UNIVERSITAT POLITÈCNICA DE CATALUNYA BARCELONATECH

Swansea University Prifysgol Abertawe

CENTRALE NANTES

Universität Stuttgart

Master of Science in Computational Mechanics

Finite Elements in Fluids

MATLAB Assignment 1: Stabilization Techniques

Submitted By:
Mario Alberto Méndez Soto

Submitted To:
Prof. Pablo Saez
Prof. Antonio Huerta

Spring Semester, 2019

1 Strong problem and boundary conditions

Convection-diffusion phenomena can be described using the following general expression:

$$\frac{\partial u}{\partial t} + \mathbf{a} \cdot \nabla u - \nabla \cdot (\nu \nabla u) + \sigma u = s(\mathbf{x}, t, u) \tag{1}$$

where:
u - unknown property
$\mathbf{a}$ - convection/transport velocity
ν - diffusivity
σ - reaction coefficient
s - source term

For the purpose of the present work, the case of steady 1-D convection-diffusion phenomena with no reaction term is considered. Thus, equation (1) can reduced to:

$$\mathbf{a} \cdot \nabla u - \nabla \cdot (\nu \nabla u) = s \tag{2}$$

Moreover, the following boundary conditions are proposed:

$$\begin{array}{c|c|c|} Problem[1] & Problem[2] & Problem[3] \\ u(0) = 0 & u(0) = 0 & u(0) = 0 \\ u(1) = 1 & u(1) = 0 & u(1) = 1 \\ s = 0 & s = 1 & s = sin(\pi x) \end{array} \tag{3}$$

2 Weak Galerkin formulation and discretization

The integral form equivalent to the governing equation (1) is obtained by multiplying by an arbitrary weighting function w and integrating over the domain of each equation:

$$\int_\Omega w(\mathbf{a} \cdot \nabla u) d\Omega - \int_\Omega w(\nabla \cdot (\nu \nabla u)) d\Omega = \int_\Omega w s d\Omega$$

Integrating by parts,

$$\int_\Omega w(\mathbf{a} \cdot \nabla u) d\Omega + \int_\Omega \nabla w \cdot (\nu \nabla u) d\Omega = \int_\Omega w s d\Omega$$

To solve the problem a structured mesh can be used with nodes given by $x_j = jh$ for $j = 0, 1, ...n$ and $h = 1/n$. Then, function u can be numerically computed using a piece-wise polynomial approximation given by the following expression:

$$u \approx u^h = \sum_{i=1}^{n+1} N_i u_i \tag{4}$$

Thereby, u_i is the respective value of the function at a grid node with the number i and N_i is the shape function, which must have a value of 1 on the grid node i and values of 0 on all other grid nodes. Depending on the degree of the shape function and using an isoparametric formulation with $\xi \in [-1, 1]$, the following are the shape functions of the elements to be used:

Linear	*Quadratic*
$N_1 = 1/2(1 - \xi)$	$N_1 = 1/2(\xi - 1)\xi$
$N_2 = 1/2(1 + \xi)$	$N_2 = 1 - \xi^2$
	$N_3 = 1/2(1 + \xi)\xi$

Imposing in weak form that $w = N_i$, the Galerkin weak formulation is obtained. Then, substituting equation (4) into the weak form, it reduces to a linear system of equations of the form:

$$(\mathbf{K_c} + \mathbf{K_d})\mathbf{u} = \mathbf{f} \tag{5}$$

Element K_{ij} of the matrices and the force vector f_i are defined as follows:

$$K_{cij} = \int_\Omega N_i(\mathbf{a} \cdot \nabla N_j)d\Omega$$
$$K_{dij} = \int_\Omega \nabla N_i \cdot (\nu \nabla N_j)d\Omega$$
$$f_i = \int_\Omega N_i s d\Omega$$

3 Stabilization

It has been shown that Galerkin method lacks enough diffusion and the numerical solution shows oscillations if the Péclet number $Pe = ah/2\nu$ is high (for the 1D convection-diffusion equation, if $Pe > 1$). In order to avoid this problem, several stabilization techniques have been introduced over the years.

