Derived Ritz Vectors, Numerical Integration

Giacomo Boffi

Dipartimento di Ingegneria Civile e Ambientale, Politecnico di Milano

May 28, 2013

Outline

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.

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 Lanczos vectors (or Derived Ritz Vectors, DRV).

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 Lanczos vectors (or Derived Ritz Vectors, DRV).

DRV's 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.

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.

If you construct a sequence of orthogonal vectors (e.g., using Gram-Schmidt algorithm) usually each new vector must be orthogonalized 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.

If you construct a sequence of orthogonal vectors (e.g., using Gram-Schmidt algorithm) usually each new vector must be orthogonalized with respect to all the other vectors. Lots of work.

Using the Lanczos procedure, when a new vector is made orthogonal with respect to the two preceding ones *only* it is found that the new vector is orthogonal to *all* the previous ones.

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.

If you construct a sequence of orthogonal vectors (e.g., using Gram-Schmidt algorithm) usually each new vector must be orthogonalized with respect to all the other vectors. Lots of work.

Using the Lanczos procedure, when a new vector is made orthogonal with respect to the two preceding ones *only* it is found that the new vector is orthogonal to *all* the previous ones.

Beware that most references to Lanczos vectors are about the original application, solving the eigenproblem for a large symmetrical matrix. Our application to structural dynamics is a bit different... let's see

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).

$$\mathbf{K}\,\boldsymbol{\ell}_1 = \mathbf{r}_0$$

2. Compute the normalization factor for the first deflected shape 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 normalizing $\boldsymbol{\ell}_1$ 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 2^{nd} 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 force shape vector. $$\mathbf{K}\,\boldsymbol{\ell}_2 = \mathbf{r}_1$$
2. Compute the contribution of the first vector to $\boldsymbol{\ell}_2$. $$\alpha_1 = \frac{\boldsymbol{\phi}_1^T \mathbf{M}\,\boldsymbol{\ell}_2}{1\text{ unit mass}}$$
3. Purify the displacements $\boldsymbol{\ell}_2$ (α_1 is dimensionally a displacement). $$\hat{\boldsymbol{\ell}}_2 = \boldsymbol{\ell}_2 - \alpha_1 \boldsymbol{\phi}_1$$
4. Compute the normalization factor. $$\beta_2^2 = \frac{\hat{\boldsymbol{\ell}}_2^T \mathbf{M}\,\hat{\boldsymbol{\ell}}_2}{1\text{ unit mass}}$$
5. Obtain the second derived Ritz vector normalizing $\hat{\boldsymbol{\ell}}_2$. $$\boldsymbol{\phi}_2 = \frac{1}{\beta_2}\hat{\boldsymbol{\ell}}_2$$

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 $\alpha_2 = \frac{\boldsymbol{\phi}_2^T\mathbf{M}\,\boldsymbol{\ell}_3}{1 \text{ unit mass}}$, $\alpha_1 = \frac{\boldsymbol{\phi}_1^T\mathbf{M}\,\boldsymbol{\ell}_3}{1 \text{ unit mass}} = \beta_2$ $\quad \hat{\boldsymbol{\ell}}_3 = \boldsymbol{\ell}_3 - \alpha_2\boldsymbol{\phi}_2 - \beta_2\boldsymbol{\phi}_1$
3. Compute the normalization 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 normalizing $\hat{\boldsymbol{\ell}}_3$. $\quad \boldsymbol{\phi}_3 = \frac{1}{\beta_2}\hat{\boldsymbol{\ell}}_3$

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 $\alpha_2 = \frac{\boldsymbol{\phi}_2^T\mathbf{M}\boldsymbol{\ell}_3}{1\text{ unit mass}}$, $\alpha_1 = \frac{\boldsymbol{\phi}_1^T\mathbf{M}\boldsymbol{\ell}_3}{1\text{ unit mass}} = \beta_2$ $\quad \hat{\boldsymbol{\ell}}_3 = \boldsymbol{\ell}_3 - \alpha_2\boldsymbol{\phi}_2 - \beta_2\boldsymbol{\phi}_1$
3. Compute the normalization 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 normalizing $\hat{\boldsymbol{\ell}}_3$. $\quad \boldsymbol{\phi}_3 = \frac{1}{\beta_2}\hat{\boldsymbol{\ell}}_3$

Note that it is not necessary to compute the contribution of the first vector, because it can be demonstrated that

$$\alpha_1 = \beta_2$$

that is, the contribution of first to third is *exactly* the normalization factor we computed to derive the second vector!

