INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 2007; **54**:779–804
Published online 14 March 2007 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/fld.1465

A 'divide-and-conquer' spatial and temporal multiscale method for transient convection–diffusion–reaction equations

Volker Gravemeier[1, 2, *, †] and Wolfgang A. Wall[2]

[1] *Emmy Noether Research Group 'Computational Multiscale Methods for Turbulent Combustion', Technical University of Munich, Boltzmannstr. 15, Garching D-85747, Germany*
[2] *Chair for Computational Mechanics, Technical University of Munich, Boltzmannstr. 15, Garching D-85747, Germany*

SUMMARY

A multiscale method for the numerical solution of transient convection–diffusion–reaction equations is proposed in the present paper. Two main goals have led to the development of the present method: a desired independence of any heuristic parameter such as the stabilization parameter in stabilized methods and a desire for a consistent multiscale approach in space *and* time. The method is constituted by solution approaches on a coarse- and a fine-scale level and by inter-scale operators for data transfer between those two levels. A particular feature of the method is that no large matrix system has to be solved. The results from three numerical test cases show that for both problematic flow regimes, that is, the regime of dominant convection and the regime of dominant convection and absorption, the present method provides completely stable solutions, which are not achieved by standard stabilized methods, particularly for the later regime. A still to be noted current shortcoming of the proposed method reveals itself in a too smooth resolution of regions with a sharp gradient in the solution field. Copyright © 2007 John Wiley & Sons, Ltd.

Received 16 January 2007; Revised 26 January 2007; Accepted 26 January 2007

KEY WORDS: computational fluid dynamics; transient convection–diffusion–reaction equation; finite element method; multiscale method; inter-scale operators

1. INTRODUCTION

Multiscale methods have already drawn much attention during the last decade and they will probably receive even more consideration in the upcoming future. The convection–diffusion equation is often used as a 'testbed' for methods later to be applied to the Navier–Stokes equations, since it reflects the 'competitive' behaviour of a convective and a diffusive (or viscous) term. Convection–

[*] Correspondence to: Volker Gravemeier, Chair for Computational Mechanics, Technical University of Munich, Boltzmannstr. 15, Garching D-85747, Germany.
[†] E-mail: vgravem@lnm.mw.tum.de

Copyright © 2007 John Wiley & Sons, Ltd.

diffusion–reaction equations may appear in two different contexts: either the additional reactive term has an actual physical meaning, or the equation can be considered a model problem for the semi-discrete (in time) convection–diffusion or Navier–Stokes equations, with the reactive term mimicking the discretized transient term. The Navier–Stokes equations also represent an intended future application area of the method proposed in this paper.

A dominating convective term usually makes a standard numerical approach (e.g. the standard Galerkin finite element method (SGFEM)) unfeasible. In the convection–diffusion–reaction equation, a reactive term is included, often posing another challenge to the numerical solution when the reactive term or the convective and the reactive term, respectively, dominate the problem. A particularly challenging problem was detected in [1] for large convection and absorption (i.e. a positive reaction coefficient in terms of the later definition of this coefficient in the present paper) in the presence of a negative streamwise gradient of the solution. Stabilized methods, although very successfully developed for and applied to convection–diffusion problems at high Peclet number, are typically not able to solve this problem without additional discontinuity capturing. Discontinuity-capturing methods developed in [2, 3] for reaction-dominated problems were successfully applied in [4] to turbulence models including 'source' terms. The reader may consult [5] for a recent overview of discontinuity capturing methods for convection–diffusion equations. An exception is the method recently proposed in [6], featuring a considerably more complicated stabilization parameter definition, though.

Some approaches for dealing with convection-dominated convection–diffusion(–reaction) equations may be found in the literature. In the context of FEMs, one of the most popular (class of) approach(es) are stabilized methods, such as SUPG [7, 8] (also known as the streamline diffusion method [9]), GLS [10], USFEM [11] (see also [12] for an improved version for convection–diffusion–reaction equations), and stabilized methods emanating from the variational multiscale method (VMM) [13], such as the ones described in [1, 14], respectively. For overviews on stabilized methods for the convection–diffusion–reaction equation, the reader is referred to [15, 16]. A related approach is the finite increment calculus (FIC), which was also recently applied to the convection–diffusion–reaction equation in [6]. Despite their successful application to these problems, stabilized methods essentially rely on the well-known stabilization parameter, which is usually denoted by τ. A rigorous derivation of this stabilization parameter, except in a heuristic manner, is still lacking, although the VMM, as well as tightly related residual-free bubble methods (see [17]), have shedded considerable light on its origin. A particular way to compute stabilization parameters from element matrices and vectors was proposed in [18]. Other non-standard spatial multiscale methods can be found in [19, 20].

An issue which has so far received relatively little attention compared to *spatial* multiscale approaches are *temporal* multiscale approaches. The problems related to a dominating reactive term due to small time-step lengths in semi-discretization methods have recently been investigated, for instance, in [21, 22]. In a survey of the building blocks of existing multiscale methods in various areas of application (see [23]) which was conducted by the authors, among others, it turned out that most of the spatial multiscale methods do not pay particular attention to the temporal aspect of the problem at all. Some of them employ cycling strategies (i.e. the use of a number of smaller time steps for the smaller scales during one larger time step for the larger scales). A particular multiscale space–time technique was developed in [24]. However, it is well known that the spatial and the temporal scale are tightly coupled with respect to the multiscale feature of the problem in most of the cases: if the spatial scale extends over various scale levels, the temporal scale will spread over several levels, too.

Copyright © 2007 John Wiley & Sons, Ltd.

DOI: 10.1002/fld

The guidelines for the development of the present method are a complete independence of any heuristic parameter within the method, on the one hand, and the addressing of both the spatial *and* the temporal multiscale aspects of the problem by appropriate schemes, on the other hand. In the course of the study in [23], a particular method appeared to be well suited for that goal, although it had been applied only to problems very different from convection(–reaction)-dominated flow problems: the gap-tooth scheme combined with projective time integration as proposed in [25]. The projective time-integration approach was separately published in [26] and recently extended to further time-integration schemes in [27]. Similarities may also be found in the heterogeneous multiscale methods proposed in [28].

The method to be presented in this paper acts on two resolution levels: a coarse-scale level where an acceptable numerical solution is usually not achievable with a SGFEM and a fine-scale level where an acceptable solution is indeed achievable. Since such a fine-scale solution over the *complete* problem domain is usually linked with prohibitive computational cost, the problem will be partitioned into a number of small (fine-scale) problems, and the assembly of the solutions to the partitioned problem constitutes the overall (coarse-scale) solution. Thus, the method is referred to as a 'divide-and-conquer' (DAC) multiscale method: the problem is first divided and then conquered in the form of a solution assembly. The key ingredients of the proposed method, besides the solution strategies on the coarse- and fine-scale level, are inter-scale operators providing the data transfer between these scale levels. For problems of fluid mechanics, it was already advocated in [29] to rather (spatially) resolve the respective problem wherever possible than to use a stabilized method. Using meshes adapted to the boundary layer to be resolved, such an approach was proposed in [30] for the convection–diffusion equation. It is emphasized that it is not required to have any knowledge concerning the location of critical areas of the underlying problem for the present method.

The remainder of this paper is organized as follows. In Section 2, the basic spatial and temporal discretization methods used within the DAC multiscale method are briefly presented: the SGFEM and the one-step-θ scheme. The DAC method is then constituted in Section 3 by describing firstly the overall strategy and secondly the components (i.e. the coarse- and fine-scale solvers and the inter-scale operators). This section is closed by a brief analysis of the computational cost and the storage requirements. Results from three numerical test cases are presented in Section 4. Finally, conclusions are drawn in Section 5.

2. BASIC SPATIAL AND TEMPORAL DISCRETIZATION METHODS

2.1. Problem statement

The strong form of the transient convection–diffusion–reaction equation may be stated as follows: find the scalar unknown ϕ such that

$$\frac{\partial \phi}{\partial t} + \mathbf{a} \cdot \nabla \phi - \kappa \Delta \phi + \sigma \phi = f \quad \text{in } \Omega \times [0, T] \tag{1}$$

where $\mathbf{a}$ denotes the given convective velocity field, κ the given positive diffusion coefficient, σ the given reaction coefficient ($\sigma{>}0$: absorption, $\sigma{<}0$: production), and f a potential source term. The problem is defined in the domain Ω with boundary Γ and during the time interval $[0, T]$. Dirichlet boundary conditions are assumed on the domain boundary:

$$\phi = g \quad \text{on } \Gamma \times [0, T] \tag{2}$$

Copyright © 2007 John Wiley & Sons, Ltd.

DOI: 10.1002/fld

The initial condition is given as

$$\phi = \phi^0 \quad \text{in } \Omega \times \{0\} \tag{3}$$

The solution to (1) is characterized by non-dimensional numbers. Two important numbers are the Peclet number $Pe = AL/\kappa$, relating convection and diffusion, and the Damköhler number $Da = \sigma L/A$, relating reaction and convection, where A and L denote the characteristic velocity and length of the problem, respectively.

2.2. Spatial discretization: standard Galerkin finite element method

The variational formulation corresponding to (1) is: find $\phi \in \mathcal{S}$ such that

$$\left(w, \frac{\partial \phi}{\partial t}\right)_\Omega + a(w, \phi) = (w, f)_\Omega \quad \forall w \in \mathcal{V}$$

where

$$a(w, \phi) = (w, \mathbf{a} \cdot \nabla \phi)_\Omega + (\nabla w, \kappa \nabla \phi)_\Omega + (w, \sigma \phi)_\Omega$$

employing the usual notation $(\cdot, \cdot)_\Omega$ for the L_2-inner product over the domain Ω. $\mathcal{S}$ and $\mathcal{V}$ denote the standard variational functional spaces for the solution and weighting function, respectively.

For an FEM, the domain Ω is partitioned into n_{el} non-overlapping elements Ω_e $(e = 1, \ldots, n_{\text{el}})$, with the characteristic element length of the discretization being h. Choosing the standard finite element (FE) function spaces $\mathcal{S}^h \subset \mathcal{S}$ and $\mathcal{V}^h \subset \mathcal{V}$, the SGFEM is given as: find $\phi^h \in \mathcal{S}^h$ such that

$$\left(w^h, \frac{\partial \phi^h}{\partial t}\right)_\Omega + a(w^h, \phi^h) = (w^h, f)_\Omega \quad \forall w^h \in \mathcal{V}^h$$

In the end, a matrix system as

$$\mathbf{M}\dot{\boldsymbol{\phi}} + (\mathbf{C} + \mathbf{K} + \sigma \mathbf{M})\boldsymbol{\phi} = \mathbf{f} \tag{4}$$

is obtained, where $\mathbf{M}$, $\mathbf{C}$, and $\mathbf{K}$ denote the mass, convective, and diffusive matrices, respectively. The vector of solution values and its temporal derivative are given by $\boldsymbol{\phi}$ and $\dot{\boldsymbol{\phi}}$, and the right-hand-side vector by $\mathbf{f}$.

The (spatial) FE solution is characterized by the element-based counterparts of the aforementioned non-dimensional numbers: the element Peclet number

$$Pe_e(h) = \frac{|\mathbf{a}|h}{2\kappa}$$

and the element Damköhler number

$$Da_e(h) = \frac{\sigma h}{|\mathbf{a}|}$$

Severe problems are encountered for high Pe_e and/or Da_e, that is, an adequate (oscillation-free) solution cannot be achieved with the SGFEM. As a remedy for particularly the high Pe_e regime, stabilized FEMs were proposed.

Copyright © 2007 John Wiley & Sons, Ltd.

