DATA STRUCTURES FOR THE NONLINEAR SUBGRID STABILIZATION METHOD

Isaac P. Santos
Lucia Catabriga
Departamento de Informática
Universidade Federal do Espírito Santo - UFES
Av. Fernando Ferrari, 514 - Vitória, ES, Brazil CEP: 29075-910
{*ipsantos,luciac*}*@inf.ufes.br*
Regina C. Almeida
Departamento de Mecânica Computacional
Laboratório Nacional de Computação Científica - LNCC/MCT
Av. Getúlio Vargas, 333 - Petrópolis, RJ, Brazil CEP: 25651-075
rcca@lncc.br

Abstract. *In this work we present a new implementation strategy for the Nonlinear Subgrid Stabilization (NSGS) method (Santos and Almeida, 2007, 2008). The NSGS method is a methodology for approximating transport problems, based on a multiscale decomposition of the approximation space. The two-scale viscosity model is obtained by adding to the Galerkin formulation a nonlinear operator acting only on the unresolved mesh scales. The resulting method is stable and does not require any tune-up parameter. However, it requires the solution of a system twice as large as that related to the final resolved scale resolution. To avoid this drawback, we propose here a new strategy to approximate the effects of the unresolved scales on the resolved scales combined with an element-based and edge-based data structures. The resulting methodology yields superior performance, which is demonstrated by a variety of numerical experiments covering the regimes of dominant advection and dominant absorption.*

Keywords: *Nonlinear subgrid viscosity, Edge-based data structure, Krylov iterative methods*

1. INTRODUCTION

The numerical solution of transport equations using the Galerkin formulation may exhibit global spurious oscillations when the diffusive term is smaller than the advective or reactive ones. More accurate and stable results can be obtained using stabilized or multiscale methods. In a general sense, each of these alternative methodologies are based on adding some type of linear stabilization to the standard Galerkin formulation and usually depends on the definition of one or more tuning parameters. The multiscale approach consists of decomposing the approximation space into a resolved coarse scale and an unresolved subgrid scale spaces such that the weak form of the problem is split into two sub-problems: one for the coarse scales and one for the subgrid scales. From this point of view, the attempts to recover stability of the resolved scale solution may be interpreted as a way of improving the simulation by considering the effects of the smallest scales on the larger ones Brezzi and Marini (2002).

The Nonlinear Subgrid Stabilization (NSGS) method is a step towards the development of two-scale methods whose stability and convergence properties do not rely on tune-up parameters. Its first version was presented in (Santos and Almeida, 2007) for advection-diffusion problems and was based on adding a nonlinear artificial viscosity term only to the subgrid scales. The amount of the subgrid viscosity depends on assuming an additive decomposition of the velocity field and solving the resolved scale equation at the element level, yielding a self adaptive method. A linear subgrid stabilization was kept when the resolved scale solution was accurate enough.

The NSGS method was extended in (Santos and Almeida, 2008) to advection-diffusion-reaction problems and some improvements were introduced in smooth regions so as to recover the Galerkin formulation for regular resolved scale solutions. To improve the convergence of the linearization process, an alternative iterative algorithm to solve the nonlinear problem was also proposed. The resulting self adapted methodology has very good stability and convergence properties for a wide range of transport problems, but presents the drawback of requiring the solution of a system twice as large as that associated with the resolved scale resolution. To overcome this drawback, in this work we present a strategy for building the NSGS method using element-based (EBE) and edge-based (EDS) implementations (Catabriga et al., 1998; Catabriga and Coutinho, 2002). The present strategy locally approximates the effects of the subgrid scales on the resolved scales by performing a static condensation of the subgrid degrees of freedom of the associated macro element. Solution efficiency is improved by using an edge based data structure and the final algebraic system is solved by using the generalized minimal residual method (GMRES) (Saad, 1995).

The outline of this paper is as follows. We briefly describe the Nonlinear Subgrid Stabilization (NSGS) method in Section 2. Section 3 presents the proposed strategy of incorporating the subgrid effects into the resolved scale and Section 4 presents the related element-based (EBE) and edge-based (EDS) data structures. Numerical examples are conducted in Section 5 and Section 6 concludes this paper by pointing out the main achievements, difficulties and perspectives.

2. THE NONLINEAR SUBGRID STABILIZATION METHOD

Consider the steady scalar advection-diffusion-reaction problem whose solution satisfies

$$-\epsilon\Delta u + \boldsymbol{\beta}\cdot\boldsymbol{\nabla} u + \sigma u = f \quad \text{in } \Omega; \tag{1}$$

$$u = 0 \quad \text{on } \partial\Omega, \tag{2}$$

where $\Omega \subset \mathbb{R}^d (d = 2)$ is an open bounded domain with a Lipschitz boundary $\partial\Omega$ and unit outward normal $\boldsymbol{n}$, $\boldsymbol{\beta}$ is the velocity field, σ is the reaction coefficient, $0 < \epsilon \ll 1$ is the (constant) diffusion coefficient, and f is the source term. For simplicity, we only consider homogeneous Dirichlet boundary conditions and it is assumed that $\boldsymbol{\beta} \in W^{1,\infty}(\Omega)$, $\sigma \in L^{\infty}(\Omega)$ and $f \in L^2(\Omega)$.

