UNIVERSITY OF MARYLAND
INSTITUTE FOR ADVANCED COMPUTER STUDIES
DEPARTMENT OF COMPUTER SCIENCE

ITERATIVE METHODS FOR STABILIZED DISCRETE CONVECTION–DIFFUSION PROBLEMS

CS-TR #3945 / UMIACS TR #98-58

YIN-TZER SHIH* AND HOWARD C. ELMAN†

Abstract. In this paper, we study the computational cost of solving the convection-diffusion equation using various discretization strategies and iteration solution algorithms. The choice of discretization influences the properties of the discrete solution and also the choice of solution algorithm. The discretizations considered here are stabilized low order finite element schemes using streamline diffusion, crosswind diffusion and shock–capturing. The latter, shock–capturing discretizations lead to nonlinear algebraic systems and require nonlinear algorithms. We compare various preconditioned Krylov subspace methods including Newton–Krylov methods for nonlinear problems, as well as several preconditioners based on relaxation and incomplete factorization. We find that although enhanced stabilization based on shock–capturing requires fewer degrees of freedom than linear stabilizations to achieve comparable accuracy, the nonlinear algebraic systems are more costly to solve than those derived from a judicious combination of streamline diffusion and crosswind diffusion. Solution algorithms based on GMRES with incomplete block–matrix factorization preconditioning are robust and efficient.

Key words. Convection–diffusion, streamline diffusion, shock–capturing, Krylov subspace, inexact Newton, preconditioning.

AMS(MOS) subject classifications. primary 65N30, 65F10

* Interdisciplinary Applied Mathematics Program, University of Maryland, College Park, MD 20742, email: yts@cs.umd.edu.

† Department of Computer Science and Institute for Advanced Computer Studies, University of Maryland, College Park, MD 20742, e-mail: elman@cs.umd.edu. This work was supported by U. S. National Science Foundation under grant DMS-9423133.

1. Introduction. Consider the two–dimensional convection–diffusion equation

$$-\varepsilon\Delta u + \beta\cdot\nabla u = f \quad \text{in } \Omega, \tag{1}$$

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

where $\beta = (\beta_1, \beta_2)$ is a flow velocity field, ε is a diffusion or viscosity coefficient, and f, g are given functions. Our concern in this paper is the efficient solution of discrete versions of this problem by iterative methods, with emphasis on the effect of discretization strategy on the overall cost of achieving a specified accuracy. We are particularly interested in cases where the solution contains steep gradients, i.e. boundary layers or internal layers.

In such cases, it is known that standard discretization techniques such as Galerkin finite elements yield inaccurate oscillatory solutions [18], [29, p. 259]. Various approaches for handling this problem, based on the addition of a judicious amount of upwinding, have been proposed. They include the streamline diffusion method [20] and variants that contain additional crosswind diffusion [23] and shock–capturing terms [8, 21, 38]. These strategies all in some way attempt to enhance the coercivity of the standard Galerkin discretization and are referred to as stabilized discretizations. These modified discretizations change the properties of the algebraic systems being solved, and therefore in all likehood they will affect the cost of solving these systems. The latter (shock–capturing) techniques are noteworthy in that the added diffusive term depends on the unknown solution, leading to a nonlinear discrete system even though the original problem is linear.

In this paper, we make a comparison of the cost effectiveness of a collection of such discretization strategies, for solving a set of benchmark problems of the form (1)–(2). In identifying cost effectiveness, our aims are twofold:

1. To compare and contrast the different discretization strategies in their capability to compute accurate solutions of benchmark problems;
2. To identify efficient solution algorithms for each discretization.

For solution algorithms, we use preconditioned Krylov subspace methods, including Newton–Krylov variants of these ideas to handle nonlinear algebraic systems. Our results indicate that the nonlinear shock–capturing discretizations yields significantly more accurate solutions than linear stabilization methods. However, the cost of solving the nonlinear systems also tends to be high. Although linear stabilizations require finer grids than nonlinear ones to achieve comparable accuracy, the overall solution costs of using linear discretizations (which include components of both streamline and crosswind diffusion) are lower.

The contents of the rest of the paper are as follows. In Section 2, we describe the linear stabilized finite element discretizations of (1)–(2) that we consider, and in Section 3, we describe the nonlinear discretizations. In Section 4, we briefly describe the Krylov subspace methods that we use to solve the discrete problems, and in Section 5, we describe some preconditioners used to speed convergence. In Section 6, we examine the results of numerical experiments on the benchmark problems.

2. Linear stabilized discretizations. In this section we describe the three linear stabilized discretizations of the problem (1)–(2) that we consider. For simplicity, we assume homogeneous Dirichlet boundary conditions on all boundaries; the ideas considered here generalize in a straightforward manner to other boundary conditions. Let $(\cdot,\cdot)$ denote the usual scalar L^2 inner product. The weak formulation of (1)–(2) is then: find $u \in H_0^1(\Omega)$ such that

$$B_g(u, v) = (f, v) \quad \text{for all } v \in H_0^1(\Omega),$$

where

$$B_g(u,v) = \varepsilon(\nabla u, \nabla v) + (u_\beta, v)$$

and $v_\beta = \beta\cdot\nabla v$ denotes the derivative in the streamline direction. Let $\mathcal{T}_h$ denote a triangulation of Ω and $\mathcal{T}_h = \{\tau_h\}$. We will restrict our attention to low order finite element spaces on $\mathcal{T}_h$. In particular, let

$$V_h^0 = \{v \mid v \in P_1(\tau_h), \forall \tau_h \in \mathcal{T}_h,\ \ v \text{ is continuous at the nodes and } v = 0 \text{ on } \partial\Omega\},$$

where $P_1(\tau_h)$ is the space of either linear or bilinear functions defined on τ_h.

2.1. Streamline diffusion method. The *streamline diffusion method* (SD) is defined [22, p. 185] as: find $u^h \in V_h^0$ such that

$$B_{sd}(u^h, v) = F_v \quad \forall v \in V_h^0, \tag{3}$$

where $B_{sd}(\cdot,\cdot)$ is the bilinear form

$$B_{sd}(u^h, v) = B_g(u^h, v) + \delta_s(u_\beta^h, v_\beta) \tag{4}$$

$$F_v = (f, v) + \delta_s(f, v_\beta), \tag{5}$$

On a uniform grid with mesh parameter h for which the mesh Péclet number $P_e = \frac{|\beta|h}{2\varepsilon}$ is greater than 1, the parameter δ_s is given by

$$\delta_s = \omega_s h \tag{6}$$

where ω_s is a fixed positive constant.[1] In practice, when using SD to solve problems with characteristic internal and boundary layers, the choice of ω_s is important. Fischer, Ramage, Silvester and Wathen [16] show that if ϱ is the angle of flow to the horizontal, the choice

$$\omega_s = \frac{1}{|\beta|}\left(\frac{1}{2} - \frac{\varepsilon}{h}|\cos\varrho|\right) \tag{7}$$

is a good one with respect to both clustering of the spectrum of the discrete operator and performance of the GMRES iterative solution algorithm.

Under the assumption $-\nabla\cdot\beta \geq d_0$ for nonnegative constant d_0, consider the energy norm

$$|||v|||_{sd}^2 = \varepsilon\|\nabla v\|_{L^2(\Omega)}^2 + \delta_s\|v_\beta\|_{L^2(\Omega)}^2 + \frac{d_0}{2}\|v\|_{L^2(\Omega)} \qquad \forall v \in V_h. \tag{8}$$

The form B_{sd} satisfies the coercivity condition

$$B_{sd}(v,v) \geq |||v|||_{sd}^2 \tag{9}$$

in which the lower bound is positive in the limit $\varepsilon \searrow 0$. In particular, the finite element discretization matrix has positive definite symmetric part and the discrete solution u^h of (3) is unique. If $f \in L^2(\Omega)$ and u is the strong solution, then Axelsson [1] and Nävert [28] have

[1] If $\mathcal{T}_h$ is not uniform or β is a variable, then let h represent the diameter of a local element, and determine δ_s elementwise as in (6) (see [22, p. 186]).

shown that for a piecewise linear finite element space there is a constant C (independent of h, δ_s and ε) such that

$$|||u - u^h|||_{sd} \le C\left(\varepsilon^{1/2}h + \delta_s^{1/2}h\right)|u|_2,$$

where $\|\cdot\|$ and $|\cdot|_2$ denote the usual L^2 norm and H^2 seminorm respectively. It is also shown in [3] that if $\delta_s = O(h)$, $\varepsilon \le ch$ and $-\nabla\cdot\beta \ge d_0 > 0$ for positive constant d_0, then the error for SD satisfies

$$\|u - u^h\| \le Ch^{3/2}|u|_2; \tag{10}$$

this is shown without a duality argument or elliptic regularity.

2.2. Streamline–crosswind diffusion method. SD suffers from excessive overshooting and undershooting of front following characteristics when discontinuities are present [21]. Johnson, Schatz and Wahlbin [23] introduced a modification of the SD discretization that improves its performance by adding artificial crosswind diffusion. The *streamline–crosswind diffusion method* (SD/CD) as generalized by Lube [26] is as follows: find $u \in V_h^0$ such that

$$B_{sd/cd}(u^h, v) = F_v \quad \forall v \in V_h^0, \tag{11}$$

where

$$B_{sd/cd}(u^h, v) = B_{sd}(u^h, v) + (\varepsilon_m - \varepsilon)(u^h_\alpha, v_\alpha), \tag{12}$$

$\alpha = (-\beta_2, \beta_1)$ is the crosswind vector and the coefficient of artificial crosswind diffusion is defined by

$$\varepsilon_m = \begin{cases} \varepsilon & \text{for } \varepsilon \ge h^{3/2} \\ h^{3/2} & \text{for } \varepsilon < h^{3/2}. \end{cases}$$

For this method with piecewise linear elements, pointwise error bounds of order $O(h^2|\log h|)$ have been obtained for special meshes in [40], where it is also shown that the width of the characteristic boundary layers and interior layers along streamlines are of order $O(h^{5/8}\log^2 h)$. See also [31, pp. 229ff.] for discussion of such results.

