UNIVERSITAT POLITÈNCICA DE CATALUNYA

Convection-diffusion equation

Rafel Perelló i Ribas

March 8, 2019

1 INTRODUCTION

In this assignment, the convection-diffusion problem has been solved in a 1D domain. Quadratic and linear elements have been used to discretise the domain. The formulation used are the Galerkin, Streamline Upwind (SU), Streamline Upwind Petrov Galerkin (SUPG) and Galerkin Least-Squares (GLS).

2 DERIVATION

First of all, the strong form of the problem is given. In this assignment, only Dirichlet boundary conditions are used, so the problem reduces to:

$$\begin{cases} au_x - \nu u_{xx} = s \quad x \in [0,1] \\ u(0) = u_0 \\ u(1) = u_1 \end{cases}$$

The first step is to obtain the weak form of the problem. For that, the equation is multiplied by a test function w and integrated over the domain:

$$\int_0^1 w \cdot (au_x - \nu u_{xx}) dx = \int_0^1 ws dx$$

The diffusive term in the integral can be integrated by parts to obtain the weak form of the problem:

$$\int_0^1 w \cdot (au_x) dx + \int_0^1 w_x \nu u_x dx = \int_0^1 ws dx + [w \nu u_x]_0^1$$

As both boundary conditions are of Dirichlet type, the last term is null. The solution is approximated as a linear combination of basis functions: $u \approx u^h = \sum_{i=1}^n N_i(x) u_i$.

With this approach, the vector of nodal values $\boldsymbol{u}$ can be obtained by the solution of the lineal system:

$$\boldsymbol{K}\boldsymbol{u} = \boldsymbol{f}$$

The Galerkin formulation uses the test functions as the same that the basis functions: $w_i = N_i$. That means that the stiffness matrix and the force vector can be calculated as:

$$\begin{aligned} K_{ij} &= \int_0^1 N_i a \frac{\partial N_j}{\partial x} dx + \int_0^1 \frac{\partial N_i}{\partial x} \nu \frac{\partial N_j}{\partial x} dx \\ f_i &= \int_0^1 N_i s dx \end{aligned}$$

This formulation is unstable for Péclet numbers larger than 1. For that reason, the SU method was designed. This method adds artificial diffusion in the direction of the velocity to the problem to stabilise the solution. The weak form is modified as follows:

$$\int_0^1 w \cdot (a u_x) dx + \int_0^1 w_x \nu u_x dx + \int_0^1 \tau (a w_x) \cdot (a u_x) dx = \int_0^1 w s dx$$

This results in the following form of the stiffness matrix and force vector:

$$\begin{aligned} K_{ij} &= \int_0^1 N_i a \frac{\partial N_j}{\partial x} dx + \int_0^1 \frac{\partial N_i}{\partial x} \nu \frac{\partial N_j}{\partial x} dx + \int_0^1 \tau \left(a \frac{\partial N_i}{\partial x} \right) \cdot \left(a \frac{\partial N_j}{\partial x} \right) dx \\ f_i &= \int_0^1 N_i s dx \end{aligned}$$

The disadvantage of this method is that is not consistent. That means that the solution does not necessarily converge to the exact solution when the mesh is refined. This is solved with methods that adds terms multiplied by the residual of the equation instead of just the advection like term. This would require the basis functions to be in $\mathscr{H}^2$. This is not the case as the first derivatives of the basis functions are discontinuous along the edges of the elements. That is solved by replacing the whole integral by the sum of the integral in each element. The added term is of the form of:

$$\sum_e \int_{\Omega_e} \tau \cdot \mathscr{P}(\omega) \cdot \mathscr{R}(u) d\Omega = \sum_e \int_{\Omega_e} \tau \cdot \mathscr{P}(\omega) \cdot \left(a \frac{\partial N_j}{\partial x} - \nu \frac{\partial^2 N_j}{\partial x^2} - s \right) dx$$