The weak formulation of the problem (1)-(2) reads: find $u \in H_0^1(\Omega)$ such that

$$B(u,v) = (f,v) \quad \forall v \in H_0^1(\Omega), \tag{3}$$

where

$$B(u,v) = \epsilon(\boldsymbol{\nabla} u, \boldsymbol{\nabla} v) + (\boldsymbol{\beta} \cdot \boldsymbol{\nabla} u, v) + (\sigma u, v) \tag{4}$$

and $(\cdot,\cdot)$ stands for the usual inner product in $L^2(\Omega)$. It is also assumed that $\sigma \geq 0$ and that there exists a constant σ_0 such that

$$\sigma - \frac{1}{2}\boldsymbol{\nabla} \cdot \boldsymbol{\beta} \geq \sigma_0 > 0. \tag{5}$$

The Lax-Milgram Lemma implies that the problem (3) has an unique solution. To define the discrete model, we consider a triangular partition $\mathcal{T}_H = \{T\}$ of the domain Ω where H stands for the mesh parameter. From each triangle $T \in \mathcal{T}_H$, four triangles are created by connecting the midpoints of the edges. We set $h = H/2$ and denote by $\mathcal{T}_h = \{T_h\}$ the resulting finer triangulation. A two-level piecewise linear finite element approximation is defined by introducing the following two spaces:

$$\begin{aligned} X_H &= \{u_H \in H_0^1(\Omega) \quad | \quad u_{H|_T} \in \mathbb{P}_1(T), \quad \forall T \in \mathcal{T}_H\}; & (6)\\ X_h &= \{u_h \in H_0^1(\Omega) \quad | \quad u_{h|_{T_h}} \in \mathbb{P}_1(T_h), \quad \forall T_h \in \mathcal{T}_h\}. & (7)\end{aligned}$$

Moreover, we introduce an additional discrete space $X_h^H \subset X_h$, such that the following decomposition holds:

$$X_h = X_H \oplus X_h^H \ , \tag{8}$$

where X_H is the resolved (coarse) scale space whereas X_h^H is the subgrid (fine) scale space. We define also the projection operator $P_H : X_h \longrightarrow X_H$ so that $X_h^H = (I - P_H)X_h$, where I is the identity operator. Then, given $u_h \in X_h$ and $u_H \in X_H$ such that u_h and u_H coincide in the coarse scale nodes, the space decomposition (8) implies that $u_h^H = (I - P_H)u_h = u_h - u_H$, $\forall u_h^H \in X_h^H$. The couple (X_H, X_h) will be referred to as the *two-level* $\mathbb{P}_1$ *setting* (see Fig. 1) Guermond (1999).

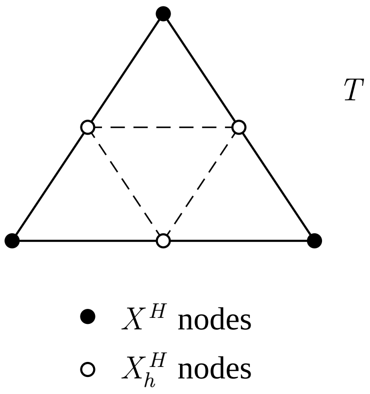


Figure 1: Schematic representation of the *two-level* $\mathbb{P}_1$ *setting*

The Nonlinear Subgrid Stabilization Method (NSGS) (see Santos and Almeida (2007, 2008) for details) is given by

$$\begin{aligned}&\text{Find } u_h \in X_h \text{ such that}\\ &B(u_h, v_h) + \sum_{T_h \in \mathcal{T}_h} D(u_H, u_h^H, v_h^H) = (f, v_h), \quad \forall v_h \in X_h,\end{aligned} \tag{9}$$

where the non-linear operator $D(u_H, u_h^H, v_h^H)$ in each element $T_h \in \mathcal{T}_h$ is given by

$$D(u_H, u_h^H, v_h^H) = \int_{T_h} \xi(u_H) \boldsymbol{\nabla} u_h^H \cdot \boldsymbol{\nabla} v_h^H d\Omega, \tag{10}$$

with

$$\xi(u_H) = \frac{1}{2}\mu(h)|\boldsymbol{\beta}_h^H|$$

and $\mu(h)$ stands for the subgrid characteristic lehgth. For the *two-level* $\mathbb{P}_1$ *setting*, we use $\mu(h) = h/2$. The subgrid velocity field $\boldsymbol{\beta}_h^H$ is given by

$$\boldsymbol{\beta}_h^H = \begin{cases} \frac{R(u_H)}{|\boldsymbol{\nabla} u_H|^2} \boldsymbol{\nabla} u_H, \text{ if } |\boldsymbol{\nabla} u_H| \neq 0; \\ 0, \quad \text{otherwise,} \end{cases} \tag{11}$$

where

$$R(u_H) = -\epsilon \Delta u_H + \boldsymbol{\beta} \cdot \boldsymbol{\nabla} u_H + \sigma u_H - f \ . \tag{12}$$

Setting $\xi(u_H) = c_b h$, with $c_b > 0$ constant, we obtain the Linear Subgrid Stabilization (SGS) method (Guermond, 1999, 2001; Layton, 2002). When $|\boldsymbol{\nabla} u_H| = 0$, the formulation (9) recovers the standard Galerkin method.

