A Two-Level Variational Multiscale Method for Convection-Diffusion Equations

Volker John [1]

Institut für Analysis und Numerik, Otto-von-Guericke-Universität Magdeburg, Postfach 4120, 39016 Magdeburg, Germany

Songul Kaya [2]

Department of Mathematics, Illinois Institute of Technology, Chicago, IL 60616, U.S.A.

William Layton [3]

Department of Mathematics, University of Pittsburgh, Pittsburgh, PA 15260, U.S.A.

Abstract

This paper studies the error in, the efficient implementation of and time stepping methods for a variational multiscale method (VMS) for solving convection-dominated problems. The VMS studied uses a fine mesh C^0 finite element space X^h to approximate the concentration and a coarse mesh discontinuous vector finite element space L^H for the large scales of the flux in the two scale discretization. Our tests show that these choices lead to an efficient VMS whose complexity is further reduced if a (locally) L^2-orthogonal basis for L^H is used. A fully implicit and a semi-implicit treatment of the terms which link effects across scales are tested and compared. The semi-implicit VMS was much more efficient. The observed global accuracy of the most straightforward VMS implementation was much better than the artificial diffusion stabilization and comparable to a streamline-diffusion finite element method in our tests.

Key words: convection-dominated convection-diffusion equation, variational multiscale method, two-level method, efficient implementation

[1] corresponding author, email: john@mathematik.uni-magdeburg.de, http://www-ian.math.uni-magdeburg.de/home/john/, phone +49 391 67 12633, fax +49 391 67 18073

[2] kaya@iit.edu, http://www.iit.edu/~kaya

[3] wjl@pitt.edu, www.math.pitt.edu/~wjl, partially supported by NSF Grant

1 The variational multiscale method

We consider a time-dependent scalar convection-diffusion equation

$$\begin{aligned} u_t - \varepsilon\Delta u + \mathbf{b}\cdot\nabla u + cu &= f && \text{in } (0,T]\times\Omega \\ u &= 0 && \text{on } [0,T]\times\partial\Omega \\ u(0,\mathbf{x}) &= u_0(\mathbf{x}) && \text{in } \Omega. \end{aligned} \tag{1}$$

Here, $\Omega\subset\mathbb{R}^d$, $d\in\{2,3\}$, is a bounded polyhedral domain. The functions $\mathbf{b}\in(L^\infty(0,T;L^\infty(\Omega)))^d$, $c\in L^\infty(0,T;L^\infty(\Omega))$ with $c(\mathbf{x},t)\ge 0$, $f\in L^2(0,T;L^2(\Omega))$, $u_0(\mathbf{x})\in H_0^1(\Omega)$ and the constant $\varepsilon>0$ are given. The use of homogeneous Dirichlet boundary conditions is only for convenience of presentation. Non-homogeneous Dirichlet boundary conditions are considered in the numerical studies. Let $X=H_0^1(\Omega)$ and let $(\cdot,\cdot)$ denote the $L^2(\Omega)$-inner product. The variational solution of (1) is a strongly differentiable map: $u:[0,T]\to X$ satisfying $u(0,\mathbf{x})=u_0(\mathbf{x})\in X$ and

$$(u_t,v)+a(u,v)=(f,v)\quad\forall\, v\in X, \tag{2}$$

where

$$a(u,v)=(\varepsilon\nabla u,\nabla v)+(\mathbf{b}\cdot\nabla u+cu,v).$$

We consider the case that ε is small compared to $\|\mathbf{b}\|_{(L^\infty(0,T;L^\infty(\Omega)))^d}$. The convection-diffusion equation (1) above occurs in many practical problems in which the diffusion coefficient is very small compared to the velocity field $\mathbf{b}$ which drives the convection, precisely the case which is most difficult to solve accurately. The solution then contains many scales composed of a complex collection of boundary and interior layers. Usual (centered) finite element methods typically produce approximate solutions with large, non-physical oscillations unless either the mesh width h is globally small with respect to the diffusion coefficient ε or enough is known about the exact solution to generate Shishkin-like meshes which are locally small with respect to ε in all transition regions in a very precise sense, [29]. Thus, various stabilizations have proven to be essential computational tools. Recently, Hughes and co-workers [12,10] have developed the variational multiscale method (VMS) which is motivated by the inherent multiscale structure of the solution of (1). We consider in this paper a method which arose from related considerations [25] and is, in fact, a VMS. The method (3) below introduces global stabilization and then anti-diffuses these effects on the large scales of the solution. Thus, effective stabilization is retained only on the smallest resolved scales (in which the non-physical oscillations occur).

DMS0207 627

Let $\mathcal{T}^H(\Omega)$ be a conforming triangulation of Ω and let $\mathcal{T}^h(\Omega)$ be a refinement of $\mathcal{T}^H(\Omega)$ or $\mathcal{T}^h(\Omega) = \mathcal{T}^H(\Omega)$. The finite element approximation of the solution of (2) is sought in the conforming finite element space $X^h \subset X$. Let L^H denote a vector-valued finite element subspace of $(L^2(\Omega))^d$. The discretization we study adds additional diffusion acting on all discrete scales and then antidiffuses on the scales resolvable on $\mathcal{T}^H(\Omega)$ as follows: find $u^h : [0,T] \to X^h$, $\mathbf{g}^H : [0,T] \to L^H$ satisfying

$$\begin{aligned} (u_t^h, v^h) + a(u^h, v^h) \\ +(\varepsilon_{\text{add}}\nabla u^h, \nabla v^h) - (\varepsilon_{\text{add}}\mathbf{g}^H, \nabla v^h) &= (f, v^h) \ \forall\ v^h \in X^h \\ (\mathbf{g}^H - \nabla u^h, \mathbf{l}^H) = 0 & \qquad \forall\ \mathbf{l}^H \in L^H. \end{aligned} \tag{3}$$

Here, ε_{add} is a non-negative function depending on the mesh size h.

