Comput. Mech. (2006) 38: 456–468
DOI 10.1007/s00466-006-0056-7

ORIGINAL PAPER

Tayfun E. Tezduyar · Sunil Sathe

Enhanced-discretization selective stabilization procedure (EDSSP)

Received: 23 January 2006 / Accepted: 14 February 2006 / Published online: 29 March 2006

Abstract The enhanced-discretization selective stabilization procedure (EDSSP) provides a multiscale framework for applying numerical stabilization selectively at different scales. The EDSSP is based on the enhanced-discretization, multiscale function space concept underlying the enhanced-discretization successive update method (EDSUM). The EDSUM is a multi-level iteration method designed for computation of the flow behavior at small scales. It has a built-in mechanism for transferring flow information between the large and small scales in a fashion consistent with the discretizations resulting from the underlying stabilized formulations. This is accomplished without assuming that the small-scale trial or test functions vanish at the borders between the neighboring large-scale elements of the enhanced-discretization zones. This facilitates unrestricted movement of small-scale flow patterns from one large-scale element to another without any constraints at the border between the two elements. The enhanced-discretization concept underlying the EDSUM can also facilitate using different stabilizations for equations or unknowns corresponding to different scales. In this paper we propose a version of the EDSSP where the SUPG and PSPG stabilizations are used for unknowns corresponding to both the large and small scales but the discontinuity-capturing stabilizations are used for unknowns corresponding to only the small scales. We also propose a version where a linear discontinuity-capturing is used for the small-scale unknowns and a nonlinear discontinuity-capturing is used for the large-scale unknowns. We evaluate the performances of these versions of the EDSSP with test problems governed by the advection–diffusion equations.

Keywords Flow simulation · Enhanced discretization · Multiscale · Selective stabilization · Discontinuity capturing

T.E. Tezduyar (✉) · S. Sathe
Mechanical Engineering,
Rice University – MS 321
6100 Main Street, Houston,
TX 77005, USA
E-mail: tezduyar@rice.edu
E-mail: sathe@rice.edu

1 Introduction

The enhanced-discretization techniques developed in recent years (see [1, 2]) for flow computations include enhanced-discretization interface-capturing technique (EDICT), enhanced-discretization space–time technique (EDSTT), and enhanced-discretization successive update method (EDSUM). These enhanced-discretization techniques are built around the core formulations that we rely on in simulation and modeling of complex flow problems. These core methods include the stabilized finite element techniques such as the streamline-upwind/Petrov-Galerkin (SUPG) [3–6] and pressure-stabilizing/Petrov-Galerkin (PSPG) [7, 8] formulations, as well as interface-tracking and interface-capturing techniques (see [1, 2, 7, 9–12]). The PSPG formulation is based on an earlier pressure-stabilizing formulation [13] that was designed for Stokes flows.

Stabilized formulations prevent numerical instabilities in solving problems with high Reynolds or Mach numbers and shocks or thin boundary layers, as well as when using equal-order interpolation functions for velocity and pressure. The SUPG and PSPG formulations are among the stabilized methods that achieve these objectives without introducing excessive numerical dissipation. For problems involving shocks or thin boundary layers, in some cases it would be desirable to supplement the SUPG and PSPG formulations with shock-capturing or discontinuity-capturing (DC) stabilizations. Using such supplementary stabilization with the SUPG formulation goes almost as far back as the development of the SUPG formulation. Use of shock-capturing and DC in the context of advection–diffusion problems and compressible flows was reported in 1986 (see [14–16]).

The EDICT was introduced in [17] to increase accuracy in representing an interface in an interface-capturing technique. In the EDICT, in the stabilized formulations of the flow and advection equations to be solved, the function spaces used are based on enhanced discretization at and near the interface. A subset of the elements in the base mesh, Mesh-1, are identified as those at and near the interface. A more refined mesh, Mesh-2, is constructed by patching together

second-level meshes generated over each element in this subset. The trial and test function spaces have components coming from Mesh-1 and Mesh-2. The subset of Mesh-1 elements over which Mesh-2 is built is re-defined not every time step but with sufficient frequency to keep the interface always enveloped in.

The EDSTT was introduced in [18, 19] as an enhancement in time discretization in the context of a space–time formulation. The trial and test function spaces have components coming from the space–time meshes Mesh-1 and Mesh-2. The EDSTT was developed to have more flexibility in carrying out time-accurate computations of fluid–structure interactions where we find it necessary to use smaller time steps for the structural dynamics part. In general, EDSTT can be used in time-accurate computations where, for whatever reason, we require smaller time steps in certain parts of the fluid domain. Test computations for the EDSTT were reported in [20].

EDSUM was proposed in [11, 19, 21, 22] as a multiscale iteration method for computation of the flow behavior at small scales. It has a built-in mechanism for transferring flow information between the large and small scales. This information transfer is consistent with the discretizations resulting from the underlying stabilized formulations. This is accomplished without assuming that the small-scale trial or test functions vanish at the borders between the neighboring large-scale elements of the enhanced-discretization zones. The small-scale flow patterns can move from one large-scale element to another without any constraints at the border between the two elements. For iterative solution of the coupled equations systems, the multiscale function space concept was exploited for designing preconditioners based on successive updates corresponding to the large and small scales (see [11, 19, 21, 22]). Test computations with the EDSUM were reported in [23].

The multiscale function space concept underlying the EDSUM also provides a multiscale framework for employing different stabilizations for equations or unknowns corresponding to different scales. Motivated by earlier work on multiscale methods (see [24–31]), the enhanced-discretization selective stabilization procedure (EDSSP) was introduced in [21–23] as a formulation where numerical stabilization is applied selectively at different scales. In a special version of the EDSSP proposed in [21–23], the SUPG and PSPG stabilizations are applied to equations corresponding to both the large and small scales but the DC stabilization is applied to equations corresponding to only the small scales. In this paper, we propose a version of the EDSSP where the SUPG and PSPG stabilizations are applied to both the large- and small-scale unknowns while the DC is applied to only the small-scale unknowns. We also propose in this paper a version where a linear DC is applied to the small-scale unknowns and a nonlinear DC is applied to the large-scale unknowns. Test computations with these ideas are reported for the first time in this paper.

In Sects. 2 and 3 we review the governing equations and the standard stabilized formulations. The enhanced-discretization stabilized formulations are described in Sect. 4, and the construction of the function spaces in Sect. 5. In Sect. 6 we describe the versions of the selective stabilization we propose. Test computations for problems governed by the advection–diffusion equation are reported in Sect. 7, and the concluding remarks are presented in Sect. 8.

2 Governing equations

2.1 Navier–Stokes equations of incompressible flows

Let $\Omega \subset I\!R^{n_{\rm sd}}$ be the spatial domain with boundary Γ, and $(0, T)$ be the time domain. The Navier–Stokes equations of incompressible flows can be written on Ω and $\forall\, t \in (0, T)$ as

$$\rho\left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u}\cdot\nabla\mathbf{u} - \mathbf{f}\right) - \nabla\cdot\boldsymbol{\sigma} = 0, \tag{1}$$

$$\nabla\cdot\mathbf{u} = 0, \tag{2}$$

where ρ, $\mathbf{u}$, and $\mathbf{f}$ are the density, velocity, and the external force, respectively. The stress tensor $\boldsymbol{\sigma}$ is defined as

$$\boldsymbol{\sigma}(p, \mathbf{u}) = -p\mathbf{I} + 2\mu\boldsymbol{\varepsilon}(\mathbf{u}). \tag{3}$$

Here p is the pressure, $\mathbf{I}$ is the identity tensor, $\mu = \rho\nu$ is the viscosity, ν is the kinematic viscosity, and $\boldsymbol{\varepsilon}(\mathbf{u})$ is the strain-rate tensor:

$$\boldsymbol{\varepsilon}(\mathbf{u}) = \tfrac{1}{2}\left((\nabla\mathbf{u}) + (\nabla\mathbf{u})^{\rm T}\right). \tag{4}$$

The essential and natural boundary conditions for Eq. (1) are represented as

$$\mathbf{u} = \mathbf{g} \quad \text{on } \Gamma_{\rm g}, \qquad \mathbf{n}\cdot\boldsymbol{\sigma} = \mathbf{h} \quad \text{on } \Gamma_{\rm h}, \tag{5}$$

where Γ_g and Γ_h are complementary subsets of the boundary Γ, $\mathbf{n}$ is the unit normal vector, and $\mathbf{g}$ and $\mathbf{h}$ are given functions. A divergence-free velocity field $\mathbf{u}_0(\mathbf{x})$ is specified as the initial condition.