DOI: 10.1002/fld

2.3. Stabilized finite element methods

Introducing the differential operator pertinent to the convection–diffusion–reaction equation as

$$\mathscr{L}\phi = \mathscr{L}_t\phi + \mathscr{L}_{\text{con}}\phi + \mathscr{L}_{\text{dif}}\phi + \mathscr{L}_{\text{rea}}\phi = \frac{\partial\phi}{\partial t} + \mathbf{a}\cdot\nabla\phi - \kappa\Delta\phi + \sigma\phi$$

and another differential operator $\mathscr{L}_{\text{sta}}$, which is to be defined below, a general stabilized FEM can be written as: find $\phi^h \in \mathscr{S}^h$ such that

$$\left(w^h, \frac{\partial\phi^h}{\partial t}\right)_\Omega + a(w^h, \phi^h) + (\mathscr{L}_{\text{sta}}w^h, \tau\mathscr{L}\phi^h)_{\Omega'} = (w^h, f)_\Omega \quad \forall w^h \in \mathscr{V}^h \tag{5}$$

where Ω' denotes the union of all element interiors $\bigcup_{e=1}^{n_{\text{el}}} \Omega_e$ and τ the stabilization parameter. Several definitions for τ may be found in the literature. For the numerical tests in the present paper, the definition in [12] is used. The differential operator $\mathscr{L}_{\text{sta}}$ in (5) is chosen as follows: $\mathscr{L}_{\text{sta}} = \mathscr{L}_{\text{con}}$ for the SUPG method, $\mathscr{L}_{\text{sta}} = \mathscr{L}$ for the GLS method, and $\mathscr{L}_{\text{sta}} = -\mathscr{L}^*$ for the USFEM and stabilized methods emanating from the VMM, respectively, where the adjoint differential operator is indicated by $\mathscr{L}^*$. In the end, additional matrix entries due to the stabilization terms have to be added to the matrix system (4).

2.4. Temporal discretization: one-step-θ scheme

Applying the one-step-θ scheme to the matrix formulation (4) of the problem yields the following equation for the vector of solution values $\boldsymbol{\phi}^{n+1}$ at the new time step $n+1$:

$$\mathbf{M}\frac{\boldsymbol{\phi}^{n+1} - \boldsymbol{\phi}^n}{\Delta t} + \theta[(\mathbf{C} + \mathbf{K} + \sigma\mathbf{M})\boldsymbol{\phi}^{n+1}] + (1-\theta)[(\mathbf{C} + \mathbf{K} + \sigma\mathbf{M})\boldsymbol{\phi}^n] = \theta\mathbf{f}^{n+1} + (1-\theta)\mathbf{f}^n$$

which may be rewritten as

$$[\mathbf{C} + \mathbf{K} + \sigma_{\text{mod}}\mathbf{M}]\boldsymbol{\phi}^{n+1} = \mathbf{f}^{n+1} + \mathbf{f}_{\text{T}}^n \tag{6}$$

where the ‘time right-hand side’ matrix is defined as

$$\mathbf{f}_{\text{T}}^n = \frac{1}{\theta\Delta t}\mathbf{M}\boldsymbol{\phi}^n - \frac{(1-\theta)}{\theta}\{[\mathbf{C} + \mathbf{K} + \sigma\mathbf{M}]\boldsymbol{\phi}^n - \mathbf{f}^n\}$$

Note that Equation (6) represents the matrix formulation of a *steady* convection–diffusion–reaction equation with modified reaction coefficient

$$\sigma_{\text{mod}} = \frac{1}{\theta\Delta t} + \sigma \tag{7}$$

compared to (1) and an additional right-hand-side term. In the present paper, it will be focused on the Crank–Nicolson scheme ($\theta = \frac{1}{2}$), the only scheme of second-order accuracy in the context of the one-step-θ family.

Copyright © 2007 John Wiley & Sons, Ltd.

DOI: 10.1002/fld

An important non-dimensional number relating the spatial discretization and the temporal discretization is the Courant number, which is defined as

$$Cr(h,\Delta t)=\frac{|\mathbf{a}|\Delta t}{h}$$

3. MULTISCALE METHOD

3.1. Strategy

Two discretization levels are considered, both spatially and temporally: the 'usual' level (coarse-scale level) and a refined level (fine-scale level).

As the usual (or coarse-scale) discretization level for a convection-dominated flow, we consider a spatial discretization with characteristic element length scale H, where the element Peclet number $Pe_e(H)$ is considerably larger than 1 (i.e. a mesh on which an adequate numerical solution is not achievable with a SGFEM). As the refined (or fine-scale) level, we consider an adequate spatial resolution, that is, a discretization with element length h, where $Pe_e(h)=1$ at maximum for all elements. Of course, such a fine-scale discretization over the *complete* problem domain Ω is usually not viable due to the huge number of degrees of freedom to be solved for in the end. However, the fine-scale discretization is used only in local subdomains ω_I of the domain Ω in the present approach (i.e. in the vicinity of each node I $(I=1,\ldots,N_{\text{nod}})$ belonging to the coarse-scale discretization, where N_{nod} is the number of coarse-scale nodes excluding nodes on the Dirichlet boundary). Each of the local subdomains ω_I is independent of all other local subdomains ω_J for $J\neq I$. Thus, a number of N_{nod} fine-scale problems has to be solved in the end, each of the resulting matrix systems being of small size. *No large matrix system on the coarse-scale level has to be solved*, as will be shown further.

Remark

The nodes on the Dirichlet boundary are accounted for strongly by directly assigning the Dirichlet values to the respective nodes and, hence, obviating any need for a creation of subdomains at those nodes. However, it is also possible to account for the Dirichlet boundary conditions weakly, as recently proposed in [31], in the context of the present method.

The spatial and temporal resolution cannot be chosen independently of each other. The time step initially chosen with respect to the coarse-scale spatial discretization ΔT must be expected to be much too large with respect to the fine-scale discretization: $Cr(h,\Delta T)\gg 1$. Therefore, as the temporal fine-scale level, we consider an adequate time-step length Δt where, at least for the numerical tests in the present paper, $Cr(h,\Delta t)=1$ at maximum. (Of course, larger values for $Cr(h,\Delta t)$ may be allowed depending on the chosen temporal discretization scheme.) A number of n_{ts} time steps with time-step length Δt are performed within each coarse-scale time step with time-step length ΔT. The coarse- and fine-scale discretizations in space and time are displayed in principle in Figure 1 for a space–time domain (1-D in space) with five inner coarse-scale nodes and respective local subdomains.

We would like to go more into detail now. For this purpose, the (discrete) coarse-scale solution, defined on the complete problem domain Ω and obtained on the coarse mesh with characteristic length H, is denoted by Φ^H, and the fine-scale solution, defined on the fine-scale domain(s) ω_I and obtained on the respective fine mesh(es) with characteristic length h, is denoted by ϕ^h. The

Copyright © 2007 John Wiley & Sons, Ltd.

DOI: 10.1002/fld

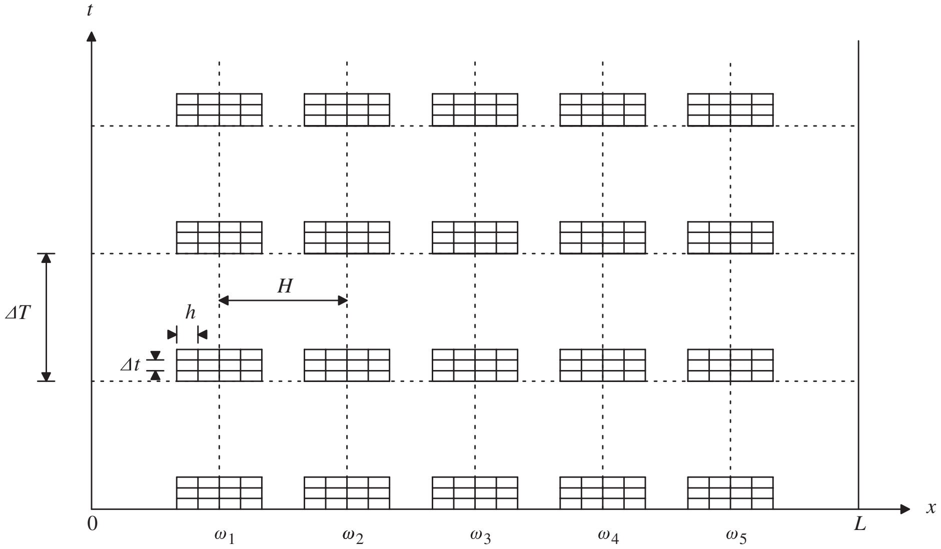

Figure 1. Coarse- and fine-scale discretizations in space–time domain (1-D in space) with the Dirichlet boundary conditions at both ends of the spatial domain.

superscripts referring to the characteristic length will be omitted below, since there is no danger of confusing Φ and ϕ. Adopting the notation in [28] for the spatial inter-scale operators, the coarse- and fine-scale variables are related to each other by a compression operator $\mathscr{Q}$ (i.e. the fine-to-coarse inter-scale operator) and a reconstruction operator $\mathscr{R}$ (i.e. the coarse-to-fine inter-scale operator) such that

$$\Phi = \mathscr{Q}\phi$$

and

$$\phi = \mathscr{R}\Phi$$

where $\mathscr{Q}$ and $\mathscr{R}$ should satisfy $\mathscr{Q}\mathscr{R} = \mathscr{I}$, with $\mathscr{I}$ denoting the identity operator.

In the present study, we would like to define two corresponding operators in time: a projection operator $\mathscr{P}$ (adopting the notation in [26]) and the so-called 'activation' operator $\mathscr{A}$. The projection operator $\mathscr{P}$ is a temporal fine-to-coarse inter-scale operator defined for a fine-scale time step n within a coarse-scale time step N as

$$\Phi^N = \mathscr{P}\Phi^{N/n}$$

and, in some sense, resembles the compression operator $\mathscr{Q}$ in space. The activation operator $\mathscr{A}$ is a temporal coarse-to-fine inter-scale operator defined as

$$\Phi^{N/n} = \mathscr{A}\Phi^N$$

and, in some sense, resembles the reconstruction operator $\mathscr{R}$ in space. Both temporal operators act on spatially coarse scales.

Copyright © 2007 John Wiley & Sons, Ltd.

DOI: 10.1002/fld

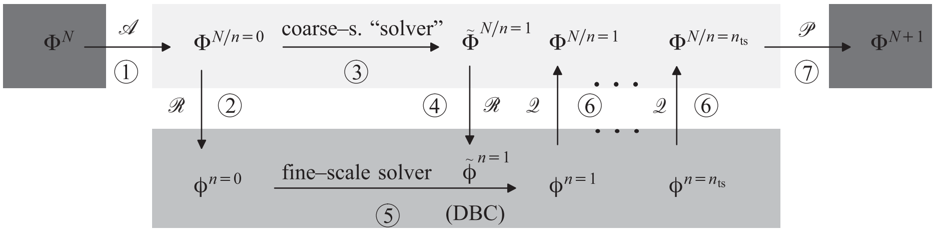

Figure 2. Strategy for one coarse-scale time step (dark grey: coarse-scale space and time; medium grey: fine-scale space and time; and light grey: coarse-scale space and fine-scale time).

