Derived Ritz Vectors, Numerical Integration Multiple Support Excitation

Giacomo Boffi

http://intranet.dica.polimi.it/people/boffi-giacomo

Dipartimento di Ingegneria Civile Ambientale e Territoriale
Politecnico di Milano

April 19, 2016

Derived Ritz Vectors

Introduction

The dynamic analysis of a linear structure can be described as a three steps procedure

1. *FEM* model discretization of the structure,

Introduction

The dynamic analysis of a linear structure can be described as a three steps procedure

1. *FEM* model discretization of the structure,
2. solution of the eigenproblem,

Introduction

The dynamic analysis of a linear structure can be described as a three steps procedure

1. *FEM* model discretization of the structure,
2. solution of the eigenproblem,
3. integration of the uncoupled equations of motion.

Introduction

The dynamic analysis of a linear structure can be described as a three steps procedure

1. *FEM* model discretization of the structure,
2. solution of the eigenproblem,
3. integration of the uncoupled equations of motion.

The eigenproblem solution is often obtained by some variation of the Rayleigh-Ritz procedure, e.g. subspace iteration that is efficient and accurate.

Introduction

The dynamic analysis of a linear structure can be described as a three steps procedure

1. *FEM* model discretization of the structure,
2. solution of the eigenproblem,
3. integration of the uncoupled equations of motion.

The eigenproblem solution is often obtained by some variation of the Rayleigh-Ritz procedure, e.g. subspace iteration that is efficient and accurate.
A proper choice of the initial Ritz base $\mathbf{\Phi}_0$ is key to efficiency. An effective reduced base is given by the so called Derived Ritz vectors (or Lanczos vectors)
DRV not only form a suitable base for subspace iteration, but can be directly used in a step-by-step procedure.

Lanczos Vectors

The Lanczos vectors are obtained in a manner that is similar to matrix iteration and are constructed in such a way that each one is orthogonal to all the others.
Usually each new vector must be orthogonalised with respect to all the other vectors, lots of work...

Lanczos Vectors

The Lanczos vectors are obtained in a manner that is similar to matrix iteration and are constructed in such a way that each one is orthogonal to all the others.
Usually each new vector must be orthogonalised with respect to all the other vectors, lots of work...

For the Lanczos vectors sequence, orthogonalising a new vector with respect to the two preceeding ones ensures that the new vector is orthogonal to *all* the previous vectors.

Computing the 1st DRV

Our initial assumption is that the load vector can be decoupled, $\mathbf{p}(x,t) = \mathbf{r}_0\, f(t)$

1. Obtain the deflected shape $\boldsymbol{\ell}_1$ due to the application of the force shape vector (ℓ's are displacements).

$$\mathbf{K}\,\boldsymbol{\ell}_1 = \mathbf{r}$$

2. Compute the normalisation factor with respect to the mass matrix (β is a displacement).

$$\beta_1^2 = \frac{\boldsymbol{\ell}_1^T \mathbf{M}\,\boldsymbol{\ell}_1}{1 \text{ unit mass}}$$

3. Obtain the first derived Ritz vector normalising $\boldsymbol{\ell}_1$ such that $\boldsymbol{\phi}_1^T \mathbf{M}\,\boldsymbol{\phi} = 1$ unit of mass (ϕ's are adimensional).

$$\boldsymbol{\phi}_1 = \frac{1}{\beta_1}\boldsymbol{\ell}_1$$

Computing the 2nd DRV

A new load vector is computed, $\mathbf{r}_1 = 1\mathbf{M}\,\boldsymbol{\phi}_1$, where 1 is a unit acceleration.

1. Obtain the deflected shape $\boldsymbol{\ell}_2$ due to the application of the new load vector.

$$\mathbf{K}\,\boldsymbol{\ell}_2 = \mathbf{r}_1$$

2. Purify the displacements $\boldsymbol{\ell}_2$ (α_1 is dimensionally a displacement).

$$\alpha_1 = \frac{\boldsymbol{\phi}_1^T\mathbf{M}\,\boldsymbol{\ell}_2}{1 \text{ unit mass}}$$

$$\hat{\boldsymbol{\ell}}_2 = \boldsymbol{\ell}_2 - \alpha_1\boldsymbol{\phi}_1$$

3. Compute the normalisation factor.

$$\beta_2^2 = \frac{\hat{\boldsymbol{\ell}}_2^T\mathbf{M}\,\hat{\boldsymbol{\ell}}_2}{1 \text{ unit mass}}$$

4. Obtain the second derived Ritz vector normalising $\hat{\boldsymbol{\ell}}_2$.

$$\boldsymbol{\phi}_2 = \frac{1}{\beta_2}\hat{\boldsymbol{\ell}}_2$$

Computing the 3rd DRV

The new load vector is $\mathbf{r}_2 = 1\mathbf{M}\,\boldsymbol{\phi}_2$, 1 being a unit acceleration.

1. Obtain the deflected shape $\boldsymbol{\ell}_3$. $\quad \mathbf{K}\,\boldsymbol{\ell}_3 = \mathbf{r}_2$
2. Purify the displacements $\boldsymbol{\ell}_3$ where $\quad \hat{\boldsymbol{\ell}}_3 = \boldsymbol{\ell}_3 - \alpha_2\boldsymbol{\phi}_2 - \beta_2\boldsymbol{\phi}_1$

$$\alpha_2 = \frac{\boldsymbol{\phi}_2^T\mathbf{M}\,\boldsymbol{\ell}_3}{1\text{ unit mass}},\quad \alpha_1 = \frac{\boldsymbol{\phi}_1^T\mathbf{M}\,\boldsymbol{\ell}_3}{1\text{ unit mass}} = \beta_2$$

3. Compute the normalisation factor. $\quad \beta_3^2 = \frac{\hat{\boldsymbol{\ell}}_3^T\mathbf{M}\,\hat{\boldsymbol{\ell}}_3}{1\text{ unit mass}}$
4. Obtain the third derived Ritz vector normalising $\hat{\boldsymbol{\ell}}_3$. $\quad \boldsymbol{\phi}_3 = \frac{1}{\beta_2}\hat{\boldsymbol{\ell}}_3$

We don't need to compute α_1 to purify $\boldsymbol{\ell}_3$, because it's equal to β_2, i.e., the normalization factor applied in the previous (second) step.

Computing the 3rd DRV

The new load vector is $\mathbf{r}_2 = 1\mathbf{M}\,\boldsymbol{\phi}_2$, 1 being a unit acceleration.

1. Obtain the deflected shape ℓ_3. $\qquad \mathbf{K}\,\ell_3 = \mathbf{r}_2$
2. Purify the displacements ℓ_3 where $\qquad \hat{\ell}_3 = \ell_3 - \alpha_2\boldsymbol{\phi}_2 - \beta_2\boldsymbol{\phi}_1$

$$\alpha_2 = \frac{\boldsymbol{\phi}_2^T\mathbf{M}\,\ell_3}{1 \text{ unit mass}}, \quad \alpha_1 = \frac{\boldsymbol{\phi}_1^T\mathbf{M}\,\ell_3}{1 \text{ unit mass}} = \beta_2$$

3. Compute the normalisation factor. $\qquad \beta_3^2 = \frac{\hat{\ell}_3^T\mathbf{M}\,\hat{\ell}_3}{1 \text{ unit mass}}$
4. Obtain the third derived Ritz vector normalising $\hat{\ell}_3$. $\qquad \boldsymbol{\phi}_3 = \frac{1}{\beta_2}\hat{\ell}_3$

We don't need to compute α_1 to purify ℓ_3, because it's equal to β_2, i.e., the normalization factor applied in the previous (second) step.