In our numerical experiments, we find that this method dramatically reduces the oscillations of discrete solutions near boundary layers and internal layers, but there are problems with smearing near sharp fronts.

2.3. Two–parameter streamline–crosswind diffusion scheme. In [35], we introduced a two–parameter variant of the SD/CD discretization. As in (12), we add crosswind diffusion to the SD operator, producing a parameterized weak formulation (denoted MSD/CD)

$$B_{msd}(u^h, v) = F_v \quad \forall v \in V_h^0, \tag{13}$$

where

$$B_{msd}(u^h, v) = B_{sd}(u^h, v) + \delta_c(u^h_\alpha, v_\alpha). \tag{14}$$

The two parameters δ_s (see (4)) and δ_c determine the amount of streamline diffusion and crosswind diffusion added to the system, respectively. For constant β_1, β_2, rather than being free parameters, these are explicitly determined so that necessary conditions for uniform convergence in l^2 of u^h with respect to ε are satisfied; see [35], [37] for discussion of these conditions. This leads to the values

$$\delta_s = \frac{h}{|\beta|^2}\left(\frac{1}{2}\frac{\beta_1^3\coth\frac{\beta_1 h}{2\varepsilon} - \beta_2^3\coth\frac{\beta_2 h}{2\varepsilon}}{\beta_1^2-\beta_2^2} - \varepsilon\right), \tag{15}$$

$$\delta_c = \frac{h}{|\beta|^2}\left(\frac{1}{2}\frac{\beta_1^2\beta_2\coth\frac{\beta_2 h}{2\varepsilon} - \beta_2^2\beta_1\coth\frac{\beta_1 h}{2\varepsilon}}{\beta_1^2-\beta_2^2} - \varepsilon\right), \tag{16}$$

for bilinear elements. For variable flows or irregular quadrilateral grids, we can define local (to element) values of δ_s and δ_c. We follow the approach given in [4]: on any element τ, let (x_τ, y_τ) denote the element center, let $\beta_\tau = \beta(x_\tau, y_\tau)$, and let h_τ be the diameter of τ_h. Then these constant values are used to define the parameters in formulas (7), (15)–(16) in the local matrix computations associated with the element τ_h.

In [35], we have shown that the form B_{msd} satisfies the coercivity condition

$$B_{msd}(v,v) \geq \varepsilon\|\nabla v\|^2 + \delta_s\|v_\beta\|^2 + \delta_c\|v_\alpha\|^2, \quad \forall v \in V_h^0,$$

so the finite element matrix has positive definite symmetric part and the discrete solution of (13) is unique. If u^h is the discrete solution obtained by MSD/CD on either bilinear or linear elements and $\beta \in W^{1,\infty}(\Omega)$ and either $\nabla\cdot\beta = 0$ or $-\nabla\cdot\beta \geq d_0 > 0$, for constant d_0, then the discretization error satisfies

$$\|u - u^h\|_{msd} \leq C\left(\varepsilon^{1/2}h + \delta_s^{1/2}h + \delta_c^{1/2}h + \delta_s^{-1/2}h^2 + h^2 + \delta_c\right)|u|_2, \tag{17}$$

for constant $C > 0$, where $\|v\|_{msd}^2 = \varepsilon\|\nabla v\|^2 + \delta_s\|v_\beta\|^2 + \delta_c\|v_\alpha\|^2$.

2.4. The algebraic systems. We identify some additional properties of the algebraic systems of equations obtained from the discretizations above. First, let

$$\mathcal{A}\mathbf{u} = b, \tag{18}$$

denote the matrix equation obtained by any of SD, SD/CD or MSD/CD. Following the notation in [16], the coefficient matrix can be expressed as

$$\begin{aligned}\mathcal{A}_{sd} &= \varepsilon\mathcal{H} + \mathcal{S} + \delta_s\mathcal{U}\\ \mathcal{A}_{sd/cd} &= \varepsilon\mathcal{H} + \mathcal{S} + \delta_s\mathcal{U} + (\varepsilon_m - \varepsilon)\mathcal{C}\\ \mathcal{A}_{msd} &= \varepsilon\mathcal{H} + \mathcal{S} + \delta_s\mathcal{U} + \delta_c\mathcal{C},\end{aligned}$$

for SD, SD/CD, MSD/CD, respectively, where

$$\mathcal{H}_{i,j} = (\nabla\phi_j, \nabla\phi_i),\ \mathcal{S}_{i,j} = (\beta\cdot\nabla\phi_j, \phi_i),\ \mathcal{U}_{i,j} = (\beta\cdot\nabla\phi_j, \beta\cdot\nabla\phi_i),\ \mathcal{C}_{i,j} = (\alpha\cdot\nabla\phi_j, \alpha\cdot\nabla\phi_i),$$

and $\{\phi_i\}_{i=1}^{(N-1)^2}$ are the finite element basis functions. If $\nabla\cdot\beta = 0$, then for each basis function ϕ_i having value 0 on the boundary, it follows from integration by parts that

$$(\beta\cdot\nabla\phi_j, \phi_i) = -(\beta\cdot\nabla\phi_i, \phi_j),$$

that is, $\mathcal{S}$ is skew-symmetric. It is then easy to see that the symmetric parts of $\mathcal{A}_{sd}, \mathcal{A}_{sd/cd}$ and $\mathcal{A}_{msd}$ are positive definite. For constant β and bilinear basis functions, the constituent

9-point stencils are as follows:

$$\mathcal{H}: \quad \begin{pmatrix} -\frac{1}{3} & -\frac{1}{3} & -\frac{1}{3} \\ -\frac{1}{3} & \frac{8}{3} & -\frac{1}{3} \\ -\frac{1}{3} & -\frac{1}{3} & -\frac{1}{3} \end{pmatrix}$$

(19)

$$\mathcal{S}: \quad \begin{pmatrix} \frac{h}{12}(-\beta_1+\beta_2) & \frac{h}{3}\beta_2 & \frac{h}{12}(\beta_1+\beta_2) \\ -\frac{h}{3}\beta_1 & 0 & \frac{h}{3}\beta_1 \\ -\frac{h}{12}(-\beta_1-\beta_2) & -\frac{h}{3}\beta_2 & \frac{h}{12}(\beta_1-\beta_2) \end{pmatrix}$$

(20)

$$\mathcal{U}: \quad \begin{pmatrix} \frac{-\beta_1^2-\beta_2^2+3\beta_1\beta_2}{6} & \frac{\beta_1^2-2\beta_2^2}{3} & \frac{-\beta_1^2-\beta_2^2-3\beta_1\beta_2}{6} \\ \frac{-2\beta_1^2+\beta_2^2}{3} & \frac{4(\beta_1^2+\beta_2^2)}{3} & \frac{-2\beta_1^2+\beta_2^2}{3} \\ \frac{-\beta_1^2-\beta_2^2-3\beta_1\beta_2}{6} & \frac{\beta_1^2-2\beta_2^2}{3} & \frac{-\beta_1^2-\beta_2^2+3\beta_1\beta_2}{6} \end{pmatrix}$$

(21)

$$\mathcal{C}: \quad \begin{pmatrix} \frac{-\beta_1^2-\beta_2^2-3\beta_1\beta_2}{6} & \frac{-2\beta_1^2+\beta_2^2}{3} & \frac{-\beta_1^2-\beta_2^2+3\beta_1\beta_2}{6} \\ \frac{\beta_1^2-2\beta_2^2}{3} & \frac{4(\beta_1^2+\beta_2^2)}{3} & \frac{\beta_1^2-2\beta_2^2}{3} \\ \frac{-\beta_1^2-\beta_2^2+3\beta_1\beta_2}{6} & \frac{-2\beta_1^2+\beta_2^2}{3} & \frac{-\beta_1^2-\beta_2^2-3\beta_1\beta_2}{6} \end{pmatrix}.$$

3. Nonlinear stabilized discretizations. In this section, we describe two nonlinear stabilization strategies based on shock capturing, whose discrete solutions display less overshooting and undershooting within numerical layers than those produced by the streamline diffusion method.

3.1. Shock capturing. Hughes, Mallet and Mizukami in [21] introduced a *shock–capturing finite element method* (SC) which adds an extra discontinuity capturing term to SD. Let $\beta_{||}$ denote the projection of the flow field onto the gradient of the discrete solution u^h, that is,

$$\beta_{||} = \frac{\beta \cdot \nabla u^h}{|\nabla u^h|^2} \nabla u^h, \quad \text{for } |\nabla u^h| \neq 0.$$

The SC method for (1) is: find $u^h \in V_0^h$ such that

$$B_{sc}(u^h, v) = F_v \quad \text{for all } v \in V_0^h, \tag{22}$$

where

$$\begin{aligned} B_{sc}(u^h, v) &= B_{sd}(u^h, v) + B_{dc}(u^h, v), & (23)\\ B_{dc}(u^h, v) &= \left(r(u^h), \delta_{sc}\beta_{||} \cdot \nabla v\right) & (24) \end{aligned}$$

and $r(u^h) = -\varepsilon\Delta u^h + \beta\cdot\nabla u^h - f$ is the discrete residual of (1). The shock capturing term of (24) depends on the residual of discrete solution, and it tends to add diffusion in regions where the gradient is large, that is, near internal and boundary layers. However, because the discontinuity capturing term depends on the discrete solution, the discrete algebraic system derived from this strategy is nonlinear. In [21], δ_s, δ_{sc} are chosen by the following formulas: let $P_{||} = \frac{h|\beta_{||}|}{2\varepsilon}$ denote the mesh Péclet number for the vector $\beta_{||}$ and let

$$\delta_s = \gamma\frac{h}{|\beta|}, \quad \text{for } \gamma = \gamma_0 \min\left(1, \frac{P}{3}\right), \tag{25}$$

$$\delta_{sc} = \max\left(0, \delta_{||} - \delta_s\right), \quad \text{for } \delta_{||} = \gamma_0\frac{h}{|\beta_{||}|}\min\left(1, \frac{P_{||}}{3}\right), \tag{26}$$

where $\gamma_0 = 1/2$ for linear and bilinear elements. The values of δ_s, δ_{sc} of (25), and (26) are determined locally in each element τ_h, using the element diameter h_τ for h and the values of β and $\beta_{||}$ at the element center.