Taking the $\mathscr{P}$ function as the advection term, the SUPG is obtained:

$$\begin{aligned} K_{ij} &= \int_0^1 N_i a \frac{\partial N_j}{\partial x} dx + \int_0^1 \frac{\partial N_i}{\partial x} \nu \frac{\partial N_j}{\partial x} dx + \sum_e \int_{\Omega_e} \tau \left(a \frac{\partial N_i}{\partial x} \right) \cdot \left(a \frac{\partial N_j}{\partial x} - \nu \frac{\partial^2 N_j}{\partial x^2} \right) dx \\ f_i &= \int_0^1 N_i s dx + \sum_e \int_{\Omega_e} \tau \left(a \frac{\partial N_i}{\partial x} \right) \cdot s dx \end{aligned}$$

The SUPG method is consistent but is preferable to add symmetric terms to the stiffness matrix. For that reason, in the GLS method, the $\mathscr{P}$ function takes the form as the residual, but only the homogeneous part:

$$\begin{aligned} K_{ij} &= \int_0^1 N_i a \frac{\partial N_j}{\partial x} dx + \int_0^1 \frac{\partial N_i}{\partial x} \nu \frac{\partial N_j}{\partial x} dx + \sum_e \int_{\Omega_e} \tau \left(a \frac{\partial N_i}{\partial x} - \nu \frac{\partial^2 N_i}{\partial x^2} \right) \cdot \left(a \frac{\partial N_j}{\partial x} - \nu \frac{\partial^2 N_j}{\partial x^2} \right) dx \\ f_i &= \int_0^1 N_i s dx + \sum_e \int_{\Omega_e} \tau \left(a \frac{\partial N_i}{\partial x} - \nu \frac{\partial^2 N_i}{\partial x^2} \right) \cdot s dx \end{aligned}$$

3 Numerical examples with linear elements

All previous method have been implemented with MATLAB language for linear and quadratic one dimensional elements. First, linear elements are presented. Two problems have been tested to compare the performance of every method. For all of them, the τ parameter have been calculated as follows:

$$\tau = \frac{h}{2a}\left(coth(Pe) - \frac{1}{Pe}\right)$$

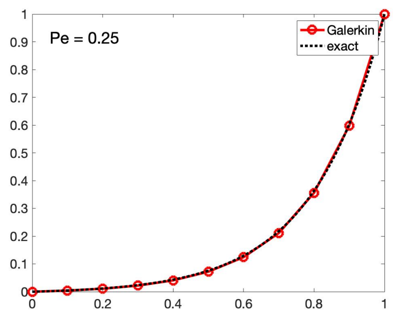
3.1 First problem

In this first problem there is no source term. The problem is:

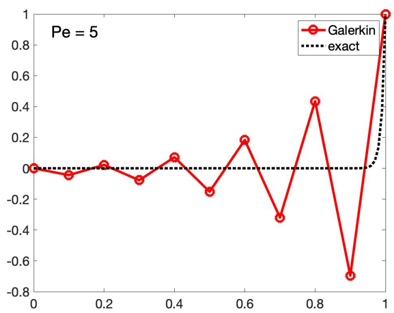
$$\begin{cases} a \cdot u_x - \nu u_{xx} = 0 \quad x \in [0,1] \\ u(0) = 0; u(1) = 1 \end{cases}$$

The numerical results are the following ones:

