Available online at www.sciencedirect.com

SCIENCE DIRECT®

Journal of Computational and Applied Mathematics 189 (2006) 362–374

JOURNAL OF COMPUTATIONAL AND APPLIED MATHEMATICS

www.elsevier.com/locate/cam

Adaptive preconditioners for nonlinear systems of equations

D. Loghin[a], D. Ruiz[b], A. Touhami[b,*]

[a] *CERFACS, 42 Avenue G. Coriolis, 31057 Toulouse, Cedex, France*
[b] *ENSEEIHT-IRIT, 2 rue C. Camichel, 31071 Toulouse, Cedex, France*

Received 22 September 2004

Abstract

The use of preconditioned Krylov methods is in many applications mandatory for computing efficiently the solution of large sparse nonlinear systems of equations. However, the available preconditioners are often sub-optimal, due to the changing nature of the linearized operator. In this work we introduce and analyse an adaptive preconditioning technique based on the Krylov subspace information generated at previous steps in the nonlinear iteration. In particular, we use an adaptive technique suggested in [J. Baglama, D. Calvetti, G.H. Golub, L. Reichel, Adaptively preconditioned GMRES algorithms, SIAM J. Sci. Comput. 20(1) (1998) 243–269] for restarted GMRES to enhance existing preconditioners with information available from previous stages in the nonlinear iteration. Numerical experiments drawn from domain decomposition techniques and fluid flow applications are used to validate the increased efficiency of our approach.

Keywords: Adaptive preconditioners; Nonlinear systems; Newton's method; Domain decomposition techniques; Augmented systems; Iterative methods

1. Problem description

In this work we are interested in the efficient solution of a sequence of linear systems of the form

$$A_m x_m = b_m, \tag{1}$$

* Corresponding author.
E-mail address: Ahmed.Touhami@enseeiht.fr (A. Touhami).

0377-0427/$ - see front matter © 2005 Elsevier B.V. All rights reserved.
doi:10.1016/j.cam.2005.04.060

where A_m is large and sparse. These problems usually arise from some linearization of a nonlinear system of equations of the form

$$F(x) = 0, \tag{2}$$

where $F : \mathbb{R}^n \to \mathbb{R}^n$ and $x \in \mathbb{R}^n$. For example, in the case of Newton's linearization of (2), A_m is some (preconditioned) approximation to the Jacobian matrix, while x_m is the new update direction and b_m is the negative residual.

In the following, we make the following assumptions:

(i) A_m are preconditioned matrices with a spectrum largely clustered;
(ii) $A_{m+1} = A_m + \mathscr{E}_m$ where $\|\mathscr{E}_m\| \to 0$ as $m \to \infty$;
(iii) we employ GMRES to solve (1).

The choice of iterative method is essential in the design of our preconditioners. In particular, ours is not the first attempt to recycle Krylov information generated by GMRES in order to accelerate convergence. Two main recurring 'recycling' themes are convergence acceleration/preconditioning in the context of (i) restarted GMRES [1–4,11–13], and, more recently, (ii) linear systems with multiple right-hand sides [5,7]. For symmetric and positive-definite problems, similar approaches can be found in [15,18,20]. We note, also, the application of these ideas to sequences of systems of type (1) [9,14].

One distinguishing feature in the quest for an adaptive preconditioner is the idea of subspace invariance. If available, one can either deflate (or shift to one) eigenvalues of the system matrix associated with this subspace. The resulting eigenvalue distribution may thus be more favourable to convergence. In this spirit, various deflation techniques and invariance-seeking algorithms have been devised. However, for realistic applications the construction of such subspaces can be rather costly.

Our approach, based on [1], is more simplistic, but achieves one of the aims of the above invariance-inspired techniques—that of shifting eigenvalues to one. As we show in the next section, this is achieved directly (without further searches), given the GMRES information in the form of a Hessenberg matrix and an Arnoldi basis. Moreover, our technique is positive-definiteness preserving.

The paper is structured as follows. In the next section we introduce our preconditioner and discuss some of its properties. We also present and discuss a simple adaptive preconditioning approach for the solution of (1). In particular, we describe how the use of block-preconditioners with GMRES can lead to cost- and performance-efficient implementations of our adaptive preconditioners. Finally, to illustrate the efficiency of our technique, we use two generic examples: domain decomposition for nonlinear elliptic problems and the steady-state, incompressible Navier–Stokes equations.

