A finite element method for scalar and vectorial interface problems

Comparison of a symmetric and non symmetric implementation with GetFEM++

Alessandra Arrigoni
Sara Francesca Pichierri

21 febbraio 2020

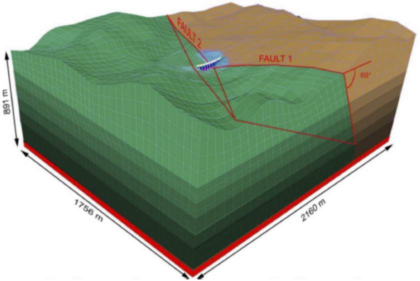
Context and goals

- **Interface problem**:
  The domain contains several areas with different physical properties. The exact solution satisfies interface conditions, as well as the usual boundary conditions.

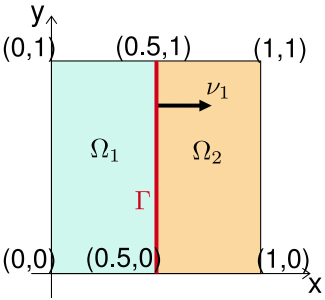
- **Goal**:
  Prove optimal convergence order for a Petrov-Galerkin FEM in two different forms, applied to an elliptic problem and a 2D linear elasticity problem.

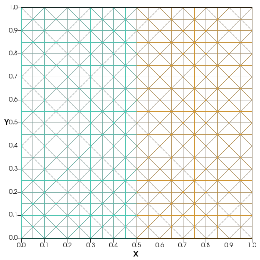
Applications

- Slip-conditions analysis between two subduction zones.
- Determination of underground cracks through surface measurements and reconstruction of the fault geometry (*inverse problem*).
- Study of flow in fractured porous media, applied to water resource management or to the isolation of radioactive waste to prevent water contamination.



The domain and its discretization





- $\Omega \subset \mathbf{R}^2$: bounded Lipschitz (unit square)
- Γ: Lipschitz curve representing the vertical interface at $x = 0.5$
- two disjoint subdomains: Ω_1 and Ω_2 such that $\Omega \setminus \Gamma = \Omega_1 \cup \Omega_2$ and $\Omega_1 \cap \Omega_2 = 0$
- $\Gamma_1 = \partial\Omega_1 \setminus \Gamma$ and $\Gamma_2 = \partial\Omega_2 \setminus \Gamma$

The triangulation is:

- conforming (no hanging nodes)
- quasiuniform, i.e. $\max_{\tau_k \in \mathcal{T}_h} h_k \lesssim \min_{\tau_k \in \mathcal{T}_h} h_k$
- aligned with Γ and conforming

Scalar elliptic problem: strong formulation

The first interface problem is the classic scalar elliptic problem, whose strong form is the following:

$$
\begin{gathered}
A_1 u_1 = f \text{ in } \Omega_1, \quad A_2 u_2 = f \quad \text{in } \Omega_2 \\
u_2 - u_1 = q_0 \quad \text{on } \Gamma \\
\frac{\partial u_2}{\partial \nu_{A_2}} - \frac{\partial u_1}{\partial \nu_{A_1}} = q_1 \quad \text{on } \Gamma \\
u_1 = g_1 \quad \text{on } \Gamma_1, \quad u_2 = g_2 \quad \text{on } \Gamma_2
\end{gathered}
$$

where A_1 and A_2 are the second order uniformly elliptic operators with smooth coefficients $a_{ij}^k \in C^2(\Omega_k')$.

Scalar elliptic problem: weak formulation

Define the Sobolev space $H^1(\Omega_1 \cup \Omega_2)$ of all functions w such that: $w|_{\Omega_1} \in H^1(\Omega_1)$ and $w|_{\Omega_2} \in H^1(\Omega_2)$.

Given $g \in H^{r-1/2}(\Gamma_1 \cup \Gamma_2)$, $q_0 \in H^{r-1/2}(\Omega)$, $q_1 \in H^{r-3/2}(\Omega)$, and $f \in L^2(\Omega)$ with $1 \leq r \leq 2$, find $u \in H^1(\Omega_1 \cup \Omega_2)$ such that:

$$a^0(u,v) + a^1(u,v) = (f,v) - < q_1, v >_\Gamma \quad \forall\, v \in H_0^1(\Omega)$$

$$[u] = q_0 \quad \text{on} \quad \Gamma, \quad u = g \quad \text{on} \quad \partial\Omega$$

where $[u] = u_2 - u_1$ and $u_i = u|_{\Omega_i}$.

