SIAM J. SCI. COMPUT.
Vol. 19, No. 1, pp. 15–26, January 1998

RESTARTED GMRES FOR SHIFTED LINEAR SYSTEMS[*]

ANDREAS FROMMER[†] AND UWE GLÄSSNER[‡]

Abstract. Shifted matrices, which differ by a multiple of the identity only, generate the same Krylov subspaces with respect to any fixed vector. This fact has been exploited in Lanczos-based methods like CG, QMR, and BiCG to simultaneously solve several shifted linear systems at the expense of only one matrix–vector multiplication per iteration. Here, we develop a variant of the restarted GMRES method exhibiting the same advantage and we investigate its convergence for positive real matrices in some detail. We apply our method to speed up "multiple masses" calculations arising in lattice gauge computations in quantum chromodynamics, one of the most time-consuming supercomputer applications.

Key words. shifted systems, GMRES, Krylov methods, QCD

AMS subject classifications. 65F10, 81T25

PII. S1064827596304563

1. Introduction.

Let $A \in \mathbf{C}^{n\times n}$ be nonsingular and nonhermitian in general and let $\alpha \in \mathbf{C}$ be such that the shifted matrix $\hat{A} = A + \alpha I$ is nonsingular, too. We consider the two systems

$$Ax = b \; , \tag{1}$$

$$\hat{A}\hat{x} = b \; . \tag{2}$$

Below, system (1) will be termed the *seed* system, whereas (2) will be called the *add* (additional) system. Such shifted systems arise in a variety of practical applications like higher-order implicit methods for solving time-dependent partial differential equations [15, 23], control theory [3, 18], and in lattice gauge computations in quantum chromodynamics (QCD) [14, 16]. The latter application was our motivation for the present study.

The Krylov subspaces

$$K_m(A, b) = \mathrm{span}\{b, Ab, A^2b, \ldots, A^{m-1}b\}$$

and

$$K_m(\hat{A}, b) = \mathrm{span}\{b, \hat{A}b, \hat{A}^2b, \ldots, \hat{A}^{m-1}b\}$$

for the seed and the add systems, respectively, are identical. Any $v \in K_m(A, b)$ can be expressed as $v = p_{m-1}(A)b$ with p_{m-1} a polynomial of degree $\leq m-1$ or, equivalently, $v = \hat{p}_{m-1}(\hat{A})b$, where

$$\hat{p}_{m-1}(t) = p_{m-1}(t - \alpha) \; .$$

[*] Received by the editors May 24, 1996; accepted for publication (in revised form) October 28, 1996.

http://www.siam.org/journals/sisc/19-1/30456.html

[†] Bergische Universität GH Wuppertal, Fachbereich Mathematik, D-42097 Wuppertal, Germany (frommer@math.uni-wuppertal.de). This author's research was partially completed while the author was visiting Stanford University, Stanford, CA. This research was partially supported by Deutsche Forschungsgemeinschaft.

[‡] Bergische Universität GH Wuppertal, Fachbereich Physik, D-42097 Wuppertal, Germany (uweg@wptsc.physik.uni-wuppertal.de).

Krylov subspace methods [7, 10, 20] for solving (1) are iterative methods for which the mth iterate x^m is contained in $x^0 + K_m(A, r^0)$, with $r^0 = b - Ax^0$ being the initial residual belonging to the initial guess x^0. Each iterative step requires at least one matrix–vector multiplication with A in order to update the calculated basis of $K_m(A, r^0)$ to $K_{m+1}(A, r^0)$. Now, assume that we take initial guesses $x^0 = 0$ and $\hat{x}^0 = 0$ for the seed and the add systems, respectively. Then the initial residuals satisfy $r^0 = \hat{r}^0 = b$ showing that the iterates x^m and $\hat{x}^m$ for the seed and the add systems are from the same Krylov subspace $K_m(A, b) = K_m(\hat{A}, b)$. Therefore, if we apply a Krylov subspace method to solve (1) and (2) simultaneously, the basis has to be calculated only once. Consequently, depending on whether the recurrencies of the Krylov subspace method in use allow it, the iterates $\hat{x}^m$ (and residuals $\hat{r}^m$) of the add system may be calculated at very low extra cost (typically some inner products and SAXPYs) without any matrix–vector multiplications.

The paper [9] showed that this is actually possible if one uses the QMR [12] or the TFQMR method [8]. For hermitian systems and the CG method similar observations were made in [25, 26], and the BiCG method was treated in [14]. All these methods rely on the Lanczos process to compute a basis for $K_m(A, r^0)$ which is identical to that produced by the Lanczos process applied to $K_m(\hat{A}, r^0)$. The latter fact has also repeatedly been reported by physicists working in QCD [1, 2].

The Lanczos process has the advantage of relying upon short recurrencies. On the other hand, in the nonhermitian case the computation of each basis vector requires one matrix multiplication with A *and* one with A^H. Thus, except for TFQMR, the above Lanczos-based methods for nonhermitian systems require an additional matrix–vector multiplication with A^H. Moreover, the Lanczos process can break down prematurely, but this can in most cases be circumvented by using a somewhat more involved look-ahead strategy [11].