Fourth Vector, etc

The new load vector is $\mathbf{r}_3 = 1\mathbf{M}\,\boldsymbol{\phi}_3$, 1 being a unit acceleration.

1. Obtain the deflected shape $\boldsymbol{\ell}_4$. $\quad \mathbf{K}\,\boldsymbol{\ell}_4 = \mathbf{r}_3$
2. Purify the displacements $\boldsymbol{\ell}_4$ where $\quad \hat{\boldsymbol{\ell}}_4 = \boldsymbol{\ell}_4 - \alpha_3\boldsymbol{\phi}_3 - \beta_3\boldsymbol{\phi}_2$

$$\alpha_3 = \frac{\boldsymbol{\phi}_3^T\,\mathbf{M}\,\boldsymbol{\ell}_4}{1\text{ unit mass}},\quad \alpha_2 = \frac{\boldsymbol{\phi}_2^T\,\mathbf{M}\,\boldsymbol{\ell}_4}{1\text{ unit mass}} = \beta_3$$

$$\alpha_1 = \frac{\boldsymbol{\phi}_1^T\,\mathbf{M}\,\boldsymbol{\ell}_4}{1\text{ unit mass}} = 0$$

3. Compute the normalisation factor. $\quad \beta_4 = \frac{\hat{\boldsymbol{\ell}}_4^T\,\mathbf{M}\,\hat{\boldsymbol{\ell}}_4}{1\text{ unit mass}}$
4. Obtain the fourth derived Ritz vector normalising $\hat{\boldsymbol{\ell}}_4$. $\quad \boldsymbol{\phi}_4 = \frac{1}{\beta_4}\hat{\boldsymbol{\ell}}_4$

The procedure used for the fourth *DRV* can be used for all the subsequent $\boldsymbol{\phi}_i$, with $\alpha_{i-1} = \boldsymbol{\phi}_{i-1}^T\mathbf{M}\,\boldsymbol{\ell}_i$ and $\alpha_{i-2} \equiv \beta_{i-1}$, while all the others purifying coefficents are equal to zero, $\alpha_{i-3} = \cdots = 0$.

The Tridiagonal Matrix

Having computed $M < N$ *DRV* we can write for, e.g., $M = 5$ that each un-normalised vector is equal to the displacements minus the purification terms

$$\begin{aligned}
\boldsymbol{\phi}_2\beta_2 &= \mathbf{K}^{-1}\mathbf{M}\,\boldsymbol{\phi}_1 - \boldsymbol{\phi}_1\alpha_1\\
\boldsymbol{\phi}_3\beta_3 &= \mathbf{K}^{-1}\mathbf{M}\,\boldsymbol{\phi}_2 - \boldsymbol{\phi}_2\alpha_2 - \boldsymbol{\phi}_1\beta_2\\
\boldsymbol{\phi}_4\beta_4 &= \mathbf{K}^{-1}\mathbf{M}\,\boldsymbol{\phi}_3 - \boldsymbol{\phi}_3\alpha_3 - \boldsymbol{\phi}_2\beta_3\\
\boldsymbol{\phi}_5\beta_5 &= \mathbf{K}^{-1}\mathbf{M}\,\boldsymbol{\phi}_4 - \boldsymbol{\phi}_4\alpha_4 - \boldsymbol{\phi}_3\beta_4
\end{aligned}$$

Collecting the $\boldsymbol{\phi}$ in a matrix $\boldsymbol{\Phi}$, the above can be written

$$\mathbf{K}^{-1}\mathbf{M}\,\boldsymbol{\Phi} = \boldsymbol{\Phi}\begin{bmatrix}\alpha_1 & \beta_2 & 0 & 0 & 0\\ \beta_2 & \alpha_2 & \beta_3 & 0 & 0\\ 0 & \beta_3 & \alpha_3 & \beta_4 & 0\\ 0 & 0 & \beta_4 & \alpha_4 & \beta_5\\ 0 & 0 & 0 & \beta_5 & \alpha_5\end{bmatrix} = \boldsymbol{\Phi}\mathbf{T}$$

where we have introduce $\mathbf{T}$, a symmetric, tridiagonal matrix where $t_{i,i} = \alpha_i$ and $t_{i,i+1} = t_{i+1,i} = \beta_{i+1}$.
Premultiplying by $\boldsymbol{\Phi}^T\mathbf{M}$

$$\boldsymbol{\Phi}^T\mathbf{M}\,\mathbf{K}^{-1}\mathbf{M}\,\boldsymbol{\Phi} = \underbrace{\boldsymbol{\Phi}^T\mathbf{M}\,\boldsymbol{\Phi}}_{\mathbf{I}}\,\mathbf{T} = \mathbf{T}$$

.

Eigenvectors

Write the unknown in terms of the reduced base $\mathbf{\Phi}$ and a vector of Ritz coordinates z, substitute in the undamped eigenvector equation, premultiply by $\mathbf{\Phi}^\mathsf{T}\mathbf{M}\,\mathbf{K}^{-1}$ and apply the semi-orthogonality relationship written in the previous slide.

1. $\omega^2\mathbf{M}\,\mathbf{\Phi}\,z = \mathbf{K}\,\mathbf{\Phi}\,z.$
2. $\omega^2\underbrace{\mathbf{\Phi}^\mathsf{T}\mathbf{M}\,\mathbf{K}^{-1}\mathbf{M}\,\mathbf{\Phi}}_{\mathbf{T}}\,z = \underbrace{\mathbf{\Phi}^\mathsf{T}\mathbf{M}\,\underbrace{\mathbf{K}^{-1}\mathbf{K}}_{\mathbf{I}}\,\mathbf{\Phi}}_{\mathbf{I}}\,z.$
3. $\omega^2\mathbf{T}\,z = \mathbf{I}\,z \quad\Rightarrow\quad \omega^2\mathbf{T}\,z = z.$

Due to the tridiagonal structure of $\mathbf{T}$, the approximate eigenvalues can be computed with very small computational effort.

Direct Integration

Write the equation of motion for a Rayleigh damped system, with $p(x,t) = \mathbf{r}\, f(t)$ in terms of the *DRV*'s and Ritz coordinates $\boldsymbol{z}$

$$\mathbf{M\Phi}\ddot{\boldsymbol{z}} + c_0\mathbf{M\Phi}\dot{\boldsymbol{z}} + c_1\mathbf{K\Phi}\dot{\boldsymbol{z}} + \mathbf{K\Phi}\boldsymbol{z} = \mathbf{r}\, f(t)$$

premultiplying by $\mathbf{\Phi}^T\mathbf{M}\,\mathbf{K}^{-1}$, substituting $\mathbf{T}$ and $\mathbf{I}$ where appropriate, doing a series of substitutions on the right member

$$\begin{aligned}\mathbf{T}(\ddot{\boldsymbol{z}} + c_0\dot{\boldsymbol{z}}) + \mathbf{I}(c_1\dot{\boldsymbol{z}} + \boldsymbol{z}) &= \mathbf{\Phi}^T\mathbf{M}\,\mathbf{K}^{-1}\mathbf{r}\, f(t)\\ &= \mathbf{\Phi}^T\mathbf{M}\boldsymbol{\ell}_1\, f(t)\\ &= \mathbf{\Phi}^T\mathbf{M}\beta_1\boldsymbol{\phi}_1\, f(t)\\ &= \beta_1\begin{Bmatrix}1 & 0 & 0 & \cdots & 0 & 0\end{Bmatrix}^T f(t).\end{aligned}$$

Using the *DRV*'s as a Ritz base, we have a set of *mildly coupled* differential equations, where external loadings directly excite the first *mode* only, and all the other *modes* are excited by inertial coupling only, with rapidly diminishing effects.