2.2 Advection–diffusion equation

As a model equation possessing some of the significant features of Eq. (1), we consider the following time-dependent advection–diffusion equation, written on Ω and $\forall\, t \in (0, T)$ as

$$\frac{\partial\phi}{\partial t} + \mathbf{u}\cdot\nabla\phi - \nabla\cdot(\nu\nabla\phi) = 0, \tag{6}$$

where ϕ represents the quantity being transported (e.g., temperature, concentration), and ν is the diffusivity, which is separate from (but in mathematical significance very comparable to) the ν representing the kinematic viscosity. The essential and natural boundary conditions associated with Eq. (6) are represented as

$$\phi = \mathrm{g} \quad \text{on } \Gamma_{\rm g}, \qquad \mathbf{n}\cdot\nu\nabla\phi = \mathrm{h} \text{ on } \Gamma_{\rm h}. \tag{7}$$

A function $\phi_0(\mathbf{x})$ is specified as the initial condition.

3 Standard stabilized formulations

3.1 Advection–diffusion equation

For the advection–diffusion equation given by Eq. (6), let us assume that we have constructed some suitably-defined finite-dimensional trial solution and test function spaces $\mathcal{S}_\phi^h$ and $\mathcal{V}_\phi^h$. The stabilized finite element formulation can then be written as follows: find $\phi^h \in \mathcal{S}_\phi^h$ such that $\forall\, w^h \in \mathcal{V}_\phi^h$:

$$\int_\Omega w^h \left(\frac{\partial \phi^h}{\partial t} + \mathbf{u}^h \cdot \nabla \phi^h\right) \mathrm{d}\Omega + \int_\Omega \nabla w^h \cdot \nu \nabla \phi^h \mathrm{d}\Omega - \int_{\Gamma_h} w^h \mathrm{h}^h \mathrm{d}\Gamma + \sum_{e=1}^{n_{\mathrm{el}}} \int_{\Omega^e} \tau_{\mathrm{SUPG}} \mathbf{u}^h \cdot \nabla w^h \left(\frac{\partial \phi^h}{\partial t} + \mathbf{u}^h \cdot \nabla \phi^h - \nabla \cdot \left(\nu \nabla \phi^h\right)\right) \mathrm{d}\Omega + S_{\mathrm{DC}} = 0, \tag{8}$$

where the DC stabilization is defined as

$$S_{\mathrm{DC}} = \sum_{e=1}^{\mathrm{n_{el}}} \int_{\Omega^e} \nabla w^h \cdot \nu_{\mathrm{DC}} \nabla \phi^h \mathrm{d}\Omega. \tag{9}$$

Here n_{el} is the number of elements, Ω^e is the domain for element e, τ_{SUPG} is the SUPG stabilization parameter, and ν_{DC} is the DC parameter. For various ways of calculating τ_{SUPG}, see [19, 32, 33]. For early examples of ways of calculating ν_{DC}, see [14, 15])

3.2 Navier–Stokes equations of incompressible flows

For the Navier–Stokes equations of incompressible flows, given by Eqs. (1) and (2), let us assume that we have some suitably-defined finite-dimensional trial solution and test function spaces for velocity and pressure: $\mathcal{S}_{\mathbf{u}}^h$, $\mathcal{V}_{\mathbf{u}}^h$, $\mathcal{S}_p^h$, and $\mathcal{V}_p^h (= \mathcal{S}_p^h)$. The stabilized finite element formulation can then be written as follows: find $\mathbf{u}^h \in \mathcal{S}_{\mathbf{u}}^h$ and $p^h \in \mathcal{S}_p^h$ such that $\forall\, \mathbf{w}^h \in \mathcal{V}_{\mathbf{u}}^h$ and $\forall\, q^h \in \mathcal{V}_p^h$:

$$\int_\Omega \mathbf{w}^h \cdot \rho \left(\frac{\partial \mathbf{u}^h}{\partial t} + \mathbf{u}^h \cdot \nabla \mathbf{u}^h - \mathbf{f}^h\right) \mathrm{d}\Omega + \int_\Omega \boldsymbol{\varepsilon}(\mathbf{w}^h) : \boldsymbol{\sigma}(p^h, \mathbf{u}^h) \mathrm{d}\Omega - \int_{\Gamma_h} \mathbf{w}^h \cdot \mathbf{h}^h \mathrm{d}\Gamma + \int_\Omega q^h \nabla \cdot \mathbf{u}^h \mathrm{d}\Omega + \sum_{e=1}^{n_{\mathrm{el}}} \int_{\Omega^e} \frac{1}{\rho} \left[\tau_{\mathrm{SUPG}} \rho \mathbf{u}^h \cdot \nabla \mathbf{w}^h + \tau_{\mathrm{PSPG}} \nabla q^h\right] \cdot \left[Ł(p^h, \mathbf{u}^h) - \rho \mathbf{f}^h\right] \mathrm{d}\Omega + S_{\mathrm{DC}} = 0, \tag{10}$$

where

$$Ł(q^h, \mathbf{w}^h) = \rho \left(\frac{\partial \mathbf{w}^h}{\partial t} + \mathbf{u}^h \cdot \nabla \mathbf{w}^h\right) - \nabla \cdot \boldsymbol{\sigma}(q^h, \mathbf{w}^h), \tag{11}$$

$$S_{\mathrm{DC}} = \sum_{e=1}^{n_{\mathrm{el}}} \int_{\Omega^e} \rho \nabla \mathbf{w}^h : \left(\boldsymbol{\kappa}_{\mathrm{DC}} \cdot \nabla \mathbf{u}^h\right) \mathrm{d}\Omega. \tag{12}$$

Here τ_{PSPG} is the PSPG stabilization parameter and $\boldsymbol{\kappa}_{\mathrm{DC}}$ is the DC tensor. For various ways of calculating τ_{PSPG}, see [19, 32, 33]. For examples of ways of calculating $\boldsymbol{\kappa}_{\mathrm{DC}}$, see [19, 33, 34].

4 Enhanced-discretization stabilized formulations

To maintain the generality of the formulations, we allow for cases where the enhanced-discretization zone does not necessarily cover the entire domain and its shape and location may change during the computation. A subset of the elements in the base mesh, Mesh-1, are identified as those constituting the enhanced-discretization zone. A more refined mesh, Mesh-2, is constructed by patching together second-level meshes generated over each element in this subset. The trial and test functions will have two components each, one coming from Mesh-1 and the second one coming from Mesh-2.

4.1 Advection–diffusion equation

At a time level n, the trial function space corresponding to ϕ_n^h is denoted by $(\mathcal{S}_\phi^h)_n$, and the test function space corresponding to the advection–diffusion equation by $(\mathcal{V}_\phi^h)_n$. The subscript n indicates that the spatial discretizations corresponding to different time levels may be different. The enhanced-discretization stabilized formulation of Eq. (6) is written as follows: given ϕ_n^h, find $\phi_{n+1}^h \in (\mathcal{S}_\phi^h)_{n+1}$, such that $\forall\, w_{n+1}^h \in (\mathcal{V}_\phi^h)_{n+1}$:

$$\int_\Omega w_{n+1}^h \left(\frac{\partial \phi^h}{\partial t} + \mathbf{u}^h \cdot \nabla \phi^h\right) \mathrm{d}\Omega + \int_\Omega \nabla w_{n+1}^h \cdot \nu \nabla \phi^h \mathrm{d}\Omega - \int_{\Gamma_\mathrm{h}} w_{n+1}^h \mathrm{h}^h \mathrm{d}\Gamma + \sum_{e=1}^{n_{\mathrm{el}}} \int_{\Omega^e} \tau_{\mathrm{SUPG}} \mathbf{u}^h \cdot \nabla w_{n+1}^h \left(\frac{\partial \phi^h}{\partial t} + \mathbf{u}^h \cdot \nabla \phi^h - \nabla \cdot \left(\nu \nabla \phi^h\right)\right) \mathrm{d}\Omega + S_{\mathrm{DC}} = 0, \tag{13}$$

where

$$S_{\mathrm{DC}} = \sum_{e=1}^{n_{\mathrm{el}}} \int_{\Omega^e} \nabla w_{n+1}^h \cdot \nu_{\mathrm{DC}} \nabla \phi^h \mathrm{d}\Omega. \tag{14}$$

The trial and test functions, at a time level n, are defined as

$$\phi_n^h = \phi_n^1 + \phi_n^2, \tag{15}$$

$$w_n^h = w_n^1 + w_n^2, \tag{16}$$

where superscripts 1 and 2 denote the components of the functions coming from Mesh-1 and Mesh-2, respectively.