- different spaces for the solution u and the test functions v
- the solution u can be discontinuous on the interface.

Linear elasticity problem: strong formulation

The second interface problem is the linear elasticity system. Given μ and λ the Lamè parameters, the elastic displacement $\mathbf{u}$ satisfies:

$$-\nabla \cdot \underline{\sigma}_1(\mathbf{u}_1) = \mathbf{f} \quad \text{in } \Omega_1, \quad -\nabla \cdot \underline{\sigma}_2(\mathbf{u}_2) = \mathbf{f} \quad \text{in } \Omega_2$$
$$[\![\mathbf{u}]\!] = \mathbf{q}_0 \quad \text{on } \Gamma$$
$$[\![\underline{\sigma}(\mathbf{u}) \cdot \boldsymbol{\nu}_1]\!] = \mathbf{q}_1 \quad \text{on } \Gamma$$
$$\mathbf{u}_1 = \mathbf{g}_1 \quad \text{on } \Gamma_1, \quad \mathbf{u}_2 = \mathbf{g}_2 \quad \text{on } \Gamma_2$$

where $[\![\mathbf{w}]\!] = \mathbf{w}_2 - \mathbf{w}_1$ and $\underline{\sigma}_k(\mathbf{u}_k)$ is the stress tensor on Ω_k:

$$\underline{\sigma}_k(\mathbf{u}_k) = 2\mu_k \, \underline{\epsilon}(\mathbf{u}_k) + \lambda_k \nabla \cdot \mathbf{u}_k \, I$$

$$\text{with} \quad \underline{\epsilon}(\mathbf{u}_k) = \frac{1}{2}(\nabla \mathbf{u}_k + \nabla \mathbf{u}_k^T) \quad \text{and} \quad I \in \mathbf{R}^{2\times 2}$$

Linear elasticity problem: weak formulation

For $\mathbf{g} \in \mathbf{H}^{r-1/2}(\Gamma_1 \cup \Gamma_2)$, $\mathbf{q}_0 \in \mathbf{H}^{r-1/2}(\Gamma)$, $\mathbf{q}_1 \in \mathbf{H}^{r-3/2}(\Gamma)$ and $\mathbf{f} \in \mathbf{L}^2(\Omega)$ with $1 \leq r \leq 2$, find $\mathbf{u} \in \mathbf{H}^1(\Omega_1 \cup \Omega_2)$ such that:

$$b(\mathbf{u}, \mathbf{v}) = (\mathbf{f}, \mathbf{v})_\Omega - \langle \mathbf{q}_1, \mathbf{v} \rangle_\Gamma, \quad \forall\, \mathbf{v} \in \mathbf{H}_0^1(\Omega)$$
$$[\![\mathbf{u}]\!] = \mathbf{q}_0 \quad \text{on} \quad \Gamma, \quad \mathbf{u} = \mathbf{g} \quad \text{on} \quad \partial\Omega$$

where the continuous bilinear form $b(\cdot, \cdot)$ is defined as:
$b(\mathbf{u}, \mathbf{v}) = b^1(\mathbf{u}, \mathbf{v}) + b^2(\mathbf{u}, \mathbf{v})$ with for $k = 1, 2$ and $\mathbf{u}, \mathbf{v} \in \mathbf{H}^1(\Omega_k)$:

$$b^k(\mathbf{u}, \mathbf{v}) = \int_{\Omega_k} \frac{\mu_k}{2}(\nabla \mathbf{u} + \nabla \mathbf{u}^T) : (\nabla \mathbf{v} + \nabla \mathbf{v}^T) + \int_{\Omega_k} \lambda_k (\nabla \cdot \mathbf{u}) \cdot (\nabla \cdot \mathbf{v})$$

Comparison of Finite Element Methods

Petrov-Galerkin Method

- ideally suitable for cases where the solution is regular everywhere except at the interface;
- smaller system (trial functions can be chosen to be continuous everywhere except at the interface).

