Master in Seismic Engineering— E.T.S.I. Industriales (U.P.M.)

Discretization Methods in Engineering

The FEM in 1D

Ignacio Romero
ignacio.romero@upm.es

September 24, 2014

Outline

1. Motivation: Review from last day.
2. Matrix form of the Ritz solution.
3. The FEM.
4. The Galerkin method.

Matrix form of the Ritz solution

We continue with the example of the elastic spring discussed last week: find the minimizer u of the potential:

$$V(u) = \int_0^L \left(\tfrac{1}{2} EA(u')^2 - f \cdot u\right) \, \mathrm{d}x - h \cdot u(L) \ ,$$

Foreseeing more complicated problems, we search for an approximation solution

$$u^h(x) = \sum_{i=1}^{r} N_i(x) d_i \ , \quad \text{with} \quad N_i(0) = 0$$

If we enforce the first order optimality condition $\frac{\partial V(u^h)}{\partial d_i} = 0$ we obtain

$$0 = \frac{\partial V(u^h)}{\partial d_i} \iff 0 = \int_0^L \left(EA(u^h)' N_i' - f \cdot N_i\right) \, \mathrm{d}x - h \cdot N_i(L)$$

Matrix form of the Ritz solution (2)

These are linear equations, so we can write, for each $i = 1, 2, \ldots r$

$$\sum_{j=1}^{r} \underbrace{\int_0^L EA\, N_i'\, N_j'\, \mathrm{d}x}_{K_{ij}} \cdot d_j = \underbrace{\int_0^L f \cdot N_i\, \mathrm{d}x + h \cdot N_i(L)}_{F_i}$$

which is just the matrix equation

$$\begin{bmatrix} K_{11} & K_{12} & K_{13} & \ldots & K_{1r} \\ K_{21} & K_{22} & K_{23} & \ldots & K_{2r} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ K_{r1} & K_{r2} & K_{r3} & \ldots & K_{rr} \end{bmatrix} \begin{Bmatrix} d_1 \\ d_2 \\ \vdots \\ d_r \end{Bmatrix} = \begin{Bmatrix} F_1 \\ F_2 \\ \vdots \\ F_r \end{Bmatrix}$$

or more compactly

$$\boxed{\mathbf{K}\mathbf{d} = \mathbf{F}}$$

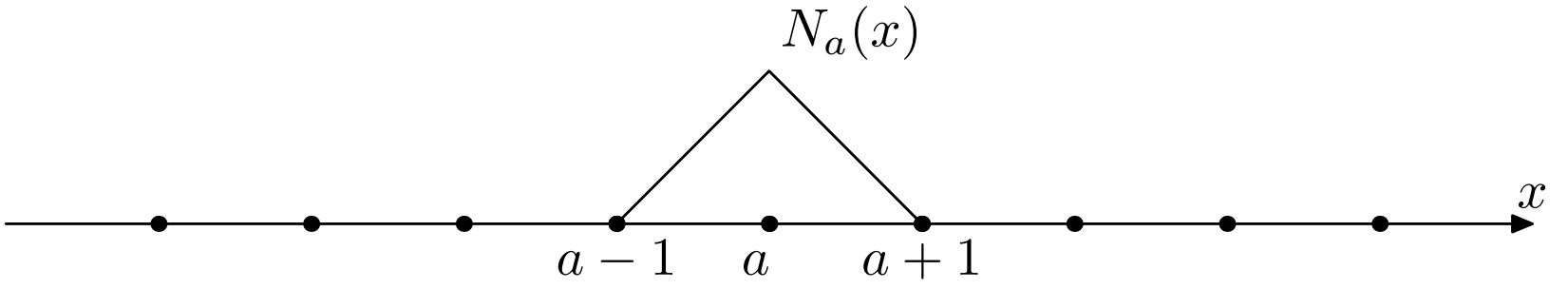
- Summary:
  - ▷ For a quadratic minimization problem ...
  - ▷ it suffices to compute the "stiffness" matrix **K** and "force" vector **F** ...
  - ▷ solve the linear system of equations $\mathbf{Kd} = \mathbf{F}$ ...
  - ▷ and reconstruct the solution $u^h(x) = \sum_{i=1}^{r} N_i(x) d_i$.

