

Numerical Stabilization of Convection-Diffusion-Reaction Problems

Ruperto P. Bonet Chaple
Delft Institute of Applied Mathematics
Faculty of Electric Engineering,
Mathematics and Computer Science
Mekelweg 4
2628 CD Delft, Netherlands
e-mail: ruperto.bonet@upc.edu
Web page: //www-ma1.upc.es

15th March 2006

1 Abstract

In this work several approximations of Petrov-Galerkin methods are considered in the framework of Convection-Diffusion-Reaction problems. Such methods have improved the Galerkin method in relation to the stabilization of the resulting numerical schemes. We discuss the behaviour of StreamLine Upwind Diffusion method (SUPG), Galerkin Least Square method (GLS) and Subgrid Scale method (SGS) by means of several tests.

2 Introduction

In many applications, it is important to design numerical methods guaranteeing that the discrete it solution satisfies the physical conditions, in particular in convection-diffusion problems very often happens that the solution have a convective nature on most of the domain of the problem, and the diffusive part of the differential operator is influenced only in certain small subdomains. In these subdomains the gradient of the solution is large and then, these solutions have a boundary layer behavior. The fact is that the numerical methods designed for elliptic problems will not work satisfactorily. In practice they usually exhibit a certain degree of instability. The goal then is to modify these numerical methods in stable form without loss accuracy.

The physical behaviour of the advection-diffusion-reaction equation is very rich. Mainly, two regimes are present: the exponential regime and the propagation regime. In the exponential regime, the roots of the characteristic polynomial are real and thus, solutions are characterized by an exponential growth or decay. In the propagation regime, the characteristic polynomial has complex conjugates roots and thus, the solutions are sinusoidal functions modulated by an exponential function.

A common source of convection-diffusion problems are linearizations of Navier-Stokes equations with large Reynolds number. Morton (1966) points out that this is by no means the only place where they arise: in this opening chapter he lists ten examples involving convection-diffusion equations that include the drift-diffusion equations of semiconductor device modelling and the Black-Scholes equation from financial modelling. He also observes that “Accurate modelling of the interaction between convective and diffusive processes is the most ubiquitous and challenging task in the numerical approximation of partial differential equations.”[13]

The numerical solution of convection-diffusion problems goes back to the 1950s (Allen and Southwell 1955), but only in the 1970s did it acquire a research momentum that has continued to this day.[13]

The numerical solution of convection-diffusion-reaction equation using Galerkin formulation normally exhibits global spurious oscillations in convection-dominated problems, especially in the vicinity of sharp gradients. Over, the last two decades, a variety of finite elements approaches have been proposed to

deal with such situations. These methods add a perturbation term to the weighing functions with the aim to get an oscillation-free solution. These terms are mesh-dependent and allow to get a consistent and stabilizing numerical scheme. Such methods have grown in popularity, especially in application to fluid dynamics. Starting with Brooks and Hughes [1] with the Streamline Upwind Petrov-Galerkin method (SUPG), a generalization was proposed for the Stokes problem [16] that circumvents the need to satisfy the Babuska-Brezzi condition [8],[3], which showed the potential of extending the idea to various applications. Designing the perturbation term as in Least-Squares formulation, the Galerkin/Least-square method (GLS) was applied to various structural and fluid problems [15]. Harari and Hughes [7] presented a Galerkin/least-squares (GLS) and gradient Galerkin/least-squares (GGLS) methods to optimize the performance of finite element formulations for the advection-diffusion equation with production. While the GGLS method is nodally exact for one-dimensional solutions, the GLS method is not. In the latter case, several choices of parameters are derived, which optimize the solutions according to certain criteria. Many effort have been done to keep a numerical method that works very well in all range of parameters. Another strategy have been to devise an augmented problem, where the small scales can be solved and the local instabilities can be removed. A number of stabilized formulations have been proposed based on the multiscale methods [14] and related work on the residual free bubbles(RSFB) by Russo [2] and Brezzi [5]. Further attempts to devised a stabilized finite element method with good stability in the presence of reactive terms are presented with the “unusual stabilized method” (USFEM) by Franca et al. [10],[11] and by Codina [12]. However, their analysis considers only negative source terms, ignoring production terms. Some advancement in this direction has been done by G.Hauke with the presentation of a subgrid scale method with a simple intrinsic time-scale parameter [6]. This formulation combines very good accuracy and stability in all physical regimes of the equation. Each formulation is cast in weighed residual form through the addition of an integral over the interior of each element.

In this work the Galerkin, SUPG, GLS and SGS method are applied to solve the advection-diffusion-reaction equation and compared with the exact solutions for the whole range of parameters and conclusions are drawn. The layout of this report is as follows. First, the steady state convection-diffusion-reaction equation is presented. Second, a brief explanation relative to each method is given. Third, numerical results are presented, and finally, a brief explanation of the MATLAB code is presented.

3 The advection-diffusion-reaction equation

The advective-diffusive-reaction problem consists of finding the scalar-valued function $\phi(\mathbf{x})$ in open domain Ω with boundary $\Gamma = \partial\Omega$, and Dirichlet boundary conditions, such that

$$-\nabla \cdot (k \nabla \phi) + \mathbf{u} \cdot \nabla \phi - s\phi = f(\mathbf{x}) \quad \text{in } \Omega, \quad (1)$$

$$\phi(\mathbf{x}) = g(\mathbf{x}) \quad \text{on } \Gamma, \quad (2)$$

where $k \geq 0$ is the diffusion coefficient, \mathbf{u} is the velocity field, $\mathcal{L}_{adv}(\phi) = \mathbf{u} \cdot \nabla \phi$ is the advective operator, f is a source function and s is the source parameter, where $s > 0$ for production and $s < 0$ for dissipation. The domain Ω is assumed to be smooth. The equation (1) includes the following particular cases:

- Advection-Diffusion equation ($s = 0$)
- Helmholtz equation ($\mathbf{u} = 0, s > 0$)
- Advection-reaction equation ($k=0$)
- Diffusion-reaction equation ($\mathbf{u} = 0$)

which describe many situations related with transport phenomena, such as mass transport and energy transport, that can be found in several problems in engineering practice. The solutions of equation (1) can be classified according to the sign of the discriminant $D = |\mathbf{u}|^2 - 4sk$. if $D \geq 0$, we are in the exponential regime (being dissipation with $s < 0$, and production with $s > 0$), and the remain cases, that is, if $D < 0$, we are in the propagation regime.

Equation (1) can be written in operational form by means of the advection-diffusion-reaction operator \mathcal{L} and its adjoint operator \mathcal{L}^* . Then,

$$\mathcal{L} = \mathbf{u} \cdot \nabla - \nabla \cdot (k \nabla) - s \quad (3)$$

$$-\mathcal{L}^* = \mathbf{u} \cdot \nabla + \nabla \cdot (k \nabla) + s \quad (4)$$

Fundamental dimensionless numbers that characterize the solution are

$$Pe = UL/k, \quad Peclet \quad (5)$$

$$Da = SL/U, \quad Damkohler \quad (6)$$

where U,S and L are the characteristic velocity, source and length of the problem. At the same time, the numerical solution is characterized by the dimensionless numbers

$$Pe = |\mathbf{u}|h/2k, \quad element \ Peclet \quad (7)$$

$$\sigma = sh/|\mathbf{u}|, \quad element \ Damk\ddot{o}hler \quad (8)$$

where h is the element size. Multiplying each member of above equality is obtained

$$2\mathbf{P}e\sigma = \frac{sh^2}{k} \quad (9)$$

The variational formulation corresponding to (1)-(2) with $g \equiv 0$ is:

Find $\phi \in H_0^1(\Omega)$ such that

$$B(\phi, w) = F(w) \quad \forall w \in H_0^1(\Omega), \quad (10)$$

with

$$B(\phi, w) = k(\nabla\phi, \nabla w) + (\mathbf{u} \cdot \nabla\phi, w) - s(\phi, w) \quad (11)$$

$$F(w) = (f, w) \quad (12)$$

where (\cdot, \cdot) indicates the integration over Ω and $H_0^1(\Omega)$ is the Sobolev space with square integrable functions and its first order derivatives, and zero value on the boundary. We denote by $\|\cdot\|_1$ the norm associated with this space and by $\|\cdot\|$ the $L_2(\Omega)$ norm. Taking $\phi = w$ in (10) in [9](4), in case of $s = 0$ is obtained the inequality

$$\|\phi\|_1 \leq \frac{1}{kC} \|f\| \quad (13)$$

and in case $s < 0$ taking $M = \min \{k, -s\}$ leads

$$\|\phi\|_1 \leq \frac{1}{M} \|f\| \quad (14)$$

This estimate indicates that small variations on the data f for small values of k (or/and small values of s), can cause large variations of the solution ϕ , which is the intrinsic problem underlying the model equation (1).