Modal Superposition or Direct Integration?

Static effects being fully taken into account by the response of the first *DRV*, only a few *DRV*'s are needed in direct integration of the equation of motion.
Furthermore special algorithms were devised for the integration of the *tridiagonal equations of motion*, that aggravate computational effort by $\approx$ 40% only with respect to the integration of uncoupled equations.

Modal Superposition or Direct Integration?

Static effects being fully taken into account by the response of the first *DRV*, only a few *DRV*'s are needed in direct integration of the equation of motion.
Furthermore special algorithms were devised for the integration of the *tridiagonal equations of motion*, that aggravate computational effort by $\approx$ 40% only with respect to the integration of uncoupled equations.
Direct integration in Ritz coordinate is the best choice when the loading shape is complex and the loading duration is relatively short.

Modal Superposition or Direct Integration?

Static effects being fully taken into account by the response of the first *DRV*, only a few *DRV*'s are needed in direct integration of the equation of motion.
Furthermore special algorithms were devised for the integration of the *tridiagonal equations of motion*, that aggravate computational effort by $\approx$ 40% only with respect to the integration of uncoupled equations.
Direct integration in Ritz coordinate is the best choice when the loading shape is complex and the loading duration is relatively short.
On the other hand, in applications of earthquake engineering the loading shape is well behaved and the duration is significantly longer, so that the savings in integrating the uncoupled equations of motion outbalance the cost of the eigenvalue extraction.

Re-Orthogonalisation

Denoting with $\mathbf{\Phi}_i$ the i columns matrix that collects the *DRV*'s computed, we define an ortogonality test vector

$$\boldsymbol{w}_i = \boldsymbol{\phi}_{i+1}^T \mathbf{M}\, \mathbf{\Phi}_i = \begin{Bmatrix} w_1 & w_2 & \ldots & w_{i-1} & w_i \end{Bmatrix}$$

that expresses the orthogonality of the newly computed vector with respect to the previous ones.
When one of the components of $\boldsymbol{w}_i$ exceeds a given tolerance, the non-exactly orthogonal $\boldsymbol{\phi}_{i+1}$ must be subjected to a Gram-Schmidt orthogonalisation with respect to all the preceding *DRV*'s.

Required Number of DRV

Analogously to the modal partecipation factor the Ritz partecipation factor $\hat{\Gamma}_i$ is defined

$$\hat{\Gamma}_i = \frac{\boldsymbol{\phi}_i^T \mathbf{r}}{\underbrace{\boldsymbol{\phi}_i^T \mathbf{M}\, \boldsymbol{\phi}_i}_{1}} = \boldsymbol{\phi}_i^T \mathbf{r}$$

(note that we divided by a unit mass).
The loading shape can be expressed as a linear combination of Ritz vector inertial forces,

$$\mathbf{r} = \sum \hat{\Gamma}_i \mathbf{M}\, \boldsymbol{\phi}_i.$$

The number of computed *DRV*'s can be assumed sufficient when $\hat{\Gamma}_i$ falls below an assigned value.

Required Number of DRV

Another way to proceed: define an error vector

$$\hat{e}_i = \mathbf{r} - \sum_{j=1}^{i} \hat{\Gamma}_j \mathbf{M}\, \boldsymbol{\phi}_j$$

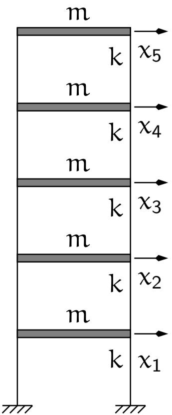
and an error norm

$$|\hat{e}_i| = \frac{\mathbf{r}^T \hat{\mathbf{e}}_i}{\mathbf{r}^T \mathbf{r}},$$

and stop at $\boldsymbol{\phi}_i$ when the error norm falls below a given value.
BTW, an error norm can be defined for modal analysis too.
Assuming normalized eigenvectors,

$$\mathbf{e}_i = \mathbf{r} - \sum_{j=1}^{i} \Gamma_j \mathbf{M}\, \boldsymbol{\phi}_j, \qquad |e_i| = \frac{\mathbf{r}^T \mathbf{e}_i}{\mathbf{r}^T \mathbf{r}}$$

Error Norms, modes



In this example, we compare the error norms using modal forces and *DRV* forces to approximate 3 different loading shapes.
The building model, on the left, used in this example is the same that we already used in different examples.

The structural matrices are $M = m \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$,

$K = k \begin{bmatrix} 2 & -1 & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 \\ 0 & 0 & -1 & 2 & -1 \\ 0 & 0 & 0 & -1 & 1 \end{bmatrix}$, $F = \frac{1}{k} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 2 & 2 & 2 & 2 \\ 1 & 2 & 3 & 3 & 3 \\ 1 & 2 & 3 & 4 & 4 \\ 1 & 2 & 3 & 4 & 5 \end{bmatrix}$.

Eigenvalues and eigenvectors matrices are:

$$\boldsymbol{\Lambda} = \begin{bmatrix} 0.0810 & 0.0000 & 0.0000 & 0.0000 & 0.0000 \\ 0.0000 & 0.6903 & 0.0000 & 0.0000 & 0.0000 \\ 0.0000 & 0.0000 & 1.7154 & 0.0000 & 0.0000 \\ 0.0000 & 0.0000 & 0.0000 & 2.8308 & 0.0000 \\ 0.0000 & 0.0000 & 0.0000 & 0.0000 & 3.6825 \end{bmatrix},$$

$$\boldsymbol{\Psi} = \begin{bmatrix} +0.1699 & -0.4557 & +0.5969 & +0.5485 & -0.3260 \\ +0.3260 & -0.5969 & +0.1699 & -0.4557 & +0.5485 \\ +0.4557 & -0.3260 & -0.5485 & -0.1699 & -0.5969 \\ +0.5485 & +0.1699 & -0.3260 & +0.5969 & +0.4557 \\ +0.5969 & +0.5485 & +0.4557 & -0.3260 & -0.1699 \end{bmatrix}$$

Error Norms, DRVs

The *DRV*'s are computed for three different shapes of force vectors,

$$\mathbf{r}_{(1)} = \begin{Bmatrix}0 & 0 & 0 & 0 & +1\end{Bmatrix}^T$$
$$\mathbf{r}_{(2)} = \begin{Bmatrix}0 & 0 & 0 & -2 & 1\end{Bmatrix}^T$$
$$\mathbf{r}_{(3)} = \begin{Bmatrix}1 & 1 & 1 & 1 & +1\end{Bmatrix}^T.$$

