DICACIM programme A.Y. 2023–24
Numerical Methods for Partial Differential Equations

Numerical solution of Advection Diffusion Reaction problems

Paola Gervasio

DICATAM, Università degli Studi di Brescia (Italy)

uniBS, May, 2024

UNIVERSITÀ DEGLI STUDI DI BRESCIA

$d = 1$, Advection Diffusion: fem_1d_adsolver.m

```
help fem_1d_adsolver
  fem_1d_adsolver: solve -mu u''+bu'+sigma u=f in Omega
  with Dirichlet and/or Neumann boundary conditions
  by either P1-fem or P2-fem on a uniform grid.

  [nodes,uh]=fem_1d_adsolver(geom,problem_data,p,Ne)
  [nodes,uh]=fem_1d_adsolver(geom,problem_data,p,Ne,degree)

  Input: geom: struct with fields:
         ....
  Output: nodes = column array with the nodes of the mesh
          uh = column array of the numerical solution
```

Problem 1 (1d Advection Diffusion (AD))

Solve the AD problem

$$\begin{cases} -\mu u'' + bu' = 0 & \text{in } \Omega = (0,1) \\ u(0) = 0, \ u(1) = 1. \end{cases}$$

(with $\mu = 1, \ 0.1, \ 0.02$ and $b = 1$) by FEM-$\mathbb{P}_1$ and FEM-$\mathbb{P}_2$. Take $Ne = 10, \ 20, \ 30, \ 40, \ 80, \ 160, \ 320$.

1. Compute the Péclet number $\mathbb{P}e = \frac{b}{\mu}\frac{h}{2}$ and verify that the numerical solution does not show oscillations when $\mathbb{P}e < 1$.
2. The exact solution of the equation is

$$u(x) = \frac{e^{xb/\mu} - 1}{e^{b/\mu} - 1}.$$

   Verify that the error $\|u - u_h\|_{H^1(\Omega)}$ converges linearly to zero for $\mathbb{P}_1-$fem and quadratically for $\mathbb{P}_2-$fem, provided that $\mathbb{P}e < 1$.

UNIVERSITÀ DEGLI STUDI DI BRESCIA

Problem 2 (1d Advection Diffusion (AD))

Solve the problem

$$\begin{cases} -\mu u'' + bu' = 0 & \text{in } \Omega = (0,1) \\ u(0) = 0, \ u(1) = 1. \end{cases}$$

(with $\mu = 0.02$ and $b = 1$) by FEM-$\mathbb{P}_1$ and FEM-$\mathbb{P}_2$, by using the **artificial diffusion** method (i.e., replace μ with $\mu_h = \mu(1 + \mathbb{P}e)$)
Take $Ne = 10, \ 20, \ 30, \ 40, \ 80, \ 160, \ 320.$

1. Verify that spurious oscillations are missing for eany value of h, even when $\mathbb{P}e > 1$.
2. The exact solution is

$$u(x) = \frac{e^{xb/\mu} - 1}{e^{b/\mu} - 1},$$

   verify that the error $\|u - u_h\|_{H^1(\Omega)}$ converges linearly to zero both for $\mathbb{P}_1-$fem and $\mathbb{P}_2-$fem

Problem 3

Solve the problem

$$\begin{cases} -\mu\Delta u + \mathbf{b}\cdot\nabla u + \sigma u = f & \text{in } \Omega = (0,1)^2 \\ u = g_D & \text{on } \partial\Omega \end{cases}$$

where $\mu = 10^{-3}$, $\mathbf{b} = [1,1]^T$, $\sigma = 0$, $f = 1$, $g_D = 0$.

1. Set $\delta = 0$ (classical Galerkin), and compute the numerical solution with $h = 1/20$ and $h = 1/80$.
2. set $\delta = 1$ and compute the numerical solution with $h = 1/20$ and $h = 1/80$.
3. Repeat the work with $\mu = 10^{-5}$ instead of $\mu = 10^{-3}$.

