Restarted Variants of DQGMRES

Kesheng Wu
Computer Science Department, University of Minnesota

I. Introduction

There are two basic types of Krylov iterations, the short recurrence methods, and the long recurrence methods [1], [10], [16]. The Conjugate Gradient (CG) method is a classic example of a short recurrence method [5]. The generalized minimum residual method (GMRES) is an example of a long recurrence method [8]. For practical uses, the long recurrences are usually either restarted or truncated [1], [12], [16]. The short recurrence methods are usually not restarted when solving a linear system. A truncated version of a long recurrence is similar to a short recurrence which is also usually not restarted. This talk exploits some characteristics of a truncated version of GMRES and shows that restarting could be potentially beneficial.

Truncated long recurrences, such as the quasi-GMRES, are shown to be effective for some linear systems [6], [9]. DQGMRES is a particular implementation of the quasi-GMRES algorithm which allows it to use a fixed amount of workspace [9]. GMRES is designed to minimize residual norm of the approximate solution on the Krylov subspace created. This optimality of GMRES method is lost when the recurrence is truncated to form the quasi-GMRES algorithm. In this talk, a number of practical techniques for monitoring the deviation of DQGMRES from GMRES are discussed. For lack of strategy to reconcile the difference, we have chosen to simply restart DQGMRES. Numerical tests show that for nonsymmetric linear systems the restarted versions of DQGMRES can perform considerably better than the original method.

II. Restarted Versions of DQGMRES

GMRES builds a Krylov subspace basis and performs the projection using the Arnoldi procedure [8]. Let $V_j = (v_1, \ldots, v_j)$ denote the Arnoldi vectors and $\bar{H}_j$ denote the upper Hessenberg matrix generated, then the Arnoldi procedure can be summarized in the following equation,

$$AV_j = V_{j+1}\bar{H}_j. \tag{1}$$

In the GMRES algorithm, the Arnoldi process is started with $v_1 = r_0/\gamma_1$, $\gamma_1 = \|r_0\|$. Let the solution at step j be $x_j = x_0 + V_j y$, then the corresponding residual is

$$r_j = b - Ax_j = r_0 - AV_j y = V_{j+1}(\gamma_1 e_1 - \bar{H}_j y). \tag{2}$$

Since V_{j+1} is an orthonormal set of vectors by construction, $\|r_j\| = \|\gamma_1 e_1 - \bar{H}_j y\|$. GMRES finds its approximate solution by minimizing $\|\gamma_1 e_1 - \bar{H}_j y\|$. This least-squares problem is solved using QR factorization via Givens rotations. The residual norm $\|r_j\|$ is known before y is actually computed. The details of how the rotations are derived and applied can be found in [8], [9].

This work was supported in part by ARPA, DOE and NSF.

The most commonly used version of GMRES is the restarted version, GMRES(m). It performs well in general [4], [7], [13], [14], [15], [17]. However on some problems, it may stagnate, i.e., the residual norm does not decrease [2], [13], [17]. Because of this, there has been considerable interest in developing different variants of GMRES. One of these variants is DQGMRES(k) [9].

Following the same notation as in [9], the restarted version DQGMRES(k) method can be written as follows.

Algorithm *1: Restarted DQGMRES(k)*

1. Start/Restart: *Choose an initial guess x_0, or use the previous solution as initial guess, compute $r_0 = b - Ax_0$, $\gamma_1 = \|r_0\|$, and $v_1 = r_0/\gamma_1$.*

2. Loop: *For $j = 1, 2, \ldots,$ until convergence do,*

1. Compute $w := Av_j$.

2. For $i = \max\{1, j-k+1\}, \ldots, j$ do $\left\{ \begin{array}{c} h_{i,j} = (w, v_i), \\ w := w - h_{ij}v_i; \end{array} \right.$

Compute $h_{j+1,j} = \|w\|$ and $v_{j+1} = w/h_{j+1,j}$.