4.2 Navier–Stokes equations of incompressible flows

At a time level n, the trial function spaces corresponding to the velocity and pressure are denoted by $(\mathcal{S}_{\mathbf{u}}^h)_n$ and $(\mathcal{S}_p^h)_n$. The test function spaces corresponding to the momentum equation and incompressibility constraint are denoted by $(\mathcal{V}_{\mathbf{u}}^h)_n$ and $(\mathcal{V}_p^h)_n$ $(= (\mathcal{S}_p^h)_n)$. The enhanced-discretization stabilized formulation of Eqs. (1) and (2) is written as follows: given $\mathbf{u}_n^h$, find $\mathbf{u}_{n+1}^h \in (\mathcal{S}_{\mathbf{u}}^h)_{n+1}$ and $p_{n+1}^h \in (\mathcal{S}_p^h)_{n+1}$, such that $\forall\, \mathbf{w}_{n+1}^h \in (\mathcal{V}_{\mathbf{u}}^h)_{n+1}$ and $\forall\, q_{n+1}^h \in (\mathcal{V}_p^h)_{n+1}$:

$$\begin{aligned}
&\int_\Omega \mathbf{w}_{n+1}^h \cdot \rho \left(\frac{\partial \mathbf{u}^h}{\partial t} + \mathbf{u}^h \cdot \boldsymbol{\nabla} \mathbf{u}^h - \mathbf{f}^h \right) \mathrm{d}\Omega \\
&+ \int_\Omega \boldsymbol{\varepsilon}(\mathbf{w}_{n+1}^h) : \boldsymbol{\sigma}(p^h, \mathbf{u}^h) \mathrm{d}\Omega \\
&- \int_{\Gamma_\mathrm{h}} \mathbf{w}_{n+1}^h \cdot \mathbf{h}^h \mathrm{d}\Gamma + \int_\Omega q_{n+1}^h \boldsymbol{\nabla} \cdot \mathbf{u}^h \mathrm{d}\Omega \\
&+ \sum_{e=1}^{n_\mathrm{el}} \int_{\Omega^e} \frac{1}{\rho} \left[\tau_{\mathrm{SUPG}} \rho \mathbf{u}^h \cdot \nabla \mathbf{w}_{n+1}^h + \tau_{\mathrm{PSPG}} \nabla q_{n+1}^h \right] \cdot \\
&\quad \left[\text{Ł}(p^h, \mathbf{u}^h) - \rho \mathbf{f}^h \right] \mathrm{d}\Omega + S_{\mathrm{DC}} = 0,
\end{aligned} \tag{17}$$

where

$$S_{\mathrm{DC}} = \sum_{e=1}^{n_\mathrm{el}} \int_{\Omega^e} \rho \boldsymbol{\nabla} \mathbf{w}_{n+1}^h : \left(\boldsymbol{\kappa}_{\mathrm{DC}} \cdot \boldsymbol{\nabla} \mathbf{u}^h \right) \mathrm{d}\Omega. \tag{18}$$

The trial and test functions, at a time level n, are defined as

$$\mathbf{u}_n^h = \mathbf{u}_n^1 + \mathbf{u}_n^2, \tag{19}$$

$$p_n^h = p_n^1 + p_n^2, \tag{20}$$

$$\mathbf{w}_n^h = \mathbf{w}_n^1 + \mathbf{w}_n^2, \tag{21}$$

$$q_n^h = q_n^1 + q_n^2. \tag{22}$$

5 Construction of the function spaces

In constructing the function spaces corresponding to a time level n, we start with a base mesh (Mesh-1), with the set of elements and nodal points denoted by ϵ_n^1 and η_n^1. The subscript n implies that Mesh-1 itself might change from one time level to another.

A second-level and more refined mesh (Mesh-2) is constructed over a subset $(\epsilon_n^1)_n^2$ of these elements. Mesh-2 is generated by patching together the second-level meshes generated over each of the elements in $(\epsilon_n^1)_n^2$. The second subscript n implies that for a given Mesh-1, which elements of this mesh are declared to be in $(\epsilon_n^1)_n^2$ might change from one time level to other. An element which might be declared to be in $(\epsilon_n^1)_n^2$ at some time level, might fall out of it at some other time, and yet come back in again some time later. For each element in ϵ_n^1, there will be a unique second-level mesh. Therefore, if an element is declared to be in $(\epsilon_n^1)_n^2$ for a second time, the refined mesh generated over that element at the earlier declaration can be reused. If an automatic mesh generator is being used to generate these second-level meshes, the cost for that mesh generation will be a one-time cost. The set of elements and nodal points for Mesh-2 are denoted by ϵ_n^2 and η_n^2.

The function ϕ_n^1 comes from a space of functions with the basis set consisting of the shape functions associated with all the nodes in η_n^1. The function ϕ_n^2 comes from a space of functions with the basis set consisting of the shape functions associated with all the nodes in η_n^2, excluding those coinciding with the nodes in η_n^1, and also excluding those at the boundaries of the zones covered by the elements in ϵ_n^2 unless those boundaries coincide with the boundaries of Ω. The sum of the two trial functions, $\phi_n^1 + \phi_n^2$, needs to satisfy the essential boundary conditions. We construct $\mathbf{u}_n^1$, $\mathbf{u}_n^2$, p_n^1, and p_n^2 in exactly the same way, except for recognizing that for p_n^1 and p_n^2 the references to essential boundary conditions do not apply.

The components of each test function are defined in the same way as we did for the trial functions, except that the test functions need to satisfy the homogeneous form of the essential boundary conditions.

We do not update $(\epsilon_n^1)_n^2$ every time step. We update it frequently enough to meet our objective of having enhanced discretization at the zones specified by some criteria. How long we can compute without re-defining this subset will depend on how much larger we decide to keep it compared to the level dictated by the criteria used. The more we exceed the level dictated, the longer we can compute before we need to re-define it again. Whenever we re-define this subset, the mesh generation cost will not be a significant one. If we are using an automatic mesh generator for the second mesh, we will be able to use and reuse the meshes which were generated (and stored) the first time these meshes were needed.

6 Selective stabilization

6.1 Advection–diffusion equation

In describing the versions of the EDSSP we propose in this paper, we first write ϕ^h as a sum of its large- and small-scale components:

$$\phi^h = \phi^1 + \phi^2. \tag{23}$$

This decomposition is essentially equivalent to what we did in Eq. (15), but without reference to a specific time level.

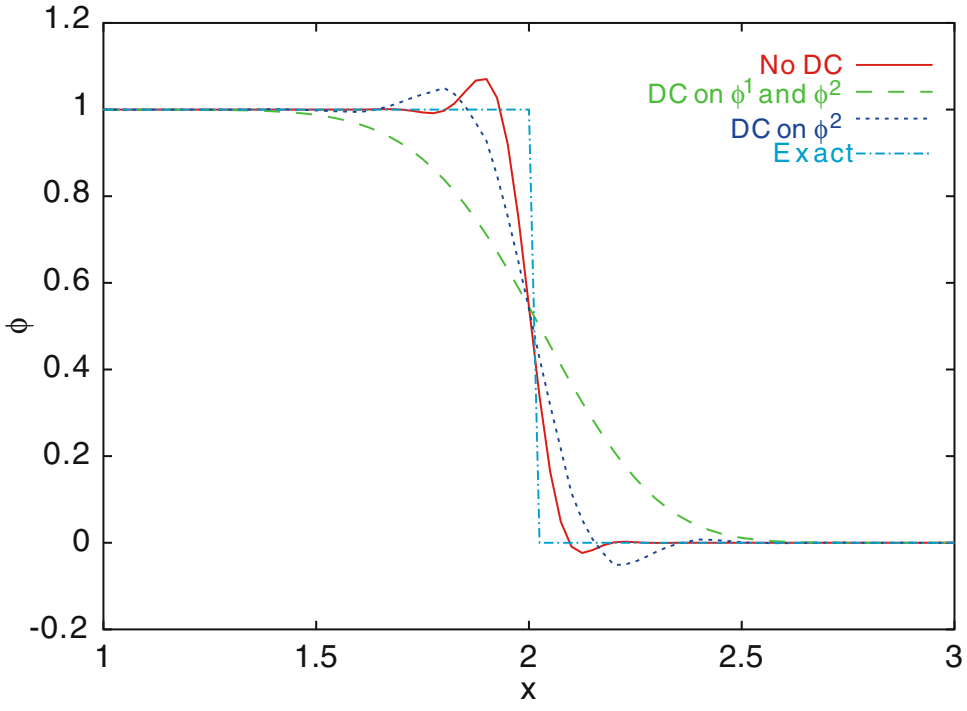

Fig. 1 1D time-dependent advection of a discontinuity. Solutions obtained with the linear DC, compared with the solution obtained with no DC