Johnson, Szepessy and Hansbo [24] and Szepessy [39] have shown that the accuracy of SC for conservation laws is of order of O($h^{3/2}$) for smooth solutions if piecewise linear functions are used.

3.2. Shock capturing with crosswind dissipation. It may happen that for some $u^h \in V_0^h$, the discontinuity capturing term $B_{dc}(u^h, u^h)$ of (24) is negative, so that negative numerical diffusion may be added to the system. An alternative that avoids this difficulty is as follows. Galeão and Dutra do Carmo in [17] modified (24) using

$$B_{dc}(u^h, v) = \left(r(u^h), \delta_{sc}\beta_r\cdot\nabla v\right) = \left(\delta_{sc}|\beta_r|^2\nabla u^h, \nabla v\right), \tag{27}$$

where $\beta_r = \frac{r(u^h)}{|\nabla u^h|^2}\nabla u^h$. For this choice, $B_{sc}(u^h, u^h) \geq 0$ for all $u^h \in V_0^h$, and (27) is identical to (24) when $f = 0$. Codina [8] refined this approach further by incorporating the crosswind direction into the discontinuity capturing term. The resulting *shock capturing with crosswind dissipation* method (SC/CD) is defined via (22)–(23) and

$$B_{dc}(u^h, v) = (\delta_{sc}|\beta_r|u^h_\alpha, v_\alpha). \tag{28}$$

Here, α is the crosswind vector as defined in Section 2.2 and $\delta_{sc}|\beta_r|$ is evaluated within each element by setting

$$\delta_{sc} = \begin{cases} \gamma_0 h \max\{0, C_0 - 1/P_{||}\} & \text{if } \nabla u^h \neq 0 \\ 0 & \text{otherwise,} \end{cases} \tag{29}$$

and $C_0 = 0.7$ for both linear and bilinear elements.

For completeness, we show that SC/CD discretization satisfies an error bound like those derived for SD in [1], [39], [40]. Assume that the quantity $|\beta_r|$ satisfies

$$q_0 \leq |\beta_r| \leq q_1 \tag{30}$$

for $q_0, q_1 > 0$, and for $-\nabla\cdot\beta \geq d_0 \geq 0$, let

$$|||v|||_{sc}^2 = \varepsilon\|\nabla v\|^2 + \delta_s\|v_\beta\|^2 + q_0\delta_{sc}\|v_\alpha\|^2 + \frac{1}{2}d_0\|v\|^2. \tag{31}$$

The following result establishes the stability of B_{sc} defined by (22)–(23), (28).

LEMMA 3.1. *The bilinear form B_{sc} satisfies*

$$B_{sc}(v,v) \geq \|\!|v|\!\|_{sc}^2 \quad \text{for any } \; v \in V_h^0. \tag{32}$$

Proof. From (8), we have

$$B_{sc}(v,v) \geq \varepsilon\|\nabla v\|^2 + \delta_s\|v_\beta\|^2 + \frac{1}{2}d_0\|v\|^2 + B_{dc}(v,v),$$

and from (28),

$$B_{dc}(v,v) \geq q_0\delta_{sc}\|v_\alpha\|^2.$$

The result follows from the definition (31). ∎

The error estimate showing for SC/CD is as follows. The proof is as in [35]. A similar result for SC can be found in [24], [39].

THEOREM 3.2. *Let u be the solution of (1)–(2) with $g = 0$ and $u \in H^2(\Omega) \cap H_0^1(\Omega)$. Let $\beta \in W^{1,\infty}(\Omega)$ and either $\nabla \cdot \beta = 0$ or $-\nabla \cdot \beta \geq d_0 > 0$, for constant d_0, and let the residual $r(u^h)$ satisfy (30), for constants q_0, q_1. If u^h is the discrete solution obtained by SC/CD, on either bilinear or linear elements, then there is a constant C such that*

$$\|\!|u - u^h|\!\|_{sc} \leq Ch^{3/2}|u|_2. \tag{33}$$

Proof. Let $\zeta = u^I - u$, where u^I is the bilinear (or linear) interpolant of u. It follows that

$$\begin{aligned} \|\zeta\| &\leq ch^2|u|_2, \quad \|\nabla\zeta\| \leq ch|u|_2 && (34)\\ \|\!|\zeta|\!\|_{sc} &\leq c\left(\varepsilon^{1/2}h + \delta_s^{1/2}h + \delta_{sc}^{1/2}h + h^2\right)|u|_2 && (35) \end{aligned}$$

(see [22, p. 176], [31, p. 232]). Setting $\eta = u^I - u^h$ yields

$$\|\!|\eta|\!\|_{sc}^2 \leq B_{sc}(\eta,\eta) = B_{sc}(\zeta,\eta) + B_{sc}(u - u^h, \eta). \tag{36}$$

The quasi–orthogonality relation holds,

$$B_{sc}(u - u^h, v) = Per(u,v) \quad \text{for all } v \in V_h^0,$$

where $Per(u,v)$ is the truncation error

$$Per(u,v) = \varepsilon(\Delta u, \delta_s v_\beta).$$

But the Poincaré inequality leads to

$$B_{sc}(u - u^h, \eta) \leq \varepsilon\delta_s^{1/2}\|\Delta u\|\delta_s^{1/2}\|\eta_\beta \leq \varepsilon\delta_s^{1/2}|u|_2\,\|\!|\eta|\!\|_{sc} \tag{37}$$

and

$$B_{sc}(\zeta,\eta) = \varepsilon(\nabla\zeta,\nabla\eta) + \delta_s(\zeta_\beta,\eta_\beta) + B_{dc}(\zeta,\eta) - (\zeta,\eta_\beta) - \int_\Omega(\nabla\cdot\beta)\zeta\eta\, d\Omega.$$

If $\nabla \cdot \beta = 0$ (divergence free case), then

$$\begin{aligned} B_{sc}(\zeta,\eta) &\leq \varepsilon\|\nabla\zeta\|\,\|\nabla\eta\| + \delta_s\|\zeta_\beta\|\,\|\eta_\beta\| + q_1\delta_{sc}\|\zeta_\alpha\|\|\eta_\alpha\| + \|\zeta\|\,\|\eta_\beta\| \\ &\leq \left(\varepsilon^{1/2}\|\nabla\zeta\| + \delta_s^{1/2}\|\zeta_\beta\| + \delta_{sc}^{1/2}q_0^{-1/2}q_1\|\zeta_\alpha\| + \delta_s^{-1/2}\|\zeta\|\right)\|\!|\eta|\!\|_{sc}. \end{aligned}$$

If $-\nabla \cdot \beta \geq d_0 > 0$, then

$$\int_\Omega (\nabla \cdot \beta)\zeta\eta \, d\Omega \leq c\|\zeta\| \, \|\eta\| \leq c'\|\zeta\| \|\!|\eta|\!\|_{sc}$$

and

$$B_{sc}(\zeta, \eta) \leq \left(\varepsilon^{1/2}\|\nabla\zeta\| + \delta_s^{1/2}\|\zeta_\beta\| + \delta_{sc}^{1/2} q_0^{-1/2} q_1 \|\zeta_\alpha\| + \delta_s^{-1/2}\|\zeta\| + c'\|\zeta\| \right) \|\!|\eta|\!\|_{sc}.$$

Use (34) to get

$$B_{sc}(\zeta, \eta) \leq c \left(\varepsilon^{1/2} h + \delta_s^{1/2} h + \delta_s^{-1/2} h^2 + h^2 + \delta_{sc}^{1/2} h \right) |u|_2 \|\!|\eta|\!\|_{sc}. \tag{38}$$

Combining (36), (37) and (38) gives

$$\|\!|\eta|\!\|_{sc} \leq c \left(\varepsilon^{1/2} h + \delta_s^{1/2} h + \delta_s^{-1/2} h^2 + h^2 + \delta_{sc}^{1/2} h \right) |u|_2.$$

Thus, using the triangle inequality and (35) – (36), it follows that

$$\begin{aligned} \|\!|u - u^h|\!\|_{sc} &\leq \|\!|\eta|\!\|_{sc} + \|\!|\zeta|\!\|_{sc} \\ &\leq C \left(\varepsilon^{1/2} h + \delta_s^{1/2} h + \delta_s^{-1/2} h^2 + h^2 + \delta_{sc}^{1/2} h \right) |u|_2. \end{aligned}$$

The result follows by taking δ_s, δ_{sc} to be of magnitude $O(h)$. ∎

3.3. The algebraic systems. The nonlinear discrete systems above can be expressed as

$$\mathcal{F}(\mathbf{u}) = \mathcal{A}_{sd}\mathbf{u} - b + \mathcal{R}(\mathbf{u}) = \mathbf{0} \tag{39}$$

where $\mathcal{R}(\mathbf{u})$ is the nonlinear shock–capturing term derived from

$$\mathcal{R}_i(u^h) = \begin{cases} \delta_{sc} \left(r(u^h), \beta_{\|} \cdot \nabla\phi_i \right) & \text{for SC} \\ \delta_{sc} \left(|\beta_r| u^h_\alpha, \alpha \cdot \nabla\phi_i \right) & \text{for SC/CD,} \end{cases}$$

where u^h is the discrete solution. Note that the nonlinear discrete function $\mathcal{F}(\mathbf{u})$ is not differentiable because it contains absolute value and maximum functions. This will influence the convergence behavior of solution algorithms for these systems; see Section 4.2.

