Accelerated Generalized Global Basis (GGB) method for nonlinear problems. ⋆

Haim Waisman [a] Jacob Fish [a,*]

[a] *Department of Civil, Mechanical and Aerospace Engineering, Rensselaer Polytechnic Institute, Troy, NY 12180-3590*

Raymond S. Tuminaro [b]

[b] *Sandia National Laboratories, PO Box 969 MS 9217, Livermore, CA 94551*

John Shadid [c]

[c] *Sandia National Laboratories, PO Box 5800 MS 1111, Albuquerque, NM 87185*

Abstract

Two strategies intended to accelerate the performance of the Generalized Global Basis (GGB) method [1] applied to nonlinear systems are presented. The GGB method accelerates a multigrid method by additional coarse grid correction that filters out slowly converging modes. Both strategies reuse 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 maximum principal angle between subspaces spanned between the previously and current prolongation operators. Numerical examples clearly indicate significant time savings in particular for the MGGB scheme.

Key words: Multilevel, Multigrid, Preconditioner, Indefinite, Nonsymmetric, GGB, GMRES

⋆ This work was supported by the ASCI program at Sandia National Laboratories under contract number DE-ACD4-94AL85000.

* Corresponding author: fishj@rpi.edu (Jacob Fish)

1 Introduction

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 and positive definite systems, standard multigrid methods are very efficient solvers due to their optimal complexity (computational 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 area research. [5], [6] utilize multigrid procedures in the context of normal equations. [7] proposes an operator (matrix) dependent black box multigrid for a single partial differential equation on structured grid problems. In [8] the authors employ a special energy minimization interpolation techniques for convection diffusion problems. Recently, an interesting idea to use a "self correcting" multigrid has been proposed [9], [10]. "self correcting" multigrid finds the algebraically smooth error components unresolved by multigrid when applied to the homogeneous problem $Ax = 0$ with a random initial guess, and adjust the coarsening process accordingly. Other approaches include a straight forward application of the multigrid method as a preconditioner to Krylov iterative solvers [11], [12], [13]. However, convergence depends on the type of multigrid method used and the spectrum of the preconditioned system [1], [13], [14]. Consequently, [14] and [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.

This work follows 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 unresolved 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 "non-converging" 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 are to be filtered. The method can

be used as a stand alone solver or as a preconditioner to Krylov methods. Krylov methods reduce the amount of modes needed.
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 Newotn'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 prolongation may be used at the current step or 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 the following section, 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 $GGB\alpha$ and MGGB strategies that employ those ideas. In Section 6, we study performance of the proposed strategies on various problems. Finally, we conclude with some remarks in Section 7.

2 Overview of the Generalized Global Basis (GGB) Method

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

$$Ku = f \tag{1}$$

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

$$S = I - M^{-1}K, \tag{2}$$

with a relaxation procedure $M \in \mathbb{R}^{N\times N}$ and the identity matrix I. Let ν_1 and ν_2 denote the number of pre- and post- smoothing, respectively. If the error after iteration i is $e^i = u - u^i$, than reduction of the error after one V-cycle is controlled by the multigrid iteration matrix $R_{\mathcal{MG}}$, given as

$$e^{i+1} = S^{\nu_2}TS^{\nu_1}e^i = R_{\mathcal{MG}}e^i. \tag{3}$$

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

$$T = I - P(RKP)^{-1}RK, \tag{4}$$

with the prolongation operator from the coarse grid to the fine grid $P : \mathbb{R}^m \rightarrow \mathbb{R}^N$, and the restriction operator $R : \mathbb{R}^N \rightarrow \mathbb{R}^m$. For symmetric systems the

restriction operator is usually taken as the transpose of the prolongation operator i.e. $R = P^T$. 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 Generalized Global Basis (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 highest eigenvalues λ_i of the multigrid iteration matrix in equation (3)

$$R_{\mathcal{MG}}\phi_i = \lambda_i\phi_i \qquad\qquad i = 1, ..., N. \tag{5}$$

The troublesome modes are the highest eigenvalues that are either not converging (indefinite) or "slow-to-converge" modes. The non converging 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

$$Q_f = span\left\{\phi_i\right\}_{i=1}^{k} = \begin{bmatrix} | & & | \\ \phi_1 & \dots & \phi_k \\ | & & | \end{bmatrix}_{N\times k} . \tag{6}$$

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] and [10]. However, as opposed to [9] and [10] the algebraically smooth error components are obtained directly from the eigenvalue problem (5). Figure 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