The second equation of (3) states that $\mathbf{g}^H = \overline{P}_H(\nabla u^h)$ where $\overline{P}_H$ is the L^2-orthogonal projection into L^H. Consider the case that ε_{add} is a non-negative constant. Using a simple orthogonality argument, [25], we find that $u^h : [0,T] \to X^h$ satisfies:

$$(u_t^h, v^h) + a(u^h, v^h) + (\varepsilon_{\text{add}}(I - \overline{P}_H)(\nabla u^h), (I - \overline{P}_H)\nabla v^h) = (f, v^h) \tag{4}$$

for all $v^h \in X^h$. Since L^H represents the large scales of the gradients, $(I - \overline{P}_H)\nabla u^h$ clearly represents the small fluctuations of ∇u^h. Thus, the method (4) introduces additional diffusion acting only on the fluctuating components of ∇u^h. In the case that ε_{add} is a constant, the Pythagorean theorem gives

$$(\varepsilon_{\text{add}}(I - \overline{P}_H)(\nabla u^h), (I - \overline{P}_H)\nabla v^h) = (\varepsilon_{\text{add}}\nabla u^h, \nabla v^h) - (\varepsilon_{\text{add}}\overline{\nabla u^h}, \overline{\nabla v^h}),$$

where

$$\overline{\nabla u^h} = \overline{P}_H(\nabla u^h).$$

Then, (4) can be rewritten in the form

$$(u_t^h, v^h) + a(u^h, v^h) + (\varepsilon_{\text{add}}\nabla u^h, \nabla v^h) - (\varepsilon_{\text{add}}\overline{\nabla u^h}, \overline{\nabla v^h}) = (f, v^h)\ \forall v^h \in X^h. \tag{5}$$

This paper studies algorithmic aspects of the two formulations (3) and (5). In both, the large scale space L^H must be chosen . If X^h is a higher order finite element space on a given mesh, one approach is to define the large scale space using lower order finite elements on the same mesh. The implementation of this choice was discussed in [17]. For low order elements, which are the only elements available in many codes, the only option is to define the large scale space L^H on a coarse mesh leading to a two-level discretization, considered herein. Low order elements are also the most common choice for diffusion-transport problems in geophysics because of the very large scales of the problems studied. The goal of this paper is to study efficient implementations of the two-level VMS idea and to delineate pros and cons of different time stepping methods for multiscale discretization.

Theoretical studies of this method began in [25] and were continued in [9,17,20–24,28]. We note that it is inspired by both physical ideas in turbulence modeling and algorithmic ideas developed for simulation of non-Newtonian fluids, [4]. It can also be thought of as a finite element realization of the method of spectral viscosity, e.g., see Maday and Tadmor [27] or Chen, Du and Tadmor [2].

Multiscale discretizations have recently attracted attention for the simulation of turbulent flows, [12,8,10,5–7]. The VMS idea is to use a variationally consistent discretization for the large scales and to stabilize only small scales. Equivalently, to add stabilization which accounts only for the effects of the unresolved solution scales upon the smallest resolved scales in the approximate solution. The first realization of this approach tested in, e.g., [11,14,6,7], uses standard finite element spaces for the large scale velocity/pressure and bubble functions to model the small scales. The implementation of this approach is straightforward because the bubble functions vanish on every face of the mesh cells, so their contribution to the global stiffness matrix can be eliminated by static condensation. On the other hand, this choice imposes a constraint for computational convenience rather than physical fidelity on the small scales that they must vanish on all mesh cell boundaries. Thus, we are consider herein a more complex model for fluctuations. With extra complexity, the question of its computational difficulty and implementation becomes more important.

2 Algorithmic aspects

We will start by considering the formulation (3) of the VMS. As discretization in time, an implicit θ-scheme is applied. This leads in the discrete time t_k to the following fully discrete equations

$$
\begin{aligned}
&(u_k^h, v^h) + \theta_1 \Delta t_k \Big[((\varepsilon + \varepsilon_{\text{add}}) \nabla u_k^h, \nabla v^h) + (\mathbf{b} \cdot \nabla u_k^h + c u_k^h, v^h) - (\varepsilon_{\text{add}} \mathbf{g}_k^H, \nabla v^h) \Big] \\
&\quad = (u_{k-1}^h, v^h) - \theta_2 \Delta t_k \Big[((\varepsilon + \varepsilon_{\text{add}}) \nabla u_{k-1}^h, \nabla v^h) + (\mathbf{b} \cdot \nabla u_{k-1}^h + c u_{k-1}^h, v^h) \\
&\qquad - (\varepsilon_{\text{add}} \mathbf{g}_{k-1}^H, \nabla v^h) \Big] + \theta_3 \Delta t_k (f_{k-1}, v^h) + \theta_4 \Delta t_k (f_k, v^h) \quad \forall v^h \in V^h \qquad (6) \\
&\qquad (\mathbf{g}_k^H - \nabla u_k^h, \mathbf{l}^H) = 0 \quad \forall \mathbf{l}^H \in L^H.
\end{aligned}
$$

Here, $\Delta t_k = t_k - t_{k-1}$. Different choices of the parameters $\theta_1, \ldots, \theta_4$ give different time stepping schemes, see Table 1. The parameters in the fractional-step θ-scheme are given by

$$\theta = 1 - \frac{\sqrt{2}}{2}, \quad \tilde{\theta} = 1 - 2\theta, \quad \tau = \frac{\tilde{\theta}}{1-\theta}, \quad \eta = 1 - \tau.$$

Table 1
Implicit θ-schemes

	θ_1	θ_2	θ_3	θ_4	t_{k-1}	t_k	Δt_k
backward Euler	1	0	0	1	t_{n-1}	t_n	Δt_n
Crank-Nicolson	0.5	0.5	0.5	0.5	t_{n-1}	t_n	Δt_n
fractional-step, step 1	$\tau\theta$	$\eta\theta$	$\eta\theta$	$\tau\theta$	t_{n-1}	$t_{n-1}+\theta\Delta t_n$	$\theta\Delta t_n$
step 2	$\eta\tilde{\theta}$	$\tau\tilde{\theta}$	$\tau\tilde{\theta}$	$\eta\tilde{\theta}$	$t_{n-1}+\theta\Delta t_n$	$t_n-\theta\Delta t_n$	$\tilde{\theta}\Delta t_n$
step 3	$\tau\theta$	$\eta\theta$	$\eta\theta$	$\tau\theta$	$t_n-\theta\Delta t_n$	t_n	$\theta\Delta t_n$