4. Solution algorithms. In this section, we briefly review some linear and nonlinear solution algorithms based on Krylov subspace methods for solving the systems obtained from the discretized schemes of Sections 2–3.

4.1. GMRES method. For the linear problems (18), we use the generalized minimal residual method (GMRES) developed by Saad and Schultz [32, 33]. Given an initial value $\mathbf{u}_0$, let $\mathbf{K}_i(\mathcal{A}, r_0)$ denote the Krylov subspace

$$\mathbf{K}_i(\mathcal{A}, r_0) = \text{span}\left\{ r_0, \mathcal{A}r_0, ..., \mathcal{A}^{i-1}r_0 \right\}. \tag{40}$$

GMRES constructs $\mathbf{u}_i \in \mathbf{u}_0 + \mathbf{K}_i(\mathcal{A}, r_0)$ whose residual norm is minimal. It constructs an orthonormal basis for $\mathbf{K}_i(\mathcal{A}, r_0)$ by the Arnoldi process, which can be viewed as a variant of the Gram-Schmidt orthogonalization procedure. A statement of the algorithm is as follows:

Algorithm 2: GMRES

Choose $\mathbf{u}_0$, compute $r_0 = b - A\mathbf{u}_0$
Let $i = 0, \rho = \|r_0\|_2, v_0 = r_0/\rho$
While ρ does not satisfy the stopping criterion, do
  $i = i + 1,\ \beta = \rho$
  The Arnoldi process/modified Gram-Schmidt orthogonalization:
    $w = Av_{i-1}$
    For $j = 1, ..., i$
      $h_{j,i} = w^T v_j$
      $w = w - h_{j,i} v_j$
    $h_{i+1,i} = \|w\|_2$
    $v_{i+1} = w/h_{i+1,i}$
  Compute $\rho = \min_{y_i} \left\| \beta e_1 - \hat{H}_i y_i \right\|_2$, for $\hat{H}_i = (h_{i,j})\,, e_1 \in \mathrm{I\!R}^i$
Enddo
$\mathbf{u}_i = \mathbf{u}_0 + V_i y_i$, for $V_i = [v_1, \ldots, v_i]$

4.2. The Newton–GMRES algorithm. For the nonlinear system (39), we use a Newton–like iteration in which the system of equations to be solved at each Newton step is solved approximately by a Krylov subspace method [5, 11, 14, 25]. We use Newton–GMRES as in [11]. That is, for the system

$$\mathcal{F}(\mathbf{u}_k) + \mathcal{F}'(\mathbf{u}_k)s_k = 0, \tag{41}$$

we compute s_k such that

$$\|\mathcal{F}(\mathbf{u}_k) + \mathcal{F}'(\mathbf{u}_k)s_k\|_2 \leq \eta_k \|\mathcal{F}(\mathbf{u}_k)\|_2, \tag{42}$$

where a GMRES-like iteration is used to enforce the criterion (42). Moreover, rather than use true GMRES of Algorithm 2, we approximate the matrix vector product by a directional derivative [7]. That is

$$\mathcal{F}'(\mathbf{u}_k)v \approx D_\sigma \mathcal{F}(\mathbf{u}_k; v) \equiv \frac{\mathcal{F}(\mathbf{u}_k + \sigma v) - \mathcal{F}(\mathbf{u}_k)}{\sigma} = \mathcal{A}_{sd} v + \frac{\mathcal{R}(\mathbf{u}_k + \sigma v) - \mathcal{R}(\mathbf{u}_k)}{\sigma}. \tag{43}$$

In practice, most entries of $\mathcal{R}$ are zero, except where the discrete residual $r(\mathbf{u})$ does not vanish. Note that we are avoiding evaluation of the Jacobian for the nonlinear system, both because it is not well–defined everywhere and because it is expensive.

The Newton–GMRES algorithm is as follows:

Algorithm 4: Inexact Newton–GMRES

Choose $\mathbf{u}_0, \gamma$ and let $k = 0,\ r = -\mathcal{F}(\mathbf{u}_0)$
While $||\mathcal{F}(\mathbf{u}_k)||_2$ does not satisfy the stopping criterion, do
  Let $k = k + 1,\ \rho = ||r||_2,\ v_1 = r/||r||_2$ and choose η_k
  While $\rho > \eta_k||\mathcal{F}(\mathbf{u}_k)||_2$, do
    Use GMRES as in Algorithm 3 with (43) for the matrix–vector product
  Enddo
  Let $\Delta\mathbf{u}_k = V_m y_m$ after m GMRES steps
  Let $\mathbf{u}_k = \mathbf{u}_{k-1} + \Delta\mathbf{u}_k, \lambda = 1, \varrho = 1 - \eta_k$ and evaluate $\mathcal{F}(\mathbf{u}_k)$
  While $||\mathcal{F}(\mathbf{u}_k)||_2 > (1 - \gamma\varrho)\,||\mathcal{F}(\mathbf{u}_{k-1})||_2$, do
    Choose $\nu \in (0, 1)$
    $\lambda = \nu\lambda, \varrho = \nu\varrho$
    $\mathbf{u}_k = \mathbf{u}_{k-1} + \lambda\Delta\mathbf{u}_k$ and evaluate $\mathcal{F}(\mathbf{u}_k)$
  Enddo
Enddo

The third *while* loop is for backtracking and forces a minimal improvement in the solution before a step is performed. For the forcing sequence $\{\eta_k\}$, we use the choice

$$\eta_k = \min\left\{\eta_{max}, \max\left(\eta_k^c, \frac{\tau}{2||F(\mathbf{u}_k)||_2}\right)\right\}, \tag{44}$$

where

$$\eta_k^c = \begin{cases} \min\left(\eta_{\max}, \gamma\frac{||F(\mathbf{u}_k)||_2^2}{||F(\mathbf{u}_{k-1})||_2^2}\right), & \text{if } \gamma\eta_{k-1}^2 \le 0.1 \\ \min\left\{\eta_{\max}, \gamma\max\left(\frac{||F(\mathbf{u}_k)||_2^2}{||F(\mathbf{u}_{k-1})||_2^2}, \eta_{k-1}^2\right)\right\}, & \text{if } \gamma\eta_{k-1}^2 > 0.1 \end{cases} \tag{45}$$

for given $\eta_{\max}, \gamma$. The strategy (45) is taken from Eisenstat and Walker [14] with modification (44) due to Kelley [25]. This prohibits the computation of an overly accurate linear solution when $\mathbf{u}_k$ is far from the solution. For the other parameters, we use $\gamma = 0.9, \eta_{max} = 10^{-4}$, σ from [12], and choose ν to minimize a quadratic polynomial function as suggested in [12, p. 126], [25, p. 142].

TABLE 1
Operation counts (multiplications) for GMRES and Newton–GMRES with matrix of dimensions N×N.

Cost at step k for linear GMRES	(k+3+1/k)N +NZ
Cost at k-th GMRES step of one inexact Newton step without backtracking	(k+4+1/k)N+F$_{ev}$+NZ

Let NZ denote the number of nonzero elements in $\mathcal{A}$, and let F$_{ev}$ represent the required operation counts for evaluating the shock capturing term $\mathcal{R}(\mathbf{u}^h)$. Here, NZ is approximately 9N for the matrix of dimensions N×N, and F$_{ev}$ is only counted where $\nabla u^h \neq 0$. A summary of operation counts for GMRES and inexact Newton–GMRES is shown in Table 1.

5. Preconditioning. Convergence of Krylov subspace methods can be significantly enhanced using preconditioners. In this section, we outline the preconditioning strategies we use. Because we are solving problems with large Reynolds numbers, we restrict our attention to easy-to-implement preconditioners of algebraic type derived from relaxation methods and incomplete factorization. (See [10], [30], for examples of alternative approaches based on multigrid.)

We consider six preconditioning strategies, for problems with an underlying rectangular grid. The first two of these could be implemented efficiently on parallel computers; the latter four take some account of the orientation of flow in the problem.

1. *Horizontal line Jacobi* (HJ). Let the grid points be ordered in a natural left-to-right, bottom-to-top ordering. Then the coefficient matrix has form

   $$\mathcal{A} = \mathcal{D} + \mathcal{L} + \mathcal{U}, \tag{46}$$

   where $\mathcal{D}$ is a tridiagonal matrix representing the connections within each equation in the horizontal direction, and $\mathcal{L}$ and $\mathcal{U}$ are strictly lower and upper triangular, respectively. The horizontal line Jacobi preconditioner is $\mathcal{M} = \mathcal{D}$.

$$\begin{array}{ccc} \times & \times & \times \\ \otimes & \otimes & \otimes \\ \times & \times & \times \end{array}$$

(a) horizontal ordering

$$\begin{array}{ccc} \times & \otimes & \times \\ \times & \otimes & \times \\ \times & \otimes & \times \end{array}$$

(b) vertical ordering

FIG. 1. *Grid points used for line Jacobi preconditionings.*

2. *Vertical line Jacobi* (VJ). Alternatively, if the grid points are ordered first from bottom-to-top and then from left to right, this corresponds to a permutation of $\mathcal{A}$,

   $$P\mathcal{A}P^T = \mathcal{D}_V + \mathcal{L}_V + \mathcal{U}_V \tag{47}$$

   and the vertical line Jacobi preconditioner is $\mathcal{M} = P^T\mathcal{D}_V P$. This preconditioning can be implemented efficiently without explicit use of the permutation.

