$\mathbb{P}^1$ - Finite Element Method for linear elasticity in 2D

16214-01 - Project: Numerical Methods for Partial Differential Equations | Fall Semester, 2025

University of Basel, Department of Mathematics and Computer Science
Prof. Dr. M. Grote | Dr. Romain Mottier

Deadline for submission: Friday, January 23, 2025

Abstract

The objective of this project is to implement $\mathbb{P}^1$-FEM to solve two-dimensional linear elasticity problems.

Table of Contents

1 Preliminaries

Notations. We use regular fonts for scalars, and boldface (resp. blackboard) fonts for vectors (resp. tensors), and for vector-valued (resp. tensor-valued) fields and spaces composed of such fields (*i.e.* $u \in \mathbb{R}$, $\mathbf{u} \in \mathbb{R}^d$, $\mathbb{u} \in \mathbb{R}^{d\times d}$). For a bounded and uniformly positive weight $\kappa \in L^\infty(\Omega)$, we define the κ-weighted L^2-inner products as

$$(u,v)_{L^2(\kappa;\Omega)} := \int_\Omega \kappa uv \,\mathrm{d}\Omega, \qquad (\mathbf{u},\mathbf{v})_{\boldsymbol{L}^2(\kappa;\Omega)} := \int_\Omega \kappa\,\mathbf{u}\cdot\mathbf{v}\,\mathrm{d}\Omega, \qquad (\mathbb{u},\mathbb{v})_{\mathbb{L}^2(\kappa;\Omega)} := \int_\Omega \kappa\,\mathbb{u} : \mathbb{v}\,\mathrm{d}\Omega,$$

for all $u, v \in L^2(\Omega)$, all $\mathbf{u},\mathbf{v} \in \mathbf{L}^2(\Omega)$, and all $\mathbb{u},\mathbb{v} \in \mathbb{L}^2(\Omega)$. In particular we will use for fourth-order tensorial weights, $\mathbb{k} \in L^\infty(\Omega, \mathbb{R}^{d\times d\times d\times d})$,

$$(\mathbb{A},\mathbb{B})_{L^2(\mathbb{k};\Omega)} := \int_\Omega (\mathbb{k} : \mathbb{A}) : \mathbb{B}\,\mathrm{d}\Omega.$$

Context. Linear elasticity is a fundamental theory in solid mechanics that describes the deformation of elastic bodies under the action of external forces. This framework is widely used to predict the mechanical behavior of structures such as beams, plates, and shells in engineering applications. Linear elasticity provides the basis for the analysis and design of mechanical components in civil, aerospace, and mechanical engineering.

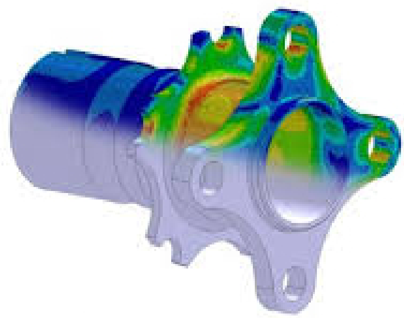

Figure 1: Example of stress distribution in a mechanical component

A classical quantity of interest is the displacement field $\mathbf{u} : \Omega \to \mathbb{R}^d$ [m] (where $d \in \{1,2,3\}$ denotes the spatial dimension considered), defined as a vector-valued function that assigns to each point $\mathbf{x} \in \Omega$ its infinitesimal displacement from the reference (undeformed) configuration. Linear elasticity is based on two fundamental assumptions:

1. **Infinitesimal strain theory (small deformations assumption):** $\|\nabla\mathbf{u}\| \ll 1$.
   The displacements $\mathbf{u}$ [m] are assumed to be much smaller than the characteristic dimensions of the body, so that its geometry and material properties (e.g., density, stiffness) remain essentially unchanged. Under this small-deformation assumption, the strain–displacement relation is linear, and higher-order terms in

the deformation gradient can be neglected. The corresponding infinitesimal strain tensor $\varepsilon(\mathbf{u})$ [ ] is defined as

$$\varepsilon(\mathbf{u}) := \nabla^{\text{SYM}}\mathbf{u} = \frac{1}{2}\big(\nabla\mathbf{u} + (\nabla\mathbf{u})^{\mathsf{T}}\big), \tag{1}$$

which measures the symmetric part of the displacement gradient and represents the linearized deformation at each material point.

