Bocconi University
Ph.D. Statistics and Computer Science
Secchi Cecilia

Elliptic PDEs FEM solver

Introduction

The purpose of this project is to solve 2-dim stationary PDEs of second order with P_1 Galerking Finite Element Method. The PDEs are of the kind

$$-\text{div}(K\nabla u) + \text{div}(\beta u) + cu = f \quad \text{in } \Omega,$$

with Dirichlet or Neumann boundary conditions on $\Gamma_D \cup \Gamma_N = \partial\Omega$; in particular:

$$u = f_d \quad \text{on} \quad \Gamma_D, \quad \nabla u \cdot \nu = g \quad \text{on} \quad \Gamma_N.$$

K is the diffusion coefficient, $\beta : \Omega \to \mathbb{R}^d$ is a vector field, and $c : \Omega \to \mathbb{R}$ is the reaction coefficient; f and $g : \partial\Omega \to \mathbb{R}^d$ and all these functions are sufficiently smooth.
From the physical point of view, this equation may represent the transport of a solute dissolved in a fluid that moves with the velocity field $\beta(x)$ and undergoes a linear chemical transformation with rate c.

Weak formulation

Let us consider the Hilbert space $\mathcal{H}_0 = \{v \in \mathbb{H}^1(\Omega) : v = 0 \text{ on } \Gamma_D\}$.
The typical weak formulation can be obtained by multiplying the equation by a test function $v \in \mathcal{H}_0$ and integrating on the domain Ω :

$$-\int_\Omega \text{div}(K\nabla u)v \, \mathrm{d}x + \int_\Omega \text{div}(\beta u)v \, \mathrm{d}x + \int_\Omega cuv \, \mathrm{d}x = \int_\Omega fv \, \mathrm{d}x.$$

Application of Green's Lemma only to the first term yields:

$$-\int_{\Gamma_N} gv \, \mathrm{d}s + \int_\Omega K\nabla u \cdot \nabla v \, \mathrm{d}x + \int_\Omega \text{div}(\beta u)v \, \mathrm{d}x + \int_\Omega cuv \, \mathrm{d}x = \int_\Omega fv \, \mathrm{d}x,$$

from which we can deduce the following weak formulation:

Find $u \in \mathcal{H}_d = \{v \in \mathbb{H}^1(\Omega) : v = f_d \text{ on } \Gamma_D\}$ such that:

$$a(u, v) = (f, v) + (g, v)_{\Gamma_N} \quad \forall v \in H_0^1, \tag{1}$$

where:

$$a(u,v) = \int_\Omega K\nabla u \cdot \nabla v \ \mathrm{d}x + \int_\Omega \mathrm{div}(\beta u)v \ \mathrm{d}x + \int_\Omega cuv \ \mathrm{d}x$$
$$(f,v) = \int_\Omega fv \ \mathrm{d}x$$
$$(g,v)_{\Gamma_N} = \int_{\Gamma_N} gv \ \mathrm{d}s.$$

The presence of only first partial derivatives suggests the use of $\mathcal{P}1-$Galerkin space, so given a triangulation $\mathcal{T}_h$ of Ω, we consider the space $\mathcal{V}_h$:

$$\mathcal{V}_h = \left\{ v \in \mathcal{C}^0(\Omega) : v|_{T_i} \in \mathcal{P}_1(T_i) \quad \forall T_i \in \mathcal{T}_h \right\}$$

where $\mathcal{P}_1$ is the set of first degree. In particular, calling $\mathcal{N}_h$ the set of n nodes of $\mathcal{T}_h$, we consider the basis of $\mathcal{V}_h$ made of pyramidal functions ϕ_i. For each triangle of vertices $\{i,j,k\}$ there are exactly 3 basis functions non constantly zero in it: $\{\phi_i, \phi_j, \phi_k\}$. Reduced to the triangle $\mathcal{T}$, the generic ϕ_i has the form $\phi_i^T(x,y) = a_i^T + b_i^T x + c_i^T y$ and $\phi_i^T(i) = 1$, $\phi_i^T(j) = 0$, $\phi_i^T(k) = 0$. Hence a general function $u_h \in \mathcal{V}_h$ can be expressed as $u_h(x,y) = \sum_{i=1}^n u_i \phi_i(x,y)$ and considering the equation (1), the Galerking formulation of the problem is:

