Numerical Stabilization of one-dimensional Convection-Diffusion Problems by Algebraic Methods

Ruperto P. Bonet Chaple
Applied Mathematic Department I, DMA1
ETSEIB-Catalonia University
Avda. Diagonal 647
08028 Barcelona, Spain
e-mail: ruperto.bonet@upc.edu
Web page: //www-ma1.upc.es

1 SUMMARY

We are focused to the task of stabilizing discrete approximations to convection-diffusion problems on non-uniform mesh. This paper reports progress on a numerical technique to compute the stabilization parameters for streamline-diffusion FEM. This technique is based on the approximate symmetrization to the global matrix of resulting linear equations system. A first approximation to stabilization parameter is obtained reducing the imaginary part of local matrix spectrum to zero. A second approximation is obtained introducing a diffusion matrix which is computed in different manner, as in iterative procedure or by calculus direct. Improvement with this procedure relative to usual SUPG can be appreciated as the flux term and /or source term are spatially variable.

KEY WORDS: stabilization, convection-diffusion problems, direct method, SUPG, iterative procedure, approximate symmetrization

2 INTRODUCTION

Convection-Diffusion process are present in many areas of engineering. In several applications are very often that the dimensionless parameter that measures the relative strength of the diffusion is quite small; so one often meets situations where thin boundary and interior layers are present and singular perturbation problems arise. Important efforts have been developed the last few decades on a variety of techniques for analyzing and overcoming these difficulties.

It is well known that the Galerkin FEM applied to convection-dominated flow problems gives unphysical oscillatory solutions [1]. To eliminate such oscillations, various 'upwind' finite element schemes have been developed [[2],[3]], among them, which has been more applied is the optimal 'upwind' / Petrov-Galerkin method. With this method the one-dimensional convection-diffusion equation with constant coefficients has exact solutions in each node [4]. A multidimensional generalization of the one-dimensional optimal theory, was done by Hughes and Brooks with the presentation of the streamline 'upwind' / Petrov-Galerkin (SUPG) method [[3],[5]].

Over the last few years a very large literature has built up on a variety of techniques for the stabilization of convection-diffusion equation. However, the control of stabilization parameter is being the principal task in whatever numerical formulation for dominant convection problems. Explicit expressions to compute this parameter have been reported early. Some of them are based on the fact that a boundary layer is usually exponential in form [6]. This procedure can only be successfully for linear two node elements. Recently an adaptive procedure via diminishing the average residual has been reported[7]. However, in this procedure an elemental residual for an enhanced solution must be calculated. This procedure has a good performance only in the cases where the Peclet number is greater than five, and the velocity flux and source term are constants. The main lack of this procedure is to find a successful way to compute the derivative of the average residual for an enhanced solution. Recently more, Torsten showed that within the layers the stabilization must be drastically reduced, using a residual free bubbles approach, but the precision is only high on very fine meshes or adaptive meshes[8].

In this work a systematic procedure to compute the stabilization parameter of central difference schemes for steady convection-diffusion problems is developed. Using $SUPG$ formulation, an iterative method to minimized the imaginary part of local matrix spectrum is presented. This approximation originates precise solutions only in case of the isotropic physical properties and uniform meshes. A spatially variable correction term is introduced in case of non-uniform mesh, which is computed in iterative form or by direct calculus. This method allows to transform the global matrix to **quasi-symmetric global matrix**. With the proposed method can be stabilized convection-diffusion problems where the velocity and source terms are spatially variables.

The layout of this paper is as follows. First, the steady state convection-diffusion equation is presented. Second, an brief explanation about $SUPG$ method is given, and the classical analytical expression to compute the artificial diffusion parameter is derived. Third, a statement of spectral problems and basic facts are presented by means of an heuristic procedure. Fourth, a local algebraic method for identification of stabilization parameter via to solve a discrete spectral problem on an uniform mesh is proposed. Fifth, an extension of this procedure to non-uniform meshes including the case of variable flux is developed. Sixth, several examples that confirm the good performance of this method on non-uniform meshes are presented.

3 SUPG METHOD

An convection-diffusion equation is an important model to study transport problem. Here, we consider the one-dimensional steady state convection-diffusion equation with Dirichlet boundary conditions. This equation can be written in the following form

$$\mathcal{L}\phi = U(x)\frac{d\phi(x)}{dx} - k\frac{d^2\phi(x)}{dx^2} = Q(x) \qquad in \quad a \leq x \leq b \tag{1}$$

where ϕ is the transported variable, U is the velocity field, k is the diffusion parameter and Q is the source term, a and b real numbers ($a < b$). The Dirichlet boundary conditions associated to the equation (1) are

$$\phi(a) = g1, \quad \phi(b) = g2 \tag{2}$$

Let us consider now a standard FEM approximations written as

$$\phi \approx \hat{\phi} = \sum_{i=0}^{\text{nel}} \mathbf{N}_i \phi_i \tag{3}$$

where nel denotes the element number at the mesh, $\mathbf{N}^i$ are the global piecewise linear basis functions and ϕ^i are the nodal values of the approximate $\hat{\phi}$ at node i. One important measure of $\hat{\phi}$ as an approximation to ϕ is the error (or algebraic error) and is given by $\mathbf{e}rr = \phi - \hat{\phi}$. Using the weighted residual procedure over the domain $\Omega = [a, b]$, the weighted residual form for $1D$ convection-diffusion problem is now written as

$$\int_{\Omega} (U(x)\frac{d\hat{\phi}}{dx} W_i + \frac{d\hat{\phi}}{dx} k \frac{dW_i}{dx}) d\Omega = \int_{\Omega} QW_i d\Omega + \int_{\Gamma} \frac{d\hat{\phi}}{dx} kW_i ds \tag{4}$$

Equation (4) is usually written in matrix form

$$\mathbf{A}\hat{\phi} = \mathbf{f} \tag{5}$$

where $\hat{\phi} = [\phi_1, \phi_2, \ldots, \phi_{(\text{nel}-1)}]^T$, and matrix $\mathbf{A}$ and vector $\mathbf{f}$ are obtained by standard assembly of the element contributions given by

$$\mathbf{A}^e_{ij} = \int_{\Omega^e} (U(x)\frac{d\hat{\phi}}{dx} W_i + \frac{d\hat{\phi}}{dx} k \frac{dW_i}{dx}) d\Omega \tag{6}$$

$$\mathbf{f}^e_i = \int_{\Omega^e} QW_i d\Omega + \int_{\Gamma} \frac{d\hat{\phi}}{dx} kW_i ds \tag{7}$$

Here, the boundary integral term is not considered because Dirichlet boundary conditions are imposed on this boundary, where the value of ϕ is prescribed. The Streamline Upwind Petrov-Galerkin ($SUPG$) form of equations ((4) - (6)) is simply obtained by making

$$W_i = N_i + \frac{\alpha_e h_e}{2} \frac{dN_i}{dx} sign(U) \qquad i = 0, 1, 2, \ldots, \text{nel} \tag{8}$$

where α_e is a local parameter and h_e is the element length. Here α_e characterizes the artificial diffusion in each element, which is added to the problem to guarantees the numerical stability, therefore, the selection of optimal parameter allows the regularizing of the numerical scheme derived by Galerkin method in presence of the dominant convection problems. The limit cases are $\alpha_e = 0$ and $\alpha_e = 1 (\alpha_e >= 1)$ for all elements. The first case corresponds to the Galerkin formulation, while the second case is relative to a diffusive purely formulation.

A finite element formulation to one-dimensional transport equation with $U (U > 0), k$ and Q constants, employing linear shape functions and uniform mesh originates the following difference scheme

$$[-\mathbf{Pe}(\alpha_e + 1) - 1]\hat{\phi}_{i-1} + [2 + 2\alpha_e(\mathbf{Pe})]\hat{\phi}_i + [-\mathbf{Pe}(\alpha_e - 1) - 1]\hat{\phi}_{i+1} + \frac{Qh_e^2}{k} = 0 \tag{9}$$

for $i = 1, 2, \ldots, \text{nel}$, where $\mathbf{Pe} = \frac{Uh_e}{2k}$ is the element Peclet number. This dimensionless number expresses the discrete relation between the convection and diffusion terms. It is necessary to obtain the α_e values that guarantees to get exact nodal values of $\hat{\phi}$ for all **Pe** values. Taking $Q = 0$ and U, k constants, the value of α_e giving the "exact" solution at the nodes can be also simply derived. From (9) leads

$$\alpha_e = \frac{(\mathbf{Pe}+1)\hat{\phi}_{i-1} - 2\hat{\phi}_i + (1-\mathbf{Pe})\hat{\phi}_{i+1}}{-Pe(\hat{\phi}_{i-1} - 2\hat{\phi}_i + \hat{\phi}_{i+1})} \tag{10}$$

In the case of $Q = 0, g_1 = 0$ and U, k constants, the exact solution of (1) has an exponential form. Substituting the exact solution of (1) into (10) gives after some algebra

$$\alpha_e = \coth(\mathbf{Pe}) - \frac{1}{\mathbf{Pe}} \tag{11}$$

It is well known that using this expression can be obtained exact nodal values of $\hat{\phi}$ for all **Pe** values in the case of U, k, Q are constants[6]. In the limit case, when the **Pe** values are very big the expression (11) becomes $\alpha_e = \alpha^* = 1 - \frac{1}{Pe}$. Usually, this expression is used also when the parameters U, k, Q have important variations within the physical domain. However, there are cases where the numerical solution resulting has appreciable errors.

The aim of this paper is to derive a discrete procedure to get α values that guarantees the stability and accuracy of the numerical solution of problems with variable velocity flux on non-uniform meshes and for all values of the Peclet numbers **Pe**. As will show in the following sections the problem is to reconstruct $\alpha(\hat{\phi}_i)$ at each element e for all **Pe** values, this means, that α represents an arbitrary value of local parameter α_e.

4 DETERMINING THE STABILIZATION PARAMETER ON AN UNIFORM MESH

In this section an heuristic procedure based on the spectral properties of the global matrix **A** will be developed. In this way, it is convenient to write the scheme (9) in matrix-vector form, using $\hat{\phi} = [\phi_1, \phi_2, \ldots, \phi_{(\mathrm{nel}-1)}]^T$ to denote the numerical approximation of ϕ at the interior nodes, therefore, the matrix $\mathbf{A} = tridiag(-\mathbf{Pe}(\alpha+1) - 1, 2 + 2\alpha(\mathbf{Pe}), -\mathbf{Pe}(\alpha - 1) - 1)$ such that $size(A) = n - 1$. For the matrix **A** given above the spectrum can be calculated explicitly, by the expression

$$\sigma(\mathbf{A}) = 2(1+\alpha\mathbf{Pe}) + 2\mathrm{i}\sqrt{\mathbf{Pe}^2 - (1+\alpha\mathbf{Pe})^2}\cos\left(\frac{\mathrm{l}\pi}{\mathrm{nel}}\right) \quad l = 1, 2, \cdots, (\mathrm{nel}-1) \tag{12}$$

where $0 \leq \alpha \leq \alpha^*$. If $l \neq \frac{\mathrm{nel}}{2}$ the zeros from the imaginary part of the spectrum $\sigma(\mathbf{A})$ are solutions of a second order algebraic equation in the variable α,

$$\alpha^2 + \frac{2}{\mathbf{Pe}}\alpha + \left(\frac{1}{\mathbf{Pe}^2} - 1\right) = 0 \tag{13}$$

which, has two real roots and only one positive root. After some algebra, employing the Vietta formula, a unique positive solution was obtained, $\alpha = \alpha^*$. This fact shows that it is possible, to compute the adequate parameter $\alpha = \alpha(\mathbf{Pe})$ reducing to zero the imaginary part of the global matrix spectrum, and solving by some numerical method the resulting equation. As we can observe in (12) that the positive solution $\alpha = \alpha^*$ is unique independently of element number taking into account to compute the spectrum. In virtue of this, we form the local matrix $\mathbf{A}_l = \mathbf{A}^l$ by the entries $\mathbf{A}(\mathbf{i},\mathbf{j})$ corresponding to nodes $\mathbf{l}$ and $(\mathbf{l}+\mathbf{1})$ at the global matrix. In the practice, we consider two equations:

$$\max[Im\{\sigma(\mathbf{A}_l)\}]^2 = 0 \quad (\gamma = 0) \qquad \text{or} \tag{14}$$

$$\max[\frac{Im\{\sigma(\mathbf{A}_l)\}}{Real\{\sigma(\mathbf{A}_l)\}}]^2 = 0 \quad (\gamma = 1) \tag{15}$$

which will be examined in explicit form as follow

$$f(\alpha) = \frac{4(\mathbf{Pe}^2 - (1+\alpha\mathbf{Pe})^2)}{4^\gamma(1+\alpha\mathbf{Pe})^{2\gamma}} = 0 \qquad \gamma = 0, 1 \tag{16}$$

where the equations (14) and (15) represent the cases $\gamma = 0$ ($\gamma = 1$) , respectively. We now discuss the existence and uniqueness of the solutions to the γ-equation (16). As we can appreciate by direct calculus,

$$f(0) = \frac{\mathbf{Pe}^2 - 1}{4^{\gamma-1}} > 0 \tag{17}$$

$$f(1) = \frac{-1 - 2\mathbf{Pe}}{4^{\gamma-1}(1+\mathbf{Pe})^{2\gamma}} < 0 \tag{18}$$

for $\mathbf{Pe}$ values greater than one ($\mathbf{Pe} > 1$). The function $f(\alpha)$ is strictly decreasing on $(0, 1]$, because the derivative function $f^{'}(\alpha) < 0$ for all α values belong to open interval $(0, 1)$. Continuity of f together $f(1) \leq 0 < f(0)$ implies that there exist an unique solution $\alpha^* \in (0, 1]$ to the γ-equation (16).

4.1 An alpha-adaptive local stabilization scheme on an uniform mesh

In this section, we propose an alpha-adaptive local stabilization scheme based on spectral properties of local matrix $\mathbf{A}_l = \mathbf{A}^l$. Now, we denotes the stabilization parameter estimation vector by $\hat{\alpha} = [\alpha^{(1)}, \alpha^{(2)}, \ldots, \alpha^{(nel)}]^T$. Next, we will use only α to denotes an arbitrary component from $\hat{\alpha}$ vector, because in case of uniform mesh this problem can be reduced to one-dimensional parameter estimation problem.

For practical purposes, we introduce a cutoff-level $\delta(\ \delta > 0)$, such that the stabilization parameter α should be chosen according to

$$f(\alpha) = \max[\frac{[Im\{\sigma(\mathbf{A}_l)\}]^2}{[Real\{\sigma(\mathbf{A}_l)\}]^{2\gamma}}] = \delta \quad (\gamma = 0,\ 1) \tag{19}$$

and then, the local matrix $\mathbf{A}_l$, will be become in δ-self-adjoint matrix, where δ is a given precision level in the numerical computations. Also, we introduce the following tolerance levels $\epsilon_1,\ \epsilon_2,\ \epsilon_3$, and the maximum number of iterations ($iter_{max}$). Tolerance levels are relative to the magnitudes $||\hat{\phi}_k - \hat{\phi}_{k-1}||_2,\ ||\hat{\alpha}_k - \hat{\alpha}_{k-1}||_2,\ ||\hat{\phi}_k - \hat{\phi}_{k-1}||_2 \times ||\hat{\alpha}_k - \hat{\alpha}_{k-1}||_2$, respectively, for $k = 2, 3, \ldots, iter_{max}$.

In the numerical experiments we calculate the local matrix $\mathbf{A}_l$ and its eigenvalues numerically using $MATLAB$ 5.2. The accuracy of these numerical results are checked with the explicit formula given in (11). In this work, the equation (19) is solved by

Newton's method or Quasi-Newton's method, which are denoted as follow: **N0**: Newton's method for solving equation (19) with $\gamma = 0$, and calculating the spectrum of matrix $\mathbf{A}_l$ numerically, and $f'(\alpha)$ in exact manner; **N1**: Newton's method for solving equation (19) with $\gamma = 1$, and calculating the spectrum of matrix $\mathbf{A}_l$ numerically, and $f'(\alpha)$ in exact manner; **Bis**: Bisection Method on the extreme left of parameters interval, while $\hat{\alpha}$ satisfies the inequality $f(\alpha) > \delta$; **QN0**: Quasi-Newton's method for solving equation (19) with $\gamma = 0$, and calculating the spectrum of matrix $\mathbf{A}_l$, and $f'(\alpha)$ numerically; **QN1**: Quasi-Newton's method for solving equation (19) with $\gamma = 1$, and calculating the spectrum of matrix $\mathbf{A}_l$, and $f'(\alpha)$ numerically.

