Adv Comput Math (2007) 27:423–448
DOI 10.1007/s10444-006-9012-5

Dynamic block GMRES: an iterative method for block linear systems

R. D. da Cunha · D. Becker

Received: 15 March 2004 / Accepted: 8 November 2005 /
Published online: 13 December 2006

Abstract We present variants of the block-GMRES(m) algorithms due to Vital and the block-LGMRES(m,k) by Baker, Dennis and Jessup, obtained with replacing the standard QR factorization by a rank-revealing QR factorization in the Arnoldi process. The resulting algorithm allows for dynamic block deflation whenever there is a linear dependency between the Krylov vectors or the convergence of a right-hand-side occurs. FORTRAN 90 implementations of the algorithms were tested on a number of test matrices and the results show that in some cases a substantial reduction of the execution time is obtained. Also a parallel implementation of our variant of the block-GMRES(m) algorithm, using FORTRAN 90 and MPI was tested on SUNFIRE 15K parallel computer, showing good parallel efficiency.

Keywords iterative methods · GMRES · block systems · parallel computing

Mathematics Subject Classification (2000) 65F10

1. Introduction

This paper describes changes to block-GMRES algorithms when replacing the ordinary QR factorization by a rank-revealing QR factorization (RRQR). The resulting algorithm allows for the detection of early convergence of one or more right-hand-sides (rhs) during the execution of the algorithm, and of linear dependency between

This work was carried out while the author was at IM/UFRGS.

R. D. da Cunha (✉)
Instituto de Matemática, Universidade Federal do Rio Grande do Sul, Rio Grande, Brazil
e-mail: rcunha@mat.ufrgs.br

D. Becker
Applied Mathematics and Computing Group, School of Engineering,
Cranfield University, Cranfield, UK
e-mail: d.becker.2002@cranfield.ac.uk

two or more Krylov vectors. In such cases, the block size is dynamically reduced accordingly; that is to say the algorithm continues but with a smaller block size. For any large problems, $AX = B$, this ability to reduce block size dynamically represents a significant savings in the memory requirements for storing the orthogonal basis. Furthermore a smaller block size substantially reduces the computational cost for generating successive orthogonal vectors of the Krylov subspace. This strategy may also help in the convergence of a block-GMRES method, since the numerical linear dependency between the Krylov vectors may lead to a higher number of iterations, as noted in [10]; our experiments have shown that in some test cases (but not always) there is a reduction in the number of iterations. The RRQR factorization using column pivoting (see, for instance, [4, 7, 8] (pp. 233–236) and [3]) requires a few more operations than the non-pivoting, standard QR factorization, in case a matrix has full column-rank. Also, the authors have developed a parallel algorithm for the RRQR factorization [5] that provide good scalability with respect to the number of processors. Thus, in the cases where early convergence does not occur, there is no significant loss of parallel efficiency in the use of the dynamic block-GMRES over the standard block-GMRES.

A brief overview of the paper is as follows: Section 2 provides a brief overview of the block-GMRES algorithms used. Section 3 describes the block-Arnoldi process in view of the use of the RRQR factorization and its connection with the block-GMRES algorithms. Section 4 presents the experimental results obtained with FORTRAN 90 implementations of the new algorithms. A parallel implementation of the dynamic block-GMRES algorithm is given in Section 5. Finally in Section 6 we present our conclusions.

2. Overview of block-GMRES algorithms

We consider the problem of solving a non-singular system of n nonsymmetric linear equations with t rhs vectors,

$$A x_{(i)} = b_{(i)}, i = 1, \ldots, t, \tag{1}$$

or, in matrix form,

$$A_{n\times n} X_{n\times t} = B_{n\times t}, \tag{2}$$

where A is typically sparse and large. The rhs vectors may be given as a result of some specific formulation; of several different experiments; or as multiple copies of a single $b_{n\times 1}$ vector. The latter case is a strategy used to accelerate the process of solving the system; using an iterative method implies the use of t different guesses to the solution, and this has implications with the use of block-GMRES algorithms, as we present in Section 4.

Our work is based on the B-GMRES(m) algorithm due to Vital [14] as presented by Saad ([12], pp. 208–212) and the B-LGMRES(m,k) by Baker et al. [1] (a block variant of the Loose GMRES method [2]). Block variants of the GMRES algorithm are based on extensions of the Arnoldi process to obtain a Krylov basis of a set of vectors, instead of a single one. In [12], three different versions of the Arnoldi process are given; we have used the block-Arnoldi algorithm with a block modified Gram-Schmidt, as detailed in Section 3. In the block-GMRES algorithms, the block-Arnoldi process is used to produce a Krylov basis such that improved approximations to the solution (i.e. with diminishing residual norms) are taken as a linear combination of the previous approximation and specific vectors in that basis, selected as the solution of a least-squares problem.

In the B-GMRES(m) algorithm, for a given initial guess X_0, the residual $R_0 = B - AX_0$ is factored in the form $R_0 = V_m R$ via the QR factorization, where V_m is unitary and $R_{t\times t}$ is upper triangular. Upon the V matrix, the block-Arnoldi algorithm produces an $(m+t)\times m$ block Hessenberg matrix H_m and an $(m+t)\times t$ matrix G whose columns are $E_1 Re_i$ (E_1 is an $(m+t)\times t$ matrix whose $t\times t$ upper principal block is the identity matrix and e_i is the ith canonical vector). Then the initial guess is refined as $X_0 + V_m Y$, where Y is obtained as the solution of the least-squares problem $||\bar{G} - \bar{H}_m Y||_2$; $\bar{G}$ is $m\times t$ and $\bar{H}_m$ is $m\times m$. The number of vectors used in the least-squares problem is mt.

In the B-LGMRES(m,k) algorithm, some k vectors $Z_i = X_i - X_{i-1}$, $i = 1, 2, \ldots, k$, are used together with the residual vectors. This would hopefully reduce the 'zig-zag' behaviour (with respect to the path taken to the solution) exhibited by GMRES(m) once the restarting takes effect. Instead of using only the residuals in the block-Arnoldi method, the initial basis vectors are augmented by the k Z_i vectors; note that k iterations of the standard GMRES(m) method should be made, computing and storing those vectors. Then the initial guess will be refined as $X_0 + W_m Y$, where W_m is an $n\times(m+k)$ matrix whose first m columns are the column vectors of V_m (as above) and the remaining k columns are the vectors Z_i, and Y is the solution of a similar least-squares problem; the number of vectors used in the least-squares problem is $(m+k)t$.

3. Block-Arnoldi process

Algorithm 1 shows the block-Arnoldi process, expressed using the modified Gram-Schmidt process. Given the following notation,

$$U_m = \left[V_1, V_2, \ldots, V_m\right], \tag{3}$$

$$H_m = \begin{pmatrix} H_{1,1} & H_{1,2} & H_{1,3} & \ldots & H_{1,m} \\ H_{2,1} & H_{2,2} & H_{2,3} & \ldots & H_{2,m} \\ & H_{3,2} & H_{3,3} & \ldots & H_{3,m} \\ & & H_{4,3} & \ldots & H_{4,m} \\ & & & \ddots & \vdots \\ & & & & H_{m+1,m} \end{pmatrix}_{(m+1)\times m} \tag{4}$$

1. Choose an unitary matrix V_1 of size $n\times t$
   for $j = 1, 2, \ldots, m$ do
2. $\quad W_j = AV_j$
   $\quad$ for $i = 1, 2, \ldots, j$ do
3. $\quad\quad H_{i,j} = V_i^T W_j$
4. $\quad\quad W_j = W_j - V_i H_{i,j}$
   $\quad$ endfor
5. $\quad$ Compute the QR factorization: $W_j = V_{j+1} H_{j+1,j}$
   endfor

Algorithm 1. *Block-Arnoldi with block-MGS.*

where H_m is a $(m+1) \times m$ block matrix and the $H_{i,j}$ are defined as in Algorithm 1 (with $H_{i,j} = 0$ when $i > j+1$), it can be observed that:

1. The following relationship holds,

$$AV_m = V_{m+1} H_m \tag{5}$$

2. H_m has the structure of a block upper Hessenberg matrix with t subdiagonals.
3. $H_{j+1,j}$, $j = 1, 2, \ldots, m$, are upper triangular blocks of dimension $t \times t$.
4. Each V_i, $i = 1, 2, \ldots, m$, has dimension $n \times t$.

However, if we replace the QR factorization in the block-Arnoldi algorithm by RRQR the last two observations may no longer hold.

3.1. Ramifications of replacing QR by RRQR in the block-Arnoldi algorithm

A RRQR factorization (with column pivoting) is of the form

$$WP = QR = Q \begin{pmatrix} R_{11} & R_{12} \\ 0 & R_{22} \end{pmatrix}, \tag{6}$$

$$\| R_{22} \| \approx 0, \tag{7}$$

where P is a permutation matrix, Q has orthonormal columns, R is upper triangular and R_{11} is of order r, the numeric rank of W. There are several different algorithms to compute the RRQR factorization with column pivoting (see [4, 7, 8] (pp. 233–236) and [3]); it is also possible to rewrite the Modified Gram-Schmidt algorithm such that numerical linear dependency between the columns may be detected. In our work we have used both this latter approach as well as the PRRQR algorithm [5] with good results.

Now suppose that on step no. 5 of the block-Arnoldi algorithm we compute W_j using the above RRQR. It then follows that by defining V_{j+1} to be the first r columns of Q and $H_{j+1,j}$ to be the first r rows of RP^T we have

$$W_j = V_{j+1} H_{j+1,j}. \tag{8}$$

It is noteworthy to observe that H_m as defined above still has the structure of a block upper Hessenberg matrix when we use RRQR in place of QR, but with $2t-1$ subdiagonals. What changes is that the number of rows in block-row i of H_m equals the rank of V_i. Similarly the number of columns in block-column j of H_m equals the rank of V_j. The fact that the $H_{j+1,j}$ submatrices are not necessarily upper triangular effects ones choice of orthogonal transformations used to reduce H_m to upper triangular form.

3.2. Ramifications of RRQR within the Block-GMRES algorithms

For the purpose of describing our algorithms we adopt the following notation:

1. RRQR $[V_{j+1}, H_{j+1,j}, d(j+1)] = W_j$, where $d(j+1) = rank\left(W_j\right)$ and the definitions of V_{j+1} and $H_{j+1,j}$ are as described in Section 3.1.
2. The matrix W of dimension $n \times r$ is denoted by $(W)_{n\times r}$.
3. The number of vectors in the Krylov basis is $l = \sum_{i=1}^{j+1} d(i)$.

