A Parallel Algorithm for Systems of Convection-Diffusion Equations

János Karátson[1], Tamás Kurics[1], and Ivan Lirkov[2]

[1] Department of Applied Analysis and Computational Mathematics, ELTE University, H-1117 Budapest, Hungary
karatson@cs.elte.hu, fantom@cs.elte.hu
[2] Institute for Parallel Processing, Bulgarian Academy of Sciences, Acad. G. Bonchev, Bl. 25A, 1113 Sofia, Bulgaria
ivan@parallel.bas.bg

Abstract. The numerical solution of systems of convection-diffusion equations is considered. The problem is described by a system of second order partial differential equations (PDEs). This system is discretized by Courant-elements. The preconditioned conjugate gradient method is used for the iterative solution of the large-scale linear algebraic systems arising after the finite element discretization of the problem. Discrete Helmholtz preconditioners are applied to obtain a mesh independent superlinear convergence of the iterative method. A parallel algorithm is derived for the proposed preconditioner. A portable parallel code using Message Passing Interface (MPI) is developed. Numerical tests well illustrate the performance of the proposed method on a parallel computer architecture.

2000 Mathematics Subject Classification: 65N12, 68W10, 65F10, 74S05.

1 Introduction

The generalized conjugate gradient (GCG) method has become the most widespread way of solving nonsymmetric linear algebraic systems arising from discretized elliptic problems, see [3] where an extensive summary is given on the convergence of the CGM. For discretized elliptic problems, the CGM is mostly used with suitable preconditioning (cf. [3]), which sometimes relies on Hilbert space theory (cf. [6]) and then provides mesh independent convergence. Moreover, it has been shown in [6] that the GCG method can be competitive with multigrid methods.

The CGM for nonsymmetric equations in Hilbert space has been studied in [4,5]: in the latter superlinear convergence has been proved in Hilbert space and, based on this, mesh independence of the superlinear estimate has been derived for FEM discretizations of elliptic Dirichlet problems. The mesh independent superlinear convergence results have been extended from a single equation to systems of PDEs in a recent paper [7] in the framework of normal operators in Hilbert space. An important advantage of the obtained preconditioning method

T. Boyanov et al. (Eds.): NMA 2006, LNCS 4310, pp. 65–73, 2007.

for systems is that one can define decoupled preconditioners, hence the size of the auxiliary systems remains as small as for a single equation, moreover, parallelization of the auxiliary systems is available. The main goal of this paper is to develop an efficient MPI parallel code using multiple processors, based on a proper summary of the theoretical result for systems of PDEs.

We consider systems of the form

$$\left.\begin{aligned} -\operatorname{div}(K_i\nabla u_i) + \mathbf{b}_i\cdot\nabla u_i + \sum_{j=1}^{l} V_{ij}u_j = g_i \\ u_{i_{|\partial\Omega}} = 0 \end{aligned}\right\} \qquad (i=1,\dots,l) \tag{1}$$

under the following

Assumptions BVP.

(i) the bounded domain $\Omega\subset\mathbb{R}^N$ is C^2-diffeomorphic to a convex domain;
(ii) for all $i,j=1,\dots,l$, $K_i\in C^1(\overline{\Omega})$, $V_{ij}\in L^\infty(\Omega)$ and $\mathbf{b}_i\in C^1(\overline{\Omega})^N$;
(iii) there is $m>0$ such that $K_i\geq m$ holds for all $i=1,\dots,l$;
(iv) letting $V=\{V_{ij}\}_{i,j=1}^{l}$, the coercivity property

$$\lambda_{\min}\left(V+V^T\right) - \max_i \operatorname{div}\mathbf{b}_i \geq 0 \tag{2}$$

holds pointwise on Ω, where $\lambda_{\min}$ denotes the smallest eigenvalue;
(v) $g_i\in L^2(\Omega)$.

Items (iii) and (iv) ensure the coercivity property (6) which is a crucial assumption for Theorem 1.

Systems of the form (1) arise, e.g., from the time discretization and Newton linearization of nonlinear reaction-convection-diffusion systems which occur frequently in meteorological air-pollution models [12].