To accelerate the convergence and get more precision is designed the Hybrid method, which consist in the application of Bisection method during the first k steps, and after a type Newton's method. For this, we introduce the notations: $\mathbf{Bis} + \mathbf{QN0}$: - Bisection Method in combination with the Quasi-Newton's method for solving equation (19) with $\gamma = 0$, and $\mathbf{Bis} + \mathbf{QN1}$- Bisection Method in combination with the Quasi-Newton's method for solving equation (19) with $\gamma = 1$.

The following scheme can be devised to obtain a stable numerical solution in an adaptive manner.

$\alpha -$ **Algorithm** :

1. **Initialization**. Input $\hat{\alpha}_0,\ \gamma,\ \delta > 0, \epsilon_1,\ \epsilon_2,\ \epsilon_3,\ iter_{max},\ \hat{\phi}_0$, set $k := 0$;

2. **Loop of calculus**. Do while ($||\hat{\phi}_k - \hat{\phi}_{k-1}||_2 > \epsilon_1$, and,$k \leq iter_{max}$), k = k + 1;

    - Form a local matrix $\mathbf{A}_l$ using **SUPG** formulation and to compute the matrix spectrum $\sigma(\mathbf{A}_l)$ (linear case). Additionally, in non-linear case, to solve the **forward** problem by **SUPG** formulation using a guess of the stabilization parameter.
    - Computation of $\max\left[Im\{\sigma(\mathbf{A}_l)\}\right]^2$, max $\left[Real\{\sigma(\mathbf{A}_l)\}\right]^{2\gamma}$ and the Jacobian function $f'(\alpha_0)$ according to the employed method.
    - Do while ($\frac{\sqrt{||f(\hat{\alpha}_k)||}}{\mathbf{Pe}^{(1-\gamma)}} > \delta$), solve δ-equation (19) for $\hat{\alpha}_{k+1}$ from iterative formula (Newton, Quasi-Newton, Bisection or Hybrid method).

3. **Convergence Tests**. Check the following inequalities

$$||\hat{\alpha}_k - \hat{\alpha}_{k-1}||_2 < \epsilon_2, \quad \text{or} \tag{20}$$

$$||\hat{\alpha}_k - \hat{\alpha}_{k-1}||_2 > \epsilon_2 \quad \text{and} \quad ||\hat{\phi}_k - \hat{\phi}_{k-1}||_2 \times ||\hat{\alpha}_k - \hat{\alpha}_{k-1}||_2 < \epsilon_3 \tag{21}$$

   with $\delta = 10^{-3}, \epsilon_1 = 10^{-12},\ \epsilon_2 = 10^{-8}$, and $\epsilon_3 = 10^{-8}$.

4. **Stop rule**. If conditions ((20) - (21)) are satisfied, or $k == iter_{max}$, exit algorithm; otherwise, goto step 2.

5 EXTENSIONS OF LOCAL STABILIZATION FORMULA TO NON-UNIFORM MESHES

In above sections the weighted residual is uniformly weighted over the interval of integration, which reflects a tacit assumption that information is equally available and equally reliable everywhere in x, however, it may be wrong, in case of the physical properties vary spatially at the domain. Next we obtain a discrete (1) on a generally non-uniform mesh

$$a = x_1 < x_2 < \ldots < x_{\text{nel}} < x_{\text{nel}+1} = b \tag{22}$$

where nel is the element number at mesh. We suppose that the mesh in (22) is a sufficiently smooth transformation of a uniform mesh with element size $\bar{h}_e$. Thus, we assume that there is a differentiable, monotonically increasing function $\psi : [a, b] \rightarrow [a, b]$ such that

$$\psi(\xi_l) = x_l \qquad l = 1, 2, \ldots, (\text{nel} + 1) \tag{23}$$

Clearly this function is invertible, so there is a corresponding Jacobian transformation $\omega = \frac{dx}{d\xi}$, being $\omega = 1$ in those elements, where element size $h_e = \bar{h}_e$. In the following, we will derive specific expressions to the test functions type W_i given in (8), that allows to catch the non-uniformity effects arisen by a non-uniform mesh or by the presence of non-homogeneous flux and/or source terms. In virtue of it the formula (8) is modified as follows:

$$W_i = N_i + \frac{\alpha_e^{nh} h_e}{2}\frac{dN_i}{dx}\text{sign}(U) = N_i + \frac{(\alpha_e + \beta(x))h_e}{2}\frac{dN_i}{dx}\text{sign}(U) \quad i = 0, 1, 2, \ldots, \text{nel} \tag{24}$$

where α_e^{nh} characterizes the anisotropic diffusion to element level. Thus, we would like to get for each $x \in \widetilde{\Omega}$, a $\beta(x)$ value such that $\alpha_e^{nh} = \alpha_e + \beta(x)$. Here $\widetilde{\Omega}$ is the union of element interiors, i.e., $\widetilde{\Omega} = \cup_{e=1}^{nel}\Omega^e$. Let us consider that the $\beta(x)$ function takes the form $\beta(x) = \beta \mathrm{p}^{\mathrm{e}}(\mathrm{x})$ where p^{e} a smooth function over an element interior and the β parameter is a scale factor to the function p. For $\beta = 0$, the numerical solution becomes in the numerical solution obtained by SUPG method (with $\alpha_e^{nh} = \alpha_e$) and for $\beta = 1$ the numerical solution is influenced by the presence of anisotropy of the mesh or non-homogeneities at the medium. Here, the β parameter and the p^{e} function are unknowns at the formula

$$\alpha_e^{nh} = \alpha_e + \beta \mathrm{p}^{\mathrm{e}}(\mathrm{x}) \tag{25}$$

Formula (25) is applied to element level, while β values can be obtained to global level. Substituting the formula (25) into (24) and repeating the procedure (3)-(4), appears the global matrix $\mathbf{A}$ in form

$$\mathbf{A} = \mathbf{D} + \mathbf{K}(\alpha^{nh}) \tag{26}$$

where $\mathbf{D}$, $\mathbf{K}(\alpha^{nh})$ are obtained by standard assembly of the element contributions given by

$$\mathbf{D}_{ij}^e = \int_{\Omega^e} U(x)\frac{dN_j}{dx}N_i dx \tag{27}$$

$$\mathbf{K}_{ij}^e(\alpha_e^{nh}) = \frac{h_e}{2}\text{sign}(U)\int_{\Omega^e} \alpha_e^{nh} U(x)\frac{dN_j}{dx}\frac{dN_i}{dx}dx + \int_{\Omega^e}\frac{dN_j}{dx}k\frac{dN_i}{dx}dx \tag{28}$$

Now, we define the matrix $\mathbf{R}(\alpha^{nh})$ by

$$\mathbf{R}(\alpha^{nh}) = \mathbf{I} - [\mathbf{K}(\alpha^{nh})]^{-1}\mathbf{A} \tag{29}$$

which express the discrepancy between the non-symmetric Galerkin matrix A, and the quasi-symmetric matrix $A(\alpha^{nh})$ derived by (25). It is clear that the matrix A to be a symmetric matrix for $||\mathbf{R}(\alpha^{nh})|| = 0$, however, it is only can be reached approximately. Well, the main goal of this paper is to develop a procedure such that the global matrix A becomes in **quasi-symmetric** matrix. In virtue of this, the problem consist in:

Given $\delta > 0$ enough small and a spatial function p(x), with $x \in \widetilde{\Omega}$ to find a β parameter, such that

$$\mathbf{R}(\alpha^{nh}) = \delta \mathbf{I} \tag{30}$$

Putting $\alpha_e = 1 - \frac{2k}{U_e h_e}$ into (30), the matrix $\mathbf{K}(\alpha^{nh}) \approx \mathbf{P} + \mathbf{K}(\beta)$, such that

$$\mathbf{P}^e_{ij} = \frac{h_e}{2}\text{sign}(U)\int_{\Omega^e} U(x)\frac{dN_j}{dx}\frac{dN_i}{dx}dx \tag{31}$$

$$\mathbf{K}^e(\beta)_{ij} = \frac{\beta h_e}{2}\text{sign}(U)\int_{\Omega^e} U(x)\frac{dN_j}{dx}\mathrm{p^e(x)}\frac{\mathrm{dN_i}}{\mathrm{dx}}\mathrm{dx} \tag{32}$$

and, then, the eigenvalue problem (30) can be written in form

$$\mathbf{D} = \delta(\mathbf{P} + \mathbf{K}(\beta)) \tag{33}$$

5.1 Determining the function p(x)

Mesh Transformation method (MT). Using the mesh transformation ψ defined in (23) we introduce the following relation:

$$x = \xi + \beta\frac{d\xi}{d\tau}\frac{h_e}{2} \tag{34}$$

where τ is an evolution time scale in the flow direction, $h_e = x_{(l+1)} - x_l$, and the factor β is the order $O(time)$. This relation expresses the displacement of each interior node along the domain originated by ψ transformation. Now, we can rewrite the test functions (8) to uniform mesh as follow

$$W_i(\xi) = N_i(\xi) + \frac{\alpha_e \bar{h}_e}{2}\frac{dN_i(\xi)}{d\xi}\text{sign}(U) \tag{35}$$

and for generally a non-uniform mesh,

$$W_i(x) = N_i(x) + \frac{\alpha_e^{nh} h_e}{2}\frac{dN_i(x)}{dx}\text{sign}(U) \tag{36}$$

Evaluating $W_i(x)$ according to (34) and taking its first approximation leads

$$W_i(x) = N_i(\xi) + \beta\frac{h_e}{2}(\frac{d\xi}{d\tau})\text{sign}(U)\frac{dN_i(x)}{d\xi}|_{x=\xi} \tag{37}$$

Using that

$$\frac{dN_i(\xi)}{d\xi} = \frac{dN_i(\xi)}{dx}\frac{dx}{d\xi} \tag{38}$$

$$\frac{dx}{d\tau} = \frac{dx}{d\xi}\frac{d\xi}{d\tau} \tag{39}$$

the equation (37) takes the form

$$W_i(x) = W_i(\xi) + \beta\frac{h_e}{2}(\frac{dx}{d\tau})\text{sign}(U)\frac{dN_i(\xi)}{dx} \tag{40}$$

and then, substituting (35) and (36) into (40) leads the relation

$$\alpha^{nh} = \alpha_e + \beta(\frac{dx}{d\tau}) \tag{41}$$

after some algebra. We can appreciate that for $\beta = 0$ or $\frac{dx}{d\tau} = 0$ this formula to be reduced to local stabilization formula (11). Therefore, the formula (41) enhances the validity of (11) to anisotropic mesh. This formula can be applied to anisotropic case, however, big variations of flux on an element may not be dissipated. Substituting (41) into equation (36) appears the formula

$$W_i(x) = N_i(x) + \frac{\alpha_e h_e}{2}\text{sign}(U)\frac{dN_i(x)}{dx} + \beta\frac{h_e}{2}(\frac{dx}{d\tau})\text{sign}(U)\frac{dN_i(x)}{dx} \tag{42}$$

Comparing this formula with the equality (24) we conclude that the smooth function p(x) takes the form

$$\mathrm{p^e(x)} = \frac{\mathrm{dx}}{\mathrm{d}\tau} \tag{43}$$

Error Equation Method (EE). When the SUPG method is applied on non-uniform meshes, the stabilization formula $\alpha_e = 1 - \frac{2k}{U_e h_e}$ is no longer valid, an then, an error equation can be computed. Using the residual free bubbles approach, Torsten derived an equation to compute the stabilization parameter in anisotropic medium, and applied it to adaptive meshes, getting a poor precision over a coarse mesh[8]. In contrast with it, on this procedure we compute the stabilization parameter $p^e(x)$ over a coarse mesh and scale it by the β parameter, which allows us to get a high precision over a coarse mesh. Next, we consider the error $\phi_{(\beta=1)} - \phi_{(\beta=0)}$ along the domain. It is clear that this error is derived by the anisotropy of mesh or the presence of non-homogeneities at the medium. Putting

$$\phi_{(\beta=1)} - \phi_{(\beta=0)} = \mathrm{p(x)}R_{(\beta=0)} = \sum_{\mathrm{e}} \mathrm{p^e(x)}R^e{}_{(\beta=0)} \tag{44}$$

and solving the problem (1)-(2) for $\phi_{(\beta=1)}$, appears the equation

$$\sum_e \int_{\widetilde{\Omega^e}} ((U^e\frac{d}{dx} - k\frac{d^2}{dx^2})(\hat{\phi}_{(\beta=0)} + R^e{}_{(\beta=0)}\mathrm{p^e}(x))\frac{h}{2}\frac{dN_i}{dx})d\Omega = \sum_e \int_{\widetilde{\Omega^e}} Q^e\frac{h}{2}\frac{dN_i}{dx}d\Omega \tag{45}$$

where U^e, Q^e, and $R^e{}_{(\beta=0)}$ are the average values of $U(x), Q(x)$, and $R_{(\beta=0)}$ at each element e, respectively. Taking the operator $\mathcal{L}^e = U^e\frac{d}{dx} - k\frac{d^2}{dx^2}$ over the interval $]a, b[$, the equation (45) takes the form

$$\sum_e \int_{\widetilde{\Omega^e}} (\mathcal{L}^e(\hat{\phi}_{(\beta=0)}) + R^e{}_{(\beta=0)}\mathcal{L}^e(\mathrm{p}^e(x)))\frac{h}{2}\frac{dN_i}{dx})d\Omega = \sum_e \int_{\widetilde{\Omega^e}} Q^e \frac{h}{2}\frac{dN_i}{dx} d\Omega \qquad (46)$$

and by consequence,

$$\mathcal{L}^e(\hat{\phi}_{(\beta=0)}) + R^e{}_{(\beta=0)}\mathcal{L}^e(\mathrm{p}^e(x)) = Q^e \quad \textit{for each element } e \qquad (47)$$

and in virtue of that $R^e{}_{(\beta=0)} = Q^e - \mathcal{L}^e(\hat{\phi}_{(\beta=0)})$, for $R^e{}_{(\beta=0)}$ non-identically null at the element e leads the equation

$$\mathcal{L}^e(\mathrm{p}^e(x)) = 1 \quad \textit{in each element } e \qquad (48)$$

to compute $\mathrm{p^e(x)}$ at element interior, being $\mathrm{p^e(x)} = 0$ for all x that belong to boundary of element e. Then, solving the constant coefficients equation (48), which have the explicit solution

$$\mathrm{p}^e(x) = \frac{x}{U^e} - (\frac{h^e}{U^e})\frac{\exp{(-\frac{U^e h^e}{k})} - \exp{(\frac{U^e}{k}(x-h^e))}}{(\exp{(-\frac{U^e h^e}{k})} - 1)} \qquad (49)$$

Also, by the error equation method **(EE)** we can verify that $p^e(x) \equiv 0$ for $R^e{}_{(\beta=0)} = 0$ with the machine precision.

5.2 Determining the β parameter

In this point, we compute the β parameter such that $\mathbf{R}(\alpha^{nh})$ matrix satisfies the equalities (30)(or (33)). Given $\mathrm{p^e(x)}$ in each element e, α_e^{nh} is approximated by

$$\alpha_e^{nh} = \alpha_e + \beta \mathrm{p}^e(\mathbf{P}) = \alpha_\mathrm{e} + 2\frac{\beta_\mathrm{g}}{\mathrm{h_e}}\mathrm{p}^e(\mathbf{P}) \qquad (50)$$

such that $\beta_g = \beta\frac{h_e}{2}$ and $\mathrm{p}^e(\mathbf{P})$, is an average value of $\mathrm{p^e(x)}$ at element e. Next, a β parameter will be computed solving the eigenvalue problem (33) in direct (or iterative) manner.