The Streamline upwind (SU) technique introduces artificial diffusion $\bar{\nu}$ as follows:

$$\int_\Omega w(\mathbf{a} \cdot \nabla u)d\Omega + \int_\Omega \nabla w \cdot ((\nu + \bar{\nu})\nabla u)d\Omega = \int_\Omega wsd\Omega$$

The latter equation can be then rewritten as:

$$\int_\Omega w(\mathbf{a} \cdot \nabla u)d\Omega + \int_\Omega \nabla w \cdot (\nu \nabla u)d\Omega + \int_\Omega \tau(\mathbf{a} \cdot \nabla w)(\mathbf{a} \cdot \nabla u)d\Omega = \int_\Omega wsd\Omega \tag{6}$$

where $\tau = \frac{\bar{\nu}}{|\mathbf{a}\|^2}$ is called the stabilization parameter. For a 1D convection-diffusion case, the values

$$\bar{\nu} = \beta \frac{ah}{2}$$
$$\beta = coth(Pe) - \frac{1}{Pe}$$

yield nodally exact solutions. These values can be obtained imposing the FEM solution from the SU formulation to coincide with the exact solution at the nodes.

Nevertheless, SU formulation is non-consistent and, for non-constant source terms, nodally exact solutions are not obtained. As a result, an entire family of consistent stabilization formulation has been proposed using the original governing equation written as $\mathcal{R}(u) = 0$:

$$\mathcal{R}(u) = \mathbf{a} \cdot \nabla u - \nabla \cdot (\nu \nabla u) + \sigma u - s = 0$$

Thus, a general stabilized consistent formulation has the following form:

$$\int_\Omega w(\mathbf{a} \cdot \nabla u) d\Omega + \int_\Omega \nabla w \cdot (\nu \nabla u) d\Omega + \int_\Omega w \sigma u d\Omega + \overbrace{\sum_e \int_{\Omega_e} \mathcal{P}(w) \tau \mathcal{R}(u) d\Omega}^{\text{stabilization term}} = \int_\Omega w s d\Omega$$

where τ is called the stabilization parameter and $\mathcal{P}(w)$ determines the method. The Streamline-Upwind Petrov-Galerkin (SUPG) formulation has:

$$\mathcal{P}(w) = \mathbf{a} \cdot \nabla w$$

whereas the Galerkin least-squares (GLS) takes:

$$\mathcal{P}(w) = \mathbf{a} \cdot \nabla w - \nabla \cdot (\nu \nabla w) + \sigma w$$

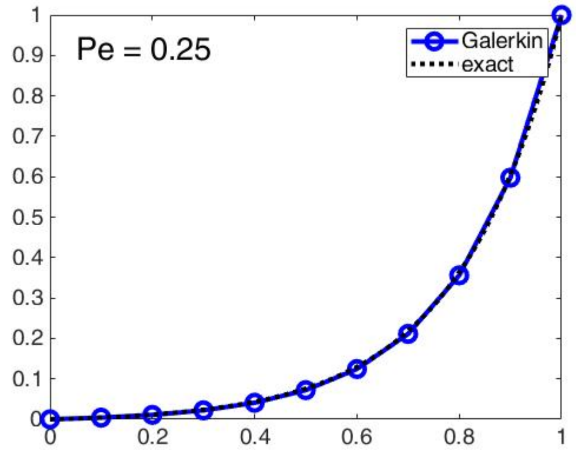
It is necessary to mention that quadratic elements and, in general, high-order finite elements present serious difficulties in convection problems because of the different behavior shown of interior and corner nodes. Therefore, in order to use quadratic elements in SPUG and GLS formulations, the stabilization parameter τ must be substituted by a diagonal matrix T which has the stabilization parameters τ and τ_c for interior and corner nodes, respectively.

4 MATLAB implementation and discussion of results

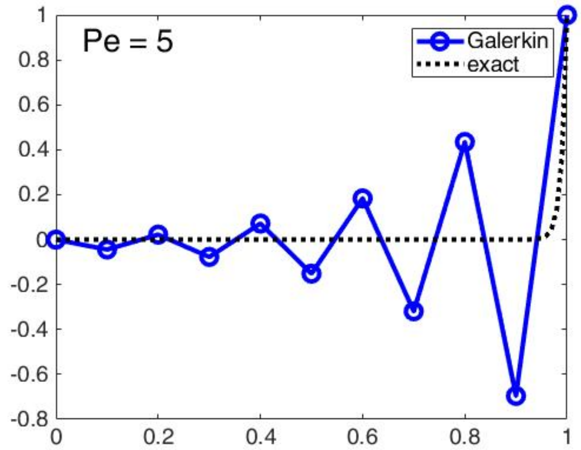
The initial code given had fully operational implementations of linear Galerkin and SU formulations to solve problems [1], [2] and [3] (see equation (3)). Firstly, both SPUG and GLS codes were written and added to the main program. Subsequently, the MATLAB implementation was modified to include quadratic elements and the possibility to solve problem [3] with a new source term defined as $s = 10e^{-5x} - 4e^{-x}$. As mentioned before, for quadratic elements different treatments were used for mid-side and corner nodes by introducing a matricial definition of the stabilization parameter.