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 $\boldsymbol{\ell}_4$.
2. Purify the displacements $\boldsymbol{\ell}_4$ where
   $\alpha_3 = \frac{\boldsymbol{\phi}_3^T\mathbf{M}\boldsymbol{\ell}_4}{1m}$
   $\alpha_2 = \frac{\boldsymbol{\phi}_2^T\mathbf{M}\boldsymbol{\ell}_4}{1m} = \beta_3$
   $\alpha_1 = \frac{\boldsymbol{\phi}_1^T\mathbf{M}\boldsymbol{\ell}_4}{1m} = 0$
3. Compute the normalization factor.
4. Obtain the fourth derived Ritz vector normalizing $\hat{\boldsymbol{\ell}}_4$.

$$\mathbf{K}\,\boldsymbol{\ell}_4 = \mathbf{r}_3$$
$$\hat{\boldsymbol{\ell}}_4 = \boldsymbol{\ell}_4 - \alpha_3\boldsymbol{\phi}_3 - \beta_3\boldsymbol{\phi}_2$$

$$\beta_4 = \frac{\hat{\boldsymbol{\ell}}_4^T\mathbf{M}\hat{\boldsymbol{\ell}}_4}{1 \text{ unit mass}}$$
$$\boldsymbol{\phi}_4 = \frac{1}{\beta_4}\hat{\boldsymbol{\ell}}_4$$

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 $\quad \hat{\boldsymbol{\ell}}_4 = \boldsymbol{\ell}_4 - \alpha_3\boldsymbol{\phi}_3 - \beta_3\boldsymbol{\phi}_2$

   $$\alpha_3 = \frac{\boldsymbol{\phi}_3^T\mathbf{M}\boldsymbol{\ell}_4}{1m}$$
   $$\alpha_2 = \frac{\boldsymbol{\phi}_2^T\mathbf{M}\boldsymbol{\ell}_4}{1m} = \beta_3$$
   $$\alpha_1 = \frac{\boldsymbol{\phi}_1^T\mathbf{M}\boldsymbol{\ell}_4}{1m} = 0$$
3. Compute the normalization factor. $\quad \beta_4^2 = \dfrac{\hat{\boldsymbol{\ell}}_4^T\mathbf{M}\hat{\boldsymbol{\ell}}_4}{1 \text{ unit mass}}$
4. Obtain the fourth derived Ritz vector normalizing $\hat{\boldsymbol{\ell}}_4$. $\quad \boldsymbol{\phi}_4 = \frac{1}{\beta_4}\hat{\boldsymbol{\ell}}_4$

Note the contributions to $\boldsymbol{\phi}_4$ from the previous vectors, in particular the contribution from $\boldsymbol{\phi}_1$ is equal to zero... also the contribution from the immediately previous vector is equal to β_3. At each step, we have to solve a linear system, that was possibly put in a triangular format, and to do two double matrix products, to find α_{i-1} and β_i.

Fourth Vector, etc

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 coefficients are equal to zero, $\alpha_{i-3} = \cdots = 0$.

The Tridiagonal Matrix

Having computed $M < N$ DRV's we can write for, e.g., $M = 5$ that each non-normalized 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 $\mathbf{\Phi}$, the above can be written

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

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 $\mathbf{\Phi}^T\mathbf{M}$

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

Eigenvectors

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

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

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 $p(\mathbf{x}, t) = \mathbf{r}\, f(t)$ in terms of the *DRV*'s and Ritz coordinates $\mathbf{z}$

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

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

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

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-Orthogonalization

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

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

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

Required Number of DRV

Analogously to the modal participation factor the Ritz participation 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

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

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.