2. **Hooke's law (linear constitutive relation between the stress and strain tensors).**
   For a linear, homogeneous, and isotropic elastic material, the Cauchy stress tensor σ [Pa] is related to the infinitesimal strain tensor $\varepsilon(\mathbf{u})$ through the Lamé parameters $\lambda > 0$, $\mu > 0$ (both in Pa) which characterize the elastic properties of the material. This fundamental law is written as:

$$\sigma(\mathbf{u}) := \mathbb{C} : \varepsilon(\mathbf{u}), \tag{2}$$

   where $\mathbb{C}$ [Pa] is the fourth-order isotropic Hooke tensor describing the material's stiffness.

   Remark 1 (Lamé parameters)**.** *The Lamé parameter μ is the shear modulus controlling the response to shear deformations, and λ is related to the material's compressibility.*

   Remark 2 (Symmetry of the Cauchy stress tensor)**.** *Component-wise, the Cauchy stress tensor is given by*

$$\sigma_{ij} = \sum_{k,l} C_{ijkl}\,\varepsilon_{kl}.$$

   If the elasticity tensor $\mathbb{C}$ satisfies the minor symmetries ($C_{ijkl} = C_{ijlk} = C_{jikl}$ *for all i, j, k, l) and the strain tensor ε is symmetric ($\varepsilon_{kl} = \varepsilon_{lk}$), then σ is symmetric as well. Indeed,*

$$\sigma_{ij} = \sum_{k,l} C_{ijkl}\varepsilon_{kl} = \sum_{k,l} C_{jikl}\varepsilon_{kl} = \sigma_{ji}.$$

   The major symmetry $C_{ijkl} = C_{klij}$ *is not needed for this result.*

Voigt's notation for the 2D isotropic linear elasticity. Since both the stress tensor σ and the strain tensor ε are symmetric, in 2D isotropic linear elasticity each of them has three independent components in two dimensions. In Voigt notation, these components are collected into column vectors as

$$\boldsymbol{\sigma} = \begin{pmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{xy} \end{pmatrix}, \qquad \boldsymbol{\varepsilon} = \begin{pmatrix} \varepsilon_{xx} \\ \varepsilon_{yy} \\ 2\varepsilon_{xy} \end{pmatrix}.$$

In this notation, the fourth-order elasticity tensor $\mathbb{C}$ can be represented by the 3×3 matrix

$$\mathbb{C} = \begin{pmatrix} \lambda + 2\mu & \lambda & 0 \\ \lambda & \lambda + 2\mu & 0 \\ 0 & 0 & \mu \end{pmatrix}, \text{ so that Hooke's law takes the matrix–vector form } \boldsymbol{\sigma} = \mathbb{C}\,\boldsymbol{\varepsilon}.$$

2 Continuous framework

Model problem. The strong formulation of the linear elasticity problem consists in finding the displacement field $\mathbf{u} : \Omega \to \mathbb{R}^d$ such that

$$\begin{aligned} -\nabla \cdot \sigma(\mathbf{u}) &= \mathbf{f} && \text{in } \Omega, \\ \mathbf{u} &= \mathbf{u}_D && \text{on } \Gamma_D, \\ \sigma(\mathbf{u}) \cdot \mathbf{n} &= \mathbf{t}_N && \text{on } \Gamma_N. \end{aligned} \tag{3}$$

where $\mathbf{f}$ $\left[\frac{\mathrm{N}}{\mathrm{m}^3}\right]$ is the body force density representing external forces per unit volume, $\mathbf{u}_D$ [m] denotes the prescribed displacement field on the Dirichlet boundary Γ_D and $\mathbf{t}_N$ $\left[\frac{\mathrm{N}}{\mathrm{m}^2}\right]$ is the prescribed traction acting on the Neumann boundary Γ_N.