Symbolic representations of the line Jacobi operators are shown in Fig. 1.

$$\begin{array}{ccc} \times & \times & \times \\ \otimes & \otimes & \otimes \\ \otimes & \otimes & \otimes \end{array}$$

(a) horizontal ordering

$$\begin{array}{ccc} \otimes & \otimes & \times \\ \otimes & \otimes & \times \\ \otimes & \otimes & \times \end{array}$$

(b) vertical ordering

FIG. 2. *Grid points used for line Gauss–Seidel preconditionings.*

3. *Horizontal line Gauss–Seidel* (HGS). The preconditioner is defined by $\mathcal{M} = \mathcal{D} - \mathcal{L}$, where $\mathcal{D}, \mathcal{L}$ are as in (46).
4. *Vertical line Gauss–Seidel* (VGS). The preconditioner is defined by $\mathcal{M} = P^T(\mathcal{D}_V - \mathcal{L}_V)P$, where $\mathcal{D}_V, \mathcal{L}_V$ are as in (47).

The line Gauss–Seidel preconditioners are represented symbolically in Figure 2.

5. *Incomplete block factorization* (IB_1). If there are n horizontal grid lines, then the coefficient matrix has the form

   $$\mathcal{A} = [\mathcal{A}_{i,j}]_{1 \le i,j \le n}\,.$$

The incomplete block factorization of Concus, Golub and Meurant [9] is

$$\mathcal{M} = (X + \mathcal{L})\, X^{-1}\, (X + \mathcal{U})\,.$$

Here

$$X = \text{blockdiag}\,[X_1, \ldots, X_n]$$

is defined by the recurrence below, where $[\cdot]^{(p)}$ denotes the matrix with half–bandwidth p.

$$\boxed{\begin{array}{l} X_1 = \mathcal{A}_{1,1} \\ \text{For } i = 1, 2, \ldots, n-1, do \\ \qquad Y_i = [X_i]^{(p)} \\ \qquad X_{i+1} = \mathcal{A}_{i+1,i+1} - \left[\mathcal{A}_{i+1,i} Y_i \mathcal{A}_{i,i+1}\right]^{(p)}. \\ \text{Enddo} \end{array}}$$

Since $\mathcal{A}$ is a positive real matrix, the sequence matrices $\{X_i\}$, $\{Y_i\}$ remain positive real and nonsingular, for sufficient large p (see [3]). In our numerical tests, we let $p = 1$ for simplicity.

6. *Incomplete block factorization 2* (IB_2). In this variant of incomplete factorization, due to Axelsson [2], the factors are expressed in term of the inverse of the block diagonal, i.e.,

$$\mathcal{M} = \left(Y^{-1} + \mathcal{L}\right) Y \left(Y^{-1} + \mathcal{U}\right),$$

where

$$Y = \text{blockdiag}\,[Y_1, \ldots, Y_n]$$

is defined as above.

TABLE 2
Operation counts (flops) for preconditioners with matrix of dimensions $N \times N$.

	VJ	HJ	VGS	HGS	IB_1	IB_2
Preprocessing cost per step	2N	2N	2N	2N	31N	31N
Substitutions	3N	3N	6N	6N	12N	12N

The operation counts for these preconditioners are shown in Table 2. The first line of the table reflects the cost of factorization (for example, of the tridiagonal matrix $\mathcal{D}$ in Jacobi preconditioning), assuming no pivoting is needed. Under this assumption, the preconditioners all have essentially the same sparsity requirements as the coefficient matrix.

For the linear discretizations of Section 2, we apply the preconditioning on the right, and solve

$$\mathcal{A}\mathcal{M}^{-1}\hat{\mathbf{u}} = b, \quad \hat{\mathbf{u}} = \mathcal{M}\mathbf{u},$$

where $\mathcal{M}$ is the preconditioning matrix. For the nonlinear problems, we precondition by replacing the directional derivative in the direction of v_i with $D_\sigma \mathcal{F}(\mathbf{u}; \mathcal{M}^{-1} v_i)$, and define the correction to be

$$\Delta\mathbf{u}_k = \mathcal{M}^{-1} V_k y_k.$$

We take as preconditioners for the nonlinear iteration approximations to the streamline diffusion operator $\mathcal{A}_{sd}$; these approximations are determined using the six approaches listed above.

6. Numerical experiments. In this section, we compare the performance of the discretization strategies of Sections 2-3 and the solution algorithms of Sections 4-5 for solving a set of benchmark problems. All experiments use bilinear shape functions on square elements on a uniform $N \times N$ element grid with $h = 1/N$, and they were performed with MATLAB Version 4.2c on a SUN SPARC–20 workstation. All discretizations were employed with 2×2 Gauss quadrature. The coefficient $\delta_s = \omega_s h$ of the streamline diffusion term for both SD and SD/CD was chosen using (7). We use $\mathbf{u}_0 = 0$ as initial guess for solving all linear systems of equations and $\mathbf{u}_0 = \mathbf{u}_{sd}$ for all nonlinear systems of equations where $\mathbf{u}_{sd}$ is the solution of the discrete problem on the given grid obtained from SD discretization.[2]

We present two types of results. First, we examine the behavior, i.e., iteration counts and operation counts, of various solution algorithms for a series of choices of parameters and mesh sizes, without regard to quality of solution. This gives a general idea of the costs of solving the discrete problems, but it ignores the fact that certain discretizations such as shock–capturing may produce more accurate solutions on a given mesh. In a second set of tests, we attempt to factor solution quality into our assessment. We define criteria to measure solution accuracy, use these criteria to identify mesh sizes for which each discretization achieves a specified accuracy, and then use "good" choices of algorithms (determined by the first set of results) to assess the effectiveness of the discretizations.

6.1. Benchmark problems. We consider two benchmark problems.

Problem 1: Characteristic and downstream boundary layers. This problem was first considered in [20] for studying a downstream boundary layer and a characteristic internal layer that propagates along the characteristics when inflow boundary conditions are discontinuous. The velocity field β is given by $(\cos\theta, \sin\theta)$, and the boundary values are as follows:

$$u = \begin{cases} 1 & \text{if } 0 \le y < 1/2,\ x = 0 \ \text{ or } \ y = 0,\ 0 \le x < 1 \\ 0 & \text{otherwise.} \end{cases}$$

The reduced problem (i.e., where $\varepsilon = 0$ in (1)) has discontinuous solution

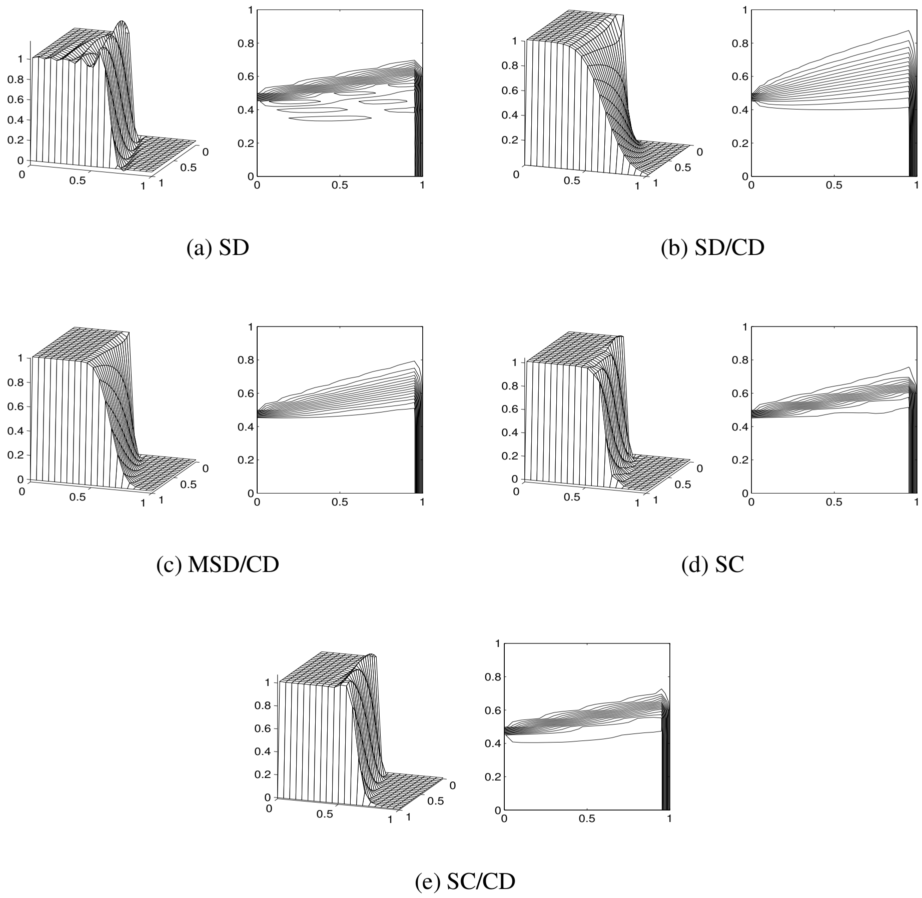
$$u = \begin{cases} 1 & y < \frac{\beta_2}{\beta_1}x + \frac{1}{2} \\ 0 & y > \frac{\beta_2}{\beta_1}x + \frac{1}{2}. \end{cases}$$

For $\varepsilon > 0$, there is an internal layer of width $O(\sqrt{\varepsilon})$ across the characteristic $y = \frac{\beta_2}{\beta_1}x + \frac{1}{2}$, and a boundary layer of width $O(\varepsilon)$ at $x = 1$ [13]. Figure 3 depicts the three–dimensional structure and contour plots of the numerical solutions obtained by the six methods tested, for $\varepsilon = 10^{-6}$, $h = 1/20$ and $\beta = (\cos 10^o, \sin 10^o)$. The three-dimensional plots are rotated 110^o to give a clearer picture of the layers.

Problem 2: Variable flow field. For our second benchmark problem, we consider a variant of the "IAHR/CEGB" workshop problem [36] in common use for testing discretization strategies (see e.g. [19], [27]). The domain is the rectangular region

$$\Omega = \{(x, y) \mid -1 < x < 1,\ 0 < y < 1\},$$

[2] We also tried $\mathbf{u}_0 = 0$ as an initial guess for solving the nonlinear systems of equations and found that this requires more Newton steps and more operation counts than those using $\mathbf{u}_{sd}$. The costs of generating $\mathbf{u}_{sd}$ are included in all performance assessments.