Figure (1) shows the solutions obtained using a linear Galerkin formulation for different Péclet numbers (varying both the physical properties a and ν and the number of elements). The results matched the behavior expected presenting with instabilities for $Pe > 1$.

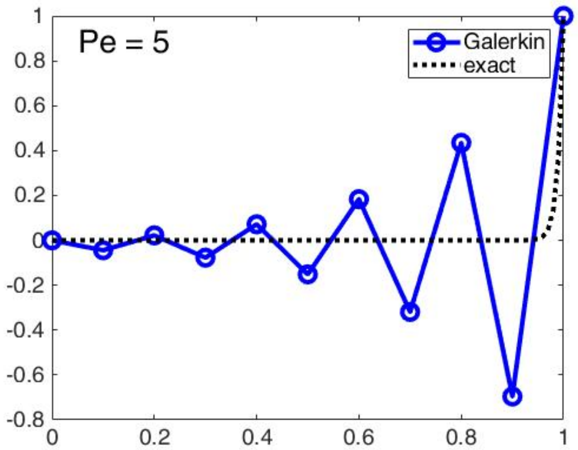
Figure (2) and (3) depict the solution curves obtained for Problem [3] with $s = sin(\pi x)$ and $s = 10e^{-5x} - 4e^{-x}$, respectively. In both cases, the solution is computed using a 10-linear-element Galerkin formulation with different stabilization techniques. As predicted by theory, SU formulation provides a solution with no oscillations for $Pe > 1$, but the solution is not consistent and a nodally exact solution is not obtained. On the contrary, SUPG and GLS give consistent numerical solution with no oscillations for the same Péclet number. Additionally, as evident in both figures for linear elements SUPG and GLS are mathematically identical.



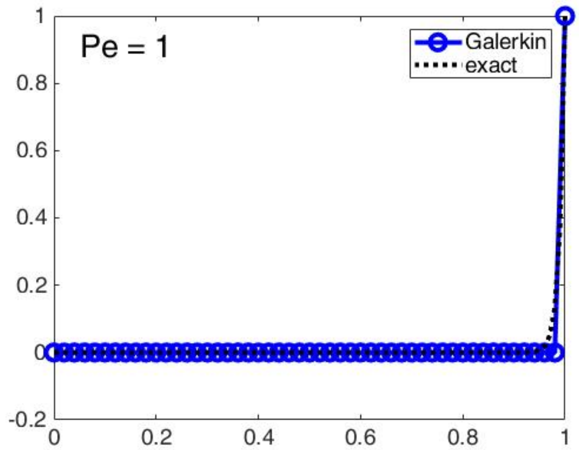
(a) Solution with $a = 1$, $\nu = 0.2$ and 10 linear elements