Both B-GMRES(*m*) and B-LGMRES(*m*,*k*) may have their block-Arnoldi part modified as above, and these modifications lead to the Dynamic Block-GMRES(*m*) – DB-GMRES(*m*) – and the Dynamic Block-LGMRES(*m*,*k*) – DB-LGMRES(*m*,*k*), as presented in Algorithms 2 and 3, respectively, where individual column vectors are indicated by indices enclosed in parentheses, as in $x_{(i)}$, $z_{(i)}$ and *tol* is some prescribed tolerance for convergence. Note that the convergence check is made at the end of

1. $R_0 = B - AX_0$
   for $q = 1, 2, \ldots$ do
2. RRQR$[\ (V_1)_{n\times d(1)}, R_{d(1)\times d(1)}, d(1)\ ] = (R_{q-1})_{n\times t}$
3. $G = [\ R_{d(1)\times d(1)}\ 0\ ]^T$
   for $j = 1, 2, \ldots, m$ do
4. $(W_j)_{n\times d(j)} = AV_j$
   for $i = 1, 2, \ldots, j$ do
5. $(H_{i,j})_{d(i)\times d(j)} = V_i^T W_j$
6. $(W_j)_{n\times d(j)} = W_j - V_i H_{i,j}$
   endfor
7. RRQR$[\ V_{j+1}, H_{j+1,j}, d(j+1)\ ] = (W_j)_{n\times d(j)}$
8. Apply previous Householder reflections to new columns of H
9. Apply Householder reflections to triangularize $(H_{j+1,j})_{(d(j)+d(j+1))\times d(j)}$, apply also to G
   endfor
10. Compute Krylov basis dimension $l = \sum_{j=1}^{m} d(j)$
11. $\beta_{(i)} = \|\ G(l+1 : l+d(1), i)\ \|_2, i = 1, 2, \ldots, d(1)$
12. Solve the least-squares problem, $\|\ \bar{G}_{l\times d(1)} - \bar{H}Y\ \|_2$
13. Compute the approximate solution $(X_q)_{n\times d(1)} = X_{q-1} + VY$
    for $i = t, t-1, \ldots, 1$ do
14. if $\beta_{(i)} < tol$ then
    $x_{(i)} \leftrightarrow x_{(t)}; b_{(i)} \leftrightarrow b_{(t)}; t = t-1$
    endif
    endfor
15. if "all systems have converged" then stop
16. $R_q = B - AX_q$
    endfor

Algorithm 2. *Dynamic Block-GMRES (m).*

the inner iterations (step 11 on Algorithm 2 and step 14 on Algorithm 3), though it could be made inside; it is also important to note that on DB-LGMRES(*m*,*k*) a

1. $R_0 = B - AX_0$
   for $q = 1, 2, \ldots$ do
      if $q \leq k$ then
2. $s = m$
      else
3. $s = m + k$
      endif
4. RRQR[$(V_1)_{n\times d(1)}, R_{d(1)\times d(1)}, d(1)$] $= (R_{q-1})_{n\times t}$
5. $G = [\ R_{d(1)\times d(1)}\ 0\ ]^T$
      for $j = 1, 2, \ldots, s$ do
         if $j \leq m$ then
6. $(W_j)_{n\times d(j)} = AV_j$
         else
7. $(W_j)_{n\times d(j)} = AZ_{j-m}$
         endif
         for $i = 1, 2, \ldots, j$ do
8. $(H_{i,j})_{d(i)\times d(j)} = V_i^T W_j$
9. $(W_j)_{n\times d(j)} = W_j - V_i H_{i,j}$
         endfor
10. RRQR[$V_{j+1}, H_{j+1,j}, d(j+1)$] $= (W_j)_{n\times d(j)}$
11. Apply previous Householder reflections to new columns of H
12. Apply Householder reflections to triangularize
         $(H_{j+1,j})_{(d(j)+d(j+1))\times d(j)}$, apply also to G
      endfor
13. Compute Krylov basis dimension $l = \sum_{j=1}^{s} d(j)$
14. $\beta_{(i)} = \| \ G(l+1 : l+d(1), i)\ \|_2, i = 1, 2, \ldots, d(1)$
15. Solve the least-squares problem, $\| \ \bar{G}_{l\times d(1)} - \bar{H}Y\ \|_2$
      if $q \leq k$ then
16. $Z_q = VY$
      else
        $Z_{1:q-1} = Z_{2:q}$; $Z_q = VY$
      endif
17. Compute the approximate solution $(X_q)_{n\times d(1)} = X_{q-1} + Z$
      for $i = t, t-1, \ldots, 1$ do
18. if $\beta_{(i)} < tol$ then
            $x_{(i)} \leftrightarrow x_{(t)}$; $b_{(i)} \leftrightarrow b_{(t)}$;
            Delete vectors $z_{(i)}, z_{(i+t)}, \ldots, z_{(i+kt)}$; $t = t - 1$
        endif
      endfor
19. if "all systems have converged" then stop
20. $R_q = B - AX_q$
   endfor

Algorithm 3. *Dynamic Block-LGMRES (m,k).*

housekeeping procedure must be made once convergence for a given rhs is detected, as k Z vectors must be deleted before proceeding with the iteration.

4. Comparison with Block-GMRES

An experiment was carried out using test problems found in the MATRIX MARKET collection [11], using FORTRAN 90 implementations of B-GMRES(m) and B-LGMRES(m,k) and our dynamic-block variants, DB-GMRES(m) and DB-LGMRES(m,k). The RRQR factorization used with the latter two is a column-pivoting modification of the Modified Gram-Schmidt process. The test problems chosen were `CDDE1` ($n = 961$), `JPWH_991` ($n = 991$), `NOS3` ($n = 960$), `ORSIRR_1` ($n = 1030$), `ORSREG_1` ($n = 2205$), `PDE900` ($n = 900$) and `RDB1250` ($n = 1250$).

We chose a random rhs B with $t = 2, 4$ and 8 columns and $X_0 = B$. The iterations were allowed to proceed until $||\, r_{(i)} \,||_2 < 10^{-5}$, $i = 1, \ldots, t$, or the number of iterations was greater than n. In this experiment the systems were solved without the use of preconditioning. We used the values $m = 10$, 20 and 30 and $k = 1, 2, 3, 4$.

Also, to establish a fair comparison, we solved the same systems with the non-block GMRES(m) and LGMRES(m,k), implemented using Householder transformations (for the Householder GMRES implementation see [15]). To ensure that the same rhs vectors were used, they were computed previously and stored in unformatted FORTRAN 90 files for later use. The computer used had a Pentium 4 1.5 GHz processor with 256 MB of main memory running a Linux O/S. The compiler used was the INTEL FORTRAN COMPLER 8.1 and the source files were compiled with optimization (`-O3`) turned on.

Tables 1–7 present the number of iterations (designated q) and the runtime (in s) for the four methods considered, solving the systems above. The fourth and fifth columns present the number of iterations (designated q) and the runtime (designated T_1) for B-GMRES(m) and B-LGMRES(m,k) whereas the number of iterations and the runtime (designated T_2) for DB-GMRES(m) and DB-LGMRES(m,k) are presented in the sixth and seventh column, according to the appropriate values of m and k; the eighth column of each table shows the ratios T_1/T_2 between the runtime of B-GMRES(m) and DB-GMRES(m) (or B-LGMRES(m,k) and DB-LGMRES(m,k)); the next three columns show the minimum and maximum number of iterations and the overall runtime (designated T_3) obtained while solving the same systems with multiple applications of GMRES(m) and LGMRES(m,k); finally, the last two columns show the ratios, T_1/T_3 between the runtime of B-GMRES(m) and GMRES(m) (or B-LGMRES(m,k) and LGMRES(m,k)) and the ratios, T_2/T_3 of DB-GMRES(m) and GMRES(m) (or DB-LGMRES(m,k) and LGMRES (m,k)). The results show that in some cases the dynamic block variants do not solve the problem faster than the block methods; in some others they provide a faster solution, and in a few cases they are much slower. When comparing the block methods with multiple applications of GMRES(m) and LGMRES(m,k), in most of the cases they are faster.

Figures 1–7 show the residual history and evolution of the basis dimension as the iterations proceed; note that the graphs show all the residuals. One can notice that the reduction on the basis dimension usually means an increase in the residual norm, which is as expected; also noticeable is that the block methods, in some cases, show an oscillating behaviour, though the residual is decreasing in an overall sense.

Table 1 Results for problem `CDDE1`.

t	m	k	B-GMRES(m) B-LGMRES(m,k)		DB-GMRES(m) DB-LGMRES(m,k)		T_1/T_2	GMRES(m) LGMRES(m)			T_1/T_3	T_2/T_3
			q	$T_1(s)$	q	$T_2(s)$		$\min q$	$\max q$	$T_3(s)$		
2	10		962	24.72	145	3.34	7.40	962	962	30.12	0.82	0.11
		1	273	8.34	230	6.70	1.24	157	162	6.16	1.35	1.09
		2	228	7.95	99	3.49	2.28	88	171	5.68	1.40	0.61
		3	62	2.40	53	2.04	1.18	48	50	2.34	1.03	0.87
		4	49	1.69	41	1.83	0.92	37	41	2.10	0.80	0.87
	20		143	10.83	25	1.77	6.12	68	84	6.60	1.64	0.27
		1	32	2.68	39	3.23	0.83	22	26	2.34	1.15	1.38
		2	30	2.71	28	2.53	1.07	27	27	2.78	0.97	0.91
		3	22	2.09	100	8.17	0.26	30	40	3.80	0.55	2.15
		4	17	1.68	16	1.61	1.04	19	21	2.20	0.76	0.73
	30		28	4.23	12	1.72	2.46	23	24	3.80	1.11	0.45
		1	19	3.08	16	2.64	1.17	16	17	2.88	1.07	0.92
		2	15	2.55	15	2.58	0.99	12	15	2.50	1.02	1.03
		3	11	1.91	33	5.25	0.36	15	17	2.96	0.65	1.77
		4	10	1.76	11	2.02	0.87	14	16	3.00	0.59	0.67
4	10		962	59.82	962	61.73	0.97	962	962	63.60	0.94	0.97
		1	272	20.27	142	10.66	1.90	157	200	13.58	1.49	0.78
		2	136	11.56	66	5.85	1.98	88	163	10.00	1.16	0.58
		3	129	12.42	53	5.13	2.42	49	126	8.36	1.49	0.61
		4	29	2.97	63	6.60	0.45	37	66	5.64	0.53	1.17
	20		157	29.67	55	9.36	3.17	73	80	13.16	2.25	0.71
		1	46	9.67	26	4.69	2.06	26	27	5.10	1.90	0.92
		2	35	8.07	13	3.07	2.63	28	30	5.94	1.36	0.52
		3	23	5.60	18	4.51	1.24	26	39	6.98	0.80	0.65
		4	13	3.22	26	5.03	0.64	18	21	4.38	0.74	1.15
	30		22	8.50	22	6.42	1.32	22	27	7.82	1.09	0.82
		1	17	6.54	19	6.04	1.08	12	16	5.12	1.28	1.18
		2	12	5.27	18	6.67	0.79	16	17	6.02	0.88	1.11
		3	9	4.06	7	3.07	1.32	15	17	6.06	0.67	0.51
		4	10	4.62	8	3.59	1.29	14	16	5.52	0.84	0.65
8	10		962	165.10	962	174.47	0.95	962	962	122.92	1.34	1.42
		1	365	74.77	125	10.61	7.05	121	229	25.72	2.91	0.41
		2	137	31.59	50	11.64	2.71	79	170	23.18	1.36	0.50
		3	70	15.39	28	6.58	2.34	51	170	14.14	1.09	0.47
		4	36	10.25	22	6.09	1.68	39	132	12.38	0.83	0.49
	20		102	54.85	90	39.78	1.38	69	86	25.54	2.15	1.56
		1	46	27.33	47	20.13	1.36	21	32	10.02	2.73	2.01
		2	17	10.23	11	6.53	1.57	23	39	10.98	0.93	0.59
		3	16	10.67	17	7.62	1.40	28	32	12.58	0.85	0.61
		4	13	8.89	15	9.42	0.94	18	22	8.68	1.02	1.09
	30		10	11.01	10	7.51	1.47	21	32	15.94	0.69	0.47
		1	7	8.61	11	8.24	1.04	13	17	10.64	0.81	0.77
		2	6	7.35	5	5.64	1.30	13	17	10.92	0.67	0.52
		3	6	7.44	5	5.60	1.33	13	18	11.30	0.66	0.50
		4	7	8.72	5	4.98	1.75	11	15	10.08	0.87	0.49