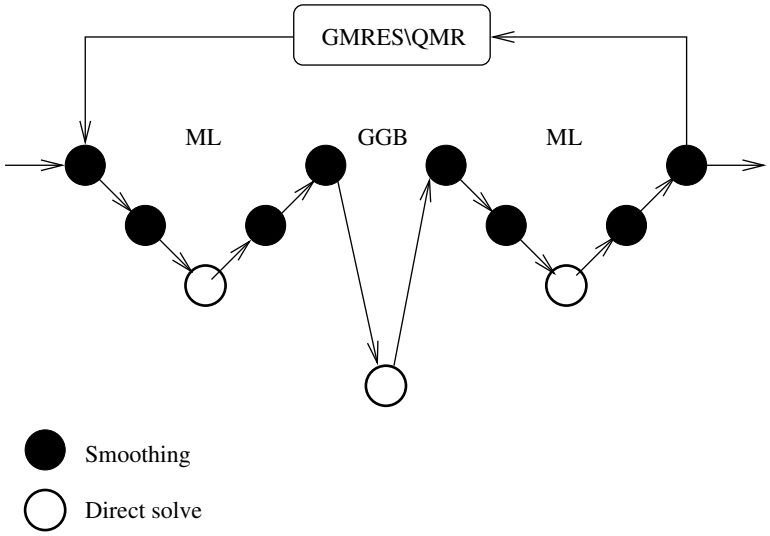


Fig. 1. Generalized Global Basis (GGB) cycle

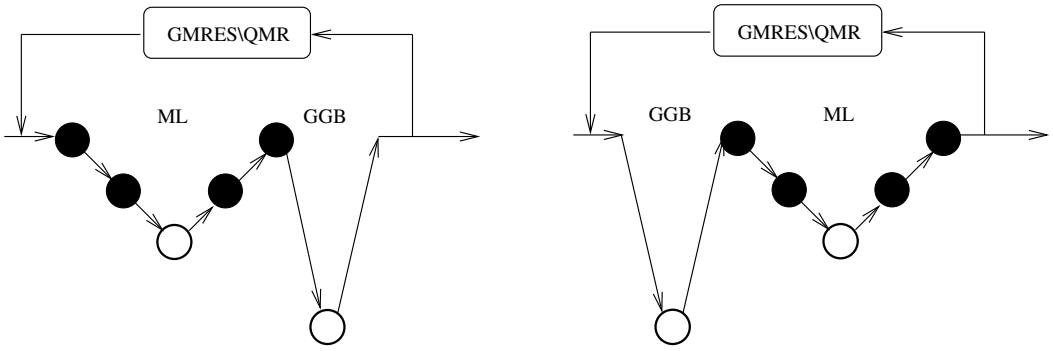
possible for nonsymmetric systems (see Figure 2). The overall error reduction of a single GGB cycle illustrated in Figure 1, without an external accelerator and one smoothing iteration at each level can be written as

$$e^{i+1} = (STS)^{\nu_2} F_{GGB} (STS)^{\nu_1} e^i = R_{\mathcal{MG}}^{\nu_2} F_{GGB} R_{\mathcal{MG}}^{\nu_1} e^i, \tag{7}$$

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

$$F_{GGB} = I - Q_f (Q_f^* K Q_f)^{-1} Q_f^* K, \tag{8}$$

where the prolongation operator Q_f and the restriction Q_f^* are spanned by the highest modes of $R_{\mathcal{MG}}$ given in (6).



(a) GGB filter on the right (b) GGB filter on the left

Fig. 2. Various GGB cycles for nonsymmetric systems

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 later, 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 (5) 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 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 gradually changing, than 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 eigenspace.
To motivate the MGGB approach we consider the following 1D nonlinear boundary value problem on the interval $\Omega = [0, 1]$ with homogeneous Dirichlet boundary conditions

$$\begin{cases} u'' + \alpha u' + \lambda e^u = 0 \\ u(x = 0) = u(x = 1) = 0 \end{cases} . \tag{9}$$

This problem is referred to as the "modified Bratu problem" [20]. The standard "Bratu problem" is obtained for $\alpha = 0$ with two known bifurcated solutions for $\lambda < \lambda_c$, no solutions for $\lambda > \lambda_c$ and a unique solution when $\lambda = \lambda_c$ [21]. To numerically solve (9) the domain is discretized into $N - 1$ segments with equal length $h = \frac{1}{N-1}$. A central difference discretization is employed leading to the following difference scheme

$$F(u_i) = \beta u_{i+1} + \gamma u_{i-1} - 2u_i + \lambda h^2 e^{u_i} = 0, \tag{10}$$

where $\beta = (1 + \frac{\alpha h}{2})$ and $\gamma = (1 - \frac{\alpha h}{2})$. Applying Newton's method to (10) yields the following sequence of linear systems to be solved for the search direction s_j:

$$\mathcal{J} s_i = -F(u_i). \tag{11}$$

$\mathcal{J} \equiv F'(u_i)$ is the Jacobian matrix, given by the following stencil

$$tridiag[\gamma,\ -2 + \lambda h^2 e^{u_i},\quad \beta]. \tag{12}$$

The singular turning point of the curve ($\alpha = 0$) is obtained for $\lambda = \lambda_c = 3.5138307$ [21]. Moreover, it is easily verified that at the turning point exactly

one eigenvalue of $\mathcal{J}$ changes its sign and the system becomes indefinite. For $\alpha > 0$ the linear system becomes nonsymmetric and will therefore pose difficulties to iterative solvers. We apply a standard multigrid method to (11). Figure 3 shows the fine and coarse meshes. We assume an odd number of fine

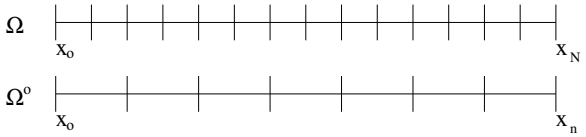


Fig. 3. Fine and coarse meshes

grid points N, with $n = \frac{N+1}{2}$ coarse grid points.
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 $R_{full} = \frac{1}{2}P_{mg}^T$ or injection

$$R_{inject} = \begin{pmatrix} 0\ 1\ 0 & & \\ & 0\ 1\ 0 & \\ & & 0\ 1\ 0 \end{pmatrix}. \tag{13}$$

are used for restriction.
The problem parameters are set to $\lambda = 3$ and $\alpha = 1.5$, with initial guess $u_0 = 2sin(\pi x)$, placing the nonlinear problem (9) in the indefinite and nonsymmetric region. One pre- and post- Gauss-Siedel smoothing iteration is performed. For the GGB filter (8) we chose four eigenvectors corresponding to the largest eigenvalues obtained from solving (5). Figures 4 and 5 show the spectrum and highest modes of the MG_{inj} iteration matrix, respectively, as the nonlinear iteration proceeds. Red circles are used to mark the largest magnitude modes picked for the GGB filter. The figures illustrate that the largestest eigenvalue move significantly as the nonlinear iteration proceeds. This implies, for this problem, that eigen information needs to be recomputed for the GGB method to perform appropriately. In fact, if the initial eigen space computed at the first Newton step is reused throughout the linear sequence, eventually the GGB method fails to converge. While this example illustrates the limitations of eigen reuse, it is worthwhile to point out that the MG_{full} iteration matrix is well behaved. Using only the eigen space information computed at the first Newton iteration, the GGB method converges quite satisfactory.
To summarize, the Bratu exercise illustrates that eigen reuse must be done carefully. In the next Section we develop a set of criteria to automatically determine when the eigen space needs to be re-computed.