Find $u_h \in \mathcal{V}_h$ such that

$$a(u_h, \phi_j) = (f, \phi_j) + (q, \phi_j)_{\Gamma_N} \quad \forall j = 1, \ldots, n \tag{2}$$

Exploiting the linearity of the scalar product, we can finally write the Galerkin problem:

Solve

$$Au = b \tag{3}$$

where

$$A_{i,j} = a(\phi_i, \phi_j), \quad b_j = (f, \phi_j) + (q, \phi_j)_{\Gamma_N}.$$

Definition of the computational mesh

The implemented module operates on meshes defined on the square domain $[-1,1]\times[-1,1]$, that is divided in triangular elements. Let `Nelem` be the number of elements (triangles) and `Nodes` the number of vertices. The computational mesh is memorized by means of two matrices:

1. `coord` $\in$ Matrix(`Nodes`, 2) contains the two coordinates `coord`$[j,1]$ and `coord`$[j,2]$ for each mesh node j;

2. `triang` $\in$ Matrix(`Nelem`, 3) contains for each element `el` the global numbering of the nodes that form `el`. In other words, `triang` is the array implementing the map from the local element vertex numbers $(1,2,3)$ to the global vertex numbers of $\mathcal{T}_h$: for element `el`, the global node number m corresponding to the local node number j ($j = 1,2,3$) is given by $m =$ `Triang`(`el`, j).

The identification of the portions of the domain boundary characterized by Dirichlet or Neumann conditions is done in the code by specifying the set of nodes that belong to Γ_D or Γ_N.

In the folder there are 5 meshes, in which a consecutive decrease (division by 2) in the maximum diameter of the triangles can be observed. In this way it is possible to study the error convergence rate of the FEM.

1 The module

The implemented module, given the mesh and the input parameters, calculates the matrices and vectors:

- Stiffness matrix for $a(u,v) = \int_\Omega K\nabla u \cdot \nabla v \, \mathrm{d}x$,

$$A_{i,j} = a(\phi_i, \phi_j) = \int_\Omega D\frac{\partial \phi_i}{\partial x}\frac{\partial \phi_j}{\partial x} + D\frac{\partial \phi_i}{\partial y}\frac{\partial \phi_j}{\partial y} = \sum_{T\in\mathcal{T}_h}\int_T a_i^T a_j^T + b_i^T b_j^T$$

- Mass matrix for $\int_\Omega cuv \, \mathrm{d}x$

$$M_{i,j} = \int_\Omega c\phi_i\phi_j dx = \sum_{T\in\mathcal{T}_h}\int_T \phi_i\phi_j dx$$

- Transport matrix for $\int_\Omega \mathrm{div}(\beta u)v \, \mathrm{d}x$

$$B_{i,j} = \int_\Omega \left(\beta_x\frac{\partial \phi_j}{\partial x} + \beta_y\frac{\partial \phi_j}{\partial y}\right)\phi_i$$

- Forcing vector for $(f,v) = \int_\Omega fv \, \mathrm{d}x$

$$b_i = (f,\phi_i) = \int_\Omega f\phi_i = \sum_{T\in\mathcal{T}_h}\int_T f\phi_i$$

  and the trapezoidal rule is used to calculate b_i^T

- Neumann vector $(g,v)_{\Gamma_N} = \int_{\Gamma_N} gv \, \mathrm{d}s$

$$b_i^N = (g,\phi_i)_{\Gamma_N} = \int_{\Gamma_N} q\phi_i = \sum_{e\in\mathcal{L}}\int_e q\phi_i$$

  where the sum is along all the edges e of the triangles embeddedin Γ_N, and the trapezoidal rule was used to evaulate $b_i^{N,e}$.