GMRES [21] is a Krylov subspace method in which the basis for $K_m(A, r^0)$ is calculated via the Arnoldi process. The mth iterate x^m is characterized by having the smallest residual possible, i.e.,

$$||b - Ax^m||_2 = \min_{x \in x_0 + K_m(A, r^0)} ||b - Ax||_2 \; .$$

Computing the next basis vector requires one multiplication with A only and premature breakdown cannot occur. However, no short recurrencies are present, so all basis vectors have to be stored and the number of inner products and SAXPYs increases linearly with m. For larger values of m the GMRES method thus becomes less efficient and one has to consider truncated modifications. A common practice is GMRES(k), where the GMRES process is restarted after every k iterations using the current iterate as a new initial guess.

Just like the Lanczos process, the Arnoldi process also computes identical basis vectors for $K_m(A, r^0)$ and $K_m(\hat{A}, r^0)$ so that the add systems can again be solved cheaply if GMRES is performed for the seed and the add systems simultaneously (see [3] and also [22] for a hybrid variant). However, the situation deteriorates as soon as we consider restarts: the GMRES residuals r^k and $\hat{r}^k$ of the seed and the add systems will generally *not* be colinear anymore, so that $K_m(A, r^k) \neq K_m(\hat{A}, \hat{r}^k)$. Thus, from the first restart on, the matrix multiplications for the add system cannot be saved anymore.

As a remedy to this situation we now propose to modify the GMRES iteration for the *add* system by forcing the residual $\hat{r}^m$ to be colinear to r^m, the residual of the seed system on which the usual GMRES iteration is performed. Then, restarts do not prevent us from saving the matrix–vector multiplications for the add system. In section 2 we will show under which conditions a residual $\hat{r}^m$ colinear to r^m will exist, and we will work out a more detailed description of the proposed method. In section 3 we will prove the convergence of our method for a positive real matrix A and $\alpha > 0$. This is precisely the case of interest for the lattice gauge theory computations for which we report the results of some numerical experiments in section 4.

2. Shifted GMRES. Any vector x^m from the affine Krylov subspace $x_0 + K_m(A, r^0)$ can be represented as

$$x^m = x^0 + q_{m-1}(A)r^0$$

with q_{m-1} a polynomial of degree $\leq m-1$. The corresponding residual $r^m = b - Ax^m$ satisfies

$$r^m = r^0 - Aq_{m-1}(A)r^0 = p_m(A)r^0 \ ,$$

where $p_m(t) = 1 - tq_{m-1}(t)$ is a polynomial of degree $\leq m$ with $p_m(0) = 1$. Adopting a similar notation for the add system we have

$$\hat{x}^m = \hat{x}^0 + \hat{q}_{m-1}(\hat{A})\hat{r}^0, \ \ \hat{r}^m = \hat{p}_m(\hat{A})\hat{r}^0 \ .$$

Now assume that the initial residuals $r^0, \hat{r}^0$ are colinear,

$$\hat{r}^0 = \beta_0 r^0, \ \ \beta_0 \in \mathbf{C} \ .$$

Then, requiring

$$\hat{r}^m = \beta_m r^m, \ \ \beta_m \in \mathbf{C}, \tag{3}$$

amounts to $\hat{p}_m(\hat{A})\hat{r}^0 = \beta_m p_m(A)r^0$ or, equivalently,

$$\beta_0 \hat{p}_m(\hat{A})r^0 = \beta_m p_m(\hat{A} - \alpha I)r^0 \ . \tag{4}$$

Together with the additional condition

$$\hat{p}_m(0) = 1 \tag{5}$$

this yields defining equations for $\hat{p}_m$ and β_m.

LEMMA 2.1. *Assume that $K_{m+1}(A, r^0)$ has dimension $m+1$ and let $\beta_0 \neq 0$. Then there exists a polynomial $\hat{p}_m$ and a complex number β_m satisfying* (4) *and* (5) *iff $p_m(-\alpha) \neq 0$. In that case,*

$$\beta_m = \beta_0 / p_m(-\alpha) \quad \textit{and} \quad \hat{p}_m(t) = p_m(t - \alpha)/p_m(-\alpha) \ .$$

Proof. By assumption, the vectors $r^0, Ar^0, \ldots, A^m r^0$ are linearly independent. Therefore, (4) is equivalent to

$$\beta_0 \hat{p}_m(t) = \beta_m p_m(t - \alpha) \ .$$

This shows that (5) can be satisfied iff $\beta_0 = \beta_m p_m(-\alpha)$ which, in turn, is fulfilled iff $p_m(-\alpha) \neq 0$ and $\beta_m = \beta_0 / p_m(-\alpha)$. □

For a given Krylov subspace method the corresponding polynomials p_m are usually not calculated in practice, but they are useful for theoretical investigations. Lemma

2.1 shows that the colinearity condition (3) cannot always be satisfied, but if it can, the residual $\hat{r}^m$ (and the corresponding vector $\hat{x}^m$) is unique.

We now work out how the iterates for the add system satisfying (3) can be practically computed when the GMRES iteration is performed on the seed system. We start with a description of the Arnoldi process which computes an orthogonal basis of $K_m(A, r)$.