We consider for convenience of presentation the two-dimensional case. The same ideas can be applied in a straightforward way to three-dimensional convection-diffusion equations. The finite element spaces are equipped with bases:

$$\begin{aligned} X^h &= \operatorname{span}\{\phi_i^h\}, \ i = 1, \ldots, N_X, \\ L^H &= \operatorname{span}\left\{ \begin{pmatrix} \psi_i^H \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ \psi_i^H \end{pmatrix} \right\}, \ i = 1, \ldots, N_L. \end{aligned} \tag{7}$$

Then, the algebraic representation of (6) looks as follows

$$\begin{aligned} \mathcal{A} \begin{pmatrix} u_k \\ g_{k,1} \\ g_{k,2} \end{pmatrix} &= \begin{pmatrix} M + \theta_1 \Delta t_k A & \theta_1 \Delta t_k B_1 & \theta_1 \Delta t_k B_2 \\ C_1 & D & 0 \\ C_2 & 0 & D \end{pmatrix} \begin{pmatrix} u_k \\ g_{k,1} \\ g_{k,2} \end{pmatrix} \\ &= \begin{pmatrix} \theta_3 \Delta t_k f_{k-1} + \theta_4 \Delta t_k f_k \\ 0 \\ 0 \end{pmatrix} \\ &\quad + \begin{pmatrix} M - \theta_2 \Delta t_k A & -\theta_2 \Delta t_k B_1 & -\theta_2 \Delta t_k B_2 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} u_{k-1} \\ g_{k-1,1} \\ g_{k-1,2} \end{pmatrix}, \end{aligned} \tag{8}$$

where

$$
\begin{aligned}
M &= (\phi_j^h, \phi_i^h)_{i,j=1,\dots,N_X}\,,\\
A &= (((\varepsilon+\varepsilon_{\text{add}})\nabla\phi_j^h, \nabla\phi_i^h) + (\mathbf{b}\cdot\nabla\phi_j^h + c\phi_j^h, \phi_i^h))_{i,j=1,\dots,N_X}\,,\\
B_1 &= -(\varepsilon_{\text{add}}\psi_j^H, (\phi_i^h)_x)_{i=1,\dots,N_X, j=1,\dots,N_L}\,,\\
B_2 &= -(\varepsilon_{\text{add}}\psi_j^H, (\phi_i^h)_y)_{i=1,\dots,N_X, j=1,\dots,N_L}\,,\\
C_1 &= -(\psi_i^H, (\phi_j^h)_x)_{i=1,\dots,N_L, j=1,\dots,N_X}\,,\\
C_2 &= -(\psi_i^H, (\phi_j^h)_y)_{i=1,\dots,N_L, j=1,\dots,N_X}\,,\\
D &= (\psi_j^H, \psi_i^H)_{i,j=1,\dots,N_L}\,.
\end{aligned}
$$

Note, the blocks B_1, B_2 have to be scaled in the same way in (8) as the block A since the additional diffusion ε_{add} has to be the same in all of these blocks.

The matrix blocks M, A and D are sparse since they are build from inner products with finite element functions from only one space. Thus, the sparsity of these blocks is a standard property. However, the sparsity of the matrix blocks B_1, B_2, C_1, C_2 depends heavily on the choice of L^H. An inner product defining an entry of these matrices does not vanish if the intersection of the support of the two factors has a positive measure

$$
\text{meas}(\text{supp}(\psi_i^H) \cap \text{supp}(\phi_j^h)) > 0, \quad \psi_i^H \in L^H, \phi_j^h \in X^h.
$$

The number of non-zero entries connected to the basis function ψ_i^H becomes the smaller, the smaller the support of ψ_i^H is. The smallest possible support is one mesh cell on $\mathcal{T}^H(\Omega)$. This can be realized if L^H is a discontinuous finite element space.

The fully implicit VMS introduces $2N_L$ additional equations for the unknowns $g_{k,1}, g_{k,2}$. A way to avoid this problem is to use a semi-implicit version of (6)

$$
\begin{aligned}
&(u_k^h, v^h) + \theta_1\Delta t_k\Big[((\varepsilon+\varepsilon_{\text{add}})\nabla u_k^h, \nabla v^h) + (\mathbf{b}\cdot\nabla u_k^h + cu_k^h, v^h)\Big]\\
&= (u_{k-1}^h, v^h) - \theta_2\Delta t_k\Big[((\varepsilon+\varepsilon_{\text{add}})\nabla u_{k-1}^h, \nabla v^h) + (\mathbf{b}\cdot\nabla u_{k-1}^h + cu_{k-1}^h, v^h)\Big]\\
&\quad +\Delta t_k(\varepsilon_{\text{add}}\mathbf{g}_{k-1}^H, \nabla v^h) + \theta_3\Delta t_k(f_{k-1}, v^h) + \theta_4\Delta t_k(f_k, v^h) \quad \forall v^h \in V^h \qquad (9)\\
&\quad (\mathbf{g}_{k-1}^H, \mathbf{l}^H) = (\nabla u_{k-1}^h, \mathbf{l}^H) \quad \forall \mathbf{l}^H \in L^H.
\end{aligned}
$$

In this semi-implicit version, the subtraction of the additional diffusion in the large scales is treated explicitly. The algebraic form of the second equation of the coupled system (9) is

$$\begin{pmatrix} D & 0 \\ 0 & D \end{pmatrix} \begin{pmatrix} g_{k-1,1} \\ g_{k-1,2} \end{pmatrix} = - \begin{pmatrix} C_1 u_{k-1} \\ C_2 u_{k-1} \end{pmatrix}. \tag{10}$$

If the mass matrix D of L^H is diagonal, that means iff the basis functions of L^H are L^2-orthogonal, the solution of this system is very simple. This property can be achieved easily for discontinuous finite element spaces, e.g., by using a basis consisting of Legendre polynomials. Inserting the solution of (10) into the first equation of (8) gives

$$\begin{aligned}(M + \theta_1 \Delta t_k A) u_k = & \theta_3 \Delta t_k f_{k-1} + \theta_4 \Delta t_k f_k + (M - \theta_2 \Delta t_k A) u_{k-1} \\ & + \Delta t_k B_1 D^{-1} C_1 u_{k-1} + \Delta t_k B_2 D^{-1} C_2 u_{k-1}. \end{aligned} \tag{11}$$

Note, the matrix A includes an additional diffusion in the diffusive term. Thus, the operator on the left hand side of (11) is stable if the amount of additional diffusion is sufficiently large. In addition, many standard solvers and preconditioners work well for such problems. The only difference to the simple artificial diffusion stabilization of a convection-diffusion equation consists in the last two terms on the right hand side of (11).

