Available online at www.sciencedirect.com
SCIENCE DIRECT®
Journal of Computational Physics 210 (2005) 274–291
JOURNAL OF COMPUTATIONAL PHYSICS
www.elsevier.com/locate/jcp

Acceleration of the generalized global basis (GGB) method for nonlinear problems ☆

Haim Waisman [a,*], Jacob Fish [a], Raymond S. Tuminaro [b], John N. Shadid [c]

[a] *Department of Civil, Mechanical and Aerospace Engineering, Rensselaer Polytechnic Institute, 44 Brinsmade Terrace, Troy, NY 12180-3590, United States*
[b] *Sandia National Laboratories, P.O. Box 969, MS 9159, Livermore, CA 94551, United States*
[c] *Sandia National Laboratories, P.O. Box 5800, MS 0316, Albuquerque, NM 87185, United States*

Received 18 June 2004; received in revised form 16 February 2005; accepted 18 April 2005
Available online 16 June 2005

Abstract

Two heuristic strategies intended to enhance the performance of the generalized global basis (GGB) method [H. Waisman, J. Fish, R.S. Tuminaro, J. Shadid, The Generalized Global Basis (GGB) method, International Journal for Numerical Methods in Engineering 61(8), 1243–1269] applied to nonlinear systems are presented. The standard GGB accelerates a multigrid scheme by an additional coarse grid correction that filters out slowly converging modes. This correction requires a potentially costly eigen calculation. This paper considers reusing previously computed eigenspace information. The GGBα scheme enriches the prolongation operator with new eigenvectors while the modified method (MGGB) selectively reuses the same prolongation. Both methods use the criteria of principal angles between subspaces spanned between the previous and current prolongation operators. Numerical examples clearly indicate significant time savings in particular for the MGGB scheme.

Keywords: Multilevel; Multigrid; Preconditioner; Indefinite; Nonsymmetric; GGB; GMRES

1. Introduction

The efficient solution of large systems of equations $Ax = b$ arising from partial differential equations remains a challenging problem for nonsymmetric and indefinite systems. For symmetric positive definite systems, standard multigrid methods are very efficient solvers due to their optimal complexity (computa-

☆ This work was supported by the ASC DOE NNSA program at Sandia National Laboratories under contract number DE-ACD4-94AL85000.
* Corresponding author. Tel.: +1 518 522 2290; fax: +1 518 276 4833.
E-mail address: waismh@rpi.edu (H. Waisman).

0021-9991/$ - see front matter © 2005 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2005.04.016

tional work is proportional to the number of unknowns). However, when the system is nonsymmetric or highly indefinite, multigrid methods may not perform as well [2]. Such systems arise in a variety of applications including linearized Navier–Stokes equations, saddle-point problems, least squares problems with constraints and systems with an indefinite constitutive tensor arising as a result of localized damage in solids. Some multilevel methods have been applied for certain weakly indefinite systems. However, the existing strategies impose restrictions on the coarse grid, requiring that these grids are sufficiently fine for the proposed algorithms to converge [3,4]. For nonsymmetric and highly indefinite systems, various methods have been proposed, yet a general and efficient methodology is still an ongoing research area. Previous work [5] proposes an operator (matrix) dependent black box multigrid scheme for a single partial differential equation on structured grid problems. Fish et al. [6,7] utilize multigrid procedures in the context of normal equations. In [8] the authors employ a special energy minimization interpolation technique for convection diffusion problems. Recently, an interesting idea to use a "self-correcting" multigrid has been proposed for symmetric systems [9,10]. This technique finds the algebraically smooth error components unresolved by multigrid when applied to the homogeneous problem $Ax = 0$ with a random initial guess, and adjusts the coarsening process accordingly. Other approaches include a straightforward application of the multigrid method as a preconditioner to Krylov iterative solvers [11–13]. However, convergence depends on the type of multigrid method used and the spectrum of the preconditioned system [1,13,14]. Consequently, Carpentieri et al. [14,15] proposed to "remove" the smallest eigenvalues of the preconditioned linear system by shifting them. This involves computations of the smallest eigenvalues which sometimes can be expensive.

The method presented here extends the global basis (GB) method [16,17] and, in particular, the generalized global basis (GGB) method [1] when applied to general nonlinear problems solved by Newton's method. The GGB method stabilizes the entire multilevel procedure by constructing an additional coarse grid correction spanned by the nonconverging and slowly converging eigenmodes of the multilevel iteration. In this sense, the GGB method fits into the "self-correcting" multigrid methodology. The idea is to filter out modes that are "nonconverging" and "slow-to-converge" and resolve them on an additional coarse grid. This accelerates the iterative process and yields rates of convergence similar to the application of the unaccelerated multilevel method applied to a positive definite system. Consequently, any multilevel method may be applied to difficult systems, assuming only a small number of those eigenmodes need to be filtered. The method can be used as a stand alone solver or as a preconditioner to Krylov methods.

In this paper, we introduce and study two strategies to reduce the setup cost associated with GGB. Our objective is to reduce the overall *CPU* time when GGB is applied to a sequence of linear systems such as those arising from nonlinear problems solved by Newton's method. Since most of the computational work is governed by the eigen computations, reuse of eigenspace information may lead to significant *CPU* time savings. The first scheme (GGBα) computes only a few eigenvectors at each linear solve and appropriately enriches an existing prolongation operator. The second strategy is a modified GGB method termed MGGB. The method predicts whether the previous filter may be used at the current step or whether a new subspace should be computed. Both strategies are based on a criterion that measures the maximum principal angle between subspaces.

The paper is organized as follows. In Section 2, a brief introduction of the GGB method is presented. In Section 3, we motivate the idea of eigenspace reuse on a simple 1D nonlinear (and nonsymmetric) modified Bratu problem. In Section 4, we discuss the *GGB*α and MGGB strategies that employ those ideas. In Section 5, we study performance of the proposed strategies on various problems. Finally, we conclude with some remarks in Section 6.

2. Overview of the GGB method

Consider a generic two-level multigrid V-cycle for the solution of linear system of equations

$$Ku = f$$

in which the system matrix $K \in \mathbb{R}^{N\times N}$ is generally nonsymmetric indefinite. Let S be defined as the smoothing iteration matrix

$$S = I - M^{-1}K,$$

with a relaxation procedure $M \in \mathbb{R}^{N\times N}$ and the identity matrix I. Let v_1 and v_2 denote the number of pre- and post- smoothing steps, respectively. If the error after iteration i is $e^i = u - u^i$, then reduction of the error after one V-cycle is controlled by the multigrid iteration matrix $R_{\mathcal{MG}}$, given as

$$e^{i+1} = S^{v_2}TS^{v_1}e^i = R_{\mathcal{MG}}e^i. \tag{1}$$

where $T \in \mathbb{R}^{N\times N}$ is the coarse grid correction given by

$$T = I - P(RKP)^{-1}RK,$$

with the prolongation operator from the coarse grid to the fine grid $P : \mathbb{R}^m \to \mathbb{R}^N$, and the restriction operator $R : \mathbb{R}^N \to \mathbb{R}^m$. For symmetric systems the restriction operator is usually taken as the transpose of the prolongation operator, i.e., $R = P^{\mathrm{T}}$. T is a projector satisfying $T = T^2$ with a spectral radius of $\rho(T) = 1$. Multilevel methods consist of two major elements: smoothing and coarse grid correction. When symmetric positive definite (SPD) systems are considered, classical iterative methods, used as smoothers, eliminate the oscillatory components of the error leaving the smooth components almost untouched. This motivates the use of a coarse grid correction, where smooth components of the error are effectively approximated on a coarser grid. However, for difficult systems such as indefinite and/or nonsymmetric systems, smoothing may leave some oscillatory modes untouched, and thus standard multilevel methods might magnify these modes rather than reducing them [2]. Recently, the GGB method for highly indefinite and nonsymmetric systems has been proposed [1]. The current paper is a direct extension of the GGB method when applied to a sequence of linear solves generated by Newton's method. The GGB method [1] is a generalization of the global basis method [16,17]. It accelerates (stabilizes) the entire multigrid procedure in the following way. It first identifies all the troublesome modes of the applied multigrid method by solving for the largest magnitude eigenvalues λ_i of the multigrid iteration matrix in Eq. (1)

$$R_{\mathcal{MG}}\phi_i = \lambda_i\phi_i, \quad i = 1,\ldots,N. \tag{2}$$

The troublesome modes are the highest eigenvalues that are either not converging (indefinite) or "slow-to-converge" modes. The nonconverging eigenvalues are those that lie outside the unit circle $|\lambda_i| > 1$, and the "slow-to-converge" are the ones that lie inside the unit circle, however very close to one, i.e. $1 - \delta < |\lambda_i| < 1$, for some small positive constant δ. For this purpose, an implicitly restarted Arnoldi method [18] from ARPACK [19] is employed. Next, based on the computed eigenvalues, say k eigenvalues, the GGB method constructs an additional coarse grid correction, with the prolongation operator spanned by the corresponding eigenvectors

$$Q_f = \mathrm{span}\{\phi_i\}_{i=1}^k = \begin{bmatrix} | & & | \\ \phi_1 & \ldots & \phi_k \\ | & & | \end{bmatrix}_{N\times k}. \tag{3}$$