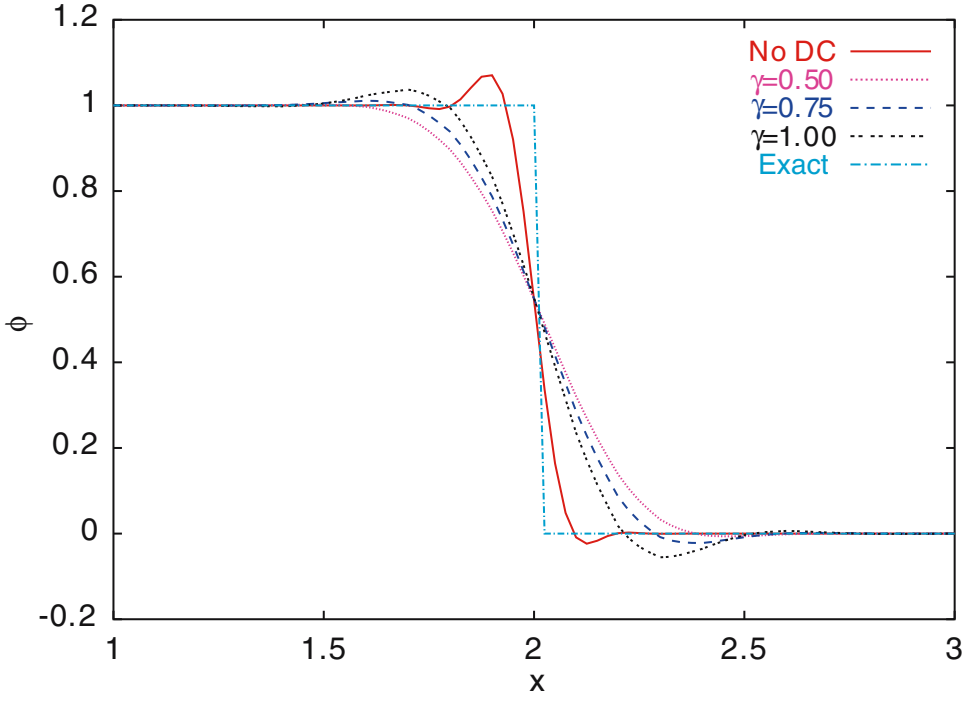

Fig. 2 1D time-dependent advection of a discontinuity. Solutions obtained with the linear DC given by Eq. (30), compared with the solution obtained with no DC

In the first version of the EDSSP we propose, the DC is applied to only the small-scale unknowns:

$$S_{\text{DC}} = \sum_{e=1}^{n_{\text{el}}} \int_{\Omega^e} \nabla w_{n+1}^h \cdot \nu_{\text{DC}} \nabla \phi^2 \, \text{d}\Omega, \tag{24}$$

where ϕ^2 is the small-scale component of ϕ^h. This has two options, one for linear DC and one for nonlinear:

$$\nu_{\text{LDC}} = \left\| \mathbf{u}^h \right\| \left(\frac{h}{2} \right), \tag{25}$$

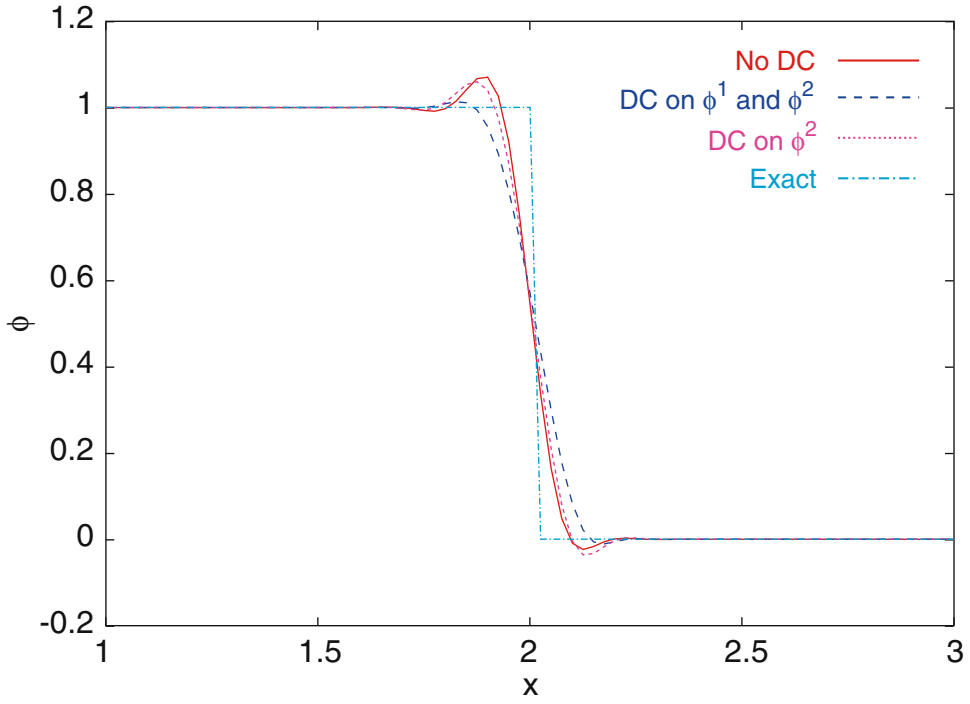

Fig. 3 1D time-dependent advection of a discontinuity. Solutions obtained with the nonlinear DC, compared with the solution obtained with no DC

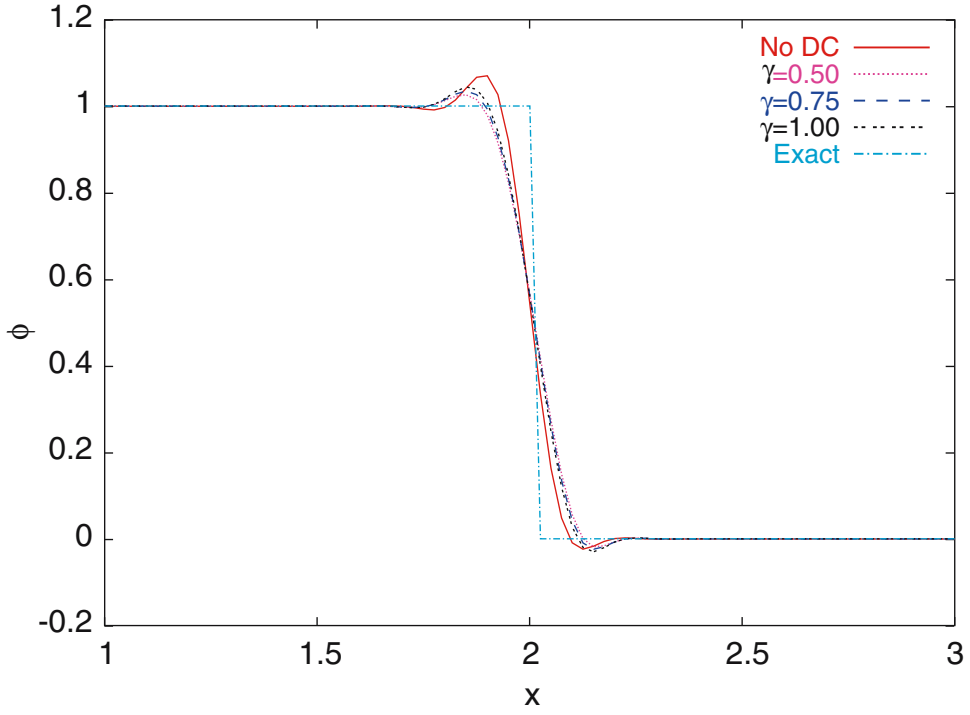

Fig. 4 1D time-dependent advection of a discontinuity. Solutions obtained with the nolinear DC given by Eq. (30), compared with the solution obtained with no DC

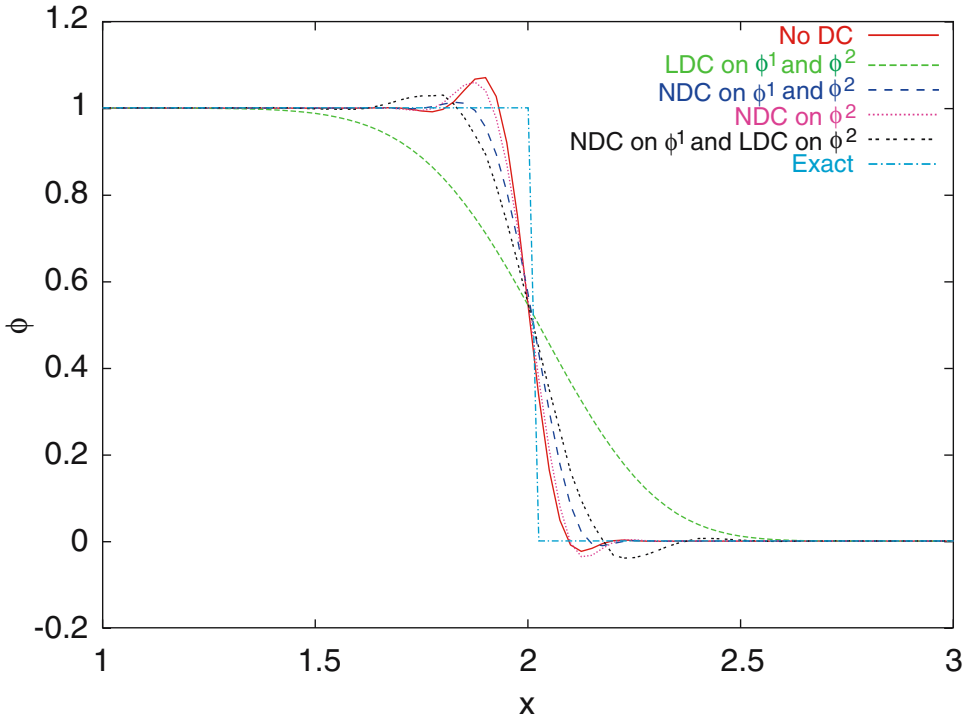

Fig. 5 1D time-dependent advection of a discontinuity. Solutions obtained with various combinations of the linear and nonlinear DC, compared with the solution obtained with no DC

$$\nu_{\text{NDC}} = \left| Y^{-1} \left(\mathbf{u}^h \cdot \nabla \phi \right) \right| \left(\frac{h}{2} \right)^2, \tag{26}$$

where h is a measure of the element length and Y is a scaling value for ϕ.