Algorithm 2.1. *Arnoldi* (r, m).

```
{ input:   initial vector $r$, subspace dimension $m$
  output:  coefficients $h_{ij} \in \mathbf{C}$, $j = 1, \dots, m$, $i = 1, \dots, j+1$
           orthogonal vectors $v^1, \dots, v^{m+1} \in \mathbf{C}^n$            }
 set $v^1 = r/||r||_2$
 for $j = 1, 2, \dots, m$
     $w^j = Av^j$
     for $i = 1, \dots, j$
         $h_{ij} = \langle w^j, v^i \rangle$
         $w^j = w^j - h_{ij}v^i$
     $h_{j+1,j} = ||w^j||_2$
     $v^{j+1} = w^j / h_{j+1,j}$
```

We define the "upper Hessenberg" matrix H_m^* by

$$H_m^* = \begin{bmatrix} h_{1,1} & h_{1,2} & \cdots & h_{1,m} \\ h_{2,1} & h_{2,2} & \cdots & h_{2,m} \\ & h_{3,2} & & \vdots \\ & & \ddots & h_{m,m} \\ & & & h_{m+1,m} \end{bmatrix} \in \mathbf{C}^{(m+1)\times m},$$

and V_m by

$$V_m = [v^1 | v^2 | \dots | v^m] \in \mathbf{C}^{n \times m} .$$

We sum up the most important properties of the Arnoldi process in the following lemma (see [17, 20], for example).

Lemma 2.2.

(i) *The vectors* v^j, $j = 1, \dots, m$, *are orthonormal.*

(ii) $K_j(A, r) = \operatorname{span}\{v^1, \dots, v^j\}$.

(iii) *For* $j = 1, \dots, m$ *we have*

$$AV_j = V_{j+1} H_j^* .$$

Note that Algorithm 2.1 is just one (the "modified Gram–Schmidt" version) out of several possible algorithmic formulations of the Arnoldi process; see [20]. Also note that we implicitly assumed that $h_{j+1,j} \neq 0$ for $j = 1, \dots, m$. The event $h_{j+1,j} = 0$ for some j represents a lucky breakdown of the Arnoldi process, since then $K_l(A, r) = K_j(A, r)$ for all $l \geq j$ and the GMRES iterate x^j is the exact solution of (1); see [20]. For simplicity, we do not consider lucky breakdowns here.

We now give an algorithmic description for the GMRES method where, for convenience, we assume a fixed maximal subspace length m. The vector e_1 denotes $e_1 = (1, 0, \dots, 0)^T \in \mathbf{C}^{m+1}$.

Algorithm 2.2. *GMRES* (x^0, m).

```
{ input:  initial guess $x^0$, maximal subspace length $m$
  output: GMRES iterate $x^m$                              }
$r^0 = b - Ax^0$, $\beta = ||r^0||_2$
call Arnoldi $(r^0, m)$, yielding $H_m^*$ and $V_m$
compute $y_m \in \mathbf{C}^m$, the minimizer of $||\beta e_1 - H_m^* y||_2$
$x^m = x^0 + V_m y_m$
```

Lemma 2.3. *The residual r^m of the GMRES iterate x^m satisfies*

$$||r^m||_2 = \min_{x \in x_0 + K_m(A, r^0)} ||b - Ax||_2 \ ,$$

and

$$r^m = V_{m+1} z_{m+1}, \ \ \textit{where} \ \ z_{m+1} = \beta e_1 - H_m^* y_m \in \mathbf{C}^{m+1} \ .$$

Proof. See [20]. □

The norm of the residual r^j of "intermediate" GMRES iterates x^j can be monitored very cheaply if one updates the QR factorization of the matrices H_j^* in the Arnoldi process. Therefore, Algorithm 2.2 can be stopped as soon as $||r^j||_2$ is small enough, replacing m by j. For details, see [20].

Now, consider the add system. Our basic assumption is that

$$\hat{r}^0 = \beta_0 r^0, \ \ \beta_0 \in \mathbf{C} \ .$$

Due to the Arnoldi process we have by Lemma 2.2

$$AV_m = V_{m+1} H_m^*,$$

which results in

$$\hat{A} V_m = V_{m+1} \hat{H}_m^* \tag{6}$$

with

$$\hat{H}_m^* = H_m^* + \alpha \begin{bmatrix} 1 & & \\ & \ddots & \\ & & 1 \\ 0 & \cdots & 0 \end{bmatrix} \in \mathbf{C}^{(m+1) \times m} \ .$$

Requiring the colinearity condition (3) for $\hat{x}^m = \hat{x}^0 + V_m \hat{y}_m$ we obtain, using Lemma 2.3 and (6),

$$\begin{aligned} & & \hat{r}^m &= \beta_m r^m \\ &\Leftrightarrow & b - \hat{A}(\hat{x}^0 + V_m \hat{y}_m) &= \beta_m V_{m+1} z_{m+1} \\ &\Leftrightarrow & \hat{r}^0 - \hat{A} V_m \hat{y}_m &= V_{m+1} z_{m+1} \beta_m \\ &\Leftrightarrow & \beta_0 r^0 - V_{m+1} \hat{H}_m^* \hat{y}_m &= V_{m+1} z_{m+1} \beta_m \\ &\Leftrightarrow & V_{m+1}(\hat{H}_m^* \hat{y}_m + z_{m+1} \beta_m) &= \beta_0 r^0 \\ &\Leftrightarrow & \hat{H}_m^* \hat{y}_m + z_{m+1} \beta_m &= \beta_0 ||r^0||_2 e_1. \end{aligned}$$