3. Update the QR factorization of $\bar{H}_j$, i.e.,

- *Apply the rotations Ω_i, $i = \max\{1, j-k\}, \ldots, j-1$ to the j-th column of $\bar{H}_j$;*
- *Compute the rotation coefficients c_j, s_j by*

$$\begin{aligned} s_i &= h_{i+1,i}/\sqrt{(h_{ii}^{(i-1)})^2 + (h_{i+1,i})^2}, \\ c_i &= h_{ii}^{(i-1)}/\sqrt{(h_{ii}^{(i-1)})^2 + (h_{i+1,i})^2}. \end{aligned} \tag{3}$$

4. Apply rotation Ω_j, to $\bar{H}_j$ and γ_j, i.e., compute

- $\gamma_{j+1} := -s_j\gamma_j$,
- $\gamma_j := c_j\gamma_j$, *and,*
- $h_{jj} := c_j h_{jj} + s_j h_{j+1,j}$.

5. $p_j = \left(v_j - \sum_{i=j-k}^{j-1} h_{ij}p_i\right)/h_{jj}$.

6. $x_j = x_{j-1} + \gamma_j p_j$.

7. • *If we have reached desired solution, e.g., $|\gamma_{j+1}|$ is small enough, stop;*

- *if a restart is desired, a residual norm minimization could be performed, then return to step 1;*
- *or else go to step 2.1 to continue with regular DQGMRES steps.*

Given this restarted DQGMRES algorithm, next we introduce four schemes of restarting and one scheme of minimizing the residual norm at the last step of the regular iteration. A review of residual norm computation will lead to these schemes.

In [9], it was shown that $|\gamma_{j+1}|$ can be used in place of the residual norm for the convergence test. For our experiment here we would like the convergence test for all the iterative solvers to be based on the exact residual norm. From equation (2), the residual can be written as

$$\begin{aligned} r_j &= V_{j+1} Q_j^T Q_j \left(\gamma_1 e_1 - \bar{H}_j y\right) \\ &= V_{j+1} Q_j^T \left(\bar{g}_j - H_j^{(j)} y\right). \end{aligned} \qquad (4)$$

where $\bar{g}_j = [\gamma_1, \ldots, \gamma_{j+1}]^T$, $H_j^{(j)} = Q_j \bar{H}_j$. The superscript (j) indicates the number of rotations applied [9]. For convenience, define $Z_{j+1} \equiv V_{j+1} Q_j^T$. For the normal DQGMRES steps, $\bar{g}_j - H_j^{(j)} y = \gamma_{j+1} e_{j+1}$, therefore

$$r_j = \gamma_{j+1} z_{j+1} \qquad (5)$$

where the vector z_{j+1} is the last column of Z_{j+1}. It is easy to verify that this last column can be updated from v_{j+1} and z_j using the following formula,

$$z_{j+1} = -s_j z_j + c_j v_{j+1}. \qquad (6)$$

The z_i's can be updated at the cost of one extra vector in memory and $3n$ operations at each step. The exact residual now can be computed using equation (5).

There is another use of equation (6) which leads to a way of minimizing the residual norm of the DQGMRES method. In deriving equation (5) it is assumed that $h_{j+1,j} = 0$ after the rotation Ω_j is applied (step 2.4 of Algorithm 1, see [9] for description of the rotations). If we apply an arbitrary rotation instead, and carry out the rest of the steps of Algorithm 1, the residual can be computed from equation (4). After we have applied all the previous rotations in step 2.3 of DQGMRES, the variables that directly influence the selection of the final rotation angle and the residual norm are:

$$h_{jj}^{(j-1)}, \quad h_{j+1,j}^{(j-1)}, \quad \gamma_j^{(j-1)}, \quad v_{j+1}, \quad z_j,$$

where the superscript $(j-1)$ indicates that the rotations Ω_i, $i = 1, \ldots, j-1$, have been applied. If instead of computing the last rotation by equation (3), we use a rotation of arbitrary angle, say θ, and solve the resulting overdetermined linear system by discarding the the last equation, the residual $\bar{r}_j \equiv \bar{g}_j - H_j^{(j)} y$ is

$$\begin{aligned} \bar{r}_j &= \Omega_\theta \left[\begin{pmatrix} \bar{g}_{j-1} \\ 0 \end{pmatrix} - H_j^{(j-1)} y \right] \\ &= -\gamma_j^{(j-1)} \left(\sin\theta + \cos\theta \frac{-h_{jj}^{(j-1)} \sin\theta + h_{j+1,j}^{(j-1)} \cos\theta}{h_{jj}^{(j-1)} \cos\theta + h_{j+1,j}^{(j-1)} \sin\theta} \right) e_{j+1} \\ &= -\frac{\gamma_j^{(j-1)} \sin\psi_j}{\cos(\psi_j - \theta)} e_{j+1}, \end{aligned}$$