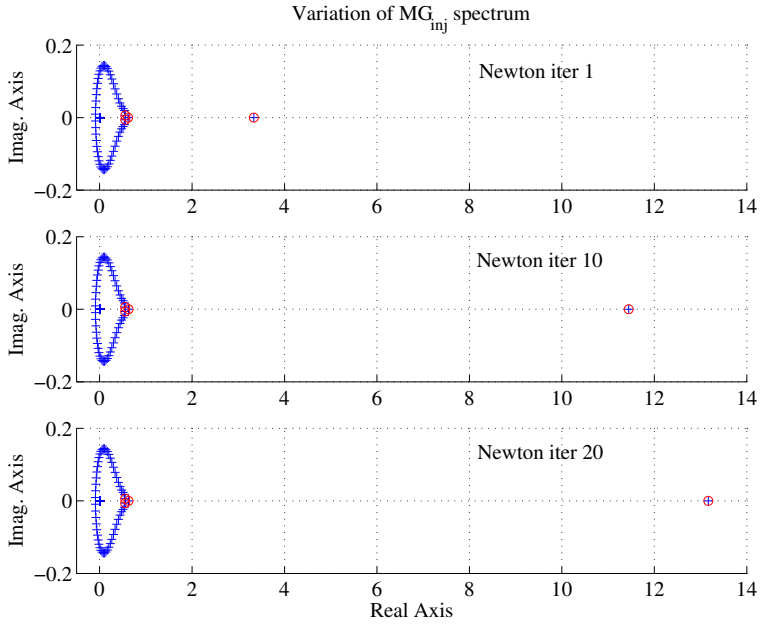


Fig. 4. Variation of the MG_{inj} iteration matrix spectrum during Newton's process for $\lambda = 3$ and $\alpha = 1.8$

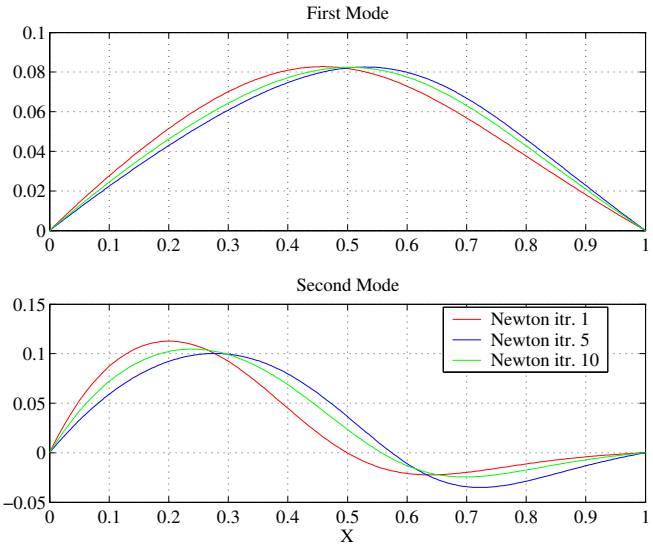


Fig. 5. Variation of first and second highest modes of MG_{inj} during Newton process for $\lambda = 3$ and $\alpha = 1.8$

4 Strategies to reuse eigenspace information

We propose two strategies that reuse already computed prolongation operator. The first is termed GGBα and the second is MGGB. GGBα is based on augmenting new information into a previously computed prolongation operator. In the first Newton iteration a GGB filter is constructed by solving the

eigen problem of the multigrid iteration matrix $R^1_{\mathcal{MG}}$ for k non-converging and "slow-to-converge" eigenvalues

$$R^1_{\mathcal{MG}} q^1_i = \lambda^1_i q^1_i \qquad i = 1, .., k, \tag{14}$$

where superscripts denote the nonlinear iteration count. 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 iteration j, may be written in the following way,

$$Q^j = \left[Q^1, U^2, ..., U^j\right], \tag{15}$$

where

$$Q^1 = \{q^1_1, q^1_2, ..., q^1_k\}, \tag{16}$$

and the U matrices are given by

$$U^j = \{q^j_1, ..., q^j_t\} \qquad j > 1, \tag{17}$$

obtained from

$$R^j_{\mathcal{MG}} q^j_i = \lambda^j_i q^j_i \qquad i = 1, .., t. \tag{18}$$

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 $GGB\alpha$ 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.

Further, if the difference between the two subspaces Q^j and U^{j+1} is close than 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. Computation of principal angles between subspaces is performed in many applications, for example data analysis, random processes, stochastic realization and more [22].

Angles between subspaces are defined in the following way (for more details see [22], [27]).