Direct form. Employing the direct calculus, we solve the eigenvalue problem (33) to element level. Substituting the element matrices $\mathbf{D}, \mathbf{P}$ and $\mathbf{K}(\beta)$ we have

$$\begin{bmatrix} -1 & 1 \\ -1 & 1 \end{bmatrix} = -\delta(1 + \beta\mathrm{p}^e(\mathbf{P}))\frac{\mathrm{sign(U^e)}}{2}\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \qquad (51)$$

Solving for β, the obtained value β_g is equal to

$$\beta_g = \frac{\max{[(2\delta^{-1}\mathrm{sign}(U^e) - 1), -(2\delta^{-1}\mathrm{sign}(U^e) + 1)]}\max h^e}{2 \mid \min_{\mathrm{p}^e(\mathbf{P})\neq 0} \mathrm{p}^e(\mathbf{P}) \mid} \qquad (52)$$

Iterative form. Employing an iterative procedure, we solve the eigenvalue problem (33) to global level. Computing the spectral radius $\rho(\mathbf{R}(\alpha^{nh}))$ the problem (33) can be solved in iterative manner by the solution of the following inequality:

$$\rho(\mathbf{R}(\alpha_k^{nh})) \leq \delta \quad k = 1, 2, \ldots \qquad (53)$$

This procedure, will allow to get a major precision at the computing process. Based on the iteration matrix $\mathbf{R}(\alpha^{nh})$ given by the formula (29) we will formulate the following propositions:

Proposition1 : The iteration matrix $\mathbf{R}(\alpha^{nh})$ has the following properties

- Is an invertible matrix
- Is a quasi-symmetric matrix
- For values $||\alpha^{nh}||$ sufficiently large, yields $||\mathbf{R}(\alpha^{nh})|| < 1$

Proposition 2: If $\alpha_e = 1 - \frac{1}{\mathbf{P}e}$ and there is an element, such that $(\frac{dx}{d\tau})^e \neq 0$, then $\hat{\phi}(\alpha^{nh})$ is a convergent succession in $||\alpha^{nh}||$ to ϕ in L_2 - norm.
Next, we construct a functions succession $\hat{\phi}(\alpha^{nh})$ by the equation

$$\hat{\phi}(\alpha^{nh}) = \mathbf{R}(\alpha^{nh})\hat{\phi}(\alpha^{nh}) + (\mathbf{I} - \mathbf{R}(\alpha^{nh}))\phi \tag{54}$$

where ϕ is the exact solution of equation (1). Also, equation (54) may be represented in the form

$$\hat{\phi}^{(1)} = \mathbf{R}(\alpha^{nh})\hat{\phi}^{(0)} + (\mathbf{I} - \mathbf{R}(\alpha^{nh}))\phi \tag{55}$$

The model equation (54) has been designed such that the exact solution, ϕ, is a fixed point of the iteration. This means that iteration does not change the exact solution:

$$\phi = \mathbf{R}(\alpha^{nh})\phi + (\mathbf{I} - \mathbf{R}(\alpha^{nh}))\phi \tag{56}$$

Subtracting these last two expressions, we find that

$$\mathbf{e}rr^{(1)} = \mathbf{R}(\alpha_0^{nh})\mathbf{e}rr^{(0)} \tag{57}$$

Repeating this argument, it follows that after m steps, the error in the mth approximation is given by

$$\mathbf{e}rr^{(m)} = \prod_{k=1}^{m}\mathbf{R}(\alpha_k^{nh})\mathbf{e}rr^{(0)} \tag{58}$$

If we now choose a particular vector norm and its associated matrix norm, it is possible to bound error after m iterations by

$$||\mathbf{e}rr^{(m)}|| \leq \prod_{k=1}^{m}||\mathbf{R}(\alpha_k^{nh})|| * ||\mathbf{e}rr^{(0)}|| \tag{59}$$

This leads us to conclude that if $||\mathbf{R}(\alpha_k^{nh})|| < 1$ for each $k = 1, 2, \ldots$, then the error is forced to zero as the iteration proceeds. Therefore, it follows that the iteration associated with the matrix $\mathbf{R}(\alpha_k^{nh})$ converges for all initial guess if and only if the spectral radius $\rho(\mathbf{R}(\alpha_k^{nh})) < 1$. The spectral radius $\rho(\mathbf{R}(\alpha_k^{nh}))$ is also called the asymptotic convergence factor when it appears in the context of iterative methods[9]. We have established the importance of the spectral radius of the iteration matrix $\mathbf{R}$ in analyzing the convergence properties of iterative methods governed by equation (54). The following scheme can be devised to obtain a stable numerical solution in an adaptive manner.

$\beta -$ **Algorithm** :

1. **Initialization**. Input $\hat{\alpha}_0$, $\beta_g^{(0)}$, $\delta > 0$, ϵ_1, ϵ_2, ϵ_3, $iter_{max}$, $\hat{\phi}_0$, set $k := 0$;
2. Form a local matrix $\mathbf{A}_l, l = 1, 2, \ldots,$ nel using the **SUPG** formulation.

3. To apply the $\alpha-$**algorithm to element level** with $\beta_g^{(0)} = 0$, until to get the optimal $\hat{\alpha} = [\alpha^{(1)}, \alpha^{(2)}, \ldots, \alpha^{(\mathrm{nel})}]^T$ parameter vector.

4. **Loop of calculus**. Do while ($||\hat{\phi}_k - \hat{\phi}_{k-1}||_2 > \epsilon_1$, and, $k \leq iter_{max}$), k = k + 1;

   - To solve the **forward** problem by **SUPG** formulation using a guess of the stabilization parameter $\hat{\alpha}_k^{nh} = \hat{\alpha}$.
   - Compute a global matrix $\mathbf{R}(\alpha_k^{nh}) = \mathbf{I} - [\mathbf{K}(\alpha^{nh})]^{-1}\mathbf{A}$ and its $\rho(\mathbf{R})$ spectral radius.
   - Do while $(\rho(\mathbf{R}(\alpha_k^{nh})) > \delta)$, solve δ-equation $f(\beta_g^{(k)}) = [\rho(\mathbf{R}(\alpha_k^{nh}))]^2 - \delta = 0$ for $\beta_g^{(k+1)}$ by **Quasi-Newton method.**
   - $\alpha_{(k+1)}^{(l)} = \alpha_k^{(l)} + \beta_g^{(k+1)}[\mathrm{p^e}(\mathbf{P})], \mathrm{l} = 1, 2, \ldots, \mathrm{nel}$.

5. **Convergence Tests**. Check the following inequality

$$|\beta_g^{(k)} - \beta_g^{(k-1)}| < \epsilon_2 \quad \text{or} \tag{60}$$

$$|\beta_g^{(k)} - \beta_g^{(k-1)}| > \epsilon_2 \quad \text{and} \quad ||\hat{\phi}_k - \hat{\phi}_{k-1}||_2 \times |\beta_g^{(k)} - \beta_g^{(k-1)}| < \epsilon_3 \tag{61}$$

6. **Stop rule**. If the condition ((60) - (61)) are satisfied, or $k == iter_{max}$, exit algorithm; otherwise, goto step 4.

6 NUMERICAL EXPERIMENTS

We now present some numerical experiments to show the effectiveness of the procedure developed above to achieve the numerical stabilization of SUPG method when solving the convection-diffusion equation (1) subject to Dirichlet boundary conditions (2), on uniform or non-uniform meshes. In our implementation, we use piecewise-linear finite element method to solve the problem numerically, and iterative or direct procedures to compute the stabilization parameter on the same mesh. The discrete systems were solved using MATLAB 5.2.

When using $\alpha-$**Algorithm** to compute the stabilization parameter on an uniform mesh the following parameters are chosen: a discrepancy level $\delta = 1e-03$, a tolerance level $\epsilon_2 = 1e-08$, and the error levels $\epsilon_1 = 1e-12$, $\epsilon_3 = 1e-08$. On this process, the initial guess $\hat{\alpha}_0 = 0.5$ is chosen to apply the $N1, QN0, QN1$ methods, while $\hat{\alpha}_0 = 0$ can be employed to initialize the computation in the other cases. Starting from the initial guess, the convergence is achieved at the parameter's space within the interval $[0, 1]$.

Many numerical experiments show the convergence phenomena through the tables presented below. On these tables, α^* stands for the optimal value α value which achieves the minimum for $||f(\alpha)||_2$, the norms $||\hat{\alpha}_k - \hat{\alpha}_{k-1}||_2$, $||\hat{\phi}_k - \hat{\phi}_{k-1}||_2$ are the residuals in the $L_2 - norm$ at the parameter's space and solution's space, respectively, and Iter is the iteration number. A discrepancy between the numerical solution ϕ and the exact solution of the direct problem is expressed by the residual $||\hat{\phi}_k - \phi_{exact}||_2$. A measure of the $\delta-$self-adjoint matrix is expressed at the column $(\frac{\sqrt{||f(\alpha^*)||}}{\mathbf{Pe}^{(1-\gamma)}})$. This property of the solutions is essential when solving convection-diffusion problems with dominant convection.

Example 1

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

Since the coefficients are constants we have a purely convection dominated problem for $k \to 0$. In this case it is no surprise that the numerical algorithms show robustness for $\mathbf{Pe} \to \infty$, as shown in Tables 1 - 3. We can notice that the optimal values α^* obtained for each $\mathbf{Pe}$ value coincides with the standard critical value used in the practice. It can be checked that the agreement with the exact solution ϕ_{exact} improves to nine decimal figures as the $\mathbf{Pe}$ values increase to 10^{10}. This result shows that the resulting FEM solution is exact at each node.

Algorithms	Iter	α^*	$\|\|\hat{\alpha}_k - \hat{\alpha}_{k-1}\|\|$	$\|\|\hat{\phi}_k - \hat{\phi}_{k-1}\|\|$	$\|\|\hat{\phi}_k - \phi_{exact}\|\|$	$\frac{\sqrt{\|\|f(\hat{\alpha}_*)\|\|}}{\mathbf{Pe}^{(1-\gamma)}}$
$N0$	10	0.8000	2.9337e-10	4.6386e-11	4.5400e-5	8.2161e-5
$N1$	45	0.8000	7.9859e-09	1.2627e-09	4.5403e-5	2.0156e-4
Bis	26	0.7812	5.8902e-09	9.4920e-10	9.5095e-3	3.6661e-1
$QN0$	44	0.8000	7.3652e-09	1.1645e-09	4.5402e-5	1.9255e-4
$QN1$	24	0.8000	8.2923e-09	1.3111e-09	4.5401e-5	1.3775e-4
$Bis + QN0$	45	0.8000	9.0036e-09	1.4236e-09	4.5403e-5	2.1768e-4
$Bis + QN1$	23	0.8000	6.6383e-09	1.0496e-09	4.5401e-5	1.2325e-4

Table 1: The comparison of efficiency for seven algorithms for δ-equation (19) with $\mathbf{Pe} = 5$

Algorithms	Iter	α^*	$\|\|\hat{\alpha}_k - \hat{\alpha}_{k-1}\|\|$	$\|\|\hat{\phi}_k - \hat{\phi}_{k-1}\|\|$	$\|\|\hat{\phi}_k - \phi_{exact}\|\|$	$\frac{\sqrt{\|\|f(\hat{\alpha}_*)\|\|}}{\mathbf{Pe}^{(1-\gamma)}}$
$N0$	10	0.9999	8.3139e-09	1.3145e-09	1.3878e-10	4.4816e-05
$N1$	133	0.9999	8.8990e-09	1.4071e-09	1.1030e-08	3.9954e-04
Bis	26	0.9999	5.8902e-09	9.3133e-10	3.4067e-06	7.0215e-03
$QN0$	28	0.9999	5.1678e-09	8.1710e-10	8.1759e-10	1.0878e-04
$QN1$	120	0.9999	8.9724e-09	1.4187e-09	9.9205e-09	3.7891e-04
$Bis + QN0$	45	0.9999	8.5631e-09	1.3540e-09	4.0604e-09	2.4241e-04
$Bis + QN1$	14	0.9999	2.1175e-10	3.3480e-11	2.6830e-14	6.2302e-07

Table 2: The comparison of efficiency for seven algorithms for δ-equation (19) with $\mathbf{Pe} = 10^4$

In the following, we compare the performance of Quasi-Newton's methods with the $N0$ method. In general, the Quasi-Newton's methods are more cheap that Newton's methods. Also, we can appreciate that the $QN1$ and $N1$ methods are very expensive, and further dependent from the $\mathbf{Pe}$ values. In contrast with it, the iteration number using the $QN0$ method, is reduced drastically as the $\mathbf{Pe}$ values increase to 10^{10}. This method shows a good convergence asymptotically and the numerical solution can be achieved within 28 iterations with a precision about $8.8132e - 10$. A cheap method can be derived using the Hybrid method. We can appreciate that the $Bis + QN1$ method reflects a better convergence than ones in all range of $\mathbf{Pe}$ values.

Algorithms	Iter	α^*	$\|\|\hat{\alpha}_k - \hat{\alpha}_{k-1}\|\|$	$\|\|\hat{\phi}_k - \hat{\phi}_{k-1}\|\|$	$\|\|\hat{\phi}_k - \phi_{exact}\|\|$	$\frac{\sqrt{\|\|f(\hat{\alpha}_*)\|\|}}{\mathbf{Pe}^{(1-\gamma)}}$
$N0$	10	1.0000	8.3184e-09	1.3153e-09	1.3885e-10	4.4828e-05
$N1$	133	1.0000	8.9858e-09	1.4208e-09	1.1145e-08	4.0162e-04
Bis	28	1.0000	5.8902e-09	9.3132e-10	8.8132e-10	1.1294e-04
$QN0$	28	1.0000	5.1307e-09	8.1123e-10	8.1124e-10	1.0835e-04
$QN1$	120	1.0000	9.1058e-09	1.4398e-09	1.0078e-08	3.8191e-04
$Bis + QN0$	28	1.0000	5.8902e-09	9.3132e-10	8.8132e-10	1.1294e-04
$Bis + QN1$	28	1.0000	5.8902e-09	9.3132e-10	8.8132e-10	1.1294e-04

Table 3: The comparison of efficiency for seven algorithms for δ-equation (19) with $\mathbf{Pe} = 10^{10}$

An detailed study of the convergence phenomena as function of **Pe** values and tolerance level ϵ_2 is revealed through from Tables 4 - 7. Convergence achieved with Quasi-Newton's algorithms are shown in Tables 4 and 6, while the convergence properties from Hybrid methods are shown in Tables 5-7.

We can observe in Tables 4 - 7 that the $QN0$ and $Bis + QN1$ methods give good approximations to the optimal α^* values for the considered example. In particular, for low **Pe** values all methods can be used without loss precision, except the $QN1$ method in case of a tolerance error $\epsilon_2 = 10^{-7}$. In the intermediate range of **Pe** values all method can be used with a tolerance error ϵ_2 lower than 10^{-5}, and the $QN0$ method is most appropriate to high accuracy. For the asymptotic case, that is, in the diffusive limit, the Hybrid methods have the best behavior. Convergence with these methods is achieved within 25 iterations starting from $\hat{\alpha}_0 = 0$ for a tolerance error $\epsilon_2 = 10^{-7}$ in L_2-norm. It confirm that when we combine the advantages of the Bisection and Quasi-Newton algorithms, we can speed up the whole iterative process.