The summary of the algorithmic aspects of using the VMS (3) is as follows:

- the additional matrix blocks B_1, B_2, C_1, C_2 and D are needed, at least one dimension of these blocks is N_L,
- the sparsest structure of B_1, B_2, C_1, C_2 is achieved if L^H is a discontinuous finite element space,
- the fully implicit approach (8) requires the additional vectors $g_{k,1}, g_{k,2}$,
- the algebraic system (8) possesses $2N_L$ additional equations,
- the semi-implicit approach (11) can be implemented easily if the basis functions of L^H are L^2-orthogonal,
- the system matrix of (11) corresponds to a very stable operator.

Now, we will consider the formulation (5) of the VMS. The application of a implicit theta-scheme leads in each discrete time to a scalar equation of the form

$$\begin{aligned}(u_k^h, v^h) + \theta_1 \Delta t_k \Big[((\varepsilon + \varepsilon_{\text{add}}) \nabla u_k^h, \nabla v^h) + (\mathbf{b} \cdot \nabla u_k^h + c u_k^h, v^h) - (\varepsilon_{\text{add}} \overline{\nabla u^h}, \overline{\nabla v^h}) \Big] \\ = (u_{k-1}^h, v^h) - \theta_2 \Delta t_k \Big[((\varepsilon + \varepsilon_{\text{add}}) \nabla u_{k-1}^h, \nabla v^h) + (\mathbf{b} \cdot \nabla u_{k-1}^h + c u_{k-1}^h, v^h) \\ - (\varepsilon_{\text{add}} \overline{\nabla u^h}, \overline{\nabla v^h}) \Big] + \theta_3 \Delta t_k (f_{k-1}, v^h) + \theta_4 \Delta t_k (f_k, v^h) \quad \forall v^h \in V^h. \end{aligned} \tag{12}$$

In this equations, the large scales $\mathbf{g}^H$ are eliminated. But the term $(\varepsilon_{\text{add}}\overline{\nabla u^h}, \overline{\nabla v^h})$ couples the variables of X^h across the mesh cells of the coarse triangulation $\mathcal{T}^H(\Omega)$. Let ϕ_i^h, ϕ_j^h be two basis functions of X^h. Since $\text{supp}(\phi_i^h) \subset \text{supp}(\overline{\nabla \phi_i^h})$, the term $(\varepsilon_{\text{add}}\overline{\nabla \phi_j^h}, \overline{\nabla \phi_i^h})$ does not vanish if

$$\text{meas}(\text{supp}(\overline{\nabla \phi_i^h}) \cap \text{supp}(\overline{\nabla \phi_j^h})) > 0. \tag{13}$$

To minimize the number of non-vanishing terms of this form, the support of the projections has to be minimized. This is achieved by using a discontinuous finite element space for L^H. The algebraic representation of (12) is

$$(M+\theta_1\Delta t_k(A+B))u_k = \theta_3\Delta t_k f_{k-1} + \theta_4\Delta t_k f_k (M-\theta_2\Delta t_k(A+B))u_{k-1}. \tag{14}$$

This can be derived also from (8) by solving for $g_{k,1}, g_{k,2}$ which shows that

$$B = -B_1 D^{-1} C_1 - B_2 D^{-1} C_2.$$

However, the matrix B in (14) is not given in this product form but it is assembled directly by evaluating terms of the form

$$(B)_{ij} = (\varepsilon_{\text{add}}\overline{\nabla \phi_j^h}, \overline{\nabla \phi_i^h}). \tag{15}$$

This can be done in the following way:

1. Compute in a pre-processing step the matrix structure of B by checking (13).
2. If $(B)_{ij}$ is a member of this matrix structure, then compute $\overline{\nabla \phi_i^h}$, and $\overline{\nabla \phi_j^h}$. Taking the basis (7) and using the ansatz

$$\overline{\nabla \phi_i^h} = \sum_{j=1}^{N_L} \phi_j \psi_j^H,$$

   one gets for the evaluation of $\overline{\nabla \phi_i^h}$

$$\sum_{j=1}^{N_L} (\psi_j^H, \psi_k^H)\phi_j = (\nabla \phi_i^h, \psi_k^H), \quad k = 1, \ldots, N_L. \tag{16}$$

   The coefficients ϕ_j can be easily computed if the system matrix is diagonal, i.e. if the basis functions of L^H are L^2-orthogonal. But even if L^H is solely a discontinuous finite element space, (16) decouples in a number of small problems which can be solved in parallel.
3. Compute the inner product (15).

The dimension of B is $N_X \times N_X$. In comparison to the matrix blocks M and A, the block B creates a substantial fill-in.