4 Stabilized finite element methods

We consider a partition \mathcal{P} of the computational domain Ω in nel elements K of size $|K|$ and boundary Γ^e , resulting a mesh partition of two-dimensional Ω into triangles and /or quadrilaterals that do not overlap. Taking a subspace

$$V_h = \{w \in H_0^1(\Omega) / w \text{ is a continuous piecewise polynomial}\}$$

The standard Galerkin finite element method for the equation (1) can be expressed as : Find $\phi_h \in V_h$ such that

$$B(\phi_h, w) = F(w) \quad \forall w \in V_h, \quad (15)$$

This method [13] originates spurious oscillations when the convective coefficient is larger than the diffusive coefficient (when $Pe > 1$) or when the source term is larger than the diffusive coefficient (when $2 * Pe * \sigma > 1$). The stabilized finite element methods for this equation can be written as : Find $\phi_h \in V_h$ such that

$$B(\phi_h, w) + S(\phi_h, w) = F(w) \quad \forall w \in V_h, \quad (16)$$

where $S(\phi, w)$ indicates the perturbation terms added to the standard variational formulation (15). These terms guaranties that consistency is kepted and numerical stability is enhanced. There are three different terms that are usually considered for this equation, namely

$$S^{supg}(\phi, w) = \sum_K \tau_K^{supg} (\mathcal{L}_{adv} w, \mathcal{L}\phi - f)_K \quad (\text{SUPG method})[1] \quad (17)$$

$$S^{gls}(\phi, w) = \sum_K \tau_K (\mathcal{L}w, \mathcal{L}\phi - f)_K \quad (\text{GLS method})[15] \quad (18)$$

$$S^{sgs}(\phi, w) = \sum_K \tau_K (-\mathcal{L}^* w, \mathcal{L}\phi - f)_K \quad (\text{SGS method})[6] \quad (19)$$

where K is the element at the discretization, τ_K^{supg} , τ_K are the stabilizing parameter to be described below and $(\cdot, \cdot)_K$ denotes integration over K . In the numerical experiments we use the following stabilizing parameters for linear elements, according to the selected method[6].

$$\tau_K^{supg} = \frac{h}{2|\mathbf{u}|} (\coth Pe - \frac{1}{Pe}) \quad (20)$$

$$\tau_K = \frac{1}{\frac{4k}{h^2} + \frac{2|\mathbf{u}|}{h} + |s|} \quad (21)$$

$$\tau_K = \frac{1}{\frac{6k}{h^2} \max(1, \frac{2|Pe|}{3}) + |s| \max(1, \frac{1}{3|Pe\sigma|})} \quad (22)$$

where the Eq. (20) comes from advection-diffusion theory, and in practice is usual to hand the asymptotic expressions

$$\tau_K^{supg} = \begin{cases} \frac{h}{2|\mathbf{u}|}, & Pe \geq 1, \\ \frac{h^2}{12k}, & Pe < 1, \end{cases} \quad (23)$$

the Eq. (21) was obtained by Codina for source terms, ($s < 0$) [12] using the maximum principle, and the Eq. (22) is derived by Franca and Valentin ($s < 0$) [11] from the convergence and stability theory. These equations have been extended and tested by Hauke [6] for all values of s .

5 Numerical Results

In this section, are compared for the advection-diffusion-reaction equations with an exact solution the given stabilized finite element methods. This comparison is carried out by means of computed solutions for a wide range of characteristic dimensionless parameters. Linear elements are employed at the discretization in case of one-dimensional problem and bilinear elements in case of two-dimensional problems. In all tests Dirichlet boundary conditions are considered and the size of computational domain is equal to 1.

First, we consider advection-diffusion equation in one-dimensional and two-dimensional cases, where the convection coefficient is dominant.

5.1 Advection-Diffusion equation with variable source

We consider the one-dimensional problem

$$200 \frac{d\phi}{dx} - k \frac{d^2\phi}{dx^2} = x^2 \quad \text{in } \Omega = (0, 1), \quad \phi(0) = 0, \quad \phi(1) = 0$$

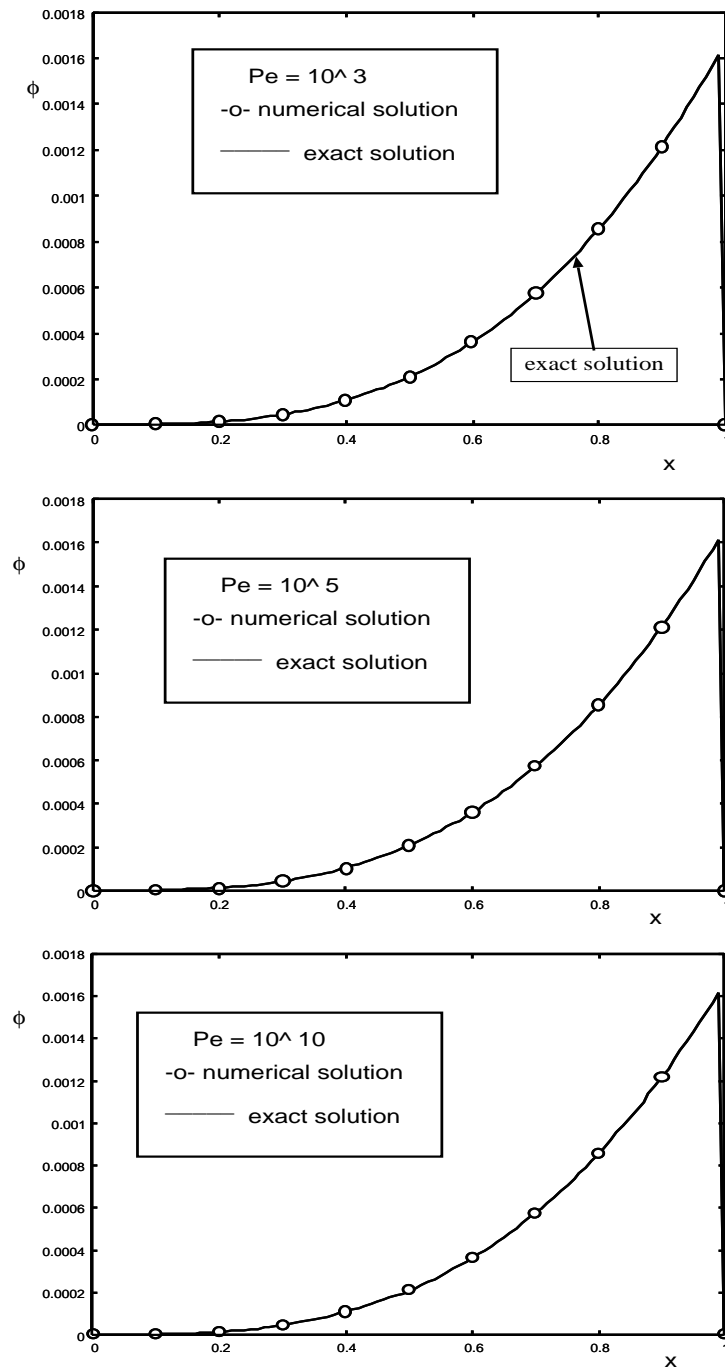


Figure 1: Solution of a one-dimensional convection-diffusion problem with $U = 200$, $Q = x^2$ and $\phi(0) = 0$, $\phi(1) = 0$. for several values of Peclet numbers. Numerical solutions obtained with 10 two node linear elements. (up) $Pe = 10^3$, (middle) $Pe = 10^5$, (down) $Pe = 10^{10}$

In one-dimensional case, we analyze the influence of the spatial variation of the source term in the behaviour of the numerical solution. We can conclude from on Figure1 that the numerical solution

obtained by SUPG method is in excellent agreement with the exact solution, independently of the values of Pe .