Discontinuous-Galerkin Method

- natural choice for discontinuous solutions;
- large number of degrees of freedom $\Longrightarrow$ increase of the computational cost for solving the associated linear system and of the memory usage.

Finite element discrete spaces

- $V_h(\Omega_k),\ k = 1, 2$ is the space of continuous piecewise linear functions on Ω_k
- V_h is the set of functions $\phi \in H^1(\Omega_1 \cup \Omega_2)$ such that $\phi|_{\Omega_k} \in V_h(\Omega_k)$.
- $V_h(\Gamma)$ is the set of piecewise linear functions on the interface
- $V_h(\Gamma_1 \cup \Gamma_2)$ is the set of functions ψ such that $\psi|_{\Gamma_k}$ is piecewise linear; ψ may be discontinuous at the intersection points.

We use the notation $\mathbf{V}_h, \mathbf{V}_h(\Omega_k), \mathbf{V}_h(\Gamma)$ and $\mathbf{V}_h(\Gamma_1 \cup \Gamma_2)$ to denote their vectorial counterparts.

Scalar elliptic problem: approximate formulation

The **approximate problem** reads as follows: in Ω find $u_h \in V_h$ such that

$$a(u_h, \phi) = (f, \phi)_\Omega - \langle q_1, \phi \rangle_\Gamma \qquad \forall\, \phi \in V_h^0$$
$$\text{with } [u_h] = q_{0,h} \text{ on } \Gamma \quad \text{and} \quad u_h = g_h \text{ on } \partial\Omega$$

where $V_h^0 = V_h \cap H_0^1(\Omega)$, $[u_h] = u_{h2} - u_{h1}$.

- g_h and $q_{0,h}$ are the projections on the finite dimensional spaces $V_h(\Gamma_1 \cup \Gamma_2)$ and $V_h(\Gamma)$ of the continuous data
- the jump condition and the Dirichlet boundary condition are assigned in a direct way
- the spaces for the test (ϕ) and the trial functions (u_h) are different $\Longrightarrow$ *Petrov-Galerkin* conforming FEM

Linear elasticity problem: approximate formulation

The **approximate problem** reads as follows: in Ω find $\mathbf{u}_h \in \mathbf{V}_h$ such that

$$b(\mathbf{u}_h, \boldsymbol{\phi}) = (\mathbf{f}, \boldsymbol{\phi})_\Omega - \langle \mathbf{q}_1, \boldsymbol{\phi} \rangle_\Gamma \qquad \forall\, \boldsymbol{\phi} \in \mathbf{V}_h^0$$

$$\text{with } [\![\mathbf{u}_h]\!] = \mathbf{q}_{0,h} \text{ on } \Gamma \quad \text{and} \quad \mathbf{u}_h = \mathbf{g}_h \text{ on } \partial\Omega$$

where $\mathbf{V}_h^0 = \mathbf{V}_h \cap \mathbf{H}_0^1(\Omega)$.

Interface conditions are usually expressed in terms of **normal** and **tangential** components with respect to the curve Γ:

$$[u_t] = q_0, \quad [u_n] = 0, \quad [(\underline{\sigma}(\mathbf{u}) \cdot \boldsymbol{\nu}_1)_t] = 0, \quad [(\underline{\sigma}(\mathbf{u}) \cdot \boldsymbol{\nu}_1)_n] = 0 \quad \text{on } \Gamma,$$

which impose a continuous stress and a discontinuous tangential displacement.

Basic method

The space V_h (trial functions) is larger than V_h^0 (test functions) $\Longrightarrow$ more columns than rows in the system.
Additional equations are retrieved by adding identity block matrices to impose the jump of the solution on Γ.

$$\begin{bmatrix} A_{1,1} & \mathbf{0} & A_{1,\Gamma} & \mathbf{0} \\ \mathbf{0} & A_{2,2} & \mathbf{0} & A_{2,\Gamma} \\ \mathbf{0} & \mathbf{0} & -I & I \\ A_{\Gamma,1} & A_{\Gamma,2} & \mathbf{0} & A_{\Gamma 2,\Gamma 2} \end{bmatrix} \begin{bmatrix} \mathbf{u}1 \\ \mathbf{u}2 \\ \mathbf{u}1_\Gamma \\ \mathbf{u}2_\Gamma \end{bmatrix} = \begin{bmatrix} \mathbf{f}1 \\ \mathbf{f}2 \\ \mathbf{q0} \\ \mathbf{f}1_\Gamma + \mathbf{f}2_\Gamma - M\mathbf{q1} \end{bmatrix}$$