	Error level ($\epsilon_2 =$)		1e-02		1e-03		1e-04
Pe	Algorithms	Iter	$\|\|\hat{\alpha}_k - \hat{\alpha}_{k-1}\|\|$	Iter	$\|\|\hat{\alpha}_k - \hat{\alpha}_{k-1}\|\|$	Iter	$\|\|\hat{\alpha}_k - \hat{\alpha}_{k-1}\|\|$
5	$QN0$	8	8.0116e-03	13	9.2670e-04	17	7.0775e-05
5	$QN1$	5	5.7529e-03	8	3.1729e-04	10	4.2569e-05
10	$QN0$	9	9.4569e-03	13	9.4221e-04	16	6.8216e-05
10	$QN1$	10	8.3737e-03	17	8.1420e-04	22	5.9606e-05
10^2	$QN0$	8	9.2941e-03	11	7.2397e-04	14	2.7422e-05
10^2	$QN1$	18	9.3859e-03	34	7.6239e-04	45	7.9254e-05
10^{10}	$QN0$	8	7.7465e-03	11	6.1634e-04	13	4.2231e-05
10^{10}	$QN1$	19	9.8312e-03	37	7.8439e-04	49	8.0732e-05

Table 4: Convergence of the Quasi-Netown algorithms with $\epsilon_3 = 1e - 08$ for several error levels ϵ_2

	Error level ($\epsilon_2 =$)	1e-02		1e-03		1e-04	
Pe	Algorithms	Iter	$\|\|\hat{\alpha}_k - \hat{\alpha}_{k-1}\|\|$	Iter	$\|\|\hat{\alpha}_k - \hat{\alpha}_{k-1}\|\|$	Iter	$\|\|\hat{\alpha}_k - \hat{\alpha}_{k-1}\|\|$
5	$Bis + QN0$	8	8.2281e-03	14	7.3788e-04	18	9.7655e-05
5	$Bis + QN1$	6	4.7965e-03	8	3.4041e-04	10	4.8751e-05
10	$Bis + QN0$	10	9.3602e-03	15	8.5570e-04	19	5.5117e-05
10	$Bis + QN1$	9	8.0241e-03	14	5.9212e-04	16	8.2180e-05
10^2	$Bis + QN0$	10	9.3602e-03	15	8.5570e-04	19	5.5117e-05
10^2	$Bis + QN1$	9	8.0241e-03	14	5.9212e-04	16	8.2180e-05
10^{10}	$Bis + QN0$	8	8.7346e-03	12	5.4592e-04	15	6.8239e-05
10^{10}	$Bis + QN1$	8	8.7346e-03	12	5.4592e-04	15	6.8239e-05

Table 5: Convergence of the Hybrid algorithms with $\epsilon_3 = 1e-08$ for several error levels ϵ_2

	Error level ($\epsilon_2 =$)	1e-05		1e-06		1e-07	
Pe	Algorithms	Iter	$\|\|\hat{\alpha}_k - \hat{\alpha}_{k-1}\|\|$	Iter	$\|\|\hat{\alpha}_k - \hat{\alpha}_{k-1}\|\|$	Iter	$\|\|\hat{\alpha}_k - \hat{\alpha}_{k-1}\|\|$
5	$QN0$	21	4.0763e-06	23	3.5425e-07	25	4.4997e-08
5	$QN1$	12	4.1254e-07	12	4.1254e-07	244	0.0000e+00
10	$QN0$	18	6.5766e-06	21	3.0073e-08	21	3.0073e-08
10	$QN1$	25	9.3763e-06	28	9.0955e-07	31	6.8734e-08
10^2	$QN0$	15	6.1725e-06	16	3.7363e-09	16	3.7363e-09
10^2	$QN1$	55	7.2113e-06	63	9.2989e-07	73	2.5898e-08
10^{10}	$QN0$	14	8.1721e-06	16	1.3999e-07	785	1.7664e-08
10^{10}	$QN1$	62	7.1804e-06	71	5.9535e-07	78	9.3036e-08

Table 6: Convergence of the Quasi-Netown algorithms with $\epsilon_3 = 1e-08$ for several error levels ϵ_2

	Error level ($\epsilon_2 =$)	1e-05		1e-06		1e-07	
Pe	Algorithms	Iter	$\|\|\hat{\alpha}_k - \hat{\alpha}_{k-1}\|\|$	Iter	$\|\|\hat{\alpha}_k - \hat{\alpha}_{k-1}\|\|$	Iter	$\|\|\hat{\alpha}_k - \hat{\alpha}_{k-1}\|\|$
5	$Bis + QN0$	23	3.5219e-06	25	9.4941e-07	27	7.3244e-08
5	$Bis + QN1$	11	8.1494e-06	12	1.3931e-07	13	4.4556e-08
10	$Bis + QN0$	22	5.2712e-06	24	6.7412e-07	25	8.2056e-08
10	$Bis + QN1$	20	1.6121e-06	21	3.2431e-07	23	4.3060e-08
10^2	$Bis + QN0$	25	7.5701e-06	29	6.7327e-07	31	5.6219e-08
10^2	$Bis + QN1$	88	2.1299e-06	100	1.0508e-04	100	1.0508e-04
10^{10}	$Bis + QN0$	18	8.5299e-06	22	5.3312e-07	25	6.6640e-08
10^{10}	$Bis + QN1$	18	8.5299e-06	22	5.3312e-07	25	6.6640e-08

Table 7: Convergence of the Hybrid algorithms with $\epsilon_3 = 1e-08$ for several error levels ϵ_2

Figure 1 shows the numerical results obtained applying Hybrid methods at the parameter space and two node linear elements at the solution space. First, we study the convergence of the stabilization parameter $\hat{\alpha}_k, k = 1, 2, \ldots$, using $Bis + QN0$ algorithm. Figure 1(left) gives the number of iterations of the Hybrid method with an initial guess $\alpha = 0$ when it converges to the critical value $\alpha^* = 1$. On the low part of this figure the corresponding L_2-norm distance between $\hat{\alpha}_k$ values and the critical value α_{exact} as function of iteration numbers is shown. As we can appreciate the line decreases monotonically until to save the desirable accuracy. In Figure 1, we have plotted the Galerkin, the exact solution and the approximate solutions $\hat{\phi}_k, k = 1, 2, \ldots$, corresponding to $\mathbf{Pe} = 10^{10}$. We start with no diffusivity (that is the Galerkin method) and within about 20 iterations converge to $SUPG$ method, with a tolerance level $\epsilon_2 = 10^{-5}$. The agreement with the exact solution is excellent.

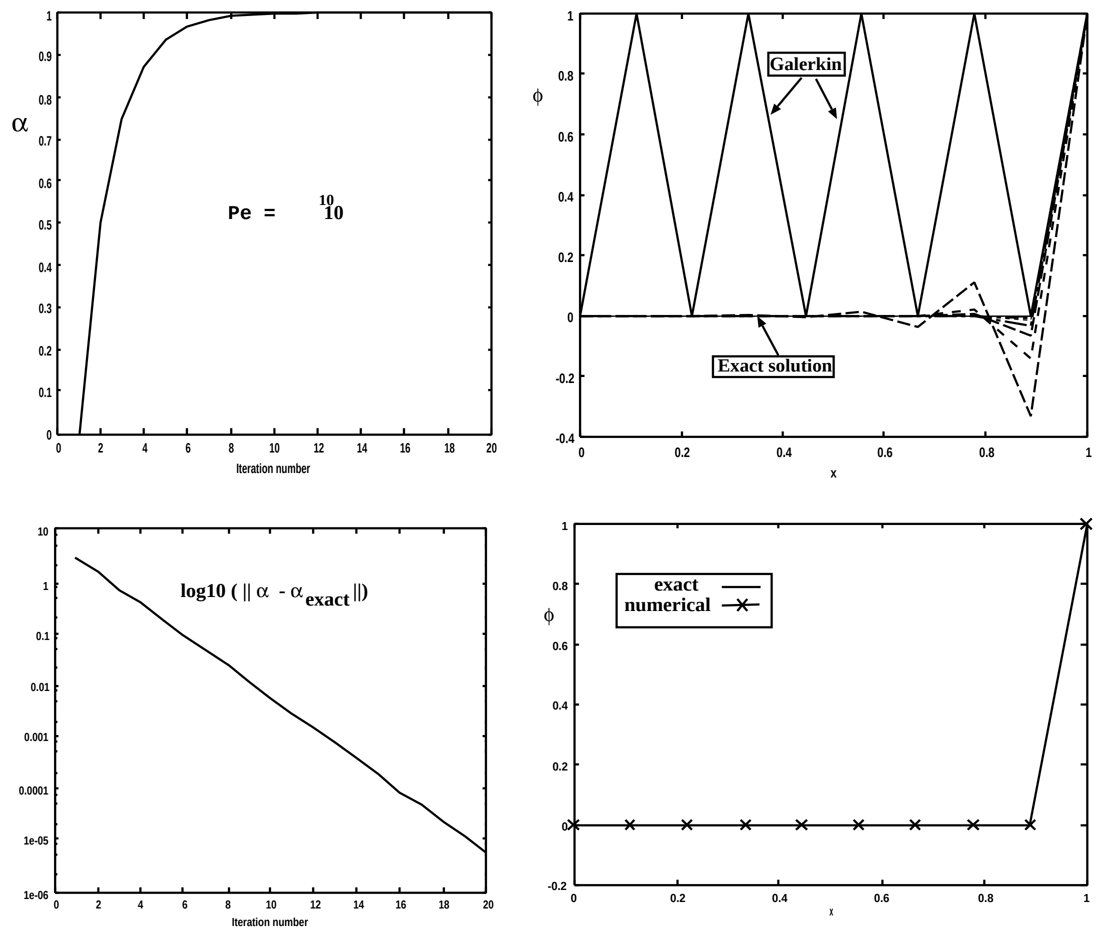


Figure 1: Solution of one-dimensional convection-diffusion problem with $U = 1$, $Q = 0$ and $\phi(0) = 0$, $\phi(1) = 1$. (left) Convergence of the $Bis{+}QN0$ algorithm for $\mathbf{Pe} = 10^{10}$ and tolerance level $\epsilon_2 = 1e - 05$ starting from $\hat{\alpha}_0 = 0$. (right) Numerical solutions obtained with 9 two node linear elements.

Next, we analyze the performance of the proposed procedures in the section 5.2 to compute the stabilization parameter in cases of velocity flux and /or source term are spatially variable. In such cases is recommended to use a non-uniform mesh over the domain. In this way, we select three family of real transformations on the interval $[a, b]$ that characterize the non-uniformity of mesh at hand. These transformations are writing now

$$x = ((b^p - a^p) * \xi + a^p)^{(1/p)} \quad with\ \xi \in [0,1],\ p \in \mathcal{R},\ b > a,\ p > 0 \tag{62}$$

$$x = a + exp(q * \xi) - (exp(q) - 2) * \xi - 1 \quad with\ q \in \mathcal{R},\ \xi \in]0,1[\tag{63}$$

$$x = \xi - s * (\xi - x_0)^q \quad with\ q \in \mathcal{R},\ s \in \mathcal{R},\ x_0,\ \xi \in]0,1[\tag{64}$$

and they are denoted by $\mathbf{f1}, \mathbf{f2}$, and **f3**, respectively.

Numerical results obtained using $\mathbf{f1}, \mathbf{f2}$, and /or **f3** transformations will be display at the tables of this section. On these tables β_g^* stands for the optimal value β_g which achieves the minimum for $||\mathbf{R}(\alpha_k^{nh})||_2$, the norms $||\hat{\phi}_k - \phi_{exact}||_2$ are the residuals in the $L_2 - norm$ at solution's space, and, it express the discrepancy between the numerical solution $\hat{\phi}$ and the exact solution ϕ. Iter is the iteration number and $\rho(\mathbf{R})$ the spectral radius. Also, $\alpha-$Iter and $\beta-$Iter denote the iteration numbers developed by $\alpha-$algorithm and $\beta-$algorithm, respectively. A measure of the $\delta-$self adjoint matrix is expressed at the column ($\rho(\mathbf{R})$). This property of the solutions is essential when solving convection-diffusion problems with dominant convection on non-uniform mesh.

Example 2

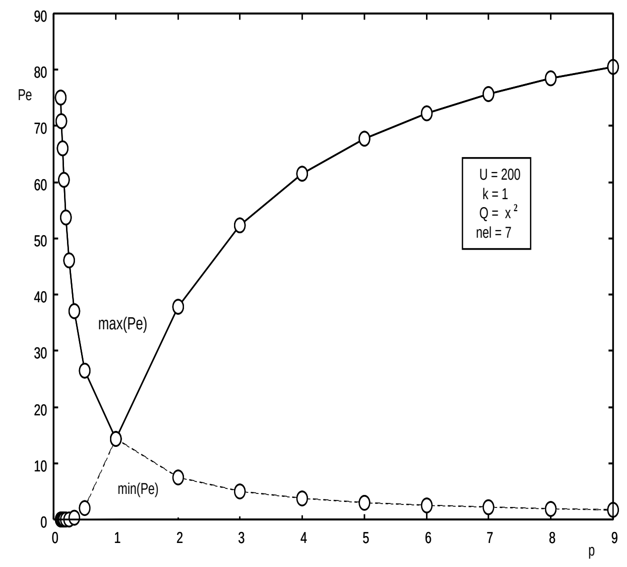
$$200\frac{d\phi}{dx} - \frac{d^2\phi}{dx^2} = x^2 \quad in\ \Omega = (0,1),\ \phi(0) = 0,\ \phi(1) = 0$$

Excellent accuracy in the numerical results on an uniform mesh were obtained using the $\alpha-$Algorithm. Next,using $\beta-$**Algorithm** we analyze of accuracy of computed solution on a non-uniform mesh. In this case we take the maximum **Pe** value (max(**Pe**)) and the minimum **Pe** value (min(**Pe**)) as reference numbers.

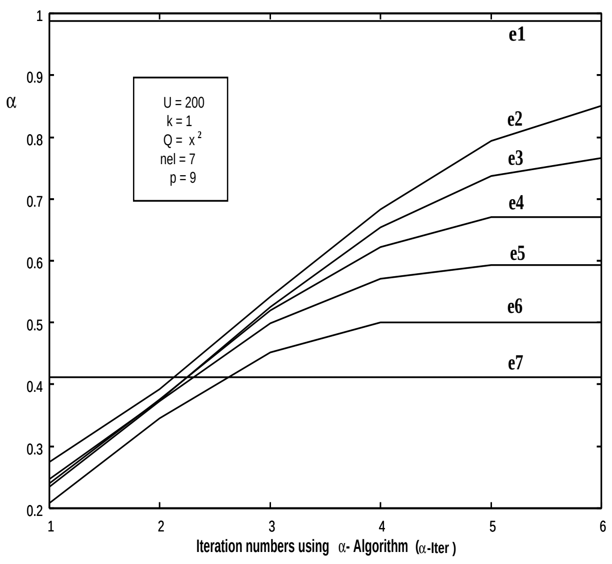
Figure 2(a) shows the maximum(minimum) values of **Pe** over meshes built by **f1** transformation for several p parameter values. It is clear that for p = 1 is obtained an uniform mesh. Taking nel = 7 linear elements, the biggest variations of **Pe** numbers are observed for small values of p , $(0 < p << 1)$ and p values close to 9.

First steps at the $\beta-$**Algorithm** are based on the application of $\alpha-$**-Algorithm** to element level until to get the optimal value $\bar{\alpha}_e$. Figure 2(b) reflects the convergence process to element level according to the **Pe** values using the **f1** transformation with p = 9. On this figure, each element has been denoted by el, where l indicates the order of element at the partition. On this process, we assume that the boundary elements have a maximum diffusivity due to the presence of Dirichlet boundary conditions at the continuous problem(1)-(2). We can appreciate that the convergence is achieved within of 6 iterations almost every elements.