We write the considered system in a short vector form using the corresponding n-tuples:

$$\left.\begin{aligned} L\mathbf{u} \equiv -\operatorname{div}(\mathbf{K}\nabla\mathbf{u}) + \mathbf{b}\cdot\nabla\mathbf{u} + V\mathbf{u} = \mathbf{g} \\ \mathbf{u}_{|\partial\Omega} = \mathbf{0} \end{aligned}\right\}, \tag{3}$$

where

$$\mathbf{u}=\begin{pmatrix} u_1 \\ \vdots \\ u_l \end{pmatrix},\ \mathbf{g}=\begin{pmatrix} g_1 \\ \vdots \\ g_l \end{pmatrix},\ \operatorname{div}(\mathbf{K}\nabla\mathbf{u}) = \begin{pmatrix} \operatorname{div}(K_1\nabla u_1) \\ \vdots \\ \operatorname{div}(K_l\nabla u_l) \end{pmatrix},\ \mathbf{b}\cdot\nabla\mathbf{u} = \begin{pmatrix} \mathbf{b}_1\cdot\nabla u_1 \\ \vdots \\ \mathbf{b}_l\cdot\nabla u_l \end{pmatrix}.$$

The FEM discretization of (3) leads to a linear algebraic system $\mathbf{L}_h\mathbf{c}=\mathbf{g}_h$. This can be solved by the GCG method using a preconditioner. In this paper we consider decoupled symmetric Helmholtz preconditioners

$$S_iu_i := -\operatorname{div}(K_i\nabla u_i) + \eta_i u_i \qquad (i=1,\dots,l) \tag{4}$$

where $\eta_i\in C(\overline{\Omega})$, $\eta_i\geq 0$ are suitable functions. The n-tuple S of the elliptic operators S_i and the corresponding matrix $\mathbf{S}_h$ can be defined in the same way as previously, hence the preconditioned form of the discretized equation is

$$\mathbf{S}_h^{-1}\mathbf{L}_h\mathbf{c} = \mathbf{f}_h \equiv \mathbf{S}_h^{-1}\mathbf{g}_h. \tag{5}$$

2 The Preconditioned Generalized Conjugate Gradient Method

Now let us consider the operator equation $Lu = g$ with an unbounded linear operator $L : D \to H$ defined on a dense domain D, and with some $g \in H$, where H is an infinite dimensional complex separable Hilbert space. We have the following

Assumptions A.

(i) The operator L is decomposed in $L = S + Q$ on its domain D where S is a self-adjoint operator in H.
(ii) S is a strongly positive operator, i.e., there exists $p > 0$ such that

$$\langle Su, u\rangle \geq p\|u\|^2 \qquad (u \in D). \tag{6}$$

(iii) There exists $\varrho > 0$ such that $\Re\langle Lu, u\rangle \geq \varrho\langle Su, u\rangle$ $(u \in D)$.
(iv) The operator Q can be extended to the energy space H_S, and then $S^{-1}Q$ is assumed to be a compact normal operator on H_S.

The generalized conjugate gradient, least square (GCG-LS) method is defined in [2]. The full version of the GCG-LS method constructs a sequence of search directions d_k and simultaneously a sequence of approximate solutions u_k. Following the terminology of [2,4], the definition also involves an integer $s \in \mathbb{N}$, further, we let $s_k = \min\{k, s\}$ $(k \geq 0)$. The full version of the algorithm for the solution of the preconditioned operator equation

$$S^{-1}Lu = f \equiv S^{-1}g \tag{7}$$

in H_S is as follows:

$$\begin{cases}
(1)\ \text{Let } u_0 \in D \text{ be arbitrary, let } r_0 \text{ be the solution of } Sr_0 = Lu_0 - g; \\
\quad d_0 = -r_0; \text{ and } z_0 \text{ be the solution of } Sz_0 = Ld_0; \\
\qquad \text{for any } k \in \mathbb{N}: \text{ when } u_k, d_k, r_k, z_k \text{ are obtained, let} \\
(2a)\ \text{the numbers } \alpha_{k-j}^{(k)} \quad (j = 0, \ldots, k) \quad \text{be the solution of} \\
\quad \sum_{j=0}^{k} \alpha_{k-j}^{(k)} \langle Sz_{k-j}, z_{k-l}\rangle = -\langle r_k, Sz_{k-l}\rangle \qquad (0 \leq l \leq k); \\
(2b)\ u_{k+1} = u_k + \sum_{j=0}^{k} \alpha_{k-j}^{(k)} d_{k-j}; \\
(2c)\ r_{k+1} = r_k + \sum_{j=0}^{k} \alpha_{k-j}^{(k)} z_{k-j}; \\
(2d)\ \beta_{k-j}^{(k)} = \langle Lr_{k+1}, z_{k-j}\rangle / \|z_{k-j}\|_S^2 \quad (j = 0, \ldots, s_k); \\
(2e)\ d_{k+1} = -r_{k+1} + \sum_{j=0}^{s_k} \beta_{k-j}^{(k)} d_{k-j}; \\
(2f)\ z_{k+1} \text{ be the solution of } Sz_{k+1} = Ld_{k+1}.
\end{cases} \tag{8}$$

When symmetric part preconditioning is used, a more simple truncated algorithm is applicable, namely the so-called GCG-LS(0) (see [4] for details), where

only the previous search direction d_k and the auxiliary vector z_k are used, so the previous ones do not have to be stored. Assumptions A imply that the operator of the preconditioned equation $S^{-1}L$ has the form $I + S^{-1}Q$, which is a compact perturbation of the identity operator, hence the following convergence result (cf. [5,7]) is applicable. Recall that a compact operator has countably many eigenvalues (with multiplicity), clustering at zero.

Theorem 1. *Let Assumptions A hold. Denoting the unique solution by u^*, the generalized conjugate gradient method applied for equation* (7) *yields for all $k \in \mathbb{N}$*

$$Q_k := \left(\frac{\|e_k\|_L}{\|e_0\|_L}\right)^{1/k} \le \frac{2}{\varrho}\left(\frac{1}{k}\sum_{i=1}^{k}\left|\lambda_i(S^{-1}Q)\right|\right) \to 0 \quad \text{as } k \to \infty \tag{9}$$

where $e_k = u_k - u^*$ is the error vector and $\lambda_i = \lambda_i(S^{-1}Q)$ ($i \in \mathbb{N}$) are the ordered eigenvalues of the operator $S^{-1}Q$ ($|\lambda_i| \ge |\lambda_{i+1}|$).

3 Superlinear Convergence for Elliptic Systems

Let us consider the Hilbert space $H = L^2(\Omega)^l$ with the inner product $\langle \mathbf{u}, \mathbf{v}\rangle = \int_\Omega \sum_{i=1}^{l} u_i \overline{v}_i$ and define the operators L and S according to (3) and (4) on the dense domain

$$D(L) = D(S) = D := \left(H^2(\Omega) \cap H_0^1(\Omega)\right)^l.$$

Now we can use the convergence theorem for this problem in the space $L^2(\Omega)^l$ by verifying that L and S satisfy Assumptions A. First, we apply Theorem 1 using the truncated algorithm when S is the symmetric part of L. Then we consider the full version (8) and use Theorem 1 for problems with constant coefficients when the normality of the preconditioned operator in the corresponding Sobolev space can be ensured.

First symmetric part preconditioning is considered, that is $S = (L + L^*)/2$. Since $Q = L - S$ is antisymmetric, it can be shown easily, that the operator $S^{-1}Q$ is antisymmetric in H_S, therefore it is normal automatically. We have for $\mathbf{u}, \mathbf{v} \in D$

$$\langle L\mathbf{u}, \mathbf{v}\rangle = \int_\Omega \left(\sum_{i=1}^{l}\left(K_i \nabla u_i \cdot \overline{\nabla} v_i + (\mathbf{b}_i \cdot \nabla u_i)\overline{v}_i\right) + \sum_{i,j=1}^{l} V_{ij} u_j \overline{v}_i\right). \tag{10}$$