Let $\mathcal{F}$ and $\mathcal{G}$ be the column space of Q^j and U^{j+1}, respectively, and let $s = dim(\mathcal{F})$ and $t = dim(\mathcal{G})$ with $s \geq k \geq t$. The principal angles $\theta_1, ..., \theta_t \in [0, \frac{\pi}{2}]$ between $\mathcal{F}$ and $\mathcal{G}$ may be defined recursively for $p = 1, ..., t$ by

$$cos(\theta_p) = max_{v \in \mathcal{G}} \; max_{w \in \mathcal{F}} \; v^T w = v_p^T w_p \tag{19}$$

subject to

$$\|v\| = \|w\| = 1, \quad v^T v_i = 0, \quad w^T w_i = 0, \quad i = 1, .., p-1 \tag{20}$$

The vectors $v_1, ..., v_t$ and $w_1, ..., w_t$ are called principal vectors. More explicitly, definition (19) follows if the subspaces $\mathcal{F}$ and $\mathcal{G}$ are orthogonalized, and

rotated such that the inner product between their columns are 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 $R^j_{\mathcal{MG}}$. For this purpose, we choose the Björck-Golub algorithm [22] 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 bases for $range(\mathcal{F})$ and $range(\mathcal{G})$, say computed by a QR factorization, i.e.,

$$\begin{aligned} Q^j &= \tilde{Q}\tilde{R}_1 \\ U^{j+1} &= \tilde{U}\tilde{R}_2 \end{aligned} \tag{21}$$

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

$$\tilde{Q}^T\tilde{U} = Z \begin{pmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_t \end{pmatrix} V^T \qquad 1 \geq \sigma_1 \geq ... \geq \sigma_t \geq 0 \tag{22}$$

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

$$\theta_i = arccos(\sigma_i) \qquad i = 1, ..., t \tag{23}$$

where $0 \leq \theta_1 \leq ... \leq \theta_t \leq \frac{\pi}{2}$ and the columns of $\tilde{Q}Z$ and $\tilde{U}V$ are the principal directions.
For computational purposes, the space of the prolongation Q_j is enriched by new U^{j+1} eigenvectors only if the maximum principal angle $\theta_t \geq \theta_{cr}$, where θ_{cr} is some tolerance angle. This strategy may be very effective if the multigrid iteration matrices vary from iteration to iteration since the enriched space might better capture the highest problematic modes. However, if the iteration matrices only slightly changes than the maximum principal angle will stay below the tolerance and computing t new eigenvectors might not be justified as they do not contribute to the space.
The second strategy, termed MGGB (modified generalized global basis), is based on the full reuse of a previously computed prolongation operator. Again, if the iteration matrices only slightly changes than full reuse may result in satisfactory convergence rate and yet the CPU time savings might be significant. Thus, the goal of this strategy is to avoid the exact computation of t eigenvectors, which was employed by GGBα strategy.
To do this, we measure how far are the vectors $\{q_i^j\}$ from being eigenvectors of $R^{j+1}_{\mathcal{MG}}$ using a Rayleigh quotient type measure

$$\frac{\|R^{j+1}_{\mathcal{MG}}q_i - \rho q_i\|_2}{|\rho|} \leq \delta \qquad i = 1, ..., k \tag{24}$$

where ρ is the Rayleigh quotient defined as $\rho = \frac{q_i^* R_{\mathcal{MG}}^{j+1} q_i}{q_i^T q_i}$, and δ is some small constant. Using the fact that $q_i^* q_i = 1$, in our case, (24) can be rewritten as

$$\frac{\|R_{\mathcal{MG}}^{j+1} q_i - \left(q_i^T R_{\mathcal{MG}}^{j+1} q_i\right) q_i\|}{|q_i^T R_{\mathcal{MG}}^{j+1} q_i|} \leq \delta \qquad i = 1, ..., k. \tag{25}$$

Moreover, measure (25) is related to the acute angle between $R_{\mathcal{MG}}^{j+1} q_i$ and q_i, $\theta = \angle(R_{\mathcal{MG}}^{j+1} q_i, q_i)$ in the following way,

$$\frac{\|R_{\mathcal{MG}}^{j+1} q_i - \rho q_i\|_2}{|\rho|} = \tan\theta. \tag{26}$$

This can be seen by starting with the definition of the acute angle

$$\cos\theta = \angle(R_{\mathcal{MG}}^{j+1} q_i, q_i) = \frac{|q_i^T R_{\mathcal{MG}}^{j+1} q_i|}{\|R_{\mathcal{MG}}^{j+1} q_i\|_2} = \frac{|\rho|}{\|R_{\mathcal{MG}}^{j+1} q_i\|_2}. \tag{27}$$

So,

$$\cos^2\theta = \frac{\rho^*\rho}{\|R_{\mathcal{MG}}^{j+1} q_i\|_2^2} = \frac{\rho^*\rho}{\|R_{\mathcal{MG}}^{j+1} q_i\|_2^2 - \rho^*\rho + \rho^*\rho} \tag{28}$$

Using the relation

$$\|R_{\mathcal{MG}}^{j+1} q_i - \rho q_i\|_2^2 = \|R_{\mathcal{MG}}^{j+1} q_i\|_2^2 - \rho^*\rho. \tag{29}$$

we arrive at

$$\cos^2\theta = \frac{\rho^*\rho}{\|R_{\mathcal{MG}}^{j+1} q_i - \rho q_i\|_2^2 + \rho^*\rho} \tag{30}$$

and (26) follows (for more details see [27]).
One can generalize the acute angle θ to an angle between the vector $R_{\mathcal{MG}}^{j+1} q_i$ and the subspace Q^j (as opposed to a single vector), assuming real eigenvectors. Define a projection of $R_{\mathcal{MG}}^{j+1} q_i$ onto the subspace Q^j

$$y = Q^j \left((Q^j)^T Q^j\right)^{-1} (Q^j)^T R_{\mathcal{MG}}^{j+1} q_i = Q^j (Q^j)^T R_{\mathcal{MG}}^{j+1} q_i. \tag{31}$$

