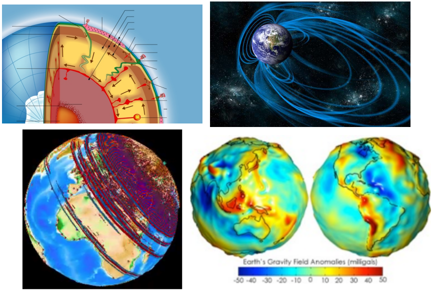
Numerical Analysis: Finite Element Method

Master 2 STPE

Emmanuel Chaljub[1]

[1] Institut des Sciences de la Terre, Univ Grenoble Alpes

Motivation: Numerical physics



- Different paths but same target (Earth)
- Observation ⇔ Physical models (forces, conservation laws, constitutive equations . . .)
- Need to validate physical models: numerical modeling can help

Motivation: Numerical physics

From physics . . . to numerical physics in details

- Physical modeling of a given problem
- Mathematical modeling: system of Partial Differential Equations
- Mathematical analysis: existence, unicity, solutions properties
- Design of a numerical method: definition of a discrete problem (Finite Difference Method, Finite Element Method)
- Numerical analysis: stability, convergence, accuracy
- Algorithm: resolution strategy of the discrete problem
- Implementation on a computer: programming (langage)
- Verification (that the numerical solution solves the discrete problem)
- Validation (of the physical model through comparison of the numerical solution to observations)

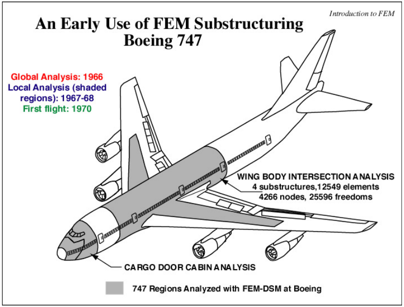
Motivation: Numerical physics

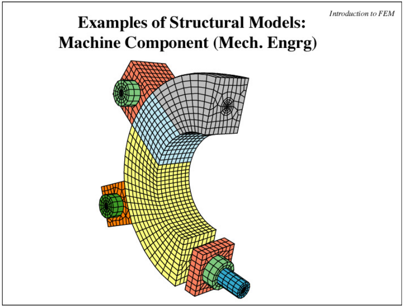
From physics . . . to numerical physics in details   This lecture

- Physical modeling of a given problem
- Mathematical modeling: system of Partial Differential Equations
- Mathematical analysis: existence, unicity, solutions properties
- Design of a numerical method: definition of a discrete problem (Finite Difference Method, Finite Element Method)
- Numerical analysis: stability, convergence, accuracy
- Algorithm: resolution strategy of the discrete problem
- Implementation on a computer: programming (langage)
- Verification (that the numerical solution solves the discrete problem)
- Validation (of the physical model through comparison of the numerical solution to observations)

History of the Finite Element Method

- 1950-1960: developed and applied in the aerospace industry (Boeing) by engineers to solve problems in complex geometries

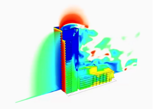




- Then extended to other engineering applications and many physical situations
- Main domains: structural dynamics, thermodynamics, fluid mechanics

History of the Finite Element Method

- Many references (books, lectures), with different focus (maths, physics, engineering, computer...)
- Many open-source or commercial codes
- Cloud platforms:
  https://www.simscale.com/



- Even phone apps:
  https://quickfem.com/

Setting up the problem to solve

The (mathematical) formulation of the problem to solve is

Find $\mathbf{u}(\mathbf{x}, t)$ such that

$$\begin{aligned} \mathcal{B}(\mathbf{u}(\mathbf{x}, t)) + \mathcal{A}(\mathbf{u}(\mathbf{x}, t)) \quad = \quad & \mathbf{F}(\mathbf{x}, t) \quad \text{for } \mathbf{x} \in \Omega, t \in [0, T] \\ & + \mathbf{BC} \quad + \mathbf{IC} \end{aligned}$$

- Ω is the physical domain (1D, 2D, 3D), possibly unbounded
- $\mathbf{u}(\mathbf{x}, t)$ is the unknown physical quantity
- $\mathcal{B}$ contains the time derivatives (usually 1st or 2nd order)
- $\mathcal{A}$ contains the space derivatives (usually 1st or 2nd order)
- $\mathbf{F}$ is the source term
- **BC** = Boundary Conditions (space)
- **IC** = Initial Conditions (time)

Example: 1D steady diffusion

Find $u(x)$ such that

$$-\frac{d}{dx}\left(k(x)\frac{du}{dx}(x)\right) = s(x) \quad \text{for } x \in \Omega$$

$$+\textbf{BC}$$

- $\Omega = [a, b]$
- $u(x)$ is the unknown steady solution (e.g. temperature field)
- $k(x)$ is the diffusivity
- $s(x)$ is the (permanent) source term
- **BC** : two kinds

$$\begin{cases} \text{Dirichlet:} & \text{condition on the solution } u(a) \text{ or } u(b) \\ \text{Neumann:} & \text{condition on the flux } \frac{du}{dx}(a) \text{ or } \frac{du}{dx}(b) \end{cases}$$

- Simplest case: $k(x) = k_0$, $s(x) = 0$: u is linear from $u(a)$ to $u(b)$

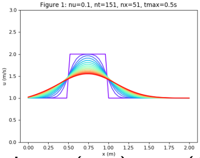
Example: 1D unsteady diffusion

Find $u(x,t)$ such that

$$\frac{\partial u}{\partial t}(x,t) - \frac{\partial}{\partial x}\left(k(x)\frac{\partial u}{\partial x}(x,t)\right) = s(x,t) \text{ for } x \in \Omega, t \in [0,T]$$

$$+\mathbf{BC} + \mathbf{IC}$$

- $\Omega = [a,b]$
- $u(x,t)$ is the unknown solution (e.g. temperature field)
- $k(x)$ is the diffusivity
- $s(x,t)$ is the source term
- **BC** : two kinds



$$\begin{cases} \text{Dirichlet:} & \text{condition on the solution } u(a,t) \text{ or } u(b,t)\ t \in [0,T] \\ \text{Neumann:} & \text{condition on the flux } \frac{du}{dx}(a,t) \text{ or } \frac{du}{dx}(b,t)\ t \in [0,T] \end{cases}$$

- **IC** : $u(x,0) = u_0(x)$

Example: 1D elastostatics

Find $u(x)$ such that

$$-\frac{d}{dx}\left(\mu(x)\frac{du}{dx}(x)\right) = f(x) \quad \text{for } x \in \Omega$$
$$+\textbf{BC}$$

- $\Omega = [a, b]$
- $u(x)$ is displacement, $\epsilon(x) = \frac{du}{dx}(x)$ is the strain
- $\mu(x)$ is the elastic modulus, $\sigma(x) = \mu(x)\epsilon(x)$ is the stress
- $f(x)$ is the (permanent) forcing term
- **BC** : two kinds

$$\begin{cases} \text{Dirichlet:} & \text{condition on the displacement } u(a) \text{ or } u(b) \\ \text{Neumann:} & \text{condition on the stress } \sigma(a) \text{ or } \sigma(b) \end{cases}$$

Example: 1D elastodynamics

Find $u(x,t)$ such that

$$\rho(x)\frac{\partial^2 u}{\partial t^2}(x,t) - \frac{\partial}{\partial x}\left(\mu(x)\frac{\partial u}{\partial x}(x,t)\right) = f(x,t) \quad \text{for } x \in \Omega$$

$$+\mathbf{BC} + \mathbf{IC}$$

- $\Omega = [a,b]$
- $u(x,t)$ is displacement, $\epsilon(x,t) = \frac{\partial u}{\partial x}(x,t)$ is the strain
- $\mu(x)$ is the elastic modulus, $\sigma(x,t) = \mu(x)\epsilon(x,t)$ is the stress
- $\rho(x)$ is mass density
- $f(x,t)$ is the forcing term
- **BC** : two kinds

$$\begin{cases} \text{Dirichlet:} & \text{condition on the displacement } u(a,t) \text{ or } u(b,t) \\ \text{Neumann:} & \text{condition on the stress } \sigma(a,t) \text{ or } \sigma(b,t) \end{cases}$$

- **IC** $\begin{cases} \text{displacement:} & u(x,0) = u_0(x) \\ \text{velocity:} & \frac{\partial u}{\partial t}(x,0) = v_0(x) \end{cases}$

Solving strategy: grid based approximation

Find $\mathbf{u}(\mathbf{x}, t)$ such that

$$\mathcal{B}(\mathbf{u}(\mathbf{x}, t)) + \mathcal{A}(\mathbf{u}(\mathbf{x}, t)) = \mathbf{F}(\mathbf{x}, t) \quad \text{for } \mathbf{x} \in \Omega, t \in [0, T]$$
$$+\mathbf{BC} \quad + \mathbf{IC}$$

- Ω is the physical domain (1D, 2D, 3D), possibly unbounded
- $\mathbf{u}(\mathbf{x}, t)$ is the unknown physical quantity
- $\mathcal{B}$ contains the time derivatives (usually 1st or 2nd order)
- $\mathcal{A}$ contains the space derivatives (usually 1st or 2nd order)
- $\mathbf{F}$ is the source term
- **BC** = Boundary Conditions (space)
- **IC** = Initial Conditions (space)

Our goal: compute $u_h(x_j, t_i)$ ...

... an approximation of $u(x, t)$ at discrete positions x_j and times t_i.