(a) SD

(b) SD/CD

(c) MSD/CD

(d) SC

(e) SC/CD

FIG. 3. *Numerical solutions and contours for Problem 1 with* $\varepsilon = 10^{-6}$, $h = 1/20$ *and* $\theta = 10^{o}$.

and the velocity field is

$$\beta = (2y(1-x^2),\ -2x(1-y^2)).$$

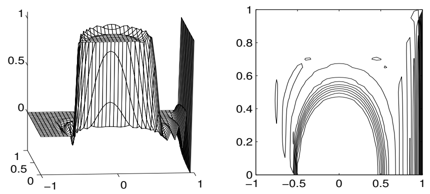
The inflow boundary is the interval $\{(x,0)\,|-1 \leq x \leq 0\}$, and Dirichlet conditions specified there represent an inlet temperature which is convected in a circular flow to the outflow boundary $\{(x,0)\,|\,0 < x < 1\}$, where natural boundary conditions

$$\frac{\partial u(x,0)}{\partial n} = 0, \quad \text{for } 0 < x < 1$$

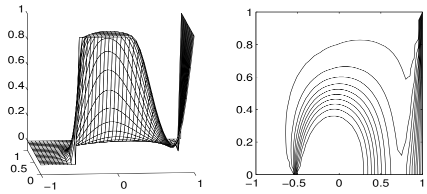
are assigned. Dirichlet boundary conditions are given on the remainder of $\partial\Omega$. There is a discontinuity in the inlet profile

$$u(x,0) = \begin{cases} 0 & -1 \leq x < -0.5 \\ 1 & -0.5 \leq x \leq 0. \end{cases} \tag{48}$$

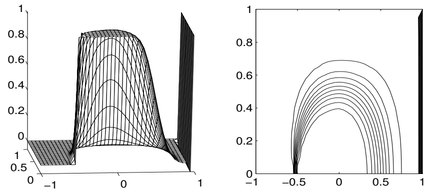
together with Dirichlet conditions $u = 0$ at $x = -1$, $u = 0$ at $y = 1$ and the value $u = 1$ (a hot wall) at $x = 1$ as in [27]; the discontinuity introduces a thin boundary layer at the right boundary. Representative pictures of the three-dimensional structure and contour plots of the



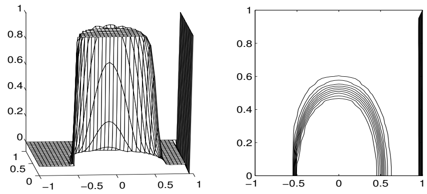
(a) SD



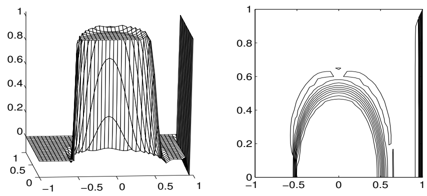
(b) SD/CD



(c) MSD/CD



(d) SC



(e) SC/CD

FIG. 4. *Numerical solutions and contours for Problem* 2, *second variant with* $\varepsilon = 10^{-6}$ *and* $h = 1/20$.

numerical solutions obtained by the six methods tested, for $\varepsilon = 10^{-6}$ and $h = 1/20$, are shown in Figure 4.

Consideration of Figures 3 and 4 gives a qualitative picture of the effectiveness of the six discretization strategies. In particular, SD without additional stabilization yields solutions with oscillations near internal layers. Linear crosswind diffusion diminishes (but does not eliminate) these overshoots and undershoots, but it also leads to excessive smearing. The parameterized MSD/CD method is somewhat more effective (less smearing) in this regard than SD/CD. The nonlinear discretizations, especially SC, yield the qualitatively best solutions, with considerably less oscillation than pure SD and less smearing of fronts than the linear crosswind diffusion schemes.

6.2. Computational results for selected example problems. We first examine the behavior of various discretizations and solution algorithms on a fixed set of meshes. In these tests, the stopping criterion was

$$\|b - \mathcal{A}\mathbf{u}_k\|_2 \le \tau_r \|b\|_2,$$

for all linear problems, and

$$\|\mathcal{F}(\mathbf{u}_k)\|_2 \le \tau_r \|b\|_2$$

for nonlinear problems, where $\tau_r = 10^{-5}$. Tables 3 and 4 show the iteration counts needed by GMRES for the three linear discretizations of Section 2. In all cases, we permit a maximum

TABLE 3
Iterations of GMRES for Problem 1 using SD, SD/CD, MSD/CD to discretize with various meshes, $\varepsilon = 10^{-6}$ and $\theta = 10^o$.

h	Discretization	Preconditioner						
		I	HJ	VJ	HGS	VGS	IB_1	IB_2
1/16	SD	33	27	14	9	7	6	9
	SD/CD	21	19	14	7	4	3	4
	MSD/CD	22	23	10	5	5	4	6
1/32	SD	58	49	21	15	9	8	16
	SD/CD	39	37	19	8	4	4	5
	MSD/CD	40	41	13	5	5	5	10
1/64	SD	100	89	33	23	13	12	30
	SD/CD	72	73	26	8	5	5	10
	MSD/CD	74	75	22	6	5	6	14

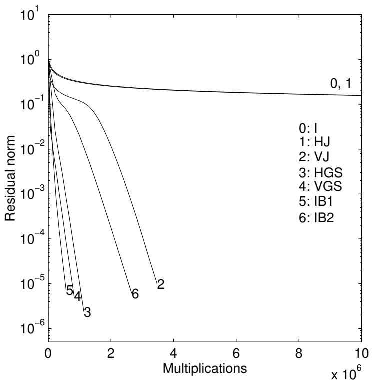
TABLE 4
Iterations of GMRES for Problem 2 using SD, SD/CD, MSD/CD to discretize with various meshes, $\varepsilon = 10^{-6}$. Results marked "$(\cdot)$" did not satisfy stopping tolerance after 200 iterations.*

h	Discretization	Preconditioner						
		I	HJ	VJ	HGS	VGS	IB_1	IB_2
1/16	SD	88	42	46	22	16	9	17
	SD/CD	65	34	33	18	9	6	10
	MSD/CD	66	28	27	17	6	5	14
1/32	SD	158	68	67	39	22	12	29
	SD/CD	115	59	55	32	12	7	14
	MSD/CD	119	46	43	32	6	5	20
1/64	SD	(200)*	110	96	72	27	17	53
	SD/CD	(200)*	88	80	58	15	9	21
	MSD/CD	(200)*	81	77	64	6	5	35

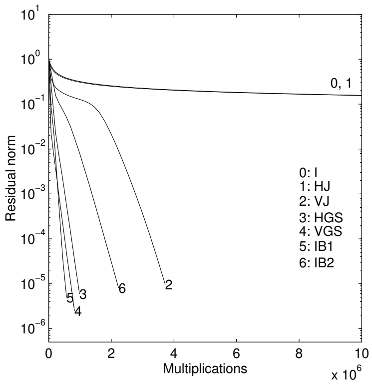
of 200 GMRES steps. The results indicate that the differences in the algebraic systems caused by the introduction of artificial diffusion do in fact influence the performance of iterative algorithms. In particular, the extra (crosswind) diffusion included in SD/CD and MSD/CD leads to linear systems that in every case require fewer iterations to solve than those produced by pure streamline diffusion. We attribute this to the enhanced coercivity produced by crosswind diffusion. It is also clear that preconditioning significantly enhances convergence speed. Among the preconditioners considered, VGS and IB_1 are most effective. For both problems, these strategies correspond most closely to "flow following" computations.[3] The line Jacobi methods are largely ineffective.

Figure 5 expands on these results by plotting the residual norm $\|b - \mathcal{A}\mathbf{u}_k\|_2$ against multiplications, for several choices of the angle of the flow for Problem 1 with MSD/CD discretization. Here, we see that the general trends observed for $\theta = 10^o$ carry over. We have

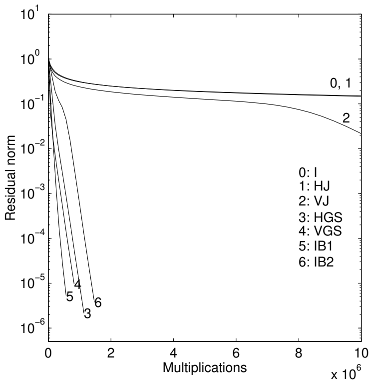
[3] In the case of IB_1, this is true for the forward substitution involving $(X + L)^{-1}$.



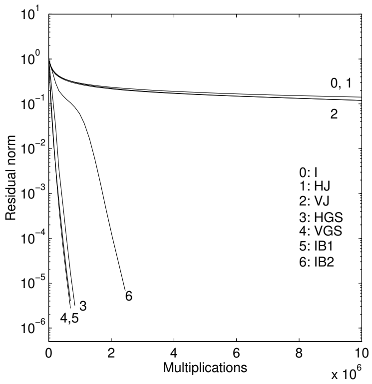
(a) Pb. 1, $\theta = 10^o, \varepsilon = 10^{-6}$



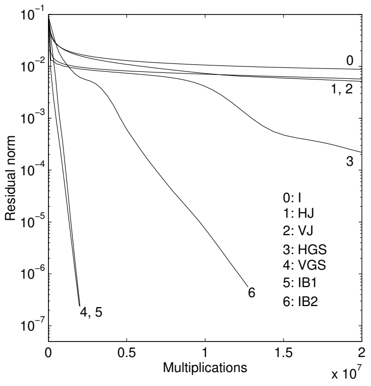
(b) Pb. 1, $\theta = 10^o, \varepsilon = 10^{-3}$



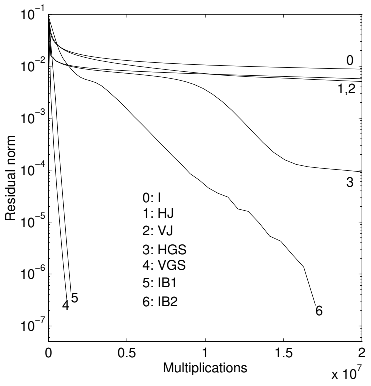
(c) Pb. 1, $\theta = 30^o, \varepsilon = 10^{-3}$



(d) Pb. 1, $\theta = 45^o, \varepsilon = 10^{-3}$



(e) Pb. 2, $\varepsilon = 10^{-3}$



(f) Pb. 2, $\varepsilon = 10^{-6}$

FIG. 5. *Convergence behavior of GMRES for Problems 1 – 2 with* $h = 1/64$*, various* ε, θ*, and MSD/CD discretization.*

also observed similar trends for different (larger) choice of ε, and for different mesh sizes.