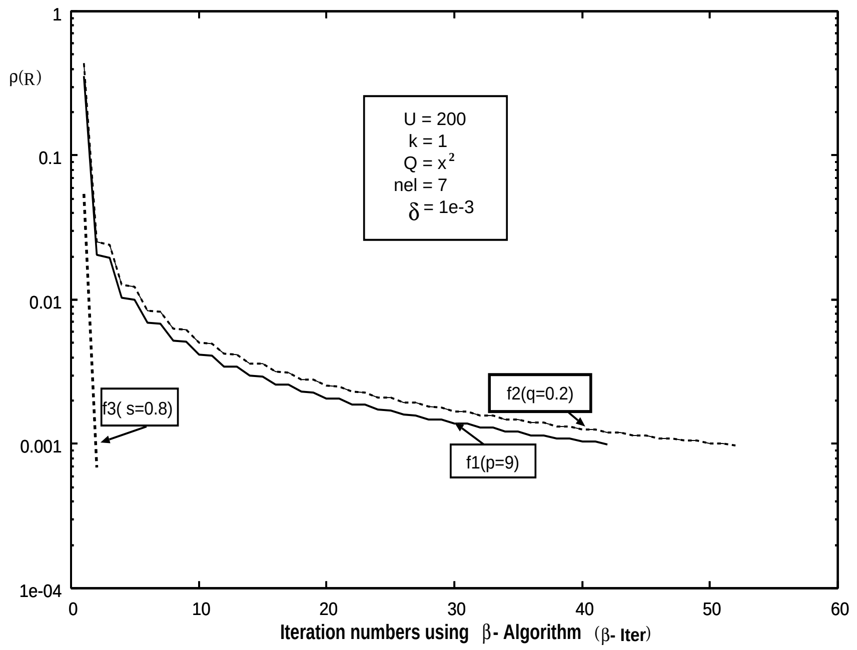
On the next step, the $\beta-$Algorithm is initialized with these values, that is, $\beta_g = 0$, as same as, $\alpha_{(k=0)}^{nh} = \bar{\alpha}_e$. Figure 2(c) shows the convergence process of the $\beta-$**Algorithm** for three different meshes, according to the chosen parameter values. With a tolerance of $\delta = 1e - 03$ at the parameter's space, the convergence is reached in all cases, and, it is significant that in the case of **f3**(s = 0.8) the convergence is achieved within 10 iterations. This results can be influenced by presence of many interior nodes at the boundary layer, located at right part of interval $[a, b]$. To know the influence of these results in the solution's space a detailed analysis is developed in advance.



(a)



(b)



(c)

Figure 2: Solution of one-dimensional convection-diffusion problem with 7 linear elements, $U = 200,\ Q = x^2,\ k = 1$ and $\phi(0) = 0,\ \phi(1) = 0$. (a) Extremum **Pe** values for several p using the mesh **f1**. (b) Convergence of the $\alpha-$algorithm for each element versus Iteration numbers ($\alpha-$Iter).(c) Convergence history using three different non-uniform meshes.

			Error level	$(\delta = 1e-03)$		
p	$\alpha-$Iter	$\beta-$Iter	β_g^*	$\rho(\mathbf{R})$	$\|\|\hat{\phi}_k - \phi_{exact}\|\|$	$\frac{\|\|\hat{\phi}_k - \phi_{exact}\|\|}{\|\|\phi_{exact}\|\|}$
1	3	1	9.3000e-01	2.0515e+00	1.8619e-05	1.4504e-02
2	7	1372	2.0247e+03	9.9979e-04	3.4385e-05	1.7872e-02
3	8	1204	1.3087e+03	9.9958e-04	3.9896e-05	1.7263e-02
4	8	1147	1.0821e+03	9.9962e-04	4.3121e-05	1.6715e-02
5	9	1133	9.7104e+02	9.9970e-04	4.4297e-05	1.5945e-02
6	9	1142	9.0457e+02	9.9995e-04	4.3798e-05	1.4952e-02
7	9	1166	8.6147e+02	9.9999e-04	4.3365e-05	1.4231e-02
8	8	1202	8.3134e+02	9.9957e-04	4.6316e-05	1.4749e-02
9	8	1247	8.0901e+02	9.9970e-04	5.5593e-05	1.7293e-02

Table 8: Convergence of $\beta-$algorithm using the mesh (62) for several p values, with error level $\delta = 1e-03$,$k = 1$, and nel = 7

			Error level	$(\delta = 1e-04)$		
p	$\alpha-$Iter	$\beta-$Iter	β_g^*	$\rho(\mathbf{R})$	$\|\|\hat{\phi}_k - \phi_{exact}\|\|$	$\frac{\|\|\hat{\phi}_k - \phi_{exact}\|\|}{\|\|\phi_{exact}\|\|}$
1	5	1	9.9300e-01	2.0759e+00	2.2690e-05	1.7983e-02
2	11	13148	2.0257e+04	9.9989e-05	3.4023e-06	1.7971e-03
3	13	11091	1.3091e+04	9.9998e-05	3.9518e-06	1.7369e-03
4	14	10156	1.0814e+04	9.9987e-05	4.3129e-06	1.6973e-03
5	14	9636	9.7053e+03	9.9990e-05	4.5690e-06	1.6684e-03
6	14	9312	9.0509e+03	9.9995e-05	4.7599e-06	1.6462e-03
7	14	9096	8.6201e+03	9.9994e-05	4.9078e-06	1.6286e-03
8	14	8946	8.3158e+03	9.9984e-05	5.0256e-06	1.6143e-03
9	14	8837	8.0884e+03	9.9997e-05	5.1217e-06	1.6025e-03

Table 9: Convergence of $\beta-$algorithm using the mesh (62) for several p values, with error levels $\delta = 1e-04$, $k = 1e-1$, and nel = 7

Tables 8-9 examine the influence of the non-uniformity of **f1** mesh in the accuracy of the computed solution for a tolerance $\delta = 1e-03$ and $\delta = 1e-04$, respectively. We can appreciate that the convergence is reached at the parameter's space in all cases examined with similar computational cost. A comparison between two tables reflects that the performance of $\beta-$**Algorithm** is related to the **Pe** values. If **Pe** values increase the accuracy at the numerical solution increases also in both space, at the parameter's space and at the solution's space. Also, we can observe that if the non-uniformity of the mesh increases the precision of numerical solution decreases not significantly. In virtue of this, a further analysis relative to the convergence of $\beta-$**Algorithm** will be done for the worst case: p = 9.

We can appreciate in Tables 8-9 that the $\beta-$**Algorithm** seems very expensive. We find out that the iteration number $\beta-$Iter is highly dependent on scales of **Pe** values, and then, it is possible to reduce drastically the iteration number without affect the precision. Tables 10-11 show that the reduction factor can be significant, and that the precision increases as the diffusive coefficient k diminishes. These results are very important to the feasibility of the proposed method.

			Error level	($\delta = 1e-03$)	(p=9)	
t	$\alpha-$Iter	$\beta-$Iter	$\beta_g^*/\mathbf{Pe}^{\mathrm{t}}$	$\rho(\mathbf{R})$	$\|\|\hat{\phi}_k - \phi_{exact}\|\|$	$\frac{\|\|\hat{\phi}_k-\phi_{exact}\|\|}{\|\|\phi_{exact}\|\|}$
0	6	1246	8.0834e+02	9.9992e-04	5.5593e-05	1.7293e-02
1	6	657	2.5444e+02	9.9939e-04	5.5778e-05	1.7351e-02
2	6	362	1.0578e+02	9.9798e-04	5.5786e-05	1.7353e-02
3	6	184	4.5577e+01	9.9968e-04	5.5787e-05	1.7354e-02
4	6	90	2.0367e+01	9.8449e-04	5.5787e-05	1.7354e-02
5	6	42	9.0701e+00	9.8651e-04	5.5788e-05	1.7354e-02

Table 10: Acceleration of convergence process of $\beta-$algorithm for several β_g scales, using the mesh (62) for p = 9 value, with error levels $\delta = 1e-03$, $k = 1$, and nel = 7

			Error level	($\delta = 1e-04$)	(p = 9)	
t	$\alpha-$Iter	$\beta-$Iter	$\beta_g^*/\mathbf{Pe}^{\mathrm{t}}$	$\rho(\mathbf{R})$	$\|\|\hat{\phi}_k - \phi_{exact}\|\|$	$\frac{\|\|\hat{\phi}_k-\phi_{exact}\|\|}{\|\|\phi_{exact}\|\|}$
0	14	8837	8.0884e+03	9.9997e-05	5.1217e-06	1.6025e-03
1	14	520	2.5415e+02	9.9959e-05	5.1554e-06	1.6130e-03
2	14	24	1.1324e+01	9.3374e-05	5.1568e-06	1.6135e-03
3	14	2	9.5120e-01	4.8441e-05	5.1569e-06	1.6135e-03
4	14	2	9.5112e-01	2.1314e-06	5.1569e-06	1.6135e-03
5	14	1	9.4111e-01	8.9452e-06	5.1569e-06	1.6135e-03

Table 11: Acceleration of convergence process of $\beta-$algorithm for several β_g scales, using the mesh (62) for p = 9 value, with error levels $\delta = 1e-04$, $k = 1e-1$, and nel = 7

			Error level	($\delta = 1e-04$)	(p = 9)	
nel	$\alpha-$Iter	$\beta-$Iter	$\beta_g^*/\mathbf{Pe}^4$	$\rho(\mathbf{R})$	$\|\|\hat{\phi}_k - \phi_{exact}\|\|$	$\frac{\|\|\hat{\phi}_k-\phi_{exact}\|\|}{\|\|\phi_{exact}\|\|}$
7	13	2	9.2589e-01	9.1216e-06	7.3671e-06	2.3034e-03
14	15	6	2.7488e+00	8.3496e-06	5.3940e-06	1.1518e-03
28	16	10	4.5674e+00	8.7835e-06	3.8748e-06	5.7578e-04
56	17	14	6.3854e+00	9.0571e-06	2.7603e-06	2.8783e-04
112	19	18	8.2038e+00	9.1860e-06	1.9588e-06	1.4389e-04

Table 12: Convergence process of $\beta-$algorithm relative to element number nel using the mesh (62) for p = 9 value, with error levels $\delta = 1e-04$, and $k = 1/\mathrm{nel}$

			Error level	$(\delta = 1e-04)$	(p=1/3)	
nel	$\alpha-$Iter	$\beta-$Iter	$\beta_g^*/\mathbf{Pe}^3$	$\rho(\mathbf{R})$	$\lVert\hat{\phi}_k - \phi_{exact}\rVert$	$\frac{\lVert\hat{\phi}_k-\phi_{exact}\rVert}{\lVert\phi_{exact}\rVert}$
7	13	2	9.2589e-01	9.1216e-06	7.3671e-06	2.3034e-03
14	15	6	2.7488e+00	8.3496e-06	5.3940e-06	1.1518e-03
28	16	10	4.5674e+00	8.7835e-06	3.8748e-06	5.7578e-04
56	17	14	6.3854e+00	9.0571e-06	2.7603e-06	2.8783e-04
112	19	18	8.2038e+00	9.1860e-06	1.9588e-06	1.4389e-04

Table 13: Convergence process of $\beta-$algorithm relative to element number nel using the mesh (62) for p = 1/3 value, with error levels $\delta = 1e-04$, and $k = 1/$nel

Another important aspect at the convergence process at the discrete medium is the influence of the element number of the mesh (nel) at this process. In this way, we select the values p = 9, and p = 1/3, with a tolerance $\delta = 1e-04$ at the parameter's space. We take $k = 1/$nel to guarantees that the $\mathbf{Pe}$ values are similar in each mesh for the prefixed p values. We can appreciate in the Tables 12-13 that the convergence is achieved within 20 iterations in all cases, although, the mesh in question is four times more fine that the initial mesh. With the increment of the element number the absolute residuals at the solution's space decreases with the same order, and the relative error decreases until one magnitude.

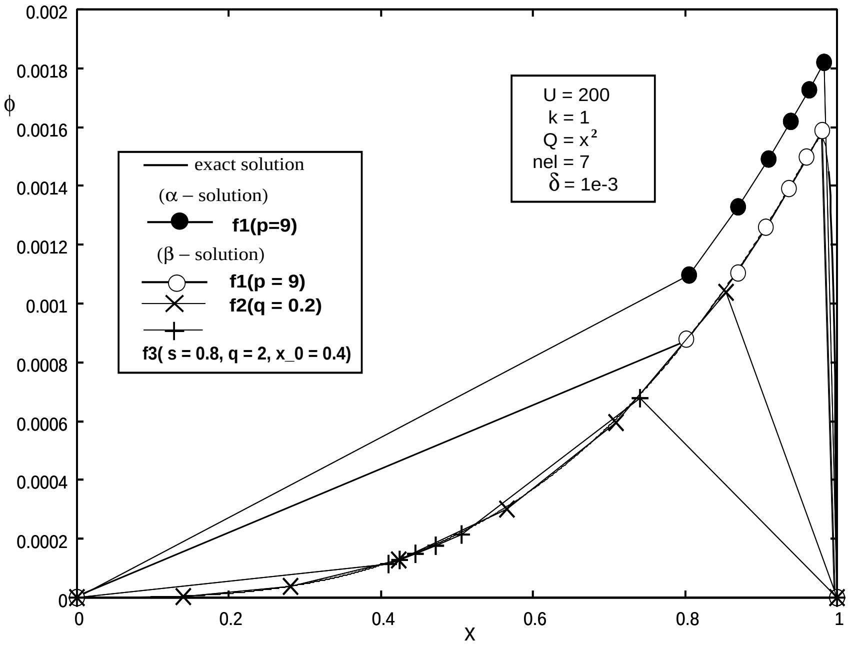


Figure 3: Solution of one-dimensional convection-diffusion problem with 7 linear elements, $U = 200$, $Q = x^2$, $k = 1$ and $\phi(0) = 0$, $\phi(1) = 0$. Comparison of numerical solutions obtained with $\beta-$ algorithm on three non-uniform meshes

In the following we select the worst case in each family given in (62)-(64) with prefixed parameter values. A comparison of the precision of numerical solutions is shown through the figures 3 and 4. Figure 3 shows a comparison between the numerical solutions derived with various non-uniform meshes and the exact solution. Numerical solutions have been obtained applying the $\beta-$**Algorithm** with a tolerance of $\delta = 1e-03$. The agreement with the exact solution is excellent. A behavior of residuals along the domain is shown in the figure 4. We can appreciate that the nodal residual increases along the interior region of domain, reaching its maximum value at the last interior node. Also, we can observe that this maximum absolute error increases as the last interior node is located very near to extreme right of the domain, however, the maximum absolute error is lower than $3.5e-05$.

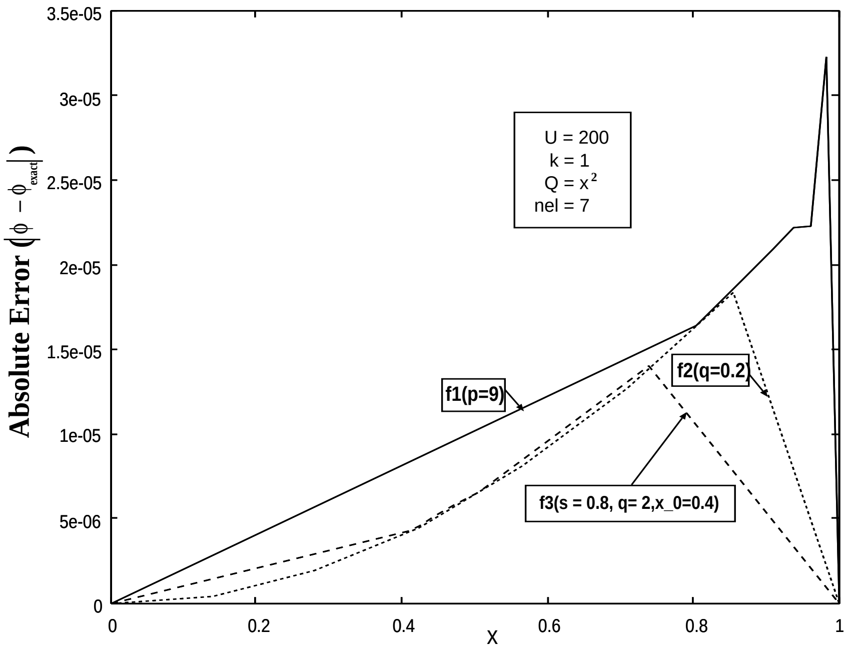
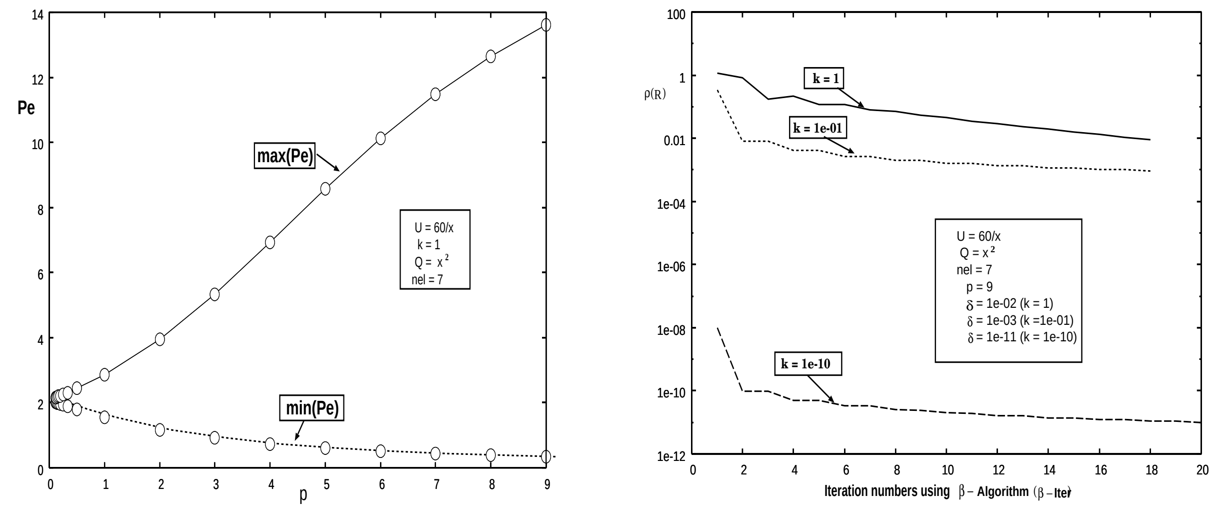


Figure 4: Solution of one-dimensional convection-diffusion problem with 7 linear elements, $U = 200,\ Q = x^2,\ k = 1$ and $\phi(0) = 0,\ \phi(1) = 0$. Comparison of absolute error between $\beta-$ numerical solutions and the exact solution vs domain.

