

MATH661 Project 1

Large system factorization and model reduction

Posted: 10/02/23

Due: 10/16/21, 11:55PM (first draft, comments will be returned, revision due on 10/30)

This project presents a typical medium-size problem involving the concepts from approximation using linear combinations: transformation of coordinates, least squares, different types of basis sets.

1 Introduction

1.1 Modeling binary interaction in science and engineering

Many systems consist of point interactions. The iconic Eiffel Tower can be modeled through the trusses linking two points with coordinates $\mathbf{x}_i, \mathbf{x}_j \in \mathbb{R}^3$. The unit vector along direction from point i to point j is

$$\mathbf{l}_{ij} = \frac{\mathbf{x}_j - \mathbf{x}_i}{\|\mathbf{x}_j - \mathbf{x}_i\|}.$$

Suppose that structure deformation changes the coordinates to $\mathbf{x}_i + \mathbf{u}_i, \mathbf{x}_j + \mathbf{u}_j$. The most important structural response is a force \mathbf{f}_i exerted on point i that can be approximated as

$$\mathbf{f}_i = \mathbf{l}_{ij} \mathbf{l}_{ij}^T (\mathbf{u}_j - \mathbf{u}_i),$$

in appropriate physical units. An opposite force $\mathbf{f}_j = -\mathbf{f}_i$ acts on point j by principle of action and reaction. Note the projector along the truss direction $\mathbf{P}_{ij} = \mathbf{l}_{ij} \mathbf{l}_{ij}^T$, allowing computation of nodal forces as

$$\mathbf{f}_i = \mathbf{P}_{ij} \mathbf{u}_j - \mathbf{P}_{ij} \mathbf{u}_i$$

$$\mathbf{f}_j = -\mathbf{P}_{ij} \mathbf{u}_j + \mathbf{P}_{ij} \mathbf{u}_i$$

Adding forces on a point from all trusses leads to the linear relation

$$\mathbf{f} = \mathbf{K} \mathbf{u} \tag{1}$$

with $\mathbf{f}, \mathbf{u} \in \mathbb{R}^m$ the forces and displacements, $\mathbf{K} \in \mathbb{R}^{m \times m}$ the stiffness matrix, $d=3$ the number of dimensions, N the number of points, and $m = dN$, the number of degrees of freedom of the structure. This is a simple example of a finite element method. The vectors \mathbf{f}, \mathbf{u} contain the forces, displacements at all nodes, e.g.,

$$\mathbf{u} = \begin{bmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \\ \vdots \\ \mathbf{u}_m \end{bmatrix},$$

with each $\mathbf{u}_i \in \mathbb{R}^d$. Assembly of the stiffness matrix \mathbf{K} is carried out by adding contributions from each truss

$$\begin{bmatrix} \vdots \\ \mathbf{f}_i \\ \vdots \\ \mathbf{f}_j \\ \vdots \end{bmatrix} = \begin{bmatrix} \ddots & \vdots & \dots & \vdots & & \\ & -\mathbf{P}_{ij} & \dots & \mathbf{P}_{ij} & & \\ & \vdots & \ddots & \vdots & & \\ & \mathbf{P}_{ij} & \dots & -\mathbf{P}_{ij} & & \\ & \vdots & \dots & \vdots & \ddots & \end{bmatrix} \begin{bmatrix} \vdots \\ \mathbf{u}_i \\ \vdots \\ \mathbf{u}_j \\ \vdots \end{bmatrix}.$$

Such point interactions arise in molecular or social interactions, cellular motility, and economic exchange much in the same form (1). The techniques in this homework are thus equally applicable to physical chemistry, sociology, biology or business management.