2. Adaptive preconditioners

Let $A \in \mathbb{R}^{n\times n}$ and let $V^{\mathrm{T}}AV = H$ denote a Hessenberg decomposition of A, where $V \in \mathbb{R}^{n\times n}$ is an orthonormal matrix with columns v_i, $1 \leqslant i \leqslant n$. Let $V = [V_k \;\; V_{n-k}]$ denote a partition of the column space of V such that $V_k^{\mathrm{T}}AV_k = H_k \in \mathbb{R}^{k\times k}$ is a nonsingular, irreducible Hessenberg matrix (i.e., with no zero subdiagonal entries). Writing $h = H_{k+1,k}$, A satisfies

$$AV_k = V_k H_k + h v_{k+1} e_k^{\mathrm{T}}$$

and

$$V^{\mathrm{T}}AV = \begin{pmatrix} H_k & V_k^{\mathrm{T}}AV_{n-k} \\ he_1e_k^{\mathrm{T}} & V_{n-k}^{\mathrm{T}}AV_{n-k} \end{pmatrix} =: \begin{pmatrix} H_k & F \\ hE & G \end{pmatrix},$$

where, with an abuse of notation, e_1, e_k are respective columns of I_{n-k}, I_k. Define now

$$M_\alpha := I_n - V_kV_k^{\mathrm{T}} + V_kH_kV_k^{\mathrm{T}} + \alpha v_{k+1}v_k^{\mathrm{T}}$$

with inverse

$$M_\alpha^{-1} = I_n - V_kV_k^{\mathrm{T}} + (V_k - \alpha v_{k+1}e_k^{\mathrm{T}})H_k^{-1}V_k^{\mathrm{T}}. \tag{3}$$

Our preconditioning approach is based on the following result.

Proposition 2.1. *Let A have the Hessenberg decomposition $V^{\mathrm{T}}AV = H$ and let M_α be defined as in* (3). *For all $\alpha \neq h$, $\lambda = 1$ is an eigenvalue ofAM_α^{-1} with multiplicity $k-1$. If $\alpha = h$, then*

$$\Lambda(AM_h^{-1}) = \{1\} \cup \Lambda(S(h)),$$

where $\lambda = 1$ has multiplicity k and

$$S(h) := V_{n-k}^{\mathrm{T}}AV_{n-k} - he_1e_k^{\mathrm{T}}H_k^{-1}V_k^{\mathrm{T}}AV_{n-k}.$$

Finally, if A is positive definite, then AM_α^{-1} is positive definite provided $h-\alpha$ is sufficiently small.

Proof. Since

$$V^{\mathrm{T}}AV = \begin{pmatrix} H_k & F \\ hE & G \end{pmatrix}, \quad V^{\mathrm{T}}M_\alpha V = \begin{pmatrix} H_k & 0 \\ \alpha E & I_{n-k} \end{pmatrix},$$

the eigenvalues of AM_α^{-1} satisfy

$$\begin{aligned} &H_kx + Fy = \lambda H_kx, \\ &hEx + Gy = \lambda\alpha Ex + \lambda y. \end{aligned}$$

Thus, if $\alpha \neq h$, $(e_i^{\mathrm{T}}, 0)^{\mathrm{T}}$, $1 \leqslant i \leqslant k-1$, are eigenvectors corresponding to $\lambda = 1$. If $\alpha = h$, $(e_k^{\mathrm{T}}, 0)^{\mathrm{T}}$ also is an eigenvector for $\lambda = 1$, while for $\lambda \neq 1$, elimination of x from the first equation leads to the eigenvalue equation $S(h)y = (G - hEH_k^{-1}F)y = \lambda y$.

Assume now that A is positive definite. Then the Schur complement $S(h)$ is positive definite. To see this, consider the real Schur decomposition $A = Q^{\mathrm{T}}TQ$; then $0 < x^{\mathrm{T}}Q^{\mathrm{T}}TQx = z^{\mathrm{T}}A^{-1}z$, $z = Q^{\mathrm{T}}TQx$. Since the inverse of A has the form

$$A^{-1} = \begin{pmatrix} * & * \\ * & S(h)^{-1} \end{pmatrix},$$

the choice $z = (0, w^{\mathrm{T}})^{\mathrm{T}}$ yields $0 < z^{\mathrm{T}}A^{-1}z = w^{\mathrm{T}}S(h)^{-1}w$ and hence $S(h)$ is positive-definite.

Now, the non-unit eigenvalues λ of AM_α^{-1} satisfy

$$(S(h) + (h-\alpha)\frac{\lambda}{\lambda - 1}EH_k^{-1}F)y = \lambda y$$

and the final statement follows from the positive-definiteness of $S(h)$ and standard perturbation theory [10]. □

Remark 2.1. We note here that our result is different, and in a sense more general, than that of Baglama et al. who considered the same preconditioned system under the assumption that V spans an invariant subspace of A. Under this assumption, the preconditioner M leaves unchanged $n-k$ eigenvalues of A and shifts the other k to 1 (see [1]).

Remark 2.2. The matrix in the last eigenvalue problem in the proof is a rank-one perturbation of $G = V_{n-k}^{\mathrm{T}} A V_{n-k}$; moreover, only the first row of G is modified. As an immediate corollary, an application of Gerschgorin's theorem reveals that the union of disks where the eigenvalues lie remains almost unchanged, with at most one having a different center and radius.