5.2 Steady advection-diffusion two-dimensional problem

We consider the test case

$$L\phi := -\nabla \cdot (k \nabla \phi) + \mathbf{u} \cdot \nabla \phi = 0 \quad \text{in } \Omega = (0, 1) \times (0, 1), \quad (24)$$

where $u = (1, 3)$ with the following boundary conditions

$$\phi(0, y) = 1, \quad \phi(x, 1) = \phi(1, y) = 0, \quad \phi(x, 0) = \begin{cases} 1 & x \leq 1/3, \\ 0 & x > 1/3. \end{cases} \quad (25)$$

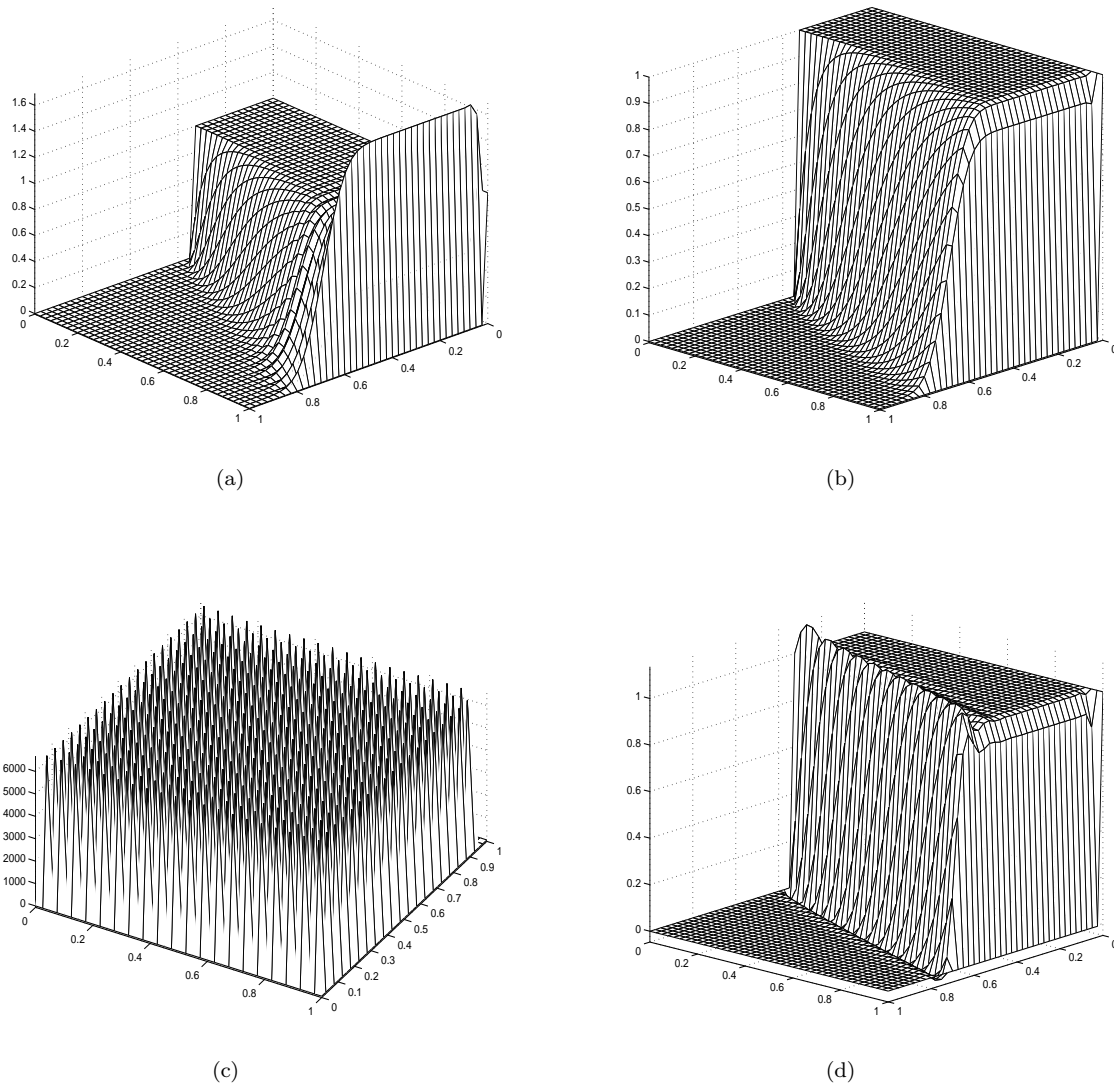


Figure 2: Solution of two-dimensional convection-diffusion problem. Advection skewed to the mesh. Numerical solutions obtained with 1600 bilinear elements. (left) Galerkin solution, (right) SUPG solution, (up) Numerical solution with $Pe = 7.9057$, (down) Numerical solution with $Pe = 7.9057e + 5$

In this case the exact solution exhibits an internal and boundary layer[4]. We compare the solution obtained by SUPG method and the solution obtained by Galerkin method. The basic mesh is made

of 1600 quadrilaterals. Figure 2 shows that the poor performance of the Galerkin method in case of convection dominant. Also, when the Peclet number is very high one observe the presence of crosswind diffusion, which is not taking into account in the stabilized methods presented above, because such methods stabilizes the solution only in the streamline direction. Other techniques relate to shock-capturing methods are necessary to implement to catch the crosswind diffusion.

5.3 One-dimensional Advection-Diffusion-Reaction Equation with constant coefficients

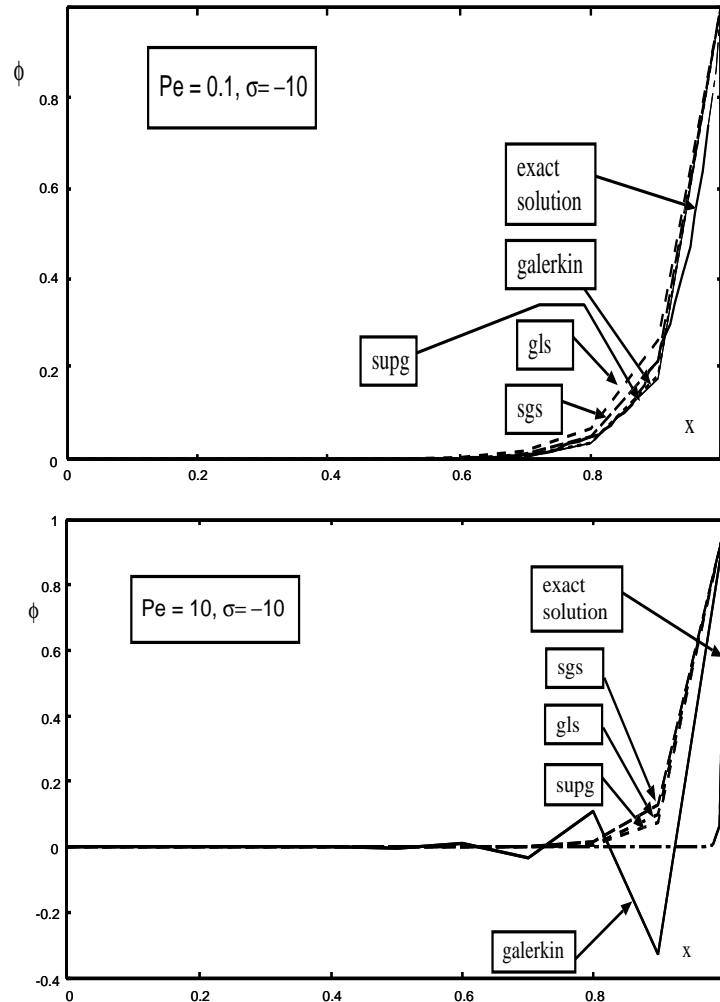


Figure 3: Solution of a one-dimensional convection-diffusion-reaction problem with Dirichlet boundary conditions $\phi(0) = 0$, $\phi(1) = 1$. for several values of Peclet and Damkhöler numbers. Numerical solutions are obtained with 10 two node linear elements.

