Derivation:

* Given the steady state convection-diffusion eq.:

$a \cdot \nabla u - \nabla \cdot (\nu \nabla u) = s \quad \text{in } \Omega$

with dirichlet boundary conditions

$u = u_d \quad \text{on } \partial\Omega$

* Multiplying by a test function w and integrating

$$\int_\Omega w(a \cdot \nabla u) - w[\nabla \cdot (\nu \nabla u)]\, d\Omega = \int_\Omega w s \, d\Omega$$

* Integrating by parts the terms with the laplacian op. yields the weak form:

no neumann BC

$$\int_\Omega w(a \cdot \nabla u)\, d\Omega + \int_\Omega \nabla w (\nu \nabla u)\, d\Omega - \int_{\partial\Omega} \frac{\partial u}{\partial n} w \, d\Gamma = \int_\Omega w s \, d\Omega$$

* Applying the Galerkin formulation where the test function w is the same as the shape function N_i and discretizing the solution where: $\hat{u} = \sum_i N_i u_i$

This yields the following in the case of one element:

$$\underbrace{\left[\int_\Omega N_j (a \cdot \nabla N_i)\, d\Omega + \int_\Omega \nabla N_j \cdot (\nu \nabla N_i)\, d\Omega\right]}_{K} u = \int_\Omega N_j \, s \, d\Omega$$

This yield the following system of eq.s:

$K u = f$

* In order to ensure stabilization artificial diffusion must be added. The stiffness matrix in this case becomes:

$$K = \int_\Omega N_j (a \cdot \nabla N_i)\, d\Omega + \int_\Omega \nabla N_j \cdot (\nu \nabla N_i)\, d\Omega + \int_{\Omega^e} P(N_j)\, \tau \, R(N_i)\, d\Omega$$

Where:

$R(N_i) = a \cdot \nabla N_i - \nabla \cdot (\nu \nabla N_i) - s$

using the SUPG method

$P(N_j) = a \cdot \nabla N_j$

using the GLS method

$P(N_j) = a \cdot \nabla N_j - \nabla \cdot (\nu \nabla N_j)$

In this assignment it is required to modify a given Matlab code in order to achieve certain functionalities. These functionalities would enable the code to:

1. Handle the source term
2. Use quadratic elements
3. Incorporate SUPG and GLS stabilization methods

The given code is able to implement the Galerkin and the Streamline Upwind methods (SU). The results are shown in the following figures respectively.

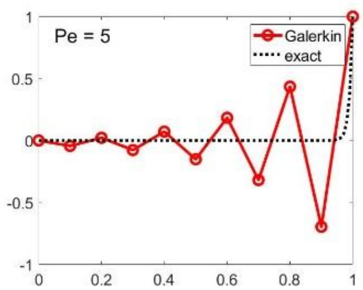

Figure 1 Glalerkin method for no source term

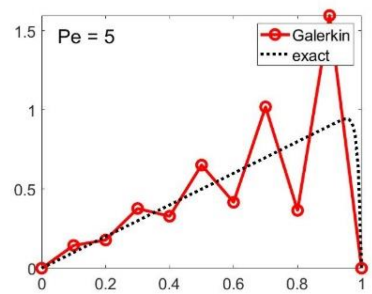

Figure 2 Glalerkin method for f = 1

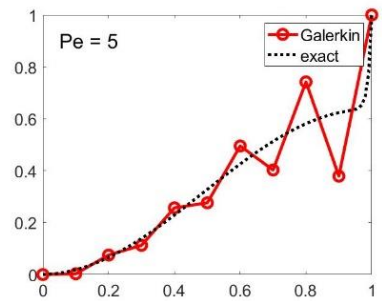

Figure 3 Glalerkin method for f = sin (pi x)