As shown in [1], the additional coarse grid is used as a multigrid filter, eliminating those troublesome modes. Therefore, this method belongs to the class of "self-correcting" multigrid methods, which find the algebraically smooth error components unresolved by multigrid [9,10]. However, as opposed to [9,10] the algebraically smooth error components are obtained directly from the eigenvalue problem (2). Fig. 1 schematically illustrates the architecture of the method used in the paper. Black circles denote local smoothing at each level, and GMRES/QMR is an outer accelerator. The GGB cycle is used to precondition Krylov methods. We note that other GGB cycles are also possible for nonsymmetric systems (see Fig. 2).

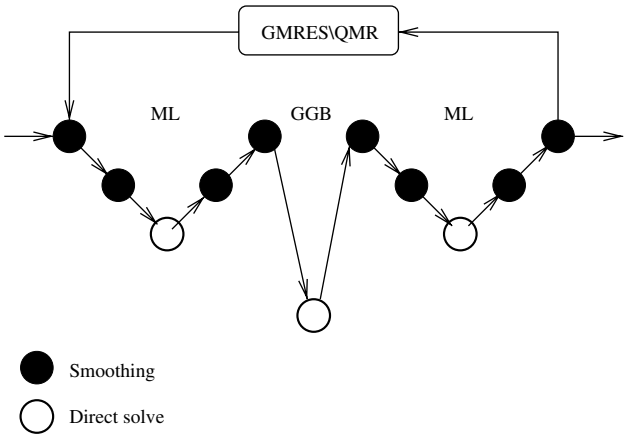

Fig. 1. Generalized global basis (GGB) cycle.

The overall error reduction of a single GGB cycle illustrated in Fig. 1, without an external accelerator and one smoothing iteration at each level can be written as

$$e^{i+1} = (\mathrm{STS})^{\nu_2} F_{\mathrm{GGB}} (\mathrm{STS})^{\nu_1} e^i = R^{\nu_2}_{\mathscr{MG}} F_{\mathrm{GGB}} R^{\nu_1}_{\mathscr{MG}} e^i,$$

where STS is the multilevel iteration matrix and ν_1, ν_2 correspond to the number of V-cycles. F_{GGB} is the additional projector (filter), given as

$$F_{\mathrm{GGB}} = I - Q_f (Q_f^* K Q_f)^{-1} Q_f^* K,$$

where the prolongation operator Q_f and the restriction Q_f^* are spanned by the highest modes of $R_{\mathscr{MG}}$ given in (3).

3. Motivation

The GGB method is useful for solving very difficult problems such as highly indefinite and/or nonsymmetric systems that require one linear solve or a sequence of linear solves. For the latter, the method is most attractive for problems with multiple right-hand sides such as linear transient problems or shift-and-invert eigenvalue problems, since the indefinite and "slow-to-converge" eigenspace given in (2) has to be computed only once at the setup phase and can later be reused throughout the entire sequence of linear solves. However, if the method is applied to a sequence of linear solves, for instance, those that arise from

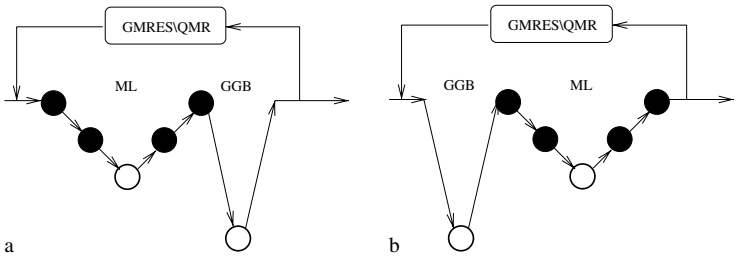

Fig. 2. Various GGB cycles for nonsymmetric systems: (a) GGB filter on the right, (b) GGB filter on the left.

nonlinear problems solved by Newton's method, then the left-hand side (the Jacobian matrix) as well as the right-hand side changes from one Newton iteration to the other. This results in an eigen computation for each linear solve in the sequence which may dominate the entire computational cost. However, if the sequence of Jacobians are slowly changing, then the eigenspace information can be reused without significantly affecting the convergence rate of the GGB method. In practice, the total linear solve iterations will sometimes increase but overall *CPU* time can be reduced due to savings resulting from reuse of the eigenspace.

In this section we motivate the idea of eigen reuse and also illustrate aspects that must be considered in developing a criteria for reuse. To do this, we consider the following 1D nonlinear boundary value problem on the interval $\Omega = [0,1]$

$$\begin{cases} u'' + \alpha u' + \lambda e^u = 0, \\ u(x=0) = u(x=1) = 0. \end{cases} \tag{4}$$

The standard "Bratu problem" is obtained for $\alpha = 0$ [20]. Central difference discretization on a uniform mesh leads to a set of nonlinear equations

$$F_i(u_{i-1}, u_i, u_{i+1}) = \beta u_{i+1} + \gamma u_{i-1} - 2u_i + \lambda h^2 e^{u_i} = 0, \quad i = 1, \ldots, N, \tag{5}$$

where N is the number of mesh nodes, h is the mesh spacing, $\beta = (1 + \frac{\alpha h}{2})$ and $\gamma = (1 - \frac{\alpha h}{2})$. Applying Newton's method to (5) yields a sequence of linear systems with a Jacobian matrix given by the stencil

$$\text{tridiag}[\gamma, \ -2 + \lambda h^2 e^{u_i}, \ \beta].$$

We apply a standard two level multigrid method to the sequence of linear systems. The mesh spacing is doubled to obtain the coarse mesh. Linear interpolation is used to transfer from coarse grid to fine grid $P_{mg} : \mathbb{R}^n \to \mathbb{R}^N$, and either a full weighting $R_{\text{full}} = \frac{1}{2} P_{mg}^{\mathrm{T}}$ or injection

$$R_{\text{inject}} = \begin{pmatrix} 0 & 1 & 0 & & & & \\ & & 0 & 1 & 0 & & \\ & & & & 0 & 1 & 0 \end{pmatrix}.$$

is used for restriction.

Table 1 illustrates the behavior of the standard multigrid method accelerated by GGB, applied to problem (4). We set $\lambda = 3$ and $\alpha = 0$, with initial guess $u_0 = 2\sin(\pi x)$, which yields a symmetric indefinite Jacobian. One pre- and post-Gauss–Siedel smoothing iteration is performed on the fine level. The GGB filter in (3) is constructed out of four eigenvectors. Specifically, Table 1 reports the relative perturbation of the highest eigenvalue and eigenvector. The last column of the table indicates the acute angle between the eigenvectors computed at iteration j and $j+1$, respectively. The acute angle θ between two vectors x_1 and x_2 is defined as

Table 1
Variation of various quantities of R_{MG} with nonlinear iteration

Newton iteration	$\frac{\lvert\lambda_1^j - \lambda_1^{j+1}\rvert}{\lvert\lambda_1^j\rvert}$		$\frac{\lVert q_1^j - q_1^{j+1}\rVert}{\lVert q_1^j\rVert}$		$\theta(q_1^j, q_1^{j+1})$	
	MG_{full}	MG_{inj}	MG_{full}	MG_{inj}	MG_{full}	MG_{inj}
1,2	0.0155	6.3075	0.0754	1.7429	4.0061	58.7466
2,3	0.0173	11.0640	0.0804	1.4135	4.4526	89.9387
3,4	0.0033	0.9579	0.0136	0.0961	0.7723	5.5056
4,5	0.0148	0.8137	0.0711	0.1243	3.4193	7.1258
5,6	0.0007	0.0587	0.0033	1.9976	0.172	5.6059

$$\cos\theta = \frac{|x_1^{\mathrm{T}} x_2|}{\|x_1\|_2 \|x_2\|_2}. \tag{6}$$

It can be seen from the table that the relative perturbation of the eigenvalues as well as the eigenvectors is small, for the case of multigrid with full weighting. This suggests that reusing the same filter Q_f in Eq. (3) should be sufficient to obtain satisfactory convergence rates. As a counter example, multigrid with injection suffers from large relative variations in eigenvectors, hence one must recompute the eigenspace. In fact, if the initial eigenspace computed at the first Newton step is reused throughout the linear sequence, eventually the GGB method fails to converge. That is, eigen reuse must be done carefully. Further, a measure based on the acute angle between the two eigenvectors may detect a large variation in the eigenspace. We also note that a variation in eigenvalues does not imply changes in eigenmodes and therefore cannot be used as appropriate measure. Nevertheless, the eigenvalues play an important role in determining the number of modes used to construct the filter (see Section 2).