The NSGS method is solved using an iterative procedure for which the SGS solution is the initial guess (with $c_b = 1$) and $\xi(u_H)$ (or $\boldsymbol{\beta}_h^H$) is delayed one iteration (see Algorithm 1). The iterative NSGS process is defined by: Given u_h^{n-1}, we find u_h^n satisfying

$$B(u_h^n, v_h) + \sum_{T_h \in \mathcal{T}_h} \int_{T_h} \xi(u_H^{n-1}) \boldsymbol{\nabla} u_h^{H;n} \cdot \boldsymbol{\nabla} v_h^H d\Omega = (f, v_h), \qquad \forall v_h \in X_h, \tag{13}$$

where $\xi(u_H^{n-1}) = c_{b,T_h}\mu(h)$ is locally adjusted depending on the residual of the resolved scale solution. The iterative NSGS procedure is given by (Algorithm 1), where u_H^i and $\boldsymbol{\beta}_h^{H;i}$ stand for the approximate resolved scale solution and the subgrid velocity field, respectively, at iteration i; *maxiter* is the maximum number of iterations and *tol* is the prescribed tolerance.

Thus, the desired resolved solution u_H is obtained when the convergence of (Algorithm 1) is attained. At each iteration, a linear system of the form $Ax = b$, with A nonsymmetric, has to be solved in order to determine u_H^i (*step 12*). The system dimension is equal to the number of degrees freedom of the mesh $\mathcal{T}_h$. As mentioned before, this is the method major drawback and may be overcome by applying the strategy proposed in the next section.

3. APPROXIMATING THE SUBGRID EFFECTS

In this section we propose a new implementation strategy for the NSGS method, in which the non-linear formulation (9) is solved for the coarse mesh $\mathcal{T}_H$. Let $\tilde{u}_H$ be the approximation of the resolved scale solution u_H. $\tilde{u}_H$ is obtained by locally approximating the effects of each

Algorithm 1

```
Require: This algorithm receive as input $u_H^0$ - calculated using SGS with $c_b = 1$
Ensure: The output is the approximate solution $u_H$
for $(each\ element\ T_h \in \mathcal{T}_h)$ do
    $c_{T_h}^0 \leftarrow c_b$
end for
$i \leftarrow 0$
repeat
    $i \leftarrow i + 1$
    for $(each\ element\ T_h \in \mathcal{T}_h)$ do
        determine $\boldsymbol{\beta}_h^{H;i-1}$
        $c_{T_h}^i \leftarrow \frac{1}{2}\left[c_{T_h}^{i-1} + \frac{1}{2}\left|\boldsymbol{\beta}_h^{H;i-1}\right|\right]$
        $c_{b,T_h} \leftarrow c_{T_h}^i$
    end for
    determine $u_H^i$
until $\left(i \leq maxiter\ or\ \max_{j=1,\dots,dof}\left|u_{H;j}^i - u_{H;j}^{i-1}\right| \leq tol\right)$
```

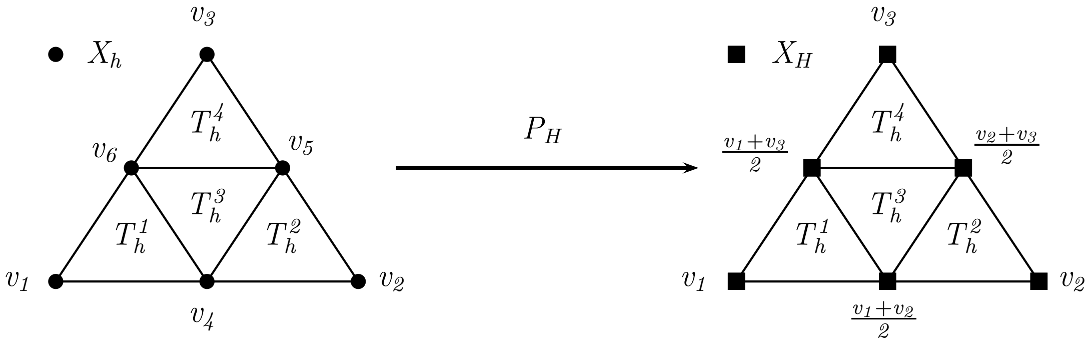


Figure 2: Macro element $T \in \mathcal{T}_H$ and its corresponding micro elements T_h^1, T_h^2, T_h^3 and $T_h^4 \in \mathcal{T}_h$: relationship among the degrees of freedom of $\mathcal{T}_H$ and $\mathcal{T}_h$, due to the projection operator P_H.