Using the operators defined above, the strategy of the proposed multiscale method is graphically summarized in Figure 2, with the numbers of the subsequently described steps plotted at the respective locations. In the course of a coarse-scale time step $N \rightarrow N+1$, seven different operations are carried out, some of them repeatedly:

1. the initial coarse-scale field for the first fine-scale time step within this coarse-scale time step has to be determined based on the coarse-scale solution at the end of the preceding coarse-scale time step: $\Phi^{N/n=0} = \mathscr{A}\Phi^{N}$;
2. the initial fine-scale field is reconstructed: $\phi^{n=0} = \mathscr{R}\Phi^{N/n=0}$;
3. the coarse-scale field is advanced to get a predicted value for the coarse-scale field at the end of the current fine-scale time step $\tilde{\Phi}^{N/n=1}$, which is needed for the fine-scale solver in the form of Dirichlet boundary conditions for the fine-scale domains (see also Section 3.2.3);
4. the predicted value for the fine-scale field at the end of the current fine-scale time step is reconstructed at the Dirichlet boundary nodes: $\tilde{\phi}^{n=1} = \mathscr{R}\tilde{\Phi}^{N/n=1}$;
5. the fine-scale field is solved for to get $\phi^{n=1}$;
6. the fine-scale field is then compressed such that the coarse-scale field at the end of the current fine-scale time step is available: $\Phi^{N/n=1} = \mathscr{Q}\phi^{n=1}$ (*repeat operations* (2)–(6) *until* $\Phi^{N/n=n_{\mathrm{ts}}}$ *is achieved*); and
7. the coarse-scale field at the end of the last fine-scale time step is then projected over the remaining time within the coarse-scale time step, $\Delta T^{\mathrm{rem}} = \Delta T - n_{\mathrm{ts}}\Delta t$, to get the coarse-scale field at the end of the current coarse-scale time step: $\Phi^{N+1} = \mathscr{P}\Phi^{N/n=n_{\mathrm{ts}}}$.

3.2. *Components*

The following components constitute the multiscale method:

- the fine-scale solver;
- the coarse-scale 'solver';
- the spatial compression operator $\mathscr{Q}$;
- the spatial reconstruction operator $\mathscr{R}$;
- the temporal projection operator $\mathscr{P}$; and
- the temporal activation operator $\mathscr{A}$.

The spatial compression operator $\mathscr{Q}$ and the temporal activation operator $\mathscr{A}$ can be explained in brief. All other components will be individually and more elaborately addressed further.

Both the *spatial compression operator* and the *temporal activation operator* are constituted by simple *injection operators*. Spatially, the result from the fine-scale calculation at a particular coarse-scale node will be injected in the form of the value at the fine-scale node which is situated at the geometrically identical location (i.e. $\Phi_I^{N/n} = \phi_i^n$, where the coarse-scale node I and the fine-scale node i coincide geometrically: $\mathbf{x}_I = \mathbf{x}_i$). The fine-scale meshes chosen in the numerical tests below will all feature such a geometrically coinciding node. If that is not the case, the value at the coarse-scale node may be obtained by an interpolation operator involving the surrounding fine-scale nodes, instead of an injection operator. Temporally, the initial value for the first fine-scale time step within a particular coarse-scale time step is simply obtained by injecting the result at the end of the preceding coarse-scale time step for each coarse-scale node I: $\Phi_I^{N/0} = \Phi_I^N$.

3.2.1. Fine-scale solver. At first, the fine-scale domain size assigned to a coarse-scale node, the fine-scale element length h, and the fine-scale time-step length Δt have to be established. The fine-scale element Peclet number $Pe_e(h)$ is fixed to be of unit value. Thus, the fine-scale element length is determined by the convective velocity field and the diffusion coefficient: $h = 2\kappa/|\mathbf{a}|$. The ratio of the coarse- and the fine-scale element lengths H and h equals the coarse-scale element Peclet number $Pe_e(H)$. Only structured meshes with the same length h in all coordinate directions are used on the fine-scale level. Therefore, after choosing the number of elements in all coordinate directions, usually the same number in all directions, the size of each fine-scale domain ω_I is determined by the lengths l_d in all d coordinate directions: $l_d = n_{el}^d h$. The domain ω_I is arranged such that the corresponding coarse-scale node I is located in the centre of the domain.

As aforementioned, the fine-scale Courant number $Cr(h, \Delta t)$ is also fixed to be of unit value. As a result, the fine-scale time-step length depends on the fine-scale element length according to $\Delta t = h/|\mathbf{a}|$. The ratio of the coarse- and the fine-scale time-step lengths ΔT and Δt equals the product of the coarse-scale Courant and Peclet number:

$$\frac{\Delta T}{\Delta t} = Cr(H, \Delta T)\frac{H}{h} = Cr(H, \Delta T)Pe_e(H)$$

With the fine-scale problem domains defined, the transient convection–diffusion–reaction equation (1) is solved independently on each of these domains. Since the domain size and the equation to be solved are identical for each individual fine-scale domain, the differences merely come into play due to the respective Dirichlet boundary (2) and initial (3) conditions. These conditions are provided by the coarse-scale solution *via* the spatial reconstruction operator (see Section 3.2.3). With the necessary conditions at hand, problem (1) is numerically solved as outlined in Sections 2.2 and 2.4 (i.e. using the SGFEM and the one-step-θ scheme with $\theta = \frac{1}{2}$). The respective matrix equation to be solved, (6), is repeated here for the convenience of the reader:

$$[\mathbf{C} + \mathbf{K} + \sigma_{\text{mod}}\mathbf{M}]\boldsymbol{\phi}^{n+1} = \mathbf{f}^{n+1} + \frac{2}{\Delta t}\mathbf{M}\boldsymbol{\phi}^n - [\mathbf{C} + \mathbf{K} + \sigma\mathbf{M}]\boldsymbol{\phi}^n + \mathbf{f}^n$$

3.2.2. Coarse-scale 'solver'. As the quotation marks should indicate, the coarse-scale 'solver' is not a solver in the sense of a usual FE solver. In contrast to, e.g. the HMM in [28], where the fine-scale model merely has to provide missing data for the coarse-scale solution process, the fine-scale solution is the only solution, and there is no actual solution procedure on the coarse-scale level in the present approach. As a result, the coarse-scale solution is basically constituted by an ensemble of local fine-scale solutions. However, for the provision of the Dirichlet boundary conditions *via* the

Copyright © 2007 John Wiley & Sons, Ltd.

DOI: 10.1002/fld

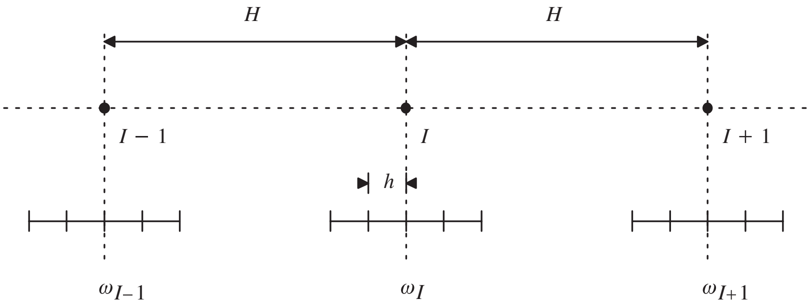

Figure 3. Coarse-scale nodes and respective local subdomains in 1-D.

spatial reconstruction operator (see Section 3.2.3), an approximation of the coarse-scale solution at the end of the current fine-scale time step has to be available. For this purpose, an explicit Forward-Euler/central-difference solution procedure based on the transient convection–diffusion–reaction equation (1) is performed. Because of the relatively small time step Δt, a sufficiently good approximation should be accomplishable even with such an explicit procedure.

The procedure will be explained in a 1-D setting. For the 2-D and 3-D case, the convective and diffusive parts have to be extended to the further coordinate directions in a similar fashion. The convection–diffusion–reaction equation in 1-D for the coarse-scale solution Φ reads

$$\frac{\partial \Phi}{\partial t} + a\frac{\partial \Phi}{\partial x} - \kappa \frac{\partial^2 \Phi}{\partial x^2} + \sigma \Phi = f \tag{8}$$

Applying the Forward-Euler/central-difference approach based on the coarse-scale nodes to (8) yields

$$\frac{\Phi_I^{N/n+1} - \Phi_I^{N/n}}{\Delta t} + a\frac{\Phi_{I+1}^{N/n} - \Phi_{I-1}^{N/n}}{2H} - \kappa\frac{\Phi_{I+1}^{N/n} - 2\Phi_I^{N/n} + \Phi_{I-1}^{N/n}}{H^2} + \sigma\Phi_I^{N/n} = f_I^{N/n}$$

and, after some calculus,

$$\begin{aligned}\Phi_I^{N/n+1} &= \Delta t f_I^{N/n} + \left(1 - \sigma\Delta t - \frac{2\kappa\Delta t}{H^2}\right)\Phi_I^{N/n} \\ &\quad + \left(\frac{a\Delta t}{2H} + \frac{\kappa\Delta t}{H^2}\right)\Phi_{I-1}^{N/n} + \left(-\frac{a\Delta t}{2H} + \frac{\kappa\Delta t}{H^2}\right)\Phi_{I+1}^{N/n}\end{aligned}$$

For evaluating a general coarse-scale node I, only the nearest-neighbour nodes $I+1$ and $I-1$ are taken into account, see Figure 3. If node I is the node nearest to the Dirichlet boundary, the value at node $I-1$ or $I+1$, respectively, is given by the (coarse-scale) Dirchlet boundary value at that node.

3.2.3. Spatial reconstruction operator. The spatial reconstruction operator $\mathscr{R}$ is constituted by an *interpolation operator*. Two different interpolation operators are considered for the numerical examples in the present paper: a linear and a quadratic interpolation. With reference to the simple

Copyright © 2007 John Wiley & Sons, Ltd.

1-D setting displayed in Figure 3, the linear interpolation may be explained as follows. For the left part of the fine-scale domain, a linear interpolation within the 'element' between node $I-1$ and node I is used to determine the values at the nodes in that part of the domain. The 'element' between node I and node $I+1$ is used for the nodes in the right part of the fine-scale domain. For the quadratic interpolation, all three nodes are taken into account, and the respective 'element' is twice as large.

Remark
The expression 'element' is deliberately put within quotation marks, since there is actually no need for setting up an element structure on the coarse-scale level due to the lack of any FE-based solution procedure on this level. However, if such an element structure is available, as it is when the present method is implemented into an existing FE code, the respective interpolation routines represent convenient tools to be exploited in this context, in particular for 2-D and 3-D applications.

Two types of inter-scale information transfer from the coarse- to the fine-scale level have to be provided. On the one hand, the initial conditions for all nodes of the fine-scale domains have to be available at the beginning of the fine-scale solution procedure. For this purpose, it is sufficient to apply the interpolation operator to the coarse-scale solution at the end of the lastly completed fine-scale time step $n-1 \rightarrow n$ (see step (2) of strategy description in Section 3.1). On the other hand, the Dirichlet conditions at the boundaries of the fine-scale domains for the current fine-scale time step $n \rightarrow n+1$ are needed (see step (4) of strategy description in Section 3.1). This is more involved, since those boundary conditions change in time as the coarse-scale solution does. Consistent with the chosen temporal discretization scheme, the Crank–Nicolson scheme, the values for the Dirichlet boundary conditions are assumed to be the mean of the solution at the lastly completed time step $n-1 \rightarrow n$ and the approximated solution at the end of the current time step $n \rightarrow n+1$, as described above: $(\phi^n + \tilde{\phi}^{n+1})/2 = (\phi^n + \mathscr{R}\tilde{\Phi}^{N/n+1})/2$.

It may easily be verified that the chosen operators for $\mathscr{Q}$ and $\mathscr{R}$ satisfy $\mathscr{Q}\mathscr{R} = \mathscr{I}$, that is,

$$\Phi = \mathscr{Q}\phi = \mathscr{Q}\mathscr{R}\Phi = \mathscr{I}\Phi$$

Applying the interpolation operator to the coarse-scale solution at a coarse-scale node I yields the same fine-scale solution value at the geometrically corresponding fine-scale node i: $\phi_i = \mathscr{R}\Phi_I$. After applying the injection operator, the same coarse-scale solution value at the coarse-scale node is obtained: $\Phi_I = \mathscr{Q}\phi_i$.

3.2.4. Temporal projection operator.

The temporal projection operator $\mathscr{P}$ is constituted by an *extrapolation operator*, as proposed in [26]. However, besides the Forward-Euler extrapolation, which is favoured in [26], a Forward-Euler/Crank–Nicolson predictor–corrector extrapolation, which is only described in an extended technical report by those authors [32], is also used, and the results obtained for both operators are compared.