The cosine of the angle between $R_{\mathcal{MG}}^{j+1} q_i$ and Q^j is given by

$$\begin{aligned}
\cos\beta &= \frac{y^T R_{\mathcal{MG}}^{j+1} q_i}{\|y\| \|R_{\mathcal{MG}}^{j+1} q_i\|} = \frac{\left(Q^j (Q^j)^T R_{\mathcal{MG}}^{j+1} q_i\right)^T R_{\mathcal{MG}}^{j+1} q_i}{\|Q^j (Q^j)^T R_{\mathcal{MG}}^{j+1} q_i\| \|R_{\mathcal{MG}}^{j+1} q_i\|} \\
&= \frac{\left((Q^j)^T R_{\mathcal{MG}}^{j+1} q_i\right)^T \left((Q^j)^T R_{\mathcal{MG}}^{j+1} q_i\right)}{\|(Q^j)^T R_{\mathcal{MG}}^{j+1} q_i\| \|R_{\mathcal{MG}}^{j+1} q_i\|} \\
&= \frac{\|(Q^j)^T R_{\mathcal{MG}}^{j+1} q_i\|}{\|R_{\mathcal{MG}}^{j+1} q_i\|}.
\end{aligned} \tag{32}$$

The square of the cosine is further simplified by,

$$\begin{aligned}\cos^2\beta &= \frac{\|(Q^j)^T R^{j+1}_{\mathcal{MG}} q_i\|^2}{\|R^{j+1}_{\mathcal{MG}} q_i\|^2} \\ &= \frac{\left(q_1^T R^{j+1}_{\mathcal{MG}} q_i\right)^2 + ... + \left(q_i^T R^{j+1}_{\mathcal{MG}} q_i\right)^2 + ... + \left(q_k^T R^{j+1}_{\mathcal{MG}} q_i\right)^2}{\|R^{j+1}_{\mathcal{MG}} q_i\|^2} \\ &= \frac{\left(q_1^T R^{j+1}_{\mathcal{MG}} q_i\right)^2 + ... + \rho^2 + ... + \left(q_k^T R^{j+1}_{\mathcal{MG}} q_i\right)^2}{\|R^{j+1}_{\mathcal{MG}} q_i\|^2}.\end{aligned} \tag{33}$$

Substituting (27) into (33) yields,

$$\begin{aligned}\cos^2\beta &= \frac{\left(q_1^T R^{j+1}_{\mathcal{MG}} q_i\right)^2 + ... + \|R^{j+1}_{\mathcal{MG}} q_i\|^2 \cos^2\theta + ... + \left(q_k^T R^{j+1}_{\mathcal{MG}} q_i\right)^2}{\|R^{j+1}_{\mathcal{MG}} q_i\|^2} \\ &= \frac{\left(q_1^T R^{j+1}_{\mathcal{MG}} q_i\right)^2}{\|R^{j+1}_{\mathcal{MG}} q_i\|^2} + ... + \cos^2\theta + ... + \frac{\left(q_k^T R^{j+1}_{\mathcal{MG}} q_i\right)^2}{\|R^{j+1}_{\mathcal{MG}} q_i\|^2}\end{aligned} \tag{34}$$

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

$$\cos^2\beta \geq \cos^2\theta = \frac{1}{\frac{\|R^{j+1}_{\mathcal{MG}} q_i - \rho q_i\|}{|\rho|} + 1}, \tag{35}$$

which relates Rayleigh quotient to an angle between subspaces. Note that the measures proposed for the MGGB method assume $|\lambda_1| \geq ... \geq |\lambda_k| \geq ... \geq |\lambda_N|$, 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 would capture this behavior.
Nevertheless, in most cases where the multigrid iteration matrix only slightly varies from one iteration to the other, the MGGB strategy is found to be the most attractive method.

5 Numerical Results

In order to illustrate the behavior of the MGGB and GGB methods, we apply the solvers to the following nonlinear (and nonsymmetric) problems: A 1D modified Bratu problem and 2D/3D, steady, thermal-convection flows. In the first problem we use a central difference discretization and standard multigrid methods, and in the second the nonsymmetric algebraic systems are generated by the MPSalsa code [23], [24] and a smooth aggregation multilevel method [25] implemented in the ML package [26], is used as preconditioner to the restarted GMRES(m) method.

We apply the following left preconditioners to GMRES for the solution of the modified Bratu problem described in (9). A standard multigrid with restriction based on full weighting and injection described in Section 3, GGB method based on the same standard multigrid method and the corresponding MGGB method. One pre- and post- Gauss-Siedel smoothing sweep is applied to all methods. We don't consider here the GGBα strategy described in Section 4. Table 1 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 $u_0 = 2sin(\pi x)$, yielding a nonlinear problem (9) that is indefinite and nonsymmetric. For the linear solve, we use the following stopping criteria $\frac{\|r^i\|_2}{\|r^0\|_2} \leq 10^{-8}$, where $r^i = f - Ku^i$ is the residual at the inner iteration i. The outer iteration is terminated when $F \leq 10^{-6}$. Results obtained for MGGB methods, use a measure based on an approximate angle between subspaces described in Section (4), with a critical angle for recomputing the GGB filter set to $\theta_{cr} = 20°$. We choose only 4 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 4 modes) cuts in almost half the required number of iterations. It is also clear from Table 1 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. Figure 6 compares the approximate $\theta_{app} = \angle(Q^j, R^{j+1}_{\mathcal{MG}} Q^j)$ and exact angle $\theta_{exa} = \angle(Q^j, U^{j+1})$ between subspaces. For both MGGB cases the prolongation was computed only once. Notice that the convection term is chosen to be less than the one presented in Section 3, however it illustrates well

Table 1
Convergence of various preconditioners to GMRES applied to modified Bratu problems of size 115, 315 and 515, respectively. The problem parameters are set to $\lambda = 3$ and $\alpha = 1.3$.