Table 2 Results for problem `JPWH_991`.

t	m	k	B-GMRES(m) B-LGMRES(m,k)		DB-GMRES(m) DB-LGMRES(m,k)		T_1/T_2	GMRES(m) LGMRES(m)			T_1/T_3	T_2/T_3
			q	$T_1(s)$	q	$T_2(s)$		$\min q$	$\max q$	$T_3(s)$		
2	10		9	0.26	9	0.27	0.96	9	9	0.34	0.76	0.79
		1	8	0.26	9	0.30	0.87	8	8	0.38	0.68	0.79
		2	7	0.25	9	0.33	0.76	8	8	0.40	0.62	0.82
		3	7	0.26	9	0.31	0.84	7	7	0.34	0.76	0.91
		4	7	0.25	8	0.29	0.86	7	7	0.32	0.78	0.91
	20		4	0.33	4	0.34	0.97	4	4	0.32	1.03	1.06
		1	4	0.35	5	0.38	0.92	4	4	0.38	0.92	1.00
		2	3	0.26	5	0.39	0.67	4	4	0.34	0.76	1.15
		3	4	0.35	4	0.37	0.95	4	4	0.36	0.97	1.03
		4	4	0.33	4	0.35	0.94	4	4	0.34	0.97	1.03
	30		2	0.32	2	0.34	0.94	2	2	0.34	0.94	1.00
		1	2	0.34	2	0.36	0.94	2	2	0.34	1.00	1.06
		2	2	0.32	2	0.34	0.94	2	2	0.36	0.89	0.94
		3	2	0.33	2	0.35	0.94	2	2	0.32	1.03	1.09
		4	2	0.32	2	0.34	0.94	2	2	0.34	0.94	1.00
4	10		8	0.55	11	0.56	0.98	9	9	0.70	0.79	0.80
		1	8	0.63	9	0.61	1.03	8	8	0.72	0.88	0.85
		2	7	0.56	9	0.74	0.76	8	8	0.76	0.74	0.97
		3	6	0.52	8	0.64	0.81	7	7	0.62	0.84	1.03
		4	7	0.62	6	0.53	1.17	7	7	0.66	0.94	0.80
	20		3	0.61	3	0.64	0.95	4	4	0.68	0.90	0.94
		1	3	0.66	3	0.69	0.96	4	4	0.72	0.92	0.96
		2	3	0.64	3	0.70	0.91	4	4	0.70	0.91	1.00
		3	3	0.60	3	0.65	0.92	4	4	0.70	0.86	0.93
		4	3	0.61	3	0.66	0.92	4	4	0.66	0.92	1.00
	30		2	0.82	2	0.86	0.95	2	2	0.64	1.28	1.34
		1	2	0.86	2	0.89	0.97	2	2	0.62	1.39	1.44
		2	2	0.83	2	0.86	0.97	2	2	0.66	1.26	1.30
		3	2	0.82	2	0.86	0.95	2	2	0.64	1.28	1.34
		4	2	0.83	2	0.86	0.97	2	2	0.64	1.30	1.34
8	10		8	1.43	10	1.29	1.11	9	9	1.40	1.02	0.92
		1	6	1.24	10	1.67	0.74	8	8	1.50	0.83	1.11
		2	7	1.60	7	1.55	1.03	7	8	1.50	1.07	1.03
		3	7	1.68	7	1.54	1.09	7	7	1.24	1.35	1.24
		4	7	1.68	7	1.55	1.08	7	7	1.20	1.40	1.29
	20		2	1.20	2	1.20	1.00	4	4	1.36	0.88	0.88
		1	2	1.22	2	1.23	0.99	4	4	1.40	0.87	0.88
		2	2	1.15	2	1.19	0.97	4	4	1.42	0.81	0.84
		3	2	1.13	2	1.16	0.97	4	4	1.32	0.86	0.88
		4	2	1.15	2	1.17	0.98	4	4	1.34	0.86	0.87
	30		2	2.37	2	2.50	0.95	2	2	1.30	1.82	1.92
		1	2	2.57	2	2.60	0.99	2	2	1.34	1.92	1.94
		2	2	2.38	2	2.44	0.98	2	2	1.28	1.86	1.91
		3	2	2.40	2	2.44	0.98	2	2	1.30	1.85	1.88
		4	2	2.36	2	2.42	0.98	2	2	1.28	1.84	1.89

Table 3 Results for problem `NOS3`.

t	m	k	B-GMRES(n) B-LGMRES(n,k)		DB-GMRES(n) DB-LGMRES(n,k)		T_1/T_2	GMRES(m) LGMRES(m)			T_1/T_3	T_2/T_3
			q	$T_1(s)$	q	$T_2(s)$		$\min q$	$\max q$	$T_3(s)$		
2	10		5	0.18	5	0.18	1.00	5	5	0.28	0.64	0.64
		1	5	0.20	5	0.20	1.00	5	5	0.30	0.67	0.67
		2	5	0.22	5	0.22	1.00	5	5	0.28	0.79	0.79
		3	4	0.15	5	0.22	0.68	5	5	0.32	0.47	0.69
		4	5	0.20	5	0.21	0.95	5	5	0.28	0.71	0.75
	20		3	0.27	3	0.29	0.93	3	3	0.32	0.84	0.91
		1	2	0.21	3	0.31	0.68	3	3	0.34	0.62	0.91
		2	3	0.29	3	0.32	0.91	3	3	0.28	1.04	1.14
		3	3	0.28	3	0.29	0.97	3	3	0.32	0.88	0.91
		4	3	0.28	3	0.29	0.97	3	3	0.30	0.93	0.97
	30		2	0.35	2	0.37	0.95	2	2	0.36	0.97	1.03
		1	2	0.38	2	0.38	1.00	2	2	0.34	1.12	1.12
		2	2	0.36	2	0.37	0.97	2	2	0.36	1.00	1.03
		3	2	0.35	2	0.38	0.92	2	2	0.32	1.09	1.19
		4	2	0.35	2	0.37	0.95	2	2	0.36	0.97	1.03
4	10		5	0.37	5	0.40	0.93	5	5	0.54	0.69	0.74
		1	5	0.40	5	0.46	0.87	5	5	0.60	0.67	0.77
		2	5	0.37	5	0.48	0.77	5	5	0.64	0.58	0.75
		3	4	0.33	5	0.49	0.67	5	5	0.62	0.53	0.79
		4	5	0.42	5	0.46	0.91	5	5	0.58	0.72	0.79
	20		3	0.64	3	0.68	0.94	3	3	0.62	1.03	1.10
		1	2	0.45	3	0.73	0.62	3	3	0.64	0.70	1.14
		2	3	0.69	3	0.73	0.95	3	3	0.62	1.11	1.18
		3	3	0.65	3	0.69	0.94	3	3	0.66	0.98	1.05
		4	3	0.65	3	0.68	0.96	3	3	0.64	1.02	1.06
	30		2	0.85	2	0.89	0.96	2	2	0.70	1.21	1.27
		1	2	0.88	2	0.92	0.96	2	2	0.74	1.19	1.24
		2	2	0.86	2	0.88	0.98	2	2	0.70	1.23	1.26
		3	2	0.86	2	0.89	0.97	2	2	0.70	1.23	1.27
		4	2	0.84	2	0.90	0.93	2	2	0.66	1.27	1.36
8	10		5	0.92	5	0.96	0.96	5	5	1.08	0.85	0.89
		1	5	0.88	5	1.06	0.83	5	5	1.16	0.76	0.91
		2	5	0.95	5	1.07	0.89	5	5	1.22	0.78	0.88
		3	4	0.86	5	1.05	0.82	5	5	1.18	0.73	0.89
		4	5	1.05	5	1.06	0.99	5	5	1.12	0.94	0.95
	20		3	1.33	3	1.39	0.96	3	3	1.26	1.06	1.10
		1	2	1.20	3	1.44	0.83	3	3	1.32	0.91	1.09
		2	3	1.34	3	1.41	0.95	3	3	1.26	1.06	1.12
		3	3	1.33	3	1.37	0.97	3	3	1.30	1.02	1.05
		4	3	1.33	3	1.38	0.96	3	3	1.26	1.06	1.10
	30		2	2.36	2	2.48	0.95	2	2	1.34	1.76	1.85
		1	2	2.56	2	2.63	0.97	2	2	1.38	1.86	1.91
		2	2	2.39	2	2.51	0.95	2	2	1.42	1.68	1.77
		3	2	2.39	2	2.52	0.95	2	2	1.34	1.78	1.88
		4	2	2.41	2	2.47	0.98	2	2	1.36	1.77	1.82

Table 4 Results for problem `ORSIRR_1`.