Example 3

$$\frac{60}{x}\frac{d\phi}{dx} - k\frac{d^2\phi}{dx^2} = x^2 \quad in\ \Omega = (1,2),\ \phi(1) = 1,\ \phi(2) = 0$$

A detailed analysis of the convergence process using the $\beta-$**Algorithm** will be done applying the linear finite elements on the interval $[1, 2]$ using the transformation **f1**. This transformation allows to get non-uniform meshes depending of the parameter value p.



(a) (b)

Figure 5: Solution of one-dimensional convection-diffusion problem with 7 linear elements, $U = 60/x,\ Q = x^2$ and $\phi(1) = 1,\ \phi(2) = 0$. (a) Extremum **Pe** values for several p using the mesh **f1**. (b) Convergence Process of spectral radius $\rho(\mathbf{R})$ by $\beta-$algorithm for different diffusivity values $k = 1,\ 1e - 01,\ 1e - 10$ using the **f1** mesh with p = 9.

Figure 5(a) shows the maximum and minimum values of **Pe** over meshes built by **f1** transformation for several p parameter values. It is clear that for p = 1 is obtained an uniform mesh. Taking nel = 7 linear elements, the variations of **Pe** numbers increase as p values increase towards 9, and therefore the quality of mesh is deteriorated as p values increase to 9, where the aspect rate of mesh is equal to 18.092 and different to the aspect rate of **Pe** values, which is equal to 29.057. This difference is due to the velocity varies spatially. Therefore, it is the worst case to be analyzed.

Figure 5(b) examines the convergence rate of spectral radius of the iteration matrix $\rho(\mathbf{R})$ in the process of calculus. Several curves as function of iteration number $\beta-$Iter are represented taking the diffusivity values k = 1, 1e − 01, 1e − 10. For the tolerance prefixed in each case, we can appreciate the convergence process has a high dependence on the diffusivity values k. In all cases, after a few iterations with strong slope, the convergence becomes very slow, and it is necessary to develop many iterations until to reach the desirable precision at the parameter's space, however, we expect that the precision of numerical solution at the solution's space increases as same as the diffusivity values k diminishes. In the following we analyze different numerical aspects relate to the convergence of $\beta-$**Algorithm**.

Tables 14-16 give us the iteration numbers corresponding to $\alpha-$**Algorithm** ($\alpha-$Iter) and $\beta-$**Algorithm**($\beta-$Iter) respectively, the optimal β_g^* value, the corresponding convergence rate of the spectral radius $\rho(\mathbf{R})$ of its iteration matrix, and the residuals at the solution's space. The spectral radius values expresses the discrepancy level at the $\delta-$ equation $[\rho(\mathbf{R}(\alpha_k^{nh}))]^2 - \delta = 0$ corresponding to each parameter value p selected. The numerical results shown have been obtained on non-uniform meshes with 7 linear elements.

			Error level	$(\delta = 1e-02)$		
p	$\alpha-$ Iter	$\beta-$ Iter	$\beta_g^*/\mathbf{Pe}^3$	$\rho(\mathbf{R})$	$\|\|\hat{\phi}_k - \phi_{exact}\|\|$	$\frac{\|\|\hat{\phi}_k-\phi_{exact}\|\|}{\|\|\phi_{exact}\|\|}$
1.0000	1	2	5.6384e-01	3.3091e-03	9.7447e-03	3.6225e-03
0.5000	1	31	4.4782e+12	8.3758e-03	6.0932e-02	2.2670e-02
0.3333	1	84	1.6014e+02	9.9961e-03	1.4241e-02	5.3005e-03
0.2500	1	87	1.1298e+02	9.9804e-03	1.2367e-02	4.6037e-03
0.2000	1	87	9.6406e+01	9.9493e-03	1.1511e-02	4.2857e-03
0.1667	1	86	8.7040e+01	9.9499e-03	1.1046e-02	4.1130e-03
0.1429	1	86	8.2462e+01	9.8720e-03	1.0655e-02	3.9675e-03
0.1250	1	85	7.8870e+01	9.9700e-03	1.0555e-02	3.9303e-03
0.1111	1	84	7.5492e+01	9.9888e-03	1.0418e-02	3.8795e-03
			Error level	$(\delta = 1e-03)$		
1	1	3	1.0345e+04	8.1042e-05	9.2445e-01	3.4366e-01
2	6	4	1.8188e+03	1.2960e-04	3.7869e-01	1.4075e-01
3	6	5	5.9400e+02	3.0325e-04	2.5991e-01	9.6811e-02
4	7	7	1.8852e+02	8.8882e-04	1.8147e-01	6.7886e-02
5	7	9	2.7852e+02	6.6032e-04	1.1610e-01	4.3693e-02
6	7	14	-2.9238e+02	7.9386e-04	6.2654e-02	2.3746e-02
7	6	22	-3.9652e+02	7.9897e-04	4.5618e-02	1.7425e-02
8	6	33	-4.4694e+02	9.8448e-04	8.2519e-02	3.1785e-02
9	6	52	-6.3395e+02	9.5512e-04	1.3026e-01	5.0609e-02

Table 14: Convergence of $\beta-$algorithm using the mesh (62) for several p values, with $k = 1$, nel $= 7$

Table 14 shows the convergence rate of $\beta-$**Algorithm** for p values in the range from zero to nine. To values of p lower than 1, is prefixed an error level $\delta = 1e-02$, and in other cases an error level $\delta = 1e-03$. In all cases examined the diffusivity coefficient is taken k = 1. For $1 < \text{p} \leq 7$ the convergence is achieved within 25 $\beta-$iterations at the parameter's space with the desirable tolerance level, however, the precision of the numerical solution is poor. This behaviour to get better slightly for $0 < \text{p} < 1$. In both cases, there is an element in which **Pe** value is lower than 1, that's mean, that the advective limit is reached at some elements. It is necessary further investigations about these aspects, which will be reported in future work.

Next, we analyze the influence of the diffusivity coefficient k at the convergence process, it is clear that if k values diminish then the **Pe** values increase at the same measure, and by consequence on it we expect an improvement in the precision in the numerical solution.

			Error level	$(\delta = 1e-02)$		
p	$\alpha-$Iter	$\beta-$Iter	$\beta_g^*/\mathbf{Pe}^3$	$\rho(\mathbf{R})$	$\|\|\hat{\phi}_k - \phi_{exact}\|\|$	$\frac{\|\|\hat{\phi}_k-\phi_{exact}\|\|}{\|\|\phi_{exact}\|\|}$
1.0000	0	1	9.5506e-01	1.8358e-04	5.2643e-04	1.9553e-04
0.5000	0	8	7.7113e+08	3.2997e-03	4.5690e-02	1.6998e-02
0.3333	0	2	9.7611e-01	1.6745e-03	2.0687e-03	7.7002e-04
0.2500	0	2	9.7580e-01	1.1708e-03	1.1805e-03	4.3954e-04
0.2000	0	2	9.7594e-01	9.9426e-04	9.0522e-04	3.3708e-04
0.1667	0	2	9.7610e-01	9.0419e-04	7.7533e-04	2.8874e-04
0.1429	0	2	9.7624e-01	8.4960e-04	7.0096e-04	2.6107e-04
0.1250	0	2	9.7635e-01	8.1300e-04	6.5324e-04	2.4331e-04
0.1111	0	2	9.7644e-01	7.8675e-04	6.2023e-04	2.3102e-04
			Error level	$(\delta = 1e-03)$		
			$\beta_g/\mathbf{Pe}^3$			
1	6	1	9.5506e-01	1.8357e-04	5.2644e-04	1.9553e-04
2	8	2	9.5165e-01	3.7401e-04	6.0260e-04	2.2310e-04
3	9	2	9.3558e-01	3.1143e-04	8.9845e-04	3.3159e-04
4	9	2	9.1789e-01	3.8973e-04	1.3059e-03	4.8058e-04
5	0	2	8.9865e-01	5.4743e-04	1.9181e-03	7.0401e-04
6	0	2	8.7878e-01	7.8933e-04	2.7294e-03	9.9952e-04
7	0	4	1.7068e+00	5.6566e-04	3.7005e-03	1.3525e-03
8	0	4	1.6655e+00	7.9628e-04	4.7197e-03	1.7221e-03
9	0	6	2.4310e+00	7.3149e-04	5.7565e-03	2.0974e-03

Table 15: Convergence of $\beta-$algorithm using the mesh (62) for several p values, $k = 1e-01$, nel $= 7$

Table 15 shows a comparison of performance of $\beta-$**Algorithm** for several p values, such that **Pe** values are multiplied by 10 due to reduction of diffusivity coefficient k a factor 10, k $= 1e-01$. We can observe in both tables the improvement obtained when $min(Pe) > 1$ independently of p value, and the optimal β_g^* is scaled by $\mathbf{Pe}^t$, with t = 3. We recall the attention that the iteration numbers ($\beta-$Iter) to be reduced drastically. With the same tolerance the spectral radius have a more uniform behavior in p than in the Table 14. Finally, the residuals at the solution's space reflect that the precision of the numerical solution to be better, except p = 0.5 case. Also, there are cases where the residuals have diminished a factor 3. We expect that this behavior to be better as k diminishes.

In case of the diffusivity coefficient k $= 1e-10$, the problem governed by (1)-(2) is purely convection dominated problem. With a prefixed error level $\delta = 1e-12$ the $\beta-$**Algorithm** converges in satisfactory manner for all p values examined, which are shown in Table 16. Only one $\beta-$iteration is necessary to achieve the convergence at the parameter's space, according to scales of **Pe** selected. For p <= 4 the precision reached at the numerical solution is excellent even for p = 0.5, and the other cases maintain the precision achieved earlier. Given the tolerance $\delta = 1e-12$ on this experiment the numerical results shown in the table 16 are very close to precision limit with the proposed method.

			Error level	$(\delta = 1e-12)$		
p	$\alpha-$Iter	$\beta-$Iter	$\beta_g^*/\mathbf{Pe}^2$	$\rho(\mathbf{R})$	$\|\|\hat{\phi}_k - \phi_{exact}\|\|$	$\frac{\|\|\hat{\phi}_k-\phi_{exact}\|\|}{\|\|\phi_{exact}\|\|}$
0.1111	9	1	1.0000e-00	1.9763e-15	2.1601e-04	8.0466e-05
0.1250	9	1	1.0000e-00	8.9804e-16	2.1652e-04	8.0652e-05
0.1429	9	1	1.0000e-00	6.2103e-16	2.1718e-04	8.0895e-05
0.1667	9	1	1.0000e-00	1.1683e-15	2.1808e-04	8.1222e-05
0.2000	9	1	1.0000e-00	1.8816e-15	2.1935e-04	8.1689e-05
0.2500	9	1	1.0000e-00	1.6157e-15	2.2132e-04	8.2408e-05
0.3333	10	1	1.0000e-00	3.2172e-15	2.2474e-04	8.3661e-05
0.5000	10	1	1.0000e-00	3.7445e-16	2.3220e-04	8.6392e-05(*)
			Error level	$(\delta = 1e-012)$		
			$\beta_g^*/\mathbf{Pe}^3$			
1	11	1	1.0000e-00	3.2473e-16	2.6069e-04	9.6837e-05
2	13	1	1.0000e-00	2.9131e-16	3.6063e-04	1.3353e-04
3	15	1	1.0000e-00	2.6015e-16	5.6159e-04	2.0729e-04
4	17	1	1.0000e-00	2.1790e-16	9.3058e-04	3.4250e-04
5	18	1	1.0000e-00	2.7249e-16	1.5148e-03	5.5609e-04
6	19	1	1.0000e-00	2.4636e-16	2.3029e-03	8.4347e-04
7	20	1	1.0000e-00	2.0075e-16	3.2338e-03	1.1821e-03
8	21	1	1.0000e-00	1.3152e-16	4.2345e-03	1.5454e-03
9	22	1	1.0000e-00	2.5282e-16	5.2462e-03	1.9118e-03

Table 16: Convergence of $\beta-$algorithm using the mesh (62) for several p values, $k = 1e-10$, nel $= 7$

			Error level	$(\delta = 1e-03)$	(p = 1/3)	
nel	$\alpha-$Iter	$\beta-$Iter	$\beta_g^*/\mathbf{Pe}^2$	$\rho(\mathbf{R})$	$\|\|\hat{\phi}_k - \phi_{exact}\|\|$	$\frac{\|\|\hat{\phi}_k-\phi_{exact}\|\|}{\|\|\phi_{exact}\|\|}$
7	4	158	8.9491e+01	9.9500e-04	1.0168e-03	3.7850e-04
14	4	150	8.1184e+01	9.9027e-04	1.5780e-03	4.1454e-04
28	4	148	7.8644e+01	9.8955e-04	2.2508e-03	4.1766e-04
56	4	148	7.7937e+01	9.9629e-04	3.1046e-03	4.0715e-04
112	4	150	7.8593e+01	9.9863e-04	4.1858e-03	3.8806e-04

Table 17: Convergence process of $\beta-$algorithm relative to element number nel using the mesh (62) with error levels $\delta = 1e-03$, and $k = 1/\text{nel}$

Now, we are interested on the behavior of proposed method relative to resolution at mesh, and the influence of several scales of $\mathbf{Pe}^t$ in the precision of numerical solution.

To examine the influence of the element number of the mesh (nel) at the convergence process we seek meshes **f1** with p = 1/3, and $k = 1/\text{nel}$. It values of k guarantees that the **Pe** values are similar in each mesh for the p values prefixed. Table 17 give us the numerical results with a error level $\delta = 1e-03$ at the parameter's space. On this experiment, the optimal β_g^* value is scaled by $\mathbf{Pe}^2$, and the element number increases until to factor four, and then, the last mesh in question is four times more fine that initial mesh. We can appreciate that the computational cost is almost equal independently of the resolution of mesh. A similar behavior is observed as well as the residuals at the parameter's space and the residuals at the solution's space. We conclude that the precision of numerical results is only slightly affected by the increment of the number of element at mesh.