The problem of having this substantial fill-in does not arise if a semi-implicit version of (12) is used

$$
\begin{aligned}
&(u_k^h, v^h) + \theta_1 \Delta t_k \Big[((\varepsilon + \varepsilon_{\text{add}}) \nabla u_k^h, \nabla v^h) + (\mathbf{b} \cdot \nabla u_k^h + c u_k^h, v^h) \Big] \\
&= (u_{k-1}^h, v^h) - \theta_2 \Delta t_k \Big[((\varepsilon + \varepsilon_{\text{add}}) \nabla u_{k-1}^h, \nabla v^h) + (\mathbf{b} \cdot \nabla u_{k-1}^h + c u_{k-1}^h, v^h) \Big] \\
&\quad + \Delta t_k (\varepsilon_{\text{add}} \overline{\nabla u^h}, \overline{\nabla v^h}) + \theta_3 \Delta t_k (f_{k-1}, v^h) + \theta_4 \Delta t_k (f_k, v^h) \quad \forall v^h \in V^h \qquad (17)
\end{aligned}
$$

To evaluate the term $(\varepsilon_{\text{add}} \overline{\nabla u^h}, \overline{\nabla v^h})$ on the right hand side, one can use the same approach as for computing the entries of B. The algebraic form of the semi-implicit system is like (11). Here, the additional terms on the right hand side of (11) are not computed by matrix-vector products but by computing the explicit projection of the finite element function and the appropriate inner products. Additional matrices are not necessary. However, the approach of assembling the matrix blocks B_1, B_2, C_1, C_2 and the computing of the matrix-vector products as in (11) seems to be easier to implement.

The algorithmic aspects for the VMS of form (5) are summarized as follows:

- for the fully implicit approach (14), the additional matrix B of dimension $N_X \times N_X$ is needed, this matrix is considerably less sparse than the other blocks,
- the sparsest structure of B is achieved if L^H is a discontinuous finite element space,
- the fully implicit approach (14) does not generate additional equations,
- matrix-vector products with the system matrix become considerably more expensive due to the additional fill-in of the matrix B,
- if L^H is a discontinuous finite element space, the computation of the projections decouples in a number of small systems which can be solved in parallel,
- for an easy computation of the projections, it is also advantageous if the basis functions of L^H are L^2-orthogonal.

The attractive semi-implicit strategy (9) and (17) has been studied in [1]. In [1], the method (17) was proven to be unconditionally stable. Concurrently, this method and stability results were obtained in a different context and to a different end by Johnson and Liu in [19]. A second order generalization is considered in [19]. Although a fully satisfactory stability proof for the second order extension is still open, preliminary analysis and computations in [19] look promising.

The finite element error in the energy norm for this semi-discrete VMS was analyzed in [9] for convection-diffusion equations and in [16] for the Navier-Stokes equations. Let $e = u - u^h$, one will obtain an optimal order of convergence for

$\|\nabla e\|_{L^2(0,T;L^2)}$ if the coarse space L^H is sufficiently fine and if u is sufficiently smooth. Denote by h and H the mesh widths for the fine and the coarse mesh, respectively, the fineness condition on L reads $h \sim H^\beta$ with $\beta \leq 2$. Beyond this, many theoretical problems are still open for (5).

3 Numerical studies

This section presents numerical studies comparing the fully implicit approach (6) and the semi-implicit approach (9). The algebraic representations of these approaches are given in (8) and (11), respectively. As mentioned above, the implementation of (6) and (9) seems to us easier than of (12) and (17). In addition, the basis equation (5) of (12) and (17) is equivalent to the VMS (3) only in the case of constant ε_{add}.

The numerical studies were carried out with the code MooNMD, [15,18]. The finite element space X^h consists of continuous piecewise linear or bilinear functions, i.e., $X^h = P_1$ on a triangular mesh $\mathcal{T}^h(\Omega)$ and $X^h = Q_1$ on a quadrilateral mesh $\mathcal{T}^h(\Omega)$. For the finite element space L^H on $\mathcal{T}^H(\Omega)$, we have used the simplest choice, namely piecewise constant functions. Note, this choice satisfies all conditions on L^H given in Section 2. The additional diffusion is chosen to be $\varepsilon_{\text{add}} = 0.1\ h$.

The efficiency of the solver for the algebraic systems is crucial for the efficiency of the whole numerical simulation. We applied a flexible GMRES method, [30], with a multigrid method as preconditioner. In the case of the scalar system (11), the preconditioner is a geometric multigrid method. We used the $F(1,1)$-cycle, ILU_β with $\beta = 1$ as smoother and the coarse grid system was solved directly by Gaussian elimination. Note, the ILU-decomposition has to be computed only once in the initial time step.

For the coupled system (8), an algebraic multigrid method was applied as preconditioner. Here, the $W(2,2)$-cycle was used, ILU_β with $\beta = 1$ as smoother and also as coarse grid solver. The algebraic multigrid method, which is described in [26], belongs to the class of aggregation methods, i.e. the unknowns of the coarse grid are defined by an appropriate clustering of the unknowns of the fine grid. A constant prolongation is applied and the restriction is defined to be the adjoint operator. The coarse grids and coarse grid matrices of the algebraic multigrid have to be constructed before the iterative solution of the linear system can start. For equi-distant time-steps, this has to be done only once at the first discrete time since the matrices are the same for all discrete times. The diagonal entry a_{ii}^H of the coarse grid matrix is the sum of all couplings of the fine grid nodes which are forming cluster i. The off diagonal entry $a_{ij}^H, i \neq j$, is the sum of all fine grid entries $a_{kl}^h, k \neq l$, where node k belongs to cluster i and node l to cluster j.

The numerical studies were performed for (1) with the prescribed solution

$$u(t, \mathbf{x}) = t^2 \cos(x_1 x_2^2), \tag{18}$$

with $\mathbf{x} = (x_1, x_2)$, $\varepsilon = 10^{-8}$, $\mathbf{b} = (2, -1)^T$, $c = 1$, $\Omega = (0,1)^2$ and $T = 10$. The non-homogeneous Dirichlet boundary conditions and the right hand side f were chosen such that $u(t, \mathbf{x})$ fulfills (1). We decided to use (18) as prescribed solution because the the Crank-Nicolson scheme is an exact time integrator in this case and all errors are due to the discretization in space and the stabilization of the convective term. The Crank-Nicolson scheme was applied with an equi-distant time step of length $\Delta t_n = 0.125$.

The initial quadrilateral grid, level 0, consists of four squares with edge length 0.5. As usual, the size of a mesh cells is the longest distance between two of its vertices and the mesh size is the maximum of the sizes of the mesh cells. Accordingly, the mesh size is $h_0 = \sqrt{2}\ 2^{-1}$. The initial triangular grid was obtained from the initial quadrilateral grid by dividing the squares using the diagonals from the left lower corner to the right upper corner. The initial grids were refined uniformly.