- Remarks:
  - ▷ The matrix **K** is always invertible, by hypothesis on the potential.
  - ▷ The larger r is, the more accurate the solution is (we will prove this).
  - ▷ However, the computational cost is also bigger.
  - ▷ This works for all Ritz bases.

The FEM, in its simplest form, can be understood as a particular instance of the Ritz method in which the basis of the trial space consists of piecewise polynomial functions with local support.

- Piecewise linear function in 1D: partition $[0, L]$ into n_{el} subdomains $[x_e, x_{e+1}]$ of length $h_e = x_{e+1} - x_e$, called elements. The points x_e are the nodal coordinates. Then, we define

$$N_a(x) = \begin{cases} \frac{x - x_a}{x_a - x_{a-1}} & \text{if } x \in [x_{a-1}, x_a] \\ \frac{x_{a+1} - x}{x_{a+1} - x_a} & \text{if } x \in (x_a, x_{a+1}] \\ 0 & \text{otherwise} \end{cases}$$



- The formulas for **K** and **F** are as before, however ...
- If $|i-j|>1$ then N_i and N_j have disjoint support, hence $K_{ij}=0$.
- As a result, **K** is sparse.

$$\mathbf{K} = \begin{bmatrix} K_{11} & K_{12} & 0 & 0 & \dots & 0 \\ K_{21} & K_{22} & K_{23} & 0 & \dots & 0 \\ 0 & K_{32} & K_{33} & K_{34} \dots & 0 & \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & K_{r,r-1} & K_{rr} \end{bmatrix}$$

This has enormous implications from the computational standpoint

- ▷ For a full matrix, the cost of solving a linear system is $\mathcal{O}(r^3)$
- ▷ For a banded matrix, the cost of solving a linear system is $\mathcal{O}(b^2 r)$

Consider now the advection-diffusion problem in $[0, L]$

$$a\,u' - \kappa u'' = f\ , \quad (\kappa > 0) \quad u(0) = 0, u(L) = 1$$

This problem is a model for the Navier-Stokes equations and it can not be obtained by minimizing a functional, hence the Ritz method can not be used.

The Galerkin method can be used to approximate the solution to this problem. For that, multiply the differential equation by a test function w with $w(0) = w(L) = 0$ and integrate by parts the second derivative

$$0 = \int_0^L (a\,u' - \kappa u'' - f) \cdot w \,\mathrm{d}x = \int_0^L (a\,u' \cdot w + \kappa u' \cdot w' - f \cdot w) \,\mathrm{d}x$$

The Garlerkin method (2)

We obtain the variational equation (or weak formulation of the problem)

$$\int_0^L (a\,u' \cdot w + \kappa\,u' \cdot w')\,\mathrm{d}x = \int_0^L f \cdot w\,\mathrm{d}x \qquad \text{for all } w$$

- ▷ The Galerkin method consists in replacing u and w in the previous equation with u^h and w^h, defined as

$$u^h = \sum_{j=1}^{r} N_j(x)\ d_j\ , \qquad w^h = \sum_{i=1}^{r} N_i(x)\ w_i\ .$$

By selecting $w^h = N_i$ it follows, for $i = 1, 2, \ldots r$

$$\sum_{j=1}^{r} \underbrace{\int_0^L (a\,N_j' \cdot N_i + \kappa\,N_i' \cdot N_j')\,\mathrm{d}x}_{K_{ij}}\,d_j = \underbrace{\int_0^L f \cdot N_i\,\mathrm{d}x}_{F_i}$$

. . . or in matrix terms, again $\mathbf{Kd} = \mathbf{F}$.

It is customary to find the FEM equations written in a matrix form such as the following:

- Define the interpolation vector $\mathbf{N}(x)$ and its derivative $\mathbf{B}(x)$ as

$$\mathbf{N} = [N_1(x)\ N_2(x)\ \ \ldots N_r(x)]\ , \qquad \mathbf{B} = \mathbf{N}'$$