- unknowns on the interface: solutions $\mathbf{u}1$ and $\mathbf{u}2$
- non symmetric, even with symmetric simple blocks
- easily generalized to cases with non uniform coefficients

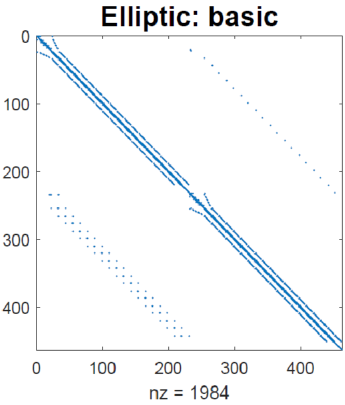
Symmetric method

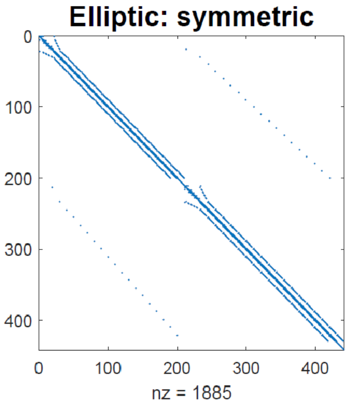
Apply a linear change of variable to reduce the system to the dimension of V_h^0 and obtain a symmetric matrix.
It requires an additional step to reconstruct the solutions from their jump and average (inverse transformation).

$$\begin{bmatrix} A_{1,1} & \mathbf{0} & A_{1,\Gamma} \\ \mathbf{0} & A_{2,2} & A_{2,\Gamma} \\ A_{\Gamma,1} & A_{\Gamma,2} & A_{\Gamma,\Gamma} \end{bmatrix} \begin{bmatrix} \mathbf{u}1 \\ \mathbf{u}2 \\ \{\mathbf{u}^h\} \end{bmatrix} = \begin{bmatrix} \mathbf{f}1 + \frac{1}{2} A_{1,\Gamma}\,\mathbf{q0} \\ \mathbf{f}2 - \frac{1}{2} A_{2,\Gamma}\,\mathbf{q0} \\ \mathbf{f}1_\Gamma + \mathbf{f}2_\Gamma - M\mathbf{q1} \end{bmatrix}$$

- unknowns on the interface: average $\{\mathbf{u}^h\} = \frac{1}{2}(\mathbf{u}1_\Gamma + \mathbf{u}2_\Gamma)$
- symmetric, only with uniform coefficients on Ω
- hard to generalize to cases with non uniform coefficients

Sparsity patterns





The matrices are not entirely symmetric because of how the Dirichlet boundary conditions are imposed.

GetFEM++ library

Open source FEM library providing a flexible framework for linear and non linear systems of PDEs. Its strengths:

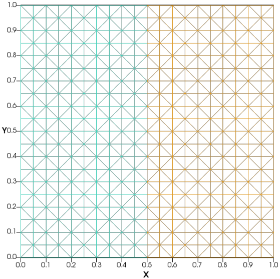
- "high level assembly": optimized compilation of the operators starting from expressions similar to weak formulations; symbolic differentiation for non linear terms;
- wide choice of FE and integration methods;
- representation of level sets for cut finite elements useful in interface problems;
- Matlab and Python interfaces and compatibility with common mesh formats.

We needed to have access to the matrices and the vectors $\Longrightarrow$ "low level assembly" of basic linear terms.

The `Bulk` class

Assumption: no explicit interface within a domain, but two distinct subdomains

- one of the boundary sides marked as interface
- same subdivision and number of elements to obtain a conforming mesh.



Collaborates with the `Problem` class via **aggregation.**

Classes for data

Used to store and evaluate data on a given mesh point: one of the fields is the `LifeV::Parser`.
Collaborate with `Problem` via **composition**.