In the next section we generalize the acute angle to a set of criteria that automatically determine when the eigenspace needs to be recomputed.

4. Strategies to reuse eigenspace information

We propose two heuristic strategies that reuse the already computed prolongation operator. The first measure is termed GGBα and it is based on constantly adding new eigen information to the filter, i.e., increasing the prolongation space with every Newton iteration. The second is MGGB and it is based on using the exact same prolongation operator (full reuse). Both strategies rely on the criteria of principal angles between subspaces spanned between the previous and current prolongation operators. Computation of principal angles between subspaces is performed in many applications, for example data analysis, random processes, stochastic realization, etc. [21].

4.1. The GGBα strategy

The GGBα strategy is based on augmenting new information into a previously computed prolongation operator, and thus enriching the filter. However, this enrichment cannot be done automatically since the new eigenvectors may be obtained from a linear combination of the previous eigenvectors causing ill conditioning of the coarse grid. To be able to safely add the new information, we develop a measure based on the principal angles between each new eigenvector and the previous eigenspace.

In the first Newton iteration a GGB filter is constructed by solving the eigen problem of the multigrid iteration matrix $R^1_{\mathscr{MG}}$ for k nonconverging and "slow-to-converge" eigenvalues

$$R^1_{\mathscr{MG}} q_i^1 = \lambda_i^1 q_i^1, \quad i = 1, \ldots, k.$$

As the nonlinear iteration proceeds and the next linear system is to be solved, we propose to compute only t eigenvalues and eigenvectors, such that $t < k$. The idea is to enrich the space of a prolongation operator with new eigenvectors. The updated operator after nonlinear iteration j may be written in the following way:

$$Q^j = [Q^1, U^2, \ldots, U^j],$$

where

$$Q^1 = \{q_1^1, q_2^1, \ldots, q_k^1\},$$

and the U matrices are given by

$$U^j = \{q_1^j, \ldots, q_t^j\}, \quad j > 1,$$

obtained from

$$R^j_{\mathscr{M}\mathscr{G}} q_i^j = \lambda_i^j q_i^j, \quad i = 1, \ldots, t. \tag{7}$$

In words, U^j contains the eigenvectors computed at the current iteration while Q^j is the accumulated prolongation operator. Thus the idea of the GGBα method is to always recompute a limited amount of eigen information and to use it to augment the prolongation operator used in the previous Newton iteration. Of course as the nonlinear iteration proceeds, the prolongation operator grows increasing the size of the coarse grid.

If the space spanned by the two subspaces Q^j and $[Q^j, U^{j+1}]$ is nearly the same then the gain associated with the new information is insignificant.

One way to measure the difference between the two subspaces is to compute the maximum principal angle between them. Angles between subspaces are defined in the following way (for more details see [21,22]). Let $\mathscr{F}$ and $\mathscr{G}$ be the column space of Q^j and U^{j+1}, respectively, and let $s = \dim(\mathscr{F})$ and $t = \dim(\mathscr{G})$ with $s \geqslant k \geqslant t$. The principal angles $\alpha_1, \ldots, \alpha_t \in [0, \frac{\pi}{2}]$ between $\mathscr{F}$ and $\mathscr{G}$ may be defined recursively for $p = 1, \ldots, t$ by

$$\cos(\alpha_p) = \max_{v \in \mathscr{G}} \max_{w \in \mathscr{F}} v^{\mathrm{T}} w = v_p^{\mathrm{T}} w_p \tag{8}$$

subject to

$$\|v\| = \|w\| = 1, \quad v^{\mathrm{T}} v_i = 0, \quad w^{\mathrm{T}} w_i = 0, \quad i = 1, \ldots, p-1.$$

The vectors $v_1, \ldots, v_t$ and $w_1, \ldots, w_t$ are called principal vectors. More explicitly, definition (8) follows if the subspaces $\mathscr{F}$ and $\mathscr{G}$ are orthogonalized, and rotated such that the inner product between their columns is maximized and reordered in an ascending order.

In our context the rank of the prolongation Q^j is much smaller than the rank of the multigrid iteration matrix $R^j_{\mathscr{M}\mathscr{G}}$. For this purpose, we choose the Björck–Golub algorithm [21] to compute the principal angles. This algorithm is based on a singular value decomposition (SVD). Let the columns of $\tilde{Q} \in \mathbb{R}^{N \times s}$ and $\tilde{U} \in \mathbb{R}^{N \times t}$ be an orthonormal basis for $range(\mathscr{F})$ and $range(\mathscr{G})$, say computed by a QR factorization, i.e.,

$$Q^j = \tilde{Q}\tilde{R}_1,$$
$$U^{j+1} = \tilde{U}\tilde{R}_2,$$

where $\tilde{Q}^{\mathrm{T}}\tilde{Q} = I_s$, $\tilde{U}^{\mathrm{T}}\tilde{U} = I_t$. Further, let

$$\tilde{Q}^{\mathrm{T}}\tilde{U} = Z \begin{pmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_t \end{pmatrix} V^{\mathrm{T}}, \quad 1 \geqslant \sigma_1 \geqslant \cdots \geqslant \sigma_t \geqslant 0$$

be the reduced SVD of $\tilde{Q}^{\mathrm{T}}\tilde{U}$, where $Z \in \mathbb{R}^{N \times s}$, $V \in \mathbb{R}^{t \times t}$. The principal angles are given by

$$\alpha_i = \arccos(\sigma_i), \quad i = 1, \ldots, t,$$

where $0 \leqslant \alpha_1 \leqslant \cdots \leqslant \alpha_t \leqslant \frac{\pi}{2}$ and the columns of $\tilde{Q}Z$ and $\tilde{U}V$ are the principal directions.

For computational purposes, the GGBα method is then defined by using the new U^{j+1} eigenvectors to enrich the space of the prolongation Q^j only if all the principal angles (not only the maximum angle) $\alpha_i \geqslant \alpha_{\mathrm{cr}}$, where α_{cr} is some tolerance angle. Considering only the maximum principal angle might lead to an ill conditioned coarse grid matrix as some vectors in U^{j+1} might lie in Q^j while others are nearly orthogonal to Q^j. Therefore, one should compute an angle between each new eigenvector $q_i^{j+1} \in U^{j+1}$ and the

previous eigenspace Q^j.[1] The computation of the angle between a vector and a subspace can be simplified in the following way. Define a projection of q_i^{j+1} onto the subspace $\tilde{Q}^j$

$$y_i = \tilde{Q}^j \left((\tilde{Q}^j)^{\mathrm{T}} \tilde{Q}^j \right)^{-1} (\tilde{Q}^j)^{\mathrm{T}} q_i^{j+1} = \tilde{Q}^j (\tilde{Q}^j)^{\mathrm{T}} q_i^{j+1}, \quad i = 1, \ldots, t. \tag{9}$$

The cosine of the angle between q_i^{j+1} and $\tilde{Q}^j$ with relation $\|q_i^{j+1}\| = 1$ is given by

$$\cos \alpha_i = \frac{|y_i^{\mathrm{T}} q_i^{j+1}|}{\|y_i\|_2 \|q_i^{j+1}\|_2} = \frac{\left| \left(\tilde{Q}^j (\tilde{Q}^j)^{\mathrm{T}} q_i^{j+1} \right)^{\mathrm{T}} q_i^{j+1} \right|}{\|\tilde{Q}^j (\tilde{Q}^j)^{\mathrm{T}} q_i^{j+1}\|_2} = \frac{\left| \left((\tilde{Q}^j)^{\mathrm{T}} q_i^{j+1} \right)^{\mathrm{T}} \left((\tilde{Q}^j)^{\mathrm{T}} q_i^{j+1} \right) \right|}{\|(\tilde{Q}^j)^{\mathrm{T}} q_i^{j+1}\|_2} = \|(\tilde{Q}^j)^{\mathrm{T}} q_i^{j+1}\|. \tag{10}$$

Note that GGBα incrementally adds eigenvectors to the projector, which leads to an increased cost to apply the filter. However, attempting to remove eigenvectors no longer needed is even more expensive. Moreover, our experience shows that only limited amount of eigenvectors are added and thus the additional cost is negligible. The GGBα strategy may be very effective if the multigrid iteration matrices vary significantly from iteration to iteration. Yet, when the eigenspace only slightly changes, computing t new eigenvectors may not be justified.