t	m	k	B-GMRES(m) B-LGMRES(m,k)		DB-GMRES(m) DB-LGMRES(m,k)		T_1/T_2	GMRES(m) LGMRES(m)			T_1/T_3	T_2/T_3
			q	$T_1(s)$	q	$T_2(s)$		$\min q$	$\max q$	$T_3(s)$		
2	10		1,031	31.67	1,031	33.45	0.95	1,031	1,031	46.76	0.68	0.72
		1	209	7.48	224	8.59	0.87	205	219	11.18	0.67	0.77
		2	117	4.63	189	8.21	0.56	202	202	12.60	0.37	0.65
		3	132	5.50	199	9.68	0.57	201	202	13.32	0.41	0.73
		4	132	6.27	199	10.73	0.58	205	215	15.06	0.42	0.71
	20		419	35.99	376	34.24	1.05	458	509	54.48	0.66	0.63
		1	75	7.17	78	7.90	0.91	89	90	11.04	0.65	0.72
		2	53	5.16	79	8.56	0.60	89	89	11.78	0.44	0.73
		3	56	6.11	75	8.74	0.70	88	89	12.48	0.49	0.70
		4	54	5.97	75	9.19	0.65	87	87	13.00	0.46	0.71
	30		184	30.97	149	26.68	1.16	124	142	27.22	1.14	0.98
		1	44	8.03	46	8.85	0.91	58	59	12.78	0.63	0.69
		2	36	6.86	44	8.90	0.77	58	58	13.20	0.52	0.67
		3	34	6.40	45	9.45	0.68	57	58	13.84	0.46	0.68
		4	33	6.77	44	9.65	0.70	58	59	14.58	0.46	0.66
4	10		1,031	73.59	1,031	78.20	0.94	1,031	1,031	94.26	0.78	0.83
		1	186	15.58	200	17.41	0.89	201	215	22.30	0.70	0.78
		2	128	11.05	173	17.34	0.64	199	201	23.64	0.47	0.73
		3	127	11.95	167	18.50	0.65	207	210	27.30	0.44	0.68
		4	132	14.91	181	22.38	0.67	200	208	29.38	0.51	0.76
	20		328	72.97	325	72.14	1.01	378	484	99.54	0.73	0.72
		1	64	15.88	66	17.12	0.93	88	93	22.12	0.72	0.77
		2	54	13.39	66	18.39	0.73	87	90	23.30	0.57	0.79
		3	53	14.78	68	20.29	0.73	85	89	24.48	0.60	0.83
		4	54	15.11	66	21.30	0.71	87	89	26.22	0.58	0.81
	30		92	40.60	83	38.01	1.07	120	186	60.60	0.67	0.63
		1	34	16.21	35	17.80	0.91	57	59	25.02	0.65	0.71
		2	31	12.82	36	18.30	0.70	57	58	26.34	0.49	0.69
		3	28	14.94	35	19.77	0.76	57	58	27.62	0.54	0.72
		4	29	15.46	35	20.71	0.75	56	58	28.46	0.54	0.73
8	10		1,031	205.41	1,031	220.69	0.93	1,031	1,031	185.94	1.10	1.19
		1	161	37.73	160	39.54	0.95	205	225	44.86	0.84	0.88
		2	107	27.31	143	39.62	0.69	193	206	46.68	0.59	0.85
		3	102	28.35	136	43.02	0.66	198	205	52.12	0.54	0.83
		4	110	37.70	150	52.30	0.72	195	207	57.24	0.66	0.91
	20		244	132.74	162	100.30	1.32	372	575	203.76	0.65	0.49
		1	54	34.79	56	37.17	0.94	85	92	43.76	0.80	0.85
		2	42	28.34	51	37.00	0.77	87	89	46.58	0.61	0.79
		3	41	30.71	52	40.02	0.77	87	89	49.50	0.62	0.81
		4	41	31.59	51	43.37	0.73	85	90	52.54	0.60	0.83
	30		68	77.77	58	66.25	1.17	110	158	103.40	0.75	0.64
		1	28	37.55	29	39.75	0.94	57	59	50.34	0.75	0.79
		2	24	32.71	28	38.60	0.85	57	59	52.10	0.63	0.74
		3	25	32.18	28	41.66	0.77	55	59	54.38	0.59	0.77
		4	24	33.22	27	42.09	0.79	57	59	57.04	0.58	0.74

Table 5 Results for problem `ORSREG_1`.

t	m	k	B-GMRES(m) B-LGMRES(m,k)		DB-GMRES(m) DB-LGMRES(m,k)		T_1/T_2	GMRES(m) LGMRES(m)			T_1/T_3	T_2/T_3
			q	$T_1(s)$	q	$T_2(s)$		$\min q$	$\max q$	$T_3(s)$		
2	10		98	6.38	98	6.80	0.94	133	147	13.78	0.46	0.49
		1	43	3.28	44	3.53	0.93	43	43	4.94	0.66	0.71
		2	35	2.90	43	3.92	0.74	43	43	5.46	0.53	0.72
		3	35	3.17	43	4.39	0.72	44	44	6.10	0.52	0.72
		4	33	3.19	43	4.76	0.67	43	43	6.52	0.49	0.73
	20		36	6.62	36	7.06	0.94	31	51	9.78	0.68	0.72
		1	21	4.21	22	4.63	0.91	22	22	5.68	0.74	0.82
		2	17	3.54	21	4.82	0.73	21	21	5.88	0.60	0.82
		3	17	3.63	21	5.09	0.71	22	22	6.28	0.58	0.81
		4	16	3.76	22	5.48	0.69	21	22	6.58	0.57	0.83
	30		21	7.50	21	8.03	0.93	18	23	8.82	0.85	0.91
		1	14	5.36	14	5.69	0.94	14	14	6.52	0.82	0.87
		2	12	4.59	14	5.94	0.77	14	14	6.74	0.68	0.88
		3	12	4.11	14	6.15	0.67	14	14	6.92	0.59	0.89
		4	12	5.04	14	6.33	0.80	14	14	7.10	0.71	0.89
4	10		118	18.82	118	19.73	0.95	80	154	24.50	0.77	0.81
		1	42	7.83	43	8.51	0.92	43	45	9.86	0.79	0.86
		2	34	6.98	43	9.65	0.72	42	44	10.84	0.64	0.89
		3	37	7.85	43	10.73	0.73	43	44	12.02	0.65	0.89
		4	33	8.16	43	11.69	0.70	42	44	12.90	0.63	0.91
	20		39	18.53	37	17.85	1.04	30	46	17.40	1.06	1.03
		1	21	11.19	21	11.60	0.96	21	22	11.22	1.00	1.03
		2	21	9.87	21	12.18	0.81	21	22	11.80	0.84	1.03
		3	17	9.86	21	13.16	0.75	21	22	12.44	0.79	1.06
		4	18	10.31	21	13.58	0.76	21	22	12.96	0.80	1.05
	30		22	20.79	22	21.90	0.95	19	22	17.26	1.20	1.27
		1	14	14.34	14	15.03	0.95	14	15	13.08	1.10	1.15
		2	12	12.17	14	15.65	0.78	14	14	13.24	0.92	1.18
		3	12	13.21	14	16.15	0.82	14	15	14.02	0.94	1.15
		4	12	13.11	14	16.71	0.78	14	15	14.32	0.92	1.17
8	10		141	66.86	130	63.49	1.05	77	100	35.84	1.87	1.77
		1	42	22.97	43	24.97	0.92	42	45	19.44	1.18	1.28
		2	33	20.85	43	28.09	0.74	42	44	21.66	0.96	1.30
		3	33	21.38	42	31.62	0.68	43	45	24.00	0.89	1.32
		4	34	24.90	42	35.16	0.71	42	44	25.70	0.97	1.37
	20		41	59.14	40	61.50	0.96	30	42	33.58	1.76	1.83
		1	20	33.16	21	34.99	0.95	21	22	22.30	1.49	1.57
		2	17	28.70	20	37.34	0.77	21	22	23.24	1.23	1.61
		3	18	28.59	20	39.83	0.72	21	22	24.70	1.16	1.61
		4	18	32.70	21	41.80	0.78	21	22	25.62	1.28	1.63
	30		21	66.04	21	67.63	0.98	18	25	36.74	1.80	1.84
		1	14	47.43	14	48.80	0.97	14	15	26.06	1.82	1.87
		2	12	39.53	14	47.19	0.84	14	15	26.86	1.47	1.76
		3	12	41.62	14	49.96	0.83	14	15	27.46	1.52	1.82
		4	12	40.30	13	49.76	0.81	14	15	27.78	1.45	1.79

Table 6 Results for problem PDE900.

t	m	k	B-GMRES(m) B-LGMRES(m,k)		DB-GMRES(m) DB-LGMRES(m,k)		T_1/T_2	GMRES(m) LGMRES(m)			T_1/T_3	T_2/T_3
			q	$T_1(s)$	q	$T_2(s)$		$\min q$	$\max q$	$T_3(s)$		
2	10		18	0.42	18	0.44	0.95	18	18	0.52	0.81	0.85
		1	16	0.44	18	0.52	0.85	16	16	0.52	0.85	1.00
		2	16	0.49	18	0.59	0.83	15	16	0.56	0.87	1.05
		3	17	0.58	19	0.68	0.85	17	17	0.70	0.83	0.97
		4	16	0.52	19	0.72	0.72	17	18	0.74	0.70	0.97
	20		9	0.59	9	0.63	0.94	8	8	0.58	1.02	1.09
		1	10	0.76	12	0.90	0.84	9	9	0.74	1.03	1.22
		2	9	0.70	9	0.76	0.92	9	9	0.76	0.92	1.00
		3	9	0.76	10	0.83	0.92	8	9	0.78	0.97	1.06
		4	9	0.76	9	0.81	0.94	8	10	0.78	0.97	1.04
	30		7	0.96	8	1.04	0.92	6	6	0.84	1.14	1.24
		1	6	0.83	6	0.94	0.88	6	7	0.94	0.88	1.00
		2	6	0.86	7	1.13	0.76	6	7	0.88	0.98	1.28
		3	6	0.92	6	0.98	0.94	6	6	0.90	1.02	1.09
		4	5	0.73	6	0.97	0.75	6	6	0.88	0.83	1.10
4	10		19	1.09	20	1.12	0.97	17	18	1.00	1.09	1.12
		1	16	1.06	17	1.21	0.88	16	17	1.14	0.93	1.06
		2	17	1.21	18	1.44	0.84	17	18	1.28	0.95	1.12
		3	19	1.46	20	1.66	0.88	16	19	1.40	1.04	1.19
		4	16	1.38	17	1.60	0.86	17	18	1.54	0.90	1.04
	20		9	1.52	9	1.63	0.93	8	9	1.24	1.23	1.31
		1	9	1.62	9	1.82	0.89	8	10	1.42	1.14	1.28
		2	9	1.54	9	1.93	0.80	8	9	1.52	1.01	1.27
		3	8	1.68	9	1.89	0.89	8	10	1.58	1.06	1.20
		4	8	1.44	8	1.78	0.81	7	9	1.44	1.00	1.24
	30		6	2.16	6	2.30	0.94	6	8	1.82	1.19	1.26
		1	6	2.00	6	2.40	0.83	6	8	2.06	0.97	1.17
		2	6	2.21	8	2.51	0.88	6	9	2.16	1.02	1.16
		3	5	1.95	5	2.05	0.95	6	7	1.82	1.07	1.13
		4	5	1.91	6	2.31	0.83	6	7	1.82	1.05	1.27
8	10		19	2.95	20	2.88	1.02	17	18	2.08	1.42	1.38
		1	15	2.64	21	3.26	0.81	14	17	2.10	1.26	1.55
		2	15	2.89	15	3.14	0.92	16	18	2.64	1.09	1.19
		3	13	2.89	16	3.49	0.83	17	19	2.86	1.01	1.22
		4	12	2.67	15	3.75	0.71	16	20	3.04	0.88	1.23
	20		9	4.29	12	4.76	0.90	8	9	2.44	1.76	1.95
		1	7	3.82	10	4.24	0.90	8	10	2.80	1.36	1.51
		2	7	3.45	9	4.10	0.84	8	9	3.04	1.13	1.35
		3	6	3.48	6	3.62	0.96	8	10	3.04	1.14	1.19
		4	7	4.12	9	4.26	0.97	7	9	2.96	1.39	1.44
	30		5	5.12	6	5.03	1.02	6	7	3.38	1.51	1.49
		1	5	4.71	7	5.95	0.79	6	7	3.92	1.20	1.52
		2	5	5.43	7	5.68	0.96	6	7	4.00	1.36	1.42
		3	4	4.42	7	5.39	0.82	6	7	3.78	1.17	1.43
		4	5	5.23	7	5.49	0.95	6	7	3.50	1.49	1.57