Tables 5 and 6 show the performance of the inexact Newton–GMRES algorithm for the nonlinear discretizations of Section 4. For the GMRES computations, we allow a maximum of 50 steps and then restart. Some of the trends displayed here are similar to those observed

TABLE 5

Summary of results for shock capturing discretization for Problem 1 with various meshes, $\varepsilon = 10^{-6}$ and $\theta = 10^{\circ}$. NS, GN and FE refer to numbers of Newton steps, GMRES steps and function evaluations, respectively. Results marked "$(\cdot)$" failed during backtracking.*

h	Discretization		Preconditioner						
			I	HJ	VJ	HGS	VGS	IB_1	IB_2
1/16	SC	NS	9	16	7	8	8	7	10
		GS	83	97	37	37	42	29	42
		FE	258	763	72	96	97	52	162
	SC/CD	NS	7	8	8	6	7	7	7
		GS	51	69	35	23	31	30	36
		FE	134	212	115	49	79	77	94
1/32	SC	NS	(20)*	(21)*	17	8	8	8	9
		GS	(145)*	(194)*	121	51	46	43	68
		FE	(1717)*	(2051)*	2269	163	134	133	274
	SC/CD	NS	8	9	9	8	8	7	10
		GS	105	125	52	37	42	29	42
		FE	294	461	200	96	97	52	162

TABLE 6

Summary of results for shock capturing discretization for Problem 2 with various meshes, $\varepsilon = 10^{-6}$. NS, GN and FE refer to numbers of Newton steps, GMRES steps and function evaluations, respectively. Results marked "$(\cdot)$" failed during backtracking.*

h	Discretization		Preconditioner						
			I	HJ	VJ	HGS	VGS	IB_1	IB_2
1/16	SC	NS	12	23	16	10	14	11	11
		GS	947	2558	1061	511	337	248	332
		FE	1035	2917	1228	567	467	316	400
	SC/CD	NS	9	14	14	12	9	9	8
		GS	517	972	1038	583	107	142	163
		FE	539	1084	1135	663	145	190	190
1/32	SC	NS	22	24	33	25	19	14	15
		GS	5238	10345	4826	5738	863	555	1205
		FE	5571	5124	11310	6108	1128	672	1341
	SC/CD	NS	15	19	14	20	10	12	9
		GS	2370	2438	1535	3165	202	324	376
		FE	2534	2686	1642	3418	255	411	418

for linear discretization: inclusion of crosswind–diffusion leads to problems that are easier to solve, and "flow following" preconditioners (VGS and IB_1) tend to be most effective. It is clear, however, that the nonlinear discretizations lead to much more difficult problems than the linear ones, requiring many more GMRES steps to satisfy similar stopping criteria on common meshes. We will include solution accuracy in our considerations in the next section.

We comment on the convergence rate of the Newton–GMRES solver. As observed in Section 4.4, the nonlinear function $\mathcal{F}(\mathbf{u})$ is not differentiable. In some tests with a stringent forcing tolerance, $\eta = 10^{-8}$, we observed a linear convergence rate and not the quadratic rate achievable for smooth functions. This accounts for the relatively large number of Newton steps required for solution. We tested several other nonlinear solvers, including Broyden's method [6], [12, p. 195] and integration of a transient problem to steady state using a forward Euler method, and found them to be both slower and less robust (see [34]). We also note that the nonlinearity in these problems is due exclusively to the (nondifferentiable) discretizations. Cf. [15] for results with similar discontinuous discretizations applied to the Euler equations; for this nonlinear problem, Newton's method converges more rapidly.

6.3. Solution quality. For both benchmark problems, the steep gradients in both internal layers and boundary layers correspond to changes in function values $u \approx 0$ and $u \approx 1$. We can measure the width of the internal layer using

$$y_u = \min_{0\le y\le 1}\left\{y \mid u^h(x_i,y) \ge \vartheta_i\right\},\;\; y_l = \max_{0\le y\le 1}\left\{y \mid u^h(x_i,y) \le 1-\vartheta_i\right\}, \quad \text{for small } \vartheta_i > 0.$$

That is $\Delta y = y_u - y_l$ is a measure of the width of the numerical internal layer and the effect of crosswind smearing at $x = x_i$. Similarly,

$$\Delta x = 1 - \max_{0<x<1}\left\{x \mid |u^h(x,y_b)| \le \vartheta_b\right\},$$

is a measure of the width of the numerical boundary layer at $y = y_b$. We will use these quantities to specify the accuracy of the discrete solutions. In particular, for Problem 1, let $x_i = 1/2$ (so that we are measuring crosswind smear at the midpoint of the internal layer), and let $\vartheta_i = 10^{-3}$. For this problem with the choices of parameters via (7), (15)–(16), (26), and (29), the accuracy of the layer is restricted by the mesh size, i.e., $\Delta x = h$ for all h and discretizations considered (see Figure 3). That is, this numerical boundary layer provides no useful information. Therefore, we restrict our attention to the internal layer for this problem. For any discretization, we can then find the largest mesh parameter h such that $\Delta y \le 0.2$, and then examine the cost of solving each problem to within this specified accuracy.[4] For Problem 2, we use both criteria with $x_i = 0$, $\vartheta_i = 10^{-2}$, $y_b = 0.2$, $\vartheta_b = 10^{-2}$.

Figure 6 plots the width of crosswind smear in Problem 1 for various mesh sizes. These results indicate that solutions obtained from the nonlinear shock capturing discretizations exhibit considerably less smearing than for the linear schemes, and that MSD/CD is the most effective among the linear strategies.

The required mesh sizes for various choices of ε and both benchmark problems are shown in Table 7. These results indicate that when ε is small, the nonlinear discretizations require considerably less resolution to achieve accuracy. (Note that the mesh sizes in Table 7 are not increasing in a regular manner as ε increases; we believe this is due to discontinuities in this dependence, as evidenced by the nonsmooth curves in Figure 6.) Now, identification of the cost of solving each discrete problem on the mesh determined by the entries of Table 7 gives an indication of the costs to compute solutions of similar accuracy. Here, we have observed that the stopping criteria for the iterative solvers used in Section 6.1 are too stringent, in the sense that the accuracy requirements used here are satisfied for less accurate discrete solutions. Therefore, in these tests, the stopping criteria are

[4] Computations with fine meshes indicate that the true layers are actually much narrower than those quantities, so that we are examining numerical effects here. See Figure 6. The choice $\Delta y = 0.2$ was determined from the value of Δy for SC on a 25×25 grid for Problem 1 with $\varepsilon = 10^{-6}$.

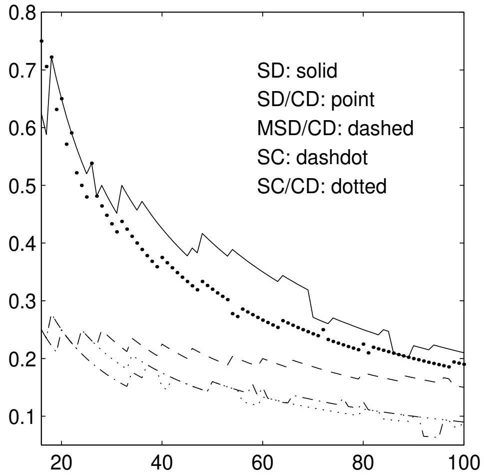


FIG. 6. *Crosswind smear width (Δy) when mesh is refined for Problem 1 with $\varepsilon = 10^{-6}, \theta = 10^{o}$ and $\vartheta_i = 10^{-3}$.*

TABLE 7
The required mesh sizes h when setting $\Delta y \leq 0.2$ at $x_i = 0.5$ in Problem 1, and $\Delta y \leq 0.2$ at $x_i = 0$ and $\Delta x \leq 0.1$ at $y_j = 0.1$ in Problem 2.

		Discretization				
	ε	SD	SD/CD	MSD/CD	SC	SC/CD
Pb. 1	10^{-6}	1/102	1/90	1/55	1/25	1/30
	10^{-5}	1/90	1/86	1/55	1/34	1/31
	10^{-4}	1/66	1/86	1/51	1/35	1/30
	10^{-3}	1/30	1/87	1/51	1/30	1/35
Pb. 2	10^{-6}	<1/110	1/90	1/75	1/21	1/30
	10^{-5}	<1/110	1/90	1/65	1/25	1/30
	10^{-4}	1/60	1/90	1/65	1/25	1/30
	10^{-3}	1/25	<1/110	1/86	1/20	1/24

1. the accuracy requirements

$$\begin{cases} \Delta y(x_i) \leq 0.2 & \text{for Problem 1} \\ \Delta y(x_i) \leq 0.2, \Delta x(y_b) \leq 0.1 & \text{for Problem 2} \end{cases}$$

2. decrease in the residual norm [25, p. 146]

$$\begin{cases} \|b - \mathcal{A}\mathbf{u}_k\| \leq \tau_a + \tau_r\|b\| & \text{for the linear discretizations} \\ \|\mathcal{F}(\mathbf{u}_k)\| \leq \tau_a + \tau_r\|b\| & \text{for the nonlinear discretizations,} \end{cases}$$

where $\tau_a = \tau_r = 0.1h^2$. The latter criterion ensures that the discrete solution achieves accuracy comparable to the truncation error of the discretization, and it is less stringent than that used in the previous section. We find the second criterion takes longer to be satisfied. We then solved each problem using a "good" solution algorithm, as determined by the results of the previous section; that is, we restrict our attention to the block incomplete factorization preconditioner IB_1. The results of these tests are shown in Tables 8 and 9. The results indicate that when ε is small, despite the fact that the nonlinear discretizations use fewer mesh points, the overall solution costs are higher because of slow convergence of the solution algorithms. These results also demonstrate the robustness and stability of the MSD/CD discretization over a variety of values of ε. The solution costs for SD discretization are large for small ε, although this method is effective for larger values of ε.

