Numerical Analysis and Computational Mathematics - 2015-2016
Section CSE - Dr. Luca Dedè
Session 11 - Linear Systems & Eigenvalues and Eigenvectors

Exercise 1 (MATLAB)
Let us consider the two dimensional *truss bridge* reported in Figure 1 composed by triangular units defined by *beams* connecting in some *nodes* (joints). We consider N_{nodes}, with N_{nodes} odd, delimiting the beams. The triangular units are representable as equilateral triangles with beams of the same length. We assume that all the beams possess the same elastic properties and they can uniquely carry axial loads. External forces can be applied only in the nodes. Our goal is to compute the displacements of the nodes of the bridge under the action of external forces.
Let us indicate with $\mathbf{u}_i \in \mathbb{R}^2$ the displacement vector of each node $i = 1, \ldots, N_{nodes}$ and the external forces with $\mathbf{f}_i^{ext} \in \mathbf{R}^2$. The internal forces $\mathbf{f}_{i,j}^{int}$ acting between the beams and the nodes depend on the displacements of the nodes i and j and the beams' orientation; they reads:

$$\mathbf{f}_{i,i-1}^{int} = k_{beam} T_a (\mathbf{u}_i - \mathbf{u}_{i-1}) \qquad \text{for } i = 2, 4, 6, \ldots, N_{nodes} - 1,$$

$$\mathbf{f}_{i,i-1}^{int} = k_{beam} T_b (\mathbf{u}_i - \mathbf{u}_{i-1}) \qquad \text{for } i = 3, 5, 7, \ldots, N_{nodes},$$

$$\mathbf{f}_{i,i-2}^{int} = k_{beam} T_h (\mathbf{u}_i - \mathbf{u}_{i-2}) \qquad \text{for } i = 3, 4, 5, \ldots, N_{nodes},$$

where k_{beam} indicates the elastic properties of the beams and:

$$T_a = \frac{1}{4} \begin{bmatrix} 1 & \sqrt{3} \\ \sqrt{3} & 3 \end{bmatrix}, \qquad T_b = \frac{1}{4} \begin{bmatrix} 1 & -\sqrt{3} \\ -\sqrt{3} & 3 \end{bmatrix}, \qquad T_h = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}.$$

We impose the equilibrium of the internal and external forces in each node of the bridge; we set:

$$\sum_{j \in I_i} \mathbf{f}_{i,j}^{int} = \mathbf{f}_i^{ext}, \qquad I_i := \{i-2, i-1, i+1, i+2\} \cap \{k\}_{k=1}^{N_{nodes}}, \qquad \text{for all } i = 1, \ldots, N_{nodes}.$$

This leads to the following discrete structural problem:

$$A\mathbf{d} = \mathbf{b},$$

where:

$$\mathbf{d} = ((\mathbf{u}_1)^T, (\mathbf{u}_2)^T, \ldots, (\mathbf{u}_{N_{nodes}})^T)^T$$

is the unknown solution vector representing the displacements of each node of the bridge. We observe that $\mathbf{d} \in \mathbb{R}^n$, with $n = 2N_{nodes}$, since a local displacement vector $\mathbf{u}_i = (u_{x,i}, u_{y,i})^T \in \mathbb{R}^2$ with two components, one horizontal $u_{x,i}$ and one vertical $u_{y,i}$, is associated to each node $i = 1, \ldots, N_{nodes}$. The matrix $A \in \mathbb{R}^{n \times n}$ is called the stiffness matrix and depends on the elastic properties of the

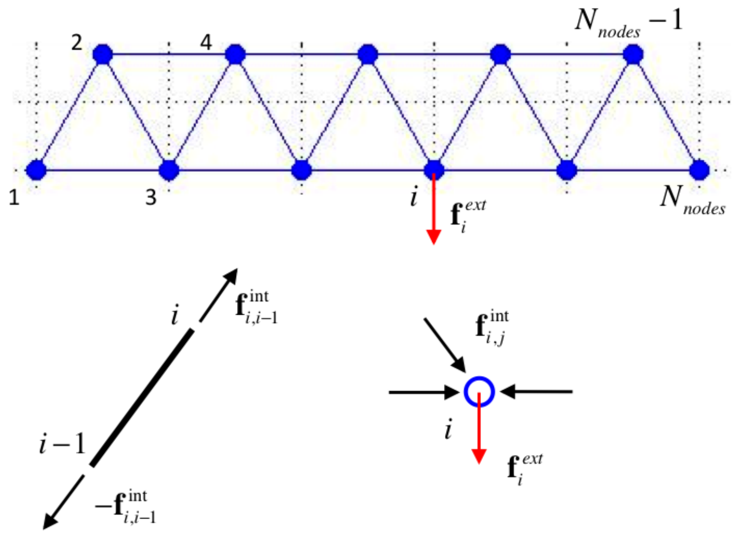

Figure 1: Truss bridge model.