In this section, we consider homogeneous Dirichlet boundary conditions ($\mathbf{u}_D = 0$) and nonhomogeneous Neumann boundary conditions. Then, we consider the following Hilbert spaces

$$H_0^1(\Omega) := \left\{ v \in H^1(\Omega) \;:\; v|_{\Gamma_D} = 0 \right\}, \qquad \boldsymbol{H}_\mathbf{0}^\mathbf{1}(\Omega) := [H_0^1(\Omega)]^d = \left\{ \mathbf{v} \in [H^1(\Omega)]^d \;:\; \mathbf{v}|_{\Gamma_D} = \mathbf{0} \right\}. \tag{4}$$

Question 1. Show that the weak formulation of the general linear elasticity problem in $\mathbb{R}^d$ is given by:
Find $\mathbf{u} \in \boldsymbol{H}_\mathbf{0}^\mathbf{1}(\Omega)$ such that,

$$(\varepsilon(\mathbf{u}) : \varepsilon(\mathbf{v}))_{\mathbb{L}^2(\mathbb{C};\Omega)} = (\mathbf{f}, \mathbf{v})_{\boldsymbol{L}^2(\Omega)} + (\mathbf{t}_N, \mathbf{v})_{\boldsymbol{L}^2(\Gamma_N)} \qquad \forall \mathbf{v} \in \boldsymbol{H}_\mathbf{0}^\mathbf{1}(\Omega).$$

Hint: Green's first identity for general tensor fields

Lemma 2.1. *Let $\Omega \subset \mathbb{R}^d$ be a domain with smooth boundary $\partial\Omega$, and let $\mathbf{T} : \Omega \to \mathbb{R}^{d\times d}$ be a sufficiently smooth tensor field and $\mathbf{v} : \Omega \to \mathbb{R}^d$ a sufficiently smooth vector field,*

$$(\nabla \cdot \mathbb{T}, \mathbf{v})_{\boldsymbol{L}^2(\Omega)} = (\mathbb{T} \cdot \mathbf{n}, \mathbf{v})_{\boldsymbol{L}^2(\partial\Omega)} - (\mathbb{T}, \nabla \mathbf{v})_{\mathbb{L}^2(\Omega)}. \tag{5}$$

Question 2. Show that in 2D the weak formulation for each of the scalar components of the displacement field $\mathbf{u} = (u_x, u_y)^\top$ is given by:

Find $u_x, u_y \in H_0^1(\Omega)$ such that,

$$\begin{aligned}
((\lambda + 2\mu)\partial_x u_x + \lambda \partial_y u_y, \partial_x v_x)_{L^2(\Omega)} + (\mu(\partial_y u_x + \partial_x u_y), \partial_y v_x)_{L^2(\Omega)} &= (f_x, v_x)_{L^2(\Omega)} + (t_{N,x}, v_x)_{L^2(\Gamma_N)}, \\
((\lambda + 2\mu)\partial_y u_y + \lambda \partial_x u_x, \partial_y v_y)_{L^2(\Omega)} + (\mu(\partial_y u_x + \partial_x u_y), \partial_x v_y)_{L^2(\Omega)} &= (f_y, v_y)_{L^2(\Omega)} + (t_{N,y}, v_y)_{L^2(\Gamma_N)},
\end{aligned}$$

$\forall v_x, v_y \in H_0^1(\Omega)$.

Hint

Use the Hooke's law for 2D isotropic linear material in Voigt notation.

3 Implementation of $\mathbb{P}^1$-FEM operators in 2D

Notations. In the following, algebraic realizations of matrices are denoted by calligraphic uppercase letters $(\mathcal{A}, \mathcal{B}, \ldots)$, while algebraic realizations of vectors are written in upright uppercase letters $(\mathrm{U}, \mathrm{V}, \ldots)$. Accordingly, the algebraic form of elliptic boundary value problems can be expressed as:

$$\mathcal{K}\,\mathrm{U} = \mathrm{F}.$$

$\mathbb{P}^1$-FEM basics. The $\mathbb{P}^1$-FEM is based on approximating the unknown solution by a continuous function that is *piecewise linear* over a triangulation of the domain. The domain $\Omega \subset \mathbb{R}^2$ is first discretized into a triangulation $\mathcal{T}_h$, where each $T \in \mathcal{T}_h$ denotes a triangular element. On each triangle T, the local approximation space is the set of affine functions, spanned by three *local shape functions* $\{\varphi_i\}_{i=1}^3$. Each φ_i is associated with a vertex of the triangle and satisfies the nodal interpolation property $\varphi_i(P_j) = \delta_{ij}$, $j = 1, 2, 3$, where $P_j = (x_j, y_j)$ denotes the j-th vertex of the element T. Any discrete solution u_h restricted to T is then expressed as

$$u_h|_T(x, y) = \sum_{i=1}^{3} \varphi_i(x, y)\, u_i, \quad \text{where } u_i \text{ are the nodal values at the vertices of } T.$$