Let now $A_\varepsilon = A + \mathscr{E}$ denote a perturbation of A with $\|\mathscr{E}\| \leqslant \varepsilon$. The ε-pseudospectrum of A is the set

$$\Lambda_\varepsilon(A) = \{z \in \mathbb{C} : z \in \Lambda(A + \mathscr{E}), \text{ for some } \mathscr{E} \text{ with } \|\mathscr{E}\| \leqslant \varepsilon\}, \tag{4}$$

whose contour length $\mathscr{L}$ enters the following standard bound on the convergence of GMRES [19]:

$$\|r^k\| \leqslant (2\pi\varepsilon)^{-1} \mathscr{L}(\partial\Lambda_\varepsilon(A)) \min_{p_k(0)=1} \max_{z\in\Lambda_\varepsilon(A)} |p_k(z)| \|r^0\|. \tag{5}$$

The following result describes the suitability of M as a preconditioner for A_ε.

Proposition 2.2. *Let A_ε, M be defined as above. Then the δ-pseudospectrum of $A_\varepsilon M^{-1}$ satisfies*

$$\Lambda_\delta(A_\varepsilon M^{-1}) \subseteq \Lambda_{\tilde{\varepsilon}+\delta}(AM^{-1})$$

for all $\delta \geqslant 0$, where $\tilde{\varepsilon} = \varepsilon/\sigma_{\min}(\widehat{H}_{k+1})$, where $\widehat{H}_{k+1} \in \mathbb{R}^{k+1\times k+1}$ is the Hessenberg matrix

$$\widehat{H}_{k+1} = \begin{pmatrix} H_k & 0 \\ \alpha e_k^{\mathrm{T}} & 1 \end{pmatrix}.$$

Proof. Let $F \in \mathbb{R}^{n\times n}$ such that $\|F\| \leqslant \delta$. Since $\Lambda(A_\varepsilon M^{-1} + F) = \Lambda(AM^{-1} + \mathscr{E}M^{-1} + F)$ with $\|\mathscr{E}M^{-1} + F\| \leqslant \varepsilon/\sigma_{\min}(\widehat{H}_{k+1}) + \delta$, the result follows from definition (4). □

Thus, one can use (5) to relate the GMRES residuals r_ε^k corresponding to coefficient matrix $A_\varepsilon M^{-1}$ to the tolerance tol used for the solution of the problem with coefficient matrix AM^{-1}. Assuming

$$\|r^k\|/\|r^0\| \leqslant (2\pi(\tilde{\varepsilon}+\delta))^{-1} \mathscr{L}(\partial\Lambda_{\tilde{\varepsilon}+\delta}(AM^{-1})) \min_{p_k(0)=1} \max_{z\in\Lambda_{\tilde{\varepsilon}+\delta}(AM^{-1})} |p_k(z)| \leqslant \mathrm{tol},$$

we find using Proposition 2.2

$$\begin{aligned} \|r_\varepsilon^k\|/\|r_\varepsilon^0\| &\leqslant (2\pi\delta)^{-1} \mathscr{L}(\partial\Lambda_\delta(A_\varepsilon M^{-1})) \min_{p_k(0)=1} \max_{z\in\Lambda_\delta(A_\varepsilon M^{-1})} |p_k(z)| \\ &\leqslant (2\pi\delta)^{-1} \mathscr{L}(\partial\Lambda_{\tilde{\varepsilon}+\delta}(AM^{-1})) \min_{p_k(0)=1} \max_{z\in\Lambda_{\tilde{\varepsilon}+\delta}(AM^{-1})} |p_k(z)| \\ &\leqslant (1 + \tilde{\varepsilon}/\delta)\mathrm{tol}. \end{aligned}$$

Thus, either a small value of ε or a large value of δ can ensure that preconditioning with M leads to convergence to a fixed tolerance in approximately the same number of steps for all perturbations $\mathscr{E}$ of maximum norm ε. We note here that larger values of δ would correspond to a large GMRES tolerance, which is often employed in the early stages of Newton–Krylov algorithms [6].

The application of the above results to (2) is evident: we solve $A_1 x_1 = b_1$ using GMRES, which yields a preconditioner $M_{(1)}$ of form (3). We then employ GMRES to solve a system with preconditioned matrix $(A_1 + \mathscr{E}_1)M_{(1)}^{-1} = A_2 M_{(1)}^{-1}$ and construct $M_{(2)}$. The general preconditioner for $A_m x_m = b_m$ is then given by

$$M_m^{-1} = M_{m-1}^{-1} M_{(m-1)}^{-1}, \quad m > 2, \tag{6}$$

with $M_2 = M_{(1)}$. The above definition raises the concern of cost as well as storage. Each of the preconditioners M_m requires the storage of, and multiplication by, bases V_m of length k_m. While we have control over the choice of k_m, smaller values usually lead to deterioration in performance. Our approach to circumventing this problem is to work with matrices that have, or have been permuted to, a block structure. Then a Schur complement approach coupled with a suitable starting guess leads to a GMRES algorithm essentially for the Schur complement problem, with the advantage that the resulting Arnoldi basis is small. An important example is the class of domain decomposition methods which lend themselves naturally to this strategy. Below we consider in some detail a generic block algorithm which is both cost- and storage-efficient while preserving the advantages of the preconditioning strategy described above.