In the second version, we re-define what we consider for the purpose of selective stabilization to be the large- and small-scale components of ϕ^h. We re-write ϕ^h as follows:

$$\phi^h = \left(\overline{\phi^h} \right)^1 + \phi^1 - \left(\overline{\phi^h} \right)^1 + \phi^2, \tag{27}$$

where $\left(\overline{\phi^h} \right)^1$ is the averaged value of ϕ^h represented in the large-scale function space. In calculating $\left(\overline{\phi^h} \right)^1$, for each large-scale node the averaging is done over the large-scale elements surrounding that node. For the purpose of selective stabilization, the large- and small-scale components of ϕ^h are considered to be $\left(\overline{\phi^h} \right)^1$ and $\phi^1 - \left(\overline{\phi^h} \right)^1 + \phi^2$. The DC is then applied to the re-defined small-scale unknowns:

$$S_{\text{DC}} = \sum_{e=1}^{n_{\text{el}}} \int_{\Omega^e} \nabla w_{n+1}^h \cdot \nu_{\text{DC}} \nabla \left(\phi^1 - \left(\overline{\phi^h} \right)^1 + \phi^2 \right) \mathrm{d}\Omega. \tag{28}$$

This version is generalized by introducing a parameter γ into the decomposition given by Eq. (27):

$$\phi^h = \gamma \left(\overline{\phi^h} \right)^1 + \phi^1 - \gamma \left(\overline{\phi^h} \right)^1 + \phi^2, \tag{29}$$

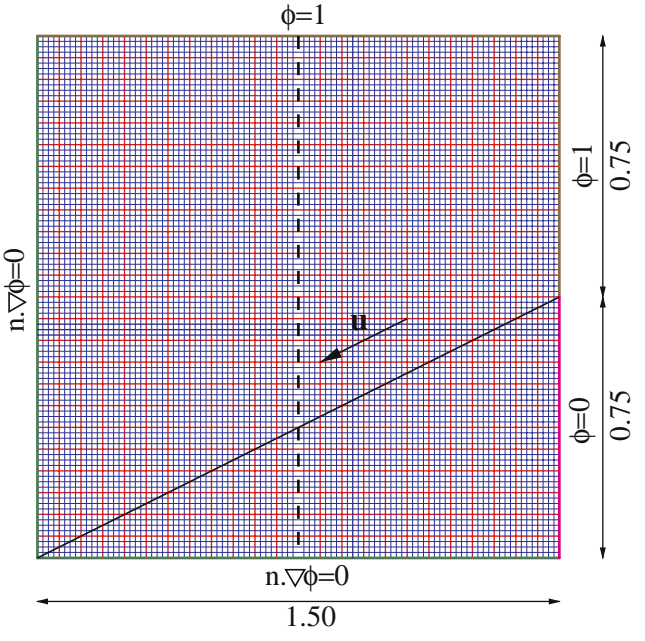

Fig. 6 2D steady-state advection with a discontinuity skew to the mesh. Problem set up. Both the large- and small-scale meshes are uniform, with 24×24 and 96×96 quadrilateral elements, respectively. The exact solution is a discontinuity along the *solid black line*. Numerical solutions are compared by plotting them along the *dashed black line* at $x = 0.75$

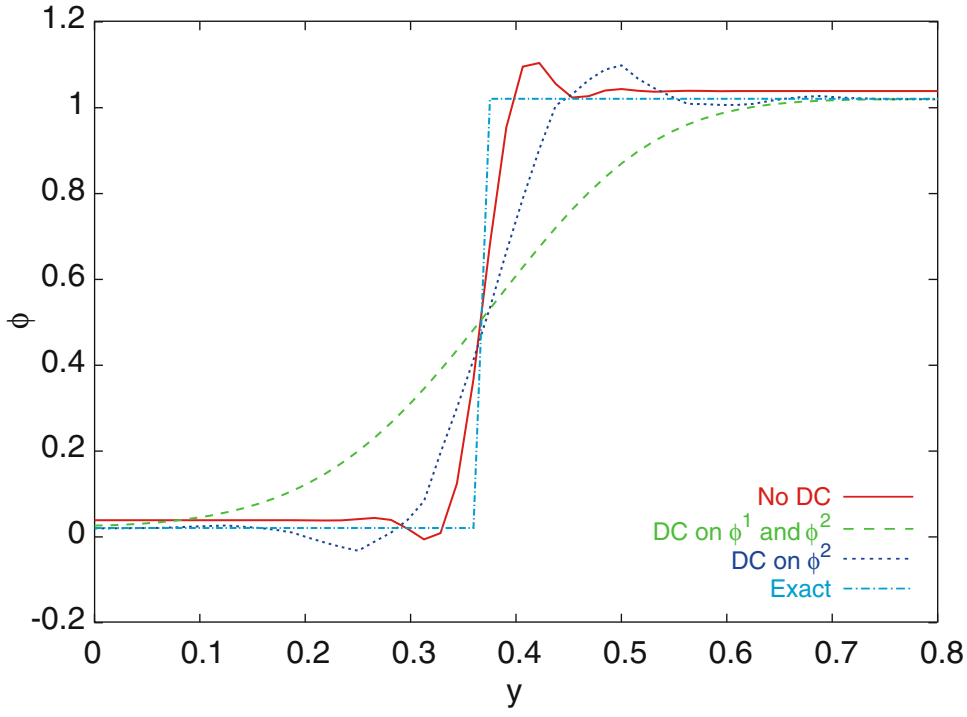

Fig. 7 2D steady-state advection with a discontinuity skew to the mesh. Solutions obtained with the linear DC, compared with the solution obtained with no DC

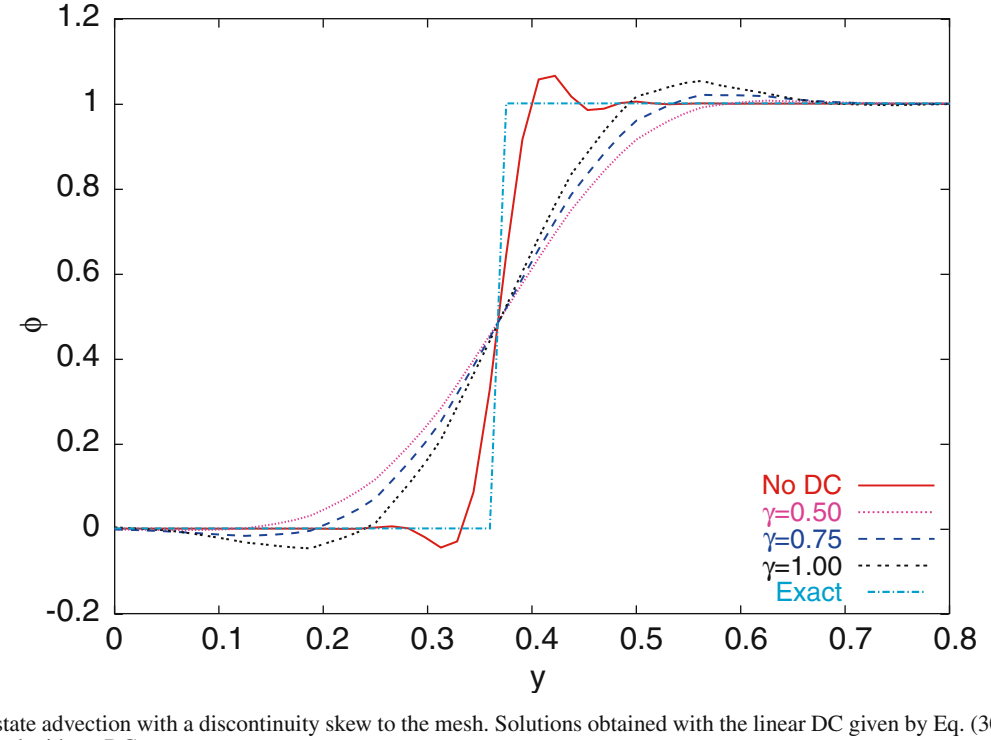

Fig. 8 2D steady-state advection with a discontinuity skew to the mesh. Solutions obtained with the linear DC given by Eq. (30), compared with the solution obtained with no DC