`Bulkdatum:` scalar (coefficients) and vectorial data (source, exact solution) linked to the interior of Ω_1 and Ω_2.

```
scalar_type getValue(const base_node & x,
                     const size_type what);
```

`BC:` scalar and vectorial data linked to the boundaries.

```
scalar_type BCNeum(const base_node& x,
    const size_type what, const size_type& flag);
```

Assumption: interface conditions as boundary conditions $\Longrightarrow$ no tailored description for the interface.

Classes for DOFs

`FEM:` description of scalar and vectorial discrete spaces. Simple wrapper of the library's class `getfem::mesh_fem` highlighting the most useful methods.
Collaborates with `Problem` via **composition**.

`LinearSystem:` SuperLU solver and methods to modify, copy and extract parts of the sparse matrices and vectors provided by the auxiliary library `Gmm++`

```
using sparseVector_Type =
            gmm::rsvector<scalar_type>;
using sparseMatrix_Type =
            gmm::row_matrix<sparseVector_Type>;
```

Collaborates with `Problem` via **aggregation**.

Operators

Several free functions to represent the bilinear forms and linear operators linked to any PDEs problem.
Pointers as parameters to comply with other methods.

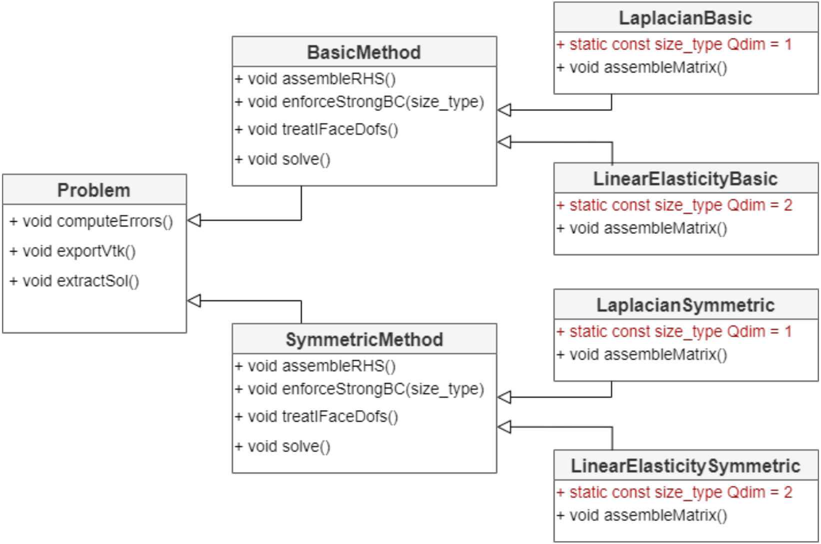
- simple evaluation on all the degrees of freedom

```
void exactSolution (scalarVectorPtr_Type V,
    const FEM& FemSol, BulkDatum& Solution);
```

- use of GetFEM++ low level assembly functions

```
void stiffness (sparseMatrixPtr_Type M,
    const FEM& FemSol, const FEM& FemCoef,
    BulkDatum& Diff, const getfem::mesh_im& im);
```

`Problem` inheritance tree



Basic vs Symmetric method - Pt. 1

All methods deal with both the scalar and the vectorial problem.

`enforceStrongBC:` modifies the rows of the matrix and the right hand side $\Longrightarrow$ symmetry is lost in both methods.
The symmetric case needs special treatment for the dofs on both the interface and the boundary of Ω_2:

```
// Add 1/2 of the jump
M_Sys.setRHSValue( ii  + j + M_nbDOF1,
                value + M_q02.at(ii+j)/2 );
```

`treatIFaceDofs:` in the basic case introduces the identity block matrices.
In the symmetric case removes additional rows and columns:

```
M_Sys.eliminateRowsColumns(dof_IFace1);
M_nbTotDOF = M_nbDOF1 + M_nbDOF2 - M_nbDOFIFace;
```

Basic vs Symmetric method - Pt. 2

`assembleMatrix:` builds the two matrices `A1` and `A2` separately and places them in the global system.
Calls either `stiffness` or `linearElasticity` operator according to the class.