2.1. A right-preconditioned block algorithm

Let $A \in \mathbb{R}^{n\times n}$ have the following block-structure:

$$A = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix}, \tag{7}$$

where $A_{ij} \in \mathbb{R}^{n_i \times n_j}$, $i, j = 1, 2$, with $n_2 \ll n_1$ and such that A_{11} is invertible. Moreover, we assume that there exists an efficient algorithm for applying the action of the inverse of A_{11}. This is the case for domain decomposition methods, for example where A_{11} is a block diagonal matrix, with blocks obtained by discretizing some PDE problem on subdomains, while A_{22} is the matrix corresponding to the boundary nodes. We also encounter the same structure in many applications of saddle-point type, whether in the field of PDE (mixed finite-element discretizations of systems of PDE) or optimization (constraint minimization programmes); moreover, in these applications also, the action of the inverse of A_{11} can often be performed efficiently.

Consider now the following block-triangular preconditioner:

$$P_R = \begin{pmatrix} A_{11} & A_{12} \\ 0 & \widehat{S} \end{pmatrix}, \tag{8}$$

where $\widehat{S}$ is an approximation to the Schur complement $S = A_{22} - A_{12}A_{11}^{-1}A_{21}$. Since

$$AP_R^{-1} = \begin{pmatrix} I & 0 \\ A_{12}A_{11}^{-1} & S\widehat{S}^{-1} \end{pmatrix},$$

an initial guess of the form

$$x^0 = P_R^{-1} b$$

yields an initial residual of the form $r^0 = (0, \tilde{r}^0)^{\mathrm{T}}$. If we seek r^k in the Krylov space $\mathscr{K}(AP_R^{-1}, r^0, k)$, then all residuals will take the same form $r^k = (0, \tilde{r}^k)^{\mathrm{T}}$. Moreover, since $\mathscr{K}(AP_R^{-1}, r^0, k) \equiv \mathrm{span}_{0 \leqslant j \leqslant k-1}\{r^j\}$, the resulting Arnoldi basis is of the form

$$V_k = \begin{pmatrix} 0 \\ \tilde{V}_k \end{pmatrix}.$$

Thus, one only needs to store and work with a basis $\tilde{V}_k$ of length n_2. In particular, for domain decomposition methods $n_2 \sim \mathrm{O}(n_1^{1/2})$; thus, the cost of GMRES preconditioned by the adaptive preconditioner M given in (6) is $\mathrm{O}(n)$ provided

$$km^2 \sim \mathrm{O}(n^{1/2}),$$

where $k = \max_{1 \leqslant i \leqslant m} k_i$ is the maximum size of the Arnoldi bases employed in M and m is the number of nonlinear steps. In practice, for many nonlinear applications of interest the value of m is typically small ($m \ll n^{1/2}$) while the preconditioner combination MP_R will guarantee a small value of k.

3. Numerical experiments

To illustrate the effectiveness of the approach described in Section 2, we consider two applications arising from the discretization of partial differential equations: a domain decomposition approach for solving a nonlinear reaction-convection-diffusion problem and the example of Navier–Stokes equations discretized with mixed finite elements. Both problems were discretized as

$$A(x)x = (\nu L + N(x))x = b,$$

where ν is a small parameter and L, N denote the linear and nonlinear parts of A. In both cases we employ two linearization methods: Picard (fixed-point) and Newton. Thus, we solved the sequence of linear systems

$$A_m x^{m+1} := (A(x^m) - \rho T(x^m))x^{m+1} = b - \rho T(x^m)x^m =: b_m,$$

where $T(x^m) = N(x^m) - N'(x^m)$, and N' denotes the Jacobian of N with respect to x. Picard and Newton linearizations correspond to choices $\rho = 0$ and $\rho = 1$, respectively. We employ the following stopping criterion suggested in [6] for the Newton iteration: at each nonlinear step m we stop after k iterations of GMRES if the residual r^k satisfies

$$\|r^k\| / \|b_m - A_m x^m\| \leqslant c \|b_m - A_m x^m\|^q, \tag{9}$$

where the choice for c, q varies from problem to problem.