(b) Solution with $a = 20$, $\nu = 0.2$ and 10 linear elements

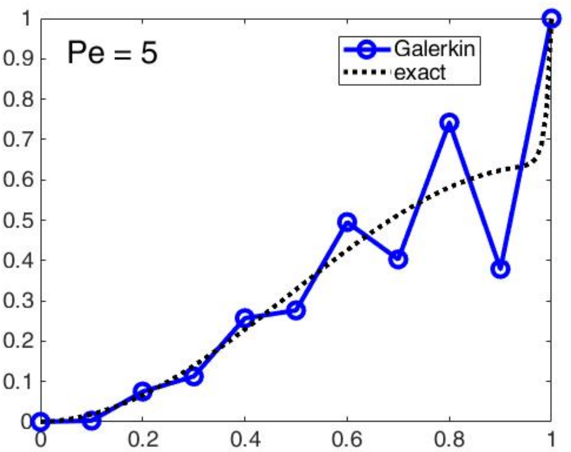


(c) Solution with $a = 1$, $\nu = 0.01$ and 10 linear elements

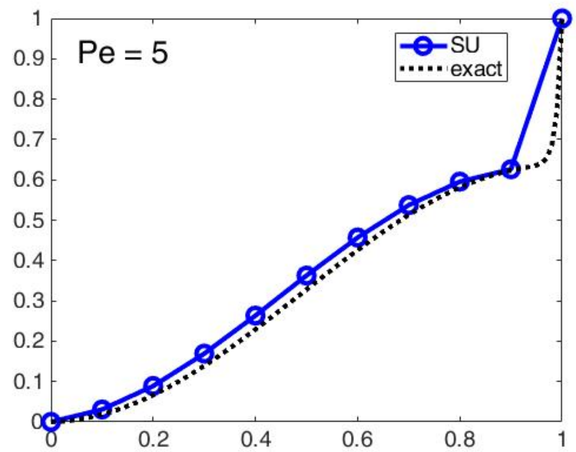


(d) Solution with $a = 1$, $\nu = 0.01$ and 50 linear elements

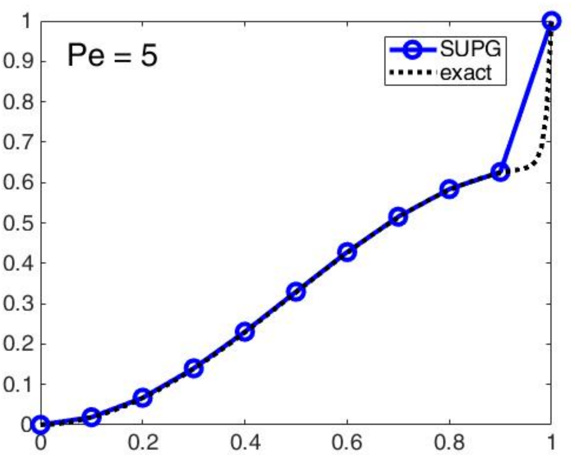
Figure 1 – Solution of problem [1] for different Péclet number using a linear Galerkin formulation