Origin of the Finite Element Method (FEM)

Find $\mathbf{u}(\mathbf{x}, t)$ such that

$$\mathcal{B}(\mathbf{u}(\mathbf{x}, t)) \; + \; \mathcal{A}(\mathbf{u}(\mathbf{x}, t)) \quad = \quad 0$$
$$+\mathbf{BC} \quad + \mathbf{IC}$$

- Assume a solution of the form $\mathbf{u}_h(\mathbf{x}, t) = \mathbf{u}_b(\mathbf{x}, t) + \Sigma_i c_i(t)\mathbf{u}_i(x, t)$
- $\mathbf{u}_b$ is a known function satisfying **BC** on the surface $\mathcal{S}$
- $\mathbf{u}_i$ are known trial functions which vanish on $\mathcal{S}$
- $\mathbf{u}_i$ are continuous (discontinuous FEM exist: DGM)
- $c_i(t)$ are unknown time-dependent coefficients
- Define the residual $\mathcal{R}(\mathbf{u}_h)(\mathbf{x}, t) = \mathcal{B}(\mathbf{u}_h(\mathbf{x}, t)) \; + \; \mathcal{A}(\mathbf{u}_h(\mathbf{x}, t))$

Your goal

Make the residual small to find the $c_i(t)$

Origin of the Finite Element Method (FEM): weighted residual + Galerkin

Find $\mathbf{u}(\mathbf{x}, t)$ such that

$$\begin{aligned} \mathcal{B}(\mathbf{u}(\mathbf{x}, t)) + \mathcal{A}(\mathbf{u}(\mathbf{x}, t)) &= 0 \\ &+\mathbf{BC} \quad +\mathbf{IC} \end{aligned}$$

- Define the residual $\mathcal{R}(\mathbf{u}_h)(\mathbf{x}, t) = \mathcal{B}(\mathbf{u}_h(\mathbf{x}, t)) + \mathcal{A}(\mathbf{u}_h(\mathbf{x}, t))$
- Define some weight functions and force the weighted residuals to vanish at any time t:

$$\langle \mathbf{w}_j, \mathcal{R}(\mathbf{u}_h)(\cdot, t) \rangle = 0 \quad \text{for all } j = 1, 2 \dots J$$

- $\langle \mathbf{u}, \mathbf{w} \rangle$: integral ($L^2$) scalar product on Ω
- $\langle \mathbf{u}, \mathbf{w} \rangle = \int_\Omega \mathbf{u}(\mathbf{x}) \cdot \mathbf{w}(\mathbf{x}) \, dx$

Galerkin method

The weight functions are chosen to be the same as the trial functions

Principle of the FEM

Original equation. Find $\mathbf{u}(\mathbf{x},t)$ such that

$$\mathcal{B}(\mathbf{u}(\mathbf{x},t)) \;+\; \mathcal{A}(\mathbf{u}(\mathbf{x},t)) \;=\; \mathbf{F}(\mathbf{x},t) \quad + \mathbf{BC} \quad + \mathbf{IC}$$

Weak, or variational form. Compute $\mathbf{u}_h(\mathbf{x},t)$ such that

$$\mathbf{u}_h(\mathbf{x},t) \;=\; \sum_{j=1}^{J} \mathbf{u}_j(t)\,\mathbf{w}_j(\mathbf{x}) \quad \text{and for all times } t \text{ and all } j = 1 \ldots J\text{:}$$

$$\int_\Omega \mathcal{B}(\mathbf{u}_h(\mathbf{x},t))\cdot\mathbf{w}_j(\mathbf{x})\,d\mathbf{x} + \int_\Omega \mathcal{A}(\mathbf{u}_h(\mathbf{x},t))\cdot\mathbf{w}_j(\mathbf{x})\,d\mathbf{x} = \int_\Omega \mathbf{F}(\mathbf{x},t)\cdot\mathbf{w}_j(\mathbf{x})\,d\mathbf{x}$$

How to proceed? Follow the recipe:

- Discretize the domain Ω into (finite) elements
- Define the trial functions: polynomials local to the elements
- Approximate the integrals (numerical integration)
- Solve the discretized system

Steady diffusion, homogeneous Dirichlet conditions

Find $u(x)$ such that

$$\begin{aligned} -\frac{d}{dx}\left(k(x)\frac{du}{dx}(x)\right) &= s(x) \quad \text{for } x \in \Omega \\ u(a) &= 0 \\ u(b) &= 0 \end{aligned} \tag{1}$$

Functional spaces

$$\begin{aligned} L^2(\Omega) &= \left\{ w : \int_\Omega w^2(x)dx < \infty \right\} \\ \mathcal{W} = H_0^1(\Omega) &= \left\{ w \in L^2(\Omega) : \frac{dw}{dx} \in L^2(\Omega), w(a) = w(b) = 0 \right\} \end{aligned}$$

Multiply Eq. (1) by w in $\mathcal{W}$ and integrate over Ω:

$$-\int_\Omega \frac{d}{dx}\left(k(x)\frac{du}{dx}(x)\right) w(x)\, dx = \int_\Omega s(x)w(x)\, dx \tag{2}$$

Integrate by parts the left hand side Toolbox:

Steady diffusion, homogeneous Dirichlet conditions

Find $u(x)$ such that

$$-\frac{d}{dx}\left(k(x)\frac{du}{dx}(x)\right) = s(x) \quad \text{for } x \in \Omega \tag{1}$$
$$u(a) = 0$$
$$u(b) = 0$$

Multiply Eq. (1) by w in $\mathcal{W}$ and integrate over Ω:

$$-\int_\Omega \frac{d}{dx}\left(k(x)\frac{du}{dx}(x)\right) w(x)\,dx = \int_\Omega s(x)w(x)\,dx \tag{2}$$

Integrate by parts the left hand side Toolbox:

$$\int_\Omega k(x)\frac{du}{dx}(x)\frac{dw}{dx}(x)\,dx - \left[k(s)\frac{du}{dx}(s)w(s)\right]_{s=a}^{s=b} = \int_\Omega s(x)w(x)\,dx \tag{3}$$

Note that because $w(a) = w(b) = 0$, the boundary term vanishes.

Find $u(x)$ such that forall w in $\mathcal{W}$

$$\int_\Omega k(x)\frac{du}{dx}(x)\frac{dw}{dx}(x)\,dx = \int_\Omega s(x)w(x)\,dx \tag{4}$$

Steady diffusion, homogeneous Dirichlet conditions

Find $u(x)$ such that (strong form)

$$
\begin{aligned}
-\frac{d}{dx}\left(k(x)\frac{du}{dx}(x)\right) &= s(x) \quad \text{for } x \in \Omega \qquad (1)\\
u(a) &= 0\\
u(b) &= 0
\end{aligned}
$$

Find $u(x)$ such that forall w in $\mathcal{W}$ (variational form)

$$
\int_\Omega k(x)\frac{du}{dx}(x)\frac{dw}{dx}(x)\,dx = \int_\Omega s(x)w(x)\,dx \qquad (2)
$$

The strong and variational (or weak) forms are equivalent (admitted).

Steady diffusion: discretization

Find $u(x)$ such that forall w in $\mathcal{W}$

$$\int_\Omega k(x)\frac{du}{dx}(x)\frac{dw}{dx}(x)\,dx = \int_\Omega s(x)w(x)\,dx \tag{3}$$

The variational formulation involves $\mathcal{W}$ which has an infinite dimension. We need to approximate it. This is the discretization process. Let us proceed with the simplest finite element approximation.

⚠ Boxed paragraphs like this one will be used for advanced material which can be overlooked in a first lecture.

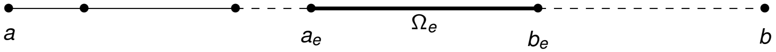
When required we will use this sign:

⏮ Back to some material that we have overlooked

Steady diffusion: discretization (mesh)

To start, we need to discretize the geometry. This is call the meshing.
Formally, we write: $\Omega = \bigcup_{e=1}^{E} \Omega_e$, with E the total number of elements.
For our 1D problem:

$$\begin{array}{rcl} \Omega & = & [a, b] \\ \Omega_e & = & [a_e, b_e] \\ a_1 = a & ; & b_E = b \end{array}$$



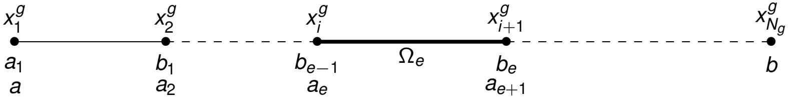
⚠ Each element Ω_e is the image of the reference 1D element $\hat{\Omega} = [-1, +1]$ by a linear mapping $\mathcal{F}_e$. If ξ is the working coordinate on $\hat{\Omega}$, then:

$$\left\{ \begin{array}{rclcl} x & = & \mathcal{F}_e(\xi) & = & \alpha_e \xi + \beta_e \\ & & \alpha_e & = & (b_e - a_e)/2 \\ & & \beta_e & = & (a_e + b_e)/2 \end{array} \right.$$

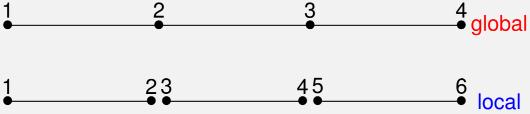
The jacobian of the mapping is denoted $\mathcal{J}_e$. It is the derivative of the mapping: $\mathcal{J}_e(\xi) = \frac{d\mathcal{F}_e}{d\xi}(\xi) = \frac{dx}{d\xi}$. In our example, it is constant (and equal to α_e). It quantifies the change of length between the reference element $\hat{\Omega}$ (length=2) and the deformed element Ω_e (length=$b_e - a_e$).

Steady diffusion: discretization (mesh)

We also need a global numbering for the corners of the elements. We assume that the total number of corners (each one counted once) is N_g. Let us denote them $x_i^g, i = 1 \dots N_g$. Each global node (aka *degree of freedom*) is related to 2 corners:



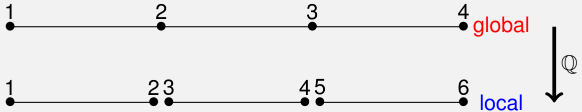
To go from the local to the global numbering, we introduce the so-called connectivity matrix, $\mathbb{Q}$. Take a simple example:



Let V_G be a vector in the global space and V_L in the local space. The connectivity matrix is such that : $V_L = \mathbb{Q} V_G$.

Steady diffusion: discretization (mesh)

⚠ To go from the local to the global numbering, we introduce the so-called connectivity matrix, $\mathbb{Q}$. Take a simple example:



Let V_G be a vector in the global space and V_L in the local space. The connectivity matrix is such that : $V_L = \mathbb{Q} V_G$.