3.1. Nonlinear reaction-convection-diffusion problem

Consider the finite-element discretization of

$$\begin{aligned} -\nu\Delta u + (\vec{b}\cdot\nabla)u + u^2 &= f \quad \text{in } \Omega, \\ u &= 0 \quad \text{on } \Gamma, \end{aligned} \tag{10}$$

where $f=1$, $\Omega=[-1,1]^2$ and $\vec{b}=(2y(1-x^2), -2x(1-y^2))$. The problem was solved for $\nu=1, 1/10$ via a domain decomposition approach, using 4, 16 and 64 subdomains. We stopped the nonlinear iteration when the criterion $\|b_m - A_m x^m\|/\|b_0 - A_0 x^0\| \leqslant 10^{-6}$ was satisfied. We used the GMRES stopping criterion (9) with $c=10^{-2}$ and $q=4/5$. The initial guess for the nonlinear iteration was the zero vector. The structure of the linearized matrices A_m is that given in (7), where A_{11} is block-diagonal, with blocks corresponding to nodes interior to each subdomain—this corresponds to a so-called non-overlapping Dirichlet–Dirichlet domain decomposition method. Since the problem is nonsymmetric, we used as a first preconditioner P_m the block-triangular matrix given in (8), where we naively chose the sparse approximation

$$\widehat{S} = A_{22} - A_{21}\,\mathrm{diag}(A_{11})^{-1}A_{12},$$

in order to test our preconditioning approach. In this case, the eigenvalues of the preconditioned system are either equal to one or equal to the eigenvalues of the preconditioned Schur complement $S\widehat{S}^{-1}$. For elliptic problems, it is known [17] that the condition number of S is of order $h^{-1}H^{-1}$ where $h \sim n^{1/2}$ is the size of the mesh and H is the average size of the subdomains. This typically leads to both mesh- and domain-dependent convergence. As we see below, the choice $\widehat{S}$ appears to remove only the second dependence, given our iterative strategy (9) for the nonlinear process.

While various domain decomposition algorithms remove this dependence for certain applications, their design is often problem-based and may require multi-level information in order to attain optimality. On the other hand, our adaptive technique is a black-box preconditioner which can either replace or enhance a Schur complement preconditioner. Moreover, as demonstrated below, our adaptive preconditioners appear to remove both mesh- and domain-dependence.

The results for the Picard iteration are presented in Tables 1 and 3. We compared the two adaptively constructed preconditioners M_m corresponding to $\alpha = 0$ and $\alpha = h$ (cf. (3)). The average number of iterations per nonlinear step (also displayed) excludes the first Picard step. This allows for a clearer and fairer comparison between P_m and $M_m P_m$, since $M_m = I$ for $m = 1$. The corresponding savings and additional storage are presented in Tables 2 and 4 (Tables 1–4). In particular, we chose to calculate the cost of the adaptive preconditioners only in matrix-vector products (matvecs) and add it to the overall number of matrix-vector products. The savings presented are thus the relative reduction in matvecs and precvecs (matvecs involving P_m^{-1}). The additional storage calculation also used as a reference the storage required by the system matrix. We note here that the cost or storage of the first preconditioner P_m is not of interest here and was not evaluated.

The performance of the adaptive technique can be summarized as follows:

(i) the mesh- or domain-dependence is virtually removed;
(ii) the savings increase with the size of the problem;
(iii) relative additional storage decreases with the size of the problem.

Table 1
Total/average number of GMRES iterations with and without adaptive preconditioning—nonlinear elliptic problem using Picard linearization

$v = 1$		$A_m P_m^{-1}$	$A_m P_m^{-1} M_m^{-1}$	
#dom	n	# its	# its ($\alpha = 0$)	# its ($\alpha = h$)
4	11,425	131/**24.2**	41/**6.2**	38/**5.6**
	45,377	181/**33.4**	54/**8.0**	50/**7.2**
	180,865	310/**48.5**	80/**10.2**	73/**9.0**
16	11,585	191/**35.6**	57/**8.8**	52/**7.8**
	45,953	272/**50.6**	76/**11.4**	72/**10.6**
	183,041	467/**73.5**	113/**14.5**	107/**13.5**
64	16,641	195/**35.4**	53/**7.0**	48/**6.0**
	66,049	283/**51.4**	72/**9.2**	68/**8.4**
	263,169	392/**71.0**	97/**12.0**	92/**11.0**

Table 2
Savings in precvecs ($P_m^{-1} v$) and matvecs ($A_m v$) and additional storage ($\times nz(A_m)$)—nonlinear elliptic problem using Picard linearization

$v = 1$		$\alpha = 0$			$\alpha = h$		
#dom	n	$P_m^{-1} v$ (%)	$A_m v$ (%)	$\times nz(A_m)$	$P_m^{-1} v$ (%)	$A_m v$ (%)	$\times nz(A_m)$
4	11,425	69	65	0.10	70	68	0.11
	45,377	70	68	0.06	72	71	0.07
	180,865	74	73	0.05	76	75	0.05
16	11,585	70	60	0.35	72	63	0.36
	45,953	72	66	0.24	74	67	0.24
	183,041	76	71	0.18	77	73	0.18
64	16,641	73	55	0.73	75	60	0.75
	66,049	75	63	0.50	76	65	0.51
	263,169	75	68	0.34	76	70	0.34

In fact, for this application the improvement in convergence is remarkable, with savings between 69–93% for precvecs and 55–90% for matvecs, while additional storage is minimal (a fraction of the storage for the system matrix between 0.1–0.3 for the finest mesh).