	Total linear solve iter.			Average No. of iter.		
Preconditioner	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 match and mismatch of the angles. In other words, the subspace $R_{\mathcal{MG}}^{j+1}Q^j$ we use to approximation of the angle θ_{app} in valid when the multigrid iteration matrix only slightly changes as the MG_{full} case.

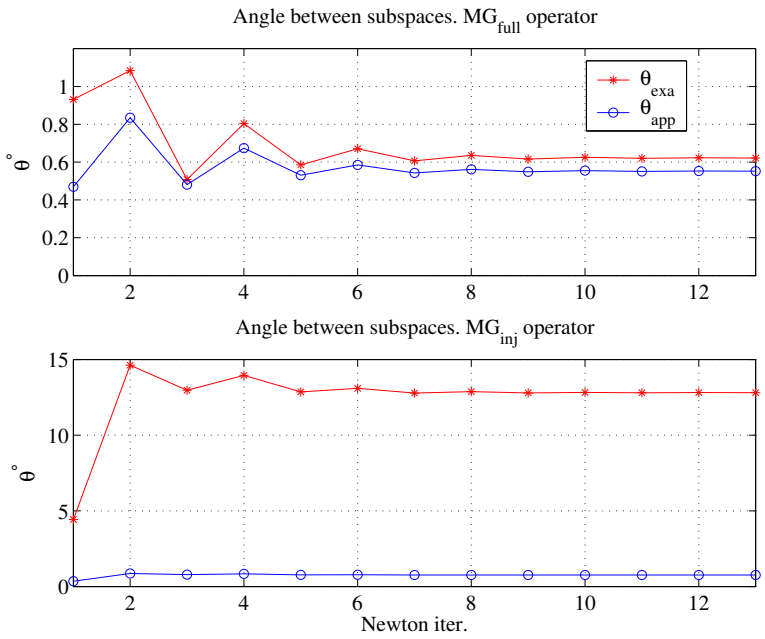


Fig. 6. Approximate and exact angle between subspace for MG_{full} and MG_{inj} iteration matrices ($N = 115$), respectively.

5.2 Steady, thermal-convection flow, preconditioned by smooth aggregation

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

$$\begin{aligned} &\text{Momentum} && \rho\mathbf{u}\cdot\nabla\mathbf{u} - \nabla\cdot\mathsf{T} - \rho\mathbf{g} = 0 && (36)\\ &\text{Total mass} && \nabla\cdot(\rho\mathbf{u}) = 0 && (37)\\ &\text{Thermal energy} && \rho\hat{C}_p\mathbf{u}\cdot\nabla T + \nabla\cdot\mathbf{q} = 0 && (38) \end{aligned}$$

The unknown quantities are $\mathbf{u}$ the fluid velocity vector, P the hydrodynamic pressure and T the temperature. ρ, $\mathbf{g}$, and $\hat{C}_p$ are respectively, the density, the gravity vector and the specific heat at constant pressure. The Boussinesq approximation is used for representing the body force term. The necessary constitutive equations for T and $\mathbf{q}$ are

$$\begin{aligned} &\text{Stress tensor} && \mathsf{T} = -P\mathbf{I} + \Upsilon = -P\mathbf{I} + \mu(\nabla\mathbf{u} + \nabla\mathbf{u}^T) && \\ &\text{Heat flux} && \mathbf{q} = -\kappa\nabla T && (39) \end{aligned}$$

where μ is the viscosity and κ is the thermal diffusivity. (36)-(38) 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 the following preconditioners to a restarted GMRES(m) method. A smooth aggregation multilevel method [25] implemented in ML package [26], a generalized global basis (GGB) method [1] based on the smooth aggregation method from ML, a GGBα method and an MGGB method. We apply a measure based on the angle between subspaces (see Section 4) for the following cases. In GGBα method the angle is used to determine whether the new computed subspace is needed to enrich the prolongation. In all examples we compute two new eigenvectors every nonlinear iteration. If $\theta > 5^\circ$ the new subspace is added. In MGGB we use an approximate angle which is used to predict whether the entire prolongation should be recomputed. If $\theta > 20^\circ$ a new filter is computed. The approximate angle is computed between the previous prolongation Q^j and two vectors given by

$$v_i = R^{j+1}_{\mathcal{MG}} q^j_i \qquad i = 1, 2 \tag{40}$$

where q^j_i are the highest eigenvectors from subspace Q^j. We employ LAPACK subroutines [28] 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 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 Figure 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 ten eigenvectors corresponding to largest magnitude eigenvalues to construct the cycle illustrated in Figure 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 $\frac{||r^i||_2}{||r^0||_2} \leq 10^{-6}$. We report results for flow in three geometries (i) 2D flow in a box (ii) 3D flow between cylinders and (iii) 3D flow in a cube.

5.2.1 2D flow in a box

Tables 2, 3 and 4 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 Rayleigh number to 1.0×10^5 and the Prandtl number to 1.0. The results are reported in Table 2, Table 3 and Table 4 for 32×32 elements with $4,356$ unknowns, 128×128 elements with $66,564$ unknowns and 256×256 elements with $264,196$ unknowns, respectively. We apply 3-levels of the aggregation method from ML [26] to the 32×32 box, 4-levels to the 128×128 box and 5-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 approximate angle between subspaces described in Section **??**.

It is clear from Table 2 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. The MGGB method converged faster and with fewer iterations than GGBα. On the 128×128 elements problem illustrated in Table 3, the MGGB method is the fastest method to converge. GGBα also performs well computing only two new eigenvectors (except for the first step). The GB method performs worse than the other methods. 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 4, are 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.

