A truncated ILU smoother for multigrid preconditioning of convection dominated flow problems

Rees, Glyn and Silvester, David and Mihajlovic, Milan

2011

MIMS EPrint: **2011.34**

Manchester Institute for Mathematical Sciences
School of Mathematics

The University of Manchester

Reports available from: http://eprints.maths.manchester.ac.uk/
And by contacting: The MIMS Secretary
School of Mathematics
The University of Manchester
Manchester, M13 9PL, UK

ISSN 1749-9097

A truncated ILU smoother for multigrid preconditioning of convection dominated flow problems

G.O. Rees*
School of Computer Science, University of Manchester,
Manchester M13 9PL, UK.
e-mail: reesg2@yahoo.co.uk

D.J. Silvester
School of Mathematics, University of Manchester,
Manchester M13 9PL, UK.
e-mail: david.silvester@manchester.ac.uk

M.D. Mihajlović
School of Computer Science, University of Manchester,
Manchester M13 9PL, UK.
e-mail: milan@cs.man.ac.uk

April 21, 2011

Abstract

Multigrid methods are known to be efficient preconditioners and solvers for linear systems obtained from discretizing second-order, scalar elliptic problems. Singular perturbations involving these problems (such as the convection-diffusion equation) introduce new properties into the discrete problem, and this typically leads to the deterioration in the effectiveness of multigrid methods using standard point smoothers when close to the perturbation limit. In this paper we propose a new smoothing strategy, based on incomplete factorisation of truncated matrices arizing in the multigrid hierarchy. The truncation procedure is based on the heuristics used to determine strong connections in the classical (Ruge-Stüben) algebraic multigrid method. We report results of tests of the new smoother both for geometric and for algebraic multigrid on benchmark problems in two and three spatial dimensions.

Keywords: convection-diffusion, preconditioning, ILU, multigrid, Krylov methods.

1 Convection-diffusion problems

This paper is about fast solvers for the convection-diffusion equation. This is a fundamental model for a number of important physical processes and phenomena in fluid mechanics and electronics. It can arise either as a stand-alone (scalar) problem, or as a part of more complex systems of partial differential equations (such as the Navier-Stokes equations, the Boussinesq equations, or the drift-diffusion equations in semiconductor modelling). The basic problem is: given a convection field $\vec{w} : \mathbb{R}^d \mapsto \mathbb{R}^d$ so that $\nabla \cdot \vec{w} = 0$ (the "wind") and a diffusion parameter $\varepsilon > 0$, we want to find the scalar function $u(\vec{x})$ that satisfies

$$-\varepsilon \nabla^2 u + \vec{w} \cdot \nabla u = 0 \qquad \text{in } \Omega \subset \mathbb{R}^d, \tag{1.1}$$

*This research was supported by the EPSRC grant no. EP/C534875/1.

subject to a suitable set of boundary conditions (BCs). The most general form of the BC is given by

$$\beta_1 u + \beta_2 \frac{\partial u}{\partial n} = g \qquad \text{on } \partial\Omega, \tag{1.2}$$

where $\frac{\partial u}{\partial n}$ denotes the normal derivative (the flux) through the domain boundary $\partial\Omega$. In this paper we restrict attention to the case when $\beta_2 = 0$ (i.e. $u = g$ on $\partial\Omega$). The equation (1.1) is usually non-dimensionalised with respect to the characteristic domain length scale L and the magnitude of the wind W ($\vec{w} = W\vec{w}_*$, $\|\vec{w}_*\| = 1$) to give the equivalent problem

$$-\nabla^2 u + \mathrm{Pe}\ \vec{w}_* \cdot \nabla u = 0 \quad \text{in } \Omega_\star, \qquad u = g \quad \text{on } \partial\Omega_\star. \tag{1.3}$$

Here, $\Omega_\star$ is a reference domain (with coordinates scaled by L) and $\mathrm{Pe} = \frac{WL}{\varepsilon} > 0$ is a non-dimensionalised scalar parameter known as the Peclet number. The Peclet number determines the relative contributions of convection and diffusion in (1.3), see e.g. [7, p. 115]. The convection-dominated case when $\mathrm{Pe} \gg 1$ is of primary interest here.

To assure numerical stability when approximating the problem (1.3), the discretization $\mathcal{T}_h$ of Ω needs to be refined (or stretched) in the parts of the domain where rapid changes in the solution occur, see [7, Ch. 3], or [9, pp. 203–206]. If solution features are not resolved (for example on the successively coarser grids generated using multigrid) then some form of stabilisation must be applied to the discrete problem. We will adopt such a strategy in a finite element method (FEM) setting in this work. To this end, we define the solution space $H^1_E(\Omega) = \{u \,|\, u \in H^1(\Omega),\ u = g \text{ on } \partial\Omega\}$, and construct finite element approximation spaces $H^1_0(\Omega) \supset S^h_0 = \mathrm{span}\{\phi_i\}_{i=1}^n$ and $H^1_E(\Omega) \supset S^h_E = \mathrm{span}\{\phi_j\}_{j=1}^{n+n_\partial}$. Our stabilized discretization of the dimensional version of (1.3) can then be stated as follows: find $u_h \in S^h_E$ such that

$$\begin{aligned} \varepsilon \int_\Omega \nabla u_h \cdot \nabla v_h \, d\Omega + \int_\Omega (\vec{w} \cdot \nabla u_h) v_h \, d\Omega \quad &+ \quad \sum_{K \in \mathcal{T}_h} \delta_K \int_K (\vec{w} \cdot \nabla u_h)(\vec{w} \cdot \nabla v_h) \, dK \\ &= \quad \varepsilon \sum_{K \in \mathcal{T}_h} \delta_K \int_K (\nabla^2 u_h)(\vec{w} \cdot \nabla v_h) \, dK \qquad \forall v_h \in S^h_0. \end{aligned} \tag{1.4}$$

The function u_h satisfying (1.4) will be referred to as the SUPG FEM solution in the sequel. Note that in the case of bilinear/trilinear (Q_1) approximation on rectangles/bricks the consistency term on the right-hand side of (1.4) is identically zero. To implement (1.4) we also have to specify the local stabilisation parameters δ_K. Herein we follow the construction[1] given in [7, p. 132] so that

$$\delta_K = \begin{cases} \frac{h_k}{2|\vec{w}_k|} \left(1 - \frac{1}{\mathrm{Pe}^k_h}\right) & \text{if } \mathrm{Pe}^k_h > 1 \\ 0 & \text{if } \mathrm{Pe}^k_h \le 1 \,. \end{cases} \tag{1.5}$$

Here, $|\vec{w}_k|$ is the ℓ_2 norm of the wind at the element centroid, h_k is a measure of the element length in the direction of the wind, and $\mathrm{Pe}^k_h := |\vec{w}_k| h_k / (2\varepsilon)$ is the element Peclet number.