The divergence theorem and the boundary conditions imply (see [4]) that

$$\langle S\mathbf{u}, \mathbf{v}\rangle = \int_\Omega \left(\sum_{i=1}^{l}\left(K_i \nabla u_i \cdot \overline{\nabla} v_i - \frac{1}{2}(\operatorname{div} \mathbf{b}_i) u_i \overline{v}_i\right) + \frac{1}{2}\sum_{i,j=1}^{l}(V_{ij} + V_{ji}) u_j \overline{v}_i\right).$$

The operator S itself falls into the type (4) if and only if

$$V_{ij} = -V_{ji} \quad (i \ne j) \quad \text{and} \quad \eta_i = V_{ii} - \frac{1}{2}(\operatorname{div} \mathbf{b}_i). \tag{11}$$

Proposition 1. (cf. [7]). *Under Assumptions BVP and condition* (11)*, Assumptions A are satisfied and therefore the truncated GCG-LS algorithm for system* (1) *converges superlinearly in the space* $H_0^1(\Omega)^l$ *according to the estimate* (9) *with the parameter* $\varrho = 1$.

Using the truncated algorithm can be beneficial, but it is a significant restriction not to have the freedom to choose the coefficients η_i of S in (4). For convection-dominated problems, large values of η_i might compensate the large $\mathbf{b}$ [8]. Now let us consider the preconditioner operator (4) with arbitrary nonnegative parameters η_i.

Proposition 2. (cf. [7]). *Assume that* $K_i \equiv K \in \mathbb{R}$, $\eta_i \equiv \eta \in \mathbb{R}$ *and* $\mathbf{b}_i \equiv \mathbf{b} \in \mathbb{R}^N$ *are constants,* $V \in \mathbb{R}^{l\times l}$ *is a normal matrix and suppose that Assumptions BVP hold. Then the full version of the preconditioned GCG-LS algorithm* (8) *for system* (1) *with the preconditioning operator* (4) *converges superlinearly in the space* $H_0^1(\Omega)^l$ *according to the estimate* (9).

Now let us consider the discretized problem (5). Then as shown in [7], the GCG method can be defined similarly as in (8), simply replacing L and S by $\mathbf{L}_h$ and $\mathbf{S}_h$, in particular, in step (2f) z_{k+1} is defined as the FEM solution of the problem $Sz_{k+1} = Ld_{k+1}$ in the considered subspace V_h. Then the right-hand side of (9) provides a mesh independent superlinear convergence estimate for the discretized problem. Besides the superlinear convergence result, the advantage of the preconditioning method (4) is that the elliptic operators are decoupled, i. e. the corresponding matrix $\mathbf{S}_h$ is symmetric block-diagonal, hence auxiliary equations for the discretized system like $\mathbf{S}_h\mathbf{z}_h = \mathbf{L}_h\mathbf{d}_h$ (step (2f) in algorithm (8)) can be divided into l parts and they can be solved simultaneously.

4 Parallelization of the GCG-LS Algorithm

The basic advantage of the proposed preconditioner is its inherent parallelism. The kth iteration of the full version of GCG-LS algorithm consists of two matrix-vector multiplications with matrix $\mathbf{L}_h$, one preconditioning step (solving a system of equations with the preconditioner), solving a system of k equations, $2k+s+2$ inner products, and $s+2$ linked triads (a vector updated by a vector multiplied by a scalar).

Let us consider a parallel system with p processors. We divide the vectors u_k, d_k, r_k, z_k (defined in (8)) in such a way that first $\left\lceil \frac{l}{p} \right\rceil$ blocks are stored in the first processor, blocks for $i = \left\lceil \frac{l}{p} \right\rceil + 1, \ldots, 2\left\lceil \frac{l}{p} \right\rceil$ in the second processor and so on. Then the preconditioning step and linked triads do not need any communication between processors. The computation of inner products requires one global communication to accumulate the local inner products computed on each processor. Communication time for computing inner products increases with the number of processors but in general it is small. The matrix-vector

multiplication requires exchanging of data between all processors. Communication time for matrix-vector multiplication depends on the size of the matrix and on the number of processors.