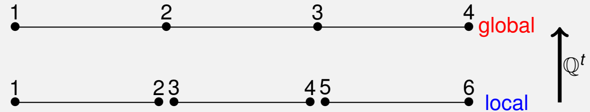
The action of $\mathbb{Q}$ is to duplicate the values of the global vector to the local vector. For the above example, the connectivity matrix reads:

$$\mathbb{Q} = \begin{pmatrix} \mathbf{1} & 0 & 0 & 0 \\ 0 & \mathbf{1} & 0 & 0 \\ 0 & \mathbf{1} & 0 & 0 \\ 0 & 0 & \mathbf{1} & 0 \\ 0 & 0 & \mathbf{1} & 0 \\ 0 & 0 & 0 & \mathbf{1} \end{pmatrix} \tag{3}$$

Exercise: compute the action of $\mathbb{Q}$ on the global vector V_G such that $V_G^t = [1, 1, 1, 1]$

Steady diffusion: discretization (mesh)

⚠ To go from the local to the global numbering, we introduce the so-called connectivity matrix, $\mathbb{Q}$. Take a simple example:



Let V_G be a vector in the global space and V_L in the local space. The connectivity matrix is such that : $V_L = \mathbb{Q} V_G$.

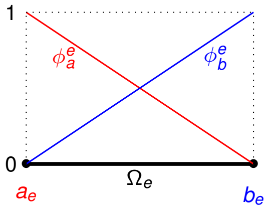
The transpose of the connectivity matrix is such that : $V_G = \mathbb{Q}^t V_L$. The action of $\mathbb{Q}^t$ is to add the values of the local vector to the global vector. $\mathbb{Q}^t$ is sometimes called the assembly matrix. For the above example, the transpose of the connectivity matrix reads:

$$\mathbb{Q}^t = \begin{pmatrix} \mathbf{1} & 0 & 0 & 0 & 0 & 0 \\ 0 & \mathbf{1} & \mathbf{1} & 0 & 0 & 0 \\ 0 & 0 & 0 & \mathbf{1} & \mathbf{1} & 0 \\ 0 & 0 & 0 & 0 & 0 & \mathbf{1} \end{pmatrix} \tag{3}$$

<u>Exercise:</u> compute the action of $\mathbb{Q}^t$ on the local vector V_L such that $V_L^t = [1, 1, 1, 1, 1, 1]$

Steady diffusion: discretization (polynomials)

Next, to approximate $\mathcal{W}$ we build a basis of functions consisting of polynomials on the elements. The simplest choice is to use first order (linear) polynomials. Each element carries two local basis polynomials ϕ_a^e and ϕ_b^e, with $\phi_a^e(a_e) = \phi_b^e(b_e) = 1$ and $\phi_a^e(b_e) = \phi_b^e(a_e) = 0$:

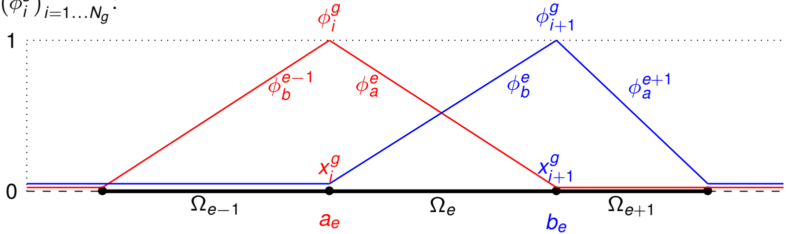


Any function ψ defined on Ω_e can be approximated through linear interpolation using the local basis:

$$\psi(x) \simeq \psi(a_e)\phi_a^e(x) + \psi(b_e)\phi_b^e(x).$$

ϕ_a^e and ϕ_b^e are called shape functions related to the points a_e and b_e, respectively.

The local polynomials are used to buid global, continuous basis functions on Ω, $(\phi_i^g)_{i=1\ldots N_g}$:



The basis functions ϕ_i^g are also called shape functions related to the global points x_i^g. They have a local support (they vanish everywhere except close to the points x_i^g).

Any function ψ defined on Ω can be approximated through linear interpolation using the global basis:

$$\psi(x) \simeq \sum_{i=1}^{N_g} \psi(x_i^g)\, \phi_i^g(x)$$

Steady diffusion: discretization (algebraic system)

Find $u(x)$ such that forall w in $\mathcal{W}$

$$\int_\Omega k(x)\frac{du}{dx}(x)\frac{dw}{dx}(x)\,dx = \int_\Omega s(x)w(x)\,dx \qquad (3)$$

Back to the variational formulation, we perform two things:

1. we decompose the unknown u and the source term s onto the global basis

$$\begin{aligned} u(x) &= \sum_{i=1}^{N_g} u(x_i^g)\,\phi_i^g(x) \\ s(x) &= \sum_{i=1}^{N_g} s(x_i^g)\,\phi_i^g(x) \end{aligned}$$

2. we write the variational formulation for each basis function ϕ_j^g

$$\sum_{i=1}^{N_g} u(x_i^g)\int_\Omega k(x)\frac{d\phi_i^g(x)}{dx}\frac{d\phi_j^g(x)}{dx}\,dx = \sum_{i=1}^{N_g} s(x_i^g)\int_\Omega \phi_i^g(x)\phi_j^g(x)\,dx$$

Steady diffusion: discretization (algebraic system)

Find $u(x_i^g)$ such that for all basis function ϕ_j^g:

$$\sum_{i=1}^{N_g} u(x_i^g) \int_\Omega k(x) \frac{d\phi_i^g(x)}{dx} \frac{d\phi_j^g(x)}{dx}\, dx = \sum_{i=1}^{N_g} s(x_i^g) \int_\Omega \phi_i^g(x) \phi_j^g(x)\, dx \qquad (3)$$

- Let $\mathbf{U}^g$ denote the vector (of size N_g) which stores the entries $(u(x_i^g))$.
- Let $\mathbf{S}^g$ denote the vector (of size N_g) which stores the entries $(s(x_i^g))$.
- Let $\mathbb{K}^g$ denote the matrix, which entries are

$$\mathbb{K}_{ij}^g = \int_\Omega k(x) \frac{d\phi_i^g(x)}{dx} \frac{d\phi_j^g(x)}{dx}\, dx$$

- Let $\mathbb{M}^g$ denote the matrix, which entries are

$$\mathbb{M}_{ij}^g = \int_\Omega \phi_i^g(x) \phi_j^g(x)\, dx$$

Exercise: Show that Eq. 3 is the j^{th} row of the algebraic system:

$$\mathbb{K}^g \mathbf{U}^g = \mathbb{M}^g \mathbf{S}^g$$

$\mathbb{M}^g$ is sometimes called the mass matrix and $\mathbb{K}^g$ the stiffness matrix, although this nomenclature only applies to elasticity problems.

Mass matrix computation

$$\mathbb{M}^g_{ij} = \int_\Omega \phi^g_i(x)\phi^g_j(x)\,dx = \sum_{e=1}^{E}\int_{\Omega_e}\phi^g_i(x)\phi^g_j(x)\,dx$$

Because the basis functions have a local support, the mass matrix is naturally sparse: Consider one shape function ϕ^g_i related to the global point $x^g_i = a_e = b_{e-1}$. The support of ϕ^g_i is restricted to the two elements Ω_{e-1} and Ω_e. The only other shape functions that have a common support are ϕ^g_{i-1} and ϕ^g_{i+1}. So, the only non-zero terms on the i-th line of matrix $\mathbb{M}$ are: $\mathbb{M}^g_{i,i-1}$, $\mathbb{M}^g_{i,i}$ and $\mathbb{M}^g_{i,i+1}$. More precisely:

$$\begin{aligned}
\mathbb{M}^g_{i,i-1} &= \int_{\Omega_{e-1}}\phi^g_i(x)\phi^g_{i-1}(x)\,dx \\
\mathbb{M}^g_{i,i} &= \int_{\Omega_{e-1}}\phi^g_i(x)\phi^g_i(x)\,dx + \int_{\Omega_e}\phi^g_i(x)\phi^g_i(x)\,dx \\
\mathbb{M}^g_{i,i+1} &= \int_{\Omega_e}\phi^g_i(x)\phi^g_{i+1}(x)\,dx
\end{aligned}$$

Using the local basis functions, ϕ^e_a and ϕ^e_b, we get:

$$\begin{aligned}
\mathbb{M}^g_{i,i-1} &= \int_{\Omega_{e-1}}\phi^{e-1}_b(x)\phi^{e-1}_a(x)\,dx \\
\mathbb{M}^g_{i,i} &= \int_{\Omega_{e-1}}\phi^{e-1}_b(x)\phi^{e-1}_b(x)\,dx + \int_{\Omega_e}\phi^e_a(x)\phi^e_a(x)\,dx \\
\mathbb{M}^g_{i,i+1} &= \int_{\Omega_e}\phi^e_a(x)\phi^e_b(x)\,dx
\end{aligned}$$

Mass matrix computation

Next, we recall that the local basis functions are:

$$\begin{cases} \phi_a^e(x) & = & \frac{x-b_e}{a_e-b_e} \\ \phi_b^e(x) & = & \frac{x-a_e}{b_e-a_e} \end{cases}$$

So the mass matrix coefficients are:

$$\mathbb{M}^g_{i,i-1} = \int_{a_{e-1}}^{b_{e-1}} \left(\frac{x-a_{e-1}}{b_{e-1}-a_{e-1}}\right)\left(\frac{x-b_{e-1}}{a_{e-1}-b_{e-1}}\right) dx$$

$$\mathbb{M}^g_{i,i} = \int_{a_{e-1}}^{b_{e-1}} \left(\frac{x-a_{e-1}}{b_{e-1}-a_{e-1}}\right)^2 dx + \int_{a_e}^{b_e} \left(\frac{x-b_e}{a_e-b_e}\right)^2 dx$$

$$\mathbb{M}^g_{i,i+1} = \int_{a_e}^{b_e} \left(\frac{x-b_e}{a_e-b_e}\right)\left(\frac{x-a_e}{b_e-a_e}\right) dx$$

Those coefficients can be computed easily, it is just a matter of integrating second order polynomials. If we look closely, the integrals look always the same (only the corners a_e and b_e change), sometimes they appear alone and sometimes they have to be added. This suggests a more clever way to present (and to perform) the calculations.

Mass matrix computation

A clever idea is to introduce a reference element and a mapping from the reference element to each element of the mesh.