Advection Diffusion Reaction, $d = 2$

```
% FEM_2D_AD Computes the GaLS-FEM-P1 approximation of
%  -mu Delta u+ b . nabla u +sigma u=f in Omega (2D domain)
%  u=g_D Dirichlet conditions on the boundary.
%
%  GaLS=Galerkin-Least-Squares stabilization
 [u,FEM]=fem_2d_ad(model,problem_data,parameters)
  Input:
         model = model object generated by MATLAB (it must contain
         field model.Mesh)
         problem_data = struct with fields:
           .mu = coefficient (constant) of 2nd order term
           .b = velocity field (constant) of 1st order term
           .....
  Output:
          u = numerical solution
          FEM = struct with fields:
           .Kc:   % matrix A0, (i,j) non-dirichlet
           .Fc:   % rhs f0, (i) non-dirichlet
           .....
```

Galerkin Least Squares GaLS stabilization

GaLS formulation: find $u_h \in V_h$:

$$a(u_h, v_h) + \sum_k \int_{T_k} Lu_h \; \tau_k \; Lv_h = F(v_h) + \sum_k \int_{T_k} f \; \tau_k \; Lv_h \qquad \forall v_h \in V_h$$

with $\tau_k = \delta \frac{h_k}{|\mathbf{b}|}$ **stabilization parameter** $\delta > 0$.

1. Because $Lu = f$, (u is the exact solution), we have
$$\sum_k \int_{T_k} Lu \; \tau_k \; Lv_h = \sum_k \int_{T_k} f \; \tau_k \; Lv_h \qquad \forall v_h \in V_h,$$ this method is strongly consistent, meaning that the exact solution satisfies the discrete equation exactly.
2. For each $v_h \in X_h$ (the space $\mathbb{P}_1$), because μ is costant, it holds $-\nabla \cdot (\mu \nabla v_h) = 0$, thus
$$\int_{T_k} Lu_h \; \tau_k \; Lv_h = \int_{T_k} (\mathbf{b} \cdot \nabla u_h + \sigma u_h) \tau_k (\mathbf{b} \cdot \nabla v_h + \sigma v_h).$$
3. If $\delta = 0$, GaLS reduces to classical Galerkin method.

Solution of Problem 3

```
model=createpde; % build the model
% build the geometry
Q1=[3,4, 0,1,1,0, 0,0,1,1]';
gd=[Q1];
ns=(char('Q1'))';
sf='Q1';
g=decsg(gd,sf,ns);
geometryFromEdges(model,g);
% build the mesh
generateMesh(model,'Hmax',1/20,...
        'GeometricOrder','linear');
% set problem_data and parameters
% call fem_2d_ad
[u]=fem_2d_ad(model,problem_data,parameters);
% plot the solution
pdeplot(model,'XYData',u,'ZData',u,...
     'Contour','on','Mesh','on')
```


UNIVERSITÀ DEGLI STUDI DI BRESCIA

Description of the output struct `FEM`

Let us consider the problem

$$\begin{cases} Lu = -\mu\Delta u + \mathbf{b}\cdot\nabla u + \sigma u = f & \text{in } \Omega \\ u = g_D & \text{on } \partial\Omega \end{cases}$$

with constant $\mu > 0$, constant $\mathbf{b}$, constant $\sigma \geq 0$.

The weak form reads: **find** $u = u_0 + Rg_D$, where:

– u_0 is the **homogeneous part** of u

$$u_0 \in H_0^1(\Omega): \quad a(u_0, v) = F(v) \quad \forall v \in H_0^1(\Omega)$$

with

$$a(u,v) = \int_\Omega \mu\nabla u\cdot\nabla v + \int_\Omega (\mathbf{b}\cdot\nabla u)v + \int_\Omega \sigma uv, \qquad F(v) = \int_\Omega fv$$

– while $Rg_D \in H^1(\Omega)$ is the **lifting of** g_D, i.e., $Rg_D = g_D$ on $\partial\Omega$.