We will start with the Forward-Euler extrapolation, since this also represents the predictor part of the predictor–corrector extrapolation. As in Figure 2, it is assumed that we have are about to start the coarse-scale time step $N \rightarrow N+1$. At the end of the last fine-scale time step $n_{\text{ts}}-1 \rightarrow n_{\text{ts}}$, the remaining time within the coarse-scale time step ΔT is $\Delta T^{\text{rem}} = \Delta T - n_{\text{ts}}\Delta t$. For simplicity, let us assume that the ratio of the coarse- and fine-scale time steps $N_{\text{ts}}/n_{\text{ts}}$ is an integer value. Then, the number of remaining fine-scale time steps to reach the end of the coarse-scale time step, which are not taken in practice, amount to $n_{\text{ts}}^{\text{rem}} = \Delta T^{\text{rem}}/\Delta t$. Those 'virtual' time steps have

Copyright © 2007 John Wiley & Sons, Ltd.

DOI: 10.1002/fld

to be bridged. The essential assumption underlying the extrapolation is that the discrete temporal derivative over the remaining fine-scale time steps is equal to the discrete temporal derivative in the last actually taken fine-scale time step:

$$\frac{\Phi_I^{N+1} - \Phi_I^{N/n_{\mathrm{ts}}}}{n_{\mathrm{ts}}^{\mathrm{rem}}\Delta t} = \frac{\Phi_I^{N/n_{\mathrm{ts}}} - \Phi_I^{N/n_{\mathrm{ts}}-1}}{\Delta t} \tag{9}$$

Solving Equation (9) for the coarse-scale solution Φ_I^{N+1} at the end of the current coarse-scale time step yields

$$\Phi_I^{N+1} = \Phi_I^{N/n_{\mathrm{ts}}} + n_{\mathrm{ts}}^{\mathrm{rem}}(\Phi_I^{N/n_{\mathrm{ts}}} - \Phi_I^{N/n_{\mathrm{ts}}-1}) \tag{10}$$

Note that Equation (10) does not directly depend on the fine-scale time-step length Δt. It is the only equation to be solved for the Forward-Euler extrapolation.

Simultaneously, Equation (10) serves as the predictor for the predictor–corrector extrapolation. In this context, the result of (10) is denoted $\tilde{\Phi}_I^{N+1}$ to indicate it being a predicted value. With $\tilde{\Phi}_I^{N+1}$ as the initial coarse-scale field, the multiscale method as described in Section 3.1 is rerun for the subsequent time step $N+1 \rightarrow N+2$ until $\Phi^{N+1/n_{\mathrm{ts}}}$ is achieved. Finally, the coarse-scale solution Φ_I^{N+1} at the end of the coarse-scale time step $N \rightarrow N+1$ is obtained as a weighted average in the form

$$\begin{aligned}\Phi_I^{N+1} = \Phi_I^{N/n_{\mathrm{ts}}} &+ \alpha n_{\mathrm{ts}}^{\mathrm{rem}}(\Phi_I^{N/n_{\mathrm{ts}}} - \Phi_I^{N/n_{\mathrm{ts}}-1}) \\ &+ (1-\alpha) n_{\mathrm{ts}}^{\mathrm{rem}}(\Phi_I^{N+1/n_{\mathrm{ts}}} - \Phi_I^{N+1/n_{\mathrm{ts}}-1})\end{aligned}$$

where α denotes the weighting factor. For a Crank–Nicolson-like scheme, α would probably be expected to be $\frac{1}{2}$ at first sight. However, the weighting factor is defined as

$$\alpha = \frac{n_{\mathrm{ts}}^{\mathrm{rem}} + 2n_{\mathrm{ts}} - 1}{2(n_{\mathrm{ts}}^{\mathrm{rem}} + n_{\mathrm{ts}})}$$

in [32], representing the value for which second-order accuracy is achieved, see [32] for elaboration. If $n_{\mathrm{ts}}^{\mathrm{rem}} \gg n_{\mathrm{ts}}$ (i.e. the remaining time within the coarse-scale time step ΔT^{rem} is close to ΔT), α is approximately $\frac{1}{2}$. This represents a target situation also from the point of view of computational efficiency, since we would like to perform as little fine-scale time steps within a coarse-scale time step as possible.

3.3. Analysis of computational cost and storage requirements

In this section, we would like to briefly analyse the important aspects computational cost and storage requirements of the proposed method in comparison to a stabilized FEM. In the DAC method, a SGFEM is applied to each local subdomain ω_I. If we assume, for instance, each ω_I to be discretized by four elements in each spatial direction, as in the numerical tests below, N_{nod} matrix systems of size $3^d \times 3^d$, where d denotes the number of spatial dimensions, have to be assembled and solved in each fine-scale time step. On the other hand, a matrix system of size $N_{\mathrm{nod}} \times N_{\mathrm{nod}}$ has to be assembled and solved in each time step in the course of a stabilized FEM. As a result, the storage requirements are substantially reduced for the DAC method, since it is not necessary to store any large matrix system.

Copyright © 2007 John Wiley & Sons, Ltd.

DOI: 10.1002/fld

Compared to a SGFEM, additional computational effort is required in a stabilized method for the elementwise evaluation of the stabilization parameter and the additional matrix entries due to the stabilization terms. Additional computational effort of the DAC method compared to a SGFEM for the solution in a local subdomain comes into play due to the coarse-scale ‘solver’ and the inter-scale operators. In this respect, the injection operators do not cause any notable cost, and the temporal projection operator is also relatively cheap in terms of computational cost. More demanding in this context are the spatial reconstruction operator (i.e. the interpolation) and the coarse-scale ‘solver’.

If one focusses on the matrix solutions for this brief analysis, the following may be stated. Of course, on a one-processor machine, the computational cost for the DAC method is much higher than the one for a stabilized FEM. If we assume the CPU time to depend on the number of degrees of freedom, which here coincides with the number of nodes, as $\mathrm{CPU} = c \times n\mathrm{dof}^{\beta}$, the ratio of the CPU time for the DAC method and the stabilized FEM is

$$\frac{\mathrm{DAC}-\mathrm{CPU}}{\mathrm{SFEM}-\mathrm{CPU}} = (2) n_{\mathrm{ts}} N_{\mathrm{nod}} \left(\frac{3^d}{N_{\mathrm{nod}}} \right)^{\beta} = (2) 3^d n_{\mathrm{ts}} N_{\mathrm{nod}}^{1-\beta} \tag{11}$$

where the constant c and the exponent β are assumed being the same for the solvers used, for simplicity. The factor 2 in (11) comes into play for the predictor–corrector scheme, where two solver calls have to be done for each time step. It is obvious that only for a small number of fine-sale time steps within each coarse-scale time step n_{ts}, a very large number of coarse-scale nodes N_{nod}, and/or an unfavourable solver-efficiency exponent β, the DAC method can become competitive on a one-processor machine. However, for large-scale problems to be solved on parallel computers, the DAC method becomes much sooner competitive, since all matrix systems can be solved independently of each other. Thus, all inter-processor communication can be eliminated within the solver stage.

4. NUMERICAL TESTS

In this section, we report three different test cases: a 1-D convection–diffusion–reaction problem, a 2-D skew-convection problem with reaction, and a 2-D rotating pulse with internal layer. The first two test cases are usually solved as stationary problems. However, they will be calculated here using instationary algorithms, with the stationary solution representing the target solution. We are interested in the convection-dominated as well as the convection–absorption-dominated regime for various boundary conditions. In all test cases, the following parameters are not changed:

- the coarse-scale Courant number: $Cr(H, \Delta T) = 1$;
- the application of the Crank–Nicolson scheme to all coarse- and fine-scale time steps; and
- the spatial discretization of the fine-scale domains ω_I by four elements in each coordinate direction.

The evaluation of the multiscale method proceeds in two steps for the first two test cases: firstly, a purely spatial divide-and-conquer (SDAC) multiscale method is tested (i.e. a mere application of the spatial inter-scale operators with the fine-scale time step used to cover the complete simulation time), and secondly, the complete (spatial and temporal) DAC multiscale method is investigated.

Copyright © 2007 John Wiley & Sons, Ltd.

DOI: 10.1002/fld

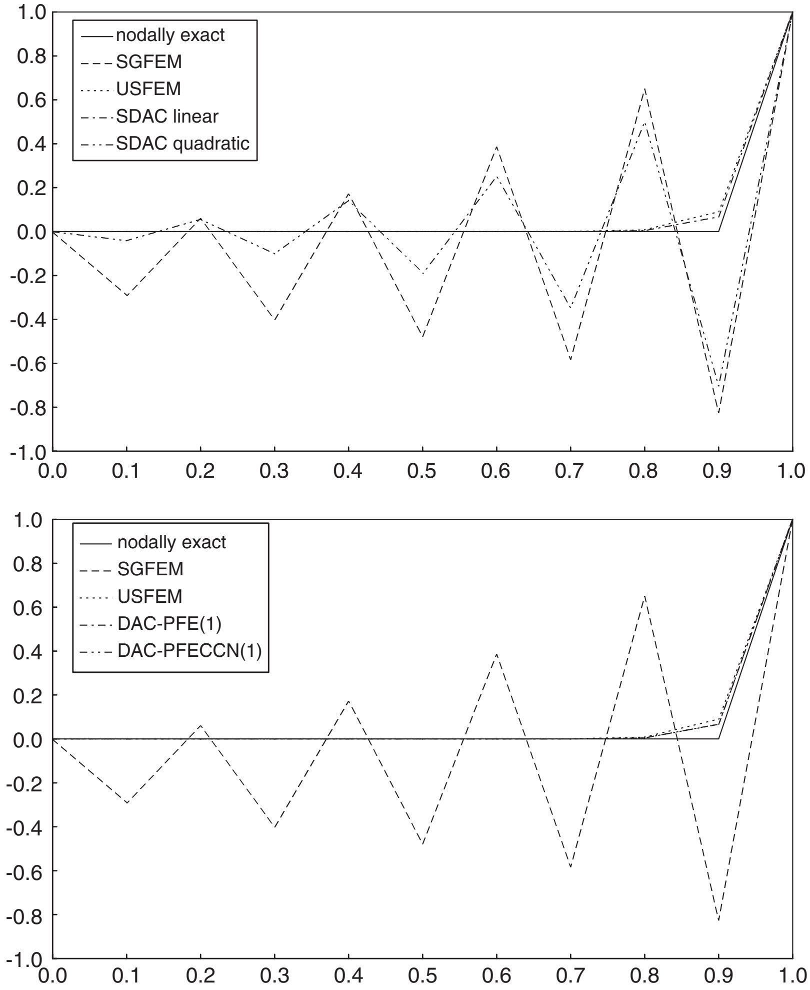

Figure 4. 1-D problem with $\phi_l = 0$ and $\phi_r = 1$: $Pe_e = 10$ and $Da_e^{\text{mod}} = 2$. Top: SDAC and bottom: DAC.

Proceeding this way, it is aimed at investigating the accuracy achievable with the spatial multiscale method and a full temporal resolution. Afterwards, it is intended to find out to which degree that accuracy may be maintained with the temporal part of the multiscale method additionally introduced. Results obtained with the multiscale methods are compared to results obtained with the SGFEM and the USFEM/VMM (with the stabilization parameter calculated according to [12]).

4.1. 1-D convection–diffusion–reaction problem

Four test series are reported for this case. At the left and right end of the problem domain $\Omega = [0, 1]$, the Dirichlet boundary conditions ϕ_l and ϕ_r, respectively, are prescribed. In the first three test series, $\phi_l = 0$ and $\phi_r = 1$, and in the fourth test series, $\phi_l = 1$ and $\phi_r = 0$. The convection velocity a, from left to right, is fixed to be 1.0 and f to be zero for all test series. The chosen simulation

Copyright © 2007 John Wiley & Sons, Ltd.