4.2. The MGGB strategy

The second strategy, termed MGGB (modified GGB), is based on the full reuse of a previously computed prolongation operator. That is, using the same GGB filter for several nonlinear iterations. Again, if the iteration matrices only change slightly, then full reuse may yield satisfactory convergence without the cost of repeating the eigen calculation. On the other hand, automatic reuse might be unsafe (see Section 3). To evaluate whether a new filter has to be computed or not, we measure how far are the previous eigenvectors (of the GGB filter) q_i^j from being also eigenvectors of the current multigrid iteration matrix $R_{\mathscr{MG}}^{j+1}$. Note that q_i^j are computed from (7), while $R_{\mathscr{MG}}^{j+1}$ denotes the current operator. This is achieved by employing a standard Rayleigh quotient estimation, popular in eigen calculations. Thus, the goal of MGGB is to avoid the exact computation of t new eigenvectors, which was employed by GGBα strategy. Later, we generalize the standard Rayleigh quotient measure to an angle between the vector $R_{\mathscr{MG}}^{j+1} q_i^j$ and the entire filter Q^j. This generalization is important due to the following. First, one can use similar algorithms for both GGBα and MGGB to compute the angle. Second, the Rayleigh quotient criteria may be somewhat relaxed since linear combination of eigenvectors might better capture the behavior.

We now describe the mathematical formulations. A Rayleigh quotient type measure between the eigenvector q_i^j and the multigrid operator $R_{\mathscr{MG}}^{j+1}$ is given by

$$\frac{\|R_{\mathscr{MG}}^{j+1} q_i - \rho_i q_i\|_2}{|\rho_i|} \leqslant \delta, \quad i = 1, \ldots, k, \tag{11}$$

where ρ_i is the Rayleigh quotient defined as

$$\rho_i = \frac{q_i^* R_{\mathscr{MG}}^{j+1} q_i}{q_i^* q_i}, \tag{12}$$

and δ is some small constant. Using the fact that $q_i^* q_i = 1$, (11) can be rewritten as

$$\frac{\|R_{\mathscr{MG}}^{j+1} q_i - (q_i^* R_{\mathscr{MG}}^{j+1} q_i) q_i\|_2}{|q_i^* R_{\mathscr{MG}}^{j+1} q_i|} \leqslant \delta, \quad i = 1, \ldots, k. \tag{13}$$

[1] In fact $\tilde{Q}^j$ can be used in the GGB method instead of Q^j.

Moreover, measure (13) is related to the acute angle (6) between $R^{j+1}_{\mathscr{MG}}q_i$ and q_i in the following way

$$\frac{\|R^{j+1}_{\mathscr{MG}}q_i - \rho_i q_i\|_2}{|\rho_i|} = \tan\theta_i. \tag{14}$$

This can be seen by writing down the acute angle and substituting the Rayleigh quotient (12)

$$\cos\theta_i = \frac{|q_i^* R^{j+1}_{\mathscr{MG}}q_i|}{\|R^{j+1}_{\mathscr{MG}}q_i\|_2} = \frac{|\rho_i|}{\|R^{j+1}_{\mathscr{MG}}q_i\|_2}.$$

So,

$$\cos^2\theta_i = \frac{\rho_i^*\rho_i}{\|R^{j+1}_{\mathscr{MG}}q_i\|_2^2} = \frac{\rho_i^*\rho_i}{\|R^{j+1}_{\mathscr{MG}}q_i\|_2^2 - \rho_i^*\rho_i + \rho_i^*\rho_i}.$$

Using the relation

$$\|R^{j+1}_{\mathscr{MG}}q_i - \rho_i q_i\|_2^2 = \|R^{j+1}_{\mathscr{MG}}q_i\|_2^2 - \rho_i^*\rho_i$$

we arrive at

$$\cos^2\theta_i = \frac{\rho_i^*\rho_i}{\|R^{j+1}_{\mathscr{MG}}q_i - \rho_i q_i\|_2^2 + \rho_i^*\rho_i}$$

and (14) follows (for more details see [22]).

One can generalize the acute angle θ_i to an angle between the vector $R^{j+1}_{\mathscr{MG}}q_i$ and the subspace Q^j (as opposed to a single vector), assuming real eigenvectors. Defining a projection of $R^{j+1}_{\mathscr{MG}}q_i$ onto the subspace Q^j using Eq. (9) and (10), one arrives at

$$\cos\beta_i = \frac{\|(Q^j)^{\mathrm{T}}R^{j+1}_{\mathscr{MG}}q_i\|}{\|R^{j+1}_{\mathscr{MG}}q_i\|_2}.$$

The square of the cosine is further simplified by

$$\begin{aligned}\cos^2\beta_i &= \frac{\|(Q^j)^{\mathrm{T}}R^{j+1}_{\mathscr{MG}}q_i\|_2^2}{\|R^{j+1}_{\mathscr{MG}}q_i\|_2^2} = \frac{\left(q_1^{\mathrm{T}}R^{j+1}_{\mathscr{MG}}q_i\right)^2 + \cdots + \left(q_i^{\mathrm{T}}R^{j+1}_{\mathscr{MG}}q_i\right)^2 + \cdots + \left(q_k^{\mathrm{T}}R^{j+1}_{\mathscr{MG}}q_i\right)^2}{\|R^{j+1}_{\mathscr{MG}}q_i\|_2^2} \\ &= \frac{\left(q_1^{\mathrm{T}}R^{j+1}_{\mathscr{MG}}q_i\right)^2 + \cdots + \rho_i^2 + \cdots + \left(q_k^{\mathrm{T}}R^{j+1}_{\mathscr{MG}}q_i\right)^2}{\|R^{j+1}_{\mathscr{MG}}q_i\|_2^2}.\end{aligned} \tag{15}$$

Substituting (6) into (15) yields

$$\begin{aligned}\cos^2\beta_i &= \frac{\left(q_1^{\mathrm{T}}R^{j+1}_{\mathscr{MG}}q_i\right)^2 + \cdots + \|R^{j+1}_{\mathscr{MG}}q_i\|_2^2\cos^2\theta_i + \cdots + \left(q_k^{\mathrm{T}}R^{j+1}_{\mathscr{MG}}q_i\right)^2}{\|R^{j+1}_{\mathscr{MG}}q_i\|_2^2} \\ &= \frac{\left(q_1^{\mathrm{T}}R^{j+1}_{\mathscr{MG}}q_i\right)^2}{\|R^{j+1}_{\mathscr{MG}}q_i\|_2^2} + \cdots + \cos^2\theta_i + \cdots + \frac{\left(q_k^{\mathrm{T}}R^{j+1}_{\mathscr{MG}}q_i\right)^2}{\|R^{j+1}_{\mathscr{MG}}q_i\|_2^2}\end{aligned}$$

and since all the terms on the right-hand side are greater or equal to zero, we get the following bound:

$$\cos^2\beta_i \geqslant \cos^2\theta_i = \frac{1}{\frac{\|R^{j+1}_{\mathscr{MG}}q_i - \rho_i q_i\|_2}{|\rho_i|} + 1},$$

which relates Rayleigh quotient to an angle between subspaces. A new eigenspace is then recomputed if $\max\{\beta_i\} \geqslant \beta_{\mathrm{cr}}$. The choice of β_{cr} depends on the problem solved. However, for the example problems given

in the next section, we find the methods almost insensitive to β_{cr}. That is, β_{cr} might affect local linear solves but will not affect the general trend. Note that the measures proposed for the MGGB method assume $|\lambda_1| \geqslant \cdots \geqslant |\lambda_k| \geqslant \cdots \geqslant |\lambda_N|$, which is a drawback to the method since no measure can take into account flipping in the order of the eigenvalues and the corresponding eigenvectors. The GGBα method does not suffer from this limitation.