Required Number of DRV

Another way to proceed: define an error vector

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

and an error norm

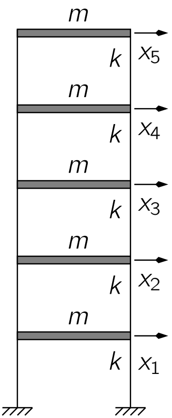
$$|\hat{e}_i| = \frac{\mathbf{r}^T \hat{\mathbf{e}}_i}{\mathbf{r}^T \mathbf{r}},$$

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 $\mathbf{M} = m \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$,

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

Eigenvalues and eigenvectors matrices are:

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

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

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 orthonormalization 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

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

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

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

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 period 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 impossible to use a conditionally stable algorithm.
In the following, two unconditionally stable algorithms will be analyzed, i.e., the constant acceleration method, that we already know, and the new Wilson's θ 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,

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

- Repeat the sub-steps detailed in the previous slide.

Modified Newton-Raphson

- Initialization

$$\mathbf{y}_0 = \mathbf{x}_i \qquad \mathbf{f}_{S,0} = \mathbf{f}_S(\text{system state})$$
$$\Delta\mathbf{R}_1 = \Delta\hat{\mathbf{p}}_i \qquad \mathbf{K}_T = \hat{\mathbf{K}}_i$$

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

$$\mathbf{K}_T \Delta\mathbf{y}_j = \Delta\mathbf{R}_j \qquad \to \Delta\mathbf{y}_j \text{ (test for convergence)}$$
$$\Delta\dot{\mathbf{y}}_j = \cdots$$
$$\mathbf{y}_j = \mathbf{y}_{j-1} + \Delta\mathbf{y}_j, \qquad \dot{\mathbf{y}}_j = \dot{\mathbf{y}}_{j-1} + \Delta\dot{\mathbf{y}}_j$$
$$\mathbf{f}_{S,j} = \mathbf{f}_S(\text{updated system state})$$
$$\Delta\mathbf{f}_{S,j} = \mathbf{f}_{S,j} - \mathbf{f}_{S,j-1} - (\mathbf{K}_T - \mathbf{K}_i)\Delta\mathbf{y}_j$$
$$\Delta\mathbf{R}_{j+1} = \Delta\mathbf{R}_j - \Delta\mathbf{f}_{S,j}$$

- Return the value $\Delta\mathbf{x}_i = \mathbf{y}_j - \mathbf{x}_i$

A suitable convergence test is

$$\frac{\Delta\mathbf{R}_j^T \Delta\mathbf{y}_j}{\Delta\hat{\mathbf{p}}_i^T \Delta\mathbf{x}_{i,j}} \le \text{tol}$$

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

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

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

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
$$\hat{h} = \theta\, h, \qquad \theta \geq 1,$$
2. compute the extended acceleration increment $\hat{\mathbf{\Delta}}\ddot{\mathbf{x}}$ at $\hat{t} = t_i + \hat{h}$,
3. scale the extended acceleration increment under the assumption of linear acceleration, $\mathbf{\Delta}\ddot{\mathbf{x}} = \frac{1}{\theta}\hat{\mathbf{\Delta}}\ddot{\mathbf{x}}$,
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 $\dot{\mathbf{x}}_0$, initialize 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, $\mathbf{K}_i = \mathbf{K}(\mathbf{x}_,\dot{\mathbf{x}}_i)$ 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 $\Delta\mathbf{p}_i = \mathbf{p}(t_i + h) - \mathbf{p}(t_i)$

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, $\mathbf{K}_i = \mathbf{K}(\mathbf{x}_, \dot{\mathbf{x}}_i)$ 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

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

3. compute

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

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

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

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

3. compute

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

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

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

A final remark

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