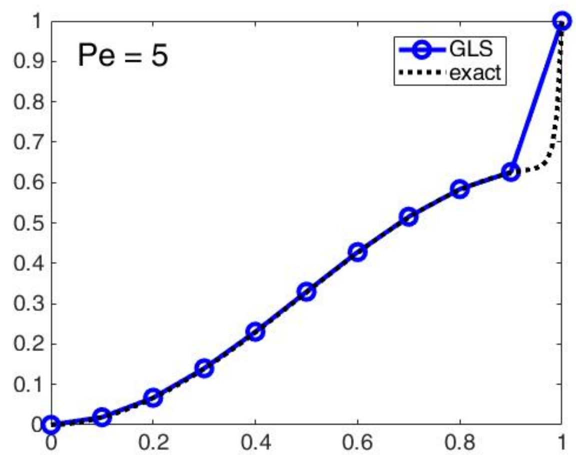
(a) Result using Galerkin formulation



(b) Result using SU formulation

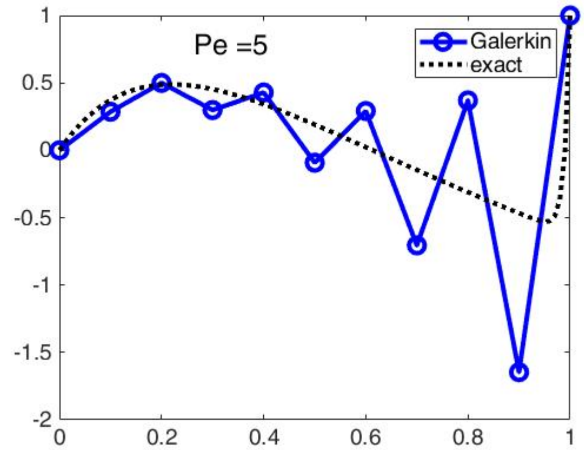


(c) Result using SUPG formulation

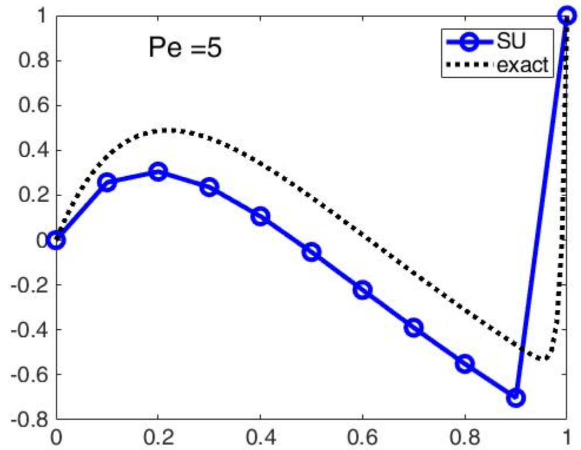


(d) Result using GLS formulation

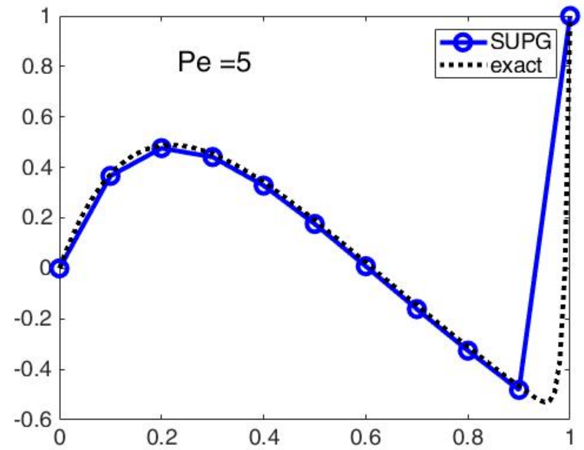
Figure 2 – Solution with source $s = sin(\pi x)$ and 10 linear elements using different stabilization formulations