5. Numerical results

In order to illustrate the behavior of the MGGB and GGBα methods, we apply the solvers to a 1D modified Bratu problem, 2D nonlinear sail (membrane) problem, 2D steady, thermal-convection flow and 3D chemical vapor deposition (CVD) reactor. The last problem is the most challenging for the multigrid solvers as different physics is used for modelling the problem.

5.1. 1D modified Bratu problem

We apply left preconditioners to GMRES for the solution of the modified Bratu problem described in (4). Two sweeps of a standard multigrid with restriction based on full weighting and injection described in Section 3 are used. Results are given for the GGB method and the corresponding MGGB method. One pre- and post-Gauss–Siedel smoothing sweep is applied within the multigrid cycle at the fine level. We do not consider here the GGBα strategy described in Section 4. Table 2 shows the convergence of the various preconditioners for a system of size $N = 315$. The problem parameters are chosen to be $\lambda = 3$ and $\alpha = 1.3$, with initial guess $u_0 = 2\sin(\pi x)$, yielding a nonlinear problem (4) that is indefinite and nonsymmetric. For the linear solve, we use the following stopping criteria $\frac{\|r^i\|_2}{\|r^0\|_2} \leqslant 10^{-8}$, where $r^i = f - Ku^i$ is the residual at the inner iteration i. The outer iteration is terminated when $\|F(u)\| \leqslant 10^{-6}$, where $F(u)$ denotes the nonlinear residual vector and u is the approximated solution vector. Results obtained for MGGB methods use a measure based on an approximate angle between a subspace and a vector described in Section (4), with a critical angle for recomputing the GGB filter set to $\theta_{cr} = 20°$. We choose only four modes to construct the filter. In general, all the multigrid methods maintain the same rate as the problem size increases, i.e., are mesh independent. Yet, adding the additional GGB operator (with only four modes) cuts in almost half the required number of iterations. It is also clear from Table 2 that GGB and MGGB with MG_{full} performs the best. In fact, both methods have similar convergence rates, yet the prolongation in the case of MGGB is computed only once at the first iteration and reused. Fig. 3 compares the maximum approximate angle $\theta_{app} = \angle(Q^j, R^{j+1}_{MG} q^j)$ and maximum exact angle $\theta_{exa} = \angle(Q^j, q^{j+1})$ between a subspace and a vector. For both MGGB cases the filter Q^j is computed only once. The figure illustrates that the vector $R^{j+1}_{MG} q^j$ we used to obtain the angle θ_{app} is valid when the multigrid iteration matrix only slightly changes as in the MG_{full} case.

Table 2
Convergence of various preconditioners to GMRES applied to modified Bratu problems of size 115, 315 and 515, respectively

Preconditioner	Total linear solve iterations			Average number of iterations		
	$N = 115$	$N = 315$	$N = 515$	$N = 115$	$N = 315$	$N = 515$
MG_{full}	133	112	102	8.31	8.00	7.84
MG_{inj}	298	265	245	18.62	18.92	18.84
GGB with MG_{full}	76	59	54	4.75	4.21	4.15
GGB with MG_{inj}	153	140	129	9.56	10.00	9.92
MGGB with MG_{full}	76	59	54	4.75	4.21	4.15
MGGB with MG_{inj}	161	145	133	10.06	10.35	10.23

The problem parameters are set to $\lambda = 3$ and $\alpha = 1.3$.

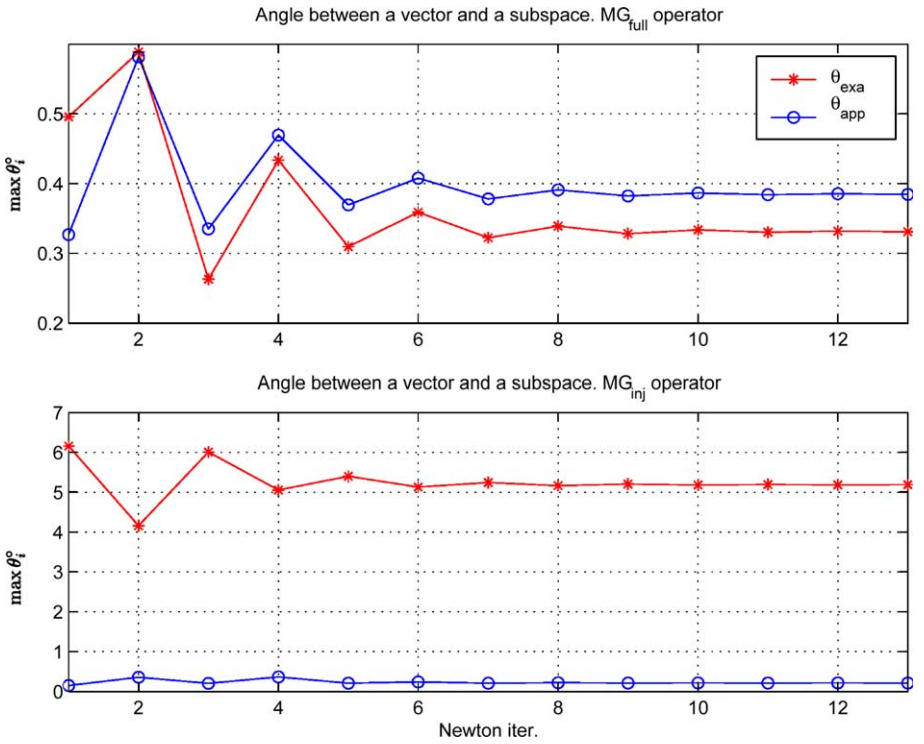

Fig. 3. Approximate and exact angle between a subspace and a vector for MG_{full} and MG_{inj} iteration matrices ($N = 315$), respectively.

5.2. A sail on nonlinear springs

We study the performance of the GGB filters applied to a 2D sail (membrane) on a nonlinear bed of springs. The sail has a small crack that is modeled with softening zones at the tips. The steady governing equation for transverse displacement u in the domain $(x, y) \in \Omega$ with boundary Γ is:

$$\nabla^2 u - K(u)u + f = 0 \quad \text{in } \Omega,$$

where f is a uniform distribution of wind load acting on the sail and K is a nonlinear spring given by $K(u) = k_0 u^{\alpha}$ where k_0 is the material constant (usually greater than zero). α describes the degree of nonlinearity. The boundary conditions for the problem are given as

$$u = 0 \quad \text{on } \Gamma,$$
$$\frac{\partial u}{\partial n} = 0 \quad \text{on } s^+, s^-,$$

where s^+ and s^- denotes the two sides of the crack. A standard Galerkin finite element discretization produces a system of nonlinear algebraic equations of the form $g(u) = f$, that are solved by Newton's method. For every Newton iteration j a global tangent stiffness matrix given by $K^j_{AB} = \frac{\partial g(u)}{\partial u}|_{(u=u^j)}$ is computed. The global tangent stiffness is assembled from local contributions of the from

$$k^j_{ab} = s_{ab} + (1+\alpha)k_0\left(u^h\right)^{\alpha} m_{ab},$$

where s_{ab} and m_{ab} are the standard stiffness and mass matrices, respectively, and u^h is the averaged approximated displacement at the nodes of the element. Fig. 4(a) shows the finite element mesh considered. The