The computations have been carried out on a PC with Intel Pentium 4 Processor, with 3 GHz.

In the first numerical study, a fixed fine mesh $\mathcal{T}^h(\Omega)$ is considered and the coarse mesh $\mathcal{T}^H(\Omega)$ is varied. The computations were performed on a quadrilateral mesh. The fine mesh is given by refinement level 6 of the initial grid such that $h_6 = \sqrt{2}\ 2^{-7}$. The number of degrees of freedom (d.o.f.) on this mesh is 16 641 (including Dirichlet nodes). The coarse mesh is varied between level $L = 1$ and $L = 6$ giving $H = \sqrt{2}\ 2^{-(L+1)}$. The results are presented in Table 2.

In order to provide an impression on the accuracy of the computed results with the VMS, the results obtained with the simple artificial diffusion stabilization and the Streamline-Diffusion FEM (SDFEM) are also presented. In the artificial diffusion stabilization, the same parameter ε_{add} was used as in the VMS. The SDFEM has in each discrete time the form, see [13,3]: find $u_k^h \in X^h$ such that for all $v^h \in X^h$

$$
\begin{aligned}
&(u_k^h, v^h) + \sum_{K\in\mathcal{T}_h} \tau_K (u_k^h, \mathbf{b}\cdot\nabla v^h)_K + \theta_1 \Delta t_k \Big[(\varepsilon \nabla u_k^h, \nabla v^h) \\
&\quad + (\mathbf{b}\cdot\nabla u_k^h + c u_k^h, v^h) + \sum_{K\in\mathcal{T}_h} \tau_K (\mathbf{b}\cdot\nabla u_k^h + c u_k^h, \mathbf{b}\cdot\nabla v^h)_K \Big] \\
&= (u_{k-1}^h, v^h) + \sum_{K\in\mathcal{T}_h} \tau_K (u_{k-1}^h, \mathbf{b}\cdot\nabla v^h)_K - \theta_2 \Delta t_k \Big[(\varepsilon \nabla u_{k-1}^h, \nabla v^h) \\
&\quad + (\mathbf{b}\cdot\nabla u_{k-1}^h + c u_{k-1}^h, v^h) + \sum_{K\in\mathcal{T}_h} \tau_K (\mathbf{b}\cdot\nabla u_{k-1}^h + c u_{k-1}^h, b\cdot\nabla v^h)_K \Big] \\
&\quad + \theta_3 \Delta t_k (f_{k-1}, v^h) + \theta_3 \Delta t_k \sum_{K\in\mathcal{T}_h} \tau_K (f_{k-1}, \mathbf{b}\cdot\nabla v^h)_K \\
&\quad + \theta_4 \Delta t_k (f_k, v^h) + \theta_4 \Delta t_k \sum_{K\in\mathcal{T}_h} \tau_K (f_k, \mathbf{b}\cdot\nabla v^h)_K .
\end{aligned}
$$

Here, $(\cdot,\cdot)_K$ denotes the inner product in $L^2(K)$ where K is a mesh cell. The parameter τ_K is computed as follows

$$
\tau_K = \frac{\alpha_K h_K}{2\|\mathbf{b}\|_2}, \quad Pe_K = \frac{\|\mathbf{b}\|_2 h_K}{2\varepsilon}, \quad \alpha_K = \coth(Pe_K) - \frac{1}{Pe_K}.
$$

Here, h_K is the size of the mesh cell K and $\|\mathbf{b}\|_2$ is the Euclidean norm of $\mathbf{b}$.

Let $e = u - u^h$. The results presented in Table 2 show that the semi-implicit approach (9) is much more efficient than the fully implicit one (6). For the fully implicit approach, we were not able to solve the linear systems of equations in the cases $L \in \{5, 6\}$. The accuracy of the results with both approaches is practically the same. The results become more accurate the finer the coarse grid becomes. Even for a very coarse space L^H, the results with the VMS are considerably more accurate than the results with the artificial diffusion stabilization. If L^H is defined on finer and finer grids, the results become more and more accurate. Finally, if the coarse space is defined on the same grid as the fine space, $L = 6$, the results are nearly as accurate as for the SDFEM. Here we like to emphasize that a very simple model for ε_{add} has been used. The accuracy of the VMS can be certainly improved by applying more sophisticated models. An important observation is that the use of finer coarse grids $\mathcal{T}^H(\Omega)$ and the corresponding increase of degrees of freedom in L^H practically does not lead to an increase in the computing times for the semi-implicit approach (9).

The second numerical tests studies the convergence of the errors if the ratio of the fine mesh width h and the coarse mesh width H is kept (nearly) constant. We tested the scalings $h \sim H^\beta$, $\beta \in \{5/4, 3/2, 7/4, 2\}$. The scaling $h \sim H^2$ was proposed in [25] for the steady state equations. This scaling is the limit case for an optimal order of convergence of $\|\nabla e\|_{L^2(0,T;L^2)}$, see the discussion of finite element error estimates at the end of Section 2. Neglecting for simplicity the factor $\sqrt{2}$ in the mesh widths, then $h = 2^{-(l+1)}$ and $H = 2^{-(L+1)}$. Let l be given,

Table 2
Results for a fixed fine mesh and varying coarse mesh

method	L	d.o.f. L^H	$\|e\|_{L^\infty(0,T;L^2)}$	$\|e\|_{L^2(0,T;L^2)}$	$\|\nabla e\|_{L^2(0,T;L^2)}$	time
(6)	1	32	1.078e-2	1.520e-2	1.637e+0	184.4
	2	128	5.927e-3	8.365e-3	1.227e+0	181.9
	3	512	3.316e-3	4.679e-3	9.141e-1	183.0
	4	2048	2.099e-3	2.960e-3	7.349e-1	373.1
(9)	1	32	1.076e-2	1.518e-2	1.621e+0	103.5
	2	128	5.960e-3	8.425e-3	1.214e+0	103.2
	3	512	3.381e-3	4.800e-3	9.050e-1	103.1
	4	2048	2.191e-3	3.130e-3	7.281e-1	102.8
	5	8192	1.762e-3	2.531e-3	6.604e-1	103.0
	6	32768	7.052e-4	1.094e-3	3.526e-1	103.1
art. diff.			1.806e-1	2.538e-1	5.490e+0	56.3
SDFEM			5.213e-4	7.344e-4	3.414e-1	89.2

one obtains $L = \beta^{-1}(l+1)-1$. Since this number is in general not an integer, we used in the computations the nearest integer to $\beta^{-1}(l+1)-1$ as value for L. The computations were performed on triangular grids with piecewise linear finite elements. The coarsest grid (level 0) consisted of 8 triangles. The degrees of freedom for the spaces X^h and L^H are given in Table 3.