In the case of Q_1 approximation the linear algebra system corresponding to (1.4) has the following structure,

$$(\varepsilon A + C + S)\, \bar{x} = \bar{f} \,. \tag{1.6}$$

In the above, A is the standard Galerkin diffusion matrix $a_{ij} = (\nabla\phi_j, \nabla\phi_i)$ and is symmetric and positive definite; C is the convection matrix $c_{ij} = (\vec{w}\cdot\nabla\phi_j, \phi_i)$, and is skew-symmetric (under the condition $\nabla\cdot\vec{w} = 0$); S is the streamline-diffusion matrix $s_{ij} = \sum_{K\in\mathcal{T}_h} \delta_k (\vec{w}\cdot\nabla\phi_j, \vec{w}\cdot\nabla\phi_i)_K$ and is symmetric semi-definite. The overall coefficient matrix $\Phi = \varepsilon A + C + S$ is not symmetric and is not guaranteed to be diagonally dominant for large values of $\mathrm{Pe} \sim 1/\varepsilon$.

[1] This specification has been extensively tested and is embodied in our IFISS software package [22].

2 Multigrid solvers and multigrid preconditioners

Our aim is to study efficient iterative solver strategies for linear algebra systems of the form (1.6). Throughout this paper we use the term *robust* if an iterative solver exhibits consistent performance (measured in terms of the numbers of iterations), irrespectively of the discretisation parameter h, and independent of the value of Pe and structure of $\vec{w}$. An iterative solver is regarded as being *black-box* if its performance does not depend on any iteration parameters, or if one can provide a priori parameter values that make the solver robust for a wide class of problems. An iterative solver is said to be *optimal* if its execution time and storage requirements scale linearly ($O(n)$) with respect to the problem size.

Single grid iterative methods, such as fixed-point iteration, see [21, Ch. 4], are not robust in general, even when combined with incomplete factorisation—such methods are not able to reduce consistently and rapidly the entire discrete Fourier spectrum of the solution error, see [3, pp. 13–26]. Robustness typically requires a multilevel approach, where a nested sequence of progressively finer grids, discrete representations of the continuous problem on these grids, and the means of communication between these grids are needed to construct an effective preconditioning/solution scheme. To fix our nomenclature: any such multilevel iterative scheme that involves full coarsening, direct discretisation of the continuous problem at coarse levels, point smoother, and full weighting interpolation is called a standard geometric multigrid (GMG) method. Finding a robust optimal solver for discrete diffusion problems (Pe = 0) is a completely straightforward exercise—the Conjugate Gradient method with a single V-cycle of GMG as preconditioner is a simple and effective strategy.

Standard multigrid methods are not robust when applied to discretized convection-diffusion problems, especially for large values of Pe and/or when $\vec{w}$ has a complicated structure (see [7, p. 194]). To address this issue, two enhancements to standard GMG have been proposed in the literature: modifying the interpolation procedure [4],[8],[19],[31], and/or modifying the relaxation procedure [1],[17],[4],[8],[10],[14],[16],[19],[27]. In the latter case effort needs to be be put into reordering the unknowns to take account of the direction of the wind. For example, in [5] the authors study the effect of different nodal orderings on the performance of Gauss-Seidel smoothing when applied to a one-dimensional convection-diffusion problem. The results demonstrate a clear advantage of downwind ordering. Further modifications of standard smoothing techniques for convection-dominated cases include block Gauss-Seidel methods [16], line Gauss-Seidel smoothing [19], and black-box downwind reordering of the unknowns [10]. The strategy of downwind nodal ordering is extended in [1] to cover GMG preconditioning of convection problems discretised with adaptive grids. Further extensions of this concept can be found in [17]. In [10] a block Gauss-Seidel smoother is considered. To allow easier blocking of the coefficient matrix, truncation of the convection part of the matrix Φ is introduced (the truncation procedure is based on the magnitude of the elements, but no specific algorithm or criterion is presented). In [27] a block Gauss-Seidel solver with blocks ordered in a cross-wind direction is studied for a specific discretisation of the convection-diffusion problem. In [31] the efficiency study of ordered and line Gauss-Seidel smoothers for GMG preconditioning of two model problems from [7] is presented.

As noted above, unstabilized FEM may produce oscillatory solutions of convection-dominated problems on coarse grids even when all the features of the exact solution are properly resolved on the finest grid. This is a key issue for any GMG preconditioning strategy as the defect equations on the coarse grids do not produce desired corrections to the solution. This issue can be addressed by the use of a (stable) operator-dependent interpolation, [32], or by following [19], and using the stabilised approximation method (1.4)–(1.5) when discretising coarse level operators. Numerical results in [19] show that if this strategy is combined with a two-directional line Gauss-Seidel smoother then a robust GMG preconditioning strategy is obtained.

This is the motivation for this paper. The objective is to build on the GMG method in [19] and to try to construct an equally robust *algebraic multigrid* (AMG) preconditioning strategy. In AMG only a linear system coefficient matrix is required as an input, and its coarse representations are then constructed in an automatic coarsening procedure, based on certain heuristic principles. For a more complete discussion see [25, App. A]. We will restrict attention to so-called "classical AMG" with Ruge-Stüben coarsening in this work. The challenge that must be faced is that robust smoother strategies for convection-dominated problems are very difficult to implement within an AMG setting. Robust smoothers developed for convection-dominated problems usually require explicit knowledge of geometric information. Even when such information is available,

it is a non-trivial task to implement line/block smoothers efficiently (e.g. in cases when unstructured or adaptive grids are used). In addition the use of black-box algorithms to compute downwind ordering (such as Tarjan's algorithm [24]) may be prohibitively expensive, especially in three dimensions [20]. Incomplete factorisations (with or without fill-in, see [21, p. 288]) are the obvious alternative to line smoothing. The appeal of using an ILU_0 smoothing for MG preconditioning comes from its effectiveness as a solver/preconditioner in highly convective cases, see [29], [30], and [26, p. 216]. Our computational experience confirms that ILU_0 smoothing is very robust for convection dominated problems. Standard two-grid Fourier analysis in [11] provides theoretical justification for this assertion. The results in Section 4 also suggest however, that there may be a simpler, more cost-effective alternative to ILU_0 smoothing. The new approach is presented in the next section.