⏮ Back to some material that we have overlooked

On the reference element, we also introduce a polynomial basis. Here we keep the simplest one: linear basis functions. We define two basis functions as the shape functions related to the corners $\{-1; 1\}$:

$$\begin{cases} \phi_{-1}(\xi) & = & \frac{1-\xi}{2} \\ \phi_{+1}(\xi) & = & \frac{1+\xi}{2} \end{cases}$$

And we transport those two functions with the mapping $\mathcal{F}_e$ to form a basis on each element:

$$\begin{cases} \phi_a^e(x) & = & \phi_{-1}\left(\mathcal{F}_e^{-1}(x)\right) & = & \phi_{-1}(\xi) \\ \phi_b^e(x) & = & \phi_{+1}\left(\mathcal{F}_e^{-1}(x)\right) & = & \phi_{+1}(\xi) \end{cases}$$

Mass matrix computation

Recall the mass matrix coefficients:

$$\begin{aligned}
\mathbb{M}^g_{i,i-1} &= \int_{\Omega_{e-1}} \phi_b^{e-1}(x)\phi_a^{e-1}(x)\,dx \\
\mathbb{M}^g_{i,i} &= \int_{\Omega_{e-1}} \phi_b^{e-1}(x)\phi_b^{e-1}(x)\,dx + \int_{\Omega_e} \phi_a^e(x)\phi_a^e(x)\,dx \\
\mathbb{M}^g_{i,i+1} &= \int_{\Omega_e} \phi_a^e(x)\phi_b^e(x)\,dx
\end{aligned}$$

We apply the change of variable $x = \mathcal{F}_e(\xi)$ to get back to the reference element:

$$\begin{aligned}
\mathbb{M}^g_{i,i-1} &= \int_{\hat\Omega} \phi_{+1}(\xi)\phi_{-1}(\xi)\,\alpha_{e-1}d\xi \\
\mathbb{M}^g_{i,i} &= \int_{\hat\Omega} \phi_{+1}(\xi)\phi_{+1}(\xi)\,\alpha_{e-1}d\xi + \int_{\hat\Omega} \phi_{-1}(\xi)\phi_{-1}(\xi)\,\alpha_e d\xi \\
\mathbb{M}^g_{i,i+1} &= \int_{\hat\Omega} \phi_{-1}(\xi)\phi_{+1}(\xi)\,\alpha_e d\xi
\end{aligned}$$

Next, we introduce the elementary mass matrix:

$$\mathbb{M}^e = \begin{pmatrix} \int_{\hat\Omega} \phi_{-1}(\xi)\phi_{-1}(\xi)\,\alpha_e d\xi & \int_{\hat\Omega} \phi_{-1}(\xi)\phi_{+1}(\xi)\,\alpha_e d\xi \\ \int_{\hat\Omega} \phi_{+1}(\xi)\phi_{-1}(\xi)\,\alpha_e d\xi & \int_{\hat\Omega} \phi_{+1}(\xi)\phi_{+1}(\xi)\,\alpha_e d\xi \end{pmatrix}$$

Mass matrix computation

Next, we introduce the elementary mass matrix:

$$\mathbb{M}^e = \begin{pmatrix} \int_{\hat{\Omega}} \phi_{-1}(\xi)\phi_{-1}(\xi)\,\alpha_e d\xi & \int_{\hat{\Omega}} \phi_{-1}(\xi)\phi_{+1}(\xi)\,\alpha_e d\xi \\ \int_{\hat{\Omega}} \phi_{+1}(\xi)\phi_{-1}(\xi)\,\alpha_e d\xi & \int_{\hat{\Omega}} \phi_{+1}(\xi)\phi_{+1}(\xi)\,\alpha_e d\xi \end{pmatrix}$$

and we combine all the elementary matrices to form a block diagonal matrix which we call the local mass matrix:

$$\mathbb{M}^\ell = \begin{pmatrix} \boxed{\mathbb{M}^1} & 0 & \cdots & 0 \\ 0 & \boxed{\mathbb{M}^2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \boxed{\mathbb{M}^E} \end{pmatrix}$$

The connection between this local matrix and the global one ($\mathbb{M}^g$) is given by … the connectivity matrix $\mathbb{Q}$: ⏮ Back to some material that we have overlooked

$$\mathbb{M}^g = \mathbb{Q}^t\, \mathbb{M}^\ell\, \mathbb{Q}$$

Stiffness matrix computation

Recall the coefficients of the stiffness matrix:

$$\mathbb{K}^g_{ij} = \int_\Omega k(x) \frac{d\phi^g_i(x)}{dx} \frac{d\phi^g_j(x)}{dx}\, dx \;=\; \sum_{e=1}^{E} \int_{\Omega_e} k(x) \frac{d\phi^g_i(x)}{dx} \frac{d\phi^g_j(x)}{dx}\, dx$$

Like the mass matrix, the only non-zero terms on row i are: $\mathbb{K}^g_{i,i-1}$, $\mathbb{K}^g_{i,i}$ and $\mathbb{K}^g_{i,i+1}$. Let us for example consider the last term:

$$\mathbb{K}^g_{i,i+1} \;=\; \int_{\Omega_e} k(x) \frac{d\phi^g_i(x)}{dx} \frac{d\phi^g_{i+1}(x)}{dx}\, dx \;=\; \int_{\Omega_e} k(x) \frac{d\phi^e_a(x)}{dx} \frac{d\phi^e_b(x)}{dx}\, dx$$

Apply the change of variable $x = \mathcal{F}_e(\xi)$:

$$\mathbb{K}^g_{i,i+1} \;=\; \int_{\hat\Omega} k(\mathcal{F}_e(\xi)) \frac{d\phi_{-1}(\xi)}{dx} \frac{d\phi_{+1}(\xi)}{dx}\, \alpha_e d\xi$$

Note that we need to compute spatial derivatives with respect to the space variable x of functions of the variable ξ. To proceed we use the chain rule. Let f be such function of ξ, we have:

$$\frac{df}{dx}(\xi) = \frac{df}{d\xi}(\xi)\, \frac{d\xi}{dx}(x) \;=\; f'(\xi)\, \xi'(x)$$

In our case, ξ is a linear function of x and $\xi'(x) = \frac{1}{\alpha_e}$. Finally, we obtain:

$$\mathbb{K}^g_{i,i+1} \;=\; \int_{\hat\Omega} k(\mathcal{F}_e(\xi))\, \phi'_{-1}(\xi)\phi'_{+1}(\xi)\, \frac{1}{\alpha_e} d\xi \qquad \text{with } \phi'_{-1}(\xi) = -\frac{1}{2} \text{ and } \phi'_{+1}(\xi) = \frac{1}{2}$$

Stiffness matrix computation

Next, we proceed like for the mass matrix. Consider the elementary stiffness matrix:

$$\mathbb{K}^e = \begin{pmatrix} \int_{\hat{\Omega}} k(\mathcal{F}_e(\xi))\phi'_{-1}(\xi)\phi'_{-1}(\xi)\,\frac{1}{\alpha_e}\,d\xi & \int_{\hat{\Omega}} k(\mathcal{F}_e(\xi))\phi'_{-1}(\xi)\phi'_{+1}(\xi)\,\frac{1}{\alpha_e}\,d\xi \\ \\ \int_{\hat{\Omega}} k(\mathcal{F}_e(\xi))\phi'_{+1}(\xi)\phi'_{-1}(\xi)\,\frac{1}{\alpha_e}\,d\xi & \int_{\hat{\Omega}} k(\mathcal{F}_e(\xi))\phi'_{+1}(\xi)\phi'_{+1}(\xi)\,\frac{1}{\alpha_e}\,d\xi \end{pmatrix}$$

and combine all the elementary matrices to form a block diagonal matrix which we call the local stiffness matrix:

$$\mathbb{K}^\ell = \begin{pmatrix} \boxed{\mathbb{K}^1} & 0 & \cdots & 0 \\ 0 & \boxed{\mathbb{K}^2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \boxed{\mathbb{K}^E} \end{pmatrix}$$

The connection between this local matrix and the global one ($\mathbb{K}^g$) is given by the connectivity matrix $\mathbb{Q}$:

$$\mathbb{K}^g = \mathbb{Q}^t\,\mathbb{K}^\ell\,\mathbb{Q}$$

System to solve

We look for the unknown vector $\mathbf{U}^g$ such that:

$$\mathbb{K}^g \mathbf{U}^g = \mathbb{M}^g \mathbf{S}^g$$

Formally we write:

$$\mathbf{U}^g = \left(\mathbb{K}^g\right)^{-1} \mathbb{M}^g \mathbf{S}^g$$

But in practice, the inverse of the matrix is never computed. There are two strategies to solve the problem:

1. Direct approaches: the matrix $\mathbb{K}^g$ is factored into a product of triangular matrices (for ex $\mathbb{L}\mathbb{U}$ decomposition, Cholevsky decomposition ($\mathbb{L}\mathbb{L}^t$), $\mathbb{Q}\mathbb{R}$ decomposition ...) and the system is solved by susbstitution.
   Direct methods can not be applied to very large problems because of the memory requirements needed to perform the decomposition.
2. Iterative approaches: we define a series of approximate solutions $\left(\mathbf{U}_n^g\right)_{n=1,\ldots}$ which converge to the solution $\mathbf{U}^g$. For example, $\mathbf{U}^g$ can be seen as the solution of a minimization problem:

$$\mathbf{U}^g = \underset{\mathbf{U}}{\mathrm{Min}} \left(\frac{1}{2}\mathbf{U}^t \mathbb{K}^g \mathbf{U} - \mathbf{U}^t \mathbb{M}^g \mathbf{S}^g\right)$$

   and a descent method (e.g. the gradient method, the conjugate gradient method) can be applied to define the successive approximate solutions:

$$\mathbf{U}_{n+1}^g \stackrel{def}{=} \mathbf{U}_n^g + \rho_n \, \mathbf{d}_n$$

   where $\mathbf{d}_n$ is a descent direction to follow and ρ_n is a step to take in the descent direction.