We recall that, for FEM-$\mathbb{P}_1$, we have:

$$X_h = \{v \in C^0(\overline{\Omega}) : \ v|_{T_k} \in \mathbb{P}_1, \ \forall T_k \in \mathcal{T}_h\}$$

while the Lagrangian basis is $\beta = \{\varphi_i\}_{i=1}^{N_h}$ (including the basis functions associated with the Dirichlet nodes). Set:

$$\mathcal{I} = \{1, 2, \ldots, N_h\}, \quad \mathcal{I}_D = \{i \in \mathcal{I} : \ \mathbf{x}_i \in \partial\Omega_D\}, \quad \mathcal{I}_0 = \mathcal{I} \setminus \mathcal{I}_D.$$

The numerical solution is

$$u_h(x) = \sum_{j=1}^{N_h} u_j \varphi_j(x) = \underbrace{\sum_{j \in \mathcal{I}_0} u_j \varphi_j(x)}_{u_{0,h}} + \underbrace{\sum_{j \in \mathcal{I}_D} u_j \varphi_j(x)}_{Rg_{D,h}}$$

with **discrete lifting**

$$Rg_{D,h}(\mathbf{x}_i) = \begin{cases} g_D(\mathbf{x}_i) & \text{if } \mathbf{x}_i \in \partial\Omega_D \\ 0 & \text{otherwise.} \end{cases}$$

Set

- A matrix: $A_{ij} = a(\varphi_j, \varphi_i)$ with $i, j \in \mathcal{I}$,
- A_0 the block of A with $i, j \in \mathcal{I}_0$,
- A_D the block of A with $i \in \mathcal{I}_0$ e $j \in \mathcal{I}_D$,
- $\mathbf{f}$ array: $f_i = F(\varphi_i)$ with $i \in \mathcal{I}$,
- $\mathbf{f}_0$ block of $\mathbf{f}$ with $i \in \mathcal{I}_0$,
- $\mathbf{g}_D$ block of $g_D(\mathbf{x}_i)$ with $i \in \mathcal{I}_D$,
- $\mathbf{u}_0 = [u_j]_{j \in \mathcal{I}_0}$, non-Dirichlet dof,
- $\mathbf{u}_D = [u_j]_{j \in \mathcal{I}_D}$, Dirichlet dof.

The algebraic formulation of the discrete Galerkin problem reads:

$$\begin{bmatrix} A_0 & A_D \\ 0 & I \end{bmatrix} \begin{bmatrix} \mathbf{u}_0 \\ \mathbf{u}_D \end{bmatrix} = \begin{bmatrix} \mathbf{f}_0 \\ \mathbf{g}_D \end{bmatrix} \qquad \begin{array}{l} \leftarrow i \in \mathcal{I}_0 \text{ non-Dirichlet nodes} \\ \leftarrow i \in \mathcal{I}_D \text{ Dirichlet nodes} \end{array}$$

We have

$$\mathbf{u}_D = \mathbf{g}_D, \quad A_0\mathbf{u}_0 + A_D\mathbf{u}_D = \mathbf{f}_0$$

and

$$A_0\mathbf{u}_0 = \mathbf{f}_0 - A_D\mathbf{g}_D.$$

Finally, we reconstruct

$$\mathbf{u} = \begin{bmatrix} \mathbf{u}_0 \\ \mathbf{0} \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \mathbf{u}_D \end{bmatrix}$$

- `FEM.Kc` contains the matrix A_0
- `FEM.Fc` contains the array $\mathbf{f}_0 - A_D\mathbf{g}_D$
- `FEM.B` contains the matrix B s.t. $B\mathbf{u}_0 = \begin{bmatrix} \mathbf{u}_0 \\ \mathbf{0} \end{bmatrix}$
- `FEM.ud` contains the array $\begin{bmatrix} \mathbf{0} \\ \mathbf{u}_D \end{bmatrix}$

To compute **u**:

```
u0=FEM.Kc\FEM.Fc; % solve FEM.Kc u0 = FEM.Fc
u=FEM.B*u0+FEM.ud; % reconstruct u
```


UNIVERSITÀ DEGLI STUDI DI BRESCIA
Paola Gervasio - DICATAM - unibs.it