where Ω_θ denotes the rotation of angle θ, and ψ_j denotes the rotation angle of Ω_j (see equation (3)). Combining this with equation (4), the residual norm after this arbitrary rotation θ is

$$\|r_j\|^2 = \left(\cos^2\theta - \eta_j \sin(2\theta) + \zeta_j^2 \sin^2\theta\right) \left(\frac{\gamma_j^{(j-1)} \sin\psi_j}{\cos(\psi_j - \theta)} \right)^2, \qquad (7)$$

where $\eta_j = z_j^T v_{j+1}$, and $\zeta_j = \|z_j\|$. It is clear that θ alone determines $\|r_j\|$ in this formula. The value of $\|r_j\|$ can be minimized by a few steps of Newton iteration. The main cost of minimizing (7) is computing η_j.

If $\theta = \pi/2$, then according to equation (7) we have

$$\|r_j\|^2_{\theta=\pi/2} = \left(\zeta_j \gamma_j^{(j-1)}\right)^2.$$

If $(j-1)$ step is a regular DQGMRES step, according to equation (5), $\|r_{j-1}\| = \zeta_j \gamma_j^{(j-1)}$. This indicates that if the above minimization is performed, then

$$\|r_j\| \le \|r_{j-1}\|.$$

We don't need to worry about the division by zero in the case $\theta = \pi/2$ and $\psi_j = 0$, because this implies that the exact solution is reached, and it can be computed [9].

It is not useful to perform the minimization at an arbitrary step of DQGMRES since the later QR factorization steps will cancel the effect. So this minimization can only be applied at the very last step of the DQGMRES iteration. If we do not restart, there is only one chance for it to be used. However if we restart, more opportunity will be created for performing this minimization. This is one reason to restart the DQGMRES method. Here are three more reasons for restarting.

.0.a DQGMRESh. We observe that the Incomplete Arnoldi process (step 2.2 of algorithm 1) approximates the upper Hessenberg matrix produced by the complete Arnoldi process with a banded upper Hessenberg matrix. From this point of view, $h_{1,j+1}$ could be used as a measure of the *approximation error*. Since this is only a heuristic measure of the error, we approximate $h_{1,j+1}$ by $v_1^T v_{j+1}$, where v_1 and v_{j+1} are produced by the DQGMRES iterations. In our implementation of this scheme, restart is performed if $|v_1^T v_{j+1}|$ becomes greater than 0.3. This scheme requires one extra vector storage to save v_1 and one inner-product to compute $v_1^T v_{j+1}$. It is called DQGMRESh(k).

.0.b DQGMRESz. Loss of orthogonality among $v_i (i = 1, \ldots, j+1)$ is a drawback of the DQGMRES algorithm. Our second strategy for restarting DQGMRES is based on an indicator of this orthogonality loss. The choice of this indicator for DQGMRES is $\zeta_j = \|z_j\|$. If V_j is an orthonormal basis set, $\zeta_j = 1$. When V_j loses its orthogonality, the effect is accumulated in ζ_j through equation (6). In our implementation of this scheme we restart when $\zeta_j > 1.5$ or $\zeta_j < 2/3$. This variant is called DQGMRESz(k).

.0.c DQGMRESr. In GMRES the residual norm will never increase, the same is not true in the DQGMRES method. One way of preventing residual norm from increasing is to discard the last step and restart when $\|r_{j+1}\| > \|r_j\|$. This generates a version of DQGMRES where the residual norm decreases monotonically. It is called DQGMRESr(k). If we don't want to discard the last step of DQGMRES, a minimization of the residual norm can be performed. This gives us the second variant where the residual norm decreases monotonically, DQGMRESrm(k).