Table 3
Degrees of freedom on the triangular grids.

level	X^h	L^H
0		16
1	25	64
2	81	256
3	289	1024
4	1089	4096
5	4225	16384
6	16641	65536
7	66049	262144
8	263169	
9	1050625	

In Tables 4 – 7, errors in several norms are presented. The orders of convergence are given with respect to h. They were computed using the values on the two finest levels.

Table 4
Results for a fixed ratio $h \sim H^{5/4}$ of the fine mesh width and the coarse mesh width

L	l	$\|e\|_{L^\infty(0,T;L^2)}$	$\|e\|_{L^2(0,T;L^2)}$	$\|\nabla e\|_{L^2(0,T;L^2)}$
1	2	5.225e-1	7.335e-1	1.619e+1
2	3	1.554e-1	2.185e-1	9.432e+0
3	4	4.443e-2	6.254e-2	5.241e+0
4	5	1.218e-2	1.719e-2	2.771e+0
5	6	3.250e-3	4.611e-3	1.422e+0
5	7	7.358e-4	1.062e-3	5.221e-1
6	8	2.046e-4	3.038e-4	2.623e-1
7	9	6.262e-5	9.767e-5	1.316e-1
order		1.708	1.637	0.995

Table 5
Results for a fixed ratio $h \sim H^{3/2}$ of the fine mesh width and the coarse mesh width

L	l	$\|e\|_{L^\infty(0,T;L^2)}$	$\|e\|_{L^2(0,T;L^2)}$	$\|\nabla e\|_{L^2(0,T;L^2)}$
1	2	5.225e-1	7.335e-1	1.619e+1
2	3	1.554e-1	2.185e-1	9.432e+0
2	4	3.901e-2	5.506e-2	3.957e+0
3	5	1.046e-2	1.481e-2	2.038e+0
4	6	2.762e-3	3.934e-3	1.035e+0
4	7	9.844e-4	1.409e-3	5.613e-1
5	8	2.655e-4	3.878e-4	2.820e-1
6	9	7.683e-5	1.168e-4	1.415e-1
order		1.789	1.731	0.995

The numerical results show that one obtains first order of convergence if $\beta \in \{5/4, 3/2, 7/4\}$. For the limit value $\beta = 2$, a first order of convergence is not yet reached. A second order convergence for $\|e\|_{L^\infty(0,T;L^2)}$ and $\|e\|_{L^2(0,T;L^2)}$ cannot be observed. These observations correspond to the available analytical results, see the end of Section 2. For the SDFEM, we found for this example first order convergence in $\|\nabla e\|_{L^2(0,T;L^2)}$ and second order in $\|e\|_{L^2(0,T;L^2)}$.

Table 6
Results for a fixed ratio $h \sim H^{7/4}$ of the fine mesh width and the coarse mesh width

L	l	$\|e\|_{L^\infty(0,T;L^2)}$	$\|e\|_{L^2(0,T;L^2)}$	$\|\nabla e\|_{L^2(0,T;L^2)}$
1	3	1.407e-1	1.983e-1	7.507e+0
2	4	3.901e-2	5.506e-2	3.957e+0
2	5	1.426e-2	2.015e-2	2.188e+0
3	6	3.748e-3	5.318e-3	1.113e+0
4	7	9.844e-4	1.409e-3	5.613e-1
4	8	4.246e-4	6.096e-4	3.422e-1
5	9	1.157e-4	1.700e-4	1.718e-1
order		1.876	1.842	0.994

Table 7
Results for a fixed ratio $h \sim H^2$ of the fine mesh width and the coarse mesh width

L	l	$\|e\|_{L^\infty(0,T;L^2)}$	$\|e\|_{L^2(0,T;L^2)}$	$\|\nabla e\|_{L^2(0,T;L^2)}$
0	1	1.696e+0	2.380e+0	2.708e+1
1	3	1.407e-1	1.983e-1	7.507e+0
2	5	1.426e-2	2.015e-2	2.188e+0
3	7	1.614e-3	2.292e-3	6.794e-1
4	9	1.980e-4	2.846e-4	2.224e-1
order		1.514	1.505	0.806

4 Summary

The paper studied a two-level variational multiscale method for convection-diffusion equations. This method possesses two parameters: an additional diffusion ε_{add} and a vector-valued coarse finite element space L^H. The two main topics of the study were the conditions on L^H which are necessary for an efficient implementation of the method and the treatment of the additional terms of the VMS within the temporal discretization. It was shown that an efficient implementation of the VMS can be achieved if L^H consists of discontinuous finite element functions and that the basis of L^H is L^2-orthogonal. In this case, the additional matrices posses a very sparse structure and the inversion of the mass matrix of L^H, which is needed in the semi-implicit approach, can be done easily. Numerical tests at a model problem showed that a semi-implicit temporal discretization, which treats the subtraction of the additional diffusion from the large scales explicitly, is much more efficient than a fully implicit discretization. The computing times in the semi-implicit VMS practically did not depend on the dimension of the coarse space L^H. This is because the computational overhead, four matrix-vector products and multiplication with the diagonal matrix D^{-1}, is small in comparison to time needed for solving the linear system. This property will be shared by other temporal discretizations which treat the last term on the left hand side of (3) explicitly. A comparison of this method with the SDFEM showed a similar order of convergence in several norms for appropriate scalings of the fine and the coarse mesh.