Table 7 Results for problem `RDB1250`.

t	m	k	B-GMRES(m) B-LGMRES(m,k)		DB-GMRES(m) DB-LGMRES(m,k)		T_1/T_2	GMRES(m) LGMRES(m)			T_1/T_3	T_2/T_3
			q	$T_1(s)$	q	$T_2(s)$		$\min q$	$\max q$	$T_3(s)$		
2	10		140	4.49	113	3.93	1.14	185	200	9.92	0.45	0.40
		1	83	3.39	95	4.18	0.81	88	122	6.26	0.54	0.67
		2	66	3.03	124	5.87	0.52	111	154	8.88	0.34	0.66
		3	70	3.43	107	5.63	0.61	149	175	12.12	0.28	0.46
		4	63	3.47	94	5.74	0.60	96	146	9.88	0.35	0.58
	20		34	3.41	26	2.62	1.30	29	39	4.38	0.78	0.60
		1	17	1.88	17	2.01	0.94	18	18	2.52	0.75	0.80
		2	15	1.75	18	2.20	0.80	16	17	2.52	0.69	0.87
		3	14	1.72	16	2.12	0.81	16	17	2.56	0.67	0.83
		4	14	1.72	15	2.05	0.84	16	17	2.68	0.64	0.76
	30		14	2.76	14	2.96	0.93	16	16	3.68	0.75	0.80
		1	12	2.54	13	2.83	0.90	12	14	3.16	0.80	0.90
		2	10	2.20	13	2.94	0.75	13	13	3.32	0.66	0.89
		3	11	2.38	11	2.67	0.89	12	13	3.20	0.74	0.83
		4	10	2.28	11	2.71	0.84	11	12	3.20	0.71	0.85
4	10		508	34.11	328	21.17	1.61	138	311	20.10	1.70	1.05
		1	105	9.67	143	10.54	0.92	94	167	16.64	0.58	0.63
		2	99	10.51	58	6.87	1.53	130	165	19.82	0.53	0.35
		3	80	9.58	808	75.35	0.13	122	154	21.42	0.45	3.52
		4	70	8.77	338	42.27	0.21	79	129	18.44	0.48	2.29
	20		32	7.98	23	5.59	1.43	32	36	8.68	0.92	0.64
		1	15	3.48	14	4.05	0.86	18	19	5.26	0.66	0.77
		2	12	3.52	14	4.32	0.81	17	18	5.08	0.69	0.85
		3	13	3.90	15	4.88	0.80	17	18	5.46	0.71	0.89
		4	13	3.96	16	5.03	0.79	16	17	5.24	0.76	0.96
	30		14	6.67	15	6.55	1.02	16	18	7.92	0.84	0.83
		1	10	5.43	13	5.82	0.93	13	15	6.74	0.81	0.86
		2	9	4.84	9	5.29	0.91	13	14	7.08	0.68	0.75
		3	9	4.74	11	5.73	0.83	11	13	6.72	0.71	0.85
		4	9	5.24	11	6.62	0.79	11	13	6.36	0.82	1.04
8	10		791	174.77	349	67.91	2.57	129	293	39.74	4.40	1.71
		1	134	32.67	309	58.78	0.56	101	241	36.82	0.89	1.60
		2	97	27.69	104	29.34	0.94	104	154	35.52	0.78	0.83
		3	89	25.87	316	82.05	0.32	93	183	40.42	0.64	2.03
		4	78	29.13	89	32.64	0.89	85	155	39.50	0.74	0.83
	20		28	20.74	20	13.47	1.54	27	35	15.96	1.30	0.84
		1	13	10.54	14	11.09	0.95	18	20	10.54	1.00	1.05
		2	11	9.06	12	10.42	0.87	17	18	10.30	0.88	1.01
		3	12	10.09	18	12.79	0.79	16	18	10.66	0.95	1.20
		4	11	9.21	13	12.25	0.75	16	18	10.56	0.87	1.16
	30		8	12.19	12	12.62	0.97	15	18	15.48	0.79	0.82
		1	7	11.40	8	12.76	0.89	13	15	13.74	0.83	0.93
		2	7	10.81	9	11.87	0.91	12	15	13.76	0.79	0.86
		3	7	11.85	9	12.31	0.96	12	14	13.40	0.88	0.92
		4	7	10.89	11	12.44	0.88	11	13	12.84	0.85	0.97

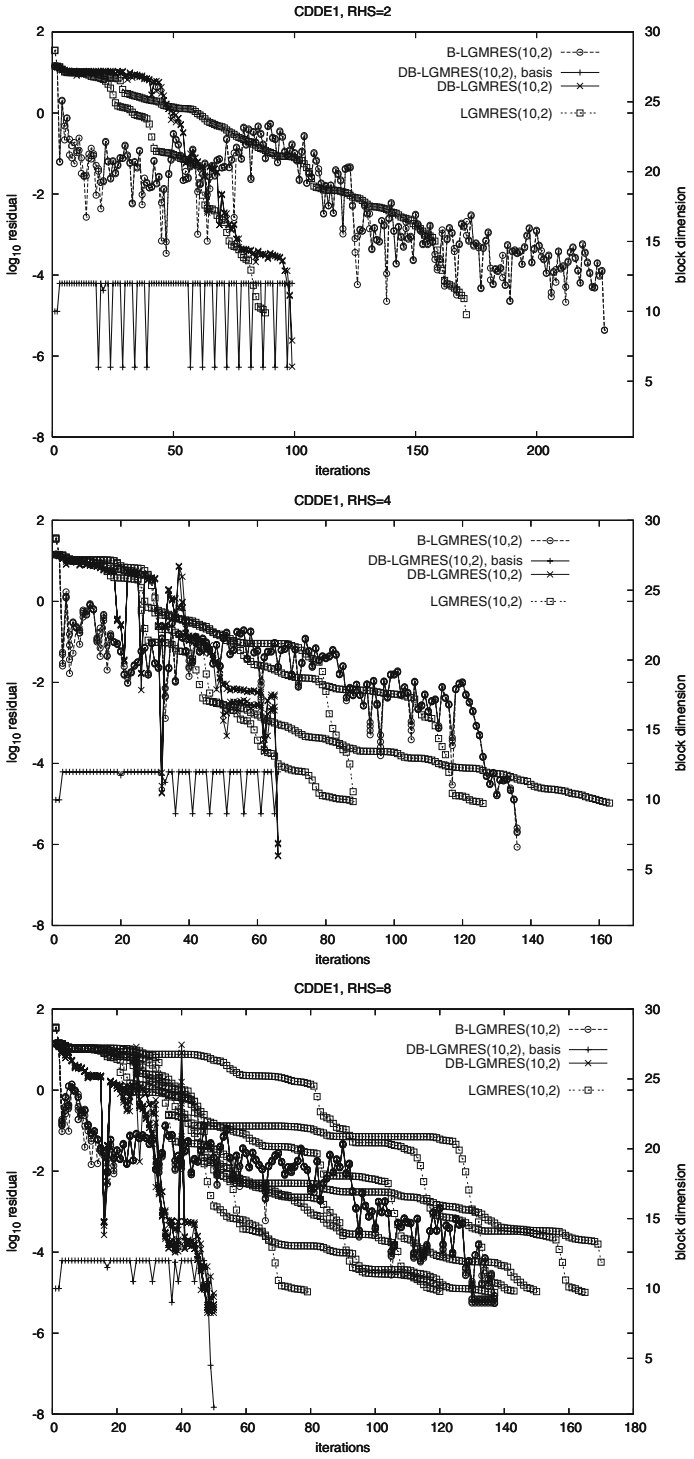

Figure 1 Residual history and evolution of basis dimension for problem `CDDE1`.