DOI: 10.1002/fld

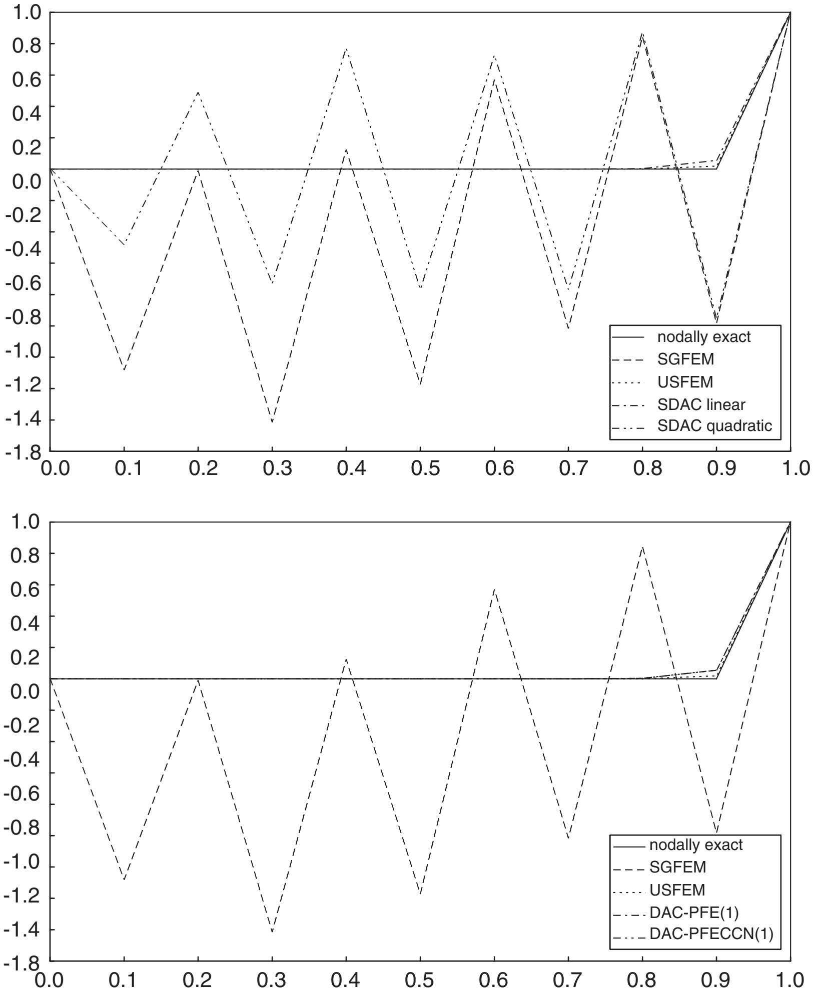

Figure 5. 1-D problem with $\phi_l = 0$ and $\phi_r = 1$: $Pe_e = 50$ and $Da_e^{\text{mod}} = 2$. Top: SDAC and bottom: DAC.

time $T = 1.0$ allows for one flow of the problem variable through Ω to reach the stationary solution. As initial condition ϕ^0, we choose a zero field with linear interpolation of a potential non-zero Dirichlet boundary condition within the element next to that boundary. The problem domain is spatially discretized by 10 uniform elements on the coarse-scale level such that $H = 0.1$. Due to the choice $Cr(H, \Delta T) = 1$, the coarse-scale time-step length is $\Delta T = 0.1$. All results are compared to the exact analytical solution interpolated from node to node.

In the first test series, the diffusion coefficient $\kappa = 0.005$ and the reaction coefficient $\sigma = 0$ are chosen, resulting in $Pe_e(H) = 10$ and, given the temporal discretization, $Da_e^{\text{mod}}(H) = 2$ based on the modified reaction coefficient as defined in (7). Thus, there is only 'artificial reaction' due to the temporal discretization in this series. The results obtained from this test series are displayed in Figure 4. As expected, the SGFEM solution exhibits oscillations at the present element Peclet

Copyright © 2007 John Wiley & Sons, Ltd.

DOI: 10.1002/fld

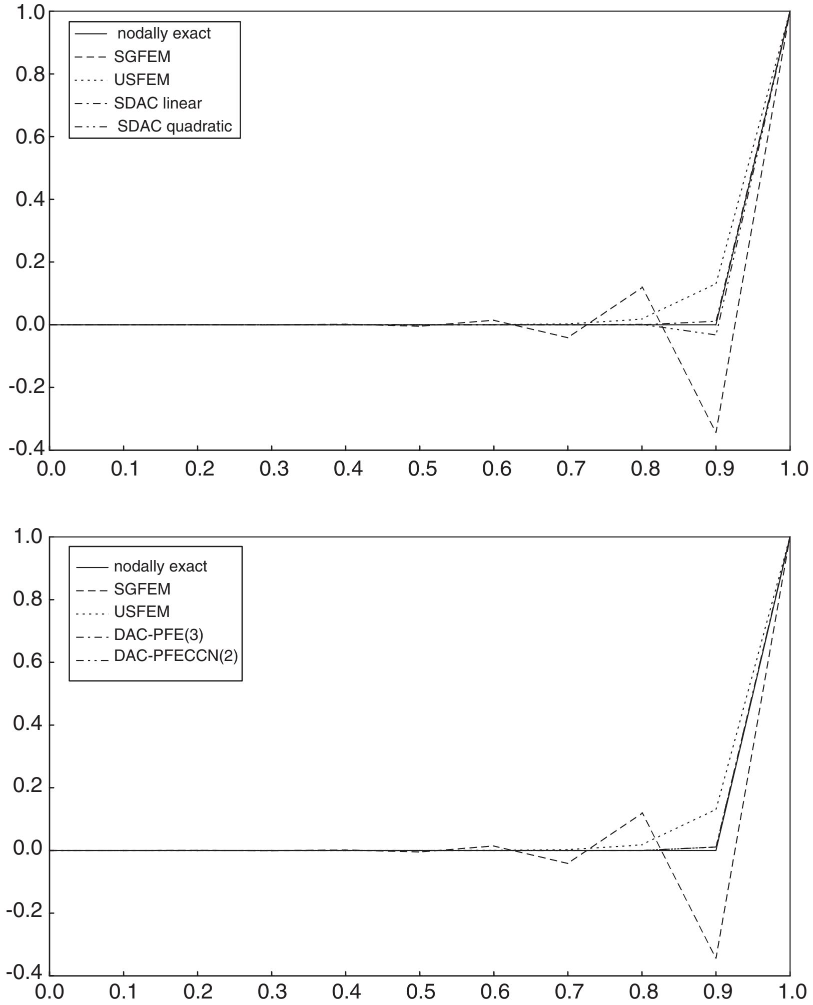

Figure 6. 1-D problem with $\phi_{\mathrm{l}} = 0$ and $\phi_{\mathrm{r}} = 1$: $Pe_e = 10$ and $Da_e^{\mathrm{mod}} = 10$. Top: SDAC and bottom: DAC.

number. The USFEM resembles the nodally exact solution with a slight deviation at the next to last node from the right boundary. The SDAC, shown in the top figure, yields a similar solution, with the deviation being even smaller, if a linear interpolation is used (SDAC linear), see Section 3.2.3. For the quadratic interpolation (SDAC quadratic), oscillations occur, although of smaller amplitude compared to the Galerkin solution. Thus, only the DAC with linear interpolation is investigated and displayed in the bottom figure. It may be observed that the accuracy of the SDAC is preserved by the DAC with one (1) fine-scale time step performed within each coarse-scale time step (i.e. the same number of time steps overall as in the non-multiscale method). This is true for both the projective Forward-Euler (PFE) approach and the projective predictor–corrector scheme with predictor Forward-Euler and corrector Crank–Nicolson (PFECCN).

Copyright © 2007 John Wiley & Sons, Ltd.

DOI: 10.1002/fld

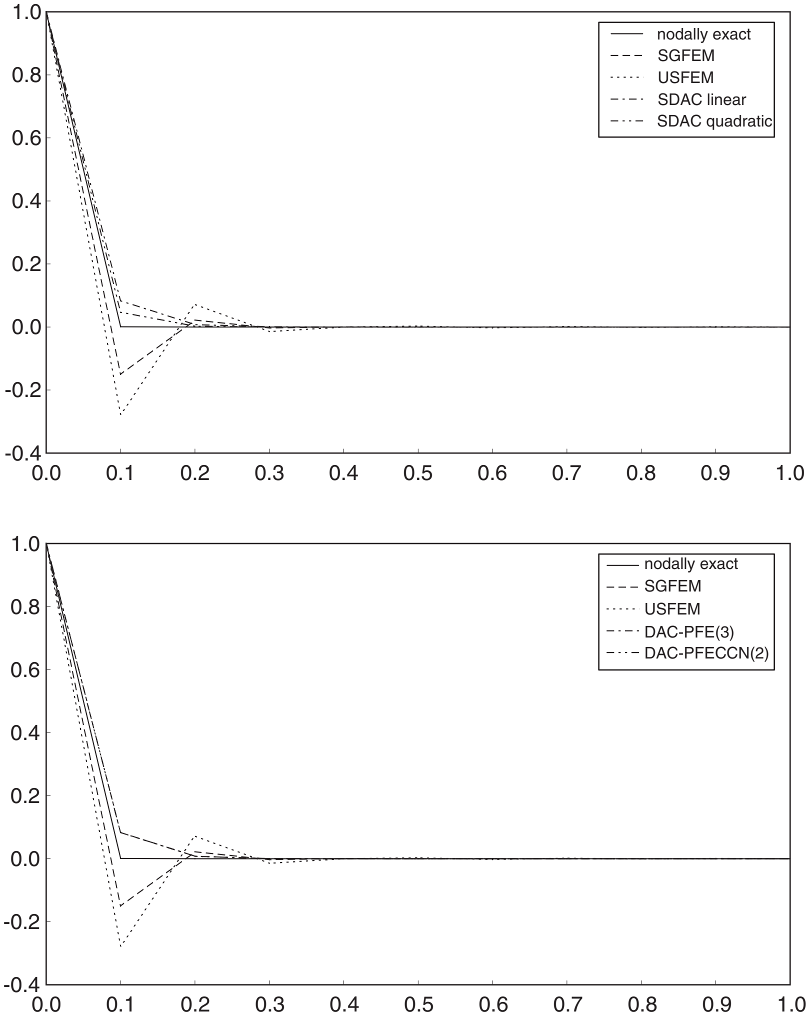

Figure 7. 1-D problem with $\phi_{\mathrm{l}}=1$ and $\phi_{\mathrm{r}}=0$: $Pe_e=10$ and $Da_e^{\mathrm{mod}}=10$. Top: SDAC and bottom DAC.

For the second test series, the diffusion coefficient is diminished by a factor of 5 compared to the first test series, resulting in $Pe_e(H)=50$ (i.e. a stronger dominance of convection). Results are reported in Figure 5. The oscillations are amplified both for the SGFEM and the SDAC with quadratic interpolation. Both the USFEM and the SDAC with linear interpolation well approximate the nodally exact solution, this time the USFEM being slightly closer to it. The accuracy obtained with the (linear) SDAC is again also achieved by both the DAC-PFE(1) and the DAC-PFECCN(1).