3 A truncated incomplete factorisation smoothing strategy

The finite difference stencils in [7, pp. 152–154]) highlight the h-dependence of the matrix C and the h-independence of A in the system (1.6). Thus, the relative contribution of C inside Φ increases when h increases (i.e. on coarser grids). This implies a loss of the diagonal dominance in Φ as $h \to \infty$ which makes standard point smoothers progressively more ineffective on coarser grids. This intuition also suggests, however, that a computationally cheap smoother can be sufficient at fine levels, and a robust (and computationally more expensive) smoother is needed only at a few coarsest levels. In practice it not going to be possible to determine the cut-off level in the grid hierarchy when a switch will be beneficial. Thus, instead, we propose to use a variable smoother which adapts itself to the local strength and characteristics of the convective field, utilising the matrix stencils at each grid node. The variability is obtained by a static analysis of the matrix elements at each grid level before the assembly and the application of the smoother. This analysis gives the information about the "relevant" off-diagonal entries in each matrix row, thus capturing additional local information about the convective field that may be missed by the standard point smoothers. In our implementation we adopt a truncation criterion based on the heuristic from classical AMG that quantifies the strength of dependence. Specifically, our modification consists in keeping a priori the diagonal entry in each row together with all off-diagonal entries that satisfy the simple dropping test:

$$|\phi_{ij}| > \alpha \max_k |\phi_{ik}| \qquad \alpha \in [0,1]. \tag{3.7}$$

The level of truncation is thus controlled by the parameter α. The smoother is then defined by applying the ILU_0 method to the truncated matrices at all MG levels. If we also introduce a damping parameter $\gamma \in (0,1]$ we can completely characterise our method using the notation $\mathrm{tILU}_0(\gamma,\alpha)$. In the two extreme cases our approach reduces to a damped ILU_0 method (for $\alpha = 0$) and damped Jacobi method (for $\alpha = 1$). The method is thus a trade-off between a computationally cheap, but ineffective (for dominant convection) Jacobi smoother, and a robust, but computationally expensive ILU_0 smoother. The new smoother adapts to the properties of the matrix stencils of Φ, typically truncating the majority of the off-diagonal entries at the finest levels, whilst keeping progressively more off-diagonal elements at coarse levels (where the application of the ILU_0 method is not prohibitively expensive). The proportion of the off-diagonal elements kept at a particular level also depends on the structure and the strength of the convection field—more entries are retained when Pe is increased.

The standard analytical procedure for estimating the contraction factors of different components of the algebraic error when an iterative solver is applied is discrete Fourier analysis, see [28, Ch. 7]. If a general splitting $\Phi = M - N$ is introduced, then the errors in two successive iterations of $\bar{x}^{[k+1]} = \bar{x}^{[k]} + M^{-1}\bar{r}^{[k]}$ satisfy $\bar{e}^{[k+1]} = E\bar{e}^{[k]}$, where $E = I - M^{-1}N$. The eigenvalue problem $E\bar{y}(\theta) = \lambda(\theta)\bar{y}(\theta)$ associated with the iteration matrix has the eigenvectors $\varphi_{j_1,j_2} = e^{i(j_1\theta_1 + j_2\theta_2)}$ ($i = \sqrt{-1}$), and θ_1, θ_2 are the points in the discrete Fourier space $\Theta(\theta) = [-\pi,\pi]^2$. The eigenvalues $\lambda(\theta)$ can be shown (see [28, p. 113], or [23, p. 14]) to satisfy

$$\lambda(\theta) = \frac{\sum_j N(j) e^{i(j_1\theta_1 + j_2\theta_2)}}{\sum_j M(j) e^{i(j_1\theta_1 + j_2\theta_2)}}, \qquad \theta \in \Theta, \tag{3.8}$$

where $j = (j_1, j_2)$ and $\theta = (\theta_1, \theta_2)$. The discrete Fourier space is subdivided into two subspaces: that of smooth eigenfunctions $\Theta_S(\theta) = \Theta(\theta) \cap \left(-\frac{\pi}{2}, \frac{\pi}{2}\right)^2$ and oscillatory eigenfunctions $\Theta_R(\theta) = \Theta(\theta) \setminus \Theta_S(\theta)$.

Fourier analysis is possible in cases when the coefficient matrix stencils are constant. In the case of a convection-diffusion problem this requirement is fulfilled only in the case of a constant, uni-directional wind. The results presented here are obtained for the problem with $\Omega = [-1, 1]^2$, $\vec{w} = (-1, 0)$, with BCs $u = 0$ on $\partial\Omega$. The Q_1 SUPG FEM discretisation on a uniform Cartesian grid with lexicographical node ordering in the negative x direction results in a constant 9-point stencil, see [7, p. 154]). The eigenvalues (3.8) for the tILU(γ, α) method are obtained exactly as in the ILU$_0$ case [28, p. 142], but with a static truncation that modifies the stencils of M. If $N_t = (LU)_t - \Phi_t$ and all statically truncated elements are $N_t^* = \Phi - \Phi_t$, then we have that $N = N_t + N_t^*$ and the eigenvalues (3.8) are given by

$$\lambda(\theta) = \frac{N}{N_t + (F_t + G_t + d)}, \tag{3.9}$$

where F_t and G_t are the strictly upper and the strictly lower triangular parts of Φ_t, respectively, and d is the diagonal entry. The eigenvalues $\lambda(\theta)$ may be determined analytically in the pure diffusion case, but for nonzero convection we use a numerical procedure to compute $\lambda(\theta)$. To this end, the Fourier space is covered by a grid of equidistant points spaced by h_θ and (3.9) is computed at each of the points. We report the results for $h_\theta = \frac{2\pi}{128}$, although we performed the computations for several grid sizes to check the convergence. In Table 3.1 we summarise the values $\sigma = |\lambda(\theta)|$ for different smoothers obtained for various values of $\mathrm{Pe} = 2/\varepsilon$.