subgrid degree of freedom belonging to $T \in \mathcal{T}_H$. To show how this approximation is performed, let us first notice that both linear and linearized operators in (13) can be built by summing up the contributions of each macro element $T \in \mathcal{T}_H$. Each 6×6 local matrix results from assembling the contributions of its corresponding micro elements $T_h^i \subset T$, $i = 1, 2, 3, 4$ (see Fig. 2). Moreover, the degrees of freedom of the micro elements are connected to the resolved scale nodes of the related macro element due to the linearized operator $D(u_H^{n-1}; u_h^{H;n}, v_h^H)$, which prevents the use of an edge based structure. This can be clearly seen by rewriting it in the following way:

$$\begin{aligned} \sum_{T \in \mathcal{T}_H} \int_T \xi(u_H^{n-1}) \boldsymbol{\nabla} u_h^{H;n} \cdot \boldsymbol{\nabla} v_h^H d\Omega &= \sum_{T \in \mathcal{T}_H} \int_T \xi(u_H^{n-1}) \boldsymbol{\nabla}(u_h^n - u_H^n) \cdot \boldsymbol{\nabla}(v_h - v_H)\, d\Omega \\ &= \sum_{T \in \mathcal{T}_H} \Big(I - II - III + IV \Big), \end{aligned} \tag{14}$$

where

$$\begin{aligned} I &= \int_T \xi(u_H^{n-1}) \boldsymbol{\nabla} u_h^n \cdot \boldsymbol{\nabla} v_h d\Omega; & II &= \int_T \xi(u_H^{n-1}) \boldsymbol{\nabla} u_h^n \cdot \boldsymbol{\nabla} v_H d\Omega; \\ III &= \int_T \xi(u_H^{n-1}) \boldsymbol{\nabla} u_H^n \cdot \boldsymbol{\nabla} v_h d\Omega \quad \text{and} & IV &= \int_T \xi(u_H^{n-1}) \boldsymbol{\nabla} u_H^n \cdot \boldsymbol{\nabla} v_H d\Omega. \end{aligned}$$

The term I is similar to the diffusive term coming from the bilinear form $B(u_h^n, v_h)$. In the other three terms (II, III, IV) the $\mathcal{T}_H$ resolved degrees of freedom are coupled with the $\mathcal{T}_h$ degrees of freedom. For each $T_h^i \subset T$, $i = 1, 2, 3, 4$, the former are obtained by projecting the latter into the macro element T using the projection operator P_H (Fig. 2).

After assembling the contributions of each micro element $T_h \in T$, the local problem associated with each $T \in \mathcal{T}_H$ may be defined as

$$K^T U^T = F^T \,. \tag{15}$$

The local vector of unknowns $U^T = \{u_{h;1}^T, u_{h;2}^T, u_{h;3}^T, u_{h;4}^T, u_{h;5}^T, u_{h;6}^T\}^t$ may also be written as $U^T = \{u_{H;1}^T, u_{H;2}^T, u_{H;3}^T, u_{h;4}^T, u_{h;5}^T, u_{h;6}^T\}^t$, since u_h and u_H coincide in the coarse scale nodes. The local system (15) can be partitioned in the following way

$$\begin{bmatrix} \mathcal{A} & \mathcal{B} \\ \mathcal{C} & \mathcal{D} \end{bmatrix} \begin{bmatrix} \mathcal{U}_1 \\ \mathcal{U}_2 \end{bmatrix} = \begin{bmatrix} \mathcal{F}_1 \\ \mathcal{F}_2 \end{bmatrix} \tag{16}$$

where $\mathcal{U}_1 = \{u_{H;1}^T, u_{H;2}^T, u_{H;3}^T\}^t$ and $\mathcal{U}_2 = \{u_{h;4}^T, u_{h;5}^T, u_{h;6}^T\}^t$. This local system shows quite well the coupling of the resolved scale and the subgrid nodes. Although we are interested in the resolved scale solution, such coupling should not be disregard. Besides, such local connection spreads to the macro element neighbors due to the support of the interpolation functions (*two-level* $\mathbb{P}_1$ *setting*). This property prevents the use of an EBE and EDS data structure. However, we may approximate the problem by disregarding the global subgrid connectivity and performing a static condensation of the unknowns $\mathcal{U}_2$ at each macro element level. Thus, assuming that the matrix $\mathcal{D}$ is nonsingular, the condensed local problem becomes

$$\mathcal{K}^T \mathcal{U}_1^T = \mathcal{F}^T \tag{17}$$

where $\mathcal{K}^T = (\mathcal{A} - \mathcal{B}\mathcal{D}^{-1}\mathcal{C})$ is the 3×3 local matrix which approximates the macro element matrix of $T \in \mathcal{T}_H$. The respective local vector is $\mathcal{F}^T = (\mathcal{F}_1 - \mathcal{B}\mathcal{D}^{-1}\mathcal{F}_2)$. After assembling the contributions of all $T \in \mathcal{T}_H$, the following new global linear system is obtained

$$\mathcal{K}\tilde{\mathcal{U}} = \mathcal{F} \,, \tag{18}$$

where

$$\mathcal{K} = \mathop{\mathbf{A}}_{T=1}^{nel} \mathcal{K}^T \; ; \qquad \mathcal{F} = \mathop{\mathbf{A}}_{T=1}^{nel} \mathcal{F}^T \; ; \qquad \tilde{\mathcal{U}} = \mathop{\mathbf{A}}_{T=1}^{nel} \mathcal{U}_1^T \; . \tag{19}$$