- Then the force vector and stiffness matrix can be expressed as

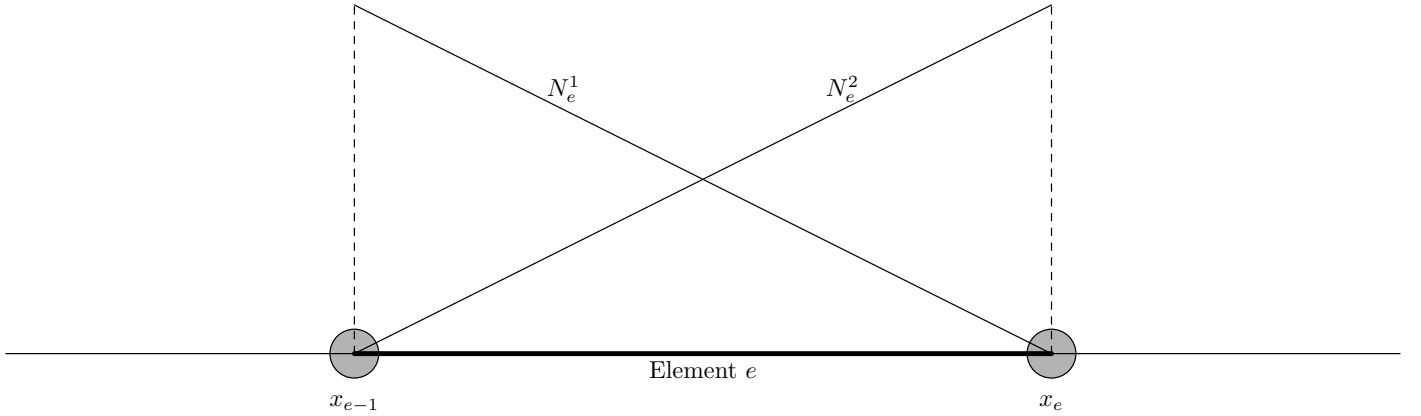
$$\mathbf{F} = \int_0^L \mathbf{N}^T(x) f \,\mathrm{d}x + \mathbf{N}(L)^T h\ , \qquad \mathbf{K} = \int_0^L \mathbf{B}^T\, EA\, \mathbf{B}\,\mathrm{d}x$$

These is the same force vector and stiffness matrix one gets from structural analysis.

- An element is a polygonal subset of the problem domain. In the case of the string, the k element is the interval $[x_{k-1}, x_k]$.
- The nodes of the element are often relabeled with a local ordering

$$x_{k-1} \to x_k^1 \qquad x_k \to x_k^2$$

- An element intersects the support of two, and only two, interpolation functions which we denote N_e^1, N_e^2. The superscript $1, 2$ is the *local* shape function numbering



- The computation of the force vector and stiffness matrix is done in an element-by-element basis. Ignoring, for simplicity, boundary terms:

$$\mathbf{F} = \begin{Bmatrix} \int_0^L N_1 f \,\mathrm{d}x \\ \int_0^L N_2 f \,\mathrm{d}x \\ \int_0^L N_3 f \,\mathrm{d}x \\ \vdots \\ \int_0^L N_{r-1} f \,\mathrm{d}x \\ \int_0^L N_r f \,\mathrm{d}x \end{Bmatrix} = \begin{Bmatrix} \int_{e_1} N_1 f \,\mathrm{d}x \\ \int_{e_1} N_2 f \,\mathrm{d}x + \int_{e_2} N_2 f \,\mathrm{d}x \\ \int_{e_2} N_3 f \,\mathrm{d}x + \int_{e_3} N_3 f \,\mathrm{d}x \\ \vdots \\ \int_{e_{r-2}} N_{r-1} f \,\mathrm{d}x + \int_{e_{r-1}} N_{r-1} f \,\mathrm{d}x \\ \int_{e_{r-1}} N_r f \,\mathrm{d}x + \int_{e_r} N_r f \,\mathrm{d}x \end{Bmatrix}$$

$$= \begin{Bmatrix} \int_{e_1} N_1 f \,\mathrm{d}x \\ \int_{e_1} N_2 f \,\mathrm{d}x \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{Bmatrix} + \begin{Bmatrix} 0 \\ \int_{e_2} N_2 f \,\mathrm{d}x \\ \int_{e_2} N_3 f \,\mathrm{d}x \\ 0 \\ \vdots \\ 0 \\ 0 \end{Bmatrix} + \begin{Bmatrix} 0 \\ 0 \\ \int_{e_3} N_3 f \,\mathrm{d}x \\ \int_{e_3} N_4 f \,\mathrm{d}x \\ \vdots \\ 0 \\ 0 \end{Bmatrix} + \cdots$$