Table 3.1: The maximum of the moduli $\sigma = \max|\lambda(\theta)|$ of the eigenvalues of the iteration matrix E for different smoothers applied to a constant wind convection-diffusion matrix of size $n = 3969$.

Pe	Gauss-Seidel	ILU$_0$	tILU$_0(1, 0.25)$	Jacobi
0	0.4203	0.1466	0.5	0.5
1000	0.5459	0.3822	0.5811	1.7510
4000	0.8455	0.9739	0.8811	1.9336

The smoothing factors in Table 3.1 are reported with no damping ($\gamma = 1$) and are in perfect agreement with previous results in [7, p. 100], [28, p. 124 & p. 143], and [23, p. 45]. Note the strong dependence of σ on the Peclet number for a fixed problem size for all smoother strategies. For $\mathrm{Pe} = 0$ the results for Jacobi and tILU$_0(1, 0.25)$ smoother are identical indicating that tILU$_0(1, 0.25)$ method reduces to Jacobi in this case. The values of σ for Gauss-Seidel and tILU$_0$ methods are comparable. It is also interesting to note that the tILU$_0$ method has a smaller contraction rate than ILU$_0$ for Pe=4000. In Fig. 3.1 we plot the discrete Fourier spectrum for four different smoothers for Pe=4000 and $n = 3969$. A visual comparison of the tILU$_0$ plot (c) with the corresponding plots for Jacobi and ILU$_0$ reveals the hybrid nature of the tILU$_0$ method.

4 Numerical experiments

In this section we present numerical results showing the performance and scalability of MG preconditioned GMRES with the new tILU$_0$ smoother. We report the GMRES iteration counts using right preconditioning and a residual reduction of 10^{-6}. The computations are performed using the `oomph-lib` software package [12], running on a dual core Intel Xeon CPU 3.60 GHz processor architecture equipped with 2 Gb of RAM. We tested both AMG and GMG, different types of grids (uniform and stretched), different node orderings and we also varied the structure of the convection field. A comparison of tILU$_0$ and other standard smoothing techniques (Jacobi, Gauss-Seidel, ILU$_0$) is our primary goal. In the experiments reported we use damped ILU$_0$, tILU$_0$, and Jacobi smoothing (all with $\gamma = 0.67$) and we compare with the standard (undamped) Gauss-Seidel smoother ($\gamma = 1$). These specific choices of γ were identified as being the "best choice" by extensive testing on a wide range of problems. Full details can be found in [20].

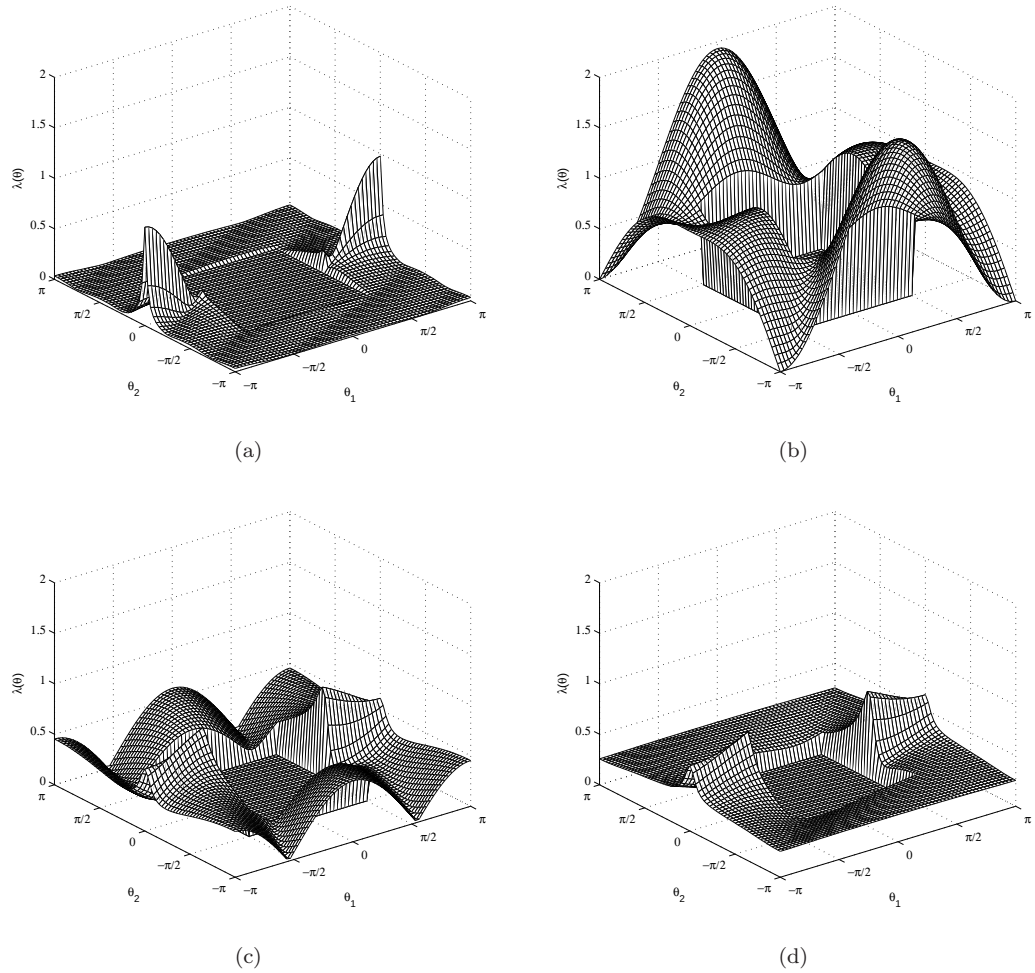

Figure 3.1: Plots of the eigenvalues $\lambda(\theta)$ (3.9) of the matrix E with $N = 3969$, Pe=4000 discretised by Q1 SUPG FEM. The Fourier space $[-\pi, \pi]^2$ is discretised by a uniformly spaced grid of points with $h_\theta = \frac{2\pi}{128}$. (a) ILU_0 smoother; (b) Jacobi smoother; (c) $\text{tILU}_0(1,0.25)$ smoother; (d) Gauss-Seidel smoother.