System to solve

When using iterative methods, the only computations to perform are matrix vector products, for example $\mathbb{K}^g \mathbf{V}^g$, where $\mathbf{V}^g$ is a vector in the global space. This writes:

$$\mathbb{K}^g \mathbf{V}^g \; = \; \mathbb{Q}^t \mathbb{K}^\ell \mathbb{Q} \, \mathbf{V}^g$$

This involves the following steps:

1. Compute the local vector $\mathbf{V}^\ell = \mathbb{Q} \, \mathbf{V}^g$. This is just a copy of the values of the global vector to define local values.
2. Compute the matrix vector product $\mathbb{K}^\ell \, \mathbf{V}^\ell$ at the local scale. This operation can be done independently on each element of the mesh. It involves elementary matrix vector products: $\mathbb{K}^e \, \mathbf{V}^e$ which can be done in parallel.
   The result of the matrix vector product $\mathbb{K}^\ell \, \mathbf{V}^\ell$ can be stored in the local vector $\mathbf{F}^\ell$.
3. Perform the assembly of the local vector to form the global vector $\mathbf{F}^g = \mathbb{Q}^t \mathbf{F}^\ell$.

Hands on

Check list

- Define the physical parameters: $[a, b]$, $k(x)$, $s(x)$.
- Define the mesh: choose E, a_e and b_e ; compute α_e ; build $\mathbb{Q}$.
- Compute the elementary mass and stiffness matrices:

$$\mathbb{M}^e = \begin{pmatrix} \int_{-1}^{+1} \frac{(1-\xi)^2}{4}\,\alpha_e d\xi & \int_{-1}^{+1} \frac{(1-\xi)(1+\xi)}{4}\,\alpha_e d\xi \\ \int_{-1}^{+1} \frac{(1+\xi)(1-\xi)}{4}\,\alpha_e d\xi & \int_{-1}^{+1} \frac{(1+\xi)^2}{4}\,\alpha_e d\xi \end{pmatrix}$$

Hands on

Check list

- Define the physical parameters: $[a, b]$, $k(x)$, $s(x)$.
- Define the mesh: choose E, a_e and b_e ; compute α_e ; build $\mathbb{Q}$.
- Compute the elementary mass and stiffness matrices:

$$\mathbb{M}^e = \begin{pmatrix} \int_{-1}^{+1} \frac{(1-\xi)^2}{4}\,\alpha_e d\xi & \int_{-1}^{+1} \frac{(1-\xi)(1+\xi)}{4}\,\alpha_e d\xi \\ \int_{-1}^{+1} \frac{(1+\xi)(1-\xi)}{4}\,\alpha_e d\xi & \int_{-1}^{+1} \frac{(1+\xi)^2}{4}\,\alpha_e d\xi \end{pmatrix}$$

This can be computed analytically:

$$\mathbb{M}^e = \begin{pmatrix} \left[\frac{\alpha_e}{4}\left(\frac{1}{3}\xi^3 - \xi^2 + \xi\right)\right]_{\xi=-1}^{\xi=+1} & \left[\frac{\alpha_e}{4}\left(-\frac{1}{3}\xi^3 + \xi\right)\right]_{\xi=-1}^{\xi=+1} \\ \left[\frac{\alpha_e}{4}\left(-\frac{1}{3}\xi^3 + \xi\right)\right]_{\xi=-1}^{\xi=+1} & \left[\frac{\alpha_e}{4}\left(\frac{1}{3}\xi^3 + \xi^2 + \xi\right)\right]_{\xi=-1}^{\xi=+1} \end{pmatrix}$$

$$\mathbb{M}^e = \begin{pmatrix} \frac{\alpha_e}{4}\left(\frac{2}{3} + 2\right) & \frac{\alpha_e}{4}\left(-\frac{2}{3} + 2\right) \\ \frac{\alpha_e}{4}\left(-\frac{2}{3} + 2\right) & \frac{\alpha_e}{4}\left(\frac{2}{3} + 2\right) \end{pmatrix} = \begin{pmatrix} \frac{2}{3}\alpha_e & \frac{1}{3}\alpha_e \\ \frac{1}{3}\alpha_e & \frac{2}{3}\alpha_e \end{pmatrix}$$

Hands on

Check list

- Define the physical parameters: $[a, b]$, $k(x)$, $s(x)$.
- Define the mesh: choose E, a_e and b_e ; compute α_e ; build $\mathbb{Q}$.
- Compute the elementary mass and stiffness matrices:

$$\mathbb{K}^e = \begin{pmatrix} \int_{-1}^{+1} k(\mathcal{F}_e(\xi))\frac{1}{4}\,\frac{1}{\alpha_e} d\xi & -\int_{-1}^{+1} k(\mathcal{F}_e(\xi))\frac{1}{4}\,\frac{1}{\alpha_e} d\xi \\ -\int_{-1}^{+1} k(\mathcal{F}_e(\xi))\frac{1}{4}\,\frac{1}{\alpha_e} d\xi & \int_{-1}^{+1} k(\mathcal{F}_e(\xi))\frac{1}{4}\,\frac{1}{\alpha_e} d\xi \end{pmatrix}$$

The computation of $\mathbb{K}^e$ depends on the knowledge of the diffusivity $k(x)$. If analytical, then the integral can be computed exactly. In general, we should resort to numerical integration Toolbox.

Here, we assume a simple case where k is constant on each element:
$k(x) = k_e, \quad$ for $x \in \Omega_e$, we have:

$$\mathbb{K}^e = \begin{pmatrix} \frac{k_e}{2\alpha_e} & -\frac{k_e}{2\alpha_e} \\ -\frac{k_e}{2\alpha_e} & \frac{k_e}{2\alpha_e} \end{pmatrix}$$

Hands on

Check list

- Define the physical parameters: $[a, b]$, $k(x)$, $s(x)$.
- Define the mesh: choose E, a_e and b_e ; compute α_e ; build $\mathbb{Q}$.
- Compute the elementary mass and stiffness matrices:
- Compute the local matrices $\mathbb{M}^\ell$ and $\mathbb{K}^\ell$
- Compute the global matrices $\mathbb{M}^g$ and $\mathbb{K}^g$
- Solve the system

Here we assume first that our problem is small enough that we can actually form the local and global matrices, and even invert the stiffness matrix.
But, in a second stage, the implementation should be more efficient and should only require to compute the matrix-vector products, without forming and storing the local and global matrices.

Steady diffusion: Verification

Find $u(x)$ such that

$$\begin{aligned} -\frac{d}{dx}\left(k(x)\frac{du}{dx}(x)\right) &= s(x) \quad \text{for } x \in \Omega \\ u(a) &= 0 \\ u(b) &= 0 \end{aligned} \tag{4}$$

Assume a constant diffusivity ($k(x) = k_0$) and a source function such that $s(x) = g''(x)$ with $g(a) = g(b) = 0$. Then, the solution of Eq. 4 is

$$u(x) = -\frac{1}{k_0} g(x)$$

<u>Exercise:</u> Compare graphically the Finite Element solution u_h and the exact solution $u(x)$ for $g(x) = \sin\left(\pi \frac{x}{L}\right)$.

Steady diffusion: Verification

For a quantitative comparison, we use the L^2 norm:

$$\|u_h - u\| \;=\; \sqrt{\int_\Omega (u_h(x) - u(x))^2 \; dx}$$

Decomposing u_h and u on the global basis, we have:

$$u_h(x) - u(x) \;=\; \sum_{i=1}^{N_g} (u_h(x_i^g) - u(x_i^g)) \; \phi_i^g(x) \;\stackrel{def}{=}\; \sum_{i=1}^{N_g} V_i \, \phi_i^g(x)$$

Next

$$\|u_h - u\|^2 \;=\; \int_\Omega \left(\sum_{i=1}^{N_g} V_i \, \phi_i^g(x) \right) \left(\sum_{j=1}^{N_g} V_j \, \phi_j^g(x) \right) \; dx$$

$$\|u_h - u\|^2 \;=\; \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} V_i V_j \int_\Omega \phi_i^g(x) \phi_j^g(x) \, dx$$

$$\|u_h - u\|^2 \;=\; \mathbf{V}^t \mathbb{M}^g \mathbf{V}$$

where $\mathbf{V}$ is the global vector which entries are $V_i = (u_h(x_i^g) - u(x_i^g))$.

Exercise: Compute the norm $\|u_h - u\|$ and study its behaviour when the number of elements E is increased.

Steady diffusion, heterogeneous Dirichlet boundary conditions

Find $u(x)$ such that

$$
\begin{aligned}
-\frac{d}{dx}\left(k(x)\frac{du}{dx}(x)\right) &= s(x) \quad \text{for } x \in \Omega \\
u(a) &= u_a \\
u(b) &= u_b
\end{aligned}
\tag{4}
$$

Idea: define $u_0(x)$ such that $u_0(a) = u_a$ and $u_0(b) = u_b$, for example:

$$
u_0(x) = \frac{x-b}{a-b}\,u_a + \frac{x-a}{b-a}\,u_b
$$

Next, define $u^\star(x) = u(x) - u_0(x)$ and rewrite Eq. 4

Find $u^\star(x)$ solution of the steady diffusion equation with homogeneous Dirichlet BC

$$
\begin{aligned}
-\frac{d}{dx}\left(k(x)\frac{du^\star}{dx}(x)\right) &= s(x) + \frac{d}{dx}\left(k(x)\frac{du_0}{dx}(x)\right) \quad \text{for } x \in \Omega \\
u^\star(a) &= 0 \\
u^\star(b) &= 0
\end{aligned}
\tag{5}
$$

The term involving u_0 can be seen as an additional source term. Note that the choice of u_0 is not unique but the solution u is.