`solve:` solves the system calling `LinearSystem::solve`.
The symmetric case needs the reconstruction of the solutions from their average and jump:

$$\mathbf{u}1_\Gamma = \{\mathbf{u}^h\} + \frac{1}{2}[\![\mathbf{u}^h]\!], \qquad \mathbf{u}2_\Gamma = \{\mathbf{u}^h\} - \frac{1}{2}[\![\mathbf{u}^h]\!].$$

Constructor of the `final` classes

```
LaplacianBasic::LaplacianBasic ( ... ):
    BasicMethod(dataFile,"laplacian",bulk1,bulk2,
        LaplacianBasic::Qdim, extSys),
    diff1( ... ),
    diff2( ... ) {}
```

Use of the inheriting constructor introduced by `C++11` $\Longrightarrow$ static variable needed to set FEM space dimension before instantiating the class.

Test 1: elliptic scalar problem with basic form

Exact solution:

$$u = \frac{1}{2}\cos(2\pi x)\sin(2\pi y) \text{ on } \Omega_1, \quad u = \cos(2\pi x)\sin(2\pi y) \text{ on } \Omega_2$$

Parameters: different diffusion coefficients ($\mu_1 = 2$, $\mu_2 = 1$). The source term is continuous on Ω and the manufactured solution is computed consequently.

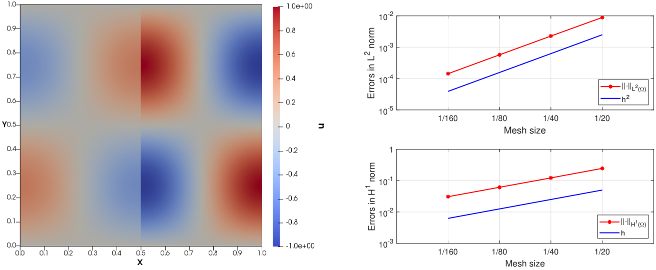


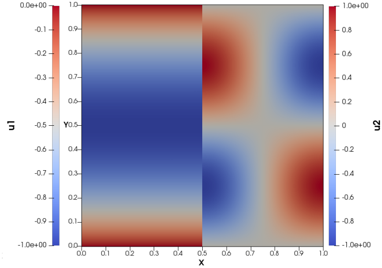
Figure: Numerical solution for a mesh with 160×160 subdivisions in the square domain and errors in the L^2-norm and H^1-norm.

Test 2: elliptic scalar problem with symmetric form

Exact solution:

$$u = 4y(y-1) \text{ on } \Omega_1, \quad u = \cos(2\pi x)\sin(2\pi y) \text{ on } \Omega_2$$

Parameters: parabolic solution with constant and uniform diffusion coefficients and periodic jumps q_0 and q_1.



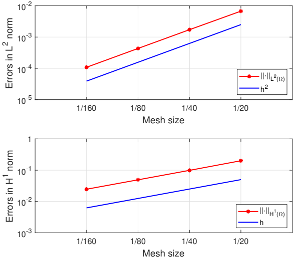


Figure: Numerical solution for a mesh with 160×160 subdivisions in the square domain and errors in the L^2-norm and H^1-norm.

Test 3: linear elasticity problem with basic form

Exact solution:

$$\mathbf{u} = [x^3, y^2 x]^T \text{ on } \Omega_1, \quad \mathbf{u} = [x^3, y^2 x + 3]^T \text{ on } \Omega_2$$

Parameters: the manufactured elastic displacement $\mathbf{u} \in \mathbf{R}^2$ is discontinuous only in the y-direction, whereas the x-component and the Cauchy stress tensor are taken continuous. We consider a homogeneous material, i.e. the Lamè parameters are uniform and constant on the whole Ω.

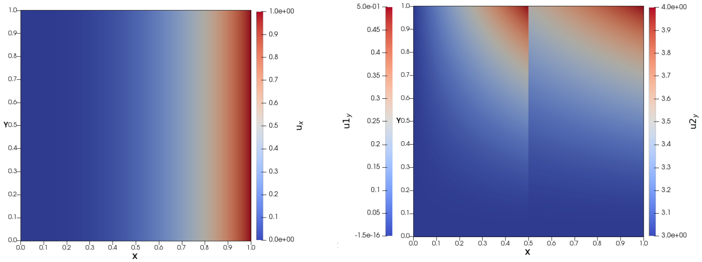


Figure: x- and y-component of the numerical solution for a mesh with 160×160 subdivisions in the square domain.

