

Finite Elements in Fluid Mechanics, Homework 1: 'Convection-Diffusion-Reaction equation'

Mauricio Barrera C.

May 22, 2017

The model equation can be taken first in its steady state form:

$$\mathbf{a} \cdot \nabla \rho + \nabla \cdot (\mu \nabla \rho) + \sigma \rho = s,$$

or

$$\mathcal{L}(\rho) - s = 0.$$

In this example a square domain of side length 1 is employed with Dirichlet boundary conditions $\rho = 1$ on $x = 0$ side and points above $y > 0.2$; and $\rho = 0$ on $x = 0$ side with points $y < 0.2$ and all points on $y = 0$ side. Zero-flux condition is taken everywhere else.

The weak form of the problem is produced by means of a weight function w applied to each term:

$$(w, \mathbf{a} \cdot \nabla \rho) + (w, \nabla \cdot (\mu \nabla \rho)) + (w, \sigma \rho) - (w, s) = 0.$$

Integration by parts and consideration of discrete spaces w^h and ρ^h make the terms in the equation above produce the following contributions to what would be the stiffness matrix in an equilibrium equation:

$$\begin{aligned} (w, \mathbf{a} \cdot \nabla \rho) &\rightarrow (\mathbf{a}; w^h, \rho^h) \rightarrow \mathbf{N}^T a_x \mathbf{N}_x + \mathbf{N}^T a_y \mathbf{N}_y, \\ (w, \nabla \cdot (\mu \nabla \rho)) &\rightarrow a(w^h, \rho^h) \rightarrow \mathbf{N}_x^T \mu \mathbf{N}_x + \mathbf{N}_y^T \mu \mathbf{N}_y, \\ (w, \sigma \rho) &\rightarrow (w^h, \sigma \rho^h) \rightarrow \mathbf{N}^T \sigma \mathbf{N}, \end{aligned}$$

where \mathbf{N} is the vector of interpolation functions ($[N_1 \ N_2 \ N_3 \ N_4]^T$ in the bilinear quadrilateral employed here), and the subscript indicates spatial derivative.

There are indeed boundary terms emerging from the integration by parts of the diffusion term. These capture the flux conditions across boundaries. Being these equal to zero, they are not elaborated on here. Furthermore, the stabilisation term (either SUPG or GLS) adds up to the stiffness matrix. There is, though, a contribution to the force vector from this fact, if there is a source term s different from zero; but the focus here is on that part of the formulation contributing to the stiffness matrix.

The SUPG stabilisation method adds the following term to the stiffness matrix (Donea and Huerta, 2003):

$$\iiint_V (\mathbf{a} \cdot \nabla w^h) \tau (\mathbf{a} \cdot \nabla \rho^h + \nabla \cdot (\mu \nabla \rho^h) + \sigma \rho^h) dV$$

whereas the contribution from SUPG to the force vector comes from the operation

$$\iint_S (\mathbf{a} \cdot \nabla w^h) \tau s dS.$$

The use of linear interpolation functions makes the contribution from the diffusion term $\nabla \cdot (\mu \nabla \rho^h)$ to the SUPG scheme disappear. A higher order element would bring this contribution back; however, that would imply the computation of a double-derivative Jacobian to produce terms of the form \mathbf{N}_{xx} , when using isoparametric functions. One such term could be developed as follows:

$$\begin{aligned} \frac{dN}{d\xi} &= \frac{dN}{dx} \frac{dx}{d\xi} + \frac{dN}{dy} \frac{dy}{d\xi} \\ \frac{d^2 N}{d\xi^2} &= \left(\frac{d}{d\xi} \left(\frac{dN}{dx} \right) \right) \frac{dx}{d\xi} + \frac{dN}{dx} \frac{d^2 x}{d\xi^2} \end{aligned}$$

$$\frac{d^2 N}{d\xi^2} = \left(\frac{d}{dx}\left(\frac{dN}{dx}\right)\right)\left(\frac{dx}{d\xi}\right)^2 + \frac{dN}{dx} \frac{d^2 x}{d\xi^2}$$

$$\frac{d^2 N}{d\xi^2} = \frac{d^2 N}{dx^2} \left(\frac{dx}{d\xi}\right)^2 + \frac{dN}{dx} \frac{d^2 x}{d\xi^2}$$

So, even though the term $\frac{d^2 N}{dx^2}$ can be solved for and all the rest of terms can be readily computed or have already been computed (like $\frac{dN}{dx}$ in a first instance of the application of the Jacobian), it can be seen that the full implementation of this process can be cumbersome.