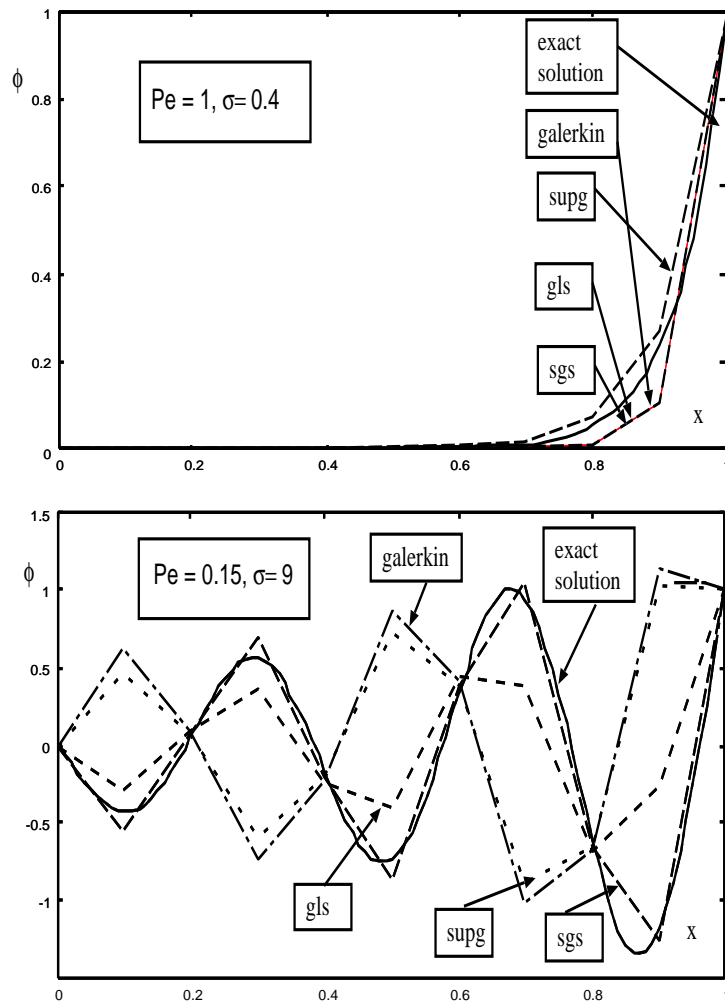


Figure 4: Solution of a one-dimensional convection-diffusion-reaction problem with Dirichlet boundary conditions $\phi(0) = 0$, $\phi(1) = 1$. for several values of Peclet and Damkhöler numbers. Numerical solutions are obtained with 10 two node linear elements.

Numerical results for this case are shown in Figures 3, 4, 5 and 6. By means of one-dimensional tests we discuss the behaviour of the numerical solution obtained by several methods, such as **SUPG**, **GLS** and a simple **SGS**. We consider the cases that including exponential and propagation regime, and in case of exponential regime are presented some case of dissipation or production.

Figure 3 shows numerical results in exponential regime with dissipation. At the first picture (up) we can observe that all above mentioned methods are closed to the analytical solution, by the contrary at the second picture (down) all above mentioned methods have a wrong near to the boundary layer. This situation appears when the diffusion and source terms are competitive. On Figure 5 (up) can be observed this situation also, using other boundary conditions. In this case all numerical solutions oscillate near to the left boundary layer, resulting in a poor precision.

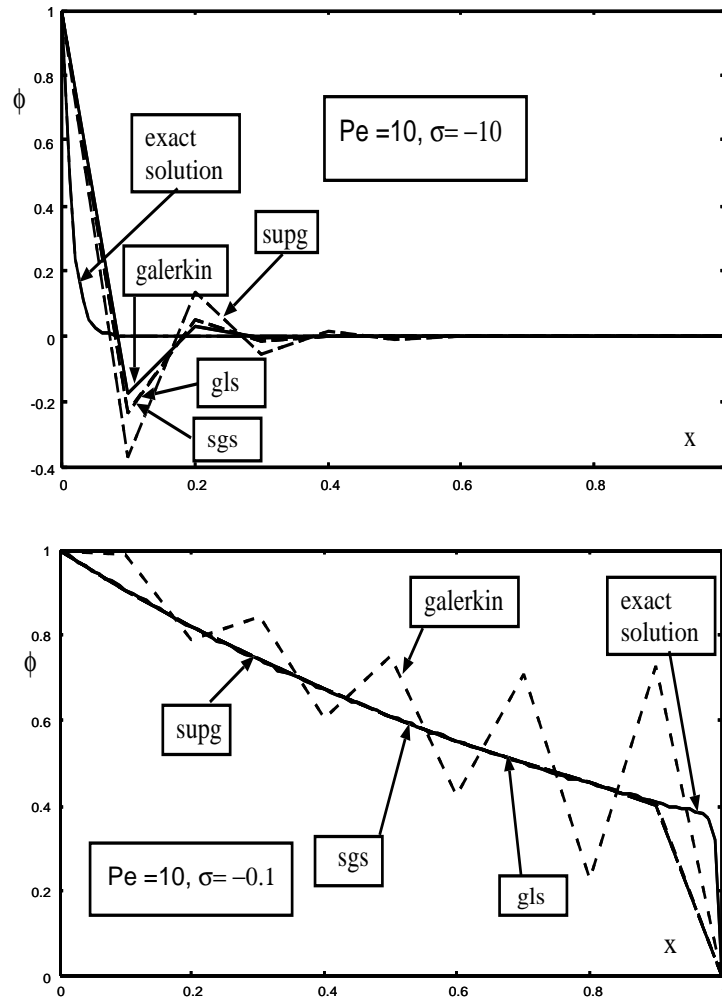


Figure 5: Solution of one-dimensional convection-diffusion-reaction problem with Dirichlet boundary conditions $\phi(0) = 1$, $\phi(1) = 0$. for several values of Peclet and Damkhöler numbers. Numerical solutions obtained with 10 two node linear elements.

Figures 5(down) and 6(up) display the numerical results in case of exponential regime, in which the diffusion mechanism is dominant relative to the dissipation/production term. This reason is the cause that the numerical solution have a good agreement except the Galerkin method. This behaviour is more noticeable on Figure 6(up), where the numerical solution by SUPG is more close to the exact solution than solution obtained by GLS and SGS methods. In case that the diffusion mechanism dominates relative to the dissipation/production mechanism in exponential regime all methods except the Galerkin method have an excellent agreement with the exact solution, as we can observe on these pictures.

Figures 4(down) and 6(down) display the numerical results in case of propagation regime ($Pe = 0.15$, $\sigma = 9$) with different boundary conditions. When the boundary condition is 1 to the left side the numerical solution by SUPG method display a curve with significant phase and amplitudes errors. Also, with the use of the GLS method leads significant errors in the amplitudes of the wave. Only a simple SGS method reports a better behaviour, as in phases as well in amplitudes. However, the numerical results are not excellent.

