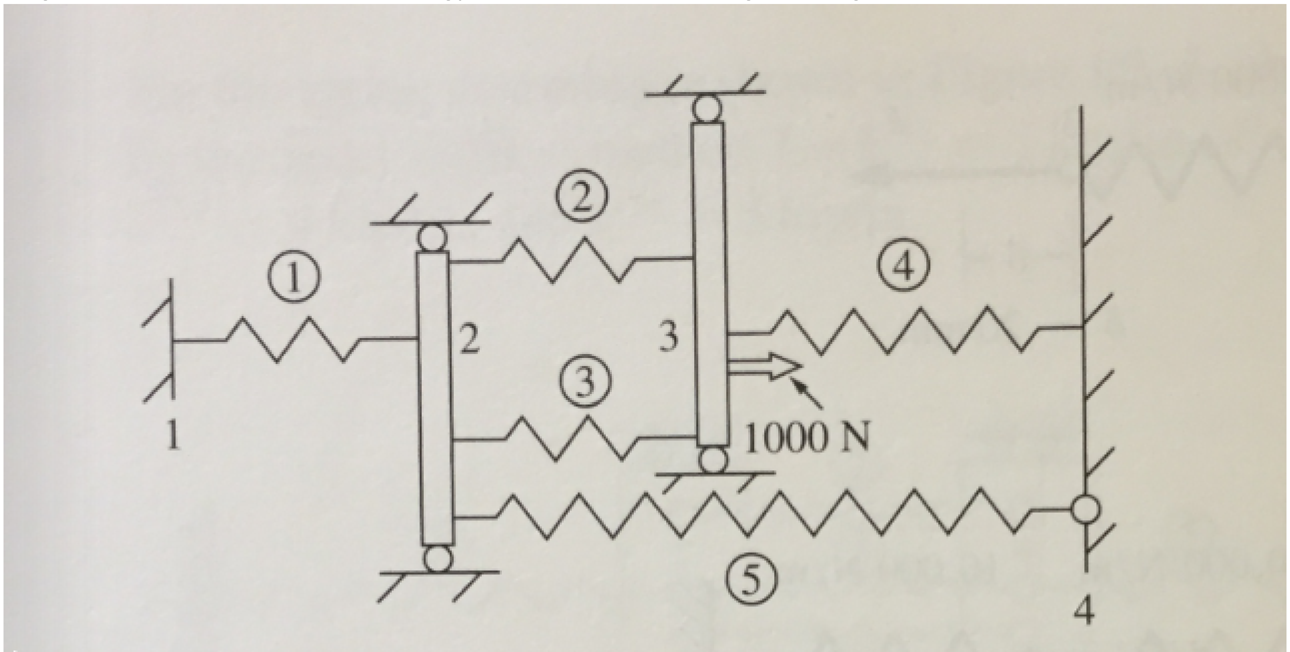
Problem 2.17

For the five-spring assemblage shown in Figure P2-17, determine the displacements at nodes 2 and 3 and the reactions at nodes 1 and 4. Assume the rigid vertical bars at nodes 2 and 3 connecting the springs remain horizontal at all times but are free to slide or displace left or right. There is an applied force at node 3 of 1000 N to the right. Use minimum potential energy method to set up the governing matrix equation.



■ Figure P2–17

$$Let\ k_1 = 500\frac{N}{mm}, \quad k_2 = k_3 = 300\frac{N}{mm}, \quad k_4 = k_5 = 400\frac{N}{mm}$$

In [1]:
```
# Import necessary packages
import sympy as sp
sp.init_printing()
```

In [2]:
```
# Create necessary symbols
k1, k2, k3, k4, k5 = sp.symbols('k_1, k_2, k_3, k_4, k_5')
force_list = sp.symbols('F1, F2, F3, F4')
displacement_list = sp.symbols('u_1, u_2, u_3, u_4')
force_column_matrix = sp.Matrix(force_list)
nodal_displacement_column_matrix = sp.Matrix(displacement_list)
```

