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:

$$[\mathbf{F}] = [\cdots \quad \hat{\boldsymbol{F}}_k \quad \cdots]$$ (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:

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

$u^{(p)}$: 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:

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

2.2 Prescribed and free DOFs:

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

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

$\mathbf{U}\big(\boldsymbol{I}_{\text{p}}(p),:\big) = \mathbf{U}_{\text{p}}(p,1)$

End loop over rows in $[\mathbf{U}_{\text{p}}]$

$\boldsymbol{I}_{\text{f}} = \{1\!:N_{\text{dof}}\} - \{\boldsymbol{I}_{\text{p}}\}$ (**Tip**: in Matlab, this operation can be done with the **`setdiff`** function)

```
If = setdiff(1:Ndof,Ip)
```

3) Solve system of equations:

3.1 Solve system:

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

$$\mathbf{U}(\boldsymbol{I}_{\text{f}},k) = [\mathbf{K}(\boldsymbol{I}_{\text{f}},\boldsymbol{I}_{\text{f}}) - \omega(k)^2\mathbf{M}(\boldsymbol{I}_{\text{f}},\boldsymbol{I}_{\text{f}})]^{-1}\big(\mathbf{F}(\boldsymbol{I}_{\text{f}},k) - \big[\mathbf{K}\big(\boldsymbol{I}_{\text{f}},\boldsymbol{I}_{\text{p}}\big) - \omega(k)^2\mathbf{M}(\boldsymbol{I}_{\text{f}},\boldsymbol{I}_{\text{f}})\big]\big\{\mathbf{U}\big(\boldsymbol{I}_{\text{p}},k\big)\big\}\big)$$

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 **`Nm`** eigenvectors (**`V`**) associated to the smallest eigenvalues (**`D`**) in magnitude:

```
[V,D] = eigs(K(If,If),M(If,If),Nm,'sm');
```

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

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

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

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

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

$$[\mathbf{\Phi}] = [\mathbf{0}]_{N_{\mathrm{dof}} \times N_{\mathrm{m}}}$$

$$\{\boldsymbol{\lambda}\} = \{\mathbf{0}\}_{1 \times N_{\mathrm{m}}}$$

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

$$\mathbf{\Phi}(\boldsymbol{I}_{\mathrm{f}}, k) = \mathbf{V}(:, k)/\mathrm{sqrt}\left(\{\mathbf{V}(:, k)\}^{\mathrm{T}}[\mathbf{M}(\boldsymbol{I}_{\mathrm{f}}, \boldsymbol{I}_{\mathrm{f}})]\{\mathbf{V}(:, k)\}\right) \quad \text{(mass-normalized vibration mode)}$$

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

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:

$\{\boldsymbol{I}_{\mathrm{m}}\} = \{\cdots\}$ ($\boldsymbol{I}_{\mathrm{m}}$ can be any set of numbers between $[1, N_{\mathrm{m}}]$, including all of them)

5.2 Project the system onto the selected set of modes:

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

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

For each mode j in $\{\boldsymbol{I}_{\mathrm{m}}\}$

$\boldsymbol{\alpha}(j,k) = \left\{\boldsymbol{\Phi}\left(:, \boldsymbol{I}_{\mathrm{m}}(j)\right)\right\}^{\mathrm{T}}\{\mathbf{F}(:,k)\}/(\lambda(j) - \omega(k)^2)$ (modal amplitude)

$\mathbf{U}^*(:,k) = \mathbf{U}^*(:,k) + \boldsymbol{\Phi}\left(:, \boldsymbol{I}_{\mathrm{m}}(j)\right)\boldsymbol{\alpha}(j,k)$

End loop over modes

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

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