By assembling the local linear interpolants from each triangular element and identifying the degrees of freedom (DOFs) associated with the same mesh node, continuity is automatically enforced at the node level. Specifically, each vertex shared by multiple elements corresponds to a single global unknown, so that the value of the discrete solution at that vertex is unique, regardless of which element's local shape functions are considered.

Remark 3. *This continuity requirement across element interfaces is specific to the standard (conforming) FEM; it is not imposed when using discontinuous Galerkin (dG) methods, where basis functions are allowed to be discontinuous between elements.*

As a result, the collection of all local $\mathbb{P}^1$ approximations forms a global finite-dimensional space of continuous, piecewise linear functions over the entire mesh. First, we define the scalar finite element space

$$V_{h0} = \{\, v_h \in C^0(\overline{\Omega}) \mid v_h|_T \in \mathbb{P}_1(T),\ \forall T \in \mathcal{T}_h,\ v_h|_{\partial\Omega} = 0 \,\}.$$

The corresponding vector-valued space for the displacement field is then

$$\mathbf{V}_{h0} = [V_{h0}]^d = \{\, \mathbf{v}_h = (v_{h,1}, \ldots, v_{h,d}) \mid v_{h,i} \in V_{h0},\ i = 1, \ldots, d \,\}.$$

The partial differential equation is finally projected onto $\mathbf{V}_{h0}$ via the weak formulation, yielding the discrete linear elasticity problem: Find $\mathbf{u}_h \in \mathbf{V}_h$ such that

$$a(\mathbf{u}_h, \mathbf{v}_h) = \ell(\mathbf{v}_h) \quad \forall \mathbf{v}_h \in \mathbf{V}_h,$$

where $a(\mathbf{u},\mathbf{v}) = (\boldsymbol{\varepsilon}(\mathbf{u}) : \boldsymbol{\varepsilon}(\mathbf{v}))_{\mathbb{L}^2(\mathbb{C};\Omega)}$ denotes the bilinear form corresponding to the elastic energy of the system, and $\ell(\mathbf{v}) = (\mathbf{f},\mathbf{v})_{\boldsymbol{L}^2(\Omega)} + (\mathbf{t}_N,\mathbf{v})_{\boldsymbol{L}^2(\Gamma_N)}$ represents the linear form associated with the external body forces and Neumann boundary conditions.

In this part, we introduce how to construct numerically the different operators involved in the linear elasticity problem, including the assembly of the stiffness matrix and the treatment of boundary conditions.

3.1 Mesh generation

An easy way to generate a structured triangular mesh on the square domain $[0,1]\times[0,1]$ is to define a function `generateMesh` that takes the number of subdivisions N along each axis as input. The first step is to create a regular grid of points using `meshgrid`, storing the coordinates in a matrix `coords`. Next, each square formed by four neighboring points is divided into two triangles, with the vertex indices stored in a matrix `elems`. Finally, the boundary nodes are identified for imposing Dirichlet conditions by selecting points whose x or y coordinates are equal to 0 or 1. The function then returns the point coordinates, the triangle connectivity, and the indices of the Dirichlet nodes.

Question 3. Following these steps, write a MATLAB function `generateMesh` that implements this procedure. Note that the mesh can be generated intially using an automatic tool, but the procedure need to be computed for subsection 3.6.