.0.d DQGMRESrz. If we compute the exact residual norm in DQGMRES, the last two restarting schemes can be realized with a minimal amount of work. We also implemented a combined scheme, i.e., restart DQGMRES if ζ_j goes out of range [2/3, 1.5] or $\|r_{j+1}\| > \|r_j\|$. We name it DQGMRESrz(k). Altogether we have four strategies for restarting DQGMRES. The residual norm minimization can be applied to all four restarting options. The following table lists the names of all ten DQGMRES routines used in our experiments.

	without minimization	*with minimization*
no restart	DQGMRES	DQGMRESm
restart on $h_{1,j+1}$	DQGMRESh	DQGMREShm
restart on $\|r_j\|$	DQGMRESr	DQGMRESrm
restart on ζ_j	DQGMRESz	DQGMRESzm
both $\|r_j\|$ *and* ζ_j	DQGMRESrz	DQGMRESrzm

III. Experiments

The linear systems used in our tests are constructed from matrices stored in the Harwell-Boeing format [3]. The right-hand sides of test linear systems are either the right-hand sides contained in the Harwell-Boeing files, or random vectors with values in the range of (-1, 1). The initial guesses are the initial guesses provided in the Harwell-Boeing files, or the zero vector.

Our matrix set includes most of the matrices from the Harwell-Boeing collection [3] and many matrices from various ftp sites set up by individual researchers. Table I lists their average sizes and maximum sizes, where $\overline{N}$ denotes the average matrix dimension, $\overline{NNZ}$ denotes average number of non-zero elements, N_{max} denotes the largest dimension of the matrices, and NNZ_{max} denotes the largest number of non-zero elements.

Altogether, there are 291 matrices, out of which 107 are symmetric and 184 are unsymmetric. In our experiment, we consider an iterative solver reached convergence if

$$\|r_j\| \leq 10^{-6}\|r_0\| + 10^{-12}.$$

A maximum of 1000 matrix-vector multiplications is allowed for any iterative method.

For all the tests we will show here, k is always 20, i.e., DQGMRES(20), DQGMRESm(20), etc. will be used in our comparisons.

Table II shows the number of unsymmetric linear systems converged using the iterative solvers. The numbers heading the columns of table II are the number of matrix-vector multiplications used. There are 184 unsymmetric systems; about 7% of them can be solved with less than 100 matrix-vector multiplications. The total number of linear systems solved by any iterative solver without preconditioning is 36. Compared against the non-restarted DQGMRES(20), DQGMRESrm(20), DQGMRESrz(20), and DQGMRESrzm(20) converged on significantly more linear systems.

On symmetric linear systems, the ten different variants of DQGMRES perform identically. No restart was performed by any of the restarted version of the DQGMRES.

MATVEC	100	200	500	1000
DQGMRES(20)	13	16	20	25
DQGMRESm(20)	13	16	20	25
DQGMRESh(20)	13	16	21	29
DQGMREShm(20)	13	16	22	28
DQGMRESz(20)	13	16	22	31
DQGMRESzm(20)	13	16	21	30
DQGMRESr(20)	13	16	27	34
DQGMRESrm(20)	13	16	26	35
DQGMRESrz(20)	13	16	26	35
DQGMRESrzm(20)	13	16	27	35

TABLE II
Number of unsymmetric linear systems solved without preconditioning.

Four fifths of the unsymmetric linear systems can not be solved without preconditioning. Preconditioning is needed. Two types of preconditioners will be used, scaling and incomplete LU factorization. This is a brief description of the four preconditioning routines used.

- Scaling. The matrix is scaled both from left and right. First, each column of the matrix is scaled by its 1-norm, then, each row is scaled by its 1-norm.
- ILU0. ILU0 is an Incomplete LU factorization that uses the original sparsity pattern of the matrix for the fill-ins.
- ILUT. The ILUT(k, ϵ) preconditioner is an incomplete LU factorization with a dual dropping strategy [11]. The number of fill-in elements in both L and U factors are limited by k, and those elements with magnitude less than ϵ of their corresponding row norms are dropped. The ILUT preconditioner used here is ILUT(10, 10^{-4}).
- ILUTP. The ILUTP(k, ϵ, τ) preconditioner is an extension of ILUT which performs column pivoting if $\tau \times (\max_{j=i+1}^{n} |a_{ij}|) > |a_{ii}|$. A series of ILUTP preconditioners are used, namely ILUTP(10, 10^{-8}, 0.02), ILUTP(20, 10^{-16}, 0.02) and ILUTP(50, 10^{-16} 0.02).

We start with a simple diagonal scaling as preconditioner. If a linear system is not converged with a simple preconditioner, a more resource consuming preconditioner will be used. This mimics a typical process where a user is selecting a preconditioner.

Table III shows the results of attempting to solve the nonsymmetric linear systems with diagonal scaling as the preconditioner. Similar to the unpreconditioned case, DQGMRESrz(20) and DQGMRESrzm(20) converged on 9 more linear system than DQGMRES(20).