Table 2
CPU time and iteration summary for thermal-convection flow in 32×32 box with $4,356$ unknowns. 3-levels of aggregation method is applied

	CPU time [sec]			Linear Iterations		Eigenvectors	
Precnd.	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	12.59	3.39	26.92	151	15.1	20	1-10
MGGB	11.00	2.27	20.63	129	12.9	40	1-4

Figure 7 compares the exact angle $\theta_{exa} = \angle(Q^j, U^{j+1})$ to the approximate angle $\theta_{app} = \angle(Q^j, R^{j+1}_{\mathcal{MG}} Q^j)$ between subspaces. We compare the angles for a

Table 3
CPU time and iteration summary for thermal-convection flow in 128×128 box with $66,564$ unknowns. 4-levels of aggregation method is applied

	CPU time [sec]			Linear Iterations		Eigenvectors	
Precnd.	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	305.88	58.20	19.02	411	41.1	18	1-10
MGGB	287.03	37.53	13.07	429	42.9	30	1,2,4

Table 4
CPU time and iteration summary for thermal-convection flow in 256×256 box with $264,196$ unknowns. 5-levels of aggregation method is applied

	CPU time [sec]			Linear Iterations		Eigenvectors	
Precnd.	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	3248.76	272.75	8.40	2058	205.8	17	1-10
MGGB	2172.93	236.80	10.90	1149	114.9	45	1-2,4

situation where the prolongation operator 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 match with the exact angle.

5.2.2 3D flow between two finite length cylinders

In this case the fluid is confined between two cylinders with an outer to inner radius ratio of $\frac{8}{3}$ and a ratio of length to outer diameter of 1. A no-slip condition is enforced on all boundaries. A hot temperature is set to the inner cylinder and a cold temperature is set to the outer cylinder. The ends of the annular region are insulated. The momentum transport, total mass conservation and energy transport are given in equations (36)-(38). Tables 5 and 6 illustrate the convergence behavior of the various preconditioners. The mesh considered in Table 5 consists of 768 elements with $5,400$ unknowns, and $32,768$ elements with $179,520$ unknowns in Table 6. The Prandtl number is set to 1.0 for both

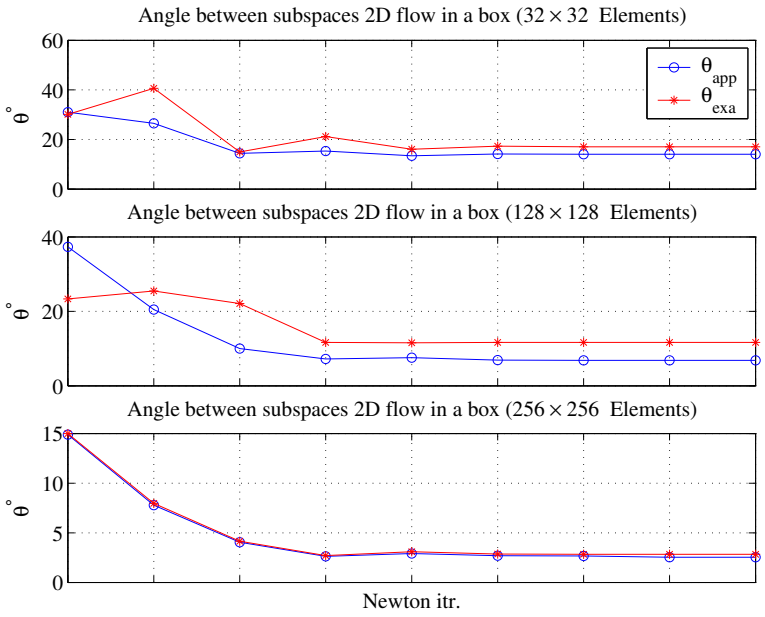


Fig. 7. Approximate and exact angle between subspace for the 32×32, 128×128 and 256×256 boxes, respectively.

problems and the Rayleigh number to 4.0×10^3 and 7.6×10^3 for the small and large problem, respectively. We apply 3-levels of aggregation from ML [26] to the 768 elements problem, and 4-levels to the $179,520$ problem. All methods are applied as preconditioners to GMRES(30) for both problems.
Table 5 indicate that MGGB converge in less CPU time than the rest of the preconditioners. On the other hand the multilevel method by itself does not perform well and converges in about twice the time of MGGB with a factor of more than 10 in iterations. Indeed, applying a GGB type filter accelerates the multilevel tremendously. Again, for the larger problem MGGB performs the best in terms of CPU time. Indeed, both MGGB and GGBα strategies accelerate the GGB method by reusing previous computed prolongation.

Table 5
CPU time and iteration summary for thermal-convection flow between cylinders with $5,400$ unknowns. 3-levels of aggregation method is applied

	CPU time [sec]			Linear Iterations		Eigenvectors	
Precnd.	Total	Eigensolver	Eigen %	Total	Average	Total	Itr. comp.
ML	95.66	-	-	1256	125.6	-	-
GGB	57.93	22.40	38.67	142	14.2	100	1-10
GGBα	54.22	9.11	16.80	346	34.6	23	1-10
MGGB	47.95	10.93	22.79	160	16	50	1-5

Table 6
CPU time and iteration summary for thermal-convection flow between cylinders with 179,520 unknowns. 4-levels of aggregation method is applied

	CPU time [sec]			Linear Iterations		Eigenvectors	
Precnd.	Total	Eigensolver	Eigen %	Total	Average	Total	Itr. comp.
ML	3074.42	-	-	717	51.21	-	-
GGB	4177.03	1697.40	40.64	280	20	140	1-14
GGBα	3435.32	800.27	23.30	313	22.35	33	1-14
MGGB	2847.50	327.99	11.52	280	20	40	1,4-6

6 Conclusions

The Generalized Global Basis (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 strategies to accelerate the GGB method applied to nonlinear problems. Both strategies, GGBα and MGGB, reuse the previously computed eigenspace based on the maximum principal angle between subspaces. Numerical examples clearly show that MGGB outperforms all methods providing significant time savings.