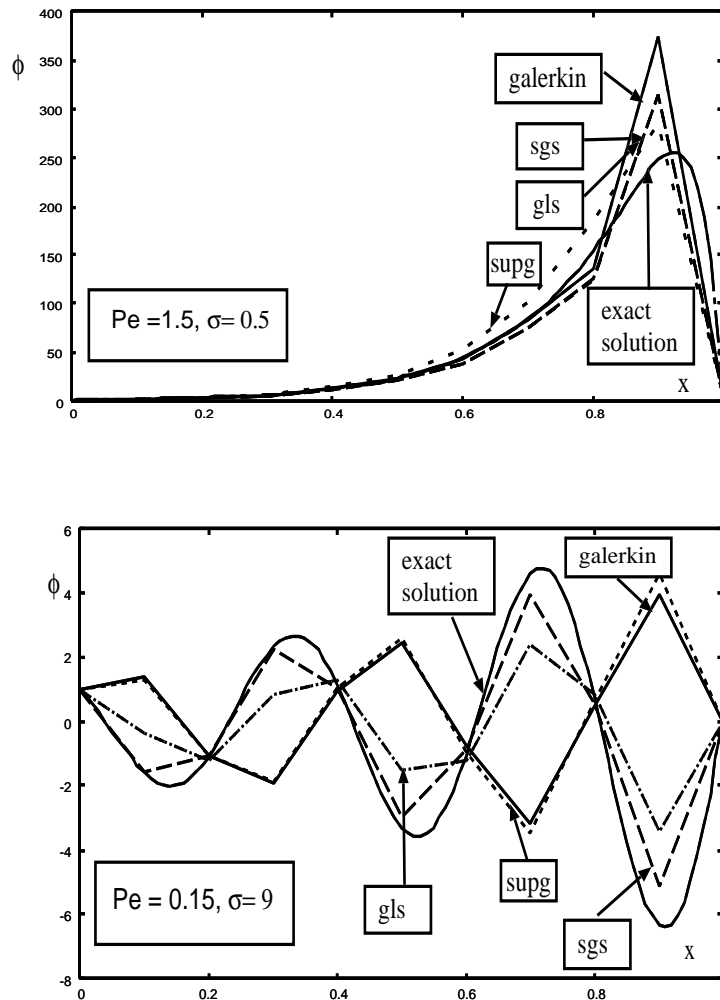


Figure 6: Solution of one-dimensional convection-diffusion-reaction problem with Dirichlet boundary conditions $\phi(0) = 1$, $\phi(1) = 0$. for several values of Peclet and Damkhöler numbers. Numerical solutions obtained with 10 two node linear elements.

We recall that the SGS method perform very well in the propagation region, in contrast the GLS method, which shows bad stability for strong production. All examined methods fail in presence of strong convection for large dissipation, as can observe the Figure 5 (up). The SUPG method has a good performance for large convection in combination with moderate source terms. Also, we can remark that the GLS method is too diffusive in comparison with the SGS method.

5.4 Steady advection-diffusion-reaction two-dimensional problem

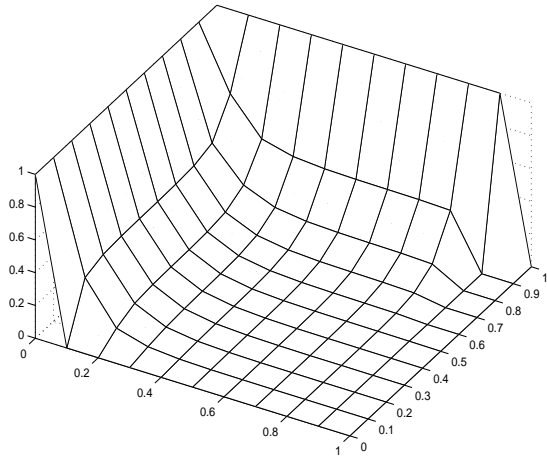
$$L\phi := -\nabla \cdot (k\nabla\phi) + \mathbf{u} \cdot \nabla\phi - s\phi = 0 \quad \text{in } \Omega = (0, 1) \times (0, 1), \quad (26)$$

where $u = (\cos(30^\circ), \sin(30^\circ))$ with the following boundary conditions

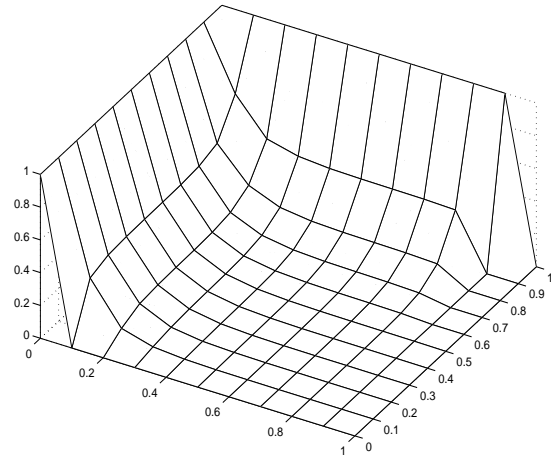
$$\phi(x, 1) = \phi(0, y) = 1, \quad \phi(x, 0) = \phi(1, y) = 0 \quad (27)$$

Numerical results in the unit square with an uniform mesh of 10×10 uniform elements are presented an compared to the Galerkin method.

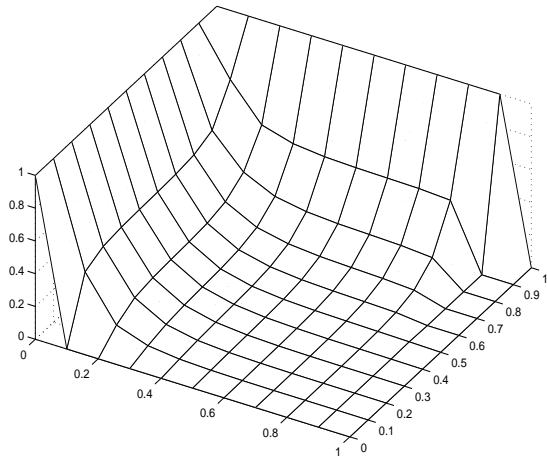
Such results are displayed for several values of Peclet number and Damkhöler number by means of Figures 7, 8,9 and 10. In all cases the advection has been considered skewed to the mesh. The Figures show a wide range of behaviour of parameters, from small convection and production to large production, or convection.



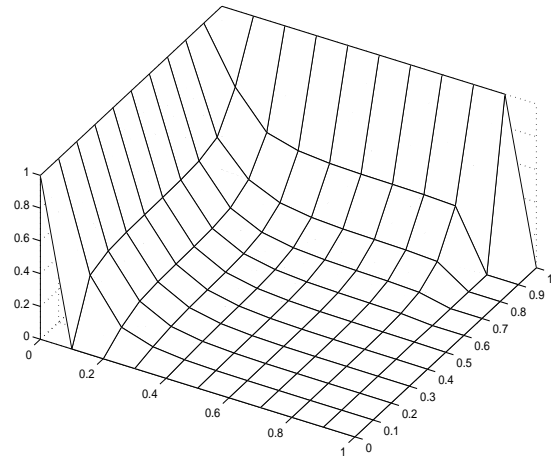
(a) Galerkin



(b) SUPG



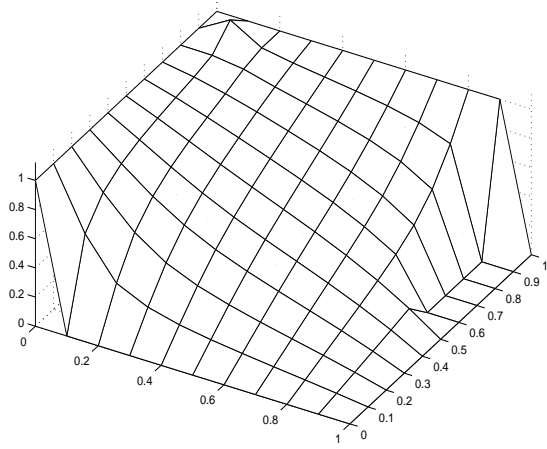
(c) GLS



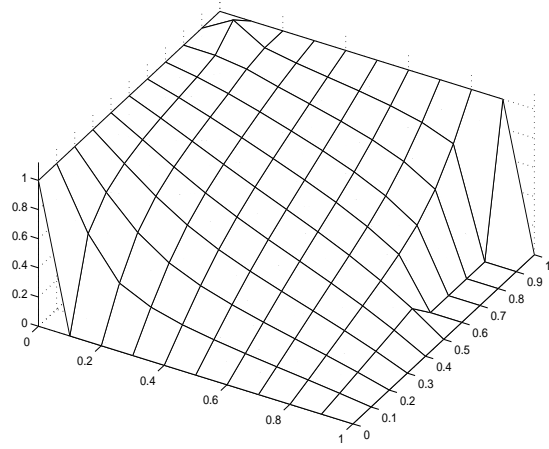
(d) SGS

Figure 7: Solution of two-dimensional convection-diffusion problem. Advection skewed to the mesh. Numerical solutions obtained with 100 bilinear elements for $Pe = 0.1, \sigma = -10$

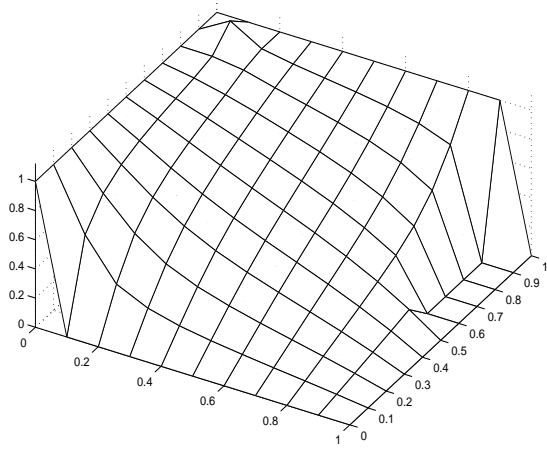
Figure 7 shows the numerical solution by different methods in case of exponential regime with dissipation. We notice that the diffusion mechanism is not important because Pe is equal to 0.1, and then, the numerical results are very close including the Galerkin method that causes good precision in case of pure advection. This behaviour has been marked in one-dimensional case, where the numerical results obtained are very closed.