Example 4.1: Double glazing problem [7, p. 119]. This is a well-known benchmark problem that involves a difficult case of a recirculating wind. It is also a physically relevant problem since it models enclosed flow scenarios. The convection-diffusion problem (1.1) is posed over the domain $\Omega = [-1, 1]^2$, subject to Dirichlet BCs $u(x = 1, y \in [-1, 1]) = 1$ (hot wall) and $u = 0$ elsewhere on $\partial\Omega$. The convection field is given by $\vec{w} = (2y(1 - x^2), -2x(1 - y^2))$, thus the Peclet number is given by $\text{Pe} = 4/\varepsilon$. The problem is discretised using Q1 SUPG FEM on a uniform Cartesian product grid. We first study how different nodal orderings affect the performance of a GMG preconditioner. In this context we study two different ordering strategies: first, a natural tree-based ordering, which arises when a fine mesh is created by successive uniform or adaptive refinement of an initial coarse mesh (see [12]), and second, a four–directional lexicographical ordering (following [7, p. 190]). The convergence results are summarised in Table 4.2 as a function of the discrete problem size n and the Peclet number. From Table 4.2 (part a) we see that in the case of the natural ordering, GMG preconditioning with all four smoothers is robust with respect to the discrete problem size. Note however, that a larger value of Pe leads to a higher iteration count in every case. Looking at the finest grid column, the $\text{tILU}_0(0.67, 0.25)$ smoother gives the shortest computation time, but this is only marginally faster than the $\text{ILU}_0(0.67)$ and Gauss-Seidel smoother results. The $\text{tILU}_0(0.67, 0.25)$ and Gauss-Seidel methods both give comparable convergence when a multi-directional lexicographical nodal ordering

is used in the case of the smaller value of Pe. Note however, that the Gauss-Seidel smoother deteriorates significantly for Pe = 8000. The $\text{tILU}_0(0.67, 0.25)$ is the clear winner in this case, and seems to be the best preconditioning strategy overall.

Table 4.2: The iteration count and, in brackets, the total execution time (in seconds) for the GMRES method right-preconditioned by GMG with different smoothers as a function of the problem size n and Peclet number Pe for different nodal orderings. Example 4.1 is discretised by Q1 SUPG FEM on a sequence of uniform, Cartesian product grids. * denotes lack of convergence within 200 iterations.

a) Natural (tree-based) ordering, V(2,2) cycle

n	3969	16129	65025	261121	1046529
Pe=2000					
$\text{ILU}_0(0.67)$	8 (0.46)	8 (1.86)	7 (7.52)	7 (30.80)	7 (126.90)
$\text{tILU}_0(0.67, 0.25)$	15 (0.47)	15 (1.89)	13 (7.37)	12 (29.51)	11 (115.97)
Jacobi(0.67)	67 (0.86)	62 (3.35)	52 (12.62)	48 (50.75)	45 (200.97)
Gauss-Seidel	16 (0.45)	14 (1.86)	13 (7.09)	12 (28.78)	12 (117.71)
Pe=8000					
$\text{ILU}_0(0.67)$	15 (0.55)	15 (2.28)	13 (8.97)	12 (35.61)	12 (144.93)
$\text{tILU}_0(0.67, 0.25)$	36 (0.67)	33 (2.47)	31 (9.54)	28 (37.86)	26 (138.32)
Jacobi(0.67)	161 (1.99)	154 (7.37)	151 (25.66)	137 (101.23)	129 (381.28)
Gauss-Seidel	39 (0.61)	36 (2.41)	35 (9.46)	29 (36.16)	27 (146.25)

b) 4-directional lexicographical ordering, V(4,4) cycle

n	3969	16129	65025	261121	1046529
Pe=2000					
$\text{ILU}_0(0.67)$	6 (0.57)	5 (2.41)	5 (10.15)	5 (42.70)	5 (178.06)
$\text{tILU}_0(0.67, 0.25)$	9 (0.51)	9 (2.17)	8 (8.53)	8 (35.17)	7 (138.57)
Jacobi(0.67)	107 (1.09)	99 (4.05)	92 (15.60)	85 (61.65)	79 (241.01)
Gauss-Seidel	11 (0.46)	9 (1.91)	9 (8.05)	8 (33.13)	7 (129.87)
Pe=8000					
$\text{ILU}_0(0.67)$	10 (0.69)	16 (3.89)	22 (20.73)	22 (87.65)	22 (365.50)
$\text{tILU}_0(0.67, 0.25)$	22 (0.75)	20 (3.01)	17 (11.78)	16 (48.69)	16 (200.60)
Jacobi(0.67)	*	*	*	*	*
Gauss-Seidel	31 (0.64)	41 (3.16)	67 (18.81)	74 (86.81)	69 (319.29)

Next we study the performance of the GMG preconditioner with different smoothers for the case of stretched grids. The solution of Example 4.1 exhibits steep boundary layers near the walls $y = -1$ and $x = 1$. Accurate resolution of these layers requires us to concentrate the grid points close to the boundaries. To achieve this, a grid stretching procedure from `oomph-lib` was applied to an initial uniform grid. The iteration counts obtained with the different smoothers are summarised in Table 4.3. From Table 4.3 we see that both $\text{ILU}_0(0.67)$ and $\text{tILU}_0(0.67, 0.25)$ smoother lead to a GMG preconditioner that performs robustly with respect to the grid size n and exhibits moderate dependence on Pe (as $O(\sqrt{\text{Pe}})$). It is worth noting that the GMG preconditioner with the Gauss-Seidel smoother is not robust in the highly convective case.

Next we examine the performance of the AMG preconditioner. We use BoomerAMG [13] from the `Hypre` library with standard two-pass Ruge-Stüben coarsening. In this case we study only the default nodal ordering. In AMG regular coarsening patterns can only be expected for discrete Poisson equation obtained from regular, tensor-product grids. In the convection cases, the properties of the coarse level matrices can be considerably different than those of the original matrix, because classical AMG performs a variant of semi-coarsening in characteristic directions, see [2]. This produces comparably more coarse levels than when full coarsening is applied to the same problem. Another interesting feature of classical AMG coarsening is that it generates

Table 4.3: The iteration count and, in brackets, the execution time (in seconds) for the GMRES method right-preconditioned by GMG with different smoothers as a function of n and Pe. The preconditioner is a V(2,2) cycle with default (tree-based) ordering. Example 4.1 is discretised by Q1 SUPG FEM on a sequence of stretched Cartesian product grids (stretching parameter $a = 0.9$).