References

[1] H. Waisman, J. Fish, R. S. Tuminaro and J. Shadid, *The Generalized global basis (GGB) method*, International Journal for Numerical Methods in Engineering (2003), to appear.

[2] A. Brandt and S. Taásan, *Multigrid methods for nearly singular and slightly indefinite problems*, in Multigrid Methods II, Hackbusch W., Trottenberg U. In Lecture Notes in Mathematics, Springer-Verlag: Berlin, 1986, pp. 99–121.

[3] J. H. Bramble, D. Y. Kwak and J. E. Pasciak , *Uniform convergence of multigrid V-cycle iterations for indefinite and nonsymmetric problems*, SIAM Journal of Numerical Analysis, 31 (1994), pp. 1746–1763.

[4] Y. Shapira, *Multigrid techniques for highly indefinite equations*, in proceedings 1995 Copper Mountain Conference on Multigrid Methods, 1995.

[5] J. Fish J, Y. Qu and A. Suvorov, *Towards robust two-level methods for indefinite systems*, International Journal for Numerical Methods in Engineering, 45 (1999), pp. 1433–1456.

[6] T. E. Giddings and J. Fish, *An algebraic two-level preconditioner for asymmetric, positive-definite systems*, International Journal for Numeri-

cal Methods in Engineering, 52 (2001), pp. 1443–1463.

[7] J. E. Dendy Jr., *Black box multigrid for nonsymmetric problems*, Applied mathematics and computation, 13 (1983), pp. 261–283.

[8] J. Wan, R. Bank and Z. Qu, *An Energy Minimization Approach to Multigrid for Convection Dominated Problems*, Copper Mountain Conference on Multigrid Methods 2003.

[9] E. Chow, *An unstructured multigrid method based on geometric smoothness*, technical report, Lawrence Livermore National Laboratory UCRL-JC-145075 (2001).

[10] M. Brezina, R. Falgouty, S. Maclachlan, T. Manteuffel, S. Mccormick and J. Ruge, *Adaptive smooth aggregation (αSA)*, submitted to SIAM Journal on Scientific Computing.

[11] R. S. Tuminaro, C. H. Tong, J. Shadid, K. D. Devine KD and D. M. Day, *On a multilevel preconditioning module for unstructured mesh Krylov solvers: Two level Schwartz*, Communications in Numerical Methods in Engineering, 18 (2002), pp. 363–389.

[12] H. Elman, V. Howle, J. Shadid and R. Tuminaro, *A parallel block preconditioner for the three-dimensional incompressible Navier-Stokes equations*, Communications in Numerical Methods in Engineering, 18 (2002), pp. 363–389.

[13] C. W. Oosterlee and T. Washio, *An evaluation of parallel multigrid as a solver and as a preconditioner for singularly perturbated problems*, Journal on scientific computing, 19 (1998), pp.87–110.

[14] B. Carpentieri, I. S. Duff and L. Giraud, *A class of spectral two-level preconditioners*, SIAM Journal on Scientific Computing, 25 (2003), pp. 749–765.

[15] B. Carpentieri, L. Giraud and S. Gratton, *Two-grid spectral preconditioning for general sparse linear systems*, CERFACS Technical Report TR/PA/04/, February 2004.

[16] J. Fish and Y. Qu, *Global-basis two-level method for indefinite systems. Part 1:convergence studies*, International Journal for Numerical Methods in Engineering, 49 (2000), pp. 439–460.

[17] Y. Qu and J. Fish, *Global-basis two-level method for indefinite systems. Part 2:computational issues*, International Journal for Numerical Methods in Engineering, 49 (2000), pp. 461–478.

[18] D. C. Sorensen, *Implicitly restarted Arnoldi/Lanczos method for large scale eigenvalue calculations*, in Parallel Numerical Algorithms, D. E. Keyes, A. Sameh, V. Venkatakrishnan eds, Kluwer: Dordrecht 1996.

[19] R. Lehoucq, D. C. Sorensen and C. Yang C, *ARPACK User's Guide: Solution of large scale eigenvalue problems with Implicitly Restarted Arnoldi methods*, SIAM: April 1998.

[20] P. N. Brown and Y. Saad, *Hybrid Krylov methods for nonlinear systems of equations*, SIAM Journal on Scientific and Statistical Computing, 11 (1990), pp. 450–481.

[21] R. Glowinski, H. B. Keller and L. Reinhart, *Continuation-*

conjugate gradient methods for the least squares solution of nonlinear boundary value problems, SIAM Journal on Scientific and Statistical Computing, 6 (1985), pp. 793–832.

[22] Å. Björck and G. H. Golub, *Numerical methods for computing angles between linear sub- spaces*, Mathematics of Computation, 27 (1973), pp. 579-594.

[23] J. Shadid, H. K. Moffat, S. A. Hutchinson, G. L. Hennigan, K. D. Devine and 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 and J. Shadid *MPSalsa-A finite element computer program for reacting flow problems. Part 2: User's guide*, Technical report SAND96-2331, Sandia National Laboratories, Sep. 1996.

[25] P. Vanek, J. Mandel, M. Brezina, *Algebraic multigrid by the smoothed aggregation for second and fourth order elliptic problems*, Computing (56) (1996), pp. 179–196.

[26] C. Tong and R. S. Tuminaro, *ML 2.0 smoothed aggregation users guide*,Technical report SAND2001-8028, Sandia National Laboratories, Dec. 2000.

[27] 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*, Ph.D Thesis, University of Colorado at Denver 2003.

[28] E. Anderson, Z. Bai, C. Bischof, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, S. Ostrouchov and D. Sorensen, *LAPACK User's Guide*, SIAM, Philadelphia , 1995.