3.2 Element Stiffness Matrix

In 2D linear elasticity, using Voigt notation, the bilinear form (representing the internal strain energy) can be expressed as

$$a(\mathbf{u},\mathbf{v}) := (\varepsilon(\mathbf{u}),\varepsilon(\mathbf{v}))_{\mathbb{L}^2(\mathbb{C};\Omega)} = \int_\Omega \boldsymbol{\varepsilon}(\mathbf{u})^\top \mathbb{C}\,\boldsymbol{\varepsilon}(\mathbf{v})\,d\Omega, \quad \text{with} \quad \boldsymbol{\varepsilon} = \begin{pmatrix} \varepsilon_{xx} \\ \varepsilon_{yy} \\ 2\varepsilon_{xy} \end{pmatrix} = \begin{pmatrix} \partial_x u_x \\ \partial_y u_y \\ \partial_x u_y + \partial_y u_x \end{pmatrix}.$$

For a linear triangular element with three nodes, the nodal displacement vector is

$$\mathrm{u}^e = [u_{1x}, u_{1y}, u_{2x}, u_{2y}, u_{3x}, u_{3y}]^\top \in \mathbb{R}^6.$$

The displacement field inside the element is approximated using linear shape functions $\varphi_i(x,y)$ associated with each node i, satisfying the interpolation property $\varphi_i(x_j,y_j) = \delta_{ij},\ \forall i,j \in \{1,2,3\}$. This gives

$$u_x(x,y) = \sum_{i=1}^{3} \varphi_i(x,y)\,u_{ix}, \quad u_y(x,y) = \sum_{i=1}^{3} \varphi_i(x,y)\,u_{iy}, \quad \forall (x,y) \in T.$$

Moreover, as each φ_i is linear, it can be written in the affine form $\varphi_i(x,y) = a_i + b_i x + c_i y,\ \forall i \in \{1,2,3\}$, where the coefficients a_i, b_i, c_i are determined by solving

$$\begin{pmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{pmatrix} \begin{pmatrix} a_1 & a_2 & a_3 \\ b_1 & b_2 & b_3 \\ c_1 & c_2 & c_3 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix},$$

and the derivatives of the shape functions are constant inside the element:

$$b_i = \frac{\partial \varphi_i}{\partial x}, \quad c_i = \frac{\partial \varphi_i}{\partial y}, \quad i = 1,2,3.$$

Using Voigt notation, the strain components can be expressed in terms of the nodal displacements:

$$\begin{aligned} \varepsilon_{xx} &= b_1 u_{1x} + b_2 u_{2x} + b_3 u_{3x}, \\ \varepsilon_{yy} &= c_1 u_{1y} + c_2 u_{2y} + c_3 u_{3y}, \\ 2\varepsilon_{xy} &= c_1 u_{1x} + b_1 u_{1y} + c_2 u_{2x} + b_2 u_{2y} + c_3 u_{3x} + b_3 u_{3y}. \end{aligned}$$

These relations can be arranged in matrix form, defining the strain–displacement matrix R:

$$\mathcal{R} = \begin{pmatrix} b_1 & 0 & b_2 & 0 & b_3 & 0 \\ 0 & c_1 & 0 & c_2 & 0 & c_3 \\ c_1 & b_1 & c_2 & b_2 & c_3 & b_3 \end{pmatrix}, \quad \boldsymbol{\varepsilon} = R\,\mathbf{u}^e.$$

Finally, the element stiffness matrix is obtained by integrating over the triangular element T:

$$\mathcal{K}^e = \int_T \mathcal{R}^\top \mathbb{C} \mathcal{R}\,dT.$$

In our case, since $\mathcal{R}$ is constant within the element, this integral can be evaluated exactly using a midpoint quadrature at the element centroid.

Question 4. Following these steps, write a MATLAB function `local_stiff_mat` that implements this procedure.

Remark 4 (High-order FEM)**.** *For higher-order finite elements, the shape functions φ_i are polynomials of degree greater than one. Their derivatives, which form the strain–displacement matrix $\mathcal{R}$, thus vary spatially within the element. Unlike linear elements, multi-point quadrature is required to accurately compute*

$$\mathcal{K}^e = \int_T \mathcal{R}(x,y)^\top \, \mathbb{C} \, \mathcal{R}(x,y) \, dT \approx \sum_{q=1}^{N_q} w_q \, \mathcal{R}(x_q,y_q)^\top \, \mathbb{C} \, \mathcal{R}(x_q,y_q),$$

where N_q is the number of quadrature points, (x_q, y_q) are the coordinates of the quadrature points within the element T, and w_q are the associated quadrature weights.