n	2209	9025	36481	146689
$h_{\min}$	0.005729	0.002474	0.001139	0.000545
$h_{\max}$	0.077604	0.039193	0.019694	0.009871
Pe=2000				
$\text{ILU}_0(0.67)$	8 (0.26)	7 (1.02)	6 (4.11)	6 (16.68)
$\text{tILU}_0(0.67, 0.25)$	9 (0.24)	8 (0.95)	8 (3.85)	7 (15.45)
Jacobi(0.67)	33 (0.34)	31 (1.32)	28 (5.17)	25 (20.55)
Gauss-Seidel	12 (0.24)	11 (0.94)	10 (3.82)	11 (15.70)
Pe=8000				
$\text{ILU}_0(0.67)$	13 (0.29)	13 (1.21)	12 (4.89)	11 (19.40)
$\text{tILU}_0(0.67, 0.25)$	17 (0.28)	16 (1.08)	15 (4.42)	13 (17.24)
Jacobi(0.67)	71 (0.54)	77 (2.25)	74 (8.45)	66 (32.77)
Gauss-Seidel	24 (0.28)	43 (1.42)	71 (7.90)	102 (44.86)

Table 4.4: The iteration count and, in brackets, the total execution time (in seconds) for the GMRES method right-preconditioned by AMG with different smoothers as a function of n and Pe. The preconditioner is a V(2,2) cycle with default (tree-based) nodal ordering. Example 4.1 is discretised by Q1 SUPG FEM on a sequence of uniform, Cartesian product grids. * denotes lack of convergence within 150 iterations.

n	3969	16129	65025	261121	1046529
Pe=8000					
$\text{ILU}_0(0.67)$	10 (0.14)	10 (0.60)	9 (2.56)	7 (10.11)	6 (40.04)
$\text{tILU}_0(0.5, 0.5)$	15 (0.12)	13 (0.48)	12 (1.99)	10 (8.02)	9 (32.88)
Jacobi(0.67)	20 (0.12)	16 (0.46)	13 (1.81)	10 (6.68)	10 (27.19)
Gauss-Seidel	17 (0.10)	16 (0.47)	14 (1.86)	10 (6.97)	10 (28.65)
Pe=40000					
$\text{ILU}_0(0.67)$	17 (0.18)	20 (0.92)	43 (7.80)	*	*
$\text{tILU}_0(0.67, 0.5)$	27 (0.17)	32 (0.88)	29 (3.68)	25 (14.98)	22 (60.22)
Jacobi(0.67)	41 (0.21)	48 (1.15)	41 (5.00)	33 (18.44)	28 (69.52)
Gauss-Seidel	36 (0.19)	51 (1.18)	70 (8.12)	*	*

coarse level matrices with larger stencils than the original matrix. By inspection it can be verified that a substantial proportion of the off-diagonal elements in coarse level matrices are small in magnitude. This makes them suitable for truncation by (3.7). Extensive testing reveals that tILU_0 smoother preserves its robustness (and consistently produces the shortest execution times) for $\alpha = 0.5$ and we adopt this value of the truncation parameter in all the AMG tests reported below.

Table 4.4 summarises the convergence characteristics of GMRES right-preconditioned by a V(2,2) cycle of AMG as a function of the problem size and the Peclet number. From Table 4.4 we see that for Pe=8000 all smoothers lead to an AMG preconditioner that is robust with respect to the grid size. The picture looks very different when Pe in increased to 40000 however. Whereas the $\text{ILU}_0(0.67)$ and Gauss-Seidel smoothers lose effectiveness, the damped Jacobi and $\text{tILU}_0(0.67, 0.5)$ smoothers are perfectly robust with respect to grid refinement.

Example 4.2: Generalised double glazing problem [18]. Next, we consider a three-dimensional problem where (1.1) is posed over the domain $\Omega = [0, 1]^3$ subject to BCs $u(x = 1, y \in [0, 1], z \in [0, 1]) = 1$

(hot wall) and $u = 0$ elsewhere on $\partial\Omega$. The convection field is given by $\vec{w} = (2x(1-x)(2y-1)z, -(2x-1)y(1-y), -(2x-1)(2y-1)z(1-z))$. An arrow plot of $\vec{w}$ is presented in Fig. 4.2. The Peclet number in this case is given by $\mathrm{Pe} = 1/(2\varepsilon)$.

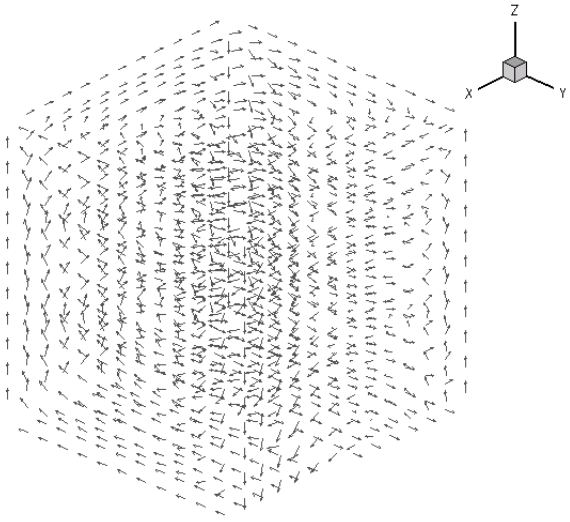

Figure 4.2: The structure of the convective field $\vec{w}$ for Example 4.2.

Here we report the convergence results for Q1 SUPG FEM discretisation of the problem on uniform grids (adaptively refined grid results can be found in [20]). Convergence results for two different nodal orderings are reported in Table 4.5. From Table 4.5, part a) we can conclude that all smoothers lead to an optimal preconditioner with respect to n. Good performance with respect to Pe is obtained when $\mathrm{ILU}_0(0.67)$ or $\mathrm{tILU}_0(0.67, 0.25)$ smoother is used. In contrast, when damped Jacobi is used as a smoother in the highly convective case, the preconditioned GMRES solver fails to converge within 200 iterations.

Next we examine the performance of AMG preconditioner. We consider only the case of natural nodal ordering. The main difficulty with the classical (two-pass) Ruge-Stüben coarsening is that it generates coarse-level operators with considerably larger stencils than that of the coefficient matrix (this is especially pronounced in three dimensions, see [2]). This makes the ILU_0 and Gauss-Seidel smoothers expensive. The convergence of the GMRES solver right-preconditioned by a V(2,2) cycle of AMG is reported in Table 4.6. Once again we see the $\mathrm{tILU}_0(0.67, 0.5)$ smoother is the best strategy in terms of overall computational cost.