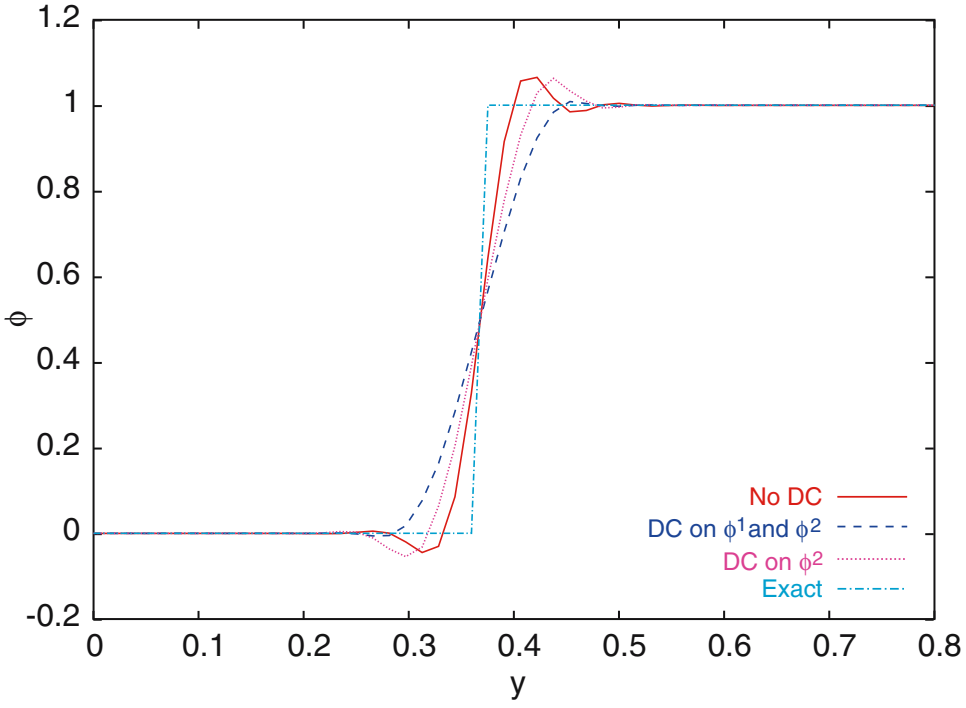

Fig. 9 2D steady-state advection with a discontinuity skew to the mesh. Solutions obtained with the nonlinear DC, compared with the solution obtained with no DC

and the DC is applied to the re-defined small-scale unknowns based on this generalization:

$$S_{\text{DC}} = \sum_{e=1}^{n_{\text{el}}} \int_{\Omega^e} \nabla w_{n+1}^h \cdot \nu_{\text{DC}} \nabla \left(\phi^1 - \gamma \left(\overline{\phi^h} \right)^1 + \phi^2 \right) \mathrm{d}\Omega. \tag{30}$$

We note that when γ=1, the decomposition given by Eq. (29) reduces to the one given by Eq. (27), and when $\gamma = 0$, the DC is applied to both the large- and small-scale unknowns. The second version of the EDSSP proposed here also has two options, one for linear DC and one for nonlinear, as given by Eqs. (25) and (26).

In the third version of the EDSSP proposed, linear DC is applied to the small-scale unknowns and nonlinear DC to the large-scale unknowns:

$$S_{\text{DC}} = \sum_{e=1}^{n_{\text{el}}} \int_{\Omega^e} \nabla w_{n+1}^h \cdot \left(\nu_{\text{NDC}} \nabla \phi^1 + \nu_{\text{LDC}} \nabla \phi^2 \right) \mathrm{d}\Omega. \tag{31}$$

6.2 Navier–Stokes equations of incompressible flows

The three versions of the EDSSP proposed for the Navier–Stokes equations of incompressible flows are defined by following a track similar to the one followed for the advection–diffusion equation. We first write $\mathbf{u}^h$ as a sum of its large- and small-scale components:

$$\mathbf{u}^h = \mathbf{u}^1 + \mathbf{u}^2. \tag{32}$$

In the first version of the EDSSP proposed, the DC is applied to only the small-scale unknowns:

$$S_{\text{DC}} = \sum_{e=1}^{n_{\text{el}}} \int_{\Omega^e} \rho \nabla \mathbf{w}_{n+1}^h : \left(\boldsymbol{\kappa}_{\text{DC}} \cdot \nabla \mathbf{u}^2 \right) \mathrm{d}\Omega. \tag{33}$$

In this version, one of the simpler ways of defining the linear DC option is to set $\boldsymbol{\kappa}_{\text{LDC}} = \nu_{\text{LDC}} \mathbf{I}$ and use for ν_{LDC} the expression given by Eq. (25) for the advection–diffusion equation. For ways of calculating $\boldsymbol{\kappa}_{\text{NDC}}$, we refer the interested reader to [19, 34, 33].

In the second version, $\mathbf{u}^h$ is re-written as

$$\mathbf{u}^h = \gamma \left(\overline{\mathbf{u}^h} \right)^1 + \mathbf{u}^1 - \gamma \left(\overline{\mathbf{u}^h} \right)^1 + \mathbf{u}^2, \tag{34}$$

and the DC is applied to the re-defined small-scale unknowns:

$$S_{\text{DC}} = \sum_{e=1}^{n_{\text{el}}} \int_{\Omega^e} \rho \nabla \mathbf{w}_{n+1}^h : \left(\boldsymbol{\kappa}_{\text{DC}} \cdot \nabla \left(\mathbf{u}^1 - \gamma \left(\overline{\mathbf{u}^h} \right)^1 + \mathbf{u}^2 \right) \right) \mathrm{d}\Omega. \tag{35}$$

In the third version, linear DC is applied to the small-scale unknowns and nonlinear DC to the large-scale unknowns:

$$S_{\text{DC}} = \sum_{e=1}^{n_{\text{el}}} \int_{\Omega^e} \rho \nabla \mathbf{w}_{n+1}^h : \left(\boldsymbol{\kappa}_{\text{NDC}} \cdot \nabla \mathbf{u}^1 + \boldsymbol{\kappa}_{\text{LDC}} \cdot \nabla \mathbf{u}^2 \right) \mathrm{d}\Omega. \tag{36}$$

7 Test computations

In this section, we evaluate the numerical performance of the proposed versions of the EDSSP in test computations with

advection problems. As test cases, we consider a 1D time-dependent problem and a 2D steady-state problem. In both cases the magnitude of the advection velocity is 1.0.

7.1 One-dimensional time-dependent advection of a discontinuity

The boundary conditions are specified as $\phi = 1$ at $x = 0$ and $\phi = 0$ at $x = 3$. The initial condition is set as $\phi = 1$ at $x = 0$ and $\phi = 0$ elsewhere. Both the large- and small-scale meshes are uniform, with 30 and 120 elements, respectively. The element length used in calculation of the SUPG and DC parameters is the one corresponding to the small-scale mesh (i.e., $h = 0.025$). The computations were carried out until $t = 2.0$, with the central-difference time-integration and a time-step size of $\Delta t = 0.0125$. At the end of this time period, in the exact solution the discontinuity is located at $x = 2.0$.

Figure 1 shows the results obtained with the linear DC.

Applying the linear DC to both ϕ^1 and ϕ^2 leads to excessive smoothing of the discontinuity and loss of accuracy. Applying the DC to only ϕ^2 yields a smoother discontinuity compared to having no DC, but we do not see an excessive smoothing. The solution is closer to the solution obtained with no DC than it is to the solution obtained with applying the DC to both scales. Figure 2 shows the results obtained with the linear DC given by Eq. (30). Because some part of ϕ^1 is considered as small-scale and subjected to the DC together with ϕ^2, the solutions are smoother than the one obtained with applying the DC to only ϕ^2. As the parameter γ decreases, more of ϕ^1 is subjected to the DC, and, in the limit, when $\gamma = 0$ the solution becomes identical to the one obtained with applying the DC to both scales. Figure 3 shows the results obtained with the nonlinear DC. The relative trends are similar to those seen in Fig. 1 for the linear DC, but the solutions are much closer to each other.

Figure 4 shows the results obtained with the nonlinear DC given by Eq. (30). The relative trends are similar to those seen in Fig. 2 for the linear DC, but the solutions are much closer to each other. Figure 5 shows the results obtained with various combinations of the linear and nonlinear DC. Using linear DC on ϕ^2 and nonlinear DC on ϕ^1 yields a result that is smoother than the solution obtained with using nonlinear DC on both scales, but we do not see the excessive smoothing we see when both scales are subjected to the linear DC.