Hence, considering the use of bilinear elements, the additional terms to the stiffness matrix from SUPG stabilisation are:

$$(\mathbf{N}_x^T \tau a_x a_x \mathbf{N}_x) + (\mathbf{N}_y^T \tau a_y a_y \mathbf{N}_y) + (\mathbf{N}_x^T \tau a_x \sigma \mathbf{N}) + (\mathbf{N}_y^T \tau a_y \sigma \mathbf{N})$$

In the GLS stabilisation method, the term $(\mathbf{a} \cdot \nabla w^h)$ which multiplies $\mathcal{L}(\rho) - s$ is substituted by $\mathcal{L}(w)$, where w is the weighting function we have been using thus far. There are more contribution terms to the stiffness matrix now, but the rationale underlying their derivation has nothing new when compared to the SUPG already presented.

Implementation of these changes in a Matlab code include modifications to the loops assembling the stiffness matrix and force terms. For instance, the SUPG loop goes as:

```
% SUPG
Ke = Ke + (nu*(Nx'*Nx+Ny'*Ny) + N_ig'*(ax*Nx+ay*Ny) + ...
    tau*(ax*Nx+ay*Ny)'*(ax*Nx+ay*Ny))*dvolu + ...
    ((sig*N_ig'*N_ig)+(tau*ax*sig*Nx'*N_ig)+(tau*ay*sig*Ny'*N_ig))*dvolu;
aux = N_ig*Xe;
f_ig = SourceTerm(aux);
fe = fe + (N_ig+tau*(ax*Nx+ay*Ny))'*(f_ig*dvolu);
```

whereas the GLS loop is:

```
% GLS
Ke = Ke + (nu*(Nx'*Nx+Ny'*Ny) + N_ig'*(ax*Nx+ay*Ny) + (sig*N_ig'*N_ig) + ...
    tau*(ax*Nx+ay*Ny)'*(ax*Nx+ay*Ny) + tau*(ax*sig*Nx+ay*sig*Ny)'*N_ig + ...
    tau*(sig*N_ig'*(ax*Nx+ay*Ny)) + (tau*sig*sig*N_ig'*N_ig))*dvolu + ...
    ((sig*N_ig'*N_ig)+(tau*ax*sig*Nx'*N_ig)+(tau*ay*sig*Ny'*N_ig))*dvolu;
aux = N_ig*Xe;
f_ig = SourceTerm(aux);
fe = fe + (N_ig+tau*(ax*Nx+ay*Ny+sig*N_ig))'*(f_ig*dvolu);
```

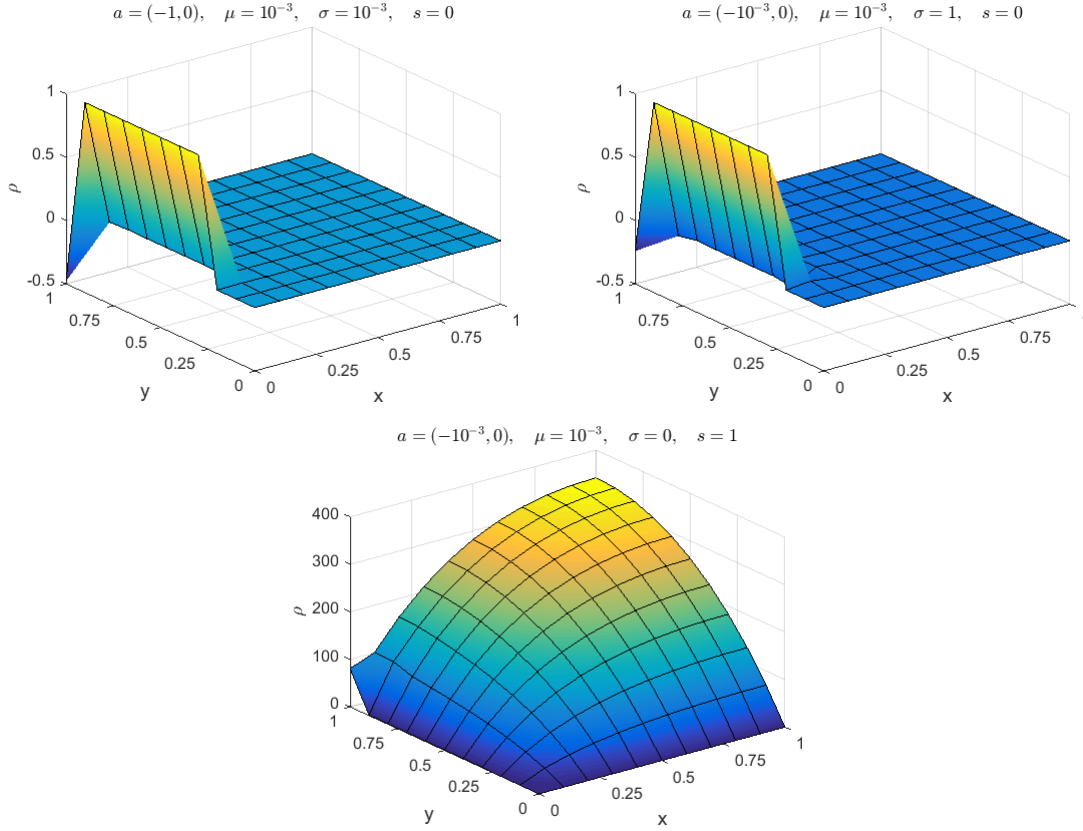
The GLS scheme option is made available to the user by inserting the above lines into an additional case conditional, just as the originally available SUPG is. In addition, the main script features a line to display the option to the user and make it part of the example data array under the parameter `example.method`:

```
disp('The following methods can be used:');
disp('    [0] Galerkin');
disp('    [1] Artificial diffusion');
disp('    [2] SUPG');
disp('    [3] GLS');
method = input('Choose a method for solving the problem: ');
example.method = method;
```