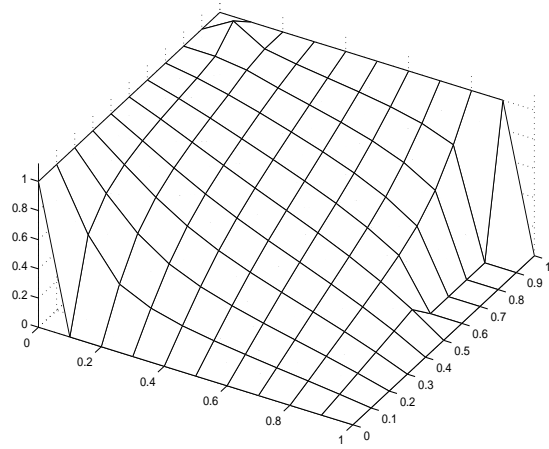
(a) Galerkin



(b) SUPG



(c) GLS



(d) SGS

Figure 8: Solution of two-dimensional convection-diffusion problem. Advection skewed to the mesh. Numerical solutions obtained with 100 bilinear elements for $Pe = 0.1, \sigma = 0.04$

Figure 8 shows the numerical solution in case of exponential regime with production. The parameters selected are $\mathbf{Pe} = 0.1, \sigma = 0.04$. The influence of source term is very small, and again the numerical results have the behaviour determined by the value of \mathbf{Pe} . Given the value of \mathbf{Pe} is equal to 0.1, like as in the above test the behaviour between all methods is the same. But, due to the production term the qualitative behaviour is very different to the dissipation case, in which the numerical solutions go down until take the value zero at the opposite boundaries.

Figure 9 shows the behaviour of the numerical solution by different methods when the mechanism of diffusion is dominant relative to dissipation mechanism. In this case there is in an exponential regime.

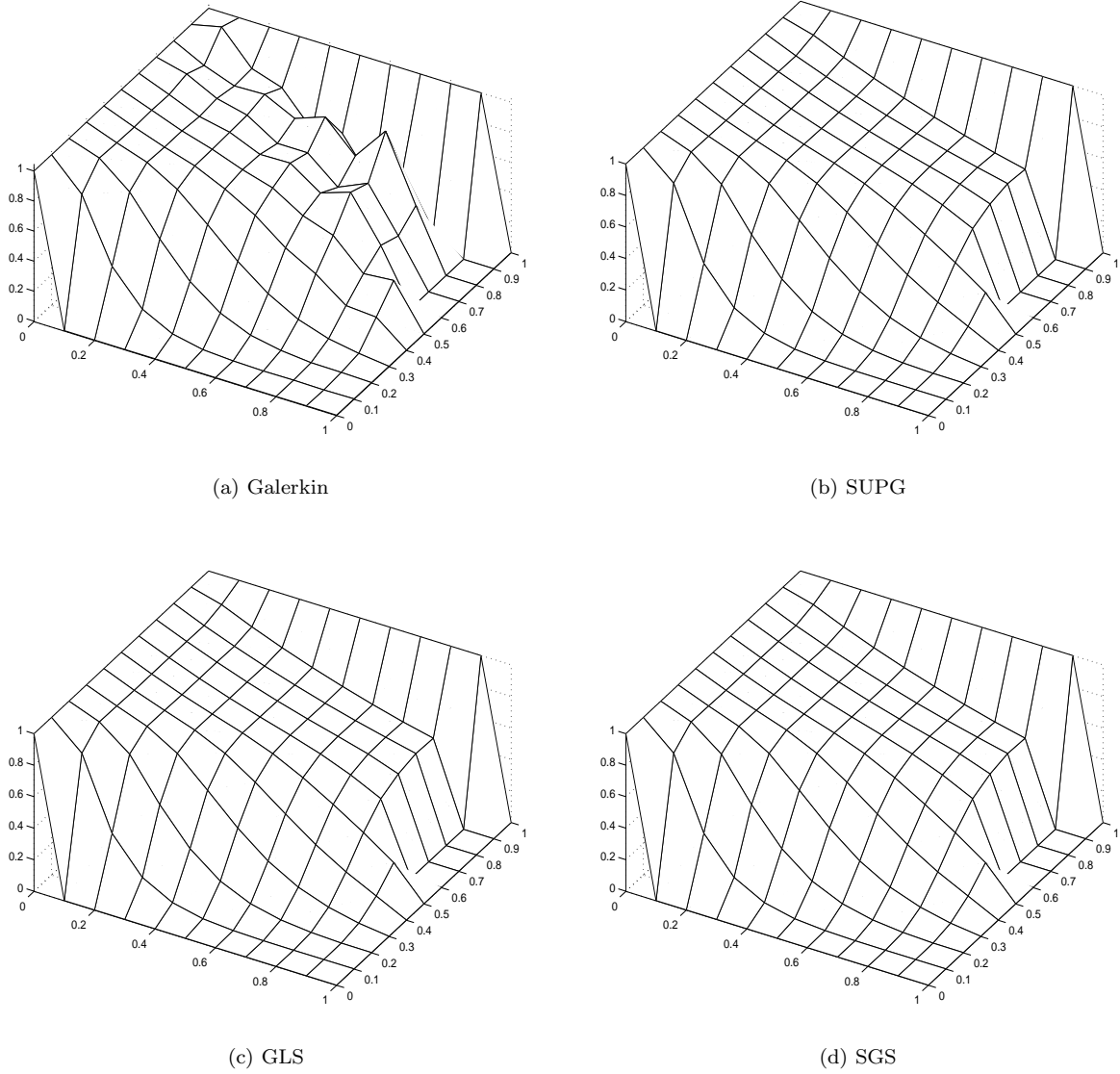
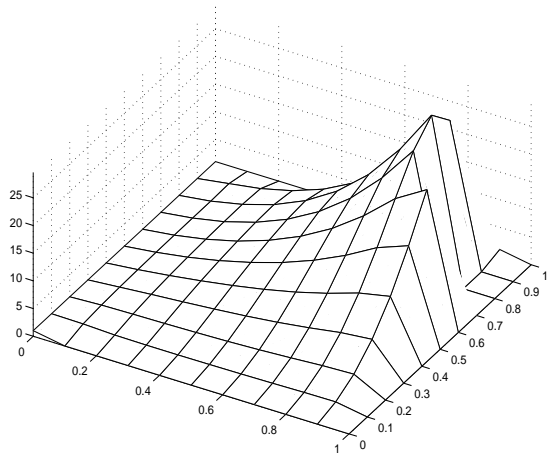


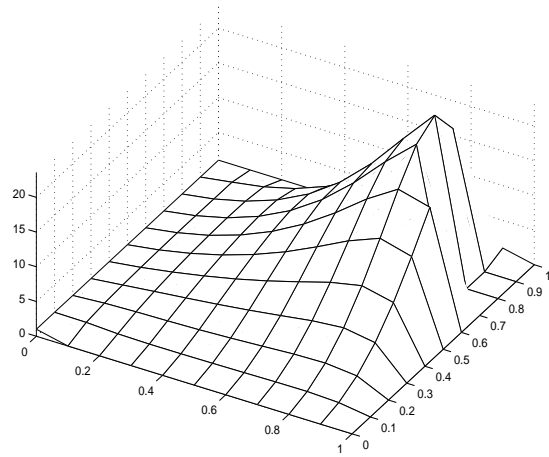
Figure 9: Solution of two-dimensional convection-diffusion problem. Advection skewed to the mesh. Numerical solutions obtained with 100 bilinear elements for $Pe = 10, \sigma = -1$

We can observe that the numerical solution by Galerkin method is a corrupted solution, and this behaviour is improved by the introduction of stabilizing terms according to the method employed. We can notice that the solutions obtained by stabilizing methods discussed here are very similar, given the principal contribution to stabilizing is the advective term, and by this reason all stabilizing methods have the same accuracy that SUPG method. Again this behaviour has been reported for one-dimensional case presented in the above section.