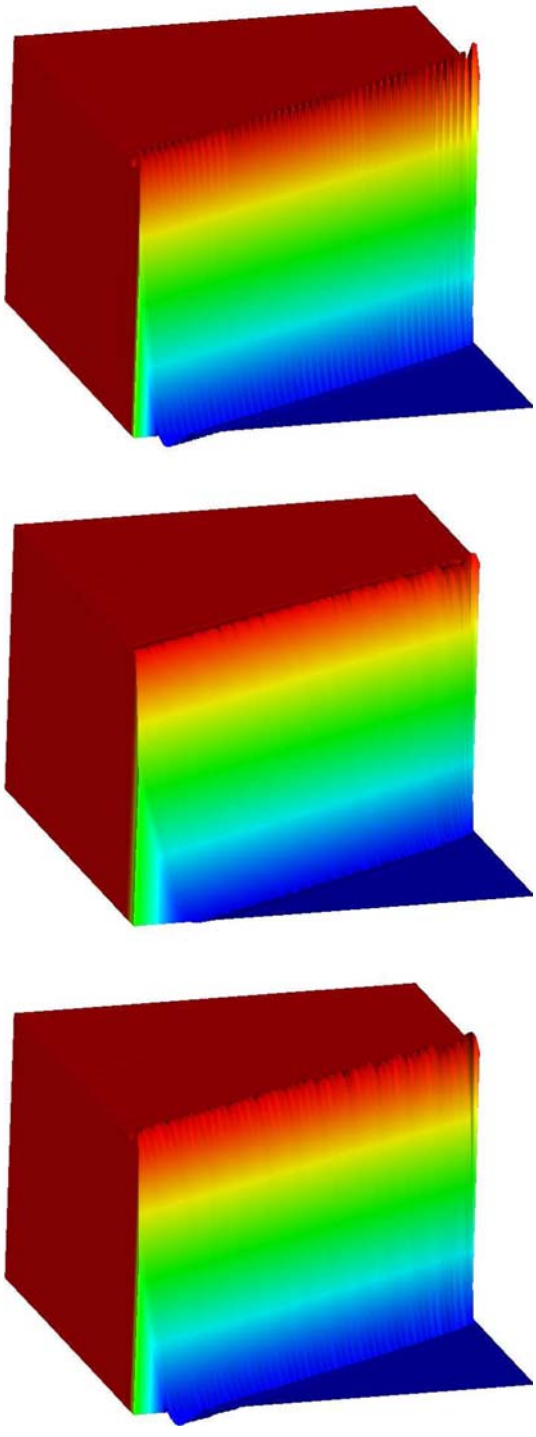

Fig. 10 2D steady-state advection with a discontinuity skew to the mesh. Solutions obtained with no DC *top*, nonlinear DC applied to both ϕ^1 and ϕ^2 *middle*, and nonlinear DC applied to only ϕ^2 *bottom*

7.2 Two-dimensional steady-state advection with a discontinuity skew to the mesh

The problem set up, including the boundary conditions, is shown in Fig. 6. Both the large- and small-scale meshes are uniform, with 24×24 and 96×96 quadrilateral elements, respectively. The element length used in calculation of the SUPG and DC parameters is based on the small-scale mesh and an element length definition named h_{UGN} (see [15, 19, 32, 33, 35]). For the constant advection velocity given in this problem, and averaging h_{UGN} over the quadrature points of each small-scale element, $h = 0.0161185$. The exact solution is a discontinuity along the solid black line shown in Figure 6. Numerical solutions are compared by plotting them along the line $x = 0.75$.

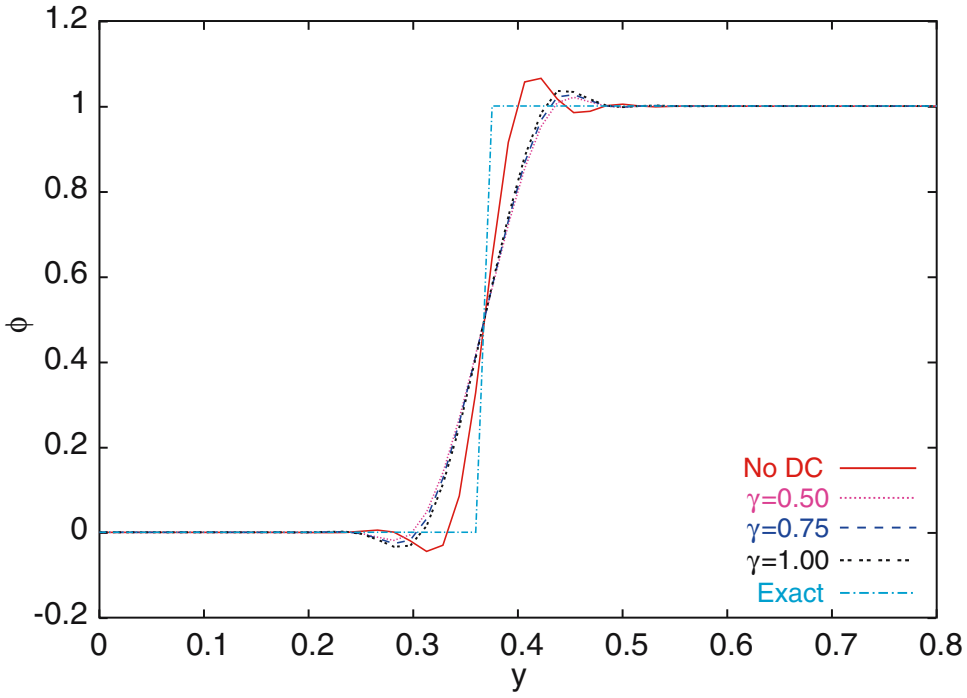

Fig. 11 2D steady-state advection with a discontinuity skew to the mesh. Solutions obtained with the nolinear DC given by Eq. (30), compared with the solution obtained with no DC

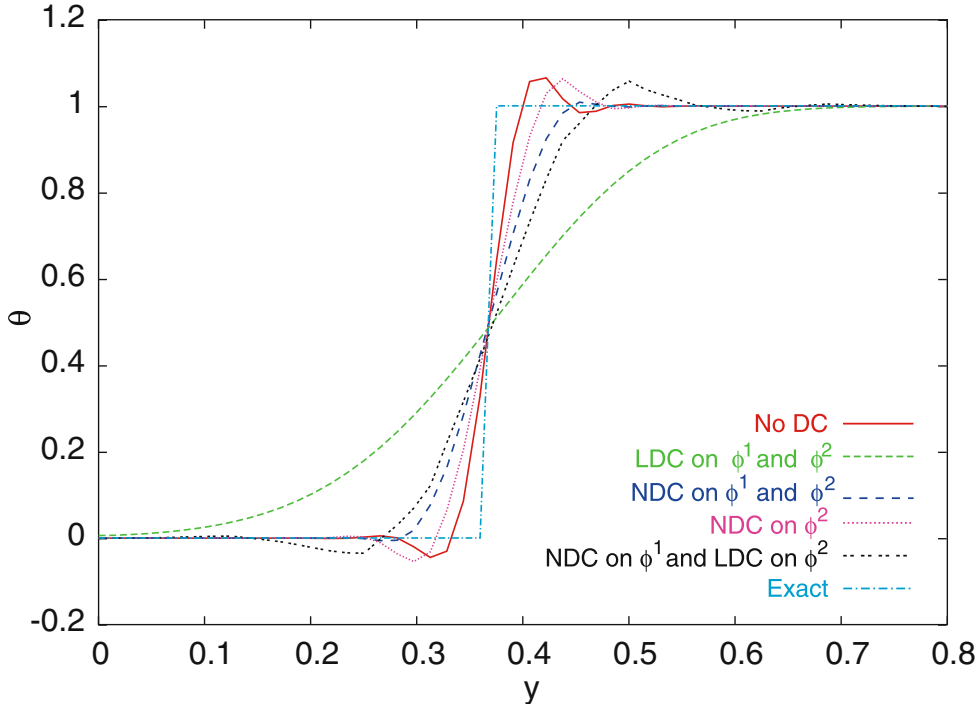

Fig. 12 2D steady-state advection with a discontinuity skew to the mesh. Solutions obtained with various combinations of the linear and nonlinear DC, compared with the solution obtained with no DC

Figure 7 shows the results obtained with the linear DC Similar to what we observed in the 1D test case, applying the linear DC to both ϕ^1 and ϕ^2 leads to excessive smoothing of the discontinuity and loss of accuracy. Applying the DC to only ϕ^2 yields a smoother discontinuity compared to having no DC, but without excessive smoothing. The solution is closer to the solution obtained with no DC than it is to the solution obtained with applying the DC to both scales.

Figure 8 shows the results obtained with the linear DC given by Eq. (30). Also similar to what we observed in the 1D test case, because some part of ϕ^1 is considered as small-scale and subjected to the DC together with ϕ^2, the solutions are smoother than the one obtained with applying the DC to only ϕ^2. Again, as the parameter γ decreases, more of ϕ^1 is subjected to the DC, and, in the limit, when $\gamma = 0$ the solution becomes identical to the one obtained with applying the DC to both scales. Figure 9 shows the results obtained with the nonlinear DC. The relative trends are similar to those seen in Figure 7 for the linear DC, but the solutions are much closer to each other. This is consistent with our observations in the 1D test case. Figure 10 shows the elevation plots for the numerical results shown in Fig. 9. Figure 11 shows the results obtained with the nonlinear DC given by Eq. (30). The relative trends are similar to those seen in Fig. 8 for the linear DC, but the solutions are much closer to each other. This is also consistent with our observations in the 1D test case. Figure 12 shows the results obtained with various combinations of the linear and nonlinear DC. Again, our observations from this Figure are consistent with those from the 1D test case. Using linear DC on ϕ^2 and nonlinear DC on ϕ^1 yields a result that is smoother than the solution obtained with using nonlinear DC on both scales, but we do not see the excessive smoothing we see when both scales are subjected to the linear DC.