In terms of the unknown quantities $\hat{y}^m$ and β_m, the last line can be written as an equation in $\mathbf{C}^{m+1}$,

$$\left[\begin{array}{c|c} \hat{H}_m^* & z_{m+1} \end{array}\right] \begin{bmatrix} \hat{y}_m \\ \beta_m \end{bmatrix} = \beta_0 ||r^0||_2 e_1 \ . \tag{7}$$

Lemma 2.4. *Let p_m be the GMRES polynomial of the seed system, i.e., $r^m = p_m(A)r^0$. Then* (7) *has a unique solution $(\hat{y}_m, \beta_m)$ iff $p_m(-\alpha) \neq 0$.*

Proof. If $p_m(-\alpha) \neq 0$ we know that $\beta_m = \beta_0/p_m(-\alpha)$ by Lemma 2.1. Furthermore, $\hat{H}_m^*$ has rank m since we assumed $h_{j+1,j} \neq 0$ for $j = 1, \ldots, m$. Thus, $\hat{y}_m$ is uniquely determined by the first m rows of

$$\hat{H}_m^* \hat{y}_m = \beta_0 ||r^0||_2 e_1 - \beta_m z_{m+1} \ .$$

If, on the other hand, (7) has a unique solution, β_m exists, and by Lemma 2.1 we know that it is equal to $\beta_0/p_m(-\alpha)$; hence $p_m(-\alpha) \neq 0$. □

In computational practice one can use the QR factorization of the upper Hessenberg matrix $[\hat{H}_m^* | z_{m+1}] = Q_{m+1} R_{m+1}$ to solve (7). Since $\hat{H}_m^*$ has maximum rank, the system (7) is not solvable iff the last row of R_{m+1} is (numerically) zero. In that case we can step m by one, update the QR factorization at very low cost, and check for the existence of a solution of (7) again. This provides a computationally viable way to adapt the dimension m such that $\hat{y}_m$ and β_m exist and are computable in a stable manner. For simplicity, we do not include such a dynamical strategy in the formulation of the algorithms to come.

For fixed m, an algorithmic description of the shifted GMRES algorithm can now be given as follows.

Algorithm 2.3. *Shifted GMRES* $(x^0, \hat{x}^0, \beta_0, m)$.

```
{ input:  initial guesses $x^0, \hat{x}^0$ for the seed and the add system such that $\hat{r}^0 = \beta_0 r^0$,
          maximal subspace length $m$,
  output: GMRES iterate $x^m$ for the seed system, iterate $\hat{x}^m$ for the add system,
          $\beta_m \in \mathbf{C}$ such that $\hat{r}^m = \beta_m r^m$ }
$r^0 = b - Ax^0$, $\beta = ||r^0||_2$
call Arnoldi $(r^0, m)$, yielding $H_m^*$ and $V_m$
{ GMRES iterate for seed system }
compute $y_m \in \mathbf{C}^m$, the minimizer of $||\beta e_1 - H_m^* y||_2$
$x^m = x^0 + V_m y_m$, $z_{m+1} = \beta e_1 - H_m^* y_m$
{ iterate for add system }
solve
   $\left[\begin{array}{c|c} \hat{H}_m^* & z_{m+1} \end{array}\right] \begin{bmatrix} \hat{y}_m \\ \beta_m \end{bmatrix} = \beta_0 \beta e_1$
$\hat{x}^m = \hat{x}^0 + V_m \hat{y}_m$
```

By our previous discussion it is clear that we have

$$b - \hat{A}\hat{x}^m = \hat{r}^m = \beta_m r^m = \beta_m V_{m+1} z_{m+1} \ .$$

For the sake of simplicity we did not include a check on the existence of $\hat{x}^m$ into Algorithm 2.3. The restarted variant of Algorithm 2.3 is now straightforward.

Algorithm 2.4. *Restarted Shifted GMRES* $(k, x^0, \hat{x}^0, \beta_0)$.

```
{ input:  $x^0, \hat{x}^0, \beta_0$ as in Algorithm 2.3, restart value $k$
  output: approximate solutions $x^{jk}, \hat{x}^{jk}$ for the seed and the add system }
for $j = 0, 1, \ldots, j_{max}$
    call Shifted GMRES($x^{jk}, \hat{x}^{jk}, \beta_{jk}, k$), yielding $x^{(j+1)k}, \hat{x}^{(j+1)k}$, and $\beta_{(j+1)k}$
        such that $\hat{r}^{(j+1)k} = \beta_{(j+1)k} r^{(j+1)k}$
    stop if $||r^{(j+1)k}||_2$ and $||\hat{r}^{(j+1)k}||_2 = \beta_{(j+1)k}||r^{(j+1)k}||_2$ are sufficiently small
```

Here, j_{max} is an a priori upper bound for the maximal number of shifted GMRES(k)-sweeps to be performed.

For $k = 1$, the restarted GMRES method reduces to the minimal residual method of [5]. Because of its simple form, we finish this section by writing down the resulting shifted algorithm, giving all scalar quantities to use explicitly.