In the rest of the experiments, we always applied scaling before performing ILU factorization. The ILU preconditioners are applied from the right. They are used in the following order,

- First apply ILU0 on the 132 unsymmetric linear system not converged with diagonal preconditioning. There were 12 them reached convergence with ILU0 preconditioner.
- Apply ILUT(10, 10^{-4}) (referred to as ILUT10) on the 120 linear systems not yet converged. This time, 18 linear systems converged.

source	# of unsymm	# of symm	$\overline{N}$	$\overline{NNZ}$	N_{max}	NNZ_{max}
HB(RUA/RSA)	95	70	1,400	22,600	15,439	543,162
FIDAP	22	13	3,400	106,000	19,716	1,096,948
Driven Cavity	26	0	2,805	83,511	4,562	138,187
U. of Florida	19	3	12,957	97,791	76,480	335,872
Heroux, Cray	0	21	3,509	181,968	3,509	181,968
Simon, NASA	12	0	27,592	1,140,150	62,424	1,853,104
Tokamak	5	0	2540	34,406	5940	83,842
Barth, NASA	4	0	14,905	960,600	15,735	1,510,325
Wigton, Boeing	1	0	3864	238,252	3864	238,252

TABLE I

General information about the test matrices.

MATVEC	100	200	500	1000
DQGMRES(20)	24	30	38	43
DQGMRESm(20)	24	30	38	43
DQGMRESh(20)	26	30	39	45
DQGMREShm(20)	26	30	39	46
DQGMRESz(20)	24	32	42	49
DQGMRESzm(20)	24	32	42	50
DQGMRESr(20)	24	32	45	51
DQGMRESrm(20)	24	32	44	51
DQGMRESrz(20)	24	33	46	52
DQGMRESrzm(20)	24	33	45	52

TABLE III

Number of unsymmetric linear systems solved with scaling.

MATVEC	50	100	200	500	1000
DQGMRES(20)	66	80	92	105	112
DQGMRESm(20)	66	80	92	105	112
DQGMRESh(20)	66	83	97	114	121
DQGMREShm(20)	66	83	97	115	123
DQGMRESz(20)	66	80	101	117	126
DQGMRESzm(20)	66	79	102	118	128
DQGMRESr(20)	66	85	100	119	126
DQGMRESrm(20)	67	83	102	118	126
DQGMRESrz(20)	66	85	101	120	126
DQGMRESrzm(20)	67	83	104	120	128

TABLE IV

Number of unsymmetric linear systems solved using the preconditioner sequence.

- Apply ILUTP(10, 10^{-8}, 0.02) (ILUTP10) on the remaining 102 linear systems still to be solved. We were able to solve 29 problems with ILUTP10.
- Apply ILUTP(20, 10^{-16}, 0.02) (ILUTP20) on the 73 unsolved linear systems. It solved another 11 systems.
- Apply ILUTP(50, 10^{-16}, 0.02) (ILUTP50) on the remaining 62 linear systems. Another 6 systems were solved.

The results of using the preconditioner sequence is shown in Table IV. Overall, after 1000 matrix-vector multiplications, DQGMRESzm(20) and DQGMRESrzm(20) are the most successful ones in terms of number of linear systems solved. The differences between the restarted variants and the non-restart version is more pronounced as the number of matrix-vector multiplications increases in Tables II, III, and IV.

Table V shows the results of applying ILU0 on the symmetric linear systems. Since the preconditioner is applied from the right, the actual iteration matrix is nonsymmetric. However, unlike the case where the matrices of the linear systems are unsymmetric, the differences among the ten DQGMRES variants are relatively small. This result is close to the unpreconditioned symmetric case where the ten variants are identical to each other.

IV. Summary

In this talk, a total of ten different variations of DQGMRES are tested on a large number of linear system from various applications. Overall, we see that the restarted variants can perform considerably better than the original version, especially if it takes more than a few hundreds of matrix-vector multiplications for DQGMRES to reach convergence. The number of linear systems converged within first 100 matrix-vector multiplications is often more than three quarters of the number of linear systems converged within 200 matrix-vector multiplications. In practice, this means that letting DQGMRES iterate more steps does not proportionally increase the chance that it will converge on a linear system. Restarting can help remedy this problem to some extend. This observation suggests that restarting other short recurrences based Krylov methods might be also beneficial.