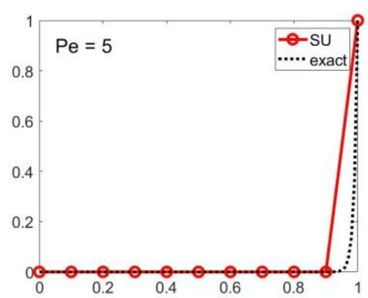

Figure 4 SU method for no source term

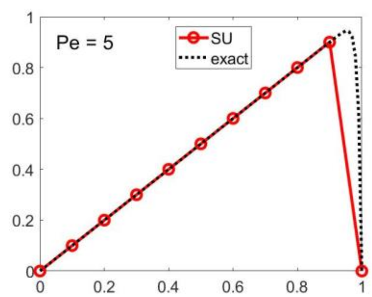

Figure 5 SU method for f = 1

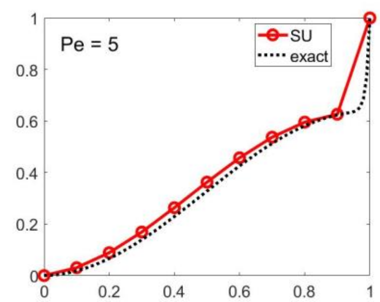

Figure 6 SU method for f = sin (pi x)

It could be observed that the Galerkin method is unstable for all cases. The SU method is more stable; however, the nodal solution is not correct where the source term is modified to a non-constant term.

Thus, the SUPG and the GLS methods are to be used as they overcome some of the short comings of the previous methods. The solution of the aforementioned methods, using linear elements, are shown in the following figures respectively.

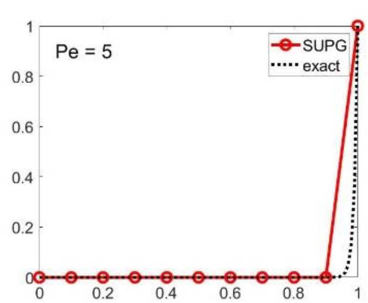

Figure 7 SUPG method for no source term

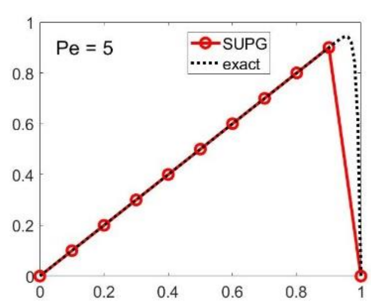

Figure 8 SUPG method for f = 1

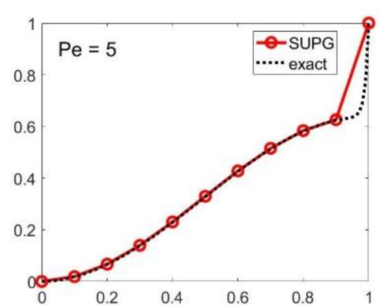

Figure 9 SUPG method for f = sin (pi x)

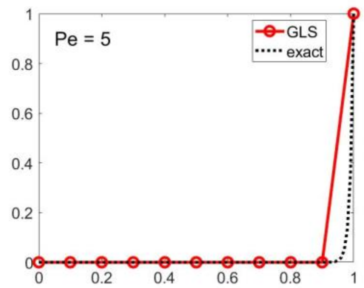

Figure 10 GLS method for no source term

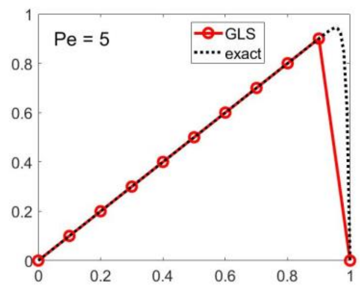

Figure 11 GLS method for f = 1

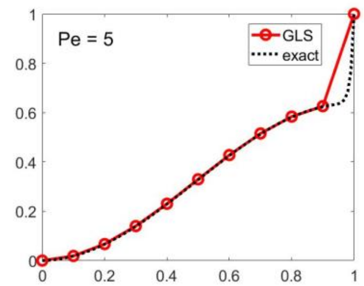

Figure 12 GLS method for f = sin (pi x)

It could be seen that both methods achieve the exact solution at the nodes. Moreover, both methods are identical which was expected when using linear elements. However, the linear interpolation between the nodes does not match with the analytical solution hence quadratic elements would be more suited to capture the entire behavior of the solution.