Algorithm 2.5. *Shifted minimal residual* $(x^0, \hat{x}^0, \beta_0)$.

```
{ input, output as in Algorithm 2.4 with $k = 1$ }
$r^0 = b - Ax^0$
for $j = 0, 1, \ldots, j_{max}$
    $p^j = Ar^j$
    $\gamma_j = \langle p^j, r^j \rangle / \langle p^j, p^j \rangle$
    $x^{j+1} = x^j + \gamma_j r^j$
    $r^{j+1} = r^j - \gamma_j p^j$
    $\beta_{j+1} = \beta_j / (1 + \gamma_j \alpha)$
    $\hat{\gamma}_j = \gamma_j \beta_{j+1}$
    $\hat{x}^{j+1} = \hat{x}^j + \hat{\gamma}_j r^j$
    stop if $||r^{j+1}||_2$ and $||\hat{r}^{j+1}||_2 = \beta_{j+1}||r^{j+1}||_2$ are sufficiently small
```

This method was devised in [16] in a different context. Due to its very low storage requirements it has become a reference method for the QCD computations to be considered in section 4.

3. Convergence.

In exact arithmetic there exists an index $m \le n$ such that the GMRES iterate x^m is the solution of the seed system (1). Then, $\hat{x}^m$ from the shifted GMRES method is the solution of the add system since $\hat{r}^m = \beta_m r^m = 0$. However, such a situation is unrealistic because of the effects of roundoff and because m would usually be impracticably large anyway.

In this section we will show that if A is positive real (i.e., $\mathrm{Re}(\langle Ax, x \rangle) > 0$ for all $x \neq 0$) and $\alpha > 0$, then the colinearity condition (3) can always be satisfied, so that $\hat{x}^m$ in the (restarted) shifted GMRES method always exists. Moreover, we will also show that the convergence of the add system is faster in the sense that its residuals are smaller than the corresponding ones of the seed system. Let us mention at this point that this is precisely the situation that we encounter in the lattice gauge theory application to be discussed in section 4.

Recall that the field of values $F(A)$ of $A \in \mathbf{C}^{n \times n}$ is defined by

$$F(A) = \{\langle Ax, x \rangle \mid x \in \mathbf{C}^n, \, ||x||_2 = 1\} \subseteq \mathbf{C} \, .$$

Clearly, if A is positive real, the field of values is contained in the right half-plane. We need the following result on the zeros of a GMRES polynomial.

LEMMA 3.1. *Let $r^m = p_m(A)r^0$ be the residual of the mth GMRES iterate for* (1) *with initial guess $x^0, r^0 = b - Ax^0, m \geq 1$. Then all zeros ζ of p_m satisfy*

$$\frac{1}{\zeta} \in F(A^{-H}) \ ,$$

where $A^{-H} = (A^H)^{-1}$.

Proof. Let ζ be any zero of p_m. Since $p_m(0) = 1$ we have $\zeta \neq 0$ and we can write

$$p_m(t) = \left(1 - \frac{t}{\zeta}\right) q_{m-1}(t) \text{ with } q_{m-1}(0) = 1 \ .$$

Denote $w = q_{m-1}(A)r^0$ and $r = Aw$. Then

$$||p_m(A)r^0||_2 = ||w - \frac{1}{\zeta}r||_2 \ .$$

For $\gamma \in \mathbf{C}$ the functional $||w - \gamma r||_2$ is minimized for

$$\gamma^* = \frac{\langle r, w\rangle}{\langle r, r\rangle} \ ,$$

and since for GMRES the norm $||p_m(A)r^0||_2$ is minimal we conclude that

$$\frac{1}{\zeta} = \gamma^* = \frac{\langle r, w\rangle}{\langle r, r\rangle} = \frac{\langle r, A^{-1}r\rangle}{\langle r, r\rangle}$$

and $\langle r, A^{-1}r\rangle / \langle r, r\rangle = \langle A^{-H}(r/||r||_2), r/||r||_2\rangle \in F(A^{-H})$. □

COROLLARY 3.2. *If A is positive real, then* $\mathrm{Re}(\zeta) > 0$.

Proof. Since $\langle Ax, x\rangle = \langle A^{-H}y, y\rangle$ for $y = Ax$ and A is nonsingular, $F(A^{-H})$ is contained in the right half-plane just like $F(A)$. By Lemma 3.1, therefore, $\mathrm{Re}(\frac{1}{\zeta}) > 0$ which is equivalent to $\mathrm{Re}(\zeta) > 0$. □

THEOREM 3.3. *Let A be positive real and $\alpha > 0$. Then the restarted shifted GMRES method* (*Algorithm* 2.4) *converges for the seed and the add systems for every restart value k. In particular, the iterates for the add system always exist. Moreover, we have*

$$||\hat{r}^{jk}||_2 \leq |\beta_0| \cdot ||r^{jk}||_2 \tag{8}$$

for all j.