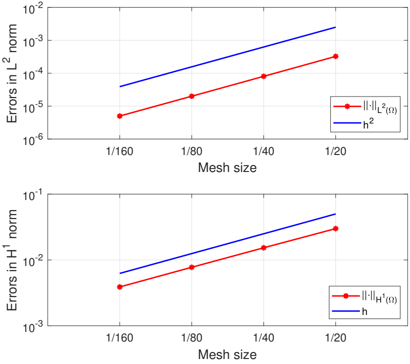


Figure: Errors in the L^2-norm and H^1-norm versus the mesh size h (log-log scale).

Test 4: linear elasticity problem with symmetric form

Exact solution:

$$\mathbf{u} = [\sin(2\pi x), \sin(2\pi y)]^T \text{ on } \Omega_1, \quad \mathbf{u} = [x^2 + \sin(2\pi x), x^2 + \sin(2\pi y)]^T \text{ on } \Omega_2.$$

Parameters: both the solution and the normal stress feature a discontinuity along the interface Γ. We consider a homogeneous material, i.e. the Lamè parameters are uniform and constant on the whole Ω.

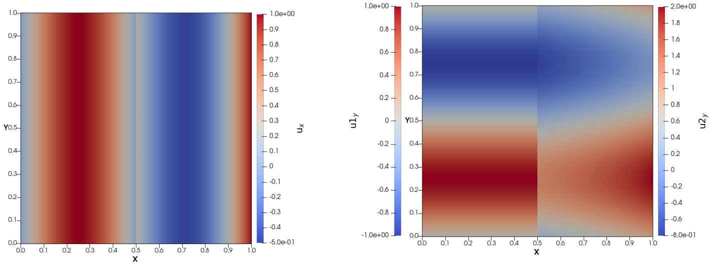


Figure: x- and y-component of the numerical solution for a mesh with 160×160 subdivisions in the square domain.

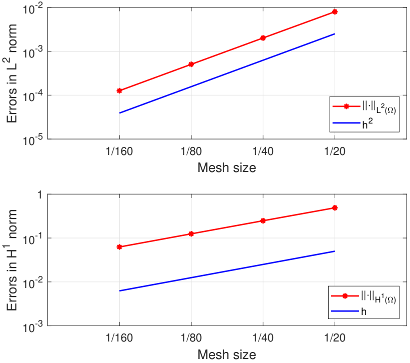
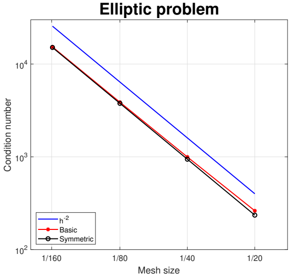


Figure: Errors in the L^2-norm and H^1-norm versus the mesh size h (log-log scale).

Condition number comparison

The **condition number** of the matrices increases as h^{-2}.
For coarser meshes it is *slightly smaller* with the symmetric method $\Longrightarrow$ there is *almost no difference* between the two methods (at least for a 2-dimensional problem).



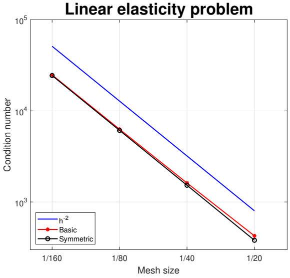


Figure: Trend of the condition numbers with the basic and the symmetric method for the laplacian and the linear elasticity problem.

Conclusions

- We verified the optimal order convergence of a Petrov-Galerkin FEM to solve linear **scalar** and **vectorial** interface problems using a mesh conforming to the crack.
- Two equivalent algebraic formulations have been implemented exploiting the **"low level assembly"** of the `GetFEM++` library $\Longrightarrow$ no remarkable differences in the performances of the two approaches, even if the symmetric system is smaller.

Extensions and future work

- Interesting to test the symmetric formulation on **a 3D problem**, where the reduced system may be much smaller than the original one $\Longrightarrow$ faster solution step.
- The same code, with small modifications, can solve problems with **Neumann boundary conditions** on parts of $\partial\Omega$ or with **higher order finite elements** defined in `GetFEM++`.
- **Other different linear problems** can be easily integrated in the hierarchy structure we designed.
- **Different computational domains** can be used, provided that the meshes are conforming to the single interface.