In practice, at each quadrature point, one evaluates the derivatives of the shape functions, constructs the corresponding $\mathcal{R}$ matrix, and forms the local contribution $\mathcal{R}^\top \mathbb{C} \mathcal{R}$ weighted by w_q. The number of quadrature points should be chosen according to the degree of the basis functions to capture the stiffness contributions accurately.

3.3 Global Assembly of the System

Once the local element operators have been computed, the next step in the finite element method is to assemble these contributions into the global stiffness matrix $\mathcal{K}$ and the global load vector F.

In two-dimensional elasticity problems, it is convenient to group the degrees of freedom (DOFs) by node so that both displacement components at a given node are stored consecutively. The global vector of unknowns is then ordered as

$$[u_{1x},\, u_{1y},\, u_{2x},\, u_{2y},\, \ldots,\, u_{nx},\, u_{ny}],$$

where n is the total number of nodes, and each pair (u_{ix}, u_{iy}) represents the displacement components of node i. For a triangular $\mathbb{P}^1$-FEM element, each element has three nodes and therefore six local DOFs. The corresponding global indices are collected in the vector

$$\mathrm{I}^e = [\, 2n_1 - 1,\, 2n_1,\, 2n_2 - 1,\, 2n_2,\, 2n_3 - 1,\, 2n_3 \,]^T,$$

where n_1, n_2, n_3 denote the global node indices of the element. The local stiffness matrix $\mathcal{K}^e$ and load vector F^e are then assembled into the global system by adding their contributions at the corresponding global indices:

$$\mathcal{K}(\mathrm{I}^e, \mathrm{I}^e) \mathrel{+}= \mathcal{K}^e, \qquad \mathrm{F}(\mathrm{I}^e) \mathrel{+}= \mathrm{F}^e.$$

Here, $\mathcal{K}(\mathrm{I}^e, \mathrm{I}^e)$ denotes the submatrix of the global stiffness matrix corresponding to the DOFs of element e, and the local load vector f^e is similarly added to the entries of F indexed by I^e.

This assembly procedure naturally handles nodes shared between elements. Nodes located at element interfaces appear in multiple I^e vectors, so the contributions from all neighboring elements are summed at the corresponding global DOFs. This ensures the continuity of the displacement field across elements, as the global matrix and vector entries associated with a shared node accumulate contributions from each element containing that node.

Once all elements have been processed, the global system takes the form

$$\mathcal{K}\,\mathrm{U} = \mathrm{F},$$

where U is the global vector of unknown nodal displacements. Finally, the boundary conditions must be applied before solving the system.

Question 5. Following these steps, write MATLAB functions `assemble_stiff_mat` that implements this procedure.

3.4 Load Vector (Source Term)

The external forces or body forces applied to the material are represented by a source term $\mathbf{f}$ in the weak form. The corresponding element load vector is defined as

$$\mathrm{F}^e = \int_T \mathcal{N}^\top f \; dT,$$

where $\mathcal{N}$ is the matrix of shape functions that interpolates the displacement field: $\mathrm{u}^e(x,y) = \mathcal{N}(x,y)\,\mathrm{u}^e$. For a linear triangular element with three nodes, $\mathcal{N}$ can be written as

$$\mathcal{N} = \begin{pmatrix} \varphi_1 & 0 & \varphi_2 & 0 & \varphi_3 & 0 \\ 0 & \varphi_1 & 0 & \varphi_2 & 0 & \varphi_3 \end{pmatrix},$$

so that the nodal forces $\mathrm{f}^e = [f_{1x}, f_{1y}, f_{2x}, f_{2y}, f_{3x}, f_{3y}]^\top$ correctly account for the body forces distributed within the element.

Question 6. Write MATLAB functions `f_midpoint` and `assemble_source_term` that computes the elemental load forces ising a midpoint quadrature at the element centroid and assemble the contribution in the global vector.