We note also here that the difference in performance between the case $v = 1$ and $v = \frac{1}{10}$ is explained through the convergence of the Picard iteration. When $v = 1$ the number of Picard iterations varied between 6 and 7, while for $v = \frac{1}{10}$ we needed 33 iterations. In both cases, the number of GMRES iterations is

Table 3
Total/average number of GMRES iterations with and without adaptive preconditioning—nonlinear elliptic problem using Picard linearization

$\nu = \frac{1}{10}$		$A_m P_m^{-1}$	$A_m P_m^{-1} M_m^{-1}$	
#dom	*n*	# its	# its ($\alpha = 0$)	# its ($\alpha = h$)
4	11,425	543/**16.7**	89/**2.5**	79/**2.1**
45,377	775/**23.8**	111/**3.03**	89/**2.3**	
	180,865	1,099/**33.8**	121/**3.2**	102/**2.6**
16	11,585	971/**29.7**	118/**3.1**	99/**2.4**
	45,953	1,375/**42.1**	144/**3.6**	117/**2.8**
	183,041	1,934/**59.3**	173/**4.2**	148/**3.5**
64	16,641	1,104/**33.7**	114/**2.8**	101/**2.3**
	66,049	1,589/**48.5**	130/**2.9**	119/**2.6**
	263,169	2,266/**69.2**	173/**3.8**	145/**2.9**

Table 4
Savings in precvecs ($P_m^{-1} v$) and matvecs ($A_m v$) and additional storage ($\times nz(A_m)$)—nonlinear elliptic problem using Picard linearization

$\nu = \frac{1}{10}$		$\alpha = 0$			$\alpha = h$		
#dom	*n*	$P_m^{-1} v$ (%)	$A_m v$ (%)	$\times nz(A_m)$	$P_m^{-1} v$ (%)	$A_m v$ (%)	$\times nz(A_m)$
4	11,425	84	80	0.25	85	81	0.31
	45,377	86	84	0.15	89	87	0.17
	180,865	89	88	0.08	91	90	0.09
16	11,585	88	78	0.84	90	81	0.94
	45,953	90	84	0.51	91	87	0.53
	183,041	91	88	0.31	92	90	0.32
64	16,641	90	73	1.73	91	74	2.05
	66,049	92	84	0.99	93	85	1.16
	263,169	92	87	0.66	93	90	0.68

similar over the first few Picard steps, with a very rapid decrease to a steady requirement of only a few GMRES iterations (2–4) per nonlinear step. Since the case $\nu = \frac{1}{10}$ requires far more nonlinear steps, the outcome is a very low average number of iterations.

For the case of Newton's iteration a similar performance behaviour was noticed which is why we chose not to include the results here. However, we employed Newton's method for the case of a more difficult problem, the Navier–Stokes equations considered below.

3.2. Navier–Stokes example

As a second application of interest, we considered the example of the steady-state incompressible Navier–Stokes equations in an open bounded domain $\Omega \subset \mathbb{R}^d$

$$-\nu\Delta u + (u \cdot \nabla)u + \nabla p = f \quad \text{in } \Omega, \tag{11a}$$

$$\nabla \cdot u = 0 \quad \text{in } \Omega, \tag{11b}$$

$$u = 0 \quad \text{on } \Gamma, \tag{11c}$$

where u is the velocity, p is the pressure and ν the viscosity parameter. The resulting nonlinear systems have the following saddle-point structure (see, e.g., [16]):

$$0 = F(x) = F(u, p) = \begin{pmatrix} f \\ 0 \end{pmatrix} - \begin{pmatrix} K(u) & B^{\mathrm{T}} \\ B & 0 \end{pmatrix} \begin{pmatrix} u \\ p \end{pmatrix}, \tag{12}$$

where $K(u) = \nu L + N(u) \in \mathbb{R}^{n_1 \times n_1}$ is nonsymmetric and possibly indefinite and $B^{\mathrm{T}} \in \mathbb{R}^{n_1 \times n_2}$ has nontrivial kernel spanned by the constant vector. Our two linearizations of (12) yield a sequence of systems of the form