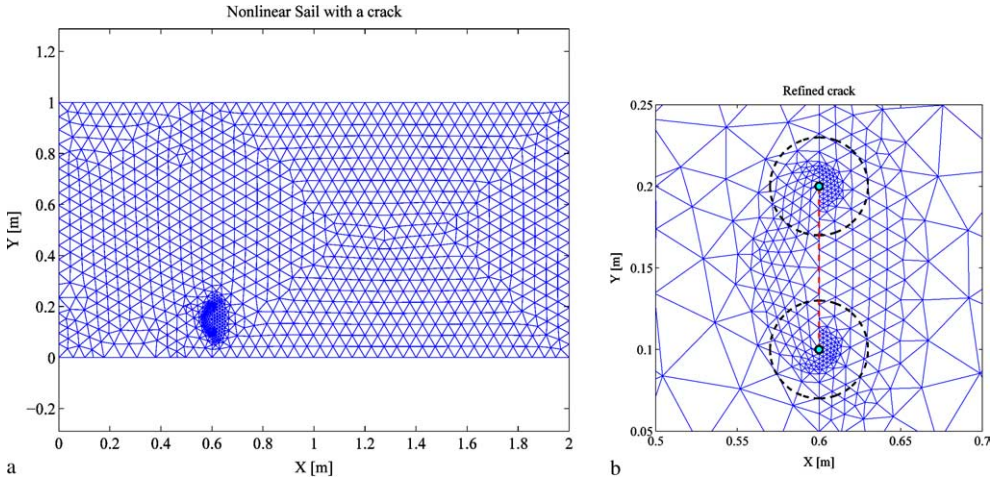

Fig. 4. A sail with a crack on nonlinear springs. (a) The mesh of the sail (2460 elements). (b) Zoom of the crack with damage zones.

sail is constrained all around and a uniform wind load is applied to all the nodes. We use 2460 elements for the model that results in systems of that 1183 unknowns. Details of the finite element mesh at the tip of the crack are shown in Fig. 4(b). The red dashed line illustrates the crack interface. The circles around the crack tips define the damage regions for which k_0 is negative. This causes the tangent stiffness matrix to become symmetric-indefinite for certain Newton iterations. We apply two cycles of a smoothed aggregation multilevel method to precondition GMRES(15), similar to two cycles used in GGB. We use two levels of the aggregation method and one Jacobi smoothing on the fine grid. The results are obtained for normalized variables $k_0 = -100$ inside the damaged zones and $k_0 = 100$ otherwise. The damaged region is set to have radius of 3 units. We use $\alpha = 2$ for the degree of nonlinearity. Fig. 5 presents the normalized displacement of the sail subjected to distributed wind load of 15 units. Table 3 summarizes the convergence behavior of the various preconditioners. We emphasize that only five modes are used by GGB every linear solve. The last two columns describe the total accumulated modes computed and the Newton iteration index in which they were computed. Again, GGB requires the minimum amount of iterations to converge, while both MGGB and GGBα performed the best of all the methods in terms of the overall time. Fig. 6 compares the maximum approximate angle $\theta_{\text{app}} = \angle(Q^j, R^{j+1}_{\mathscr{MG}} q^j)$ and maximum exact angle $\theta_{\text{exa}} = \angle(Q^j, q^{j+1})$ between a subspace and a vector. The results show good agreement in particular for the last Newton steps. This suggest that the approximated angle used by MGGB can effectively detect large variations in the GGB filter.

5.3. Thermal-convection flow in a box

In this section, we demonstrate the performance of GGB, GGBα and MGGB methods applied to steady, thermal-convection flow. The governing PDEs are the following Navier–Stokes with thermal energy equations

$$\text{Momentum} \quad \rho(\mathbf{u}\cdot\nabla)\mathbf{u} - \nabla\cdot\mathsf{T} - \rho\mathbf{g} = 0, \tag{16}$$

$$\text{Total mass} \quad \nabla\cdot(\rho\mathbf{u}) = 0, \tag{17}$$

$$\text{Thermal energy} \quad \rho\hat{C}_{\text{p}}(\mathbf{u}\cdot\nabla)T + \nabla\cdot\mathbf{q} = 0. \tag{18}$$

The unknown quantities are $\mathbf{u}$ the fluid velocity vector, P the hydrodynamic pressure and T the temperature. ρ, $\mathbf{g}$, and $\hat{C}_{\text{p}}$ are, respectively, the density, the gravity vector and the specific heat at constant pressure.

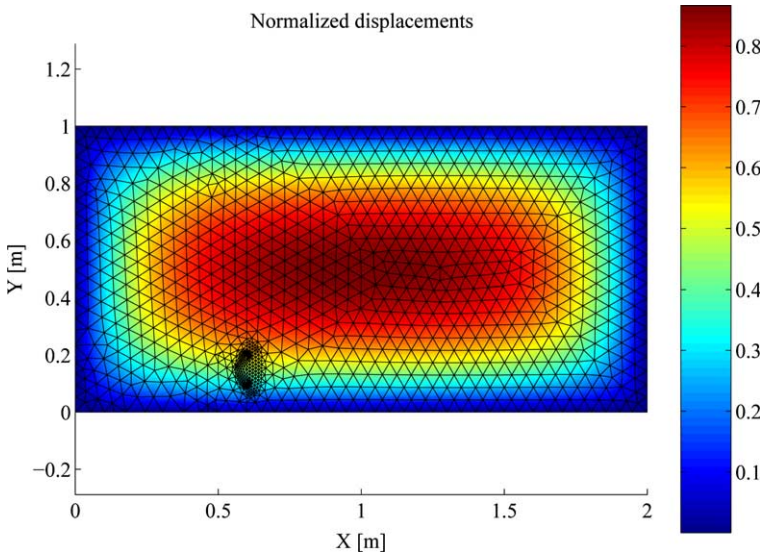

Fig. 5. Normalized displacements.

Table 3
CPU time and iteration summary for the cracked sail on nonlinear springs

Preconditioners	*CPU* time (s)			Linear iterations		Eigenvectors	
	Total	Eigensolver	Eigen %	Total	Average	Total	Itr. comp.
ML	54.7	–	–	1889	157.4	–	–
GGB	46.8	26.35	56.30	212	17.6	60	1–12
GGBα	37.9	19.73	52.07	280	23.3	13	1–12
MGGB	35.4	12.66	35.76	215	17.9	30	1–5,7

The Boussinesq approximation is used for representing the body force term. The necessary constitutive equations for T and $\mathbf{q}$ are

$$\text{Stress tensor} \quad \mathsf{T} = -P\mathbf{I} - \frac{2}{3}\mu(\nabla \cdot u)\mathbf{I} + \mu(\nabla \mathbf{u} + \nabla \mathbf{u}^{\mathrm{T}}),$$

$$\text{Heat flux} \quad \mathbf{q} = -\kappa \nabla T,$$

where μ is the viscosity and κ is the thermal diffusivity. Eqs. (16)–(18) are approximated by a Galerkin least squares formulation. The resulting nonlinear system of equations gives rise to a system of coupled, nonlinear and nonsymmetric algebraic equations. We employ MPSalsa [23,24] to generate the system of equations.

To solve the linear systems arising from Newton's method, we use right preconditioning with a restarted GMRES(m) method. A smooth aggregation multilevel method [25] implemented in ML package [26] is used for the multigrid method. This scheme is accelerated with a GGB, GGBα and MGGB methods. We apply a measure based on the angle between a vector and a subspace (see Section 4) for the following cases. In the GGBα method the angle is used to determine whether the new computed eigenvector is needed to enrich the prolongation. In all examples we compute two new eigenvectors every nonlinear iteration. If $\theta_i > 5°$ the new eigenvector is added. In MGGB we use a maximum approximate angle which is used to predict whether the entire prolongation should be recomputed. If $\theta > 20°$ a new filter is computed. The approximate angle is

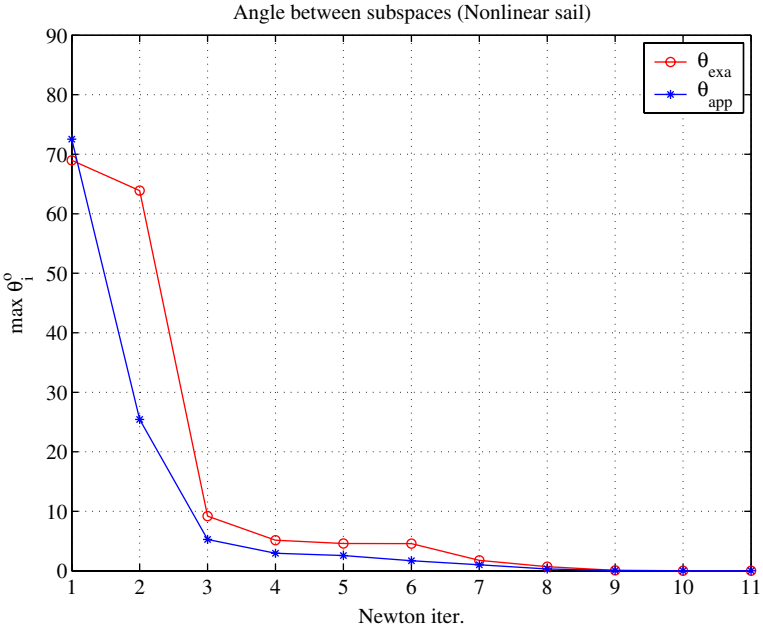

Fig. 6. Approximate and exact angle between subspace and a vector.

computed between the previous prolongation Q^j and the approximate vector $R^{j+1}_{\mathcal{M}\mathcal{G}}q^j_1$ where $q^j_1 \in Q^j$ corresponds to the highest eigenvalue. We employ LAPACK subroutines [27] to compute the maximum principal angle. The angle is computed numerically by first obtaining an orthogonal basis (QR factorization based on Householder triangularization) and second using an SVD type approach (see Section 4 for more details).