An extension of this method to the Navier-Stokes equations requires similar algorithmic considerations as for scalar convection-diffusion equations. Based on the results obtained in this paper, we will use discontinuous finite elements for the space L^H and the semi-implicit approach. For the Navier-Stokes equations, one has to pay attention to an appropriate choice of the additional viscosity, using non-linear models like, e.g., the Smagorinsky model [31]. The application of this kind of VMS to turbulent flows will be future work.

References

[1] M. Anitescu, W. Layton, and F. Pahlevani. Implicit for local effects and explicit for nonlocal effects is unconditionally stable. Technical report, TR-MATH 03-17, ETNA, 2003.

[2] G. Q. Chen, Q. Du, and E. Tadmor. Spectral viscosity approximations for multi-dimensional scalar conservation laws. *Math. Comp.*, 61:629–643, 1993.

[3] R. Codina. Comparison of some finite element methods for solving the diffusion-convection-reaction equation. *Comput. Methods Appl. Mech. Engrg.*, 156:185 – 210, 1998.

[4] M. Fortin, R. Guenette, and R. Pierre. Numerical analysis of the EVSS method. *Comp. Meth. Appl. Mech. Eng.*, 143:79–95, 1997.

[5] L.P. Franca and A. Nesliturk. On a two-level finite element method for the incompressible Navier-Stokes equations. *Int. J. Numer. Methods Engrg.*, 52:433 – 453, 2001.

[6] V. Gravemeier. *The variational multiscale method for laminar and turbulent incompressible flow.* PhD thesis, Institute of Structural Mechanics, University of Stuttgart, 2003.

[7] V. Gravemeier, W.A. Wall, and E. Ramm. A three-level finite element method for the instationary incompressible Navier-Stokes equation. *Comput. Methods Appl. Mech. and Engrg.*, 193:1323 – 1366, 2004.

[8] J. L. Guermond. Stabilization of Galerkin approximations of transport equations by subgrid modelling. *M2AN*, pages 1293–1316, 1999.

[9] N. Heitmann. *Subgrid stabilization of evolutionary diffusive transport problems.* PhD thesis, University of Pittsburgh, 2004.

[10] T. J. Hughes, L. Mazzei, and K. E. Jensen. Large eddy simulation and the variational multiscale method. *Comput. Visual Sci.*, 3:47–59, 2000.

[11] T. J. Hughes, A. Oberai, and L. Mazzei. Large eddy simulation of turbulent channel flows by the variational multiscale method. *Phys. Fluids*, 13:1784–1799, 2001.

[12] T.J.R. Hughes. Multiscale phenomena: Green's functions, the Dirichlet-to-Neumann formulation, subgrid-scale models, bubbles and the origin of stabilized methods. *Comp. Meth. Appl. Mech. Engrg.*, 127:387 – 401, 1995.

[13] T.J.R. Hughes and A.N. Brooks. A multidimensional upwind scheme with no crosswind diffusion. In T.J.R. Hughes, editor, *Finite Element Methods for Convection Dominated Flows, AMD vol.34*, pages 19 – 35. ASME, New York, 1979.

[14] T.J.R. Hughes, L. Mazzei, A.A. Oberai, and A.A. Wray. The multiscale formulation of large eddy simulation: decay of homogeneous isotropic turbulence. *Phys. Fluids*, 13:505 – 512, 2001.

[15] V. John. *Large Eddy Simulation of Turbulent Incompressible Flows. Analytical and Numerical Results for a Class of LES Models*, volume 34 of *Lecture Notes in Computational Science and Engineering*. Springer-Verlag Berlin, Heidelberg, New York, 2004.

[16] V. John and S. Kaya. Finite element error analysis of a variational multiscale method for the Navier-Stokes equations. Preprint, 2004.

[17] V. John and S. Kaya. A finite element variational multiscale method for the Navier-Stokes equations. *SIAM J. Sci. Comp.*, 2004. in press.

[18] V. John and G. Matthies. MooNMD - a program package based on mapped finite element methods. *Comput. Visual. Sci.*, 6:163 – 170, 2004.

[19] H. Johnston and J.-L. Liu. Accurate, stable and efficient Navier-Stokes solvers based on explicit treatment of the pressure term, 2004. to appear in J. Comput. Phys.

[20] S. Kaya. Numerical Analysis of Subgrid-scale eddy viscosity for the Navier-Stokes equations. Technical report, University of Pittsburgh, 2002.

[21] S. Kaya and W. Layton. Subgrid-scale eddy viscosity methods are variational multiscale methods. Technical report, University of Pittsburgh, 2002.

[22] S. Kaya and B. Riviere. A discontinuous subgrid eddy viscosity method for the time dependent Navier-Stokes equations, 2003. submitted to SINUM.

[23] S. Kaya and B. Riviere. A two level method for solving the steady-state Navier-Stokes equations. Technical report, University of Pittsburgh, 2004.

[24] W. Layton. Variational multiscale methods and subgrid scale eddy viscosity. In H. Deconinck, editor, *32nd, 2002 - 2006, Computational Fluid Dynamics-Multiscale Methods*. Von Karman Institute for Fluid Dynamics, Rhode-Saint-Genèse, Belgium, 2002.

[25] W. J. Layton. A connection between subgrid scale eddy viscosity and mixed methods. *Appl. Math. and Computing*, 133:147–157, 2002.

[26] R. Lonsdale. An algebraic multigrid solver for the Navier-Stokes equations on unstructured meshes. *Int. J. Num. Meth. Heat Fluid Flow*, 3:3 – 14, 1993.

[27] Y. Maday and E. Tadmor. Analysis of the spectral vanishing method for periodic conservation laws. *SIAM Journal on Numerical Analysis*, 26:854–870, 1989.

[28] F. Pahlevani. *Sensitivity analysis of eddy viscosity models.* PhD thesis, University of Pittsburgh, 2004.

[29] H.-G. Roos, M. Stynes, and L. Tobiska. *Numerical Methods for Singularly Perturbed Differential Equations*. Springer, 1996.

[30] Y. Saad. A flexible inner-outer preconditioned GMRES algorithm. *SIAM J. Sci. Comput.*, 14(2):461–469, 1993.

[31] J.S. Smagorinsky. General circulation experiments with the primitive equations. *Mon. Weather Review*, 91:99 – 164, 1963.