In the third series, ‘actual’ reaction is added by setting $\sigma=80.0$ compared to the first test series. As a result, $Da_e^{\mathrm{mod}}(H)=10$ based on the modified reaction coefficient. The results from this test series are provided in Figure 6. Besides the expected ‘Galerkin oscillations’ and an undershoot at the next to last node from the right boundary for the quadratic SDAC, it may be observed

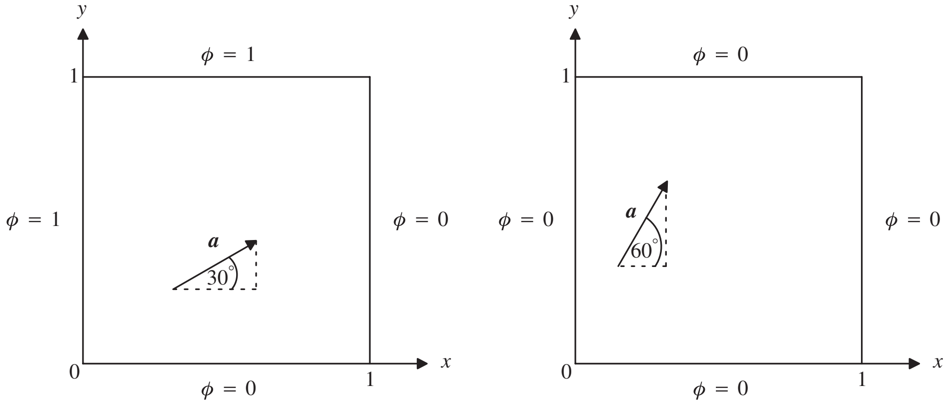

Figure 8. Problem domain with flow direction and the Dirichlet boundary conditions for skew convection. Left: test series 1–3 and right: test series 4.

that the USFEM considerably deviates from the nodally exact solution: we have already entered a problematic regime of large convection and large absorption for the stabilized method. The SDAC with linear interpolation does an excellent job, almost reproducing the nodally exact solution. With the temporal multiscale approach added, this accuracy can again be preserved. This time, however, three and two fine-scale time steps within each coarse-scale time step for PFE amd PFECCN, respectively, are necessary to achieve a non-oscillatory solution.

Finally, in the fourth series, we merely change the Dirichlet boundary conditions at the left and right end of the problem domain. This constitutes the (even more) problematic situation addressed in [1]: large convection and absorption in the presence of a negative streamwise gradient of the solution. The results from this particularly interesting test series are given in Figure 7. It is shown that the USFEM yields oscillations at the left boundary of even larger amplitude than the SGFEM. In the present test series, the quadratic SDAC exhibits no oscillations. Moreover, it yields a prediction even better than the linear SDAC. The accuracy is maintained after adding the temporal multiscale part both for the linear and the quadratic (not shown) interpolation.

4.2. Skew convection with reaction

This test case will be investigated analogously to the previous test case, that is, four test series are also reported for this case. The problem domain $\Omega = [0, 1] \times [0, 1]$ is displayed in Figure 8. On the left-hand side of Figure 8, the Dirichlet boundary conditions along with the direction of the convective velocity vector are depicted for the first three test series. The respective situation for the fourth test series is shown on the right-hand side of Figure 8. The norm of the convection velocity a is fixed to be 1.0 for all test series. The right-hand side f is assumed to be zero for the first three test series, but a source term $f = 1.0$ is included in the fourth test series. The chosen simulation time $T = 1.2$ allows for one flow of the problem variable through the domain along the prescribed velocity direction to reach the stationary solution. As initial condition ϕ^0, a zero field is chosen with linear interpolation of potential non-zero Dirichlet boundary conditions within the elements next to that boundary. The problem domain is spatially discretized by 10×10 uniform elements on the coarse-scale level such that $H = 0.1$ and, consequently, $\Delta T = 0.1$ as in the previous test case.

Copyright © 2007 John Wiley & Sons, Ltd.

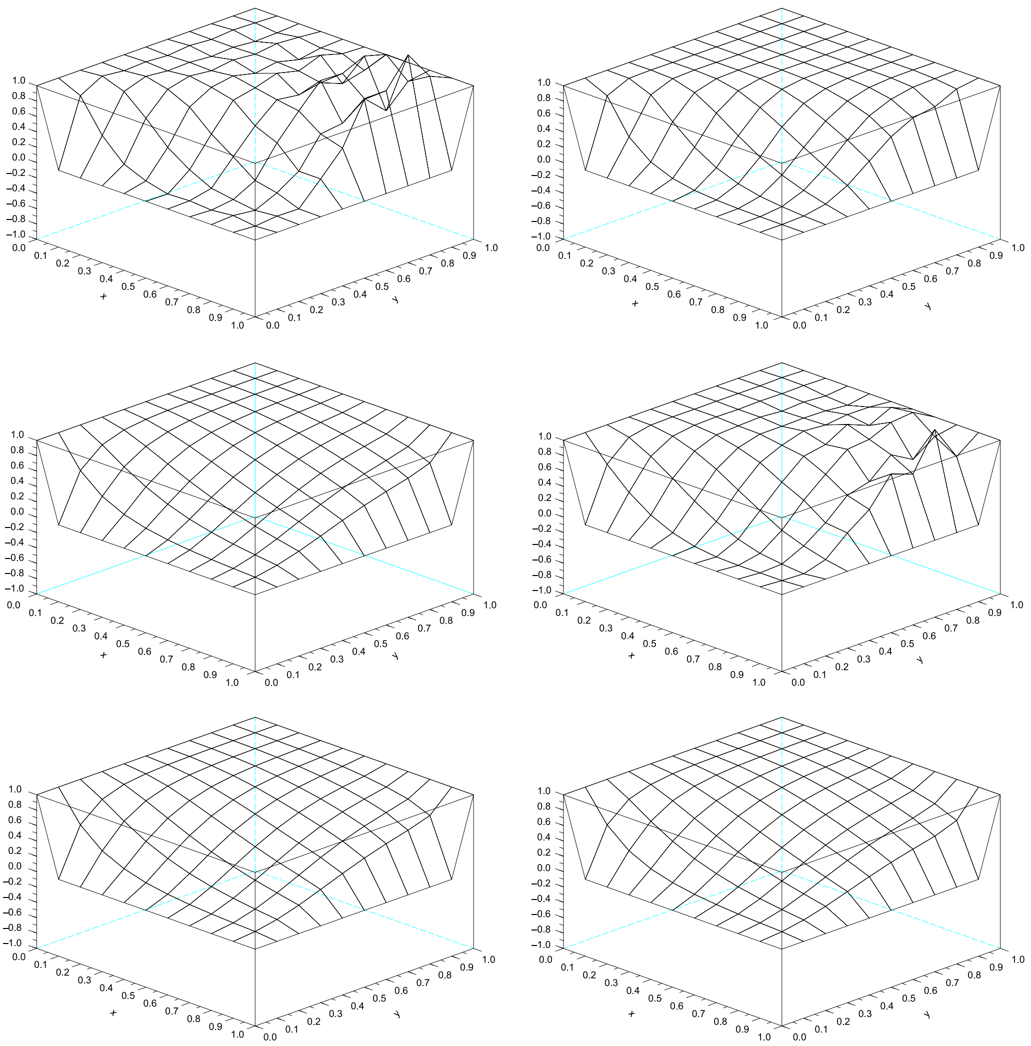

Figure 9. Skew convection at $Pe_e = 10$ and $Da_e^{\text{mod}} = 2$. From left to right and top to bottom: SGFEM; USFEM; SDAC linear; SDAC quadratic; DAC-PFE(3); and DAC-PFECCN(2).

The first test series resembles the first test series of the previous test case, that is, $Pe_e(H) = 10$ and $Da_e^{\text{mod}}(H) = 2$. The results obtained from this test series are displayed in Figure 9. Oscillations are observable for the solutions obtained with the SGFEM and the SDAC with quadratic interpolation. All other solutions are free of any solutions. The SDAC solution with linear interpolation is reproduced by DAC-PFE(3) and DAC-PFECCN(2). (In contrast to the respective 1-D series, more than one fine-scale time step within each coarse-scale time step is necessary for the comparable 2-D case.) An observable difference between the USFEM solution and the (S)DAC solutions is a more smoothed solution *via* the (S)DAC methods in the gradient area where the solution changes from one to zero.

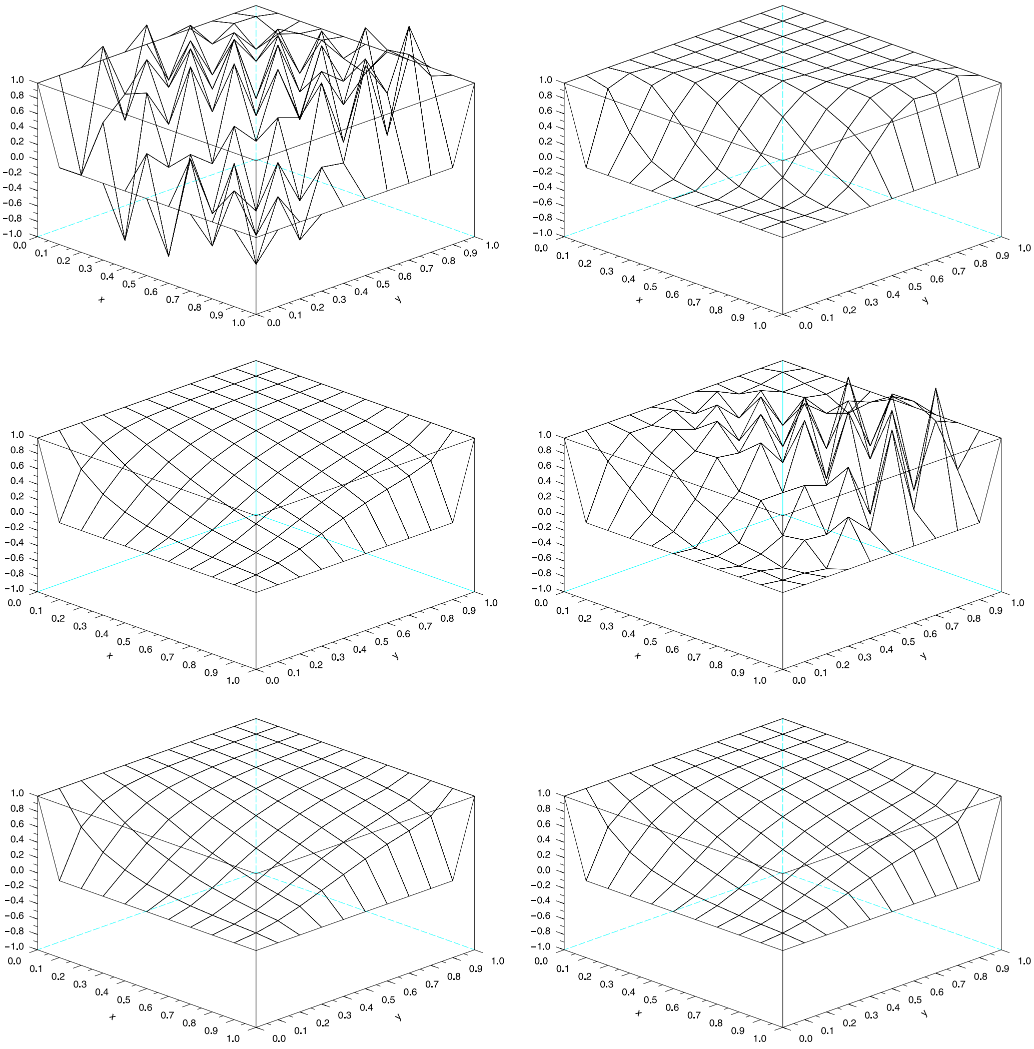

Figure 10. Skew convection at $Pe_e = 50$ and $Da_e^{\text{mod}} = 2$. From left to right and top to bottom: SGFEM; USFEM; SDAC linear; SDAC quadratic; DAC-PFE(11); and DAC-PFECCN(6).

Results from the second test series, where $Pe_e(H) = 50$, are reported in Figure 10. As in the respective 1-D series, the oscillations are amplified both for the standard Galerkin method and the SDAC with quadratic interpolation. Even the USFEM produces slight oscillations in the gradient area, particularly towards the boundary $x = 1.0$, with a maximum value of 1.1, which is larger than the expected maximum value of 1.0. The linear SDAC and DAC methods, again yielding similar results, provide a smoothed solution with no oscillations. However, a larger number of fine-scale time steps within each coarse-scale time step is needed to guarantee a stable solution. The actual number of required fine-scale time steps appears to be in a linear dependence on the Pe_e, but further investigations are necessary for confirmation.

Copyright © 2007 John Wiley & Sons, Ltd.