To conclude this section, we would like to quantify the savings in computational effort arising from the truncation heuristic (3.7). A suitable measure for the truncation of matrices in the MG hierarchy is the *truncation ratio* defined by

$$\eta = \frac{\sum_{\ell=1}^{L} \mathrm{nnz}(\tilde{A}_\ell)}{\sum_{\ell=1}^{L} \mathrm{nnz}(A_\ell)}, \tag{4.10}$$

where $\tilde{A}_\ell$, $\ell = 1, \ldots, L$ are the truncated matrices at all levels.

In Table 4.7 we present the truncation statistics for a two-dimensional problem for two different values of Pe and four different values of α. The reported values are the numbers of retained non-zero elements in the entire grid hierarchy and, in brackets, the percentage of retained elements. Note that for $\alpha = 0$ we have the ILU_0 method and for $\alpha = 1$ the Jacobi method (thus, for a 9-point stencil $\eta \sim \frac{1}{9}$, i.e. 11%). The truncation statistics for the case of AMG coarsening are reported for a three-dimensional example in Table 4.8. Note that the stencil size of the original matrix is 27, so $\eta \sim 3.7\%$ in Jacobi's case when full coarsening is used. This value can be much smaller in the case of AMG coarsening, due to semi-coarsening and much larger stencils in

Table 4.5: The iteration count and, in brackets, the total execution time (in seconds) for the GMRES method right-preconditioned by GMG with different smoothers as a function of n and Pe for different nodal orderings. Example 4.2 is discretised by Q1 SUPG FEM on a sequence of uniform, Cartesian product grids. * denotes lack of convergence within 200 iterations.

a) Natural (tree-based) ordering, V(2,2) cycle

n	1331	12167	103823	857375
Pe=250				
$\mathrm{ILU}_0(0.67)$	5 (0.48)	5 (4.04)	5 (32.87)	5 (268.55)
$\mathrm{tILU}_0(0.67, 0.25)$	8 (0.45)	8 (3.50)	8 (27.77)	8 (222.99)
Jacobi(0.67)	13 (0.48)	14 (3.87)	12 (29.98)	12 (238.94)
Gauss-Seidel	6 (0.44)	6 (3.45)	6 (27.33)	6 (220.46)
Pe=1000				
$\mathrm{ILU}_0(0.67)$	7 (0.52)	7 (4.31)	7 (35.51)	7 (290.24)
$\mathrm{tILU}_0(0.67, 0.25)$	11 (0.46)	11 (3.62)	11 (28.63)	11 (230.34)
Jacobi(0.67)	45 (0.64)	*	*	*
Gauss-Seidel	10 (0.48)	11 (3.63)	10 (28.60)	10 (230.52)

b) 6-directional lexicographical ordering, V(6,6) cycle

n	1331	12167	103823	857375
Pe=250				
$\mathrm{ILU}_0(0.67)$	3 (0.67)	3 (6.10)	3 (53.07)	3 (464.59)
$\mathrm{tILU}_0(0.67, 0.25)$	4 (0.47)	5 (3.86)	5 (31.35)	5 (252.14)
Jacobi(0.67)	21 (0.49)	22 (3.84)	19 (29.92)	18 (236.27)
Gauss-Seidel	3 (0.46)	4 (3.87)	4 (31.43)	4 (254.25)
Pe=1000				
$\mathrm{ILU}_0(0.67)$	4 (0.74)	4 (6.77)	4 (59.32)	4 (512.05)
$\mathrm{tILU}_0(0.67, 0.25)$	6 (0.51)	6 (4.01)	6 (33.21)	6 (272.91)
Jacobi(0.67)	78 (0.69)	*	*	*
Gauss-Seidel	6 (0.48)	6 (3.96)	6 (32.38)	6 (264.04)

Table 4.6: The iteration count and, in brackets, the total execution time (in seconds) for the GMRES method right-preconditioned by AMG with different smoothers as a function of n and Pe. The preconditioner is a V(2,2) cycle with default (tree-based) nodal ordering. Example 4.2.1 is discretised by Q1 SUPG FEM on a sequence of uniform, Cartesian product grids.

n	1331	12167	103823	857375
Pe=1000				
$\mathrm{ILU}_0(0.67)$	4 (0.17)	4 (4.24)	5 (65.23)	5 (839.94)
$\mathrm{tILU}_0(0.67, 0.5)$	6 (0.09)	7 (1.60)	7 (25.28)	7 (360.94)
Jacobi(0.67)	7 (0.07)	8 (1.56)	7 (22.38)	8 (343.19)
Gauss-Seidel	6 (0.07)	6 (1.35)	7 (23.17)	7 (329.43)
Pe=5000				
$\mathrm{ILU}_0(0.67)$	6 (0.16)	7(3.75)	7 (80.21)	7 (1550.34)
$\mathrm{tILU}_0(0.67, 0.5)$	8 (0.09)	9 (1.53)	9 (21.13)	9 (284.68)
Jacobi(0.67)	10 (0.08)	13 (1.73)	13 (24.88)	14 (353.83)
Gauss-Seidel	8 (0.07)	10 (1.47)	12 (23.78)	13 (368.41)

Table 4.7: Truncation statistics for Example 4.1 as a function of Pe and the truncation parameter α. The discrete convection-diffusion operators with $n = 261121$ is obtained from Q1 SUPG FEM discretisation on a uniform grid. The results are presented in the format $\eta[\%]$.

Pe	$\alpha = 0$	$\alpha = 0.25$	$\alpha = 0.5$	$\alpha = 1$
2000	3115281 (100)	727545 (23)	440533 (14)	347489 (11)
8000	3115281 (100)	951577 (31)	619169 (20)	347489 (11)

Table 4.8: Truncation statistics for Example 4.2 as a function of Pe and the truncation parameter α. The discrete convection-diffusion operators with $n = 857375$ is obtained from Q1 SUPG FEM discretisation on a uniform grid. The results are presented in the format $\eta[\%]$.

Pe	$\alpha = 0$	$\alpha = 0.25$	$\alpha = 0.5$	$\alpha = 1$
250	100	0.26	0.26	0.26
5000	100	1.00	0.55	0.45

the coarse-level operators. We observe that the $\text{tILU}_0(*, 0.5)$ smoother has a computational cost that is very close to that of the Jacobi smoother.