For the three force shapes, we have of course different sets of *DRV*'s

$$\mathbf{\Phi}_{(1)} = \begin{bmatrix} +0.1348 & +0.3023 & +0.4529 & +0.5679 & +0.6023 \\ +0.2697 & +0.4966 & +0.4529 & +0.0406 & -0.6884 \\ +0.4045 & +0.4750 & -0.1132 & -0.6693 & +0.3872 \\ +0.5394 & +0.1296 & -0.6794 & +0.4665 & -0.1147 \\ +0.6742 & -0.6478 & +0.3397 & -0.1014 & +0.0143 \end{bmatrix},$$

$$\mathbf{\Phi}_{(2)} = \begin{bmatrix} -0.1601 & -0.0843 & +0.2442 & +0.6442 & +0.7019 \\ -0.3203 & -0.0773 & +0.5199 & +0.4317 & -0.6594 \\ -0.4804 & +0.1125 & +0.5627 & -0.6077 & +0.2659 \\ -0.6405 & +0.5764 & -0.4841 & +0.1461 & -0.0425 \\ -0.4804 & -0.8013 & -0.3451 & -0.0897 & -0.0035 \end{bmatrix},$$

$$\mathbf{\Phi}_{(3)} = \begin{bmatrix} +0.1930 & -0.6195 & +0.6779 & -0.3385 & +0.0694 \\ +0.3474 & -0.5552 & -0.2489 & +0.6604 & -0.2701 \\ +0.4633 & -0.1805 & -0.5363 & -0.3609 & +0.5787 \\ +0.5405 & +0.2248 & -0.0821 & -0.4103 & -0.6945 \\ +0.5791 & +0.4742 & +0.4291 & +0.3882 & +0.3241 \end{bmatrix}.$$

Error Norm, comparison

	Error Norm					
	Forces $\mathbf{r}_{(1)}$		Forces $\mathbf{r}_{(2)}$		Forces $\mathbf{r}_{(3)}$	
	modes	*DRV*	modes	*DRV*	modes	*DRV*
1	0.643728	0.545454	0.949965	0.871794	0.120470	0.098360
2	0.342844	0.125874	0.941250	0.108156	0.033292	0.012244
3	0.135151	0.010489	0.695818	0.030495	0.009076	0.000757
4	0.028863	0.000205	0.233867	0.001329	0.001567	0.000011
5	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000

Reduced Eigenproblem using DRV base

Using the same structure as in the previous example, we want to compute the first 3 eigenpairs using the first 3 *DRV*'s computed for $\mathbf{r} = \mathbf{r}_{(3)}$ as a reduced Ritz base, with the understanding that $\mathbf{r}_{(3)}$ is a reasonable approximation to inertial forces in mode number 1.

The *DRV*'s used were printed in a previous slide, the reduced mass matrix is the unit matrix (by orthonormalisation of the *DRV*'s), the reduced stiffness is

$$\hat{\mathbf{K}} = \mathbf{\Phi}^T \mathbf{K}\, \mathbf{\Phi} = \begin{bmatrix} +0.0820 & -0.0253 & +0.0093 \\ -0.0253 & +0.7548 & -0.2757 \\ +0.0093 & -0.2757 & +1.8688 \end{bmatrix}.$$

The eigenproblem, in Ritz coordinates is

$$\hat{\mathbf{K}}\, \boldsymbol{z} = \omega^2 \boldsymbol{z}.$$

A comparison between *exact* solution and Ritz approximation is in the next slide.

Reduced Eigenproblem using DRV base, comparison

In the following, hatted matrices refer to approximate results.

The eigenvalues matrices are

$$\mathbf{\Lambda}=\begin{bmatrix}0.0810 & 0 & 0\\ 0 & 0.6903 & 0\\ 0 & 0 & 1.7154\end{bmatrix}\quad\text{and}\quad\hat{\mathbf{\Lambda}}=\begin{bmatrix}0.0810 & 0 & 0\\ 0 & 0.6911 & 0\\ 0 & 0 & 1.9334\end{bmatrix}.$$

The eigenvectors matrices are

$$\mathbf{\Psi}=\begin{bmatrix}+0.1699 & -0.4557 & +0.5969\\ +0.3260 & -0.5969 & +0.1699\\ +0.4557 & -0.3260 & -0.5485\\ +0.5485 & +0.1699 & -0.3260\\ +0.5969 & +0.5485 & +0.4557\end{bmatrix}\quad\text{and}\quad\hat{\mathbf{\Psi}}=\begin{bmatrix}+0.1699 & -0.4553 & +0.8028\\ +0.3260 & -0.6098 & -0.1130\\ +0.4557 & -0.3150 & -0.4774\\ +0.5485 & +0.1800 & -0.1269\\ +0.5969 & +0.5378 & +0.3143\end{bmatrix}.$$

Numerical Integration

Introduction to Numerical Integration

When we reviewed the numerical integration methods, we said that some methods are unconditionally stable and others are conditionally stable, that is the response *blows-out* if the time step h is great with respect to the natural preriod of vibration, $h > \frac{T_n}{a}$, where a is a constant that depends on the numerical algorithm.
For *MDOF* systems, the relevant T is the one associated with the highest mode present in the structural model, so for moderately complex structures it becomes impossibile to use a conditionally stable algorithm.
In the following, two unconditionally stable algorithms will be analysed, i.e., the constant acceleration method, that we already know, and the new Wilson'sθ method.

Constant Acceleration, preliminaries

- The initial conditions are known:

$$\mathbf{x}_0, \quad \dot{\mathbf{x}}_0, \quad \mathbf{p}_0, \quad \rightarrow \quad \ddot{\mathbf{x}}_0 = \mathbf{M}^{-1}(\mathbf{p}_0 - \mathbf{C}\,\dot{\mathbf{x}}_0 - \mathbf{K}\,\mathbf{x}_0).$$

- With a fixed time step h, compute the constant matrices

$$\mathbf{A} = 2\mathbf{C} + \frac{4}{h}\mathbf{M}, \qquad \mathbf{B} = 2\mathbf{M}, \qquad \mathbf{K}^{+} = \frac{2}{h}\mathbf{C} + \frac{4}{h^2}\mathbf{M}.$$

Constant Acceleration, stepping

- Starting with $i = 0$, compute the effective force increment,

$$\Delta\hat{\mathbf{p}}_i = \mathbf{p}_{i+1} - \mathbf{p}_i + \mathbf{A}\dot{\mathbf{x}}_i + \mathbf{B}\ddot{\mathbf{x}}_i,$$

the tangent stiffness $\mathbf{K}_i$ and the current incremental stiffness,

$$\hat{\mathbf{K}}_i = \mathbf{K}_i + \mathbf{K}^+.$$

- For linear systems, it is

$$\Delta\mathbf{x}_i = \hat{\mathbf{K}}_i^{-1}\Delta\hat{\mathbf{p}}_i,$$

for a non linear system $\Delta\mathbf{x}_i$ is produced by the modified Newton-Raphson iteration procedure.