			Error level	($\delta = 1e - 03$)	(p = .1429)	
t	$\alpha-$Iter	$\beta-$Iter	$\beta_g^*/\mathbf{Pe}^{\mathrm{t}}$	$\rho(\mathbf{R})$	$\lVert\hat{\phi}_k - \phi_{exact}\rVert$	$\frac{\lVert\hat{\phi}_k - \phi_{exact}\rVert}{\lVert\phi_{exact}\rVert}$
2	1	959	3.5443e+03	9.9914e-04	2.3920e-03	8.9047e-04
3	1	827	1.1317e+03	9.9928e-04	2.3845e-03	8.8766e-04
4	1	562	3.6145e+02	9.9825e-04	2.3771e-03	8.8490e-04
5	1	276	1.1582e+02	9.9537e-04	2.3698e-03	8.8217e-04
6	1	106	3.7211e+01	9.9100e-04	2.3627e-03	8.7956e-04
7	1	36	1.1888e+01	9.9374e-04	2.3575e-03	8.7760e-04
8	1	12	3.8920e+00	9.7368e-04	2.3496e-03	8.7466e-04
9	1	4	1.2961e+00	9.4152e-04	2.3421e-03	8.7188e-04
10	1	2	6.5188e-01	6.0473e-04	2.3431e-03	8.7225e-04

Table 18: Acceleration of convergence process of $\beta-$algorithm for several β_g scales, using the mesh (62) for p = .1429 value, with error levels $\delta = 1e-03$, $k = 1$, nel = 7, maximum **Pe** value =3.4569, and minimum **Pe** value =2.5677

Another practical aspect in the implementation of $\beta-$**Algorithm** is the adequate selection of the $\mathbf{Pe}^t$ scales. These scales allows in many cases that the iterative procedure to be more cheaper. Table 18 exhibits the convergence properties at the parameter's pace and solution's space for meshes **f1**(p = .1429) with an error level $\delta = 1e - 03$. On this experiment **Pe** values slowly vary from an element to other. It is very significant to note that the iterative process requires only 3 iterations to converge at the parameter's space, and even more, that reached precision is not sensible to selected scale as we can appreciate in the behavior of residuals at the table.

Figures 6 and 7 show a comparison between the numerical solutions and exact solution for different values of diffusivity k = 1, 1e − 01, 1e − 10 using a mesh **f1** with 7 linear elements, taken p = 9 and p = 0.9 respectively. As we can appreciate in the following that the correspondence between the $\beta-$numerical solution and the exact solution is excellent almost every cases, except in case of p = 9,
rmk = 1, in which a little discrepancy is noted, however, also in this case the qualitative behavior is excellent.

A measure of the spatially nodal error is shown in the figure 8, the upper figure for p = 9 and the lower figure for p = 0.9. We note the absolute error takes its maximum value at the interior node where the velocity $U = 60/x$ takes the minimum value. We can appreciate that using the mesh **f1**(p = 0.9) the accuracy of the numerical results obtained is better than those obtained with the mesh **f1**(p = 9), independently of diffusivity coefficient k values.

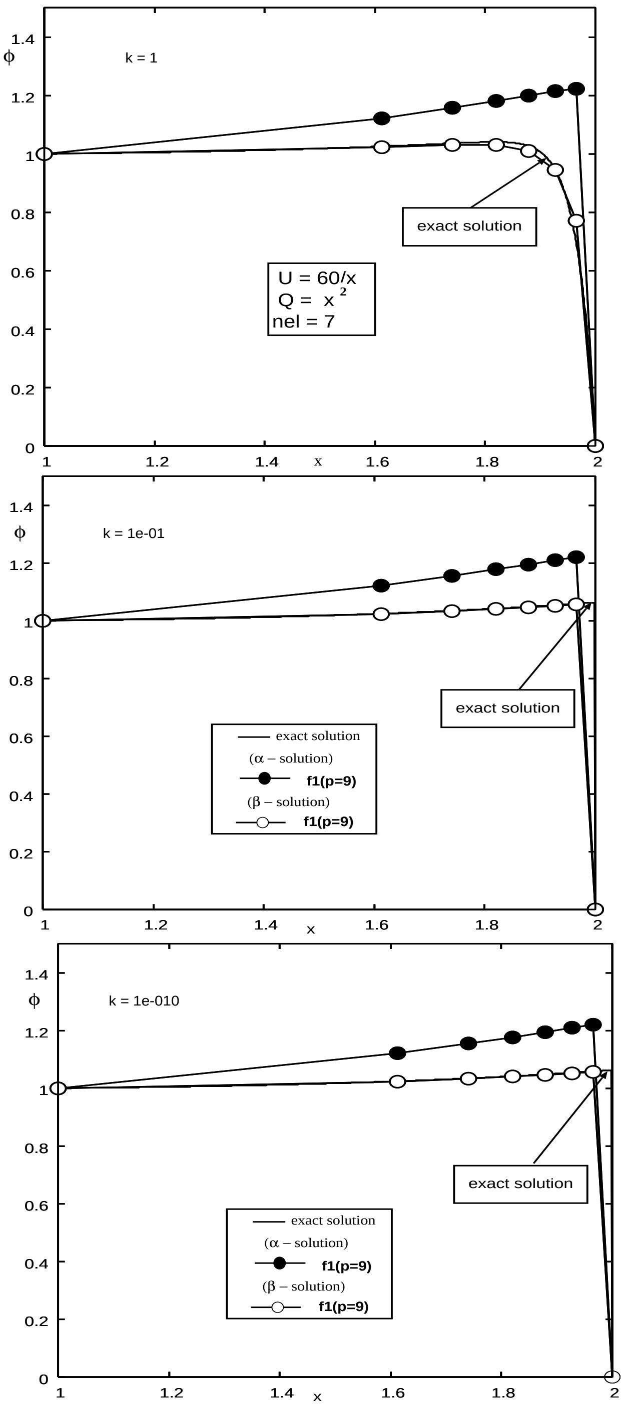


Figure 6: Solution of one-dimensional convection-diffusion problem with 7 linear elements, $U = 60/x$, $Q = x^2$ and $\phi(1) = 1$, $\phi(2) = 0$ using the mesh **f1** with p = 9. (up) k= 1, (middle) k = 1e-01, (down) k= 1e-10.

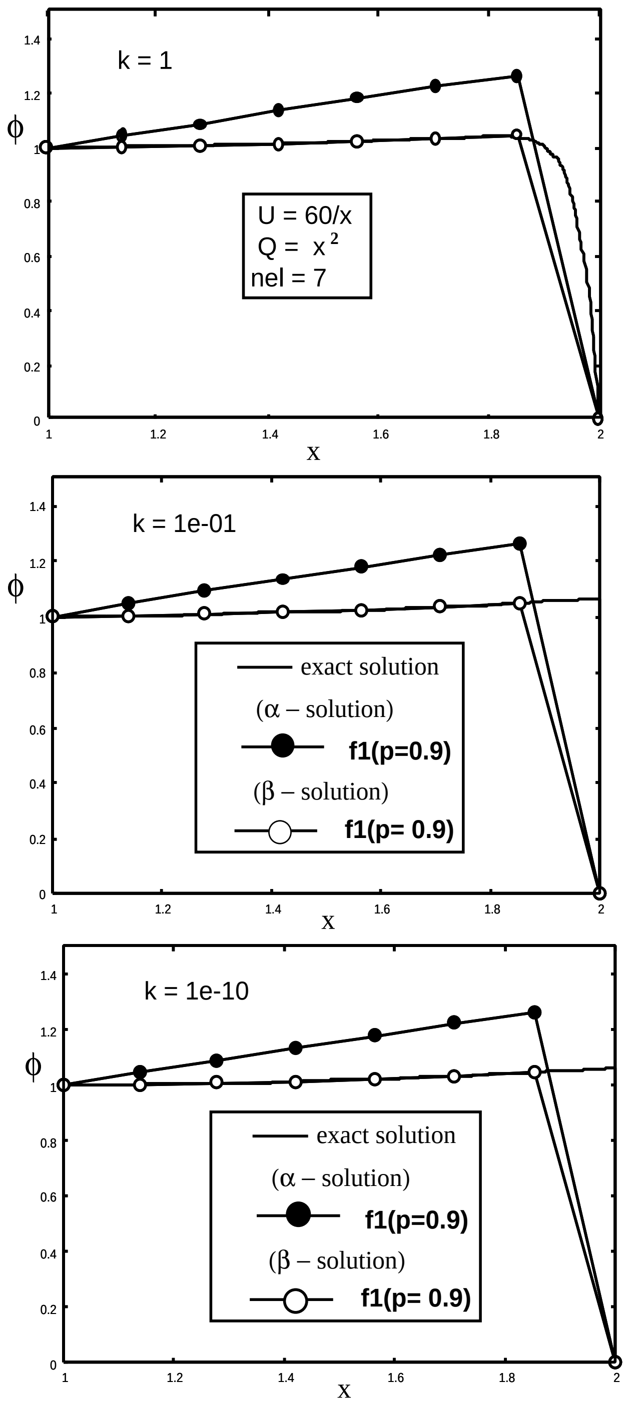
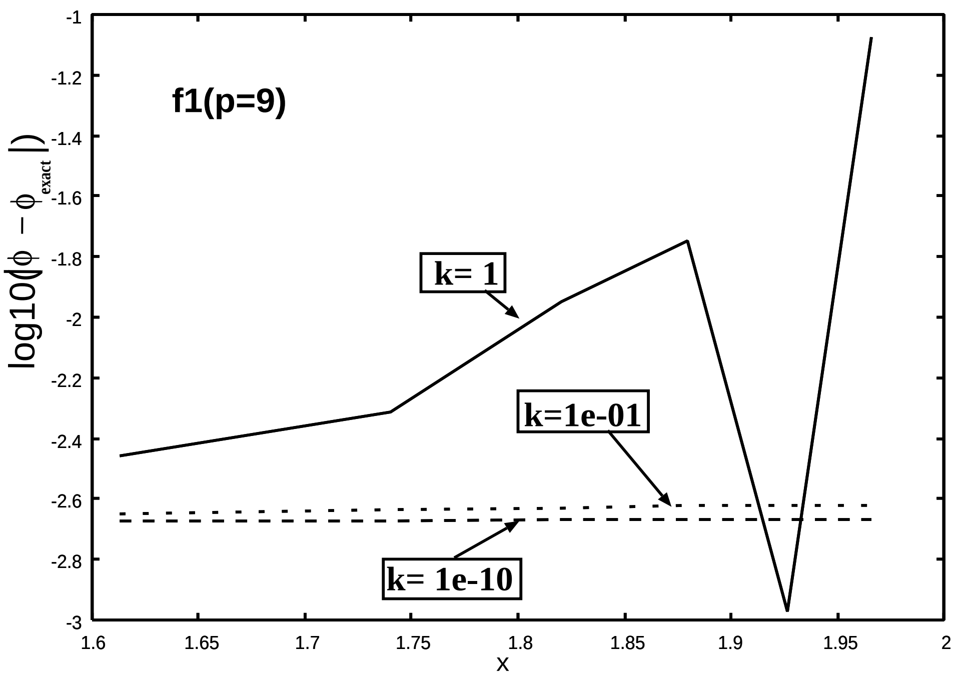


Figure 7: Solution of one-dimensional convection-diffusion problem with 7 linear elements, $U = 60/x,\ Q = x^2$ and $\phi(1) = 1,\ \phi(2) = 0$ using the mesh **f1** with p = 0.9. (up) k= 1, (middle) k = 1e-01, (down) k= 1e-10.



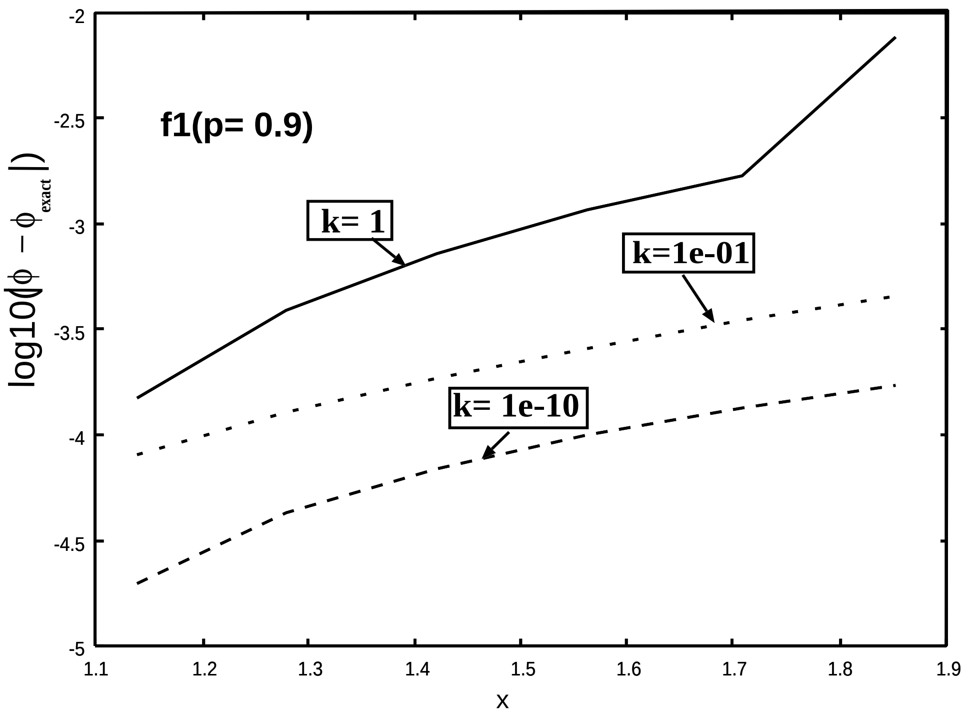


Figure 8: Solution of one-dimensional convection-diffusion problem with 7 linear elements, $U = 60/x,\ Q = x^2$ and $\phi(1) = 1,\ \phi(2) = 0$ using the mesh **f1**. Comparison of Absolute Errors between $\beta-$solution and the exact solution along the domain, for different diffusivity values $k = 1,\ 1e-01,\ 1e-10$. (up) p = 9, (down) p = 0.9.

Example 4

$$10x\frac{d\phi}{dx} - k\frac{d^2\phi}{dx^2} = x^2 \quad in\ \Omega = (1,2),\ \phi(1) = 0,\ \phi(2) = \log(2)$$

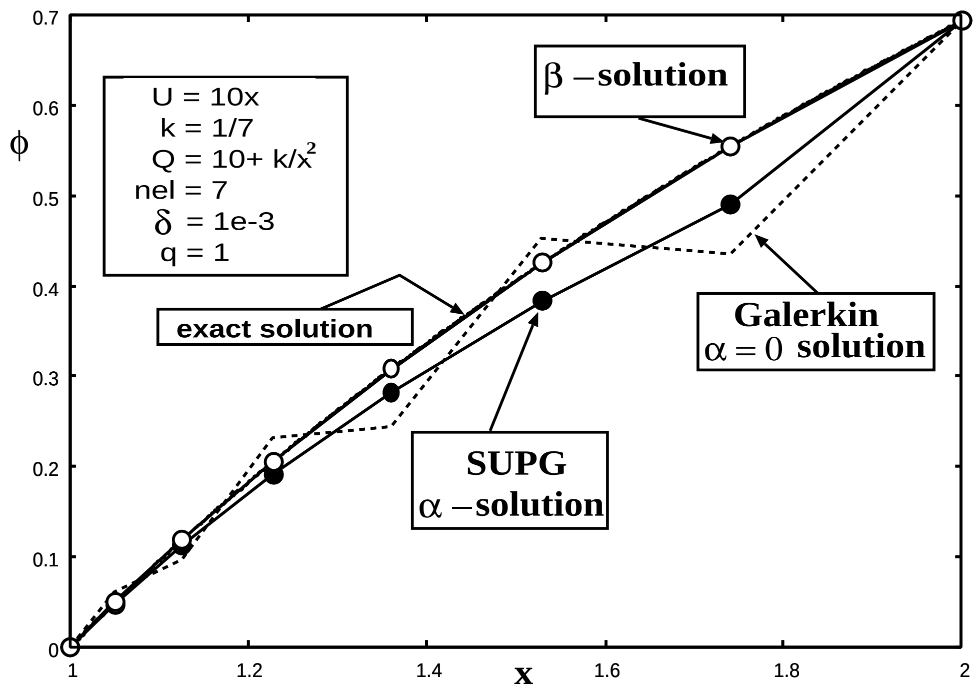


Figure 9: Solution of one-dimensional convection-diffusion problem with 7 linear elements, $U = 10x,\ Q = 10 + k/x^2$ and $\phi(1) = 0,\ \phi(2) = \log(2)$ using $\beta-$algorithm for the mesh **f2**(q = 1).