Steady diffusion, mixed boundary conditions

Find $u(x)$ such that

$$-\frac{d}{dx}\left(k(x)\frac{du}{dx}(x)\right) = s(x) \quad \text{for } x \in \Omega \tag{6}$$
$$u(a) = 0$$
$$k(b)\frac{du}{dx}(b) = \varphi_b$$

Functional spaces (only Dirichlet BC are enforced)

$$L^2(\Omega) = \left\{ w : \int_\Omega w^2(x)dx < \infty \right\}$$
$$\mathcal{W} = \left\{ w \in L^2(\Omega) : \frac{dw}{dx} \in L^2(\Omega), w(a) = 0 \right\}$$

Multiply Eq. (6) by w in $\mathcal{W}$, integrate over Ω and integrate by parts:

$$\int_\Omega k(x)\frac{du}{dx}(x)\frac{dw}{dx}(x)\,dx - \left[k(s)\frac{du}{dx}(s)w(s)\right]_{s=a}^{s=b} = \int_\Omega s(x)w(x)\,dx$$

Steady diffusion, mixed boundary conditions

Find $u(x)$ such that

$$\begin{aligned} -\frac{d}{dx}\left(k(x)\frac{du}{dx}(x)\right) &= s(x) \quad \text{for } x \in \Omega \\ u(a) &= 0 \\ k(b)\frac{du}{dx}(b) &= \varphi_b \end{aligned} \tag{6}$$

Multiply Eq. (6) by w in $\mathcal{W}$, integrate over Ω and integrate by parts:

$$\int_\Omega k(x)\frac{du}{dx}(x)\frac{dw}{dx}(x)\,dx - \left[k(s)\frac{du}{dx}(s)w(s)\right]_{s=a}^{s=b} = \int_\Omega s(x)w(x)\,dx$$

Because $w(a) = 0$, the boundary term reduces to $k(b)\frac{du}{dx}(b)w(b)$. To "impose" the boundary condition at $s = b$, we simply replace the flux by its value φ_b, the problem reads:

Find $u(x)$ such that forall w in $\mathcal{W}$

$$\int_\Omega k(x)\frac{du}{dx}(x)\frac{dw}{dx}(x)\,dx - \varphi_b w(b) = \int_\Omega s(x)w(x)\,dx$$

Steady diffusion, mixed boundary conditions

Find $u(x)$ such that (strong form)

$$\begin{aligned} -\frac{d}{dx}\left(k(x)\frac{du}{dx}(x)\right) &= s(x) \quad \text{for } x \in \Omega \\ u(a) &= 0 \\ k(b)\frac{du}{dx}(b) &= \varphi_b \end{aligned} \tag{6}$$

Find $u(x)$ such that forall w in $\mathcal{W}$ (variational form)

$$\int_\Omega k(x)\frac{du}{dx}(x)\frac{dw}{dx}(x)\,dx - \varphi_b w(b) = \int_\Omega s(x)w(x)\,dx \tag{7}$$

Again, the strong and weak forms are equivalent (admitted). In particular, the solution of the weak variational form does satisfy the Neumann BC at $s = b$. Neumann BC are said to be natural in the FEM. On the contrary, Dirichlet BC are said to be essential, in the sense that they have to be enforced through explicit requirements for the basis functions.

Exercise1: implement the case of mixed homogeneous Dirichlet-Neuman boundary conditions ($u(a) = \varphi_b = 0$). Compare with the case of homogeneous Dirichlet boundary conditions ($u(a) = u(b) = 0$).

Exercise2: implement the case of mixed Dirichlet-Neuman boundary conditions with $u(a) = 0$ and $\varphi_b \neq 0$.

Unsteady diffusion, homogeneous Dirichlet conditions

Find $u(x,t)$ such that

$$
\begin{aligned}
\frac{\partial u}{\partial t}(x,t) - \frac{\partial}{\partial x}\left(k(x)\frac{\partial u}{\partial x}(x,t)\right) &= s(x,t) \quad \text{for } x \in \Omega \text{ and } t \in [0,T] \qquad (8)\\
u(a,t) &= 0 \quad \forall t \in [0,T]\\
u(b,t) &= 0 \quad \forall t \in [0,T]\\
u(x,0) &= u_0(x) \quad \forall x \in \Omega
\end{aligned}
$$

Variational formulation (in space)

$$
\begin{aligned}
L^2(\Omega) &= \left\{ w : \int_\Omega w^2(x)dx < \infty \right\}\\
\mathcal{W} = H_0^1(\Omega) &= \left\{ w \in L^2(\Omega) : \frac{dw}{dx} \in L^2(\Omega), w(a) = w(b) = 0 \right\}
\end{aligned}
$$

For all times t, find $u(x,t)$ such that forall w in $\mathcal{W}$

$$
\int_\Omega \frac{\partial u}{\partial t}(x,t)w(x)\,dx + \int_\Omega k(x)\frac{\partial u}{\partial x}(x,t)\frac{dw}{dx}(x)\,dx = \int_\Omega s(x,t)w(x)\,dx \qquad (9)
$$

Unsteady diffusion: discretization

Find $u(x,t)$ such that forall w in $\mathcal{W}$

$$\int_\Omega \frac{\partial u}{\partial t}(x,t)w(x)\,dx + \int_\Omega k(x)\frac{\partial u}{\partial x}(x,t)\frac{dw}{dx}(x)\,dx = \int_\Omega s(x,t)w(x)\,dx \qquad (10)$$

1. For each time t we decompose the unknown $u(x,t)$, its time derivative $\frac{\partial u}{\partial t}(x,t)$ and the source term $s(x,t)$ onto the global basis

$$u(x,t) = \sum_{i=1}^{N_g} u(x_i^g,t)\,\phi_i^g(x) \quad ; \quad \frac{\partial u}{\partial t}(x,t) = \sum_{i=1}^{N_g} \frac{\partial u}{\partial t}(x_i^g,t)\,\phi_i^g(x)$$

$$s(x,t) = \sum_{i=1}^{N_g} s(x_i^g,t)\,\phi_i^g(x)$$

2. we write the variational formulation for each basis function ϕ_j^g

$$\sum_{i=1}^{N_g} \frac{\partial u}{\partial t}(x_i^g,t) \int_\Omega \phi_i^g(x)\phi_j^g(x)\,dx + \sum_{i=1}^{N_g} u(x_i^g,t) \int_\Omega k(x)\frac{d\phi_i^g(x)}{dx}\frac{d\phi_j^g(x)}{dx}\,dx = \sum_{i=1}^{N_g} s(x_i^g,t) \int_\Omega \phi_i^g(x)\phi_j^g(x)\,dx$$

Unsteady diffusion: discretization

Find $u_i^g(x,t)$ and $\partial u/\partial t(x_i^g,t)$ such that forall ϕ_j^g in $\mathcal{W}$

$$\sum_{i=1}^{N_g} \frac{\partial u}{\partial t}(x_i^g,t) \int_\Omega \phi_i^g(x)\phi_j^g(x)\,dx \quad + \quad \sum_{i=1}^{N_g} u(x_i^g,t) \int_\Omega k(x) \frac{d\phi_i^g(x)}{dx}\frac{d\phi_j^g(x)}{dx}\,dx$$
$$= \sum_{i=1}^{N_g} s(x_i^g,t) \int_\Omega \phi_i^g(x)\phi_j^g(x)\,dx \qquad (10)$$

- Let $\mathbf{U}^g(t)$ denote the time-dependent vector which stores the entries $u(x_i^g,t)$
- Let $\frac{d\mathbf{U}^g}{dt}(t)$ denote the time-dependent vector which stores the entries $\frac{\partial u}{\partial t}(x_i^g,t)$
- Let $\mathbf{S}^g(t)$ denote the time-dependent vector which stores the entries $s(x_i^g,t)$
- Let $\mathbf{U}_0^g$ denote the vector which stores the entries $u_0(x_i^g)$

Eq. (10) is the j-th row of the system of ordinary differential equations:

$$\left\{ \begin{array}{rcl} \mathbb{M}^g \frac{d\mathbf{U}^g}{dt}(t) + \mathbb{K}^g\mathbf{U}^g(t) & = & \mathbb{M}^g\mathbf{S}(t) \\ \mathbf{U}^g(0) & = & \mathbf{U}_0^g \end{array} \right. \qquad (11)$$

Unsteady diffusion: time integration

Find $\mathbf{U}^g(t)$ such that

$$\left\{\begin{array}{rcl} \mathbb{M}^g \frac{d\mathbf{U}^g}{dt}(t) + \mathbb{K}^g \mathbf{U}^g(t) & = & \mathbb{M}^g \mathbf{S}(t) \\ \mathbf{U}^g(0) & = & \mathbf{U}_0^g \end{array}\right. \tag{12}$$

To solve the system in time, we introduce a series of discrete time steps: $t_0 = 0, \ldots, t_n, t_{n+1} = t_n + \Delta t, \ldots t_N = T$, where Δt is the time step, which we assume to be constant.

Let $\mathbf{U}_n^g$ and $\mathbf{S}_n$ respectively denote the approximate solution and source term at time t_n. The simplest way to compute the solution at the next time step is to apply an explicit Euler method:

$$\mathbf{U}_{n+1}^g = \mathbf{U}_n^g + \Delta t \frac{d\mathbf{U}^g}{dt}(t_n)$$

The time derivative $\frac{d\mathbf{U}^g}{dt}(t_n)$ can be computed (or approximated) from Eq. 12. For example if we know the inverse of the mass matrix, the Euler method reads:

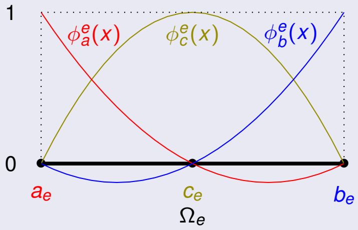
$$\begin{array}{rcl} \mathbf{U}_{n+1}^g & = & \mathbf{U}_n^g + \Delta t \left(\mathbf{S}_n - \left(\mathbb{M}^g\right)^{-1} \mathbb{K}^g \mathbf{U}_n^g \right) \\ \mathbf{U}_{n+1}^g & = & \left(\mathbb{I}^g - \Delta t \left(\mathbb{M}^g\right)^{-1} \mathbb{K}^g \right) \mathbf{U}_n^g + \Delta t \, \mathbf{S}_n \end{array}$$

where $\mathbb{I}^g$ is the identity matrix . The matrix $\mathbb{P}^g = \left(\mathbb{I}^g - \Delta t \left(\mathbb{M}^g\right)^{-1} \mathbb{K}^g \right)$ can be called the propagator matrix.

Exercise: implement the explicit Euler method on the example of your choice (source and initial conditions). Check the stability of the method.

Quadratic Finite Elements

Local polynomial basis



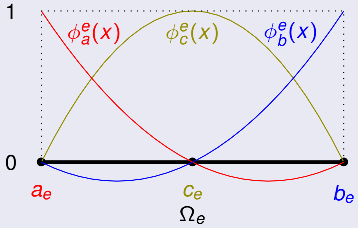
- Add a central point c_e
- Define second order shape functions: $\phi_a^e(x)$, $\phi_c^e(x)$, $\phi_b^e(x)$.