$$A_m x^{m+1} := \begin{pmatrix} K_m & B^{\mathrm{T}} \\ B & 0 \end{pmatrix} \begin{pmatrix} u^{m+1} \\ p^{m+1} \end{pmatrix} = \begin{pmatrix} f \\ 0 \end{pmatrix} - \begin{pmatrix} \rho T(u^m) & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} u^m \\ p^m \end{pmatrix} =: b_m,$$

where $K_m = K(u^m) - \rho T(u^m)$ and $T(u^m) = N(u^m) - N'(u^m)$, with N the nonlinear part of K. As a first level of preconditioning we employ the following block preconditioner:

$$P_m = \begin{pmatrix} \widehat{K}_m & B^{\mathrm{T}} \\ 0 & -\widehat{S} \end{pmatrix}, \tag{13}$$

where $\widehat{K}_m$ approximates K_m and $\widehat{S}$ the negative Schur complement $S = BK_m^{-1}B^{\mathrm{T}}$. We used an inner GMRES algorithm with block Gauss–Seidel preconditioning to solve systems with $\widehat{K}_m$, while for $\widehat{S}$, we employed the same choice as in [8] which given by

$$\widehat{S}^{-1} = M_p^{-1} F_p A_p^{-1},$$

where M_p, F_p, A_p are, respectively, mass, convection-diffusion and laplacian matrices assembled on the pressure space (see [8] for details). In particular, it is shown in [8] that for low-Reynolds number flows, the preconditioned system has eigenvalues bounded independently of the size of the problem. However, their clustering may not be optimal. As we show below, our second level of preconditioning will improve convergence dramatically.

We present results for a standard test problem, the driven cavity flow, for three values of the Reynolds number: 10, 100, 1000. The choice $c = 10^{-2}$ in our stopping criterion (9) does not affect the number of nonlinear iterations in our tests if we take $q = \frac{3}{4}$ except for $Re = 1000$ where we set $q = 1$. We stopped the Newton (outer) iteration when $\|F(x^m)\|/\|F(x^0)\| \leqslant 10^{-10}$. The initial guess for the Newton iteration was the zero vector except for the flow at $Re = 1000$, in which case we used the result of three Picard iterations as a starting approximation to the solution.

As described in Section 2, in order to reduce storage, at each nonlinear iteration the GMRES starting vector was $x = P_m^{-1} b_m$, which ensures that at each GMRES iteration the residual has the form $r^{\mathrm{T}} = (0, r_p^{\mathrm{T}})$.

Table 5
Driven cavity flow: total/average number of GMRES iterations with and without adaptive preconditioning

Re	*n*	$A_m P_m^{-1}$	$A_m P_m^{-1} M_m^{-1}$
10	4,771	113/**21.2**	51/**8.8**
	18,755	91/**21.0**	49/**10.5**
	74,371	98/**22.7**	56/**12.2**
100	4,771	227/**30.3**	102/**12.4**
	18,755	208/**32.5**	103/**15.0**
	74,371	174/**32.6**	97/**17.2**
1000	4,771	1,064/**105.0**	408/**23.0**
	18,755	952/**118.0**	411/**27.8**
	74,371	718/**118.2**	375/**32.5**

Table 6
Driven cavity flow: Savings in precvecs ($P_m^{-1} v$) and matvecs ($A_m v$) and additional storage ($\times nz(A_m)$)

Re	*n*	$P_m^{-1} v$ (%)	$A_m v$ (%)	$\times nz(A_m)$
10	4,771	55	45	0.21
	18,755	46	37	0.19
	74,371	42	31	0.20
100	4,771	55	37	0.44
	18,755	50	32	0.41
	74,371	44	25	0.38
1000	4,771	62	36	1.12
	18,755	57	32	1.02
	74,371	48	25	0.87

The resulting Arnoldi basis vectors have also the same form, a fact which allows us to store a basis V of size n_2. Bearing in mind that for standard P2–P1 discretizations $n_1 = 8n_2$ in 2 dimensions and $n_1 = 24n_2$ in 3 dimensions, the storage reduction is significant. Moreover, the cost of applying the adaptive preconditioner becomes almost negligible, as Table 5 shows. Finally, we note that relaxing the action of P^{-1} (say, by an approximate inversion of K_m) the exact Arnoldi basis will be of full length. However, we found that storing only a basis of length n_2 does not lead to deterioration in performance, provided the stopping criterion employed for the solution of linear systems with P is of the same order as the outer GMRES tolerance (Table 6).

Table 5 displays the number of GMRES iterations for the case where only P_m is employed and for the case where the adaptive preconditioner (6) is additionally used with $\alpha = h$. As before, we evaluate the extra cost in matvecs involving the system matrix A_m, with the resulting savings evaluated in precvecs and matvecs.