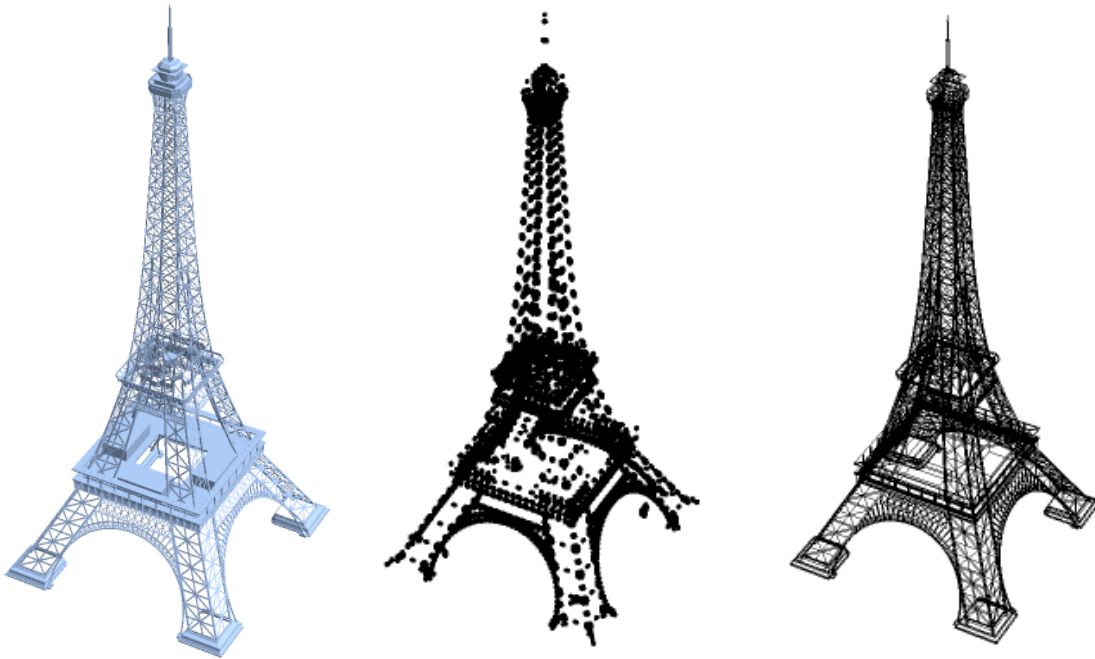


Figure 1. (Left) Eiffel Tower model. (Center) Model nodes. (Right) Model edges.

1.2 Problem data

Problem setup is carried through the following steps.

- Add required Julia packages.
- Load the data.

The following are typical instructions: open a file, load data, close the file.

```
∴ ddir = homedir()*"/courses/MATH661/data/EiffelTower";  
∴ using MAT  
∴
```

Load the model point coordinates, display the first 3 node coordinates

```
∴ dfile = matopen(ddir*"/EiffelPoints.mat");  
∴ X=read(dfile,"Expression1");  
∴ close(dfile);  
∴ NX,ndims=size(X);
```

```
∴ X[1:3,:]
```

$$\begin{bmatrix} 12.563704490661621 & -12.563704490661621 & 37.89190673828125 \\ -12.563704490661621 & -12.563704490661621 & 37.89190673828125 \\ -14.107048034667969 & -14.107048034667969 & 37.89190673828125 \end{bmatrix} \quad (2)$$

```
∴
```

Load the model edges (links between the nodes) as floating point numbers (Matlab .mat file format)

```
∴ dfile = matopen(ddir*"/EiffelLines.mat");
```

```
∴ fll=read(dfile,"Expression1");
```

```
∴ close(dfile);
```

```
∴
```

Display model size

```
∴ NL,nseg=size(fll); [NX ndims NL nseg]
```

$$[26464 \ 3 \ 31463 \ 2] \quad (3)$$

```
∴
```

The model contains $N_X = 26464$ nodes (points) each with $d=3$ coordinates and N_L edges (links) between node pairs. Transform the links to integers and display the first 4 edges.

```
∴ L=zeros(Int64,NL,nseg);
```

```
∴ for l=1:NL
```

```
    L[l,:] = floor.(Int64,fll[l,:])
```

```
end
```

```
∴ L[1:4,:]
```

$$\begin{bmatrix} 519 & 520 \\ 520 & 521 \\ 521 & 519 \\ 540 & 541 \end{bmatrix} \quad (4)$$

```
∴
```

The above states the first edge is from node $i = 519$ to node $j = 520$.