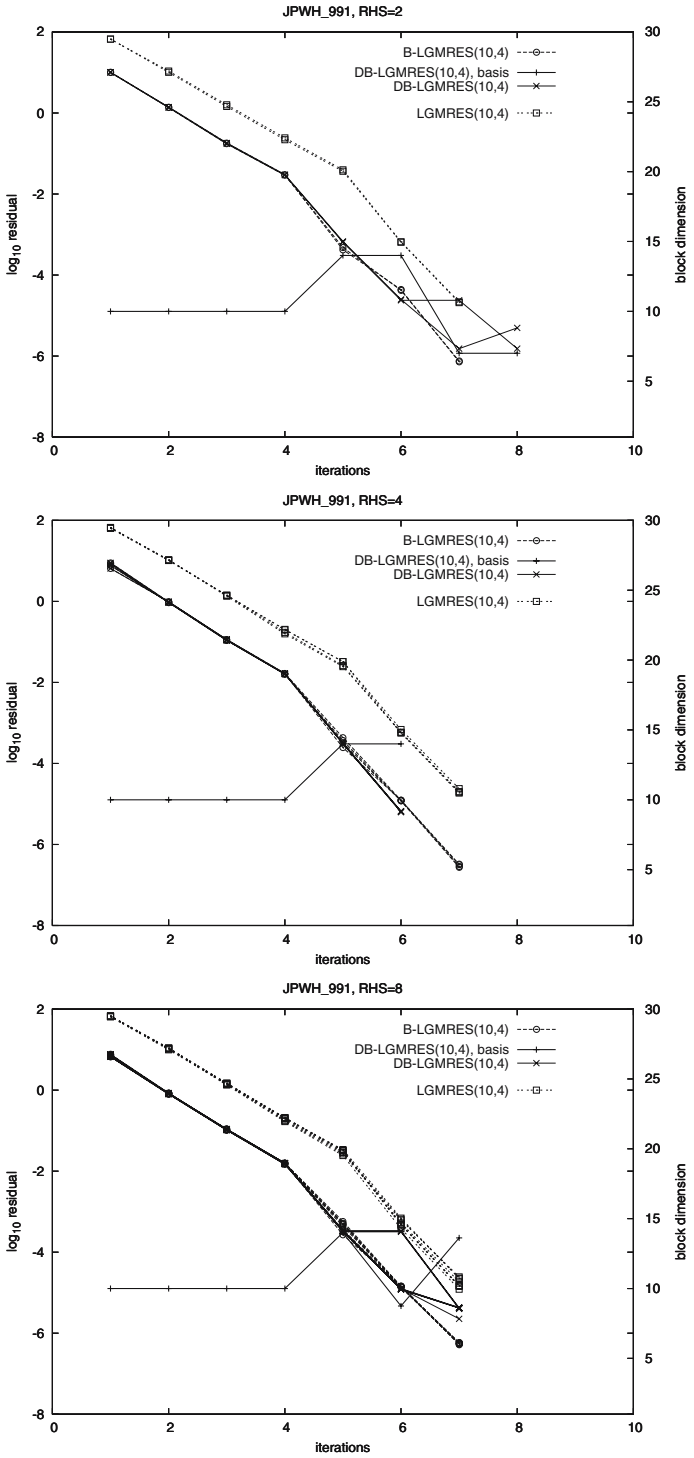

Figure 2 Residual history and evolution of basis dimension for problem `JPWH_991`.

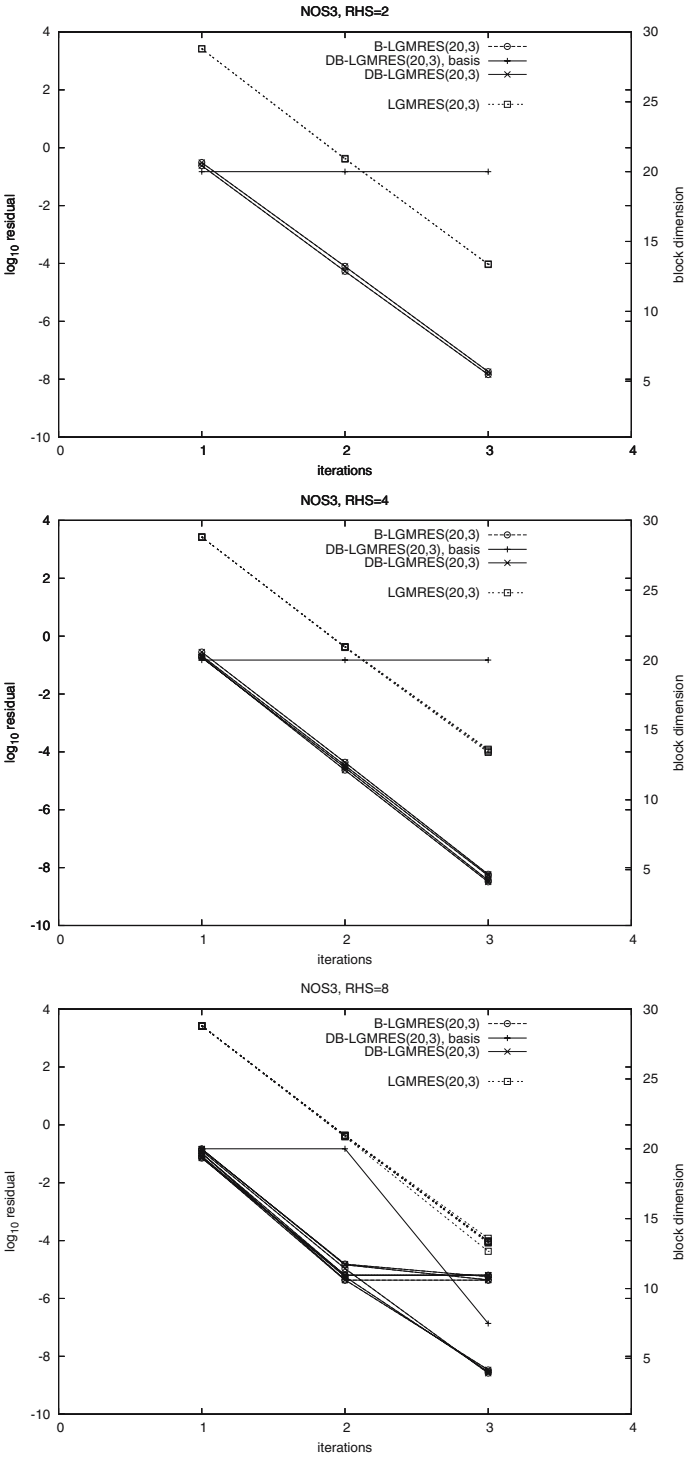

Figure 3 Residual history and evolution of basis dimension for problem `NOS3`.

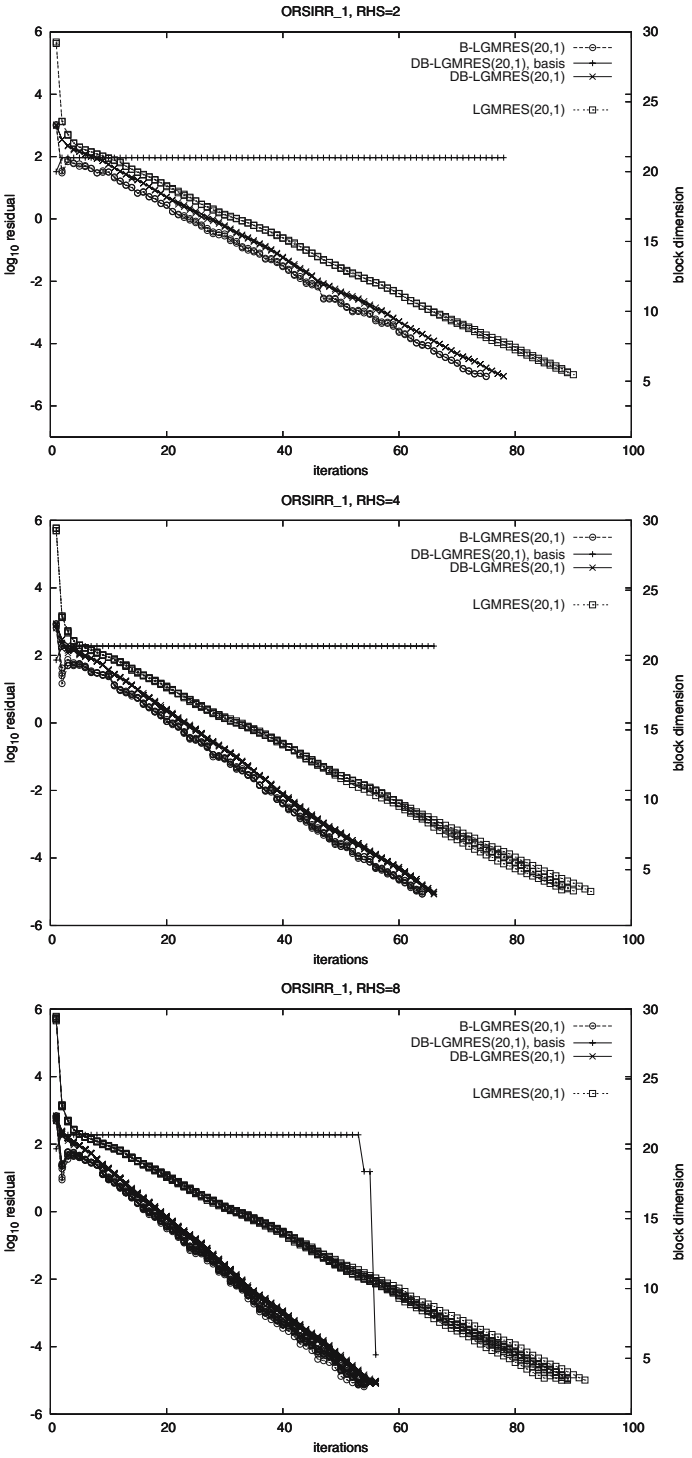

Figure 4 Residual history and evolution of basis dimension for problem `ORSIRR_1`.