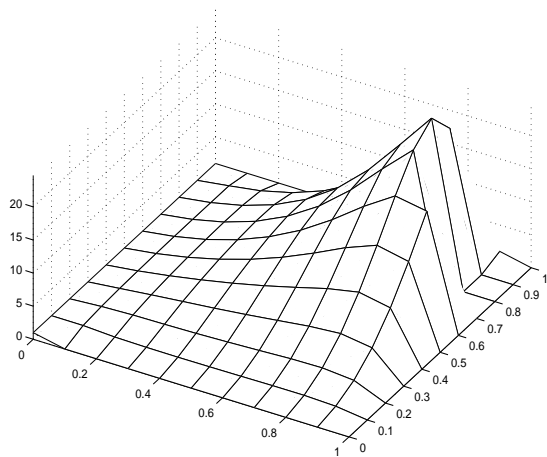
Figure 10 shows another case in exponential regime with production. In this case the value of Pe is significative, and then the difference between the stabilizing methods and the Galerkin method is noticeable. All stabilizing methods reduce the peak value in 20 percent approximately. This result show the influence of the stabilizing term in the numerical solution.



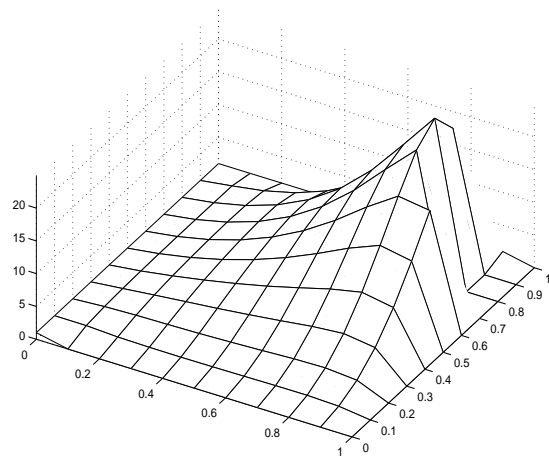
(a) Galerkin



(b) SUPG



(c) GLS



(d) SGS

Figure 10: Solution of two-dimensional convection-diffusion problem. Advection skewed to the mesh. Numerical solutions obtained with 100 bilinear elements for $Pe = 1.5, \sigma = 0.5$

6 Implementation Details

In this work various stabilizing methods have been validated by means of using of FEM-MATLAB program, which has been written in MATLAB, as intermediate step to write it in Fortran language. The program have been written in the philosophy of Finite Element Methods on a rectangular mesh, composed by quadrilaterals at this moment. In this way, several functions appear in whatever code of FEM and vary according the kind of element used at the discretization, and the procedure to compute the integrals that lead in the process of calculus, among them we have the following functions:

- **element.m** -compute derivatives of functions and Jacobian at Gaussian points **AJ**.
- **shapebq.m**-compute the the gaussian points, the corresponding weights and the shape functions **S(I,J)** and its derivatives at the Gaussian points in the master element.

The heart of the program is in the functions named **perturbation-par**, **transpo** and **rhsn**. These programs are written for the discretization of the transient equation relate to equation (1). It appear in form

$$\frac{\partial \phi}{\partial t} + \mathcal{L}\phi = f(x) \quad \text{in } \Omega$$

By applying of the θ -method the temporal derivative of the above equation takes the form

$$\frac{\partial \Phi^n}{\partial t} = f(x_{n-\frac{1}{2}}) - \{\theta(\mathcal{L}(\Phi^n) + (1-\theta)\mathcal{L}(\Phi^{n-1}))\}$$

where Φ^n represents the numerical solution associated to the ϕ values at the time $t = n\Delta t$. Computing the temporal derivative in backward form $\frac{\partial \Phi^n}{\partial t} = \frac{\Phi^n - \Phi^{n-1}}{\Delta t}$ and using the stabilized finite elements method which can be identified in the program FEM-MATLAB by the parameter **imeth** the above equation appear in a weak form

$$\mathbf{M}\Phi_h^n + \Delta t\theta(B(\Phi_h^n, w) + \hat{S}(\Phi_h^n, w)) = \hat{F}(w) + [\mathbf{M}\Phi_h^{n-1} - \Delta t(1-\theta)(B(\Phi_h^{n-1}, w) + \hat{S}(\Phi_h^{n-1}, w))]$$

$\hat{S}(\Phi_h^{n-1}, w)$ is related to the expressions ((17) - (19)) without the contribution of the source term, and $\hat{F}(w) = F(w) + \text{stabilization terms}$.

The function **transpo** compute the elemental local matrices by Galerkin method (**GALER** , the parameter τ to the stabilization terms obtained by the function **perturbation-par**, and the elemental contribution from the stabilizing terms according to method employed (**ESTAB**), and finally adds local contribution to the global matrix denotes here by **G** ,it is a advective-diffusive-reaction global matrix at the step $t = n\Delta t$.

```
GALER = zeros(NNE,NNE);
```

```
for J=1:NNE,
```

```
for I=1:NNE,
```

```
DIFU = ZERO;
```

```
ADVE = ZERO;
```

```
REAC = ZERO;
```

```
MASS = ZERO;
```

```
for L=1:NG,
```

```
DIFU = DIFU + kappa*(DSDX(L,I)*DSDX(L,J) + DSDY(L,I)*DSDY(L,J))/AJ(L);
```

```
ADVE = ADVE + S(L,I)*(UG(L)*DSDX(L,J) + VG(L)*DSDY(L,J));
```

```
MASS = MASS + S(L,I)*S(L,J)*AJ(L);
```

```
REAC = REAC + react*S(L,I)*S(L,J)*AJ(L);
```

```
end
```

```
GALER(I,J) = DIFU + ADVE - REAC;
```

```
end
```

```
RR(J) = MASS;
```

```
end
```

```
ESTAB = zeros(NNE,NNE);
```

```
for J=1:NNE,
```

```
for I=1:NNE,
```

```
DIFU = ZERO;
```

```
ADVE = ZERO;
```

```
REAC = ZERO;
```

```
MASS = ZERO;
```

```
for L = 1:NG
```

```
DIFU = DIFU + (UG(L)*DSDX(L,J) + VG(L)*DSDY(L,J))*(UG(L)*DSDX(L,I) + VG(L)*DSDY(L,I))/AJ(L);
```

```
ADVE = ADVE + (assign*S(L,I)*(UG(L)*DSDX(L,J) + VG(L)*DSDY(L,J)) -  
(UG(L)*DSDX(L,I) + VG(L)*DSDY(L,I))*S(L,J));
```

```

REAC = REAC + asign*react*S(L,I)*S(L,J)*AJ(L);
end
ESTAB(I,J) = tau*(DIFU + react*(ADVE - REAC));
end
end

```

where **NG**-represents the number of Gauss points to integrate the elemental matrices.