Due to nonsymmetry our algebraic multigrid experience with smoothed aggregation shows that the best performance is obtained when piecewise constants are used as grid interpolants (unsmoothed aggregation). We also apply two cycles of the aggregation method to precondition GMRES in order to have a fair comparison to the GGB cycle (see Fig. 1). For all problems, one pre- and post-ILU(0) smoothing iteration is applied on each level, excluding the coarse one. On the coarsest level a direct solve is applied.

The eigensolver used for GGB [1], GGBα and MGGB methods is the implicitly restarted Arnoldi method implemented in ARPACK [19]. We initially compute 10 eigenvectors corresponding to largest magnitude eigenvalues to construct the cycle illustrated in Fig. 1. The accuracy of the eigensolver is set to 10^{-4} and the restarted Arnoldi space is set to 50. The tolerance of the linear solve is set to $\frac{\|r^j\|_2}{\|r^0\|_2} \leqslant 10^{-6}$. We report results for 2D flow in a box.

Tables 4–6 illustrate the convergence behavior of the various preconditioners with GMRES(40) applied to a thermal-convection flow in a box. A no-slip condition is enforced on all surfaces. A hot temperature is set on one side of the box and a cold temperature is set on the other side. We set the Rayleigh number to 1.0×10^5 and the Prandtl number to 1.0. The results are reported in Tables 4–6 for 32×32 elements with 4356 unknowns, 128×128 elements with 66,564 unknowns and 256×256 elements with 264,196 unknowns, respectively. We apply three levels of the aggregation method from ML [26] to the 32×32 box, four levels to the 128×128 box and five levels to the 256×256 box. Note that MGGB recomputes the eigenspace only at certain Newton iterations as indicated in the last column of the tables. The computation is based on the angle between a subspace and a vector described in Section 4.

It is clear from Table 4 that the fastest preconditioner to converge is the smoothed aggregation multilevel method from ML [26], however the minimum number of iterations is obtained by the GGB preconditioner.

Table 4
CPU time and iteration summary for thermal-convection flow in 32×32 box with 4356 unknowns

Preconditioners	*CPU* time (s)			Linear iterations		Eigenvectors	
	Total	Eigensolver	Eigen %	Total	Average	Total	Itr. comp.
ML	8.69	–	–	180	18.0	–	–
GGB	13.88	5.73	41.28	105	10.5	100	1–10
GGBα	10.92	3.02	31.04	163	16.3	14	1–10
MGGB	11.00	2.27	20.63	129	12.9	40	1–4

Three levels of aggregation method are applied.

Table 5
CPU time and iteration summary for thermal-convection flow in 128×128 box with 66,564 unknowns

Preconditioners	*CPU* time (s)			Linear iterations		Eigenvectors	
	Total	Eigensolver	Eigen %	Total	Average	Total	Itr. comp.
ML	324.31	–	–	825	82.5	–	–
GGB	335.30	108.68	32.41	355	35.5	110	1–10
GGBα	326.44	61.85	18.95	456	45.6	13	1–10
MGGB	287.03	37.53	13.07	429	42.9	30	1,2,4

Four levels of aggregation method are applied.

Table 6
CPU time and iteration summary for thermal-convection flow in 256×256 box with 264,196 unknowns

Preconditioners	*CPU* time (s)			Linear iterations		Eigenvectors	
	Total	Eigensolver	Eigen %	Total	Average	Total	Itr. comp.
ML	4389.14	–	–	3873	387.3	–	–
GGB	2329.39	777.61	33.38	838	83.8	150	1–10
GGBα	2935.90	288.19	9.82	1741	174.1	15	1–10
MGGB	2172.93	236.80	10.90	1149	114.9	45	1–2,4

Five levels of aggregation method are applied.

Both MGGB and GGBα methods converge in almost the same *CPU* time, yet MGGB requires fewer iterations. On the 128×128 elements problem illustrated in Table 5, the MGGB method is the fastest method to converge. GGBα also performs well computing only two new eigenvectors (except for the first step). As expected, the GGB method performs the best in terms of iteration count. The performance of the preconditioners on the largest problem, presented in Table 6, is quite interesting. The convergence of ML deteriorates and the number of iterations and *CPU* time for convergence is much higher than those obtained by the GGB family. The best performance in terms of *CPU* time is obtained for MGGB. An important observation is that both GGBα and MGGB methods reduce the amount of work done by the eigensolver compared to the GGB method.

Fig. 7 compares the exact angle $\theta_{\text{exa}} = \angle(Q^j, q_1^{j+1})$ with $q_1^{j+1} \in U^{j+1}$ to the approximate angle $\theta_{\text{app}} = \angle(Q^j, R_{\mathscr{MG}}^{j+1} q_1^j)$ with $q_1^j \in Q^j$ between a subspace and a vector. We compare the angles for a situation where the GGB filter is computed only once in the first iteration. It can be seen that the approximation is valid in the region where the multigrid iteration matrices only slightly vary (Newton iterations six and higher) for the 32×32 and 128×128 problems. Nevertheless, the general behavior is well captured. In the 256×256 box the approximated angle matches with the exact angle.

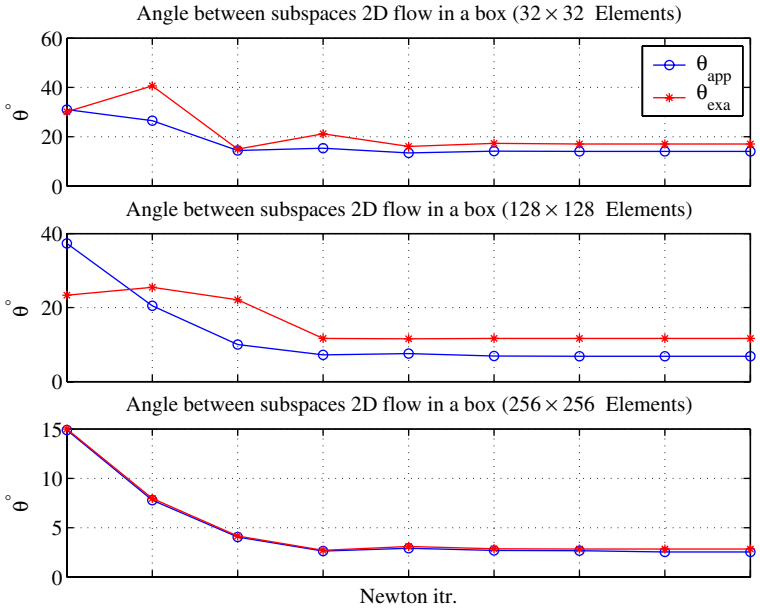

Fig. 7. Approximate and exact angle between a subspace and a vector for the 32×32, 128×128 and 256×256 boxes, respectively.

5.4. Chemical vapor deposition reactor

Finally, we present the convergence of the multigrid solvers applied to a 3D chemical vapor deposition of silicon in a horizontal rotating disk reactor. The mathematical model describing the reactor consist of the incompressible Navier–Stokes equations for a variable-density fluid given in Eqs. (16)–(18) and a species mass balance equation solved for $N_g - 1$ gas-phases species:

$$\rho(u \cdot \nabla) Y_k = \nabla \cdot \mathrm{j}_k + W_k \dot{\omega}_k \quad \text{for } k = 1, \ldots, N_g - 1,$$

where Y_k is the mass fraction of the kth species, j_k is the flux of species k relative to the mass averaged velocity u, and $\dot{\omega}_k$ is the molar rate of production of species k from gas-phase reactions. The nonlinear systems of equations are generated by MPSalsa [23,24] (see [28] for detailed problem description). Due to the chemical reactions, this problem is more challenging to iterative solvers as different physics has different character and the stabilization is applied only to the fluid part.