TABLE 8

The required iterations (GMRES steps) when setting $\Delta y \le 0.2$ at $x_i = 0.5$ in Problem 1, $\Delta y = 0.2$ at $x_i = 0$ and $\Delta x = 0.1$ at $y_b = 0.1$ in Problem 2. "(.)" refers to numbers of nonlinear function evaluations.

		Discretization				
	ε	SD	SD/CD	MSD/CD	SC	SC/CD
Pb. 1	10^{-6}	18	7	6	21 (54)	23 (56)
	10^{-5}	18	11	5	23 (107)	18 (74)
	10^{-4}	14	10	5	21 (102)	17 (94)
	10^{-3}	5	5	5	11 (54)	13 (65)
Pb. 2	10^{-6}	31	11	5	66 (83)	63 (90)
	10^{-5}	23	11	5	58 (85)	67 (94)
	10^{-4}	14	10	5	166 (220)	55 (82)
	10^{-3}	7	14	9	14 (28)	7 (16)

TABLE 9

The required operations (Mflops) when setting $\Delta y = 0.2$ at $x_i = 0.5$ in Problem 1, $\Delta y = 0.2$ at $x_i = 0$ and $\Delta x = 0.1$ at $y_b = 0.1$ in Problem 2.

		Discretization				
	ε	SD	SD/CD	MSD/CD	SC	SC/CD
Pb. 1	10^{-6}	8.03	1.96	0.53	3.08	5.62
	10^{-5}	4.46	1.27	0.51	8.49	6.94
	10^{-4}	1.57	1.27	0.44	8.35	6.51
	10^{-3}	0.15	1.30	0.51	3.68	6.90
Pb. 2	10^{-6}	>84.7	6.70	1.97	17.7	45.0
	10^{-5}	>26.8	6.70	1.48	26.0	26.8
	10^{-4}	4.02	5.98	1.48	14.1	23.1
	10^{-3}	0.30	>13.56	4.82	4.17	4.83

REFERENCES

[1] O. Axelsson. On the numerical solution of convection dominated convection-diffusion problems. In K. I. Gross, editor, *Mathematics Methods in Energy Research*, pages 3–21. SIAM, Philadelphia, 1984.

[2] O. Axelsson. Incomplete block-matrix factorization preconditioning methods. the ultimate answer? *J. Comp. Appl. Math.*, 12/13:3–18, 1985.

[3] O. Axelsson, V. Eijkhout, B. Polman, and P. Vassilevski. Incomplete block-matrix factorization iterative methods for convection-diffusion problems. *BIT*, 29:867–889, 1989.

[4] A. Brooks and T. J. R. Hughes. Streamline upwind/Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations. *Comp. Meth. Appl. Mech. Engng.*, 32:199–259, 1982.

[5] P. N. Brown and Y. Saad. Hybrid krylov methods for nonlinear systems of equations. *SIAM J. Sci. Stat. Comp.*, 11:450–481, 1990.

[6] C. G. Broyden. A class of methods for solving nonlinear simultaneous equations. *Math. Comput.*, 19:577–593, 1965.

[7] T. F. Chan and K. R. Jackson. Nonlinear precondtioned Krylov space methods for discrete Newton algorithms. *SIAM J. Sci. Stat. Comp.*, 5:533–542, 1984.

[8] R. Codina. A discontinuity-capturing crosswind-dissipation for the finite element solution of the convection-diffusion equation. *Comp. Meths. Appl. Mech. Engrg.*, 110:325–342, 1993.

[9] P. Concus, G. H. Golub, and G. Meurant. Block preconditioning for the conjugate gradient method. *SIAM J. Sci. Stat. Comput.*, 6:220–252, 1985.

[10] P.M. de Zeeuw. Matrix-dependent prolongations and restrictions in a blockbox multigrid solver. *J. Comput. Appl. Math.*, 33:1–27, 1990.

[11] R. S. Dembo, S. C. Eisenstat, and T. Steihaug. Inexact Newton methods. *SIAM J. Numer. Anal.*, 19:400–408, 1982.

[12] D. E. Dennis, Jr. and R.B. Schnabel. *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. Prentice-Hall, Englewood Cliffs, 1983.

[13] W. Eckhaus. Boundary layers in linear elliptic singular perturbation problems. *SIAM Review*, 14:225–270, 1972.

[14] S. C. Eisenstat and H. F. Walker. Choosing the forcing terms in an exact Newton's method. *SIAM J. Sci. Comput.*, 17:16–32, 1996.

[15] D. Feng and T. H. Pulliam. Tensor–GMRES methods for large systems of nonlinear equations. *SIAM J. Optim.*, 7:757–779, 1997.

[16] B. Fischer, A. Ramage, D. Silvester, and A.J. Wathen. Towards parameter-free streamline upwinding for advection-diffusion problems. Technical Report 37, Department of Mathematics, University of Strathclyde, 1996.

[17] A. C. Galeão and E. G. Dutra Do Carmo. A consistent approximate upwind Petrov–Galerkin methods for convection-dominated problems. *Comput. Methods Appl. Mech. Engng.*, 68:83–95, 1988.

[18] P. M. Gresho and R. L. Lee. Don't suppress the wiggles – they're telling you something. *Computers and Fluids*, 9:223–253, 1981.

[19] P. W. Hemker. Mixed defect correction iteration for the accurate solution of the convection diffusion equation. In W. Hackbusch and U. Trottenberg, editors, *Multi-grid Methods*, pages 485–501. Springer-Verlag, Berlin, 1982.

[20] T. J. R. Hughes and A. Brooks. A multidimensional upwind scheme with no crosswind diffusion. In T. J. R. Hughes, editor, *Finite Element Methods for Convection Dominated Flows*. AMSE, New York, 1979.

[21] T. J. R. Hughes, M. Mallet, and A. Mizukami. A new finite element formulation for computational fluid dynamics: II. Beyond SUPG. *Comp. Meths. Appl. Mech. Engrg.*, 54:341–355, 1986.

[22] C. Johnson. *Numerical Solution of Partial Differential Equations by the Finite Element Method*. Cambridge University Press, New York, 1987.

[23] C. Johnson, A.H. Schatz, and L.B. Wahlbin. Crosswind smear and pointwise errors in streamline diffusion finite element methods. *Math. Comp.*, 49:25–38, 1987.

[24] C. Johnson, A. Szepessy, and P. Hansbo. On the convergence of shock-capturing streimline diffusion finite element methods for hyperbolic conservation laws. *Math. Comp.*, 54:107–130, 1990.

[25] C. T. Kelley. *Iterative Methods for Linear and Nonlinear Equations*. SIAM, Philadelphia, 1995.

[26] G. Lube. An asymptotically fitted finite element method for convection dominated convection-diffusion-reaction problems. *Math. Mech.*, 72:189–200, 1992.

[27] K. W. Morton. *Numerical Solution of Convection-Diffusion Problems*. Chapman & Hall, London, 1996.

[28] U. Nävert. *A Finite Element Method for Convection-Diffusion Problems*. PhD thesis, Chalmers University of Technology, 1982.

[29] A. Quarteroni and A. Valli. *Numerical Approximation of Partial Differential Equations*. Springer-Verlag, New York, 1994.

[30] A. A. Reusken. Fourier analysis of a robust multigrid method for two–dimensional convection–diffusion equations. *Numer. Math.*, 71:365–398, 1995.

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

[32] Y. Saad and M. H. Schultz. Conjugate gradient-like algorithms for solving nonsymmetric linear systems. *Math. Comp.*, 44:417–424, 1985.

[33] Y. Saad and M. H. Schultz. GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM J. Sci. Stat. Comput.*, 7:856–869, 1986.

[34] Y. Shih. *Upwind Finite Element Solutions for Convection-Diffusion Problems*. PhD thesis, University of Maryland, College Park, Interdisciplinary Applied Mathematics Program, 1998.

[35] Y. Shih and H. C. Elman. Modified streamline diffusion schemes for the convection–diffusion problems.

Technical Report CS-TR-3835, University of Maryland Institute for Advanced Computer Studies, 1997. To appear in *Comp. Meth. Appl. Mech. Engng.*

[36] R. M. Smith and A. G. Hutton. The numerical treatment of advection – a performance comparison of current methods. *Numer. Heat Transfer*, 5:439–461, 1982.

[37] M. Stynes and L. Tobiska. Necessary L^2-uniform convergence conditions for difference schemes for two dimensional convection-diffusion problems. *Computers Math. Applic.*, 29:45–53, 1995.

[38] A. Szepessy. *Convergence of a Finite Element Method for Hyperbolic Conservation Laws*. PhD thesis, Chalmers University of Technology, 1989.

[39] A. Szepessy. Convergence of a shock-capturing streamline diffusion finite element method for a scalar conservation law in two dimensions. *Math. Comp.*, 53:527–545, 1989.

[40] G. Zhou and R. Rannacher. Pointwise superconvergence of the streamline diffusion finite element method. *Numer. Methods Partial Diff. Equations*, 12:123–145, 1996.