Proof. The convergence for the seed system is a well-known result on restarted GMRES for positive real matrices, see [20, 21]. To prove (8) we have to show that $|\beta_{jk}| < |\beta_0|$ for all j. From Lemma 2.1 we know

$$\beta_{(j+1)k} = \beta_{jk}/p_{k,j}(-\alpha), \ \ j = 0, 1, \ldots,$$

where $p_{k,j}$ is the GMRES polynomial for the jth restart; i.e.,

$$r^{(j+1)k} = p_{k,j}(A)r^{jk}, \ \ j = 0, 1, \ldots.$$

So all we have to show is $|p_{k,j}(-\alpha)| > 1$ for all j, since then, by induction,

$$|\beta_{(j+1)k}| \leq |\beta_{jk}| \leq \cdots \leq |\beta_0| \ .$$

By Corollary 3.2 we have

$$p_{k,j}(t) = \prod_{i=1}^{k}(1 - t/\zeta_{j,i}) \text{ with } \mathrm{Re}(\zeta_{j,i}) > 0, \, i = 1, \ldots, k \ .$$

Consequently, for $\alpha > 0$,

$$|1+\alpha/\zeta_{j,i}| = |\zeta_{j,i}+\alpha| \; / \; |\zeta_{j,i}| \; > \; 1 \text{ for } i = 1, \ldots, k,$$

so that

$$|p_{k,j}(-\alpha)| = \prod_{i=1}^{k} |1+\alpha/\zeta_{j,i}| > 1. \qquad \square$$

Note that usually we have $\beta_0 = 1$ through the choice $x^0 = \hat{x}^0 = 0$, so that (8) may be interpreted as the add system converging more rapidly than the seed system.

4. Application in QCD. Lattice gauge theory is a discretization of QCD which is universally believed to be the fundamental physical theory of the strong interaction between matter. In the Wilson fermion [24] framework, lattice gauge computations require the solution of systems of equations of the form

$$(mI - D)x = b \; , \tag{9}$$

representing a periodic nearest neighbor coupling on a four-dimensional space-time lattice. At each lattice point we have 12 degrees of freedom, so that the total size of system (9) for an $n_1 \times n_2 \times n_3 \times n_4$ lattice is $n = 12\, n_1 n_2 n_3 n_4$. Typical current values for n_1 to n_4 are in the range of 16 to 32, so that n is at least 10^6. Solving the systems from (9) is a computationally intensive task, and a significant part of today's total supercomputing capacity is actually spent in these computations. The matrix D is complex and nonhermitian, the coupling coefficients contained in D being the result of a stochastic Monte Carlo simulation. For $m > m_c$, the critical value of m, the matrix $mI - D$ is positive real. From the QCD point of view, it is particularly interesting to solve (9) for several (typically around five) values of m larger than but close to m_c and the same right-hand side b. This situation is precisely the one for which we were able to prove the feasibility and efficiency of the shifted GMRES method in section 3.

If we order the lattice points in the usual checkerboard ("odd-even") manner, the matrix D splits into the form

$$D = \left[\begin{array}{c|c} 0 & D_{eo} \\ \hline D_{oe} & 0 \end{array}\right] \; .$$

Partitioning correspondingly,

$$x = \begin{pmatrix} x_e \\ x_o \end{pmatrix}, \;\; b = \begin{pmatrix} b_e \\ b_o \end{pmatrix} \; ,$$

we can form the Schur complement in (9) to get

$$(m^2 I - D_{oe} D_{eo}) x_o = (m b_o + D_{oe} b_e) \tag{10}$$

and

$$x_e = \frac{1}{m}(b_e + D_{eo} x_o) \; .$$

The transition from (9) to the *odd-even-reduced* system (10) may be interpreted as a preconditioning process which, as computational experience shows, usually reduces the number of iterations in a Krylov subspace method by a factor of 2 to 3 [6, 13]. In the QCD community, the above odd-even preconditioning is considered to be the only successful preconditioner for (9) so far, since it still parallelizes very efficiently and does not increase the overall cost for a matrix multiplication. The odd-even reduced system (10) again exhibits a shifted structure, now with respect to m^2. The new right-hand side $mb_o + D_{oe}b_e$, however, will depend on m, unless $b_o = 0$. The latter case occurs for so-called *point sources* where b is just some unit vector. Point sources *are* of physical interest. For general right-hand sides, our shifted GMRES method can still be applied, at the expense of doubling the total cost, by considering the two systems

$$\begin{aligned} (m^2 I - D_{oe}D_{eo})x_o &= b_o \ , \\ (m^2 I - D_{oe}D_{eo})x_o &= D_{oe}b_e \ . \end{aligned} \tag{11}$$

A linear combination of their solutions yields the solution of (10).

In the following tables we compare the performance of our shifted GMRES(k) method for (10) (five values of m simultaneously) against a computation where the systems were solved one after the other using GMRES(k) and taking the result for the previous value of m as an initial guess for the next system to solve. This approach is termed *serial* GMRES in Figure 1. For the first value of m, the initial guess was $x^0 = 0$, just as for all initial guesses in the shifted GMRES(k) method.

The linear system we used was for a comparatively small lattice of size 8^4 (so that our numerical experiments could be done on a Sun SparcStation 20). It represents a physically meaningful configuration obtained via the standard hybrid Monte Carlo (HMC) method [4]. The right-hand side b was a point source, our stopping criterion was to test whether the relative residual $\|r^k\|_2/\|b\|_2$ was less than 10^{-8}.