- The state vectors at the end of the step are

$$\mathbf{x}_{i+1} = \mathbf{x}_i + \Delta\mathbf{x}_i, \qquad \dot{\mathbf{x}}_{i+1} = 2\frac{\Delta\mathbf{x}_i}{h} - \dot{\mathbf{x}}_i$$

Constant Acceleration, new step

- Increment the step index, $i = i + 1$.
- Compute the accelerations using the equation of equilibrium,

$$\ddot{x}_i = \mathbf{M}^{-1}(\mathbf{p}_i - \mathbf{C}\,\dot{x}_i - \mathbf{K}\,x_i).$$

- Repeat the substeps detailed in the previous slide.

Modified Newton-Raphson

- Initialization

$$y_0 = x_i \qquad f_{S,0} = f_S(\text{system state})$$
$$\Delta R_1 = \Delta\hat{p}_i \qquad K_T = \hat{K}_i$$

- For $j = 1, 2, \ldots$

$$K_T \Delta y_j = \Delta R_j \qquad \rightarrow \Delta y_j \text{ (test for convergence)}$$
$$\Delta\dot{y}_j = \cdots$$
$$y_j = y_{j-1} + \Delta y_j, \qquad \dot{y}_j = \dot{y}_{j-1} + \Delta\dot{y}_j$$
$$f_{S,j} = f_S(\text{updated system state})$$
$$\Delta f_{S,j} = f_{S,j} - f_{S,j-1} - (K_T - K_i)\Delta y_j$$
$$\Delta R_{j+1} = \Delta R_j - \Delta f_{S,j}$$

- Return the value $\Delta x_i = y_j - x_i$

A suitable convergence test is

$$\frac{\Delta R_j^T \Delta y_j}{\Delta\hat{p}_i^T \Delta x_{i,j}} \leqslant \text{tol}$$

Wilson's Theta Method

The linear acceleration method is significantly more accurate than the constant acceleration method, meaning that it is possible to use a longer time step to compute the response of a *SDOF* system within a required accuracy.
On the other hand, the method is not safely applicable to *MDOF* systems due to its numerical instability.

Wilson's Theta Method

The linear acceleration method is significantly more accurate than the constant acceleration method, meaning that it is possible to use a longer time step to compute the response of a *SDOF* system within a required accuracy.
On the other hand, the method is not safely applicable to *MDOF* systems due to its numerical instability.
Professor Ed Wilson demonstrated that simple variations of the linear acceleration method can be made unconditionally stable and found the most accurate in this family of algorithms, collectively known as *Wilson's θ methods*.

Wilson's θ method

Wilson's idea is very simple: the results of the linear acceleration algorithm are *good enough* only in a fraction of the time step.

Wilson demonstrated that his idea was correct, too...

The procedure is really simple,

Wilson's θ method

Wilson's idea is very simple: the results of the linear acceleration algorithm are *good enough* only in a fraction of the time step.
Wilson demonstrated that his idea was correct, too...
The procedure is really simple,

1. solve the incremental equation of equilibrium using the linear acceleration algorithm, with an extended time step

$$\hat{h} = \theta\, h, \qquad \theta \geqslant 1,$$

Wilson's θ method

Wilson's idea is very simple: the results of the linear acceleration algorithm are *good enough* only in a fraction of the time step.
Wilson demonstrated that his idea was correct, too...
The procedure is really simple,

1. solve the incremental equation of equilibrium using the linear acceleration algorithm, with an extended time step

$$\hat{h} = \theta\, h, \qquad \theta \geqslant 1,$$

2. compute the extended acceleration increment $\hat{\Delta}\ddot{x}$ at $\hat{t} = t_i + \hat{h}$,

Wilson's θ method

Wilson's idea is very simple: the results of the linear acceleration algorithm are *good enough* only in a fraction of the time step.
Wilson demonstrated that his idea was correct, too...
The procedure is really simple,

1. solve the incremental equation of equilibrium using the linear acceleration algorithm, with an extended time step

$$\hat{h} = \theta\, h, \qquad \theta \geqslant 1,$$

2. compute the extended acceleration increment $\hat{\Delta}\ddot{x}$ at $\hat{t} = t_i + \hat{h}$,
3. scale the extended acceleration increment under the assumption of linear acceleration, $\Delta\ddot{x} = \frac{1}{\theta}\hat{\Delta}\ddot{x}$,

Wilson's θ method

Wilson's idea is very simple: the results of the linear acceleration algorithm are *good enough* only in a fraction of the time step. Wilson demonstrated that his idea was correct, too...
The procedure is really simple,

1. solve the incremental equation of equilibrium using the linear acceleration algorithm, with an extended time step

$$\hat{h} = \theta\, h, \qquad \theta \geqslant 1,$$

2. compute the extended acceleration increment $\hat{\Delta}\ddot{x}$ at $\hat{t} = t_i + \hat{h}$,
3. scale the extended acceleration increment under the assumption of linear acceleration, $\Delta\ddot{x} = \frac{1}{\theta}\hat{\Delta}\ddot{x}$,
4. compute the velocity and displacements increment using the reduced value of the increment of acceleration.

Wilson's θ method description

Using the same symbols used for constant acceleration.
First of all, for given initial conditions $\mathbf{x}_0$ and $\dot{\mathbf{x}}_0$, initialise the procedure computing the constants (matrices) used in the following procedure and the initial acceleration,

$$\begin{aligned}
\ddot{\mathbf{x}}_0 &= \mathbf{M}^{-1}(\mathbf{p}_0 - \mathbf{C}\,\dot{\mathbf{x}}_0 - \mathbf{K}\,\mathbf{x}_0),\\
\mathbf{A} &= 6\mathbf{M}/\hat{h} + 3\mathbf{C},\\
\mathbf{B} &= 3\mathbf{M} + \hat{h}\mathbf{C}/2,\\
\mathbf{K}^{+} &= 3\mathbf{C}/\hat{h} + 6\mathbf{M}/\hat{h}^2.
\end{aligned}$$

Wilson's θ method description

Starting with $i = 0$,

1. update the tangent stiffness, $\mathbf{K}_i = \mathbf{K}(x,\dot{x}_i)$ and the effective stiffness, $\hat{\mathbf{K}}_i = \mathbf{K}_i + \mathbf{K}^+$,
   compute $\hat{\Delta}\hat{\mathbf{p}}_i = \theta\Delta\mathbf{p}_i + \mathbf{A}\dot{x}_i + \mathbf{B}\ddot{x}_i$,
   with $\Delta\mathbf{p}_i = \mathbf{p}(t_i + h) - \mathbf{p}(t_i)$

Wilson's θ method description

Starting with $i = 0$,

1. update the tangent stiffness, $\mathbf{K}_i = \mathbf{K}(\mathbf{x}, \dot{\mathbf{x}}_i)$ and the effective stiffness, $\hat{\mathbf{K}}_i = \mathbf{K}_i + \mathbf{K}^+$,
   compute $\hat{\Delta}\hat{\mathbf{p}}_i = \theta\Delta\mathbf{p}_i + \mathbf{A}\dot{\mathbf{x}}_i + \mathbf{B}\ddot{\mathbf{x}}_i$,
   with $\Delta\mathbf{p}_i = \mathbf{p}(t_i + h) - \mathbf{p}(t_i)$
2. solve $\hat{\mathbf{K}}_i\hat{\Delta}\mathbf{x} = \hat{\Delta}\hat{\mathbf{p}}_i$, compute

$$\hat{\Delta}\ddot{\mathbf{x}} = 6\frac{\hat{\Delta}\mathbf{x}}{\hat{h}^2} - 6\frac{\dot{\mathbf{x}}_i}{\hat{h}} - 3\ddot{\mathbf{x}}_i \rightarrow \Delta\ddot{\mathbf{x}} = \frac{1}{\theta}\hat{\Delta}\ddot{\mathbf{x}}$$