- Define the system matrix \mathbf{K} .

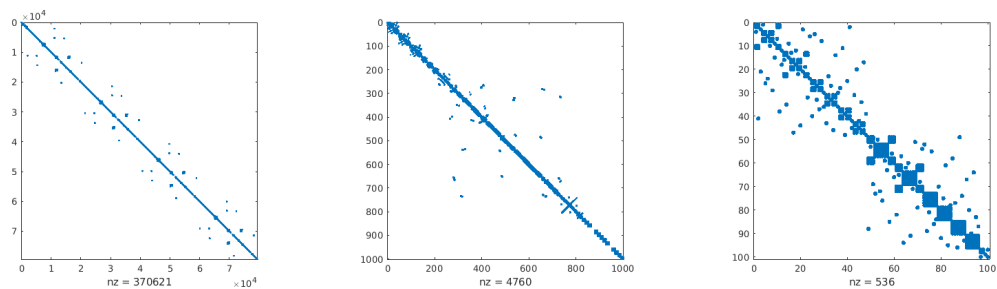


Figure 2. Structure of \mathbf{K} at various magnification factors

The data contains N_X points in \mathbb{R}^d , $d=3$, and N_L line segments defined by 2 endpoints. The total number of degrees of freedom is $m=dN_X$. Of the possible $N_X^2 \cong 7 \times 10^8$ linkages between points in the structure, only $N_L=31463$ are present.

```
∴ m=3*NX; [NX NX^2 NL m m^2]
```

$$\begin{bmatrix} 26464 & 700343296 & 31463 & 79392 & 6303089664 \end{bmatrix} \quad (5)$$

```
∴
```

The matrix \mathbf{K} can be formed by loops over the trusses as shown in the algorithm below.

Algorithm \mathbf{K} assembly

```
for k=1 to NL
  i=Lk,1; j=Lk,2
  l=Xj-Xi; l=l/||l||; P=l*lT
  p=d(i-1); q=d(j-1)
  for id=1 to d
    for jd=1 to d
      K[p+id,p+jd]=K[p+id,p+jd]-P[id,jd]
      K[p+id,q+jd]=K[p+id,q+jd]+P[id,jd]
      K[q+id,p+jd]=K[q+id,p+jd]+P[id,jd]
      K[q+id,q+jd]=K[q+id,q+jd]-P[id,jd]
    end
  end
end
```

$$\begin{bmatrix} \vdots \\ \mathbf{f}_i \\ \vdots \\ \mathbf{f}_j \\ \vdots \end{bmatrix} = \begin{bmatrix} \vdots & \dots & \vdots \\ -\mathbf{P}_{ij} & \dots & \mathbf{P}_{ij} \\ \vdots & \ddots & \vdots \\ \mathbf{P}_{ij} & \dots & -\mathbf{P}_{ij} \\ \vdots & \dots & \vdots \end{bmatrix} \begin{bmatrix} \vdots \\ \mathbf{u}_i \\ \vdots \\ \mathbf{u}_j \\ \vdots \end{bmatrix}.$$

Define a function to carry out the assembly process in d dimensions that returns a full matrix \mathbf{K} .

```
∴ function assembleK(d,L,X)
  NL = size(L)[1]; n=maximum(L); m=d*n; K=zeros(m,m)
  for k=1:NL
    i=L[k,1]; j=L[k,2]
    l=X[j,1:d]-X[i,1:d]; l=l/norm(l)
    p=d*(i-1); q=d*(j-1); P=l*l'
    for id=1:d
      for jd=1:d
        K[p+id,p+jd] = K[p+id,p+jd] - P[id,jd]
        K[p+id,q+jd] = K[p+id,q+jd] + P[id,jd]
        K[q+id,p+jd] = K[q+id,p+jd] + P[id,jd]
        K[q+id,q+jd] = K[q+id,q+jd] - P[id,jd]
      end
    end
  end
  return K
end;
```

```
∴ XB
```