Instead of giving the parameter m directly, it is of common use in QCD to write down its inverse $\kappa = 1/m$. For our configuration, the critical value of κ, i.e., $\kappa_c = 1/m_c$, is known to be somewhat larger than 0.155. In our experiments we use values of m corresponding to the five κ-values

$$\kappa_1 = 0.151,\ \kappa_2 = 0.152,\ \kappa_3 = 0.153,\ \kappa_4 = 0.154,\ \kappa_5 = 0.155\ . \tag{12}$$

The first row of Figure 1 gives results for the minimal residual method (GMRES(1)). It represents the relative norm $||r^k||_2/||b||_2$ of the residual as a function of the number of matrix–vector multiplications (left diagram) and as a function of the actual run time (right diagram). The full line with its five “peaks” corresponds to the serial GMRES method applied to the five values of κ in the order given in (12), and we see that taking the result for one m as an initial guess for the next one reduces the initial residual to $\approx 10^{-2}$. The dashed line gives the relative norm of the largest residual (corresponding to κ_5) in the shifted GMRES(1) method. We see that the shifted GMRES method reduces the total number of matrix–vector multiplications by roughly a factor of 2. The right diagram shows that these savings translate into similar savings in run time.

The second and third rows in Figure 1 give analogous diagrams for GMRES(4) and GMRES(16). We note that for GMRES(16) the shifted method now performs approximately three times as fast as the serial method. These results also show that for larger values of κ, GMRES(16) performs more than two times better than the minimal residual algorithm. Therefore, in the absence of memory restrictions, GMRES(16) can be regarded as the most efficient method, also outperforming the BiCGStab algorithm considered in [13]. This observation is complemented by results in [19] which show that larger restart values in GMRES do not yield further significant improvements.

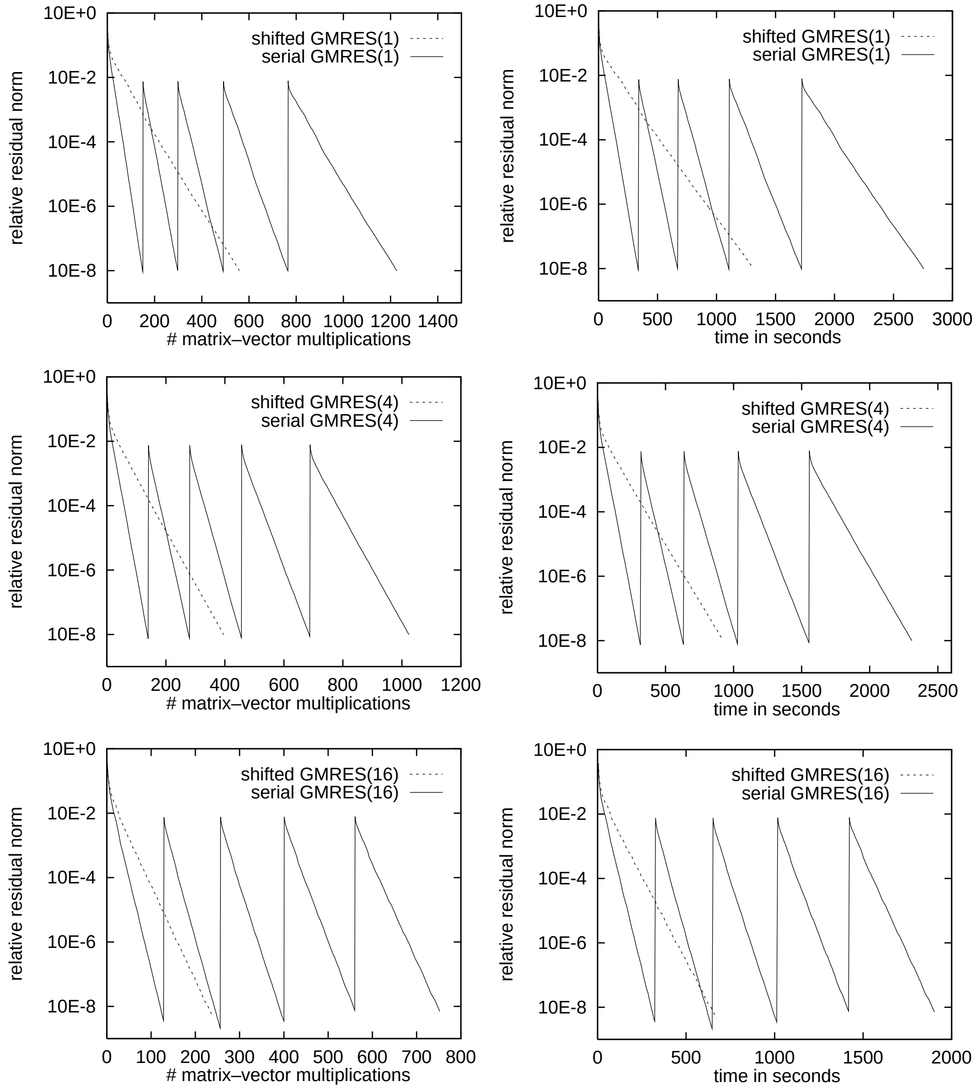

FIG. 1. *Convergence history for serial and shifted GMRES(k) for $k = 1, 4, 16$. The relative residual norm $\|r^k\|_2/\|b\|_2$ is plotted against the number of matrix–vector multiplications and run time.*

Acknowledgments. The configuration used in our computations was provided by Thomas Lippert from the Department of Physics at Wuppertal University. Peter Fiebach and Benedikt Großer from the Department of Mathematics did all the programming. The work of the first author was completed partly during his visit to Stanford in April 1996. This author wishes to express his gratitude to Gene Golub for his warm hospitality.