The problem consist of 8999 hexahedral (eight nodes) elements that corresponds to a problem size of 87,400 unknowns. Each node contains five flow variables and three chemical variables. We use GMRES(40) to accelerate the solvers. We apply four levels of aggregation from ML [26] (unsmoothed aggregation). On

Table 7
CPU time and iteration summary for CVD reactor with 87,400 unknowns

Preconditioners	*CPU* time (s)			Linear iterations		Eigenvectors	
	Total	Eigensolver	Eigen %	Total	Average	Total	Itr. comp.
ML	7173.19	–	–	1461	243.50	–	–
GGB	5502.57	1263.90	22.96	654	109	60	1–6
GGBα	6065.17	688.4	11.35	866	144.33	13	1–3
MGGB	5111.75	633.5	12.39	689	114.83	30	1–3

Four levels of aggregation method are applied.

the fine level we use one symmetric Gauss–Seidel smoothing iteration while on the rest (excluding the coarsest grid) we smooth with an ilu(0). A direct solve is applied on the coarse grid. The GGB filter is constructed from only 10 modes. We use similar measures for GGBα and MGGB as for the thermal-convection problem. The tolerance of the linear solve is set to $\frac{\|r^i\|_2}{\|r^0\|_2} \leqslant 10^{-6}$. The convergence results are reported in Table 7. It is shown that for this problem GGB enhances the aggregation method both in terms of iterations and overall *CPU* time. We note that even though MGGB requires more iterations to converge compared to GGB, the savings in eigensolver calculations makes it significantly faster than the other methods considered. Thus at the end we find MGGB to be the most attractive variant of GGB.

6. Conclusions

The GGB method [1] provides robustness to multilevel methods applied to difficult systems (indefinite and nonsymmetric). The efficiency of the method hinges on the highest eigenmodes computations. We study two heuristic strategies to accelerate the GGB method applied to nonlinear problems. Both strategies, GGBα and MGGB, reuse the previously computed eigenspace based on the angle between a subspace and a vector. Numerical examples clearly show that MGGB performs the best of the methods considered and provides significant time savings.

References

[1] H. Waisman, J. Fish, R.S. Tuminaro, J. Shadid, The Generalized Global Basis (GGB) method, International Journal for Numerical Methods in Engineering 61 (8) (2004) 1243–1269.
[2] A. Brandt, S. Ta'asan, Multigrid methods for nearly singular and slightly indefinite problems, in: W. Hackbusch, U. Trottenberg (Eds.), Lecture Notes in Mathematics 1228: Multigrid Methods II, Springer, Berlin, 1985, pp. 100–122.
[3] J.H. Bramble, D.Y. Kwak, J.E. Pasciak, Uniform convergence of multigrid V-cycle iterations for indefinite and nonsymmetric problems, SIAM Journal of Numerical Analysis 31 (6) (1994) 1746–1763.
[4] Y. Shapira, Multigrid techniques for highly indefinite equations, in: N.D. Melson, T.A. Manteuffel, S.F. McCormick, C.C. Douglas (Eds.), Seventh Copper Mountain Conference on Multigrid Methods, vol. CP 3339, NASA, Hampton, VA, 1996, pp. 689–705.
[5] J.E. Dendy, Black box multigrid for nonsymmetric problems, Applied Mathematics and Computation 13 (3–4) (1983) 261–283.
[6] J. Fish, Y. Qu, A. Suvorov, Towards robust two-level methods for indefinite systems, International Journal for Numerical Methods in Engineering 45 (10) (1999) 1433–1456.
[7] T.E. Giddings, J. Fish, An algebraic two-level preconditioner for asymmetric, positive-definite systems, International Journal for Numerical Methods in Engineering 52 (12) (2001) 1443–1463.
[8] J. Wan, R. Bank, Z. Qu, An energy minimization approach to multigrid for convection dominated problems, in: Eleventh Copper Mountain Conference on Multigrid Methods, 2003.
[9] E. Chow, An unstructured multigrid method based on geometric smoothness, Numerical Linear Algebra with Applications 10 (5-6) (2003) 401–421.
[10] M. Brezina, R. Falgout, S. Maclachlan, T. Manteuffel, S. Mccormick, J. Ruge, Adaptive smooth aggregation (αSA), SIAM Journal on Scientific Computing 25 (6) (2004) 1896–1920.
[11] R.S. Tuminaro, C.H. Tong, J. Shadid, K.D. Devine, D.M. Day, On a multilevel preconditioning module for unstructured mesh Krylov solvers: two level Schwartz, Communications in Numerical Methods in Engineering 18 (6) (2002) 383–389.
[12] H. Elman, V. Howle, J. Shadid, R. Tuminaro, A parallel block preconditioner for the three-dimensional incompressible Navier–Stokes equations, Journal of Computational Physics 187 (2) (2003) 504–523.
[13] C.W. Oosterlee, T. Washio, An evaluation of parallel multigrid as a solver and as a preconditioner for singularly perturbated problems, SIAM Journal of Scientific Computing 19 (1) (1998) 87–110.
[14] B. Carpentieri, I.S. Duff, L. Giraud, A class of spectral two-level preconditioners, SIAM Journal on Scientific Computing 25 (2) (2003) 749–765.
[15] B. Carpentieri, L. Giraud, S. Gratton, Additive and multiplicative two-level spectral preconditioning for general linear systems, Technical Report TR/PA/04/38, CERFACS, 2004.
[16] J. Fish, Y. Qu, Global-basis two-level method for indefinite systems. Part 1: convergence studies, International Journal for Numerical Methods in Engineering 49 (3) (2000) 439–460.

[17] Y. Qu, J. Fish, Global-basis two-level method for indefinite systems. Part 2: computational issues, International Journal for Numerical Methods in Engineering 49 (3) (2000) 461–478.
[18] D.C. Sorensen, Implicitly restarted Arnoldi/Lanczos method for large scale eigenvalue calculations, in: D.E. Keyes, A. Sameh, V. Venkatakrishnan (Eds.), Parallel Numerical Algorithms, Kluwer Academic Publishers, Dordrecht, 1996.
[19] R. Lehoucq, D.C. Sorensen, C. Yang, ARPACK User's Guide: Solution of Large Scale Eigenvalue Problems with Implicitly Restarted Arnoldi Methods, SIAM, 1998.
[20] R. Glowinski, H.B. Keller, L. Reinhart, Continuation-conjugate gradient methods for the least squares solution of nonlinear boundary value problems, SIAM Journal on Scientific and Statistical Computing 6 (4) (1985) 793–832.
[21] Å. Björck, G.H. Golub, Numerical methods for computing angles between linear sub-spaces, Mathematics of Computation 27 (1973) 579–594.
[22] M.E. Argentati, Principal angles between subspaces as related to Rayleigh quotient and Rayleigh Ritz inequalities with applications to eigenvalue accuarcy and an eigenvalue solver, PhD thesis, University of Colorado, 2003.
[23] J. Shadid, H.K. Moffat, S.A. Hutchinson, G.L. Hennigan, K.D. Devine, A.G. Salinger, MPSalsa-A finite element computer program for reacting flow problems. Part 1 – Theoretical development, Technical Report SAND95-2752, Sandia National Laboratories, May, 1996.
[24] A. Salinger, K. Devine, G. Hennigan, H. Moffat, S. Hutchinson, J. Shadid, MPSalsa-A finite element computer program for reacting flow problems. Part 2: User's guide and examples, Technical Report SAND96-2331, Sandia National Laboratories, 1996.
[25] P. Vanek, J. Mandel, M. Brezina, Algebraic multigrid by the smoothed aggregation for second and fourth order elliptic problems, Computing 56 (3) (1996) 179–196.
[26] C. Tong, R.S. Tuminaro, ML 2.0 smoothed aggregation user's guide, Technical Report SAND2001-8028, Sandia National Laboratories, 2000.
[27] E. Anderson, Z. Bai, C. Bischof, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, S. Ostrouchov, D. Sorensen, LAPACK User's Guide, SIAM, 1995.
[28] J.N. Shadid, R.S. Tuminaro, K.D. Devine, G.L. Hennigan, P.T. Lin, Performance of fully coupled domain decomposition preconditioners for finite element transport/reaction simulations, Journal of Computational Physics 205 (1) (2005) 24–47.