$$\mathbf{F} = \begin{Bmatrix} \mathbf{f}^1_{e_1} \\ \mathbf{f}^2_{e_1} \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{Bmatrix} + \begin{Bmatrix} 0 \\ \mathbf{f}^1_{e_2} \\ \mathbf{f}^2_{e_2} \\ 0 \\ \vdots \\ 0 \\ 0 \end{Bmatrix} + \begin{Bmatrix} 0 \\ 0 \\ \mathbf{f}^1_{e_3} \\ \mathbf{f}^2_{e_3} \\ \vdots \\ 0 \\ 0 \end{Bmatrix} + \cdots = \mathop{\mathbf{A}}_{e=1}^{r-1} \mathbf{f}_e$$

where $\mathbf{A}$ is the assembly operator that puts each entry of the element force vector $\mathbf{f}^e$ into the large vector $\mathbf{F}$.

- The element force vector is

$$\mathbf{f}_e = \int_{x_{e-1}}^{x_e} \begin{Bmatrix} N_e^1 \\ N_e^2 \end{Bmatrix} f \, \mathrm{d}x$$

with $N_e^1 = N_{e-1}$ and $N_e^2 = N_e$

- The computation of the stiffness matrix follows the same idea

$$\mathbf{K} = \mathop{\mathbf{A}}_{k=1}^{r-1} \mathbf{k}_e$$

where the element stiffness matrix is

$$\mathbf{k}_e = \int_{x_{e-1}}^{x_e} EA \begin{bmatrix} (N_e^1)'\,(N_e^1)' & (N_e^1)'\,(N_e^2)' \\ (N_e^2)'\,(N_e^1)' & (N_e^2)'\,(N_e^2)' \end{bmatrix} \mathrm{d}x$$

The element point of view (5)

The idea behind the element point of view is that instead of calculating the force and stiffness as in

```
for a = 1 : nnode
    compute ∫_0^L N^a f dx
    for b = 1 : nnode
        compute ∫_0^L EA (N^a)'(N^b)' dx
    end for
end for
```

we do

```
for e = 1 : nel
    compute f_e and assemble it into F
    compute k_e and assemble it into K
end for
```

- Instead of referring everything to the element $[x_{e-1}, x_e]$ which could be of varying size, it is more convenient to refer all the expressions to the biunit element $[-1, +1]$.
- Define the shape functions in the biunit segment

$$\tilde{N}^1(\xi) = \tfrac{1}{2}(1-\xi)\ , \qquad \tilde{N}^2(\xi) = \tfrac{1}{2}(1+\xi)$$

- Define the map $\varphi_e : [-1, +1] \to [x_{e-1}, x_e]$ by

$$\varphi(\xi) = \tilde{N}^1(\xi)\, x_{e-1} + \tilde{N}^2(\xi)\, x_e = \frac{x_e - x_{e-1}}{2}(\xi + 1) + x_{e-1}$$

Then the shape functions in the elements and their derivatives are

$$\tilde{N}^1(\xi) = N_e^1(\varphi_e(\xi))\ , \qquad \tilde{N}^2(\xi) = N_e^2(\varphi_e(\xi))\ ,$$
$$(\tilde{N}^1)' = (N_e^1)'\, J\ , \quad (\tilde{N}^2)' = (N_e^2)'\, J\ ,$$

with

$$J = \frac{\mathrm{d}\, x}{\mathrm{d}\, \xi} = \frac{\mathrm{d}\, \varphi_e}{\mathrm{d}\, \xi} = (\tilde{N}^1)'\, x_{e-1} + (\tilde{N}^2)'\, x_e\ .$$