5 Concluding remarks

A new smoothing methodology for multigrid preconditioning of discrete convection-diffusion problems is shown herein to be a robust alternative to existing techniques—especially when implemented within an AMG framework. The asymptotic behaviour with respect to grid size and Peclet number essentially mirrors that of ILU_0 smoothing, but the computational cost is significantly reduced. The new smoother also has favourable parallelisation properties, similar to those of the simplest Jacobi smoother. Future work will study its efficiency in the context of "black-box" preconditioning of unsteady Navier-Stokes [15] and Boussinesq [6] flow problems.

References

[1] J. Bey, G. Wittum: Downwind numbering: robust multigrid for convection-diffusion problems, *Appl. Numer. Math.*, **23**(1997), 177–192.

[2] J. Boyle, M. Mihajlović, J. Scott: `HSL_MI20`: an efficient AMG preconditioner for finite element problems in 3D, *Int. J. Numer. Meth. Engnrg*, **82**(1)(2010), 64–98.

[3] W. Briggs, V. Henson and S. McCormick: *Multigrid tutorial*, Second edition, SIAM Philadelphia, 2000.

[4] W. Dahmen, S. Müller, T. Schlinkmann: On adaptive multigrid solver for convection-dominated problems, *SIAM J. Sci. Comput.*, **23**(3)(2001), 781–804.

[5] H.C. Elman, M.P. Chernesky: Ordering effects on relaxation methods applied to the discrete one-dimensional convection-diffusion equation, *SIAM J. Numer. Anal.*, **30**(5)(1993), 1268–1290.

[6] H. Elman, M. Mihajlović, D. Silvester: Fast iterative solvers for buoyancy driven flow problems, *J. Comput. Phys.*, **230**(10)(2011), 3900–3914.

[7] H. Elman, D. Silvester, A. Wathen: *Finite Elements and Fast Iterative Solvers with applications in incompressible fluid dynamics*, Oxford University Press, Oxford, 2005.

[8] B. Fisher, A. Ramage, D. Silvester, A. Wathen: On parameter choice and iterative convergence for stabilised discretisations of advection-diffusion problems, *Comput. Methods Appl. Mech. Engnrg*, **179**(1999), 179–195.

[9] P.M. Gresho, R.L. Sani: *Incompressible Flow and the Finite Element method: Volume 1: Advection-Diffusion*, John Wiley, Chichester, 1998.

[10] W. Hackbusch, T. Probst: Downwind Gauss-Seidel Smoothing for Convection Dominated Problems, *Numer. Linear Algebra Appl.*, **4**(2)(1997), 85–102.

[11] P.W. Hemker: The incomplete LU-decomposition as a relaxation method in multi-grid algorithms, *In: Computational and Asymptotic Methods for Boundary and Interior Layers,* (J.J.H. Miller, ed.), 1980, 306–311.

[12] M. Heil, A.L. Hazel: `oomph-lib` – an object-oriented multi-physics finite-element library, In: Fluid-Structure Interaction (M. Schäffer, H.-J. Bungartz, eds.), Springer, 2006, pp. 19–49. (`oomph-lib` is available as open-source software at `http://www.oomph-lib.org/`).

[13] V.E. Henson, U.M. Yang: BoomerAMG: A parallel algebraic multigrid solver and preconditioner, *Appl. Numer. Math.*, **41**(2002), 155-177.

[14] K. Johannsen: A Robust Smoother for Convection-Diffusion Problem with Closed Characteristics, *Computing*, **65**(2000), 203–225.

[15] D. Kay, P. Gresho, D. Griffiths and D. Silvester: Adaptive time-stepping for incompressible flow; part II: Navier-Stokes equations, *SIAM J. Sci. Comput.*, **32**(2010), 111–128.

[16] H. Kim, J. Xu, L. Zikatanov: A multigrid method based on graph matching for convection-diffusion equations, *Numer. Linear Algebra Appl.*, **10**(2003), 181–195.

[17] S. Le Borne: Ordering Techniques for Two– and Three–Dimensional Convection-Dominated Elliptic Boundary Value Problems, *Computing,* **64**(2000), 123–155.

[18] Y. Notay: An aggregation-based algebraic multigrid method, *Electronic Transactions on Numerical Analysis*, **37**(2010), 123–146.

[19] A. Ramage: A multigrid preconditioner for stabilised discretisations of advection-diffusion problem, J. Comput. Appl. Math., **110**(1999), 187–203.

[20] G.O. Rees: Efficient "black-box" multigrid solvers for convection-dominated problems, Ph.D Thesis, The University of Manchester, 2011.

[21] Y. Saad: *Iterative methods for Sparse Linear Systems*, 2nd ed., SIAM, Philadelphia, 2003.

[22] D. Silvester, H. Elman and A. Ramage: Incompressible Flow and Iterative Solver Software (IFISS) version 3.1, 2011 (see `http://www.manchester.ac.uk/ifiss/`).

[23] Syamsudhuha: A study of multigrid methodology for the convection-diffusion equation, Ph.D thesis, UMIST, 2002.

[24] R. Tarjan: Depth-first search and linear graph algorithms, *SIAM J. Comput.*, **1**(1972), 146–160.

[25] U. Trottenberg, C.W. Oosterlee, A. Schüller: *Multigrid*, Academic Press, London, 2001.

[26] S. Turek: *Efficient Solvers for Incompressible Flow Problems*, Springer, Berlin, 1999.

[27] F. Wang, J. Xu: A crosswind block iterative method for convection dominated problems, *SIAM J. Sci. Comput.*, **21**(2)(1999), 620–645.

[28] P. Wesseling: *An Introduction to Multigrid Methods*, John Wiley, Chichester, 1992.

[29] G. Wittum: On the robustness of ILU smoothing, *SIAM J. Sci. Stat. Comput.*, **10**(4)(1989), 699–717.

[30] G. Wittum: Linear iterations as smoothers in multigrid methods: Theory with applications to incomplete decompositions, *Imp. Comput. Sci. Engnrg*, **1**(1989), 180–215.

[31] C.-T. Wu, H.C. Elman: Analysis and comparison of geometric and algebraic multigrid for convection-diffusion equations, *SIAM J. Sci. Comput.*, **28**(6)(2006), 2208–2228.

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