In all situations the stabilisation parameter τ is taken as (Donea and Huerta, 2003):

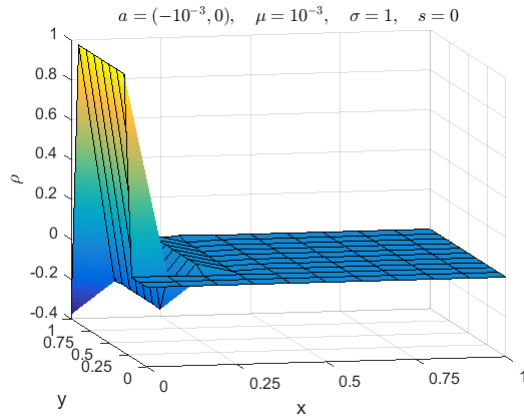
$$\tau = \frac{h}{2a} \left(1 + \frac{9}{Pe^2} + \left(\frac{h\sigma}{2a}\right)^2\right)^{-\frac{1}{2}},$$

where Pe stands for the non-dimensional Peclet number and h is a characteristic element size. Some simulation results using the SUPG scheme are set out below:



The effect of the convection velocity and its negative x direction can be appreciated in the first two cases. The similarity of the plots, in view of the changes in reaction coefficient σ , indicate that the convection dominates the equilibrium state of the convection-diffusion process. The last plot, however, shows that being there a source, this is the overly dominant factor: the convective and diffusive mechanisms have a small influence on the overall response. This may be a matter of the units used for each of these quantities and how feasible they can be in actual life.

In regards of the GLS scheme, only the second condition, namely $a = (-10^{-3}, 0), \mu = 10^{-3}, \sigma = 1, s = 0$, showed some difference worth of a remark:



Albeit the general conclusions in regards of the physical phenomenon stay the same, the GLS simulation shows a tendency to oscillate when compared to SUPG. Being both consistent stabilisation schemes, it can be suggested that the latter introduces some more numerical diffusive action into the simulation.

As regards the unsteady convection-diffusion-reaction problem, the presence of the diffusion term prescribes the use of single step time marching schemes, unless these involves only first order derivatives in time. One alternative is the use of multi-step time integration schemes using one of several techniques available, such as the use of Padé approximations (Donea and Huerta, 2003). The so-called implicit Padé schemes may arranged in the form:

$$\frac{\Delta \rho}{\Delta t} - \mathbf{W} \Delta \rho_t = \mathbf{w} \rho_t^n$$

Among these we have the one-stage second order accurate R_{11} for which:

$$\mathbf{W} = \frac{1}{2}, \quad \mathbf{w} = 1$$

And the two-stage, fourth order accurate R_{22} , where

$$\mathbf{W} = \frac{1}{24} \begin{bmatrix} 7 & -1 \\ 13 & 5 \end{bmatrix}, \quad \mathbf{w} = \frac{1}{2} \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