Among the restarted variants, DQGMRESz, DQGMRESzm, DQGMRESr, DQGMRESrm, DQGMRESrz, DQGMRESrzm all showed that they can work well in practice. In particular, DQGMRESrzm combines three successful strategies, it outperforms other methods tested in many cases. DQGMRESr seems to the least complex of

MATVEC	50	100	200	500
DQGMRES(20)	30	36	52	63
DQGMRESm(20)	30	36	52	63
DQGMRESh(20)	30	36	52	63
DQGMREShm(20)	30	36	52	63
DQGMRESz(20)	30	37	54	63
DQGMRESzm(20)	30	36	57	63
DQGMRESr(20)	30	36	59	64
DQGMRESrm(20)	30	36	58	64
DQGMRESrz(20)	30	36	58	64
DQGMRESrzm(20)	30	36	57	64

TABLE V
Number of symmetric linear systems solved with ILU0 preconditioning.

the restarted variants, it is almost as effective as DQGMRESrzm.

References

[1] Owe Axelsson. *Iterative Solution Methods*. Press Syndicate of the University of Cambridge, Cambridge, UK, 1994.

[2] Peter N. Brown. A theoretical comparison of the Arnoldi and GMRES algorithms. *SIAM J. Sci. Statist. Comput.*, 12(1):58–78, 1991.

[3] I. S. Duff, R. G. Grimes, and J. G. Lewis. Sparse matrix test problems. *ACM Trans. Math. Soft.*, pages 1–14, 1989.

[4] W. D. Gropp and D. E. Keyes. Parallel performance of domain-decomposed preconditioned Krylov methods for PDE's with locally uniform refinement. *SIAM J. Sci. Statist. Comput.*, 13:128–145, 1992.

[5] M. R. Hestenes and E. Stiefel. Methods of Conjugate Gradient for solving linear systems. *Journal of Research of the National Bureau of Standards*, 49:409–436, 1952.

[6] Zhongxiao Jia. On IGMRES(q): incomplete generalized minimal residual methods for large unsymmetric linear systems. Technical Report 94-047, Department of Mathematics, University of Bielefeld, Sonderforschungsbereich 343, 1994. Last revision March, 1995.

[7] Wayne Joubert. A robust GMRES-based adaptive polynomial preconditioning algorithm for nonsymmetric linear systems. *SIAM J. Sci. Comput.*, 15(2):427–439, March 1994.

[8] Y. Saad and M. H. Schultz. GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM J. Sci. Statist. Comput.*, 7:856–869, 1986.

[9] Y. Saad and K. Wu. DQGMRES: a quasi-minimal residual algorithm based on incomplete orthogonalization. Technical Report UMSI-93/131, Minnesota Supercomputing Institute, University of Minnesota, Minneapolis, 1993.

[10] Yousef Saad. Krylov subspace methods on supercomputers. *SIAM J. Sci. Statist. Comput.*, 10:1200–1232, 1989.

[11] Yousef Saad. ILUT: a dual threshold incomplete ILU factorization. *Numerical Linear Algebra with Applications*, 1:387–402, 1994. Technical Report 92-38, Minnesota Supercomputer Institute, University of Minnesota, 1992.

[12] Yousef Saad. *Iterative Methods for Sparse Linear Systems*. PWS publishing, Boston, MA, 1996.

[13] J. N. Shadid and R. S. Tuminaro. A comparison of preconditioned nonsymmetric Krylov methods on a large-scale MIMD machine. *SIAM J. Sci. Comput.*, 15:440–459, 1994.

[14] T. E. Tezduyar, M. Behr, S. K. Aliabadi, S. Mittal, and S. E. Ray. A new mixed preconditioning method for finite element computations. *Computer Methods in Applied Mechanics and Engineering*, 99:27–42, 1992.

[15] C. Vuik and H. A. van der Vorst. A comparison of some GMRES-like methods. *Lin. Alg. Appl.*, 160:131–162, 1992.

[16] Rüdiger Weiss. A theoretical overview of Krylov subspace methods. *Applied Numerical Mathematics*, 19:33–56, 1995.

[17] Ulrike Meier Yang. A new family of iterative solvers for nonsymmetric linear system. Technical Report 1407, CSRD, University of Illinois at Urbaba-Champaign, 1994.