beams and the pattern of the triangular units. The matrix A is sparse and possesses the following block pattern:

$$
A = k_{beam} \begin{bmatrix}
(T_h+T_a) & -T_a & -T_h & & & & \\
-T_a & (T_h+T_a+T_b) & -T_b & -T_h & & & \\
-T_h & -T_b & (2T_h+T_a+T_b) & -T_a & -T_h & & \\
 & -T_h & -T_a & (2T_h+T_a+T_b) & -T_b & -T_h & \\
 & & \ddots & \ddots & \ddots & \ddots & \ddots \\
 & & & & & & \\
 & & & & -T_h & -T_b & (T_h+T_b)
\end{bmatrix}.
$$

The vector $\mathbf{b} \in \mathbb{R}^n$ contains the external forces $\mathbf{f}_i^{ext} \in \mathbf{R}^2$ acting on each node $i = 1, \ldots, N_{nodes}$ and reads:

$$
\mathbf{b} = ((\mathbf{f}_1^{ext})^T, (\mathbf{f}_2^{ext})^T, \ldots, (\mathbf{f}_{N_{nodes}}^{ext})^T)^T.
$$

The nodes are numbered from left to right.

a) Set $N_{nodes} = 29$ and $k_{beam} = 10^3$. Assign the stiffness matrix A by means of the MATLAB function `bridge_stiffness_matrix.m` (we remark that the matrix A is sparse) and visualize the truss bridge by means of the MATLAB function `plot_bridge.m`. Plot the pattern of the matrix A by using the MATLAB command `spy`, verify that the matrix is symmetric, and that it is singular (use the MATLAB command `condest` to estimate the conditioning number of the sparse matrix).

b) In order to make the structural problem well-posed we need to constrain at least three unknowns of the vector $\mathbf{d}$ of the nodes displacements. In particular, for the bridge, it is convenient to impose zero displacements in the first node for both the components ($u_{x,1} = 0$ and $u_{y,1} = 0$) and a null vertical displacement in the last node ($u_{y,N_{nodes}} = 0$). Impose these constraints for the system $A\mathbf{d} = \mathbf{b}$ in order to obtain the system of reduced size $\widetilde{A}\widetilde{\mathbf{d}} = \widetilde{b}$, such that $\widetilde{A} \in \mathbb{R}^{\widetilde{n}\times\widetilde{n}}$, $\widetilde{\mathbf{d}}$, $\widetilde{\mathbf{b}} \in \mathbb{R}^{\widetilde{n}}$, where $\widetilde{n} = n - 3$. Verify that the matrix $\widetilde{A}$ is nonsingular.

c) Impose the external forces $\mathbf{f}_i^{ext} = (0, -1)^T$ for each node $i = 1, 3, \ldots, N_{nodes}$ and compute the displacement vector $\widetilde{\mathbf{d}}$ by means of the MATLAB command \. Build the vector of the displacements $\mathbf{d}$ from $\widetilde{\mathbf{d}}$ an visualize the deformed configuration of the bridge by means of the MATLAB function `plot_bridge.m`.

d) Let us suppose to be interested in determining the displacements of the nodes for the bridge defined at point a) under several different external loads $\mathbf{f}_i^{ext}$ for $i = 1, \ldots, N_{nodes}$. Suitably use the *LU* factorization method to solve the linear system $\widetilde{A}\widetilde{\mathbf{d}} = \widetilde{\mathbf{b}}$ for 10 different (random) external loads. (*Hint*: use the MATLAB command `lu` and the functions `backward_substitutions.m` and `forward_substitutions.m`).

e) Repeat point c) by considering now some iterative methods for the solution of the linear system $\widetilde{A}\widetilde{\mathbf{d}} = \widetilde{\mathbf{b}}$. Firstly, use the MATLAB function `gmres` implementing the GMRES method with a tolerance for the stopping criterion equal to $tol = 5 \cdot 10^{-5}$ and report the number of iterations required to converge to the approximate solution (*Hint*: use the following command `gmres( Ar, br, [], tol, nr )` with `Ar` corresponding to $\widetilde{A}$, `br` to $\widetilde{\mathbf{b}}$, and `nr` to $\widetilde{n}$).

Similarly, use the MATLAB function `pcg` to solve the linear system with the conjugate gradient method (without preconditioning); set $tol = 5 \cdot 10^{-5}$ and the maximum number of iterations equal to 1000. Report the number of iterations required to converge.