As we know the selection of adequate mesh for a prefixed problem is the principal task at the finite element method. Several numerical experiments confirm a certain improvement in the performance of proposed method using the mesh **f2**(q = 1), and the convergence is achieved within few iterations: 6 iterations with the $\alpha-$**Algorithm** and 5 iterations with the $\beta-$**Algorithm**, respectively. The computed solution is obtained with a precision of $7.7117e-04$.

Figure 9 displays the curves corresponding to the numerical solutions which have been obtained by the Galerkin method ($\alpha_{nh} = 0$), usual SUPG method ($\alpha_{nh} = \bar{\alpha}_e$) and stabilized SUPG method by the formula(42). A comparison between them and the exact solution confirm the good performance of the proposed method and the local stabilization formula (42), as well as, the significant improvement relative to usual SUPG method. A further knowledge of behavior of residuals is shown in figure 10, which plot the $log10(|\phi - \phi_{exact}|)$ versus x, that is, the spatial behavior of absolute errors for each method. It is clear the superiority of the proposed method.

6.1 Computing the stabilizing parameter β by direct methods

In this section the nodal accuracy of the numerical $\beta-$solution on non-uniform meshes is examined over three different test problems. We compare the performance of the direct

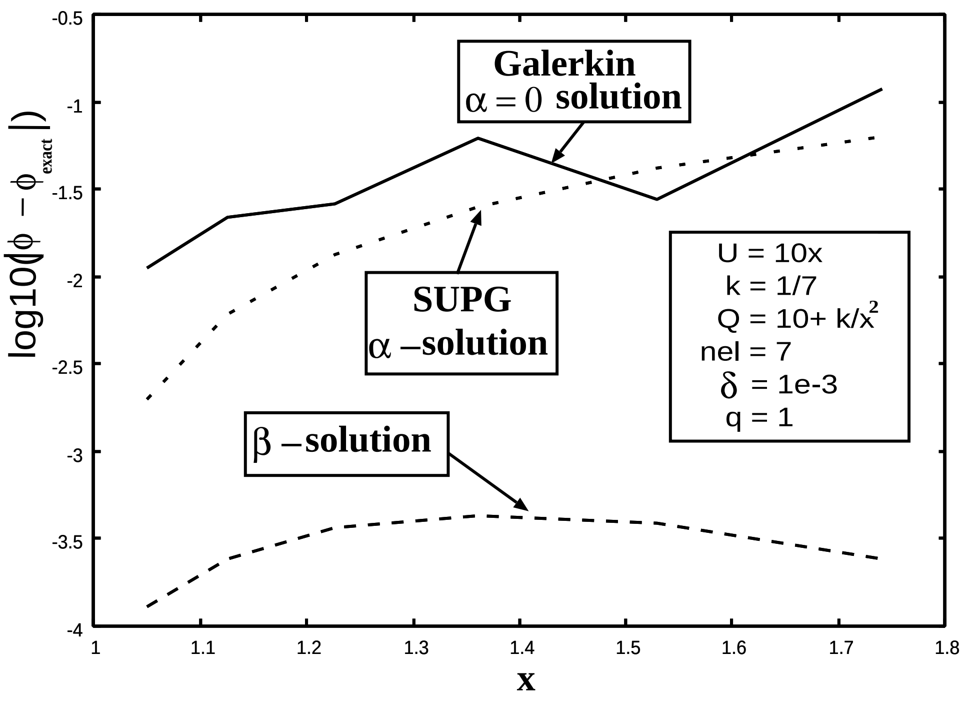


Figure 10: Solution of one-dimensional convection-diffusion problem with 7 linear elements, $U = 10x$, $Q = 10 + k/x^2$ and $\phi(1) = 0$, $\phi(2) = \log(2)$ using $\beta-$algorithm. Comparison of Absolute Errors between $\beta-$solution and the exact solution along the domain, for the mesh **f2**(q = 1).

method given by the formula (52) with the function $p(x)$ chosen according to (43) and according to the solution (49) of the local equation (48). The discrete systems were solved by direct method using MATLAB 5.2

The results of our numerical tests are presented in Tables 19-24. First we note that the errors at solution's space are almost independent of the diffusivity parameter k, in case of the convection term is dominant. Thus, the $\beta-$solution by direct method can be used with adapted meshes. Also, we can observe that using the formula (52) the precision of numerical results increases significantly.

	k	10^{-4}	10^{-6}	10^{-8}	10^{-10}
nel	$\rho(\mathbf{R})$	$\|\|\hat{\phi}_\beta - \phi_{exact}\|\|$	$\|\|\hat{\phi}_\beta - \phi_{exact}\|\|$	$\|\|\hat{\phi}_\beta - \phi_{exact}\|\|$	$\|\|\hat{\phi}_\beta - \phi_{exact}\|\|$
16	3.59e+00	9.8684e-02	9.8687e-02	9.8687e-02	9.8687e-02
32	7.45e+00	6.8455e-02	6.8459e-02	6.8459e-02	6.8459e-02
64	1.51e+01	4.6218e-02	4.6223e-02	4.6223e-02	4.6223e-02
128	3.04e+01	3.0690e-02	3.0696e-02	3.0696e-02	3.0696e-02
256	6.10e+01	2.0171e-02	2.0178e-02	2.0178e-02	2.0178e-02
512	1.22e+02	1.3174e-02	1.3183e-02	1.3183e-02	1.3183e-02
1024	2.44e+02	8.5732e-03	8.5832e-03	8.5833e-03	8.5833e-03
2048	4.89e+02	5.5672e-03	5.5789e-03	5.5791e-03	5.5791e-03

Table 19: $\beta-$solution for Example 2 with $\beta = 1$ and $p(x)$ chosen according to (43)

	k	10^{-4}	10^{-6}	10^{-8}	10^{-10}
nel	$\rho(\mathbf{R})$	$\|\|\hat{\phi}_\beta - \phi_{exact}\|\|$	$\|\|\hat{\phi}_\beta - \phi_{exact}\|\|$	$\|\|\hat{\phi}_\beta - \phi_{exact}\|\|$	$\|\|\hat{\phi}_\beta - \phi_{exact}\|\|$
16	5.0266e+00	3.0285e-01	3.0285e-01	3.0285e-01	3.0285e-01
32	1.0152e+01	2.8843e-01	2.8843e-01	2.8843e-01	2.8843e-01
64	2.0355e+01	2.7191e-01	2.7191e-01	2.7191e-01	2.7191e-01
128	4.0735e+01	2.5503e-01	2.5503e-01	2.5503e-01	2.5503e-01
256	8.1482e+01	2.3859e-01	2.3859e-01	2.3859e-01	2.3859e-01
512	1.6297e+02	2.2291e-01	2.2291e-01	2.2291e-01	2.2291e-01
1024	3.2595e+02	2.0813e-01	2.0813e-01	2.0813e-01	2.0813e-01
2048	6.5190e+02	1.9427e-01	1.9427e-01	1.9427e-01	1.9427e-01

Table 20: $\beta-$solution for Example 2 with $\beta = 1$ and $p(x)$ chosen according to (49)

	k	10^{-4}	10^{-6}	10^{-8}	10^{-10}
nel	$\rho(\mathbf{R})$	$\|\|\hat{\phi}_\beta - \phi_{exact}\|\|$	$\|\|\hat{\phi}_\beta - \phi_{exact}\|\|$	$\|\|\hat{\phi}_\beta - \phi_{exact}\|\|$	$\|\|\hat{\phi}_\beta - \phi_{exact}\|\|$
16	1.4072e-10	3.6664e-03	3.6656e-03	3.6656e-03	3.6656e-03
32	1.1136e-10	2.4206e-03	2.4194e-03	2.4194e-03	2.4194e-03
64	8.8659e-11	1.4214e-03	1.4198e-03	1.4198e-03	1.4198e-03
128	7.2591e-11	7.3099e-04	7.2863e-04	7.2861e-04	7.2861e-04
256	6.2278e-11	3.2872e-04	3.2538e-04	3.2535e-04	3.2535e-04
512	5.6595e-11	1.3400e-04	1.2927e-04	1.2922e-04	1.2922e-04
1024	5.4476e-11	5.4557e-05	4.7831e-05	4.7764e-05	4.7763e-05
2048	5.4822e-11	2.6860e-05	1.7213e-05	1.7119e-05	1.7118e-05

Table 21: $\beta-$solution for Example 3 with β computed by the formula (52) and $p(x)$ chosen according to (43)

	k	10^{-4}	10^{-6}	10^{-8}	10^{-10}
nel	$\rho(\mathbf{R})$	$\|\|\hat{\phi}_\beta - \phi_{exact}\|\|$	$\|\|\hat{\phi}_\beta - \phi_{exact}\|\|$	$\|\|\hat{\phi}_\beta - \phi_{exact}\|\|$	$\|\|\hat{\phi}_\beta - \phi_{exact}\|\|$
16	1.6282e-09	3.6664e-03	3.6656e-03	3.6656e-03	3.6656e-03
32	2.1161e-09	2.4206e-03	2.4194e-03	2.4194e-03	2.4194e-03
64	2.8493e-09	1.4214e-03	1.4198e-03	1.4198e-03	1.4198e-03
128	4.0374e-09	7.3098e-04	7.2863e-04	7.2860e-04	7.2860e-04
256	6.1010e-09	3.2871e-04	3.2537e-04	3.2534e-04	3.2534e-04
512	9.9048e-09	1.3398e-04	1.2925e-04	1.2920e-04	1.2920e-04
1024	1.7219e-08	5.4500e-05	4.7774e-05	4.7707e-05	4.7706e-05
2048	3.1620e-08	2.6710e-05	1.7067e-05	1.6973e-05	1.6972e-05

Table 22: $\beta-$solution for Example 3 with β computed by the formula (52) and $p(x)$ chosen according to (49)

	k	10^{-4}	10^{-6}	10^{-8}	10^{-10}
nel	$\rho(\mathbf{R})$	$\|\|\hat{\phi}_\beta - \phi_{exact}\|\|$	$\|\|\hat{\phi}_\beta - \phi_{exact}\|\|$	$\|\|\hat{\phi}_\beta - \phi_{exact}\|\|$	$\|\|\hat{\phi}_\beta - \phi_{exact}\|\|$
16	4.9710e-09	1.2658e-04	1.2625e-04	1.2624e-04	1.2624e-04
32	9.1854e-09	4.4990e-05	4.4507e-05	4.4502e-05	4.4502e-05
64	1.7585e-08	1.6392e-05	1.5686e-05	1.5679e-05	1.5679e-05
128	3.4373e-08	6.5541e-06	5.5286e-06	5.5184e-06	5.5183e-06
256	6.7946e-08	3.4134e-06	1.9258e-06	1.9113e-06	1.9111e-06
512	1.3509e-07	2.7418e-06	5.9023e-07	5.6964e-07	5.6944e-07
1024	2.6938e-07	3.0140e-06	7.2490e-08	1.0263e-07	1.0294e-07
2048	5.3795e-07	3.5574e-06	8.4084e-07	8.8330e-07	8.8366e-07

Table 23: $\beta-$solution for Example 4 with β computed by the formula (52) and $p(x)$ chosen according to (43)

	k	10^{-4}	10^{-6}	10^{-8}	10^{-10}
nel	$\rho(\mathbf{R})$	$\|\|\hat{\phi}_\beta - \phi_{exact}\|\|$	$\|\|\hat{\phi}_\beta - \phi_{exact}\|\|$	$\|\|\hat{\phi}_\beta - \phi_{exact}\|\|$	$\|\|\hat{\phi}_\beta - \phi_{exact}\|\|$
16	5.5283e-09	9.1187e-05	8.9893e-05	8.9880e-05	8.9880e-05
32	1.0432e-08	3.3624e-05	3.1791e-05	3.1773e-05	3.1773e-05
64	2.0194e-08	1.3858e-05	1.1253e-05	1.1227e-05	1.1227e-05
128	3.9701e-08	7.7129e-06	3.9888e-06	3.9522e-06	3.9518e-06
256	7.8704e-08	6.7250e-06	1.4005e-06	1.3488e-06	1.3483e-06
512	1.5671e-07	7.9803e-06	4.1302e-07	3.4149e-07	3.4078e-07
1024	3.1271e-07	1.0549e-05	1.8455e-07	2.8974e-07	2.9080e-07
2048	6.2348e-07	1.4094e-05	1.0824e-06	1.2334e-06	1.2349e-06

Table 24: $\beta-$solution for Example 4 with β computed by the formula (52) and $p(x)$ chosen according to (49)

7 CONCLUSIONS

In this work direct and iterative procedures have been presented to compute the local stabilization parameter approximately. Approximate symmetrization of global matrix have been done in case of physical properties vary along the domain. The obtained scheme allows to compute the local parameter α that stabilizes the solution derived by Galerkin method when the convection term is dominant in the framework of $SUPG$ formulation. A feasible and efficient numerical implementation on a uniform mesh and using a basis of linear functions have been done using the spectral properties of local matrix. Numerical examples are examined to confirm the good performance of this method in case of uniform mesh. An extension of proposed method to non-uniform mesh is also presented, and it is applied to solve at the discrete medium a variable coefficients problem, which includes the spatial variability of velocity and source term. From the numerical experiments we can conclude that the $\alpha-$ algorithm via $QN0$ and $Bis + QN1$ have a good accuracy on an uniform mesh. Numerical results show that the extension of this algorithm to non-homogeneous case was successfully, specially, when the minimum **Pe** value is greater than one. Good results were obtained by application of direct method on coarse mesh and on adapted meshes. Future works are directed to get approximate symmetrization to convection-diffusion-reaction problems.

APPENDIX

Nomenclature

- ϕ = scalar field
- $\hat{\phi}$ = nodal approximation to ϕ
- U = velocity
- k = diffusion parameter
- Q = source term
- nel = element number
- N^i = global piecewise linear basis functions
- Ω = finite element computation domain
- Γ = boundary surface
- $\mathbf{A}$ = global matrix
- $\mathbf{f}$ = independent term
- $\mathbf{e}rr$ = error
- $\sigma(A_l)$ local matrix spectrum

ACKNOWLEDGMENTS

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.

References

[1] Akira Mizukami, Thomas J.R. Hughes. A petrov-galerkin finite element method for convection - dominated flows: an accurate upwinding technique for satisfying the maximum principle. *Computer methods in applied mechanics and engineering*, 50:181–193, 1985.

[2] J.C.Heinrich, P.S.Huyakorn, O.C. Zienkiewicz and A.R.Mitchell. An 'upwind' finite element scheme for two-dimensional convective transport equation. *IJNME*, 11:131–143, 1977.

[3] T.J.R. Hughes and A. Brooks. *A multidimensional upwind scheme with no crosswind diffusion*, volume 34 of *AMD*. Finite Element Methods for convection dominated flows, ASME, New York, 1979.

[4] I.Christie, D.F.Griffiths, A.R.Mitchell and O.C. Zienkiewicz. Finite element methods for second order differential equations with significant first derivatives. *IJNME*, 10:1389–1396, 1976.

[5] 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.

[6] Zienkiewicz O.C., Taylor R.L. *El método de los elementos finitos. Mecánica de Sólidos y Fluidos. Dinámica y no linealidad, volumen 2 §§15.7-15.14*, volume Parte II §§15.7-15.14. McGraw-Hill/Interamericana de España, S.A., 1994. Ecuaciones linealizadas de aguas poco profundas y olas.

[7] Eugenio Oñate. Derivation of stabilized equations for advective-diffusive transport and fluid flow problems. *Comput. Methods Appl. Mech. Engrg.*, 151(1/2):233–267, 1998.

[8] Tortsen LinB. Anisotropic meshes and streamline-diffusion stabilization for convection-diffusion problems. *Communications in Numerical Methods in Engineering*, 21:515–525, 2005.

[9] William L. Briggs,Van Emden Henson, Steve F. McCormick. *A multigrid Tutorial.* SIAM, 2000.