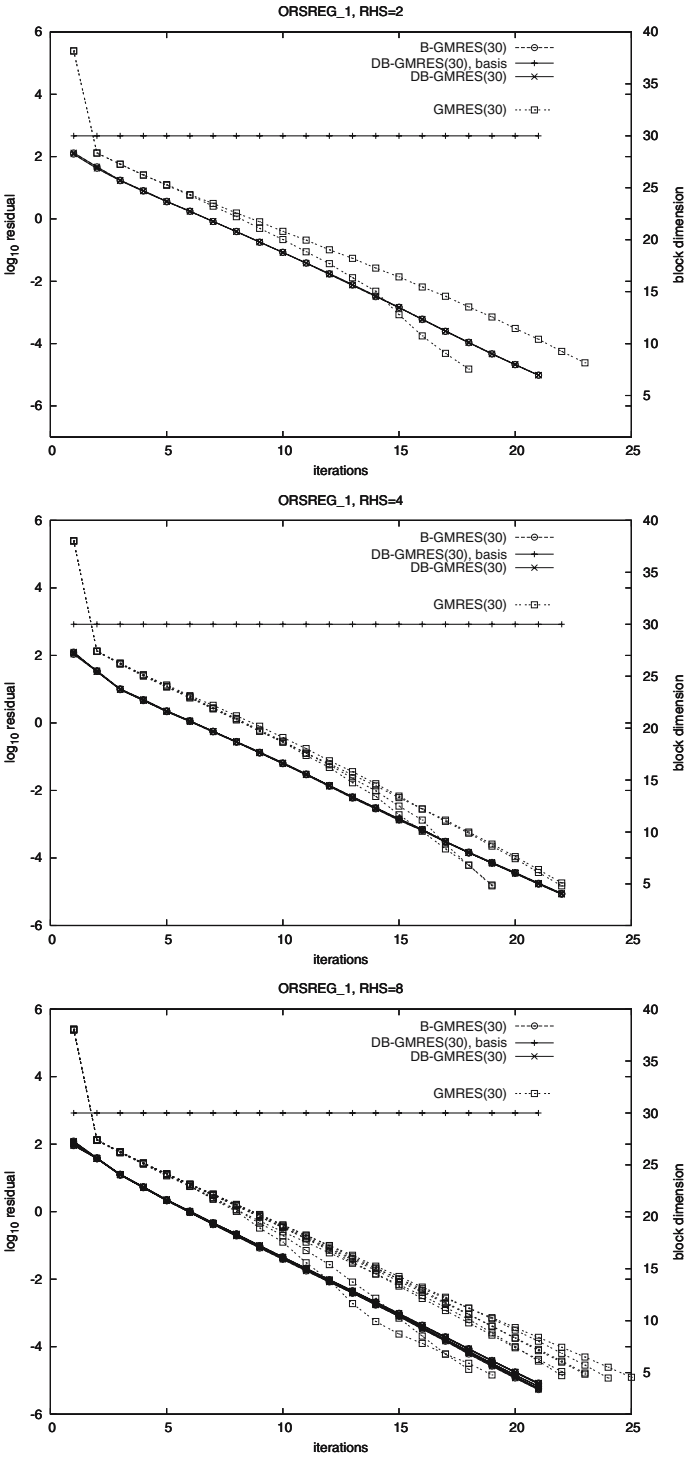

Figure 5 Residual history and evolution of basis dimension for problem `ORSREG_1`.

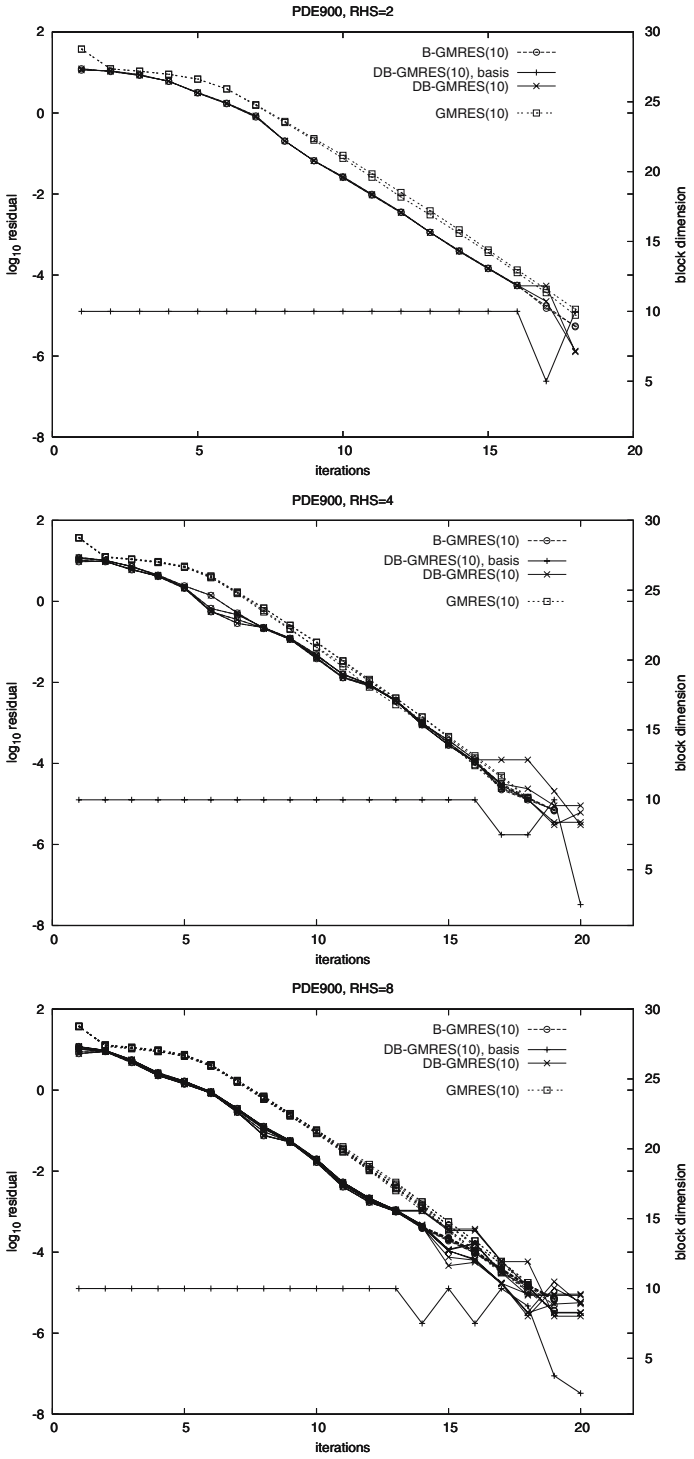

Figure 6 Residual history and evolution of basis dimension for problem PDE900.

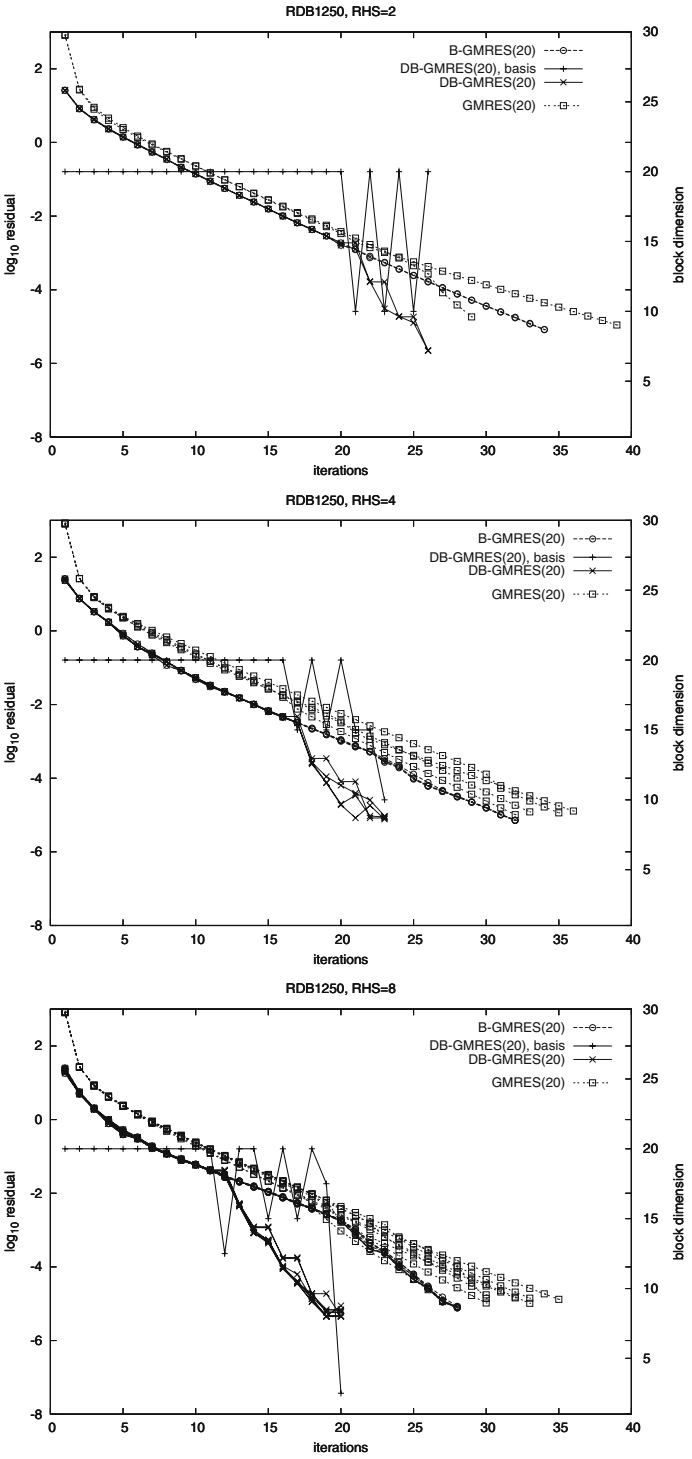

Figure 7 Residual history and evolution of basis dimension for problem `RDB1250`.

5. Parallel implementation details

A parallel implementation of the DB-GMRES(m) algorithm was made in FORTRAN 90 using the strategy employed in the PARALLEL ITERATIVE METHODS (PIM) package by da Cunha and Hopkins [6], suitably extended to cope with a multiple right-hand-sides linear solver. This means that the matrix–matrix product $U = AV$ and the parallel reduction and inner-products are left as user-supplied, external subroutines that are called inside DB-GMRES(m). The implementation is thus fairly general, with the ability to solve systems with any special structure and/or of any sparsity level, including a dense system, on any parallel architecture, including vector, shared-memory and distributed-memory computers.

The PIM package offers a restarted GMRES subroutine for the single right-hand-side case. Our DB-GMRES(m) implementation is based on that subroutine. Most notable changes is the use of Householder reflections to produce the H_k matrix and, of course, the use of the RRQR factorization instead of QR. Also, the use of FORTRAN 90 allows a straightforward management of dynamic memory allocation;[1] this is of extreme importance as we may release portions of memory once convergence for some right-hand-sides is detected.

DB-GMRES(m) is implemented in such a way that it can be used either in a parallel or sequential fashion, depending on the way the user-supplied external subroutines mentioned above are written. The algorithm is based on the *single-program*, *multiple-data* paradigm and we have partitioned the data in the following manner: scalars are present in *all* processors; vectors have their elements partitioned across the processors; and matrices are partitioned by rows. Exceptions to this are those matrices which are stored locally in every processor, the H_k, G and R, and the system coefficient matrix A, which may be partitioned as the user sees fit. Note that this strategy allows the use of a contiguous, cyclic or block-cyclic actual partitioning of the elements and rows of matrices, though DB-GMRES(m) is not aware of that. The DB-GMRES(m) implementation is presented in single- and double-precision for real and complex arithmetic, and uses BLAS and LAPACK routines.

The RRQR factorization uses the PRRQR parallel algorithm (see [5]), which is based on the RRQR algorithm found in [8] (pp. 233–236). The inner QR factorization is performed in parallel also via the PQR algorithm [5]. This uses a row-blocked partition of a matrix $W_{n\times t}$ ($n \gg t$) across p processors and then proceeds in two phases: first, every processor triangularizes its own block of rows of W, without communication between them (a perfectly parallel step); second, the triangular blocks but the topmost one (i.e., R in Equation (6)) are annihilated using a parallel reduction operation (i.e. as if traversing a binary tree upwards to its root processor, the one storing the topmost triangular block). The PQR algorithm has a floating-point operations complexity of order $O(t^2)$ and a communications complexity of order $O(\log_2 p)$.