Exercise 2 (MATLAB)

Let us consider the matrix $A \in \mathbb{R}^{n \times n}$ such that $A = \begin{bmatrix} 5 & -2 & -1 & 0 \\ -2 & 5 & -1 & -1 \\ -1 & -1 & 4 & -1 \\ 0 & -1 & -1 & 5 \end{bmatrix}$, with $n = 4$.

a) Write a MATLAB function `power_method.m` which implements the *power method* to compute the largest (in modulus) eigenvalue of a general matrix $A \in \mathbb{C}^{n \times n}$. The function should return as outputs the approximated largest (in modulus) eigenvalue $\lambda_1^{(k_c)}$ and the number of iterations k_c required to converge. The inputs should be the matrix A, the initial vector $\mathbf{x}^{(0)} \in \mathbb{C}^n \neq \mathbf{0}$, the tolerance *tol* for the stopping criterion based on the relative difference of successive approximated eigenvalues ($\frac{|\lambda_1^{(k)} - \lambda_1^{(k-1)}|}{|\lambda_1^{(k)}|} < tol$ for $k \geq 1$), and the maximum number of iterations k_{max}. Use the following template:

```
function [ lambda, x, k ] = power_method( A, x0, tol, kmax )
% POWER_METHOD power method for the computation of the largest eigenvalue
% (in modulus) of the matrix A (\lambda_1). We assume that A is square,
% |\lambda_1| > |lambda_i| for i=2,...n, and \lambda_1 non zero
% Stopping criterion based on the relative difference of successive
% iterates of the eigenvalue.
%  [ lambda, x ] = power_method( A, x0, tol, kmax )
%  Inputs: A      = matrix (n x n)
%          x0     = initial vector (n x 1)
%          tol    = tolerance for the stopping criterion
%          kmax   = maximum number of iterations
%  Output: lambda = computed (largest) eigenvalue
%          x      = computed eigenvector correspoding to lambda
%          k      = number of iterations
%

return
```

b) Compute the largest (in modulus) eigenvalue of the matrix A by means of the MATLAB function `power_method.m` by setting $\mathbf{x}^{(0)} = (1, \dots, 1)^T$, $tol = 10^{-6}$, and the maximum number of iterations $k_{max} = 100$. Report the number of iterations k_c required to converge, the corresponding approximated eigenvalue $\lambda_1^{(k_c)}$, and the relative error $e^{(k_c)} = \dfrac{|\lambda_1 - \lambda_1^{(k_c)}|}{|\lambda_1|}$, where λ_1 is the guess of the eigenvalue computed by means of the MATLAB command `eig`.

c) By recalling that the eigenvalues of A are the reciprocal of those of A^{-1} (if A is nonsingular), use the MATLAB function `power_method.m` to compute the smallest (in modulus) eigenvalue of A, say λ_n; use the same inputs indicated at point b).

d) Suitably use the power method with *shift* to compute the smallest (in modulus) eigenvalue of A; report the value of the shift used and the number of iterations required to converge to λ_n.

Exercise 3 (MATLAB)
Let us consider the structural model for the bridge of Exercise 1. Let us assume that at each node of the bridge is attributed a mass m_{node} for which a diagonal mass matrix can be defined, say $M \in \mathbb{R}^{n\times n} = \text{diag}(m_{node}, \dots, m_{node})$.

a) Assign the stiffness matrix $A \in \mathbb{R}^{n\times n}$ for the model of the bridge with $N_{nodes} = 29$ nodes by means of the MATLAB function `bridge_stiffness_matrix.m` and the corresponding sparse mass matrix $M \in \mathbb{R}^{n\times n}$, with $n = 2N_{nodes}$; set $m_{node} = 1$ and $k_{beam} = 4 \cdot 10^2$. Then, suitably constrain three displacements in the model of the bridge as at point b) of Exercise 1 in order to obtain the generalized eigenvalue problem $\widetilde{A}\widetilde{\mathbf{x}} = \widetilde{\lambda}\widetilde{M}\widetilde{\mathbf{x}}$, with $\widetilde{\lambda} \in \mathbb{C}$, $\widetilde{\mathbf{x}} \in \mathbb{C}^{\widetilde{n}}$, and $\widetilde{A}, \widetilde{M} \in \mathbb{R}^{\widetilde{n}\times\widetilde{n}}$, where $\widetilde{n} = n - 3$. Verify that the matrices $\widetilde{A}$ and $\widetilde{M}$ are nonsingular.

b) Compute the 10 smallest (in modulus) eigenvalues of the generalized eigenvalue problem $\widetilde{A}\widetilde{\mathbf{x}} = \widetilde{\lambda}\widetilde{M}\widetilde{\mathbf{x}}$ by suitably using the MATLAB command `eigs`. By using the MATLAB function `plot_bridge.m` visualize the eigenmodes corresponding to the first, fourth, seventh, and tenth eigenvalue[1]. (*Hint*: consider the eigenmodes as displacement vectors associated to the nodes; observe that the size of $\widetilde{\mathbf{x}}$ is $\widetilde{n} = n - 3 < n$ and build $\mathbf{x}$ from $\widetilde{\mathbf{x}}$ by recalling the displacement constraints introduced at point a)).

[1]Since the matrix $\widetilde{A}$ is symmetric and positive definite, all the eigenvalues are real and positive.