REFERENCES

[1] I. Barbour, N. Behlil, P. Gibbs, M. Rafiq, K. Moriarty, and G. Schierholz, *Updating fermions with the Lanczos method*, J. Comput. Phys., 68 (1987), pp. 227–236.

[2] A. Burkitt and A. Irving, *Inversion of the fermion matrix and the equivalence of the conjugate gradient and Lanczos algorithms*, Computer Phys. Comm., 59 (1990), pp. 447–454.

[3] B. Datta and Y. Saad, *Arnoldi methods for large Sylvester-like observer matrix equations, and an associated algorithm for partial spectrum assignment*, Linear Algebra Appl., 154-156 (1991), pp. 225–244.

[4] S. Duane, A. Kennedy, B. Pendleton, and D. Roweth, *Hybrid Monte Carlo*, Phys. Lett. B, 195 (1987) pp. 216–222.

[5] S. Eisenstat, H. Elman, and M. Schultz, *Variational iterative methods for nonsymmetric systems of linear equations*, SIAM J. Numer. Anal., 20 (1983), pp. 345–357.

[6] S. Fischer, A. Frommer, U. Glässner, Th. Lippert, G. Ritzenhöfer, and K. Schilling, *A parallel SSOR preconditioner for lattice QCD*, Comput. Phys. Comm., 98 (1996), pp. 20–34.

[7] R. Fletcher, *Conjugate gradient methods for indefinite systems*, in Proc. of the Dundee Biennal Conference on Numerical Analysis 1974, G.A. Watson, ed., Springer-Verlag, New York, 1975, pp. 73–89.

[8] R. Freund, *A transpose-free quasi-minimal residual algorithm for non-hermitian linear systems*, SIAM J. Sci. Statist. Comput., 14 (1993), pp. 470–482.

[9] R. Freund, *Solution of shifted linear systems by quasi-minimal residual iterations*, in Numerical Linear Algebra, L. Reichel, A. Ruttan, and R.S. Varga, eds., de Gruyter, Berlin, 1993, pp. 101–121.

[10] R. Freund, G. Golub, and N. Nachtigal, *Iterative solution of linear systems*, Acta Numerica, 1992, pp. 57–100.

[11] R. Freund, M. Gutknecht, and N. Nachtigal, *An implementation of the look-ahead Lanczos algorithm for non-hermitian matrices*, SIAM J. Sci. Statist. Comput., 14 (1993), pp. 137–158.

[12] R. Freund and N. Nachtigal, *QMR: A quasi-minimal residual method for non-Hermitian linear systems*, Numer. Math., 60 (1991), pp. 315–339.

[13] A. Frommer, V. Hannemann, B. Nöckel, Th. Lippert, and K. Schilling, *Accelerating Wilson fermion matrix inversions by means of the stabilized biconjugate gradient algorithm*, Internat. J. Modern Phys. C, 5 (1994), pp. 1073–1088.

[14] A. Frommer, B. Nöckel, S. Güsken, Th. Lippert, and K. Schilling, *Many masses on one stroke: Economic computation of quark propagators*, Internat. J. Modern Phys. C, 6 (1995), pp. 627–638.

[15] E. Gallopoulos and Y. Saad, *Efficient parallel solution of parabolic equations: implicit methods on the Cedar multicluster*, in Proc. Fourth SIAM Conf. Parallel Processing for Scientific Computing, J. Dongarra, P. Messina, D. C. Sorensen, and R. G. Voigt, eds., SIAM, Philadelphia, PA, 1990, pp. 251–256.

[16] U. Glässner, S. Güsken, Th. Lippert, G. Ritzenhöfer, K. Schilling, and A. Frommer, *How to compute Green's functions for entire mass trajectories within Krylov solvers*, Internat. J. Modern Phys. C, 7 (1996), pp. 635–644.

[17] G. Golub and C. Van Loan, *Matrix Computations*, The Johns Hopkins University Press, Baltimore, MD, 1990.

[18] A. Laub, *Numerical linear algebra aspects of control design computations*, IEEE Trans. Automat. Control, AC-30 (1985), pp. 97–108.

[19] G. Ritzenhöfer, private communication, 1995.

[20] Y. Saad, *Iterative Methods for Sparse Linear Systems*, PWS Publishing, Boston, MA, 1996.

[21] Y. Saad and M. Schultz, *GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems*, SIAM J. Sci. Statist. Comput., 7 (1986), pp. 856–869.

[22] V. Simoncini and E. Gallopoulos, *A hybrid GMRES method for nonsymmetric systems with multiple right-hand sides*, J. Comput. Appl. Math., 66 (1996), pp. 457–469.

[23] R. Sweet, *A parallel and vector variant of the cyclic reduction algorithm*, SIAM J. Sci. Statist. Comput., 9 (1988), pp. 89–105.

[24] K. Wilson, *Quarks and strings on a lattice*, in New Phenomena in Subnuclear Physics, A. Zichichi, ed., Plenum, New York, 1975, pp. 69–142.

[25] D. Young, *The search for 'high-level' parallelism for iterative sparse linear system solvers*, in Parallel Supercomputing: Methods, Algorithms and Applications, G. Carey, ed., John Wiley, Chichester, 1989, pp. 89–105.

[26] D. Young and B. Vona, *On the use of rational iterative methods for solving large sparse linear systems*, Appl. Numer. Math., 10 (1992), pp. 261–278.