3.5 Boundary Conditions

Boundary conditions play a crucial role in elasticity problems. Without proper constraints on the displacement (Dirichlet conditions), the structure may undergo rigid body motions, leading to an ill-posed problem with an undetermined solution. Properly prescribing displacements or supports ensures that the model is physically meaningful and that internal strains and stresses are correctly determined. Neumann conditions, on the other hand, allow the application of external forces or tractions, enabling the simulation of realistic loading scenarios.

Dirichlet Boundary Conditions. Dirichlet boundary conditions prescribe the displacement of the material along a portion of the boundary $\Gamma_D \subset \partial\Omega$. In the finite element implementation, these conditions are enforced by modifying the global system of equations. Specifically, the rows and columns of the global stiffness matrix corresponding to prescribed degrees of freedom are adjusted, and the right-hand side is modified to impose the known displacements. Various strategies exist, such as direct substitution or the use of large penalty coefficients, but the direct modification of the stiffness matrix is the most common in practice.

Neumann Boundary Conditions. Neumann boundary conditions prescribe surface tractions $\mathbf{t}_N$ along a portion of the boundary $\Gamma_N \subset \partial\Omega$. Similarly to the source term, the corresponding nodal contributions are computed at the element level as

$$\mathrm{F}^e_{\text{boundary}} = \int_{\Gamma_N} \mathcal{N}^\top \mathbf{t}_N \, d\Gamma,$$

then, these contributions are added to the source term.

Question 7. Implement two MATLAB functions for boundary conditions: `applyNeumann` to assemble Neumann contributions into the global load vector, and `applyDirichlet` to impose Dirichlet conditions by modifying the stiffness matrix and load vector.

3.6 Test case with analytical solution

To verify the implementation of the 2D linear elasticity FEM operators, we consider a simple test case on the unit square domain $\Omega = (0,1)\times(0,1)$ with homogeneous Dirichlet boundary conditions on all edges. The exact displacement field is chosen as

$$\mathbf{u}_{\text{exact}}(x,y) := \begin{pmatrix} \sin(\pi x)\sin(\pi y) \\ 0 \end{pmatrix},$$

so that the corresponding body force $\mathbf{f}$ can be computed analytically from the linear elasticity problem:

$$-\nabla\cdot\sigma(\mathbf{u}_{\text{exact}}) = \mathbf{f} \quad \text{in } \Omega, \qquad \mathbf{u}_{\text{exact}} = 0 \quad \text{on } \partial\Omega.$$

After solving the linear system for the nodal displacements $\mathbf{U}$, the accuracy of the FEM solution can be assessed by computing the standard L^2 and H^1 errors:

$$\|\mathbf{u}_h - \mathbf{u}_{\text{exact}}\|^2_{\boldsymbol{L}^2(\Omega)} := \sum_{T\in\mathcal{T}_h} \|\mathbf{u}_h - \mathbf{u}_{\text{exact}}\|^2_{\boldsymbol{L}^2(T)}, \tag{6a}$$

$$|\mathbf{u}_h - \mathbf{u}_{\text{exact}}|^2_{\boldsymbol{H}^1(\Omega)} := \sum_{T\in\mathcal{T}_h} \|\nabla(\mathbf{u}_h - \mathbf{u}_{\text{exact}})\|^2_{\boldsymbol{L}^2(T)}. \tag{6b}$$

In practice, for $\mathbb{P}^1$ elements, these integrals are evaluated numerically on each element using a midpoint quadrature at the element barycenter. Note that the local H^1 semi-norm of an element is computed by evaluating the Frobenius norm of the gradient of the error, $\nabla(\mathbf{u}_h - \mathbf{u}_{\text{exact}})$.

Question 8. First, compute the analytical expression of the load vector corresponding to the chosen exact solution. Then, perform a convergence study over a sequence of refined meshes and report both the error curves and the observed convergence rates with reference slopes for both $\boldsymbol{H}^1$ and $\boldsymbol{L}^2$ norms.

4 Application Case

Linear elasticity modeling is an essential tool in mechanical and civil engineering for predicting the structural response of solid components under various loading conditions. Typical applications include the deformation of mechanical parts under stress, the design of bridges and buildings, and the study of stress concentrations near holes or notches in structural elements. In all these situations, finite element simulations allow engineers to assess potential failure zones, optimize material usage, and ensure the structural integrity of complex geometries.