Final form of the integrals

The integrals in the biunit interval can be written as

$$\begin{aligned}\mathbf{f}_e &= \int_{x_{e-1}}^{x_e} f(x) \begin{Bmatrix} N_e^1(x) \\ N_e^2(x) \end{Bmatrix} \mathrm{d}x \\ &= \int_{-1}^{+1} f(\boldsymbol{\varphi}_e(\xi)) \begin{Bmatrix} \tilde{N}^1(\xi) \\ \tilde{N}^2(\xi) \end{Bmatrix} J(\xi)\, \mathrm{d}\xi\end{aligned}$$

$$\begin{aligned}\mathbf{k}_e &= \int_{x_{e-1}}^{x_e} EA \begin{bmatrix} (N_e^1)'\,(N_e^1)' & (N_e^1)'\,(N_e^2)' \\ (N_e^2)'\,(N_e^1)' & (N_e^2)'\,(N_e^2)' \end{bmatrix} \mathrm{d}x \\ &= \int_{-1}^{+1} EA \begin{bmatrix} (\tilde{N}^1)'\,(\tilde{N}^1)' & (\tilde{N}^1)'\,(\tilde{N}^2)' \\ (\tilde{N}^2)'\,(\tilde{N}^1)' & (\tilde{N}^2)'\,(\tilde{N}^2)' \end{bmatrix} \frac{1}{J}\, \mathrm{d}\xi \\ &= \int_{-1}^{+1} \frac{EA}{4} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \frac{1}{J}\, \mathrm{d}\xi\end{aligned}$$

- Computers cannot, in general, perform analytical integrals of complex functions.
- On the other hand, one does not need an exact evaluation of the integrals since the approximation of the spaces gives already unavoidable errors.
- Integrals are then approximated by $quadrature$

$$\int_{-1}^{1} f(\xi)\,\mathrm{d}\xi \approx \sum_{l=1}^{Q} f(\xi_l)\,W_l$$

- The points ξ_l are the quadrature points, W_l are the weights.
- The accuracy depends on the order Q and the choice of points/weights.

The most accurate (per quadrature point) is the so-called Gauss quadrature.

Number of points, n	Points, x_i	Weights, w_i
1	0	2
2	$\pm 1/\sqrt{3}$	1
3	0	$8/9$
	$\pm\sqrt{15}/5$	$5/9$
4	$\pm\sqrt{\left(3-2\sqrt{6/5}\right)/7}$	$\frac{18+\sqrt{30}}{36}$
	$\pm\sqrt{\left(3+2\sqrt{6/5}\right)/7}$	$\frac{18-\sqrt{30}}{36}$
5	0	$128/225$
	$\pm\frac{1}{3}\sqrt{5-2\sqrt{10/7}}$	$\frac{322+13\sqrt{70}}{900}$
	$\pm\frac{1}{3}\sqrt{5+2\sqrt{10/7}}$	$\frac{322-13\sqrt{70}}{900}$

- ▷ The Gauss rule of p points integrates exactly polynomials up to order $2p-1$.

Since the fem integrals are done over elements $[x_{k-1}, x_k]$ of dimension $h_k = x_k - x_{k-1}$ we have

$$\int_{x_{k-1}}^{x_k} f(x)\,\mathrm{d}x = \int_{-1}^{1} f(x_{k-1} + \frac{h}{2}(\xi + 1))\frac{h}{2}\,\mathrm{d}\xi = \int_{-1}^{1} \tilde{f}(\xi)\,\mathrm{d}\xi$$

which is precisely of the form suitable for quadrature.

- The finite element method can be recovered from the Ritz method by means of a particular choice of solution space.
- In addition to all the nice properties of the Ritz method, the FEM results in sparse "stiffness" matrices, a property with very important practical consequences.
- In the FEM, the terms in the "stiffness" and "force" vector can be easily computed from "local" information (no need to know the solution at every node).
- The Ritz method can only be applied to problems derived from the minimization of a functional. There is a more general discretization technique, the Galerkin method, that provides a systematic way to construct approximate solutions to problems expressed in weak form.
- If the solution space of the Gakerkin method is chosen to be piecewise polynomial with compact support, it leads to finite element approximations.
- In minimization problems, the Galerkin and the Ritz method lead exactly to the same approximations.

- The finite element equations are, ultimately, programmed in a computer which works with matrices and vectors, which is why it is useful to reformulate all of them in matrix form.
- The "element point of view" is the most common and computationally efficient.
- Integrals are evaluated with quadratures, among which the Gauss quadrature is the most accurate one.

Master in Seismic Engineering— E.T.S.I. Industriales (U.P.M.)

Discretization Methods in Engineering

The FEM in 1D

Ignacio Romero
ignacio.romero@upm.es

September 24, 2014