Wilson's θ method description

Starting with $i = 0$,

1. update the tangent stiffness, $\mathbf{K}_i = \mathbf{K}(x,\dot{x}_i)$ and the effective stiffness, $\hat{\mathbf{K}}_i = \mathbf{K}_i + \mathbf{K}^+$,
   compute $\hat{\Delta}\hat{\mathbf{p}}_i = \theta\Delta\mathbf{p}_i + \mathbf{A}\dot{x}_i + \mathbf{B}\ddot{x}_i$,
   with $\Delta\mathbf{p}_i = \mathbf{p}(t_i + h) - \mathbf{p}(t_i)$
2. solve $\hat{\mathbf{K}}_i\hat{\Delta}x = \hat{\Delta}\hat{\mathbf{p}}_i$, compute

$$\hat{\Delta}\ddot{x} = 6\frac{\hat{\Delta}x}{\hat{h}^2} - 6\frac{\dot{x}_i}{\hat{h}} - 3\ddot{x}_i \rightarrow \Delta\ddot{x} = \frac{1}{\theta}\hat{\Delta}\ddot{x}$$

3. compute

$$\Delta\dot{x} = (\ddot{x}_i + \frac{1}{2}\Delta\ddot{x})h$$

$$\Delta x = \dot{x}_i h + (\frac{1}{2}\ddot{x}_i + \frac{1}{6}\Delta\ddot{x})h^2$$

Wilson's θ method description

Starting with $i = 0$,

1. update the tangent stiffness, $\mathbf{K}_i = \mathbf{K}(x, \dot{x}_i)$ and the effective stiffness, $\hat{\mathbf{K}}_i = \mathbf{K}_i + \mathbf{K}^+$,
   compute $\hat{\Delta}\hat{\mathbf{p}}_i = \theta\Delta\mathbf{p}_i + \mathbf{A}\dot{x}_i + \mathbf{B}\ddot{x}_i$,
   with $\Delta\mathbf{p}_i = \mathbf{p}(t_i + h) - \mathbf{p}(t_i)$
2. solve $\hat{\mathbf{K}}_i \hat{\Delta}x = \hat{\Delta}\hat{\mathbf{p}}_i$, compute

$$\hat{\Delta}\ddot{x} = 6\frac{\hat{\Delta}x}{\hat{h}^2} - 6\frac{\dot{x}_i}{\hat{h}} - 3\ddot{x}_i \rightarrow \Delta\ddot{x} = \frac{1}{\theta}\hat{\Delta}\ddot{x}$$

3. compute

$$\Delta\dot{x} = (\ddot{x}_i + \frac{1}{2}\Delta\ddot{x})h$$

$$\Delta x = \dot{x}_i h + (\frac{1}{2}\ddot{x}_i + \frac{1}{6}\Delta\ddot{x})h^2$$

4. update state, $x_{i+1} = x_i + \Delta x$, $\dot{x}_{i+1} = \dot{x}_i + \Delta\dot{x}$, $i = i + 1$, iterate restarting from 1.

A final remark

The Theta Method is unconditionally stable for $\theta > 1.37$ and it achieves the maximum accuracy for $\theta = 1.42$.

Multiple Support Excitation

Definitions

Consider the case of a structure where the supports are subjected to *assigned* displacements histories, $u_i = u_i(t)$.
To solve this problem, we start with augmenting the degrees of freedom with the support displacements.
We denote the superstructure *DOF* with x_T, the support *DOF* with x_g and we have a global displacement vector x,

$$x = \begin{Bmatrix} x_T \\ x_g \end{Bmatrix}.$$

The Equation of Motion

Damping effects will be introduced at the end of our manipulations. The equation of motion is

$$\begin{bmatrix}\mathbf{M} & \mathbf{M}_g \\ \mathbf{M}_g^T & \mathbf{M}_{gg}\end{bmatrix}\begin{Bmatrix}\ddot{\mathbf{x}}_T \\ \ddot{\mathbf{x}}_g\end{Bmatrix}+\begin{bmatrix}\mathbf{K} & \mathbf{K}_g \\ \mathbf{K}_g^T & \mathbf{K}_{gg}\end{bmatrix}\begin{Bmatrix}\mathbf{x}_T \\ \mathbf{x}_g\end{Bmatrix}=\begin{Bmatrix}0 \\ \mathbf{p}_g\end{Bmatrix}$$

where $\mathbf{M}$ and $\mathbf{K}$ are the usual structural matrices, while $\mathbf{M}_g$ and $\mathbf{M}_{gg}$ are, in the common case of a lumped mass model, zero matrices.

Static Components

We decompose the vector of displacements into two contributions, a static contribution and a dynamic contribution, attributing the *given* support displacements to the static contribution.

$$\begin{Bmatrix} x_T \\ x_g \end{Bmatrix} = \begin{Bmatrix} x_s \\ x_g \end{Bmatrix} + \begin{Bmatrix} x \\ 0 \end{Bmatrix}$$

where x is the usual *relative displacements* vector.

Determination of static components

Because the $\mathbf{x}_g$ are given, we can write two matricial equations that give us the static supertructure displacements and the forces we must apply to the supports,

$$\mathbf{K}\mathbf{x}_s + \mathbf{K}_g\mathbf{x}_g = 0$$
$$\mathbf{K}_g^T\mathbf{x}_s + \mathbf{K}_{gg}\mathbf{x}_g = \mathbf{p}_g$$

From the first equation we have

$$\mathbf{x}_s = -\mathbf{K}^{-1}\mathbf{K}_g\mathbf{x}_g$$

and from the second we have

$$\mathbf{p}_g = (\mathbf{K}_{gg} - \mathbf{K}_g^T\mathbf{K}^{-1}\mathbf{K}_g)\mathbf{x}_g$$

Determination of static components

Because the $\mathbf{x}_g$ are given, we can write two matricial equations that give us the static supertructure displacements and the forces we must apply to the supports,

$$\mathbf{K}\mathbf{x}_s + \mathbf{K}_g\mathbf{x}_g = 0$$
$$\mathbf{K}_g^T\mathbf{x}_s + \mathbf{K}_{gg}\mathbf{x}_g = \mathbf{p}_g$$

From the first equation we have

$$\mathbf{x}_s = -\mathbf{K}^{-1}\mathbf{K}_g\mathbf{x}_g$$

and from the second we have

$$\mathbf{p}_g = (\mathbf{K}_{gg} - \mathbf{K}_g^T\mathbf{K}^{-1}\mathbf{K}_g)\mathbf{x}_g$$

The support forces are zero when the structure is isostatic or the structure is subjected to a rigif motion.