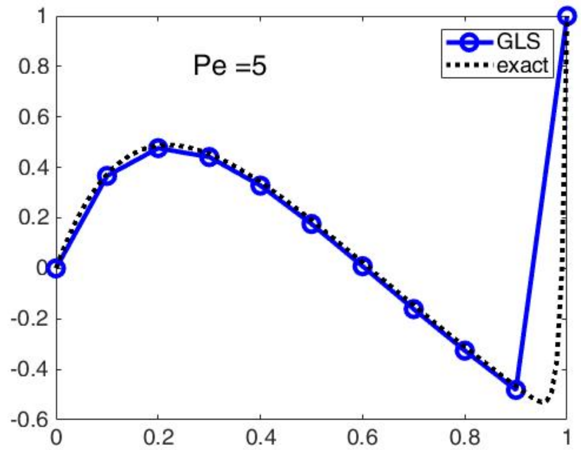
(a) Result using Galerkin formulation



(b) Result using SU formulation

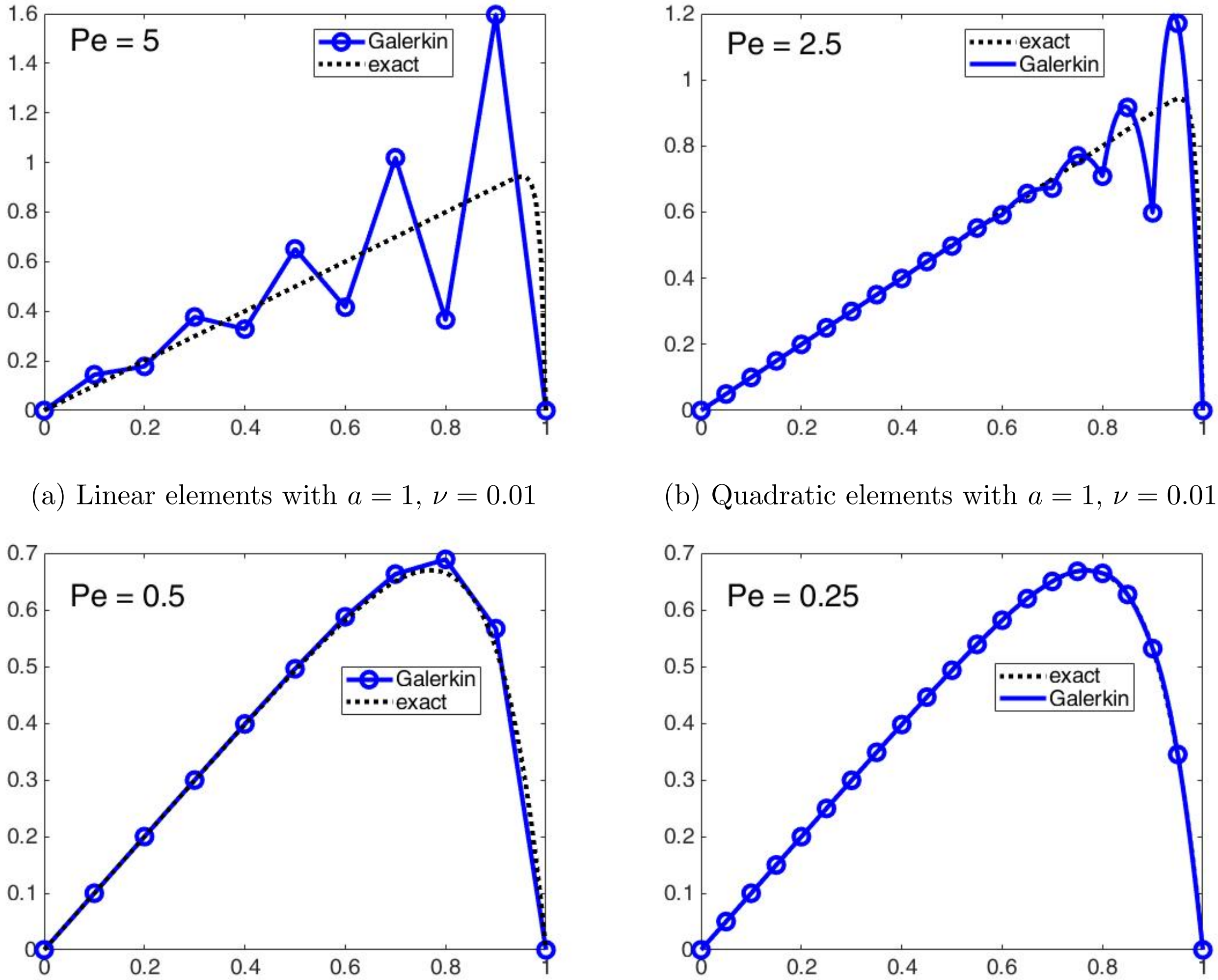


(c) Result using SUPG formulation



(d) Result using GLS formulation

Figure 3 – Solution with source $s = 10e^{-5x} - 4e^{-x}$ and 10 linear elements using different stabilization formulations



(a) Linear elements with $a = 1$, $\nu = 0.01$

(b) Quadratic elements with $a = 1$, $\nu = 0.01$

(c) Linear elements with $a = 1$, $\nu = 0.1$

(d) Quadratic elements with $a = 1$, $\nu = 0.1$

Figure 4 – Solution of problem [2] with 10 elements using Galerkin formulation

Figure (4) shows the results for different Péclet numbers using Galerkin formulation with linear and quadratic elements. The obtained results match the behavior predicted by the theory: for $Pe > 1$ the solutions with Galerkin implementation present with oscillations regardless of the degree of the shape function. It is worth mentioning that in the case of quadratic elements, the actual Péclet number is halved using the same number of elements and resulting in fewer oscillations in comparison with linear elements.

Although GLS and SUPG techniques have different formulations in the case of quadratic elements, the numerical difference can be subtle and difficult to spot (see Figure (5)). Nevertheless, it is important to mention that GLS can be more effective in the case of convection-diffusion problems in higher-dimensions, because of the symmetric nature of the term added in the GLS technique, which affects the time needed to solve the linear system of equations. The time difference can become more significant for 1-D problems if the number of nodes is increased, for instance, if 2,000

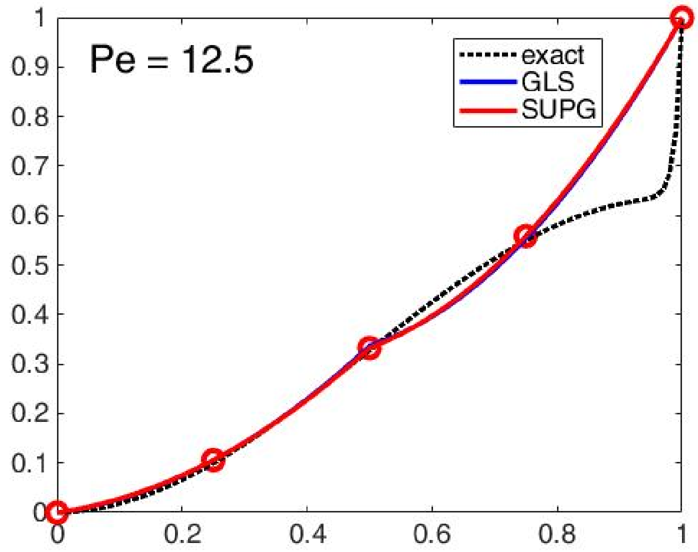


Figure 5 – Comparison of GLS and SUPG solutions for Problem [3] with 2 quadratic elements

quadratic elements are used, the computer[1] needs around 1.34 seconds to find a solution using SUPG whereas 0.85 seconds are needed if GLS is selected. Thus, for larger problems (including 2-D and 3-D) this difference in efficiency can be an important aspect to be considered.

[1]CPU Intel Core i-7 2.80 GHz - 4 Cores, 16-GB RAM