In either of these schemes ρ_t can be solved for from the original governing equations and then substituted into $\Delta\rho_t$ and ρ_t^n above. The result is arranged so that $\Delta\rho$ are the nodal variables and thus moved to the left hand side of the time-discretised equation:

$$\frac{\Delta\rho}{\Delta t} + \mathbf{W}(\mathbf{a} \cdot \nabla + \mu\nabla^2 + \sigma)\Delta\rho = \mathbf{W}s^{n+1} + (\mathbf{I} - \mathbf{W})s^n - (\mathbf{a} \cdot \nabla + \mu\nabla^2 + \sigma)\rho^n$$

Although superscripts n and $n + 1$ have been used, they actually represent advance by fractional steps, in view of the multi-stage nature of the integration scheme.

Explicit schemes based on Padé approximations cannot be arranged in this way. In a two-stage explicit scheme R_{20} can be rearranged as follows:

$$\begin{aligned} \rho^{n+\frac{1}{2}} &= \rho^n + \frac{\Delta t}{2} \rho_t^n \rightarrow \frac{\rho^{n+\frac{1}{2}} - \rho^n}{\Delta t} = \frac{1}{2} \rho_t^n \rightarrow \frac{\Delta^{n+\frac{1}{2}}\rho}{\Delta t} = \frac{1}{2} \rho_t^n \\ \rho^{n+1} &= \rho^n + \Delta t \rho_t^{n+\frac{1}{2}} \rightarrow \frac{\rho^{n+1} - \rho^n}{\Delta t} = \rho_t^{n+\frac{1}{2}} \rightarrow \frac{\Delta^{n+1}\rho}{\Delta t} = \rho_t^{n+\frac{1}{2}} \end{aligned}$$

Again, ρ_t is solved from the governing differential equation and substituted in the expressions above. It can be seen that solution of nodal unknown $\Delta\rho$ is simpler in this explicit scheme, even though it is solved for a fraction of a step at a time.

The spatial discretisation is given, in its general pattern, by the Galerkin weighted residual method. This gives the outline of matrix terms that will also be used in a balanced spatial discretisation. As discussed earlier, consideration of SUPG or GLS only adds up (consistent) terms emerging upon application of the Galerkin method alone. Hence, only the matrix terms thereby produced are shown here:

$$\begin{aligned} &(w, \frac{\Delta\rho}{\Delta t}) + \mathbf{W}(c(\mathbf{a}; w, \Delta\rho) + a(w, \Delta\rho) + (w, \sigma\Delta\rho)) = \\ &-(c(\mathbf{a}; w, \rho^n) + a(w, \rho^n) + (w, \sigma\rho^n)) + (w, \mathbf{w}s^{n+1} + (\mathbf{1} - \mathbf{w})s^n) + (w, \mathbf{w}h^{n+1} + (\mathbf{1} - \mathbf{w})h^n), \end{aligned}$$

where h represents the prescribed flux at boundaries.

The matrix terms emerging are:

$$\begin{aligned} (w, \frac{\Delta\rho}{\Delta t}) &\rightarrow \mathbf{M} = \iiint_V \mathbf{N}^T \mathbf{N} dV \\ c(\mathbf{a}; w, \Delta\rho) &\rightarrow \mathbf{C} = \iiint_V \mathbf{N}^T \mathbf{N}_x dV \\ a(w, \Delta\rho) &\rightarrow \mathbf{K} = \iiint_V \mathbf{N}_x^T \mu \mathbf{N}_x dV \\ (w, \sigma\Delta\rho) &\rightarrow \mathbf{B} = \iiint_V \mathbf{N}^T \sigma \mathbf{N} dV. \end{aligned}$$

Finally, the Galerkin scheme expression also produces boundary and body source terms that make up the force vector:

$$\begin{aligned} (w, \mathbf{w}h^{n+1} + (\mathbf{1} - \mathbf{w})h^n) &\rightarrow \mathbf{H} = \iint_S \mathbf{N}^T h dS \\ (w, \mathbf{w}s^{n+1} + (\mathbf{1} - \mathbf{w})s^n) &\rightarrow \mathbf{f} = \iiint_V \mathbf{N}^T s dV. \end{aligned}$$