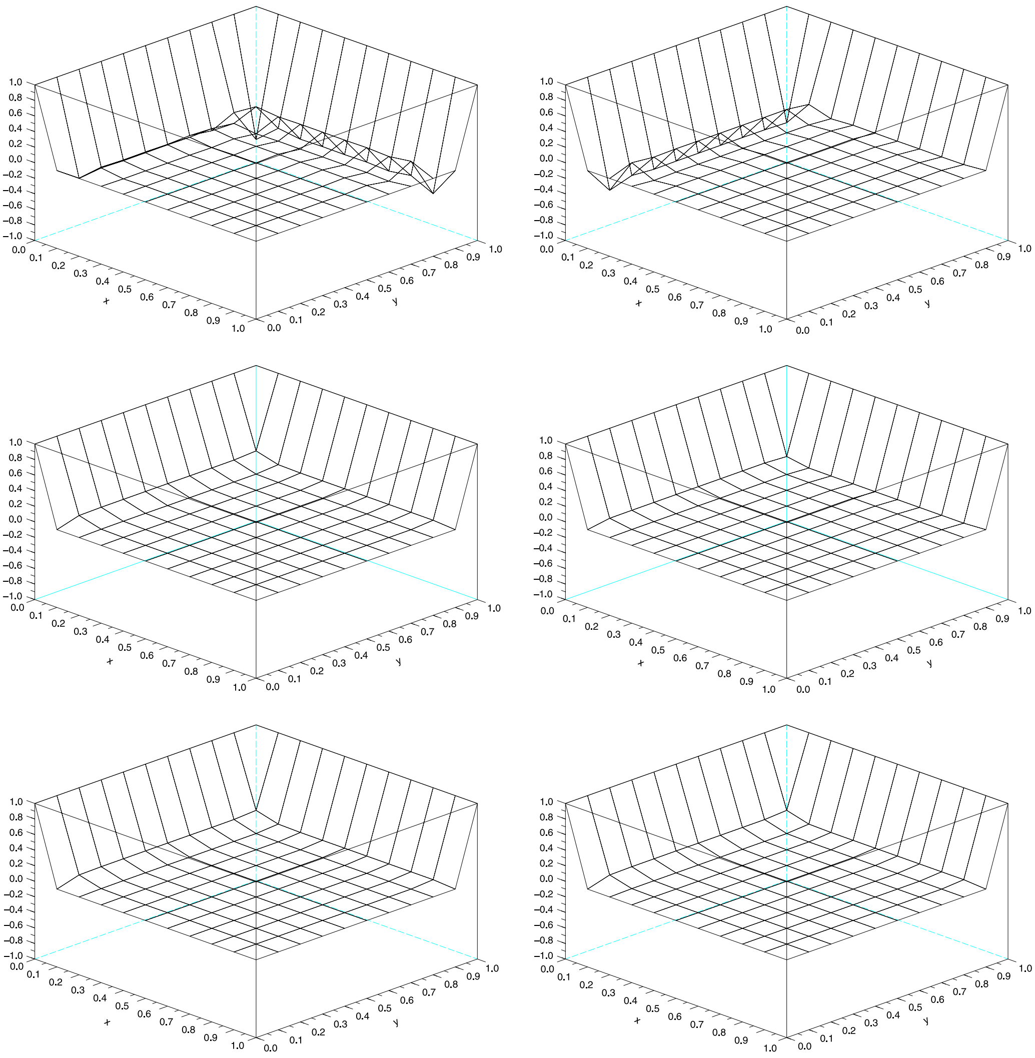

Figure 11. Skew convection at $Pe_e = 10$ and $Da_e^{\mathrm{mod}} = 10$. From left to right and top to bottom: SGFEM; USFEM; SDAC linear; SDAC quadratic; DAC-PFE(3); and DAC-PFECCN(2).

In the third series, we enter the problematic situation with large convection and absorption in 2-D, see Figure 11 for the results. It is shown that the USFEM yields oscillations towards the boundary $x = 0.0$, while the SGFEM produces oscillations towards the boundary $y = 1.0$. All (S)DAC methods are free of any oscillations and well approximate the expected solution field. The SDAC with quadratic interpolation appears to be closest to it.

In the fourth test series, we pose another challenge to the methods by changing the boundary conditions, the direction of the convection velocity, and adding a non-zero right-hand side compared to the second test series. The results are displayed in Figure 12. Huge oscillations occur when

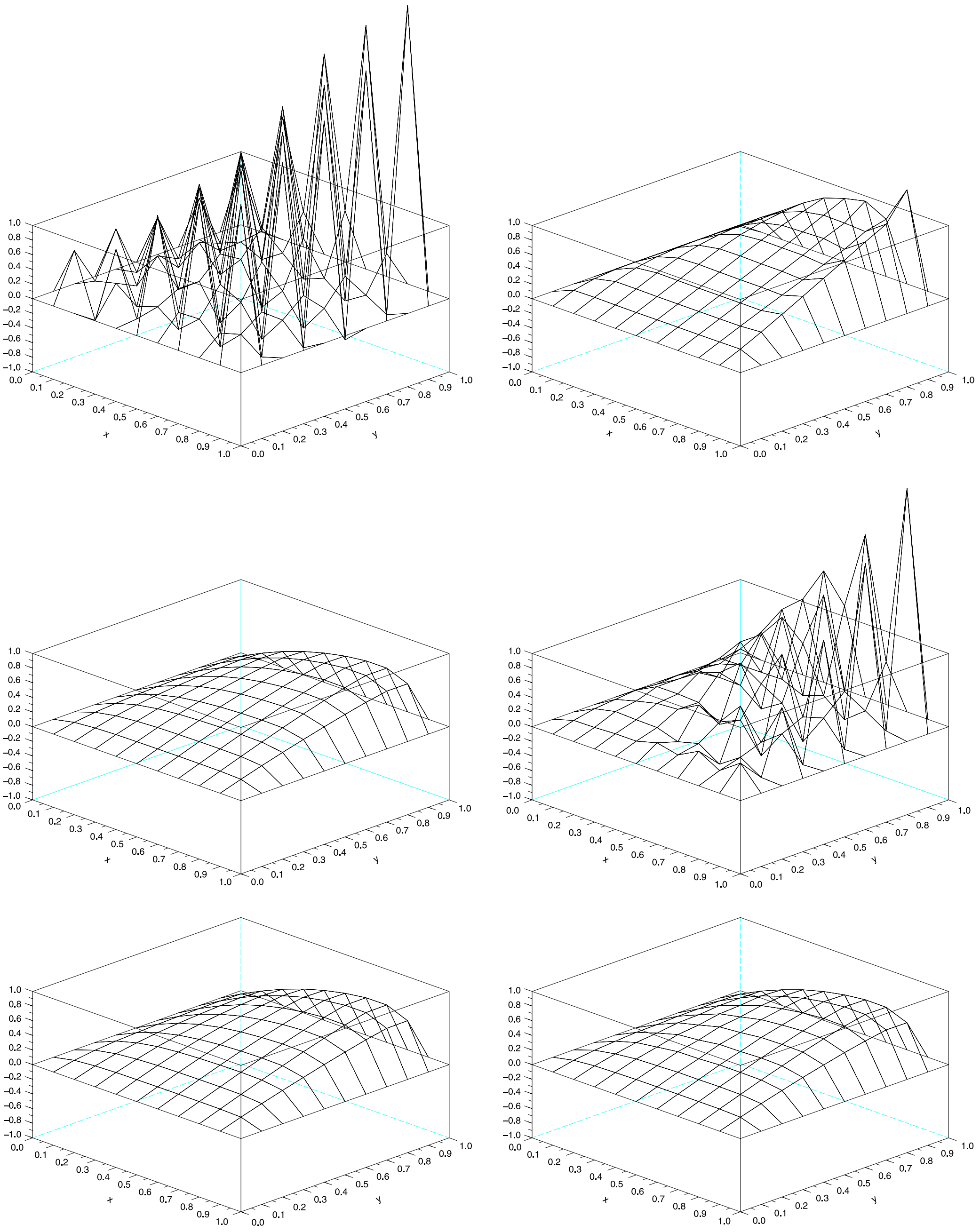

Figure 12. Skew convection at $Pe_e = 50$ and $Da_e^{\mathrm{mod}} = 2$ with source term. From left to right and top to bottom: SGFEM; USFEM; SDAC linear; SDAC quadratic; DAC-PFE(10); and DAC-PFECCN(5).

Copyright © 2007 John Wiley & Sons, Ltd.

DOI: 10.1002/fld

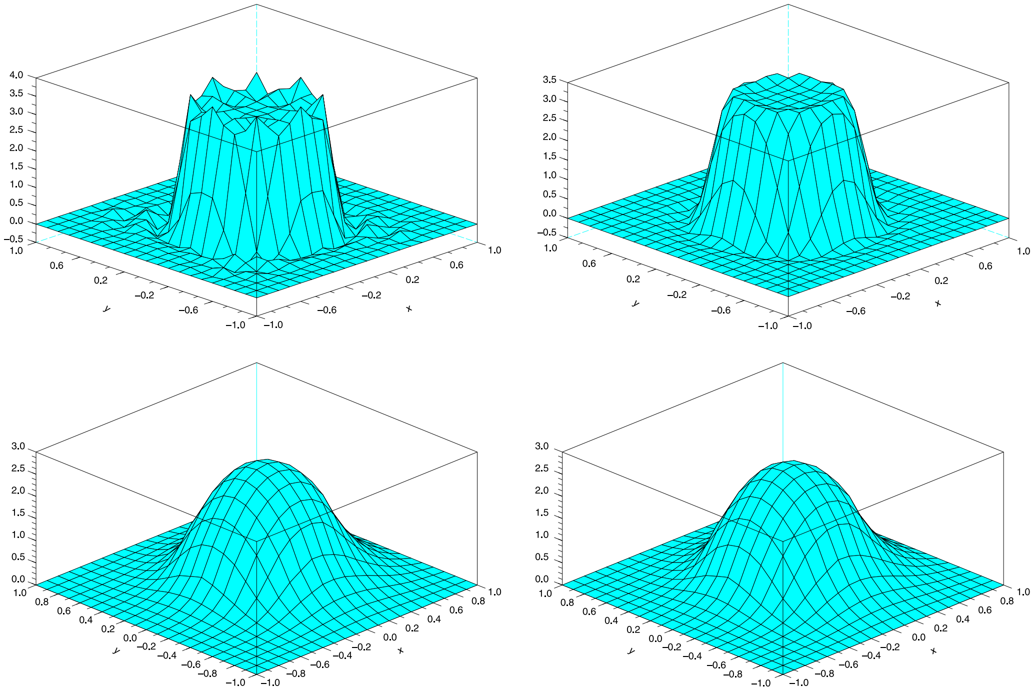

Figure 13. Rotating pulse with internal layer at $Pe_e = 50$ and $Da_e^{\text{mod}} = 2$. From left to right and top to bottom: SGFEM; USFEM; DAC-PFE(10); and DAC-PFECCN(6).

using the SGFEM and the SDAC with quadratic interpolation. Even the USFEM yields substantial oscillations towards the outflow boundary. The solutions obtained with the linear SDAC and, after adding the temporal part of the multiscale method, the DAC-PFE(10) as well as the DAC-PFECCN(5) again show no oscillations and produce a smoothed solution which probably predicts slightly too low values in the gradient area. The maximum value obtained with the help of the DAC method is about $\phi_{\max} = 0.71$, whereas the actual maximum value should be expected to be slightly below 1.0, a value the USFEM also would produce approximately, in case the remaining oscillation was taken out of account.