The PRRQR algorithm calls PQR only once; the resulting matrix R is broadcast to all processors once it is obtained and then, in case a column near-dependency is detected, column swapping and re-triangularization is made by all processors independently (again, a perfectly parallel step).

[1] The PIM package is written in FORTRAN 77.

5.1. Scalability tests

In this section we present experimental results obtained with our parallel implementation of DB-GMRES(m), using as test problem the linear system derived from the finite-differences discretization of the convection–diffusion partial differential equation (taken from [13]):

$$-\epsilon(u_{xx} + u_{yy}) + (\cos\alpha)u_x + (\sin\alpha)u_y = 0 \tag{9}$$

It is discretized on an unit square, where α was taken as $-\pi/6$ and $u = x^2 + y^2$ on the boundary. This equation was discretized on square meshes with $l = 200$ and $l = 400$, leading to systems of $n = 40000$ and $n = 160000$ linear equations, respectively. The number of rhs taken in these tests was $t = 4$. The rhs vectors were taken such that the solution is $x_{(i)} = (1, 1, \ldots, 1)^T$. The initial guesses were the zero vectors. These systems were solved using a Neumann polynomial preconditioner of first degree, up to a tolerance of 10^{-6}. The sequential and parallel codes employed single-precision floating-point computations throughout, and were tested on a SunFire 15K parallel computer, equipped with UltraSPARC III 1.2 GHz processors, located at the Cambridge-Cranfield High Performance Computing Facility. The Forte Developer 7 Fortran 95 compiler and the LAM implementation of MPI were used.

The parallel implementations of B-GMRES(m) and DB-GMRES(m), with respect to this particular system, utilized a geometric partition of the discretized domain, similar to that used in [6], extended to deal with several rhs. The vectors are mapped to the same domain and, as now there are t rhs vectors, we view each one of these vectors as a domain, one placed above the other. The matrix–vector products required during the iterations are then computed using the nearest-neighbour approach used in [6]; however, to reduce the number of messages exchanged between the processors, we send a single message containing t parts (of size l) of any vector involved in the data exchange.

Table 8 shows the number of iterations and run time (in s) obtained with our implementations of B-GMRES(m) and DB-GMRES(m). We note that for these systems, there will not be deflation caused by early convergence of one rhs (as they are all the same). However, there will be deflation in the Krylov vector basis, because the residual vectors will be the same (in a numerical sense); at most $(t-1)m$ Krylov

Table 8 Number of iterations required for convergence and run time (in seconds) for B-GMRES(m) and DB-GMRES(m), and normalized gain (run time/number of iterations) of DB-GMRES(m) over B-GMRES(m), measured on 8 processors.

n	m	B-GMRES(m)		DB-GMRES(m)		Normalized gain
		q	$T(s)$	q	$T(s)$	
40,000	4	1,730	121.93	67	2.57	1.84
	8	420	73.23	183	12.11	2.63
160,000	4	2,000[a]	721.04	66	13.87	1.72
	8	1,436	1, 155.91	69	19.89	2.79

[a]Diverged.

vectors will be dropped by the (P)RRQR algorithm (due to round-off errors). We note that at least one column per block will remain. Therefore, DB-GMRES(m) should be much better than B-GMRES(m). Of course one should stress that this experiment shows an extreme situation. We believe that these reductions would also occur in a situation where a subset of the rhs vectors are the same, say l; then $(l-1)m$ Krylov vectors corresponding to these would be expressed by at most m Krylov vectors and therefore a substantial reduction in the run time would occur. We also note that B-GMRES(m) takes considerably more iterations than DB-GMRES(m).

Table 9 shows the run time (in s) and the speed-ups obtained by our sequential and parallel implementations of B-GMRES(m) and DB-GMRES(m). To analyse the scalability, we have restricted the number of iterations to 3 and compared the performance of these two methods. At the same time, we have also limited the number of iterations of two rhs to 2 and 1, respectively, only for DB-GMRES(m); the idea being that of considering early convergence for some rhs at different iterations. The results presented show that the parallelization of the methods, made in terms of the vector operations led to a good overall performance. One can also notice that the speed-ups provided by DB-GMRES(m) are smaller than that of B-GMRES(m), considering the same number of iterations; in this case, the number of vector operations performed by the former is higher, caused by the use of the RRQR

Table 9 Parallel performance of B-GMRES(m) and DB-GMRES(m).

$m=4$		Run time (s)			Speed-up		
n	p	A	B	C	A	B	C
40,000	1	1.67	0.82	0.77	–	–	–
	2	0.7	0.41	0.33	2.39	2.00	2.33
	4	0.43	0.24	0.17	3.88	3.42	4.53
	8	0.22	0.13	0.11	7.59	6.31	7.00
	16	0.24	0.19	0.18	6.96	4.32	4.28
160,000	1	10.10	5.18	3.76	–	–	–
	2	4.35	2.18	1.65	2.32	2.38	2.28
	4	2.02	1.03	0.75	5.00	5.03	5.01
	8	1.27	0.68	0.51	7.95	7.62	7.37
	16	0.84	0.38	0.33	12.02	13.63	11.39
$m=8$		Run time (s)			Speed-up		
n	p	A	B	C	A	B	C
40,000	1	3.55	1.14	1.21	–	–	–
	2	1.78	0.63	0.50	1.99	1.81	2.42
	4	0.95	0.31	0.25	3.74	3.68	4.84
	8	0.53	0.19	0.17	6.70	6.00	7.12
	16	0.44	0.32	0.33	8.07	3.56	3.67
160,000	1	23.43	8.60	6.58	–	–	–
	2	9.37	3.72	2.75	2.50	2.31	2.39
	4	5.29	1.43	1.22	4.43	6.01	5.39
	8	2.67	1.16	0.84	8.78	7.41	7.83
	16	1.51	0.60	0.65	15.52	14.33	10.12

A: B-GMRES(m), $q=3$ iterations for all rhs,
B: DB-GMRES(m), $q=3$ iterations for all rhs,
C: DB-GMRES(m), $q=3,3,2,1$ iterations for rhs vectors 1, 2, 3 and 4, respectively.

factorization. Another interesting result is that, in the presence of deflation caused by early convergence of some rhs, the speed-up of DB-GMRES(m) increases; again this can be explained by the smaller amount of floating-point operations to be carried out once convergence for some rhs is achieved.

6. Final remarks

We have presented two new block iterative methods for systems of linear equations with multiple right-hand-sides, based on the Block-GMRES(m) and Block-LGMRES(m,k) methods. The main advantage of the proposed methods is the capability of early detection of convergence for some right-hand-sides and of linear dependency between the Krylov vectors by means of the RRQR factorization which has replaced the QR factorization in the block-Arnoldi method. This in turn leads in some cases to a reduction on the overall computational work, though the experimental results have shown that in some cases the use of the RRQR factorization is not adequate. We believe that this may be remedied with the detection of an increase in the residual norms and/or a smaller rate of convergence which would then indicate that either the RRQR factorization should not be employed or an increase of the basis dimension is to be taken for the next iterations. The latter option has been shown to provide good results for unblocked GMRES variants (see [9]) and we intend to apply this technique to our DB-GMRES(m) and DB-LGMRES(m,k) algorithms.

Acknowledgments We would like to thank Dr. James C. Patterson for suggesting the use of a rank-revealing QR factorization inside the GMRES iteration, and for the useful discussions about the subject.

The authors would also like to thank the Cambridge-Cranfield High Performance Computing Facility for allowing our use of its computing facilities. We also thank the referee for his very useful comments which have helped improve this paper.

References

[1] Baker, A., Dennis, J., Jessup, E.: An efficient block variant of GMRES. Technical Report CU-CS-957-03. Department of Computer Science, College of Engineering and Applied Science, University of Colorado at Boulder, (2003a)
[2] Baker, A., Jessup, E., Manteuffel, T.: A technique for accelerating the convergence of restarted GMRES. Technical Report CU-CS-945-03. Department of Computer Science, College of Engineering and Applied Science, University of Colorado at Boulder, (2003b)
[3] Bischof, C., Quintana-Ortí, G.: Computing rank-revealing QR factorizations of dense matrices. ACM Trans. Math. Software **24**(2), 226–253 (1998)
[4] Chan, T.: Rank revealing QR factorizations. Linear Algebra Appl. **88/89**, 67–82 (1987)
[5] da Cunha, R., Becker, D., Patterson, J.: New parallel (rank-revealing) QR factorization algorithms. In: Monien, B., Feldmann, R. (eds.) Euro-Par 2002, pp. 677–686. Springer-Verlag, Berlin (2002)
[6] da Cunha, R., Hopkins, T.: The parallel iterative methods (PIM) package for the solution of systems of linear equations on parallel computers. Appl. Numer. Math. **19**(1–2), 33–50 (1995)
[7] Foster, L.: Rank and null space calculations using matrix decomposition without column interchanges. Linear Algebra Appl. **74**, 47–71 (1986)
[8] Golub, G., Van Loan, C.: Matrix Computations, 2nd edn. Johns Hopkins University Press, Baltimore (1989)
[9] Gonçalez, T.: Adaptive algorithms for the GMRES(m) method. MS dissertation (in Portuguese), Programa de Pós-Graduação em Matemática Aplicada, UFRGS. Supervisor: R.D. da Cunha, (2005)

[10] Li, G.: A block variant of the GMRES method on massively parallel processors. Parallel Comput. **23**(8), 1005–1019 (1997)
[11] Mathematical and Computational Sciences Division, Information Technology Laboratory, National Institute of Standards and Technology. Matrix market: A visual repository of test data for use in comparative studies of algorithms for numerical linear algebra. http://math.nist.gov/MatrixMarket/, 2003
[12] Saad, Y.: Iterative Methods for Sparse Linear Systems, 2nd edn. SIAM, Philadelphia (2003)
[13] Sonneveld, P.: CGS, a fast Lanczos-type solver for nonsymmetric linear systems. SIAM J. Sci. Statist. Comput. **10**, 36–52 (1989)
[14] Vital, B.: Etude de quelques méthodes de résolution de problémes linéaires de grande taille sur multiprocesseur. Ph.D. thesis, Université de Rennes, (1990)
[15] Walker, H.: Implementation of the GMRES method using householder transformations. SIAM J. Sci. Statist. Comput. **9**(1), 152–163 (1989)