The following modifications to the code where implemented to incorporate quadratic elements. First, the Laplacian term in the added diffusion term could no longer be neglect, hence the second derivative of the shape functions is to be used. The jacobian is to be applied to the second derivative of the shape functions is required to perform the gauss legendre quadrature. The connectivity matrix is to be modified in order to accommodate the extra node of the quadratic element compared to the linear element. The stability parameter must be also modified in order to be compatible with the quadratic element. Thus, two different parameters exist: one for the corner nodes and one for the mi-side nodes. The following are the equations to be used:

$$\tau_{\mathrm{m}} = \beta\, h/(2a) \qquad\qquad \tau_{\mathrm{c}} = \beta_{\mathrm{c}}\, h/(2a)$$

Where:

$$\beta = \coth Pe - 1/Pe \qquad\qquad \beta_{\text{corner}} = \frac{(\coth Pe - 1/Pe) - (\cosh Pe)^2(\coth 2Pe - 1/(2Pe))}{1 - (\cosh Pe)^2/2}$$

Finally, the postprocessing had to be modified in order to be able to plot curves between the nodes instead of connecting them with lines. Splines where used in this case; however, ploting the shape functions themselves would yield a more accurate results.

The follwing are the results obtained using quadrating elements applying the SUPG and the GLS methods respectivelly.

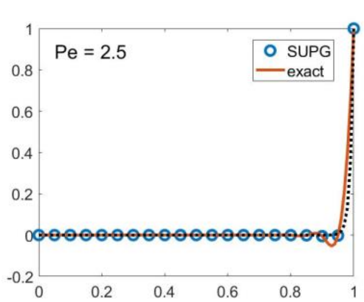

Figure 13 Quad SUPG method for no source term

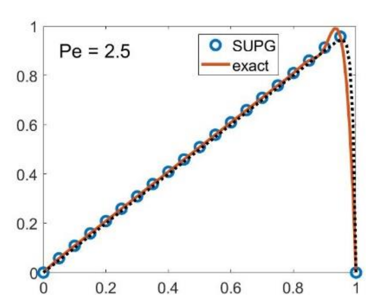

Figure 14 Quad SUPG method for f = 1

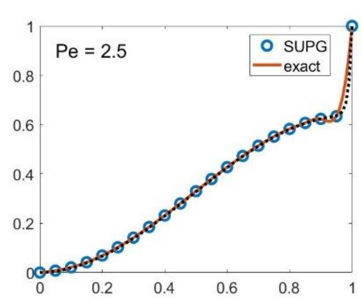

Figure 15 Quad SUPG method for f = sin (pi x)

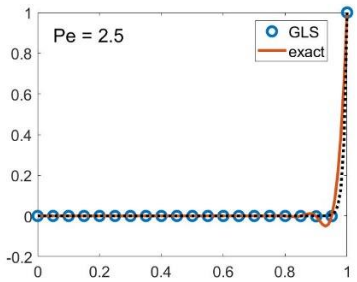

Figure 16 Quad GLS method for no source term

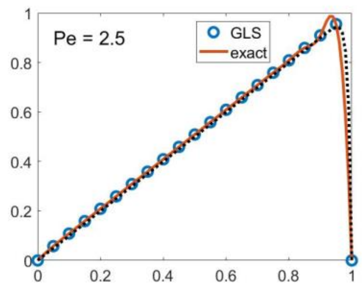

Figure 17 Quad GLS method for f = 1

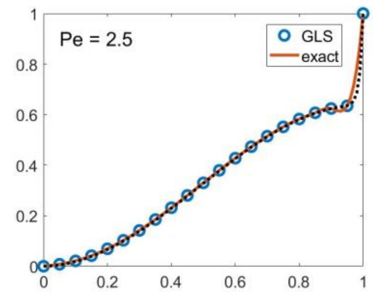

Figure 18 Quad GLS method for f = sin (pi x)

It could be seen that quadratic elements achieve exact solution at the nodes; while the behavior of the solution aside from the nodes is almost identical to the exact solution.