In this expression, nel is the total number of macro elements of the coarse mesh $\mathcal{T}_H$. The linear system (18) is then solved for the resolved scale unknown vector $\tilde{\mathcal{U}} = \{\tilde{u}_{H;1}, \ldots, \tilde{u}_{H;\mathrm{Neq}_H}\}^t$, where Neq_H is the total number of resolved scale nodes. Such approach yields a remarkable decrease of the computational cost, which can also be improved by using an EBE or EDS data structure.

The resulting methodology can be seen as an approximation of the original NSGS method. Although lacking a full mathematical explanation, the numerical results are promising, as will be discussed in the following.

4. ELEMENT-BASED AND EDGE-BASED STRUCTURES

Element-based and Edge-based structures have been extensively used in finite element implementations, resulting in considerable improvements comparing to standard implementations. The success of this solution strategy requires an efficient implementation of matrix-vector products and the choice of a suitable preconditioner. Generally, the edge-based data structure reduces processing time and requires around one half of the storage area to hold the coefficient matrix when compared to an enhanced element-based implementation. We perform an implementation of the NSGS method using element-based and edge-based implementations.

The conventional finite element data structure associates to each triangle e is its connectivity, that is, the mesh nodes I, J and K. In the edge-based data structure each edge s is associated to the adjacent elements e and f, thus to the nodes I, J, K and L, as shown in Figure 3.

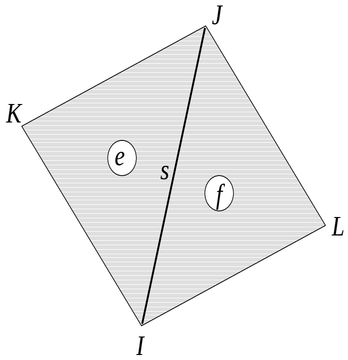


Figure 3: Elements adjacent to edge *s*, formed by nodes *I* e *J*.

Each element matrix can be disassembled into its contributions to three edges, s, $s+1$ and $s+2$, with connectivities IJ, JK and KI, that is,

$$\underbrace{\begin{bmatrix} \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet \end{bmatrix}}_{\text{element } e} = \underbrace{\begin{bmatrix} \times & \times & 0 \\ \times & \times & 0 \\ 0 & 0 & 0 \end{bmatrix}}_{\text{edge } s} + \underbrace{\begin{bmatrix} 0 & 0 & 0 \\ 0 & \times & \times \\ 0 & \times & \times \end{bmatrix}}_{\text{edge } s+1} + \underbrace{\begin{bmatrix} \times & 0 & \times \\ 0 & 0 & 0 \\ \times & 0 & \times \end{bmatrix}}_{\text{edge } s+2} , \tag{20}$$

where • and × are matrix coefficients defined from (13). Thus, all the contributions belonging to edge s will be present in the adjacent elements e and f. The resulting edge matrix is the sum of the corresponding sub-element matrices containing all the contributions to nodes I and J, that is,

$$\underbrace{\begin{bmatrix} \circ & \circ \\ \circ & \circ \end{bmatrix}}_{\text{edge } s} = \underbrace{\begin{bmatrix} \times & \times \\ \times & \times \end{bmatrix}}_{\text{element } e} + \underbrace{\begin{bmatrix} \times & \times \\ \times & \times \end{bmatrix}}_{\text{element } f} . \tag{21}$$

Considering a conventional elementwise description of a given finite element mesh, the topological informations are manipulated, generating a new edge-based mesh description. Thus, the assembled global matrix given in equation (19) may be written now as,

$$\mathcal{K} = \mathop{\mathbf{A}}_{s=1}^{nedges} \mathcal{K}^s \tag{22}$$

where $nedges$ is the total number of edges of the macro mesh $\mathcal{T}_H$. The edge matrix $\mathcal{K}^s$ is obtained from the contributions of all the element matrices $\mathcal{K}^T$ that share the edge s.

In the element by element (EBE) implementation strategy, the coefficients of the global matrix are stored in each macro element matrix as defined by (19). The global matrix $\mathcal{K}$ is stored in a compact form of size $nel \times 9$. On the other hand, in the edge-based (EDS) strategy the coefficients of the global matrix, defined by (22), are also stored in a compact form of size $nedge \times 4$.