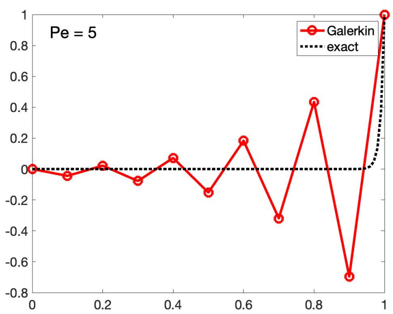
3.1.1 Galerkin method



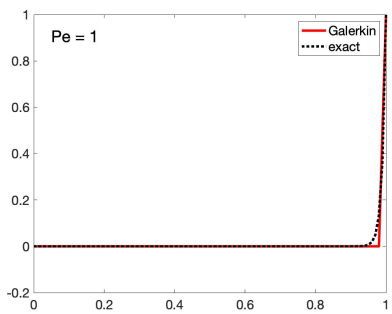
(a) a=1, ν=0.2, 10 linear elements



(b) a=20, ν=0.2, 10 linear elemetns



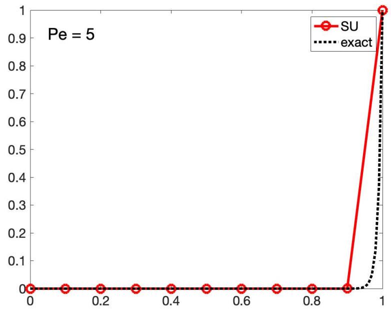
(c) a=1, ν=0.01, 10 linear elements



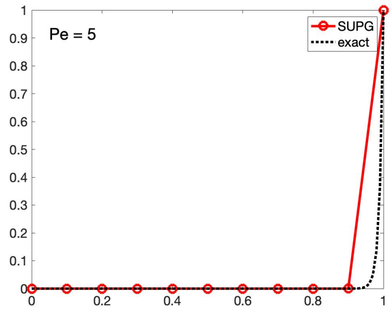
(d) a=1, ν=0.01, 50 linear elemetns

3.1.2 Stabilised methods

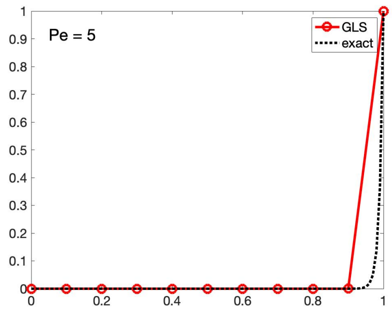
The parameters used are: a=1, ν=0.01, 10 linear elements



(a) SU method



(b) SUPG method

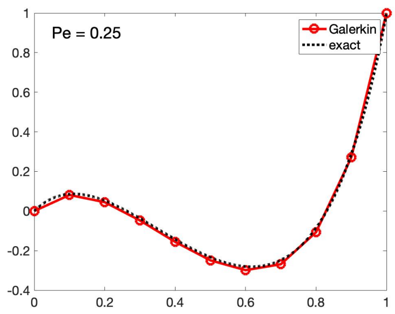


(c) GLS method

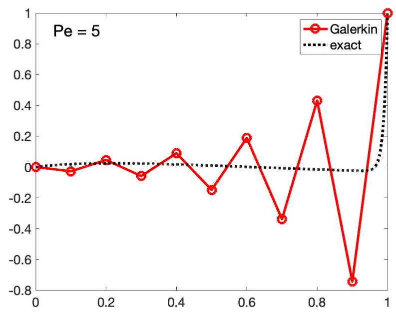
3.2 Second problem

The problem is now identical but it has a force term of the form $s = 10e^{-5x} - 4e^{-x}$. It has been solved using Galerkin and stabilised formulations:

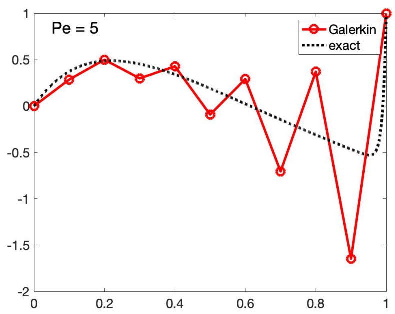
3.2.1 Galerkin method



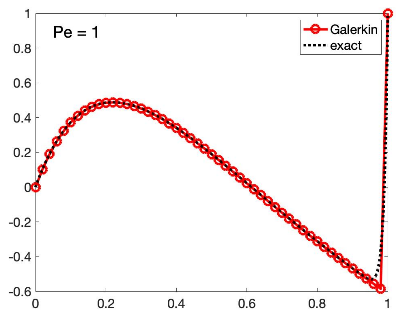
(a) a=1, ν=0.2, 10 linear elements



(b) a=20, ν=0.2, 10 linear elemetns



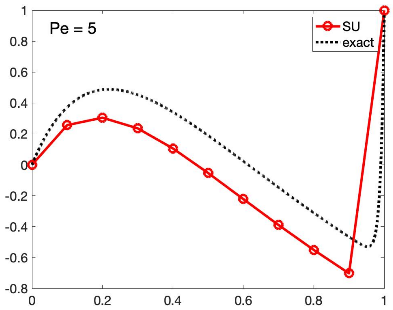
(c) a=1, ν=0.01, 10 linear elements



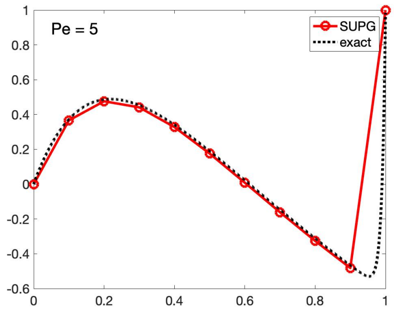
(d) a=1, ν=0.01, 50 linear elemetns

3.2.2 *Stabilised methods*

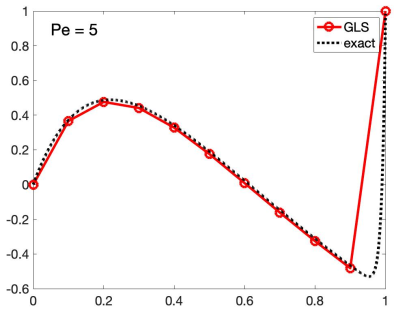
The parameters used are: a=1, ν=0.01, 10 linear elements



(a) SU method



(b) SUPG method



(c) GLS method

3.3 conclusions

In the solutions obtained it can be seen, as expected, that the Galerkin method is unstable for Péclet numbers greater than 1. About the rest of the methods, they give exact nodal solutions using linear elements if the source term is null. Otherwise, they no long give exact nodal results and, in particular, it can be noted the non-consistent nature of the SU method.

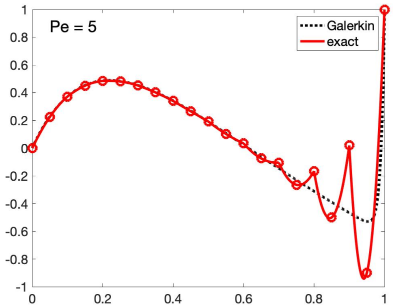
4 Numerical examples with quadratic elements

To include quadratic elements the following changes have been made in the code:

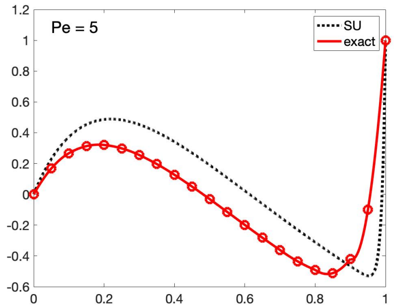
1. Add the second derivatives term in the diffusion.
2. Change the nodal coordinates and the connectivity matrices
3. Add the parabolic shape in the plot of the results.

The forced problem have been solved using 10 quadratic elements:

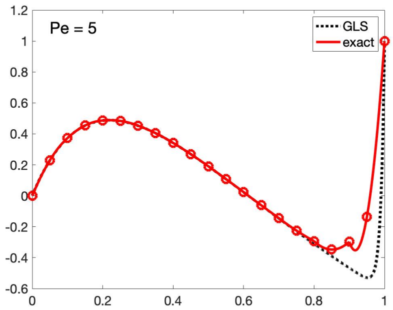
4.1 Numerical results of quadratic elements



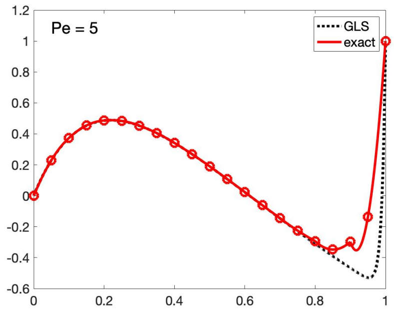
(a) Galerkin method



(b) SU method



(c) SUPG method



(d) GLS method

4.1.1 Conclusions

The Galerkin method is still unstable as the Péclet number is greater than 1. About the rest, the SU method is evidently non-consistent and SUPG and GLS lose some accuracy although having more degrees of freedom. That is due to the fact that in order to preserve nodal exact values when no force term is given, the τ parameter should be modified. In particular, when referred to the corner nodes of the element the value of the added viscosity coefficient should be modified [1]:

$$\tau_{corner} = \frac{h}{2a}\frac{(coth(Pe) - 1/Pe) - (cosh(Pe))^2 \cdot (coth(2Pe) - 1/(2Pe))}{1 - (cosh(Pe))^2/2}$$

This can be done by using a matrix form of the value of τ for each mode instead of a scalar. However, this option has not been implemented in the code.

REFERENCES

[1] Jean Donea and Antonio Huerta. *Finite Element Method for Flow Problems.* John Wiley & Sons, Ltd, 2005.