Any function ψ defined on Ω_e can be approximated through quadratic interpolation using the local basis:

$$\psi(x) \simeq \psi(a_e)\phi_a^e(x) + \psi(c_e)\phi_c^e(x) + \psi(b_e)\phi_b^e(x)$$

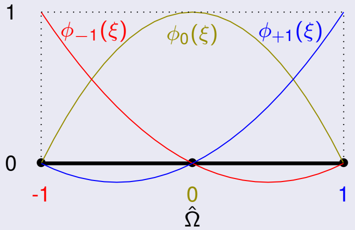
Quadratic Finite Elements

Local polynomial basis



- Add a central point c_e
- Define second order shape functions: $\phi_a^e(x)$, $\phi_c^e(x)$, $\phi_b^e(x)$.
- ... in practice they are defined on the reference element ...

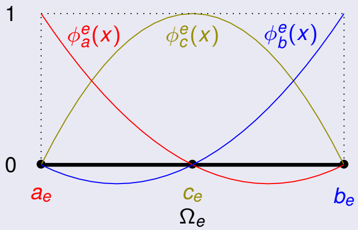
Local polynomial basis on the reference element



- $\phi_{-1}(\xi) = \xi(\xi - 1)/2$
- $\phi_0(\xi) = -(\xi + 1)(\xi - 1)$
- $\phi_{+1}(\xi) = \xi(\xi + 1)/2$

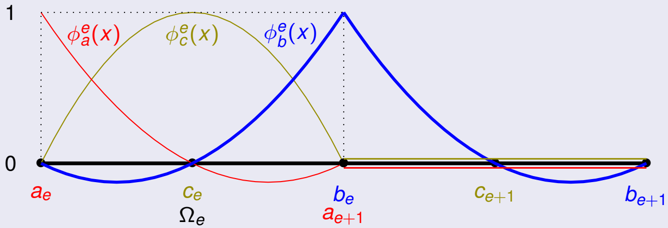
Quadratic Finite Elements

Local polynomial basis



- Add a central point c_e
- Define second order shape functions: $\phi_a^e(x)$, $\phi_c^e(x)$, $\phi_b^e(x)$.
- . . . and extended to continuous global basis functions . . .

Global polynomial basis



Quadratic Finite Elements

Elementary mass matrix

$$\mathbb{M}^e = \begin{pmatrix} \int_{\hat\Omega} \phi_{-1}(\xi)\phi_{-1}(\xi)\,\alpha_e d\xi & \int_{\hat\Omega} \phi_{-1}(\xi)\phi_{0}(\xi)\,\alpha_e d\xi & \int_{\hat\Omega} \phi_{-1}(\xi)\phi_{+1}(\xi)\,\alpha_e d\xi \\ \int_{\hat\Omega} \phi_{0}(\xi)\phi_{-1}(\xi)\,\alpha_e d\xi & \int_{\hat\Omega} \phi_{0}(\xi)\phi_{0}(\xi)\,\alpha_e d\xi & \int_{\hat\Omega} \phi_{0}(\xi)\phi_{+1}(\xi)\,\alpha_e d\xi \\ \int_{\hat\Omega} \phi_{+1}(\xi)\phi_{-1}(\xi)\,\alpha_e d\xi & \int_{\hat\Omega} \phi_{+1}(\xi)\phi_{0}(\xi)\,\alpha_e d\xi & \int_{\hat\Omega} \phi_{+1}(\xi)\phi_{+1}(\xi)\,\alpha_e d\xi \end{pmatrix}$$

Elementary stiffness matrix

$$\mathbb{K}^e = \begin{pmatrix} \int_{\hat\Omega} k_e(\xi)\phi'_{-1}(\xi)\phi'_{-1}(\xi)\,\frac{d\xi}{\alpha_e} & \int_{\hat\Omega} k_e(\xi)\phi'_{-1}(\xi)\phi'_{0}(\xi)\,\frac{d\xi}{\alpha_e} & \int_{\hat\Omega} k_e(\xi)\phi'_{-1}(\xi)\phi'_{+1}(\xi)\,\frac{d\xi}{\alpha_e} \\ \int_{\hat\Omega} k_e(\xi)\phi'_{0}(\xi)\phi'_{-1}(\xi)\,\frac{d\xi}{\alpha_e} & \int_{\hat\Omega} k_e(\xi)\phi'_{0}(\xi)\phi'_{0}(\xi)\,\frac{d\xi}{\alpha_e} & \int_{\hat\Omega} k_e(\xi)\phi'_{0}(\xi)\phi'_{+1}(\xi)\,\frac{d\xi}{\alpha_e} \\ \int_{\hat\Omega} k_e(\xi)\phi'_{+1}(\xi)\phi'_{-1}(\xi)\,\frac{d\xi}{\alpha_e} & \int_{\hat\Omega} k_e(\xi)\phi'_{+1}(\xi)\phi'_{0}(\xi)\,\frac{d\xi}{\alpha_e} & \int_{\hat\Omega} k_e(\xi)\phi'_{+1}(\xi)\phi'_{+1}(\xi)\,\frac{d\xi}{\alpha_e} \end{pmatrix}$$

with $k_e(\xi) = k(\mathcal{F}_e(\xi))$.

$$\begin{cases} \phi_{-1}(\xi) & = & \xi(\xi-1)/2 \\ \phi_0(\xi) & = & -(\xi+1)(\xi-1) \\ \phi_{+1}(\xi) & = & \xi(\xi+1)/2 \end{cases} \qquad \begin{cases} \phi'_{-1}(\xi) & = & \xi - 1/2 \\ \phi'_0(\xi) & = & -2\xi \\ \phi'_{+1}(\xi) & = & \xi + 1/2 \end{cases}$$

$\mathbb{M}^e$ and $\mathbb{K}^e$ (for piecewise constant k) can be computed analytically, by hand or using a symbolic calculation tool (e.g. https://www.wolframalpha.com).

Higher order Finite Elements

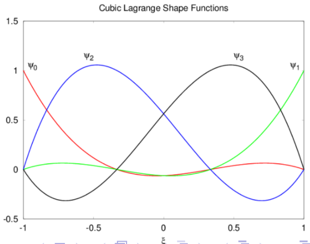
- To define a Finite Element approximation with order N polynomials, we define a set of $(N+1)$ distinct points, $(\mu_i)_{i=0\ldots N}$ on the reference element $\hat{\Omega}$. To ease the implementation of the continuity conditions between elements, we require that the first and last points are the corners of the elements:

$$-1 = \mu_0 < \mu_1 < \ldots < \mu_N = +1$$

- The shape functions are defined as the Lagrange interpolants of those points:

$$\phi_i(\xi) = \frac{\prod_{\substack{j=0\\ j\neq i}}^{N}(\xi - \mu_j)}{\prod_{\substack{j=0\\ j\neq i}}^{N}(\mu_i - \mu_j)} = \frac{(\xi-\mu_0)\ldots(\xi-\mu_{i-1})(\xi-\mu_{i+1})\ldots(\xi-\mu_N)}{(\mu_i-\mu_0)\ldots(\mu_i-\mu_{i-1})(\mu_i-\mu_{i+1})\ldots(\mu_i-\mu_N)}$$

Exercise: compute and plot the shape functions for evenly distributed points (μ_i) and N>2.



To compute the coefficients of the elementary mass and stiffness matrices using a numerical integration formula based on the K integration points $(\nu_k)_{k=1\dots K}$ and integration weights $(\rho_k)_{k=1\dots K}$:

$$
\begin{aligned}
\mathbb{M}^e_{ij} &= \int_{\hat{\Omega}} \phi_i(\xi)\phi_j(\xi)\alpha_e d\xi \simeq \alpha_e \sum_{k=1}^{K} \rho_k \phi_i(\nu_k)\phi_j(\nu_k) \\
\mathbb{K}^e_{ij} &= \int_{\hat{\Omega}} \phi_i'(\xi)\phi_j'(\xi)\frac{d\xi}{\alpha_e} \simeq \frac{1}{\alpha_e} \sum_{k=1}^{K} \rho_k \phi_i'(\nu_k)\phi_j'(\nu_k)
\end{aligned}
$$

- It is of interest to use the integration points as the set of points to define the Lagrange interpolants. For example, using the integration points involved in Gauss numerical quadrature yield polynomial bases with very good approximation properties (see Toolbox).
  This is the basis of the spectral element method.

Practices

Mandatory

1. Do all the Exercises of the present document

Choose among the following:

2. Implement a second-order Finite Element approximation of the 1D steady diffusion equation, compare with the analytical solution. Study the L^2 norm of the error (between the FE and the exact solution) when the number of elements is increased.
3. Implement a second-order Finite Element approximation of the 1D unsteady diffusion equation. Compare with a Finite Difference approximation of the same problem.
4. Implement a Finite Element approximation of the 1D elastodynamics equation (wave equation). Compare with a Finite Difference approximation of the same problem.

Instructions

Write a report and attach the codes you have written. Send everything to: `Emmanuel.Chaljub@univ-grenoble-alpes.fr` by Monday, November 7, 2022.

Toolbox: Integration by parts

back

scalar functions Let f and g stand for 2 scalar functions of $x \in [a, b]$.

$$\begin{aligned}
(fg)'(x) &= f'(x)g(x) + f(x)g'(x) \\
\int_a^b (fg)'(x)dx &= \int_a^b \left(f'(x)g(x) + f(x)g'(x)\right) dx \\
[fg(s)]_{s=a}^{s=b} = f(b)g(b) - f(a)g(a) &= \int_a^b f'(x)g(x)dx + \int_a^b f(x)g'(x)dx
\end{aligned}$$

Integration by parts

$$\int_a^b f'(x)g(x)dx = -\int_a^b f(x)g'(x)dx + [fg(s)]_{s=a}^{s=b}$$

Higher dimensions Let $\mathbf{v}$ stand for a vectorial function and w a scalar function of $\mathbf{x} \in \Omega \subset \mathbb{R}^d$. The surface is $\partial\Omega$, with unit normal vector $\mathbf{n}$.
∇w is the gradient of w, $\nabla \cdot \mathbf{v}$ is the divergence of $\mathbf{v}$.