5. NUMERICAL RESULTS

In this section we evaluate the numerical performance of the proposed Condensed NSGS on some academic test cases. In all of them the computational domain $\Omega = (0,1) \times (0,1)$ is discretized by using triangular meshes with 20×20, 40×40 and 80×80 cells, where each cell is subdivided in two triangles (macro elements). Besides comparing the resolved solution in the lights of the condensed and the original NSGS, we investigate the efficiency of both EBE and EDS implementations of the new methodology. The tolerance of the GMRES algorithm is 10^{-7}. It is used 50 vectors in the Krylov basis in examples 1 and 2 and 10 vectors in the example 3. The convergence of the NSGS method is attained for a prescribed tolerance ($tol = 10^{-2}$ to the examples 1 and 2, and $tol = 10^{-4}$ to the example 3) or $maxiter = 30$.

5.1 Example 1: Advection-diffusion problem

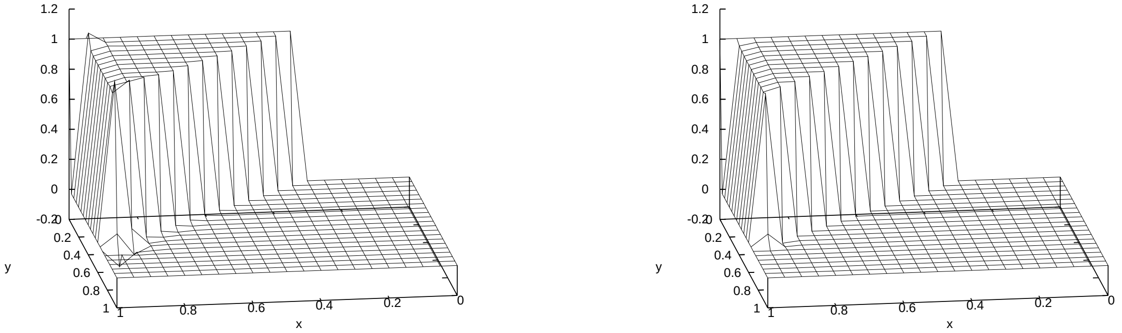
This example simulates a two-dimensional advection dominated advection-diffusion problem with $\epsilon = 10^{-12}$, $\boldsymbol{\beta} = (1,1)$ and $f = \sigma = 0$. The Dirichlet boundary conditions are given by

$$u(0,y) = u(1,y) = u(x,1) = 0 \quad \text{and} \quad u(x,0) = \begin{cases} 1, & x \leq 0.3\ ; \\ 0, & x > 0.3. \end{cases}$$

These conditions yield a solution with an internal layer in the direction of the velocity field starting at $(0.3, 0.)$ and an exponential external layer at $x = 1$.

Figure 4 shows the standard NSGS solution and the EBE/EDS-based condensed NSGS solution using the coarsest mesh (20×20). Thus, the mesh $\mathcal{T}_H$ is composed by 800 macro elements with 361 unknowns, while the mesh $\mathcal{T}_h$ is formed by 3200 elements with 1521 unknowns. Note that the EBE/EDS-based condensed NSGS solution, obtained on $\mathcal{T}_H$, is quite accurate.

Table 1 presents the computational performance of the element-based (EBE) and edge-based (EDS) data structures, respectively, for the condensed NSGS method. In Tab. 1, Neq$_H$ is the number of unknowns related to coarse mesh and the CPU times are reported. The edge-based approach is about 60% faster than the element-based one. For $\mathcal{T}_H$ with 20×20 and 40×40, the CPU times for the standard NSGS (resolved for $\mathcal{T}_h$, direct solver) is 196.6 and 13814.75 seconds, respectively. Thus, either using EBE or EDS, remarkable computational improvements are obtained with the condensed NSGS strategy.



(a) standard NSGS

(b) EBE/EDS-based condensed NSGS

Figure 4: Example 1 - NSGS method solutions.

Table 1: Example 1: Computational costs - EBE and EDS structures

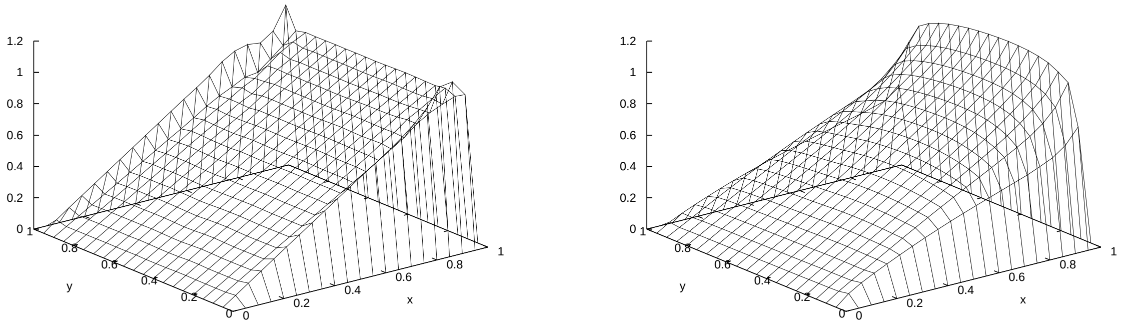
Mesh($\mathcal{T}_H$)	Neq$_H$	Time (s)	
		EBE	EDS
20×20	361	6.36	3.79
40×40	1521	55.82	33.63
80×80	6241	243.27	148.65

5.2 Example 2: Advection-diffusion problem with source term

This example simulates a two-dimensional advection dominated advection-diffusion problem with $\epsilon = 10^{-9}$, $\boldsymbol{\beta} = (1, 0)$, $\sigma = 0$ and a constant source term $f = 1$. We set $u = 0$ on all boundary so that the exact solution is a 45^o slope, possessing parabolic layers at $y = 0$, $y = 1$ and an exponential layer at $x = 1$.