$$\begin{bmatrix} 0 \\ 1 \end{bmatrix} \quad (6)$$

```
∴ LB
```

$$\begin{bmatrix} 1 & 2 \end{bmatrix} \quad (7)$$

```
∴ assembleK(1,LB,XB)
```

$$\begin{bmatrix} -1.0 & 1.0 \\ 1.0 & -1.0 \end{bmatrix} \quad (8)$$

```
∴ size(LB)
```

$$\begin{bmatrix} 1 \\ 2 \end{bmatrix} \quad (9)$$

```
∴
```

The above has been precomputed for the Eiffel Tower model. It is wasteful, and often impossible due to memory constraints, to store the entire matrix of possible couplings between points $\mathbf{K} \in \mathbb{R}^{m \times m}$ ($m^2 \cong 6.3 \times 10^9$) since most of the matrix would consist of zero entries. Rather, only the nonzero elements corresponding to the linked nodes are stored through what is known as a *sparse matrix*. There are various techniques to store sparse matrices, one of which is the *coordinate format* defined by a vector of row indices i , a vector of column indices j and a vector of values v such that the (i, j) element of the matrix \mathbf{K} is

$$k_{ij} = K(i(k), j(k)) = v(k).$$

The following instructions load the sparse representation of \mathbf{K} .

```
∴ dfile = matopen(ddir*"/KCOO.mat");
```

```
∴ flrows=read(dfile,"rows");
```

```
∴ flcols=read(dfile,"cols");
```

```
∴ vals=read(dfile,"vals");
```

```
∴ close(dfile);
```

```
∴ nz=size(flrows)[1]
```

```
370621
```

```
∴ rows=zeros(Int64,nz,1); cols=zeros(Int64,nz,1);
```

```
∴ for l=1:nz
```

```
    rows[l] = floor(Int64,flrows[l])
```

```
    cols[l] = floor(Int64,flcols[l])
```

```
end
```

```
∴ [minimum(rows) maximum(rows) minimum(cols) maximum(cols)]
```

$$[1 \ 79392 \ 1 \ 79392] \quad (10)$$

```
∴ using SparseArrays
```

```
∴ K=sparse(rows[:,1],cols[:,1],vals[:,1]);
```

```
∴ size(K)
```

$$\begin{bmatrix} 79392 \\ 79392 \end{bmatrix} \quad (11)$$

1.3 Utility routines

- Draw the deformed structure in figure n_f , given node coordinates $\mathbf{X} \in \mathbb{R}^{N_x \times d}$, displacements $\mathbf{u} \in \mathbb{R}^m$, $m = dN_x$, skipping k_s nodes (to reduce drawing time).
- Draw the deformed structure in figure n_f , given node coordinates $\mathbf{X} \in \mathbb{R}^{N_x \times d}$, displacements $\mathbf{U} \in \mathbb{R}^{N_x \times d}$, $m = dN_x$, skipping k_s nodes.

1.4 Simplified “A” model

The full Eiffel Tower model is quite complex. It is convenient to build familiarity with modeling binary interactions using the much simpler two-dimensional ($d=2$) 8-node model shown in Fig. 3

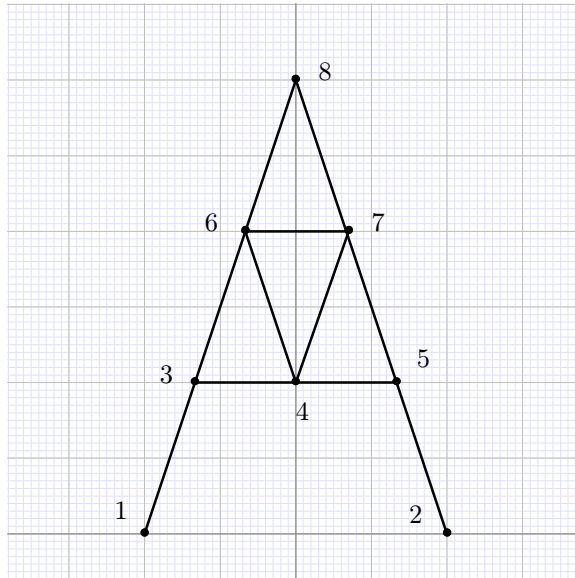


Figure 3. "A" model of Eiffel tower

- Define the data for this model and compute K .

```
∴ x5 = (2.0/3.0)*2; x7 = (1.0/3.0)*2; [x5 x7]
```