Integration by parts

$$\int_\Omega \nabla \cdot \mathbf{v}(\mathbf{x})w(\mathbf{x})d\mathbf{x} = -\int_\Omega \mathbf{v}(\mathbf{x}) \cdot \nabla w(\mathbf{x})d\mathbf{x} + \int_{\partial\Omega} \mathbf{v} \cdot \mathbf{n}(s)w(s)ds$$

Toolbox: Integration by parts

back

Higher dimensions Let $\mathbf{v}$ stand for a vectorial function and w a scalar function of $\mathbf{x} \in \Omega \subset \mathbb{R}^d$. The surface is $\partial\Omega$, with unit normal vector $\mathbf{n}$.
∇w is the gradient of w, $\nabla \cdot \mathbf{v}$ is the divergence of $\mathbf{v}$.

Integration by parts

$$\int_\Omega \nabla \cdot \mathbf{v}(\mathbf{x}) w(\mathbf{x}) d\mathbf{x} = - \int_\Omega \mathbf{v}(\mathbf{x}) \cdot \nabla w(\mathbf{x}) d\mathbf{x} + \int_{\partial\Omega} \mathbf{v} \cdot \mathbf{n}(s) w(s) ds$$

Particular case: $\mathbf{v}(\mathbf{x}) = \nabla u(\mathbf{x})$

Green's first identity

$$\int_\Omega \Delta u(\mathbf{x}) w(\mathbf{x}) d\mathbf{x} = - \int_\Omega \nabla u(\mathbf{x}) \cdot \nabla w(\mathbf{x}) d\mathbf{x} + \int_{\partial\Omega} \nabla u \cdot \mathbf{n}(s) w(s) ds$$

Δu is the Laplacian of u, $\nabla u \cdot \mathbf{n}$ is the normal derivative of u (i.e. derivative along the direction orthogonal to $\partial\Omega$).
Green's first identity = divergence theorem applied to $w \nabla u$.

Toolbox: Numerical Integration

back

Problem

- We know the values f_i of a given function f at some discrete positions (or time or whatever input variable) x_i.
- We want to compute the integral of f at those points . . .
- . . . without having to approximate the function.

Toolbox: Numerical Integration

back

Solution: numerical integration

- We seek a formula like below

$$\int_\Omega f(x)\, dx \simeq \sum_{i=0}^{n} \rho_i\, f(\xi_i)$$

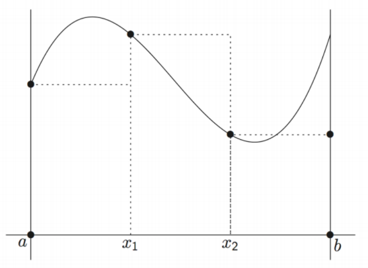
- The ξ_i are called integration points.
- The ρ_i are called integration weights.
- The accuracy of the formula can be estimated with known functions, for example polynomials: if we replace f with a polynomial of order p, for which p value is the integration exact?

back

Exemple 1: rectangles

- The interval $[a, b]$ is split into segments (of same size):

$$H = (b-a)/n, \quad x_0 = a, \quad x_{i+1} = x_i + H, \quad x_n = b$$



We replace f by a constant on each segment.

- Left Formula

$$\int_a^b f(x)dx \simeq \sum_{i=0}^{n-1} H\, f(x_i)$$

- Right Formula

$$\int_a^b f(x)dx \simeq \sum_{i=1}^{n} H\, f(x_i)$$

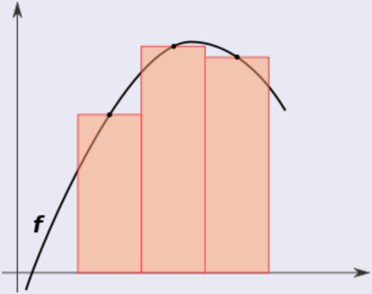
Toolbox: Numerical Integration

back

Exemple 1: rectangles

- The interval $[a, b]$ is split into segments (of same size):

$$H = (b-a)/n, \quad x_0 = a, \quad x_{i+1} = x_i + H, \quad x_n = b$$



We replace f by a constant on each segment.
⚠: requires to evaluate f at the mid-point

Mid point Formula

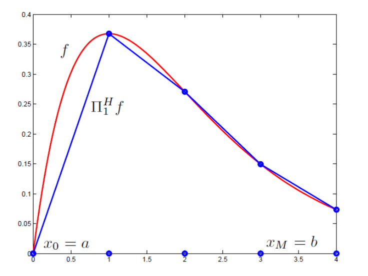
$$\int_a^b f(x)dx \simeq \sum_{i=0}^{n-1} H\, f(\hat{x}_i), \quad \hat{x}_i = \frac{x_i + x_{i+1}}{2}$$

back

Exemple 2: trapezoidal rule

- The interval $[a, b]$ is split into segments (of same size):

$$H = (b-a)/n, \quad x_0 = a, \quad x_{i+1} = x_i + H, \quad x_n = b$$



We replace f by a line on each segment.

Trapezoidal Rule

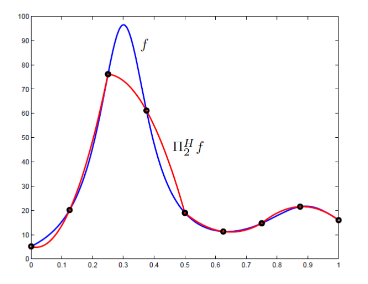
$$\int_a^b f(x)dx \simeq \frac{H}{2} f(x_a) + \sum_{i=1}^{n-1} H f(x_i) + \frac{H}{2} f(x_b)$$

 back

Exemple 3: Simpson method

- The interval $[a, b]$ is split into segments (of same size):

$$H = (b-a)/n, \quad x_0 = a, \quad x_{i+1} = x_i + H, \quad x_n = b$$



We replace f by a parabola on each segment.
⚠: requires to evaluate f at the mid-point

Simpson Formula

$$\int_a^b f(x)dx \simeq \frac{H}{6} \sum_{i=0}^{n-1} \left(f(x_{i-1}) + 4\, f(\hat{x}_i) + f(x_i)\right), \quad \hat{x}_i = \frac{x_i + x_{i+1}}{2}$$

back

Integration error

Assume that the function at hand is regular enough (derivatives exist to any order and are continuous)

- Rectangles (Left or Right): $\texttt{error} \propto H$
- Rectangles (Mid-point): $\texttt{error} \propto H^2$
- Trapezoidal Rule: $\texttt{error} \propto H^2$
- Simpson Formula: $\texttt{error} \propto H^4$

In practice, we will therefore consider a large number of segments (to get H small) and we will choose the best formula according to its accuracy (Trapezoidal or Simpson).

To reach higher accuracy with less integration points, we need to resort to Gauss type methods. . .

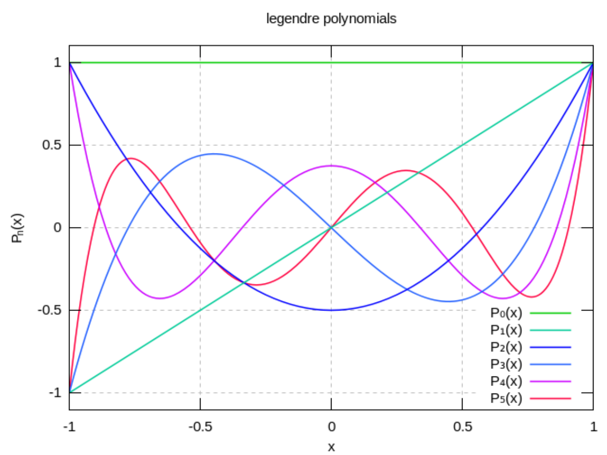
Toolbox: Numerical Integration

back

Gauss numerical quadrature

back to High-order FEM

- We consider the $(N+1)$ integration points $(\xi)_{i=0}^{N}$ which are the zeros of the Legendre polynomials of order N on [-1,1]



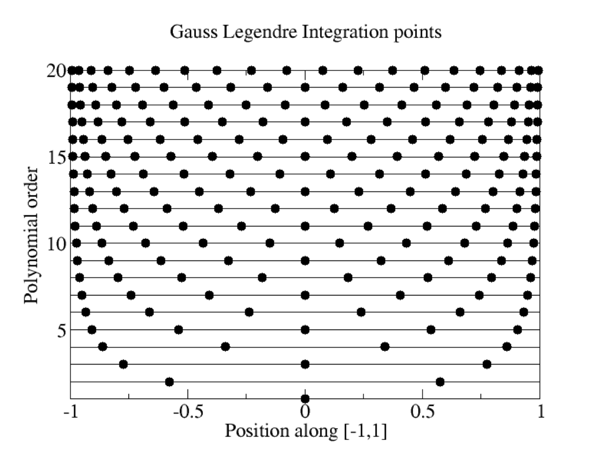
Toolbox: Numerical Integration

back

Gauss numerical quadrature

back to High-order FEM

- We consider the $(N+1)$ integration points $(\xi)_{i=0}^{N}$ which are the zeros of the Legendre polynomals of order N on [-1,1]



Toolbox: Numerical Integration

back

Gauss numerical quadrature

back to High-order FEM

- We consider the $(N+1)$ integration points $(\xi)_{i=0}^{N}$ which are the zeros of the Legendre polynomals of order N on [-1,1]
- There exist $(N+1)$ weights $(\omega)_{i=0}^{N}$ such that the integration formula

$$\int_{-1}^{1} f(x)dx = \sum_{i=0}^{N} \omega_i \, f(\xi_i)$$

is exact for any polynomial of order $2N+1$.

Toolbox: Numerical Integration

back

Gauss numerical quadrature

back to High-order FEM

- We consider the $(N+1)$ integration points $(\xi)_{i=0}^{N}$ which are the zeros of the Legendre polynomals of order N on [-1,1]
- There exist $(N+1)$ weights $(\omega)_{i=0}^{N}$ such that the integration formula

$$\int_{-1}^{1} f(x)dx = \sum_{i=0}^{N} \omega_i \, f(\xi_i)$$

is exact for any polynomial of order $2N+1$.