- Dirichlet condition: Dirichlet boundary conditions are imposed using the penalty approach. This is obtained by substituting to the diagonal entry of the linear system matrix a large value, e.g. $R_{\max} = 10^{15}$, and setting the corresponding right-hand-side value to the Dirichlet value multiplied by $R_{\max}$.

- L2 error: after having computed the numerical solution, if the analytical one is available, the L2 error is computed as

$$e_h = \left[\int_\Omega (u_h - u)^2 dx \right]^{1/2} = \left[\sum_{T \in \mathcal{T}_h} \int_T (u_h - u)^2 dx \right]^{1/2}$$

Then there is a function that assembles the final linear system, so that its solution can be found, hence obtaining the numerical solution.

Stabilization

From Céa Lemma we have the following error estimate:

$$\|u - u_h\|_{H^1} \leq C_\Omega \frac{\|K\|_\infty + C_\Omega \|\beta\|_\infty + C_\Omega^2 \|c\|_\infty}{\inf K} \|u - v\|_{H^1} \qquad \forall v \in \mathcal{V}_h \quad (4)$$

So in a convection dominated system, where $\|K\|_\infty \ll \|\beta\|_\infty$, the error may be extremely large. For this reason if we are in this situation, the problem can be modified adding a component $\in \mathcal{O}(h)$ in order to reinforce the diffusion term and thus stabilize the problem without losing the convergence of the FEM ([1]).
The idea of the Streamline upwind stabilization is to add an artificial diffusion coefficient along the streamlines of the convection field β, as it is only along the velocity vector field that the upwind idea is most effective.
For this reason we add the matrix S to the linear system, that becomes:

$$(A + S)u = b$$

where:

$$S_{i,j} = s(\phi_i, \phi_j) = \delta \int_\Omega (\beta \cdot \nabla \phi_i)(\beta \cdot \nabla \phi_j).$$

and δ is a positive parameter to be tuned in order to trade off between stabilization and correctness of the solution.
The stabilization matrix is implemented in the code:

- $S_{i,j} = \delta \int_\Omega (\beta_x \frac{\partial \phi_i}{\partial x} + \beta_y \frac{\partial \phi_i}{\partial y})(\beta_x \frac{\partial \phi_j}{\partial x} + \beta_y \frac{\partial \phi_j}{\partial y})$

Difficulties of the project

The main difficulty encountered was in assembling the matrices that compose the linear system to be solved $Au = b$.
The process of building the matrix A starts by defining local elementwise matrices and build the global contributions by summing up these pieces.
The general i, j entry of A is:

$$a_{ij} = \sum_{T \in \mathcal{T}_h(\Omega)} a_{ij}^{(T)}$$

where matrix $A^{(T)}$ is of dimension $n^{(T)} \times n^{(T)}$, where $n^{(T)}$ is the number of nodes in the element.
For example, consider a triangle of nodes i, j, m, given in this order in the table

stored in matrix `triang`. Then element $a_{11}^{(T)}$ of the local matrix must be summed up to element a_{ii}^{T} of the global element, since i corresponds to the first node in `triang`, while the local element $a_{12}^{(T)}$ will be summed up to the global element a_{ij}, and so forth.

Unsolved and unsatisfactory

One possible improvement to the code would be to identify edge vertices for any given mesh, and not just in the addressed case of the square. In fact, the creation of the linear system works for any 2D mesh as long as it is triangular and the two matrices (`coord` and `triag`) are given, and identifying boundary nodes ad hoc is a limitation.

The next goal of the implemented module would be to also solve nonstationary PDEs, since the constraint of no time derivatives excludes many interesting differential equations.

References

[1] Alexander N Brooks and Thomas JR Hughes. Streamline upwind/petrov-galerkin formulations for convection dominated flows with particular emphasis on the incompressible navier-stokes equations. *Computer methods in applied mechanics and engineering*, 32(1-3):199–259, 1982.

[2] Claes Johnson. *Numerical solution of partial differential equations by the finite element method*. Courier Corporation, 2012.

[3] Alfio Quarteroni and Silvia Quarteroni. *Numerical models for differential problems*, volume 2. Springer, 2009.