4.3. Rotating pulse with internal layer

The problem domain for this test case is $\Omega = [-1, 1] \times [-1, 1]$. Homogeneous Dirichlet boundary conditions are assumed on all boundaries. A velocity vector field $\mathbf{a} = \rho(r)(y, -x)$ with the function $\rho(r) = 1 - r^2$ (radius $r = \sqrt{x^2 + y^2}$ from the origin of the coordinate system) for $r \leqslant 1.0$ and $\rho(r) = 0$ for $r > 1.0$ is prescribed. The maximum velocity $a = 0.385$ occurs at a radius $r = 0.577$. The right-hand side is assumed to be $f = 1.0$ if $r \leqslant 0.5$ and zero elsewhere. The simulation is run until $T = 3.1 \approx \pi$. A zero field is chosen as initial condition ϕ^0. The problem domain is spatially discretized by 20×20 uniform elements on the coarse-scale level such that $H = 0.1$ and, consequently, $\Delta T = 0.1$ as in the previous test cases. The diffusion coefficient is chosen to be $\kappa = 0.0003849$, resulting in $Pe_e(H) = 50$. Only artificial reaction due to the temporal discretization

Copyright © 2007 John Wiley & Sons, Ltd.

DOI: 10.1002/fld

is considered ($Da_e^{\text{mod}}(H) = 2$), since we have already learned about the good performance of the DAC method in the presence of large absorption for the afore investigated problems. In the present case, we would rather like to reveal the still existing shortcoming of the DAC method in the case of large convection without large absorption.

The results obtained from this test case are shown in Figure 13. The interior layer to be resolved appears at a radius of about 0.5. The SGFEM is not able to resolve the interior layer without oscillations due to the high element Peclet number. The maximum and minimum values of the solution field obtained with the SGFEM are $\phi_{\max} = 3.64$ and $\phi_{\min} = -0.35$, respectively. The USFEM removes most of the oscillations and provides a good approximation of the (expected) solution field, although some slight oscillations still occur ($\phi_{\max} = 3.30$, $\phi_{\min} = -0.15$). The DAC-PFE and DAC-PFECCN remove all oscillations ($\phi_{\min} = 0.0$), but the shortcoming is also clearly notable: the interior layer is smoothed out too much. The maximum values, which is located exactly at the origin of the coordinate system in contrast to the other methods, are $\phi_{\max} = 2.43$ and 2.47, respectively, and thus about 25% smaller than the maximum value predicted by the USFEM.

5. CONCLUSIONS

A multiscale method has been presented which aims at providing a stable and accurate solution of transient convection–diffusion–reaction problems. The basic feature of the presented method culminates in its name: 'divide and conquer', that is, the problem is partitioned into small (fine-scale) problems, and the assembly of the solutions to the partitioned problems constitutes the overall (coarse-scale) solution. Phrased in the wording of [25], where the basic idea for such an approach has been formulated, although not aimed at the particular problem of a convection–diffusion–reaction equation, one may describe the situation as follows: we cannot solve (spatially) large-scale problems over long time intervals, so we solve (spatially) small-scale problems over short time intervals and try to cover the 'gaps' in space and time.

Two main goals have led to the development of the present method: a desired independence of any heuristic parameter such as the stabilization parameter in stabilized methods and a desire for a coherent multiscale strategy in space *and* time, which cannot be found in most other multiscale approaches in the literature. The components of the present method are all standard and relatively simple: the SGFEM and the one-step-θ scheme for the fine-scale solution, an extrapolation for advancing the coarse scales and the spatial and temporal inter-scale operators are constituted by relatively simple injection, interpolation, and extrapolation operators. In terms of computational cost, it has been analysed in Section 3.3 that the proposed method can become competitive to stabilized methods for large-scale computations carried out on parallel computers. In terms of storage requirements and solver technology, the presented method is less demanding in any case due to the fact that it is not necessary to store a large matrix and to employ a sophisticated solver.

The new approach has been tested for three test cases, of which the first two were further divided into several test series with varying element Peclet and Damköhler numbers. It was intended to investigate the two most problematic regimes: the 'classical' challenging regime of large convection and the regime of large convection *and* large absorption. The latter one was identified in [1] to be particularly challenging in the case of an additional presence of a negative streamwise gradient of the solution. In fact, stabilized finite element methods have not been able to guarantee stable solutions for this regime. As a result of all test cases performed, a major

Copyright © 2007 John Wiley & Sons, Ltd.

DOI: 10.1002/fld

achievement in terms of stability and a still to be noted shortcoming in terms of accuracy may be noted for the present method.

The present method with linear interpolation as the spatial reconstruction operator provides solutions which are *completely* free of oscillations for *any* regime. In contrast, standard stabilized methods suffer from (slight) oscillations in the large-convection regime, as observed for the 2-D test cases. Moreover, they are not able to provide an acceptable solution in the large-convection-and-absorption regime. The notable shortcoming of the proposed multiscale method reveals itself in the too smooth resolution of the regions where a sharp gradient of the solution field occurs. The reason for this may be traced back, among other things, to the choice of a simple linear interpolation as the spatial reconstruction operator. The authors are confident that already slightly more sophisticated reconstruction operators may provide a better resolution. The investigation of this option will be the subject of future work.

The temporal projection operator, representing the other of the two most important inter-scale operators investigated in the present study, is able to fully 'preserve' the spatial approximation quality in time, if a sufficient number of small-scale time steps within each coarse-scale time step is taken. Both the Forward-Euler extrapolation and the Forward-Euler/Crank–Nicolson predictor–corrector extrapolation work equally well in the sense that the predictor–corrector extrapolation, which needs twice the computational effort compared to the Forward-Euler extrapolation for one time step, required about half the number of time steps in all test cases. Since the computational effort is comparable, as a result, it is suggested to use the Forward-Euler extrapolation due to its slightly simpler implementation.

REFERENCES

1. Hauke G. A simple subgrid scale stabilized method for the advection–diffusion–reaction equation. *Computer Methods in Applied Mechanics and Engineering* 2002; **191**:2925–2947.
2. Tezduyar TE, Park YJ. Discontinuity capturing finite element formulations for nonlinear convection–diffusion–reaction equations. *Computer Methods in Applied Mechanics and Engineering* 1986; **59**:307–325.
3. Tezduyar TE, Park YJ, Deans HA. Finite element procedures for time-dependent convection–diffusion–reaction systems. *International Journal for Numerical Methods in Fluids* 1987; **7**:1013–1033.
4. Corsini A, Rispoli F, Santoriello A, Tezduyar TE. Finite element procedures for time-dependent convection–diffusion–reaction systems. *Computational Mechanics* 2006; **38**:356–364.
5. John V, Knobloch P. On discontinuity-capturing methods for convection–diffusion equations. In *Proceedings of the Sixth European Conference on Numerical Mathematics* (*ENUMATH 2005*), Bermudez de Castro A, Gomez D, Quintela P, Salgado P (eds). Springer: Berlin, 2006; 336–344.
6. Onate E, Miquel J, Hauke G. Stabilized formulation for the advection–diffusion–absorption equation using finite calculus and linear finite elements. *Computer Methods in Applied Mechanics and Engineering* 2006; **195**:3926–3946.
7. Brooks AN, Hughes TJR. Streamline upwind/Petrov–Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier–Stokes equations. *Computer Methods in Applied Mechanics and Engineering* 1982; **32**:199–259.
8. Hughes TJR, Tezduyar TE. Finite element methods for first-order hyperbolic systems with particular emphasis on the compressible Navier–Stokes equations. *Computer Methods in Applied Mechanics and Engineering* 1984; **45**:217–284.
9. Johnson C, Nävert U, Pitkäranta P. Finite element methods for linear hyperbolic problems. *Computer Methods in Applied Mechanics and Engineering* 1984; **45**:285–312.
10. Hughes TJR, Franca LP, Hulbert M. A new finite element formulation for computational fluid dynamics: VIII. the Galerkin/least-squares method for advective–diffusive equations. *Computer Methods in Applied Mechanics and Engineering* 1989; **73**:173–189.
11. Franca LP, Farhat C. Bubble functions prompt unusual stabilized finite element methods. *Computer Methods in Applied Mechanics and Engineering* 1995; **123**:299–308.

Copyright © 2007 John Wiley & Sons, Ltd.

DOI: 10.1002/fld

12. Franca LP, Valentin F. On an improved unusual stabilized finite element method for the advective–reactive–diffusive equation. *Computer Methods in Applied Mechanics and Engineering* 2000; **190**:1785–1800.
13. Hughes TJR. Multiscale phenomena: Green's functions, the Dirichlet-to-Neumann formulation, subgrid scale models, bubbles and the origins of stabilized methods. *Computer Methods in Applied Mechanics and Engineering* 1995; **127**:387–401.
14. Hauke G, Doweidar MH. Fourier analysis of semi-discrete and space–time stabilized methods for the advective–diffusive–reactive equation: II. SGS. *Computer Methods in Applied Mechanics and Engineering* 2005; **194**:691–725.
15. Codina R. Comparison of some finite element methods for solving the diffusion–convection–reaction equation. *Computer Methods in Applied Mechanics and Engineering* 1982; **156**:185–210.
16. Donea J, Huerta A. *Finite Element Methods for Flow Problems*. Wiley: Chichester, 2003.
17. Brezzi F, Franca LP, Hughes TJR, Russo A. $b=\int g$. *Computer Methods in Applied Mechanics and Engineering* 1997; **145**:329–339.
18. Tezduyar TE, Osawa Y. Finite element stabilization parameters computed from element matrices and vectors. *Computer Methods in Applied Mechanics and Engineering* 2000; **190**:411–430.
19. Tezduyar TE, Sathe S. Enhanced-discretization successive update method (EDSUM). *International Journal for Numerical Methods in Fluids* 2005; **47**:633–654.
20. Tezduyar TE, Sathe S. Enhanced-discretization selective stabilization procedure (EDSSP). *Computational Mechanics* 2006; **38**:456–468.
21. Harari I. Stability of semidiscrete formulations for parabolic problems at small time steps. *Computer Methods in Applied Mechanics and Engineering* 2004; **193**:1491–1516.
22. Förster C, Wall WA, Ramm E. On residual based stabilisation methods for transient problems at small time increments. Preprint, 2007.
23. Gravemeier V, Lenz S, Wall WA. Towards a framework for multiscale methods in computational mechanics: building blocks of existing methods. Preprint, 2007.
24. Tezduyar TE, Sathe S. Enhanced-discretization space–time technique (EDSTT). *Computer Methods in Applied Mechanics and Engineering* 2004; **193**:1385–1401.
25. Kevrekidis IG, Gear CW, Hyman JM, Kevrekidis PG, Runborg O, Theodoropoulos C. Equation-free, coarse-grained multiscale computation: enabling microscopic simulators to perform system-level analysis. *Communications in Mathematical Sciences* 2003; **1**:715–762.
26. Gear CW, Kevrekidis IG. Projective methods for stiff differential equations: problems with gaps in their eigenvalue spectrum. *SIAM Journal on Scientific Computing* 2003; **24**:1091–1106.
27. Lee SL, Gear CW. Second-order accurate projective integrators for multiscale problems. *Journal of Computational and Applied Mathematics* 2007; **201**:258–274.
28. E W, Engquist B. The heterogeneous multiscale methods. *Communications in Mathematical Sciences* 2003; **1**:87–132.
29. Gresho PM, Lee RL. Don't suppress the wiggles—they're telling you something. *Computers and Fluids* 1981; **9**:223–235.
30. Hegarty AF, Miller JJH, O'Riordan E, Shishkin GI. Special meshes for finite difference approximations to an advection–diffusion equation with parabolic layers. *Journal of Computational Physics* 1995; **117**:47–54.
31. Bazilevs Y, Hughes TJR. Weak imposition of Dirichlet boundary conditions in fluid mechanics. *Computers and Fluids* 2007; **36**:12–26.
32. Gear CW, Kevrekidis IG. Projective methods for stiff differential equations: problems with gaps in their eigenvalue spectrum. *NEC Research Institute 2001-029*, Princeton, NJ, 2001.

Copyright © 2007 John Wiley & Sons, Ltd.

DOI: 10.1002/fld