Going back to the EOM

We need the first row of the two matrix equation of equilibrium,

$$\begin{bmatrix} \mathbf{M} & \mathbf{M}_g \\ \mathbf{M}_g^T & \mathbf{M}_{gg} \end{bmatrix} \begin{Bmatrix} \ddot{x}_T \\ \ddot{x}_g \end{Bmatrix} + \begin{bmatrix} \mathbf{K} & \mathbf{K}_g \\ \mathbf{K}_g^T & \mathbf{K}_{gg} \end{bmatrix} \begin{Bmatrix} x_T \\ x_g \end{Bmatrix} = \begin{Bmatrix} 0 \\ p_g \end{Bmatrix}$$

substituting $x_T = x_s + x$ in the first row

$$\mathbf{M}\ddot{x} + \mathbf{M}\ddot{x}_s + \mathbf{M}_g\ddot{x}_g + \mathbf{K}x + \mathbf{K}x_s + \mathbf{K}_g x_g = 0$$

by the equation of static equilibrium, $\mathbf{K}x_s + \mathbf{K}_g x_g = 0$ we can simplify

$$\mathbf{M}\ddot{x} + \mathbf{M}\ddot{x}_s + \mathbf{M}_g\ddot{x}_g + \mathbf{K}x = \mathbf{M}\ddot{x} + (\mathbf{M}_g - \mathbf{M}\mathbf{K}^{-1}\mathbf{K}_g)\ddot{x}_g + \mathbf{K}x = 0.$$

Influence matrix

The equation of motion is

$$\mathbf{M}\ddot{\mathbf{x}} + (\mathbf{M}_g - \mathbf{M}\mathbf{K}^{-1}\mathbf{K}_g)\ddot{\mathbf{x}}_g + \mathbf{K}\mathbf{x} = 0.$$

We define the *influence matrix* $\mathbf{E}$ by

$$\mathbf{E} = -\mathbf{K}^{-1}\mathbf{K}_g,$$

and write, reintroducing the damping effects,

$$\mathbf{M}\ddot{\mathbf{x}} + \mathbf{C}\dot{\mathbf{x}} + \mathbf{K}\mathbf{x} = -(\mathbf{M}\mathbf{E} + \mathbf{M}_g)\ddot{\mathbf{x}}_g - (\mathbf{C}\mathbf{E} + \mathbf{C}_g)\dot{\mathbf{x}}_g$$

Simplification of the EOM

For a lumped mass model, $\mathbf{M}_g = 0$ and also the efficace forces due to damping are really small with respect to the inertial ones, and with this understanding we write

$$\mathbf{M}\ddot{\mathbf{x}} + \mathbf{C}\dot{\mathbf{x}} + \mathbf{K}\mathbf{x} = -\mathbf{M}\mathbf{E}\ddot{\mathbf{x}}_g.$$

Significance of $\mathbf{E}$

$\mathbf{E}$ can be understood as a collection of vectors $\mathbf{e}_i$, $i = 1, \ldots, N_g$ (N_g being the number of *DOF* associated with the support motion),

$$\mathbf{E} = \begin{bmatrix} \mathbf{e}_1 & \mathbf{e}_2 & \cdots & \mathbf{e}_{N_g} \end{bmatrix}$$

where the individual $\mathbf{e}_i$ collects the displacements in all the *DOF* of the superstructure due to imposing a unit displacement to the support *DOF* number i.

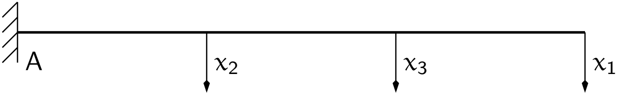
Significance of $\mathbf{E}$

This understanding means that the influence matrix can be computed column by column,

- in the general case by releasing one support *DOF*, applying a unit force to the released *DOF*, computing all the displacements and scaling the displacements so that the support displacement component is made equal to 1,
- or in the case of an isostatic component by examining the instantaneous motion of the 1 *DOF* rigid system that we obtain by releasing one constraint.

EOM example

We want to determine the influence matrix $\mathbf{E}$ for the structure in the figure above, subjected to an assigned motion in B.



First step, put in evidence another degree of freedom x_3 corresponding to the assigned vertical motion of the support in B and compute, using e.g. the PVD, the flexibility matrix:

$$\mathbf{F} = \frac{L^3}{3EJ}\begin{bmatrix} 54.0000 & 8.0000 & 28.0000 \\ 8.0000 & 2.0000 & 5.0000 \\ 28.0000 & 5.0000 & 16.0000 \end{bmatrix}.$$

EOM example

The stiffness matrix is found by inversion,

$$\mathbf{K} = \frac{3EJ}{13L^3}\begin{bmatrix} +7.0000 & +12.0000 & -16.0000 \\ +12.0000 & +80.0000 & -46.0000 \\ -16.0000 & -46.0000 & +44.0000 \end{bmatrix}.$$

We are interested in the partitions $\mathbf{K}_{xx}$ and $\mathbf{K}_{xg}$:

$$\mathbf{K}_{xx} = \frac{3EJ}{13L^3}\begin{bmatrix} +7.0000 & +12.0000.0000 \\ +12.0000 & +80.0000.0000 \end{bmatrix}, \quad \mathbf{K}_{xg} = \frac{3EJ}{13L^3}\begin{bmatrix} -16 \\ -46 \end{bmatrix}.$$

The influence matrix is

$$\mathbf{E} = -\mathbf{K}_{xx}^{-1}\mathbf{K}_{xg} = \frac{1}{16}\begin{bmatrix} 28.0000 \\ 5.0000 \end{bmatrix},$$

please compare $\mathbf{E}$ with the last column of the flexibility matrix, $\mathbf{F}$.

Response analysis

Consider the vector of support accelerations,

$$\ddot{\mathbf{x}}_g = \{\ddot{x}_{gl}, \qquad l = 1, \ldots, N_g\}$$

and the effective load vector

$$\mathbf{p}_{eff} = -\mathbf{M}\mathbf{E}\ddot{\mathbf{x}}_g = -\sum_{l=1}^{N_g} \mathbf{M}\mathbf{e}_l \ddot{x}_{gl}(t).$$

We can write the modal equation of motion for mode number n

$$\ddot{q}_n + 2\zeta_n\omega_n\dot{q}_n + \omega_n^2 q_n = -\sum_{l=1}^{N_g} \Gamma_{nl}\ddot{x}_{gl}(t)$$

where

$$\Gamma_{nl} = \frac{\boldsymbol{\psi}_n^T \mathbf{M}\mathbf{e}_l}{M_n^*}$$

Response analysis, continued

The solution $q_n(t)$ is hence, with the notation of last lesson,

$$q_n(t) = \sum_{l=1}^{N_g} \Gamma_{nl} D_{nl}(t),$$

D_{nl} being the response function for ζ_n and ω_n due to the ground excitation $\ddot{x}_{gl}$.

Response analysis, continued

The total displacements x_T are given by two contributions, $x_T = x_s + x$, the expression of the contributions are

$$x_s = E x_g(t) = \sum_{l=1}^{N_g} e_l x_{gl}(t),$$

$$x = \sum_{n=1}^{N} \sum_{l=1}^{N_g} \psi_n \Gamma_{nl} D_{nl}(t),$$

and finally we have

$$x_T = \sum_{l=1}^{N_g} e_l x_{gl}(t) + \sum_{n=1}^{N} \sum_{l=1}^{N_g} \psi_n \Gamma_{nl} D_{nl}(t).$$

Forces

For a computer program, the easiest way to compute the nodal forces is

a) compute, element by element, the nodal displacements by x_T and x_g,
b) use the element stiffness matrix compute nodal forces,
c) assemble element nodal loads into global nodal loads.