4.1 Test case setting

In this section, we consider a two-dimensional elasticity problem on a circular plate with three internal holes, arranged in such a way that the overall geometry resembles a *smiley face*, as illustrated in Figure 2. We consider the following boundary conditions:

i) **on the external boundary**, we impose homogeneous Dirichlet boundary conditions (Γ_D in red in Figure 2),

$$\mathbf{u} = \mathbf{0} \quad \text{on } \Gamma_D.$$

ii) **on the inner boundaries (holes)**, we impose inhomogeneous Neumann boundary conditions (Γ_N in blue in Figure 2),

$$\boldsymbol{\sigma}(\mathbf{u}) \cdot \mathbf{n} = \mathbf{t}_N \quad \text{on } \Gamma_N,$$

where $\mathbf{t}_N = (0, t_y)$ represents a prescribed traction acting upward or downward depending on the hole, with t_y being a constant value. This mimics the effect of localized pressure acting on the internal surfaces of the material.

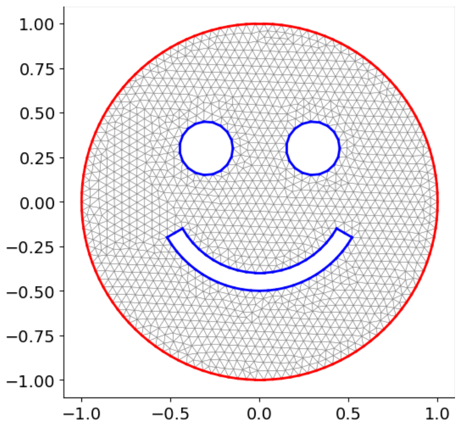

Figure 2: Mesh of the smiley-face geometry with Dirichlet boundary conditions on the outer circle (red) and Neumann boundary conditions on the internal holes (blue).

4.2 Post-processing and visualization

Once the displacement field $\mathbf{u}$ has been obtained, it is essential to compute and visualize derived physical quantities, such as the strain and stress tensors, to interpret the mechanical behavior of the structure.

Computation of strain and stress fields. From the nodal displacements, one can compute for each element, the strain tensor $\boldsymbol{\varepsilon}$ and the corresponding stress tensor $\boldsymbol{\sigma}$, obtained through Hooke's law. These quantities are first evaluated at the element level. To obtain smoother fields, we perform an averaging at the node level: for each node, the contributions of all neighboring elements are summed, typically weighted by the area of each element, and then divided by the total area associated with the node. For each node i, shared by elements $T \in \omega_i$,

$$\boldsymbol{\varepsilon}_i := \frac{\sum_{T\in\omega i} |T|\, \boldsymbol{\varepsilon}_T}{\sum_{T\in\omega_i} |T|}, \quad \boldsymbol{\sigma}_i := \frac{\sum_{T\in\omega_i} |T|\, \boldsymbol{\sigma}_T}{\sum_{T\in\omega_i} |T|}.$$

where ω_i denotes the patch of node i, (*i.e.* the set of all elements connected to it). This procedure transforms the piecewise-constant, discontinuous element stresses into a continuous field over the mesh, providing smoother and more interpretable results for visualization and post-processing.

Shear energy density. The elastic shear energy density quantifies the amount of distortion or shape change in the material caused by shear forces. It can be defined by,

$$E := \left(\frac{\mu}{24(\mu+\lambda)^2} + \frac{1}{8\mu} \right) (\sigma_{xx} + \sigma_{yy})^2 + \frac{1}{2\mu} (\sigma_{xy}^2 - \sigma_{xx}\sigma_{yy}). \tag{7}$$

High values of E indicate regions where the material is strongly deformed in shape, which often corresponds to stress concentrations and potential failure zones. Visualizing this quantity helps identify areas of high shear and interpret the mechanical response of the structure.

4.3 Objectives

Question 9. Solve the linear system resulting from the $\mathbb{P}^1$-FEM discretization of this test case and visualize the elastic shear energy density.

Question 10. Provide a critical analysis of the test case setting and the results obtained. Discuss potential improvements or alternative approaches that could enhance the accuracy or efficiency of the simulation.