The standard NSGS solution and the edge-based condensed NSGS solution using the mesh with 20×20 cells are shown in Fig. 5. The standard NSGS method presents some oscillations in the neighborhood of the parabolic layers ($y = 0$ and $y = 1$). The condensed NSGS solution presents a diffusive behavior in the these layers and a small overshoot in $x = 1$. However, there is no oscillation at the externals layers.

Table 2 presents the computational performance for the condensed NSGS method considering both data structures. The edge-based approach is about 60% faster than the element-based one, as in the example 1. For $\mathcal{T}_H$ with 20×20 and 40×40, the CPU times for the standard NSGS (resolved for $\mathcal{T}_h$, direct solver) is 128.79 and 13873.37 seconds, respectively.



(a) standard NSGS

(b) EBE/EDS-based condensed NSGS

Figure 5: Example 2 - NSGS method solutions.

Table 2: Example 2: Computational costs: EBE and EDS structures

Mesh($\mathcal{T}_H$)	Neq$_H$	Time (s)	
		EBE	EDS
20×20	361	3.23	2.11
40×40	1521	94.22	55.11
80×80	6241	2236.63	1450.08

5.3 Example 3: Diffusion - reaction problem

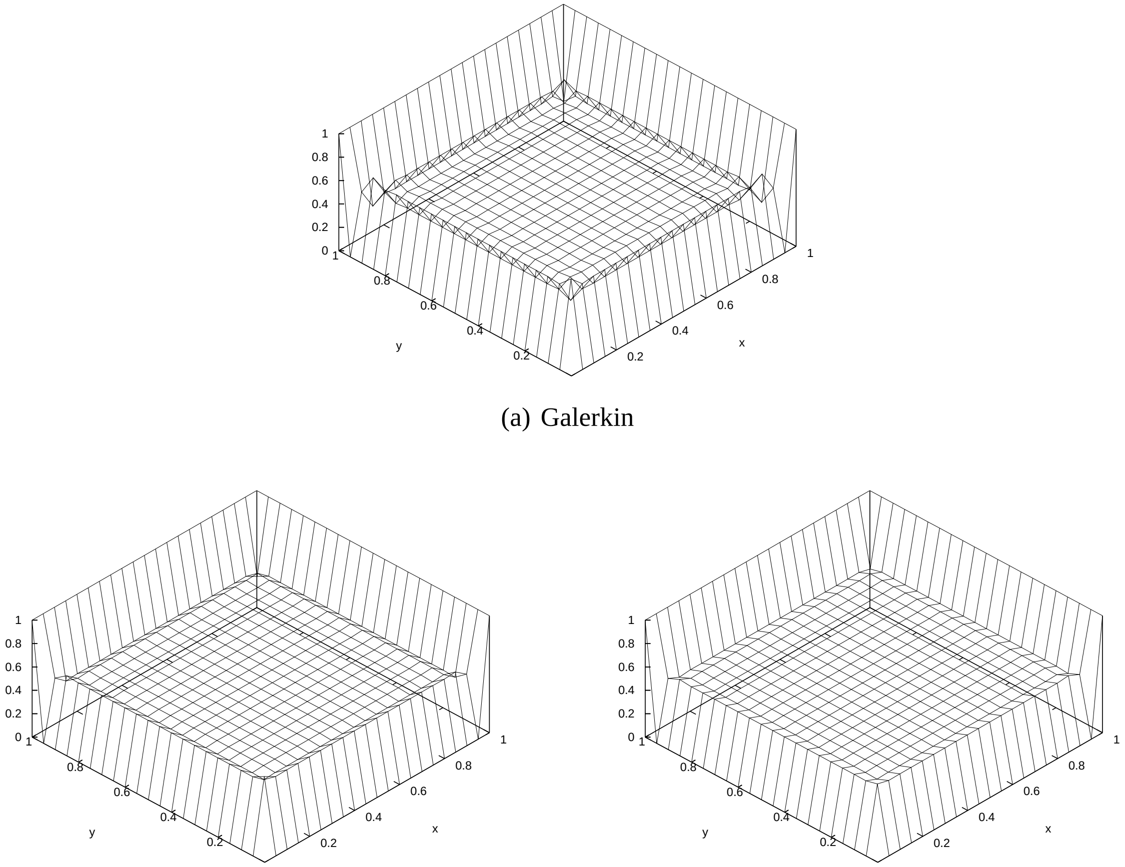
In this example, we consider the singular perturbed case where $\epsilon = 10^{-12}$, $\sigma = 1$ and $f = 0.5$, with the following boundary conditions: $u(x, 0) = u(0, y) = 0$ and $u(x, 1) = u(1, y) = 1$.