In [3]:

```
# Formulate strain energy of the system
U = 1/2*(k1*sp.Pow((displacement_list[1]-displacement_list[0]),2)\
        +k2*sp.Pow((displacement_list[2]-displacement_list[1]),2)
        +k3*sp.Pow((displacement_list[2]-displacement_list[1]),2)
        +k4*sp.Pow((displacement_list[3]-displacement_list[2]),2)
        +k5*sp.Pow((displacement_list[3]-displacement_list[1]),2))

# Formulate potential energy by external nodal forces
sig = -sp.transpose(force_column_matrix)*nodal_displacement_column_matrix
sig = sig[0]

# Formulate total potential energy of the assemblage
potential_energy = U-sig
potential_energy
```

Out[3]:

$$F_1 u_1 + F_2 u_2 + F_3 u_3 + F_4 u_4 + 0.5k_1(-u_1 + u_2)^2 + 0.5k_2(-u_2 + u_3)^2 + 0.5k_3(-u_2 +$$
$$+ 0.5k_4(-u_3 + u_4)^2 + 0.5k_5(-u_2 + u_4)^2$$

In [4]:

```
n = len(displacement_list)
global_stiffness_matrix = sp.zeros(n, n)

# Populate global stiffness Matrix with appropriate partial derivatives
for i, axis_i_displacement in enumerate(displacement_list):
    for j, axis_j_displacement in enumerate(displacement_list):
        global_stiffness_matrix[i, j] = sp.expand(sp.diff(potential_energy, di
splacement_list[i])).coeff(displacement_list[j], 1)
global_stiffness_matrix
```

Out[4]:

$$\begin{bmatrix} 1.0k_1 & -1.0k_1 & 0 & 0 \\ -1.0k_1 & 1.0k_1 + 1.0k_2 + 1.0k_3 + 1.0k_5 & -1.0k_2 - 1.0k_3 & -1.0k_5 \\ 0 & -1.0k_2 - 1.0k_3 & 1.0k_2 + 1.0k_3 + 1.0k_4 & -1.0k_4 \\ 0 & -1.0k_5 & -1.0k_4 & 1.0k_4 + 1.0k_5 \end{bmatrix}$$

In [5]:
```
# populate global stiffness matrix with numbers
global_stiffness_matrix = global_stiffness_matrix.subs({k1:500,\
                                                        k2:300,
                                                        k3:300,
                                                        k4:400,
                                                        k5:400})

# populate force matrix with numbers
force_column_matrix = force_column_matrix.subs({force_list[1]:0,\
                                               force_list[2]:1000})

# populate nodal displacement matrix with numbers
nodal_displacement_column_matrix = nodal_displacement_column_matrix.subs({disp
lacement_list[0]:0,\
                                                                          disp
lacement_list[-1]:0})
global_stiffness_matrix
```

Out[5]:

$$\begin{bmatrix} 500.0 & -500.0 & 0 & 0 \\ -500.0 & 1500.0 & -600.0 & -400.0 \\ 0 & -600.0 & 1000.0 & -400.0 \\ 0 & -400.0 & -400.0 & 800.0 \end{bmatrix}$$

In [6]:
```
force_column_matrix
```

Out[6]:

$$\begin{bmatrix} F_1 \\ 0 \\ 1000 \\ F_4 \end{bmatrix}$$

In [7]:
```
nodal_displacement_column_matrix
```

Out[7]:

$$\begin{bmatrix} 0 \\ u_2 \\ u_3 \\ 0 \end{bmatrix}$$

Final Ans:

The governing matrix equation is:

$$\begin{bmatrix} 500 & -500 & 0 & 0 \\ -500 & 1500 & -600 & -400 \\ 0 & -600 & 1000 & -400 \\ 0 & -400 & -400 & 800 \end{bmatrix} * \begin{Bmatrix} 0 \\ u_2 \\ u_3 \\ 0 \end{Bmatrix} = \begin{Bmatrix} F1 \\ 0 \\ 1000 \\ F4 \end{Bmatrix}$$