The subroutine **rhsn** computes the matrix G^{n-1} and local contributions of the source term, which are charge at the column vector **B**.

```

for I=1:NNE,
IN = NO(I,IE);
GALER = ZERO;
ESTAB-SUPG = ZERO;
ESTAB-S = ZERO;
for L=1:NG,
GALER = GALER + SG(L)*S(L,I)*AJ(L);
ESTAB-SUPG = ESTAB-SUPG + tau*( SG(L)*(UG(L)*DSDX(L,I) + VG(L)*DSDY(L,I)));
end
BV = ZERO;
for J=1:NNE,
for L=1:NG,
BV = BV + tau*(asign*S(L,J)*RG(L)*SG(L)*S(L,I)*AJ(L));
end
ESTAB-S = ESTAB-S + BV;
end
B(IN) = B(IN) + GALER + ESTAB-SUPG + ESTAB-S; end
end

```

The function **perturbation-par** computes the stabilizing parameter according to the selection of parameter formula, by means of the index **istab**, which is display as follows:

```

function [tauu,asign] = perturb-par(NDF,NG,NNE,NE,NN,NO,CO,S,DSDX,DSDY,
DIFU,U0,VS,IE,UG,VG,X,Y,AJ,REACT,DAMK,istab,imeth)

```

```

tauu = 0;
if (imeth==0), asign = 0;end
if (imeth==1), asign = 0;end
if (imeth==2), asign = -1;end
if (imeth==3), asign = 1;end
global ZERO
global ONE
global TWO
global FOUR
global EIGHT
global SU

```



```

ZERO = 0;
ONE = 1;
TWO= 2.0;
FOUR = 4;
EIGHT = 8;
h1x = (X(2) + X(3)-X(1)-X(4))/TWO;
h1y = (Y(2) + Y(3)-Y(1)-Y(4))/TWO;
h2x = (X(3) + X(4)-X(1)-X(2))/TWO;
h2y = (Y(3) + Y(4)-Y(1)-Y(2))/TWO;
VX = (U0(1) + U0(2) + U0(3) + U0(4))/FOUR;
VY = (VS(1) + VS(2) + VS(3) + VS(4))/FOUR;
VMOD = VX*VX + VY*VY;
VMOD = sqrt(VMOD);
COX = VX/VMOD;
COY = VY/VMOD;
h1 = COX*h1x + COY*h1y;
h2 = COX*h2x + COY*h2y;
hele = abs(h1) + abs(h2);
PEC = VMOD*hele/(2*DIFU);
if PEC == 0,
ALPHA = 0;
V2MOD
elseif PEC < 1,
ALPHA = 1/tanh(PEC) - 1/PEC;
else
ALPHA = PEC/3;
end
for I=1:NNE,
for L=1:NG,
V2MOD = UG(L)*UG(L) + VG(L)*VG(L);
V2MOD = sqrt(V2MOD);
if abs(V2MOD) < 1e-12,
assign = 0;
else
if (istab==1),
tauu = ALPHA*(hele/(2*V2MOD));
elseif (istab==2),
tauu = 1/(4*DIFU/hele2 + 2*VMOD/hele + abs(REACT));
elseif (istab==3),
Pe1 = 1/(PEC/3*abs(DAMK));
Pe2 = 2*PEC/3;
mkk = (6*DIFU/(hele2));
tauu = 1/(mkk*max(1,Pe2) + abs(REACT)*max(1,Pe1));
else
end
end
end
end
return

```

Finally a linear equation system must be solved in each time step, and the numerical solution is obtained. The implementation of the algebraic part at this moment is very easy, taking into account that this code is only an intermediate step to professional programming. In virtue of this the storage of global matrix is in compact form, computed the bandwidth of matrix, the linear system has been solved by direct method employing the inverse matrix operator from MATLAB, and the numerical solution is computed step by step. For a professional code it is necessary to implement Iterative methods and keep the matrix in skyline format.

7 Conclusions

Numerical methods that use central differences on uniform meshes don not have a mechanism for stabilizing solutions in exponential layers or in the propagation regime, which exhibits wild oscillations near internal or boundary layers, and thus, unphysical results in certain cases. As we show in this work, several strategies can be handled by modifying the approximation of convective terms or source terms, such that the resulting operator is monotone, and the resulting global matrix a an M-matrix. When this is obtained, the perturbation terms introduced originates an improvement in the behaviour of numerical solution inside and near the layers, and the computed solutions are more precise. All methods revised in this work are very dependent of the choice of stabilizing parameter, which has been derived in several manners. However, there is no optimal projection method, neither an optimal stabilizing parameter for all ranges of scales that are present in a advection-diffusion-reaction problem.

Based on the numerical experience gained with this work, it is convenient to make a further investigation in improved the characterization of the subgrid scale. Numerical results confirm the good performance of this method relative to SUPG and GLS method in a wide range of scales.

8 Acknowledgements

This work has received financial support from Ministerio de Ciencia y Tecnologia (McyT, España) through Programa Ramonycajal and from Polytechnic Univeristy from Cataluña (Barcelona, España). We made extensive use of software distributed by the Free Software / GNU-Proyect:Linux ELF-OS, Octave, Tgif from William C.Cheng, and others. The author want to express his gratitude to Kees Vuik and F.J.Vermolen by your opinions and suggestions.

References

- [1] A.Brooks and T.J.R.Hughes. Streamline upwind Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations. *Comput. Meths. Appl. Mech. Engrg.*, 32:199–259, 1982.
- [2] A.Russo. Bubble stabilization of the finite element methods for the linearized incompressible Navier-Stokes equation. *Comput. Methods. in Applied Mech. and Eng.*, 132:335–343, 1996.
- [3] F. Brezzi. On the existence, uniqness and approximation of saddle-point problems arising from Lagrange multipliers. *Rairo Ser. Rouge*, 8:129–151, 1974.
- [4] A. Russo F. Brezzi, L.D.Marini. On the choice of a Stabilizing Subgrid for Convection-Diffusion Problems. *Comput. Methods Appl. Mech. Engrg.*, 194(2-5):127–148, 2005.
- [5] F. Brezzi, L.P. Franca, T.J.R. Hughes, and A. Russo. $b = \int g$. *Rairo Ser. Rouge*, 145:329–339, 1997.
- [6] G.Hauke. A simple subgrid scale stabilized method for the advection-diffusion-reaction equation. *Comput. Methods. in Applied Mech. and Eng.*, 191:2925–2947, 2002.
- [7] Harari I., Hughes Thomas J.R. Stabilized finite element methods for steady advection-diffusion with production. *Computer Methods in Applied Mechanics and Engineering*, 115:165–191, 1994.
- [8] I. Babuska. The finite elment method with Lagrangian multipliers. *Numer. Math.*, 20:179–192, 1973.
- [9] A. Masud L.P.Franca, G.Hauke. *Stabilized Finite Elements Methods*, chapter Finite Element Methods: 1970’s And beyond. CIMNE, Barcelona, Spain, 2003.
- [10] L.P.Franca, C.Farhat. Bubble functions prompt unusual stabilized finite element methods. *Comput. Methods. in Applied Mech. and Eng.*, 123:299–308, 1995.
- [11] L.P.Franca, F. Valentin. On an improved unusual stabilized finite element method for the advection-reactive-diffusive equation. *Comput. Methods. in Applied Mech. and Eng.*, 190:1785–1800, 2001.
- [12] R.Codina. Comparison of some finite element methods for solving the diffusion-convection-reaction equation. *Comput. Methods. in Applied Mech. and Eng.*, 156:185–210, 1998.

- [13] Martin Stynes. Steady-state convection-diffusion problems. *Acta Numerica*, pages 445–508, 2005.
- [14] T.J.R. Hughes. Multiscale phenomena: Green’s functions, the Dirichlet-to-Neumann formulation, subgrid scale models, bubbles and the origins of stabilized methods. *Comput.Methods in Applied Mechanics and Engineering*, 127:387–401, 1995.
- [15] T.J.R. Hughes, L.P. Franca, and G.M. Hulbert. A new finite element formulation for fluid dynamics: Viii. The Galerkin/least-squares method for the advection-diffusive equations. *Comput. Methods Appl. Mech. Engrg.*, 73:173–189, 1989.
- [16] T.J.R. Hughes, L.P. Franca, and M. Balestra. A new finite element formulation for computational fluid dynamics: V. circumventing the Babuska-Brezzi condition: a stable Petrov-Galerkin formulation of the Stokes problem accomodating equal-order interpolations. *Comput. Methods Appl. Mech. Engrg.*, 59:85–99, 1986.