The Galerkin solution obtained on macro mesh 20×20 is shown in Fig. 6(a), which presents some locallized oscillations in the neighborhood of the external layers. Fig. 6(b) and 6(c) show the corresponding standard and condensed NSGS solution, respectively. Although some oscillations still remain at external layers, the edge-based condensed NSGS solution is more accurate than the NSGS standard one.

The computational performance for the condensed EBE and EDS based NSGS implementation are presented in Tab. 3. For this example the element-based data structure is more effective than the edge-based, that is, the EBE CPU time for all meshes is smaller than the EDS CPU time. Usually, EDS is more effective than EBE due to the matrix-vector product float-point operations. However, as the number of the GMRES iterations is very small in the present example, the solution required fewer number of matrix-vector products float-point operations. Thus, the computational costs are mainly concentrated on the generation of the edge-based mesh.

Table 3: Example 3: Computational costs: EBE and EDS structures

Mesh($\mathcal{T}_H$)	Neq(H)	Time (s)	
		EBE	EDS
20×20	361	0.07	0.75
40×40	1521	2.78	2.82
80×80	6241	11.58	11.37



(a) Galerkin

(b) standard NSGS

(c) EBE/EDS-based condensed NSGS

Figure 6: Example 3 - Galerkin and NSGS methods solutions .

6. CONCLUSIONS

We propose a new implementation strategy for the Nonlinear Subgrid Stabilization (NSGS) method. The NSGS method is a free parameter two-level subgrid method with good stability and convergence properties, owing to a subgrid viscosity that acts only on the unresolved scales. It requires the solution of linear systems associated with a mesh with characteristic length $h = H/2$ to obtain a solution whose resolution is H. The Condensed NSGS method proposed here introduces a procedure to overcome this drawback. This is performed by locally approximating the effects of the unresolved scale, yielded by a static condensation of the not null subgrid degrees of freedom at the macro element level. Then, the Condensed NSGS method can be seen as an approximation of the original NSGS method. Although lacking a full mathematical explanation, such simple procedure results in a remarkable computational gain. Moreover, some numerical experiments indicate that it yields extra regularization. Since the resolved scale solution is obtained directly for the macro finite element mesh, we incorporate the EBE and EDS data structures, yielding additional computational efficiency.

Contrasting other techniques that use tune-up parameters to improve accuracy, the present free Condensed EBE/EDS NSGS method combines simplicity with computational efficiency. Some important issues associated with the convergence of the nonlinear procedure, the mathematical and numerical analysis, application to nonlinear and transient problems remain to be investigated in forthcoming works.

Acknowledgements

The first author would like to thank the fellowship provided by the Brazilian Government, through the Agency CNPq/Brazil Pos-Doc Junior (PDJ) 151649/2007-3. The second and third authors acknowledge the financial support from the Agency CNPq/Brazil, contract/grant numbers 620165/2006-5 and 305730/2007-0, respectively.

REFERENCES

Brezzi, F. & Marini, L. D., 2002. Augmented spaces, two-level methods and stabilizing subgrids. *International Journal for Numerical Methods in Fluids*, vol. 40, pp. 31–46.

Catabriga, L. & Coutinho, A., 2002. Implicit SUPG solution of Euler equations using edge-based data structures. *Computer Methods in Applied Mechanics and Engineering*, vol. 191, pp. 3477–3490.

Catabriga, L., Martins, M. A. D., Coutinho, A. L. G. A., & Alves, J. L. D., 1998. Clustered edge-by-edge preconditioners for non-symmetric finite element equations. In Onate, E. & Idelsohn, S. R., eds, *Computational Mechanics: New Trends and Applications*, pp. 1–14, Buenos Aires, Argentina. 4th World Congress on Computational Mechanics, CD-ROM.

Guermond, J.-L., 1999. Stabilization of galerkin approximations of transport equation by subgrid modeling. *Mathematical Modelling and Numerical Analysis*, vol. 33, pp. 1293–1316.

Guermond, J.-L., 2001. Subgrid stabilization of galerkin approximations of linear monotone operators. *IMA Journal of Numerical Analysis*, vol. 21, pp. 165–197.

Layton, W. J., 2002. A connection between subgrid scale eddy viscosity and mixed methods. *Appl. Math. and Comput.*, vol. 133, pp. 147–157.

Saad, Y., 1995. *Iterative methods for sparse linear systems*. PWS Publishing Company.

Santos, I. P. & Almeida, R. C., 2007. A nonlinear subgrid method for advection-diffusion problems. *Computer Methods in Applied Mechanics and Engineering*, vol. 196, pp. 4771–4778.

Santos, I. P. & Almeida, R. C., 2008. A free parameter subgrid scale model for transport problems. *Computer Methods in Applied Mechanics and Engineering*.