8 Concluding remarks

We described the EDSSP, which provides a multiscale framework for applying numerical stabilization selectively at different scales. The EDSSP is based on the enhanced-discretization, multiscale function space concept underlying the EDSUM, which is a multi-level iteration method designed for computation of the flow behavior at small scales. The EDSUM has a built-in mechanism for transferring flow information between the large and small scales without assuming that the small-scale trial or test functions vanish at the borders between the neighboring large-scale elements of the enhanced-discretization zones. This facilitates unrestricted movement of small-scale flow information from one large-scale element to another without any constraints at the border between the two elements. The multiscale function space concept underlying the EDSUM can also facilitate using different stabilizations for equations or unknowns corresponding to different scales.

In this paper we proposed a version of the EDSSP where the SUPG and PSPG stabilizations are used for unknowns corresponding to both the large and small scales but the DC stabilizations are used for unknowns corresponding to only the small scales. We also proposed a version where a linear DC is used for the small-scale unknowns and a nonlinear DC is used for the large-scale unknowns. We evaluated the performances of these versions of the EDSSP with advection–diffusion test problems involving discontinuous solutions. The results from these test computations confirm that applying linear DC to only the small-scale unknowns yields smoother solutions compared to having no DC, but we do not see the excessive smoothing we see when both scales are subjected to the linear DC. The trends are similar with the nonlinear DC, but the differences are not so large. The results also confirm that using linear DC on the small-scale unknowns and nonlinear DC on large-scale unknowns yields solutions that are smoother than the solutions obtained with using nonlinear DC on both scales, but we do not see the excessive smoothing we see when both scales are subjected to the linear DC.

Acknowledgements This work was supported by the US Army Natick Soldier Center and NASA Johnson Space Center.

References

1. Tezduyar TE (2005) Finite elements in fluids: stabilized formulations and moving boundaries and interfaces. Comput Fluids, (published online)
2. Tezduyar TE (2006) Finite elements in fluids: special methods and enhanced solution techniques. Comput Fluids (published online)
3. Hughes TJR, Brooks AN (1979) A multi-dimensional upwind scheme with no crosswind diffusion. In: Hughes TJR (ed) Finite element methods for convection dominated flows, AMD-vol.34. ASME, New York pp 19–35
4. Brooks AN, Hughes TJR (1982) Streamline upwind/Petrov–Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier–Stokes equations. Comput Methods Appl Mech Eng 32:199–259
5. Tezduyar TE, Hughes TJR (1983) Finite element formulations for convection dominated flows with particular emphasis on the compressible Euler equations. In: Proceedings of AIAA 21st aerospace sciences meeting, AIAA Paper 83-0125, Reno, Nevada
6. Hughes TJR, Tezduyar TE (1984) Finite element methods for first-order hyperbolic systems with particular emphasis on the compressible Euler equations. Comput Methods Appl Mech Eng, 45: 217–284
7. Tezduyar TE (1992) Stabilized finite element formulations for incompressible flow computations. Adv Appl Mech 28: 1–44
8. Tezduyar TE, Mittal S, Ray SE, Shih R (1992) Incompressible flow computations with stabilized bilinear and linear equal-order-interpolation velocity-pressure elements Comput Methods Appl Mech Eng, 95: 221–242
9. Tezduyar TE, Behr M, Liou J (1992) A new strategy for finite element computations involving moving boundaries and interfaces – the deforming-spatial-domain/space–time procedure: I. The concept and the preliminary numerical tests. Comput Methods Appl Mech Eng 94: 339–351
10. Tezduyar TE, Behr M, Mittal S, Liou J (1992) A new strategy for finite element computations involving moving boundaries and interfaces – the deforming-spatial-domain/space–time procedure: II. Computation of free-surface flows, two-liquid flows, and flows with drifting cylinders. Comput Methods Appl Mech Eng 94: 353–371
11. Tezduyar TE (2001) Finite element methods for flow problems with moving boundaries and interfaces. Arch Comput Methods Eng 8: 83–130.
12. Tezduyar T (2002) Interface-tracking and interface-capturing techniques for computation of moving boundaries and interfaces. In: Proceedings of the 5 World Congress on computational mechanics. On-line publication: http://wccm.tuwien.ac.at/, Paper-ID: 81513, Vienna
13. Hughes TJR, Franca LP, Balestra M (1986) A new finite element formulation for computational fluid dynamics: V. Circumventing the Babuška–Brezzi condition: a stable Petrov–Galerkin formulation of the Stokes problem accommodating equal-order interpolations Comput Methods Appl Mech Eng 59: 85–99

14. Hughes TJR, Mallet M, Mizukami A (1986) A new finite element formulation for computational fluid dynamics: II. Beyond SUPG: Comput Methods Appl Mech Eng, 54: 341–355
15. Tezduyar TE, and Park YJ (1986) "Discontinuity capturing finite element formulations for nonlinear convection-diffusion-reaction equations. Comput Methods Appl Mech Eng 59:307–325
16. Hughes TJR, Mallet M (1986) A new finite element formulation for computational fluid dynamics: IV. A discontinuity-capturing operator for multidimensional advective-diffusive systems. Computer Methods Appl Mech Eng, 58: 329–339
17. Tezduyar T, Aliabadi S, Behr M (1998) Enhanced-discretization interface-capturing technique (EDICT) for computation of unsteady flows with interfaces. Comput Methods Appl Mech Eng 155: 235–248
18. Tezduyar T (2001) Finite element interface-tracking and interface-capturing techniques for flows with moving boundaries and interfaces In: Proceedings of the ASME symposium on fluid-physics and heat transfer for macro- and micro-scale gas-liquid and phase-change flows (CD-ROM). ASME Paper IMECE2001/HTD-24206, ASME, New York
19. Tezduyar TE (2004) Finite element methods for fluid dynamics with moving boundaries and interfaces In: Stein E, De Borst R, Hughes TJR, (eds), Encyclopedia of computational mechanics, volume 3: fluids, Chapter 17: Wiley , Newyork
20. Tezduyar TE, Sathe S (2004) Enhanced-discretization space-time technique (EDSTT). Comput Methods Appl Mech and Eng, 193: 1385–1401
21. Tezduyar TE (2003) Stabilized finite element methods for flows with moving boundaries and interfaces. HERMIS Int J Comput Math Appl 4: 63–88
22. Tezduyar TE (2004) Moving boundaries and interfaces. In: Franca LP, Tezduyar TE, Masud A, (eds) Finite element methods: 1970's and beyond 205–220, CIMNE, Barcelona, pp 205–220
23. Tezduyar TE, Sathe S (2005) Enhanced-discretization successive update method (EDSUM) Int J Numer Methods Fluids, 47: 633–654
24. Hughes TJR (1995) Multiscale phenomena: Green's functions, the Dirichlet-to-Neumann formulation, subgrid scale models, bubbles, and the origins of stabilized methods. Comput Methods Appl Mech Eng 127 387–401
25. Hughes TJR, Mazzei L, Oberai AA, Wray A (2001) The multiscale formulation of large eddy simulation: decay of homogeneous isotropic turbulence. Phys Fluids 13:0 505–512
26. Hughes TJR, Oberai AA, Mazzei L (2001) Large eddy simulation of turbulent channel flows by the variational multiscale method. Phys Fluids 13: 1784–1799
27. Brezzi F, Marini LD (2002) Augmented spaces, two-level methods, and stabilising subgrids. Int J Numer Methods Fluids 40:31–46
28. Masud A, Khurram R (2003) A multiscale method for advection–diffusion equation. Comput Methods Appl Mech Eng 192: 1–24
29. Masud A, Bergman L (2004) Application of multiscale finite element methods to the solution of the Fokker–Planck equation. Computer Methods Applied Mech Eng 195: 1–16
30. A. Masud and K. Xia, A variational multiscale method for computational inelasticity: application to superelasticity in shape memory alloys. Comput Methods Appl Mech Eng(in press)
31. Masud A, Khurram RA (2006) A multiscale finite element method for the incompressible Navier–Stokes equations. Comput Methods Appl Mech Eng 195:1750–1777
32. Tezduyar TE, Osawa Y (2000) Finite element stabilization parameters computed from element matrices and vectors. Comput Methods Appl Mech Eng 190: 411–430
33. Tezduyar TE (2003) Computation of moving boundaries and interfaces and stabilization parameters Int J Numer Methods Fluids 43:555–575
34. Tezduyar TE (2001) Adaptive determination of the finite element stabilization parameters. In: Proceedings of the ECCOMAS computational fluid dynamics conference 2001 (CD-ROM), Swansea, Wales
35. Tezduyar T, Sathe S (2003) Stabilization parameters in SUPG and PSPG formulations. J Comput Appl Mech 4: 71–88