

Algorithm for frequency analysis



UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH

Escola Superior d'Enginyeries Industrial,
Aeroespacial i Audiovisual de Terrassa

Algorithm for frequency analysis

1) Input data:

1.1 Global system matrices:

- Stiffness matrix: $[\mathbf{K}]$
- Mass matrix: $[\mathbf{M}]$

Tip: For increased performance in Matlab, initialize $[\mathbf{K}]$ and $[\mathbf{M}]$ as **sparse** matrices before the assembly (i.e., use the **sparse** function instead of the **zeros** function):

```
K = sparse(Ndof,Ndof);  
M = sparse(Ndof,Ndof);
```

1.2 Global forces vector:

- External forces amplitudes:

$$[\mathbf{F}] = [\cdots \quad \hat{\mathbf{F}}_k \quad \cdots] \quad (\text{as many rows as DOFs of the problem})$$

- Frequencies:

$$\{\boldsymbol{\omega}\} = \{\cdots \quad \omega_k \quad \cdots\}$$

1.3 Boundary conditions:

- Fix nodes and DOFs matrix:

$$[\mathbf{U}_p] = \begin{bmatrix} u^{(p)} & \vdots & n^{(p)} & j^{(p)} \\ & & & \\ & & & \\ & & & \\ & & & \vdots \end{bmatrix} \left. \vphantom{\begin{bmatrix} u^{(p)} & \vdots & n^{(p)} & j^{(p)} \\ & & & \\ & & & \\ & & & \vdots \end{bmatrix}} \right\} \begin{array}{l} \text{Number of} \\ \text{prescribed DOFs} \end{array}$$

$u^{(p)}$: value of prescribed displ./rot. (p)

$n^{(p)}$: global node # assigned to (p)

$j^{(p)}$: degree of freedom assigned to (p)

Algorithm for frequency analysis

2) Boundary conditions:

2.1 Initialization:

$$[\mathbf{U}] = [\mathbf{0}]_{N_{\text{dof}} \times N_{\omega}} \quad (N_{\omega} \text{ is the total number of frequencies})$$

2.2 Prescribed and free DOFs:

For each row p in $[\mathbf{U}_p]$

$$\mathbf{I}_p(p) = 6(\mathbf{U}_p(p, 2) - 1) + \mathbf{U}_p(p, 3) \quad (\text{vector with prescribed degrees of freedom})$$

$$\mathbf{U}(\mathbf{I}_p(p), :) = \mathbf{U}_p(p, 1)$$

End loop over rows in $[\mathbf{U}_p]$

$$\mathbf{I}_f = \{1: N_{\text{dof}}\} - \{\mathbf{I}_p\} \quad (\text{Tip: in Matlab, this operation can be done with the } \mathbf{setdiff} \text{ function})$$

$$\mathbf{I}_f = \mathbf{setdiff}(1:\text{Ndof}, \mathbf{I}_p)$$

3) Solve system of equations:

3.1 Solve system:

For each column k in $[\mathbf{F}]$

$$\mathbf{U}(\mathbf{I}_f, k) = [\mathbf{K}(\mathbf{I}_f, \mathbf{I}_f) - \omega(k)^2 \mathbf{M}(\mathbf{I}_f, \mathbf{I}_f)]^{-1} (\mathbf{F}(\mathbf{I}_f, k) - [\mathbf{K}(\mathbf{I}_f, \mathbf{I}_p) - \omega(k)^2 \mathbf{M}(\mathbf{I}_f, \mathbf{I}_p)] \{\mathbf{U}(\mathbf{I}_p, k)\})$$

End loop over columns in $[\mathbf{F}]$

Algorithm for frequency analysis

4) Modal analysis:

4.1 Solve eigenvalues and eigenvectors problem. In Matlab, for **sparse** stiffness and mass matrices, use the following command to obtain the first N_m eigenvectors (\mathbf{V}) associated to the smallest eigenvalues (\mathbf{D}) in magnitude:

```
[V,D] = eigs(K(I_f,I_f),M(I_f,I_f),N_m,'sm');
```

Note: For non-sparse (full) matrices, use eig function instead:

```
[V,D] = eig(K(I_f,I_f),M(I_f,I_f));
```

Tip: Make sure $[\mathbf{K}]$ and $[\mathbf{M}]$ are symmetric before using eigs/eig functions:

$$[\mathbf{K}] = ([\mathbf{K}] + [\mathbf{K}]^T)/2$$

$$[\mathbf{M}] = ([\mathbf{M}] + [\mathbf{M}]^T)/2$$

4.2 Obtain squared natural frequencies (λ_k) and vibration modes ($\hat{\phi}_k$):

$$[\Phi] = [\mathbf{0}]_{N_{\text{dof}} \times N_m}$$

$$\{\lambda\} = \{\mathbf{0}\}_{1 \times N_m}$$

For each column k in $[\mathbf{V}]$

$$\Phi(I_f, k) = \mathbf{V}(:, k) / \sqrt{\{\mathbf{V}(:, k)\}^T [\mathbf{M}(I_f, I_f)] \{\mathbf{V}(:, k)\}} \quad (\text{mass-normalized vibration mode})$$

$$\lambda(k) = \mathbf{D}(k, k) \quad (\equiv \omega^2 \text{ squared natural frequency})$$

End loop over columns in $[\mathbf{V}]$

Algorithm for frequency analysis

5) Model-order reduction:

5.1 Select set of modes to project the system:

$\{I_m\} = \{\dots\}$ (I_m can be any set of numbers between $[1, N_m]$, including all of them)

5.2 Project the system onto the selected set of modes:

$[U^*] = [0]_{N_{\text{dof}} \times N_\omega}$ (approximate solution vector)

For each column k in $[F]$

For each mode j in $\{I_m\}$

$$\alpha(j, k) = \{\Phi(:, I_m(j))\}^T \{F(:, k)\} / (\lambda(j) - \omega(k)^2) \quad (\text{modal amplitude})$$

$$U^*(:, k) = U^*(:, k) + \Phi(:, I_m(j)) \alpha(j, k)$$

End loop over modes

End loop over columns in $[F]$

Note: The model-order reduction procedure is also applicable to static problems, just by setting $\omega(k) = 0$ and making $\{F(:, k)\}$ the corresponding (static) force vector.