The gains for this problem also are considerable. In particular, the performance improvement is better gleaned from the reduction in precvecs (between 42–48% for the largest problem), since in this case they are more expensive than the matvecs. This is generally the case in practice, as efficient preconditioning comes at a price. We also note here that the additional preconditioner preserves the independence of the size of the problem achieved by the first preconditioning technique. Finally, we notice an increase in precvec savings with increasing Reynolds number. The storage increases also, though we note that for all *Re*, the extra storage required does not exceed the amount required for the matrix itself. In general, one can use a storage-limiting strategy; we found that this always leads to deterioration in performance.

4. Summary

In this work we addressed the issue of constructing adaptively preconditioners for sequences of linear systems arising from nonlinear systems of equations. Our technique, though seemingly similar to other Krylov subspace approaches (notably [1]), is different in that it does not try to construct an invariant subspace, or extract some eigenvalue information. The preconditioners used are guaranteed to shift eigenvalues to one, while preserving positivity in the case of positive-definite matrices. Moreover, our approach is based on, and incorporates, a first level of preconditioning, which for many applications of importance is available. Thus, our adaptive preconditioners can be seen as a performance enhancing tool. We demonstrated that in certain situations this enhancement can be remarkable. In particular, domain decomposition methods for nonlinear problems lend themselves quite naturally to our adaptive technique. This leads us to conjecture that a black-box approach is possible, where algebraic domain-decomposition approaches for nonlinear systems can be combined with our adaptive method to deal with the resulting Schur complement problems. We expect future work to validate and contribute to this conjecture.

References

[1] J. Baglama, D. Calvetti, G.H. Golub, L. Reichel, Adaptively preconditioned GMRES algorithms, SIAM J. Sci. Comput. 20 (1) (1998) 243–269.
[2] K. Burrage, J. Erhel, On the performance of various adaptive preconditioned GMRES strategies, Numer. Linear Algebra Appl. 5 (1998) 101–121.
[3] K. Burrage, J. Erhel, B. Pohl, A. Williams, A deflation technique for linear systems of equations, SIAM J. Sci. Comput. 19 (4) (1998) 1245–1260.
[4] E. de Sturler, Truncation strategies for optimal Krylov subspace methods, SIAM J. Numer. Anal. 36 (3) (1999) 864–889.
[5] B. Carpentieri, I.S. Duff, L. Giraud, A class of spectral two-level preconditioners, SIAM J. Sci. Comput. 25 (2003) 749–765.
[6] R.S. Dembo, S.C. Eisenstat, T. Steihaug, Inexact Newton methods, SIAM J. Numer. Anal. 19 (2) (1982) 400–408.
[7] I.S. Duff, L. Giraud, J. Langou, E. Martin, Using spectral low rank preconditioners for large electromagnetic calculations, Technical Report TR/PA/03/95, CERFACS, Toulouse, France, 2003.
[8] H.C. Elman, D. Loghin, A.J. Wathen, Preconditioning techniques for Newton's method for the incompressible Navier–Stokes equations, BIT 43 (2003) 961–974.
[9] P. Fischer, Projection techniques for iterative solution of a $x = b$ with successive right-hand sides, Comput. Methods Appl. Mech. Engrg. 163 (1998) 193–204.
[10] P. Lancaster, M. Tismenetsky, The Theory of Matrices, Academic Press, New York, 1985.
[11] R.B. Morgan, A restarted GMRES method augmented with eigenvectors, SIAM J. Matrix Anal. Appl. 16 (1995) 1154–1171.

[12] R.B. Morgan, Implicitly restarted GMRES and Arnoldi methods for nonsymmetric systems of equations, SIAM J. Matrix Anal. Appl. 21 (2000) 1112–1135.
[13] R.B. Morgan, GMRES with deflated restarting, SIAM J. Sci. Comput. 24 (2002) 20–37.
[14] M. Parks, E. de Sturler, G. Mackey, D. Johnson, S. Mait, Recycling Krylov subspaces for sequences of linear systems, Technical Report UIUCDCS-R-2004-2421, University of Illinois, March 2004.
[15] B.N. Parlett, A new look at the Lanczos algorithm for solving symmetric systems of linear equations, Linear Algebra Appl. 29 (1980) 323–346.
[16] A. Quarteroni, A. Valli, Numerical Approximation of Partial Differential Equations, Springer, Berlin, 1994.
[17] A. Quarteroni, A. Valli, Domain Decomposition Methods for Partial Differential Equations, Numerical Mathematics and Scientific Computation, Oxford University Press, Oxford, 1999.
[18] Y. Saad, On the Lanczos method for solving symmetric linear systems with several right-hand sides, Math. Comp. 178 (1987) 651–662.
[19] L.N. Trefethen, Approximation theory and numerical linear algebra, in: J.C. Mason, M.G. Cox (Eds.), Algorithms for approximation II, Chapman & Hall, London, 1990.
[20] C. Vuik, Fast iterative solvers for the discretized incompressible Navier–Stokes equations, Technical Report 93–98, Deflt University of Technology, 1993.