That said, let's see the analytical development...

Forces

The forces on superstructure nodes due to deformations are

$$\mathbf{f}_s = \sum_{n=1}^{N}\sum_{l=1}^{N_g} \Gamma_{nl}\mathbf{K}\boldsymbol{\psi}_n D_{nl}(t)$$

$$\mathbf{f}_s = \sum_{n=1}^{N}\sum_{l=1}^{N_g} (\Gamma_{nl}\mathbf{M}\boldsymbol{\psi}_n)(\omega_n^2 D_{nl}(t)) = \sum\sum \mathbf{r}_{nl}A_{nl}(t)$$

the forces on support

$$\mathbf{f}_{gs} = \mathbf{K}_g^T\mathbf{x}_T + \mathbf{K}_{gg}\mathbf{x}_g = \mathbf{K}_g^T\mathbf{x} + \mathbf{p}_g$$

or, using $\mathbf{x}_s = \mathbf{E}\mathbf{x}_g$

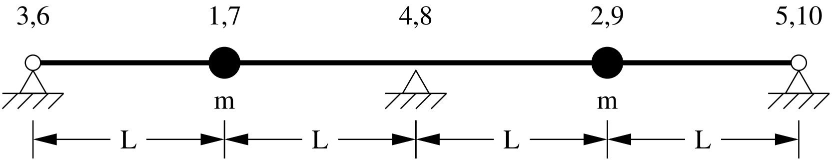
$$\mathbf{f}_{gs} = (\sum_{l=1}^{N_g} \mathbf{K}_g^T\mathbf{e}_l + \mathbf{K}_{gg,l})x_{gl} + \sum_{n=1}^{N}\sum_{l=1}^{N_g} \Gamma_{nl}\mathbf{K}_g^T\boldsymbol{\psi}_n D_{nl}(t)$$

Forces

The structure response components must be computed considering the structure loaded by all the nodal forces,

$$\mathbf{f} = \begin{Bmatrix} \mathbf{f}_s \\ \mathbf{f}_{gs} \end{Bmatrix}.$$

Example



The dynamic *DOF* are x_1 and x_2, vertical displacements of the two equal masses, x_3, x_4, x_5 are the imposed vertical displacements of the supports, $x_6, \ldots, x_{10}$ are the rotational degrees of freedom (removed by static condensation).

Example

The stiffness matrix for the 10x10 model is

$$\mathbf{K}_{10\times 10} = \frac{EJ}{L^3}\begin{bmatrix}
12 & -12 & 0 & 0 & 0 & 6L & 6L & 0 & 0 & 0\\
-12 & 24 & -12 & 0 & 0 & -6L & 0 & 6L & 0 & 0\\
0 & -12 & 24 & -12 & 0 & 0 & -6L & 0 & 6L & 0\\
0 & 0 & -12 & 24 & -12 & 0 & 0 & -6L & 0 & 6L\\
0 & 0 & 0 & -12 & 12 & 0 & 0 & 0 & -6L & -6L\\
6L & -6L & 0 & 0 & 0 & 4L^2 & 2L^2 & 0 & 0 & 0\\
6L & 0 & -6L & 0 & 0 & 2L^2 & 8L^2 & 2L^2 & 0 & 0\\
0 & 6L & 0 & -6L & 0 & 0 & 2L^2 & 8L^2 & 2L^2 & 0\\
0 & 0 & 6L & 0 & -6L & 0 & 0 & 2L^2 & 8L^2 & 2L^2\\
0 & 0 & 0 & 6L & -6L & 0 & 0 & 0 & 2L^2 & 4L^2
\end{bmatrix}$$

Example

The first product of the static condensation procedure is the linear mapping between translational and rotational degrees of freedom, given by

$$\vec{\phi} = \frac{1}{56L} \begin{bmatrix} 71 & -90 & 24 & -6 & 1 \\ 26 & 12 & -48 & 12 & -2 \\ -7 & 42 & 0 & -42 & 7 \\ 2 & -12 & 48 & -12 & -26 \\ -1 & 6 & -24 & 90 & -71 \end{bmatrix} \vec{x}.$$

Example

Following static condensation and reordering rows and columns, the partitioned stiffness matrices are

$$\mathbf{K} = \frac{EJ}{28L^3}\begin{bmatrix} 276 & 108 \\ 108 & 276 \end{bmatrix},$$

$$\mathbf{K}_g = \frac{EJ}{28L^3}\begin{bmatrix} -102 & -264 & -18 \\ -18 & -264 & -102 \end{bmatrix},$$

$$\mathbf{K}_{gg} = \frac{EJ}{28L^3}\begin{bmatrix} 45 & 72 & 3 \\ 72 & 384 & 72 \\ 3 & 72 & 45 \end{bmatrix}.$$

The influence matrix is

$$\mathbf{E} = \mathbf{K}^{-1}\mathbf{K}_g = \frac{1}{32}\begin{bmatrix} 13 & 22 & -3 \\ -3 & 22 & 13 \end{bmatrix}.$$

Example

The eigenvector matrix is

$$\mathbf{\Psi} = \begin{bmatrix} -1 & 1 \\ 1 & 1 \end{bmatrix}$$

the matrix of modal masses is

$$\mathbf{M}^\star = \mathbf{\Psi}^\mathsf{T}\mathbf{M}\mathbf{\Psi} = m\begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}$$

the matrix of the non normalized modal partecipation coefficients is

$$\mathbf{L} = \mathbf{\Psi}^\mathsf{T}\mathbf{M}\mathbf{E} = m\begin{bmatrix} -\frac{1}{2} & 0 & \frac{1}{2} \\ \frac{5}{16} & \frac{11}{8} & \frac{5}{16} \end{bmatrix}$$

and, finally, the matrix of modal partecipation factors,

$$\mathbf{\Gamma} = (\mathbf{M}^\star)^{-1}\mathbf{L} = \begin{bmatrix} -\frac{1}{4} & 0 & \frac{1}{4} \\ \frac{5}{32} & \frac{11}{16} & \frac{5}{32} \end{bmatrix}$$

Example

Denoting with $D_{ij} = D_{ij}(t)$ the response function for mode i due to ground excitation $\ddot{x}_{gj}$, the response can be written

$$\boldsymbol{x} = \begin{pmatrix} \psi_{11}\left(-\frac{1}{4}D_{11}+\frac{1}{4}D_{13}\right)+\psi_{12}\left(\frac{5}{32}D_{21}+\frac{5}{32}D_{23}+\frac{11}{16}D_{22}\right) \\ \psi_{21}\left(-\frac{1}{4}D_{11}+\frac{1}{4}D_{13}\right)+\psi_{22}\left(\frac{5}{32}D_{21}+\frac{5}{32}D_{23}+\frac{11}{16}D_{22}\right) \end{pmatrix}$$
$$= \begin{pmatrix} -\frac{1}{4}D_{13}+\frac{1}{4}D_{11}+\frac{5}{32}D_{21}+\frac{5}{32}D_{23}+\frac{11}{16}D_{22} \\ -\frac{1}{4}D_{11}+\frac{1}{4}D_{13}+\frac{5}{32}D_{21}+\frac{5}{32}D_{23}+\frac{11}{16}D_{22} \end{pmatrix}.$$