$$\left[1.3333333333333333 \quad 0.6666666666666666 \right] \quad (12)$$

```
∴ X=[-2 0; 2 0; -x5 2; 0 2; x5 2; -x7 4; x7 4; 0 6]
```

$$\begin{bmatrix} -2.0 & 0.0 \\ 2.0 & 0.0 \\ -1.3333333333333333 & 2.0 \\ 0.0 & 2.0 \\ 1.3333333333333333 & 2.0 \\ -0.6666666666666666 & 4.0 \\ 0.6666666666666666 & 4.0 \\ 0.0 & 6.0 \end{bmatrix} \quad (13)$$

```
∴ L=[ 1 3; 3 6; 6 8; 8 7; 7 5; 5 2; 6 7; 3 4; 4 5; 6 4; 4 7]
```

$$\begin{bmatrix} 1 & 3 \\ 3 & 6 \\ 6 & 8 \\ 8 & 7 \\ 7 & 5 \\ 5 & 2 \\ 6 & 7 \\ 3 & 4 \\ 4 & 5 \\ 6 & 4 \\ 4 & 7 \end{bmatrix} \quad (14)$$

```
∴ KA=assembleK(2,L,X);
```

```
∴
```

1.5 Enforcing boundary conditions

- When structures are attached to their environment, boundary conditions have to be enforced in the linear system $\mathbf{f} = \mathbf{K}\mathbf{u}$. A common technique is set the diagonal element of \mathbf{K} for the node i subjected to boundary conditions to a large value B such that it effectively decouples from the system, and set $f_i = Bg_i$ where g_i is the desired boundary value.

2 Track 1 and 2 common problems

1. Construct $\mathbf{X}_A, \mathbf{L}_A, \mathbf{K}_A$ (node coordinates, edges, stiffness matrix) for the “A” model. For this small model a sparse representation of \mathbf{K}_A is not required.
2. Start your investigation of the system by finding the $n = 10$ largest singular values and eigenvalues of \mathbf{K}, \mathbf{K}_A . Plot the first $p = 3$ singular modes and eigenmodes.
3. Find the linear combination \mathbf{s} of the first $n = 10$ singular modes that best approximates a force $\mathbf{f} = (z \cos \theta, z \sin \theta, 0)$ on the structure by solving the problem

$$\min_{\mathbf{s} \in \mathbb{R}^n} \|\mathbf{f} - \mathbf{U}\mathbf{s}\|$$

Randomly choose θ . This is the response of the structure to wind along direction θ . Form and solve the analogous problem for the “A” model first.

4. Find the linear combination \mathbf{s} of the first $n = 10$ eigenmodes modes that best approximates a force $\mathbf{f} = (z \cos \theta, z \sin \theta, 0)$ on the structures. Form and solve the analogous problem for the “A” model first.
5. Replace singular modes or eigenmodes by orthogonal \mathbf{Q} , with $\mathbf{Q}\mathbf{R} = [\mathbf{f} \mathbf{K}\mathbf{f} \mathbf{K}^2\mathbf{f} \dots \mathbf{K}^9\mathbf{f}]$ (known as Krylov modes). Find the linear combination \mathbf{w} of the first 10 Krylov modes that best approximates a force $\mathbf{f} = (z \cos \theta, z \sin \theta, 0)$ on the structure. Form and solve the analogous problem for the “A” model first.

3 Track 2 task

Carry out a bibliographic search for review papers on reduced order modeling. Choose a paper you find appealing and draft a two-paragraph summary. Compare the approach you find in the literature to problems 3 to 5 above.