5 Numerical Experiments

In this section we report the results of the experiments executed on a Linux cluster consisting of 4 dual processor PowerPCs with G4 450 MHz processors, 512 MB memory per node. The developed parallel code has been implemented in C and the parallelization has been facilitated using the MPI library [10,11]. We use the LAPACK library [1] for computing the Cholesky factorization of the preconditioner and for solving the linear systems arising in GCG-LS. The optimization options of the compiler have been tuned to achieve the best performance. Times have been collected using the MPI provided timer. In this paper we report the best results from multiple runs.

The first test problem is a class of systems (1) with $l = 2, 3, \ldots, 10$ equations, where $\mathbf{b}_i = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$ and the matrix V is skew-symmetric with elements which are randomly generated constants. Our second test problem comes from the time discretization and Newton linearization of a nonlinear reaction-convection-diffusion system of 10 equations, used in meteorological air-pollution models [12]. Since the run times here have proved to be very similar to the case of a random 10×10 matrix in the first test problem, we will only present the test results for the first problem.

In what follows, we analyze the obtained parallel time T_p on p processors, relative parallel speed-up $S_p = \frac{T_1}{T_p} \leq p$ and relative efficiency $E_p = \frac{S_p}{p} \leq 1$.

In our experiments we used a stopping criterion $\|r_k\| \leq 10^{-14}$. Table 1 shows the required number of iterations. The obtained parallel time T_p on p processors is presented in Tables 2 and 3. Here l denotes the number of equations. The first column consists of the number of processors. The execution time for problems with $h^{-1} = 32, 64, 128, 192, 256$ in seconds is shown in the next columns. The execution times of the full and truncated version of the algorithm are similar. Because of that we put in Table 3 execution times only for systems of 8 and 10 equations. One can see that for relatively small problems, the execution time on

Table 1. Number of iterations

1/h	l									
	1	2	3	4	5	6	7	8	9	10
8	9	10	11	12	12	12	13	13	14	14
16	9	10	12	12	13	13	13	14	14	14
32	9	10	12	12	13	13	14	14	14	14
64	9	10	12	12	13	13	14	14	14	14
128	9	10	12	12	13	13	14	14	14	14

Table 2. Execution time for full version of GCG-LS

p	h^{-1}			
	32	64	128	256
$l = 2$				
1	0.13	1.06	11.30	130.06
2	0.46	0.99	6.50	69.31
$l = 3$				
1	0.22	1.91	19.05	207.86
2	0.55	1.47	13.24	143.40
3	0.60	1.39	8.41	79.30
$l = 4$				
1	0.32	2.64	25.62	648.18
2	0.63	1.86	14.43	332.55
3	0.62	1.67	14.58	149.23
4	0.65	1.66	10.05	84.37
$l = 5$				
1	0.43	3.44	32.73	912.90
2	0.66	2.26	20.79	216.12
3	0.68	2.10	16.25	153.08
4	0.69	1.95	16.31	155.75
5	0.76	2.06	12.38	94.59
$l = 6$				
1	0.54	3.96	39.92	1237.71
2	0.74	2.59	22.10	219.50
3	0.75	2.22	17.15	156.95
4	0.76	2.24	18.09	161.69
5	0.82	2.19	19.06	165.57
6	0.86	2.27	14.98	105.21

p	h^{-1}				
	32	64	128	192	256
$l = 7$					
1	0.66	5.13	47.11	171.49	1479.28
2	0.79	3.17	28.60	103.44	667.80
3	0.77	2.74	23.54	82.53	227.45
4	0.82	2.70	19.14	62.73	166.62
5	0.88	3.55	20.95	66.59	361.98
6	0.94	2.80	21.71	68.22	176.53
7	0.97	2.78	18.56	51.21	119.14
$l = 8$					
1	0.79	5.96	54.17	306.79	1725.53
2	0.86	3.74	29.99	104.48	771.83
3	0.84	3.30	25.52	86.95	233.69
4	0.86	3.08	19.95	64.44	170.92
5	0.94	3.55	22.14	69.20	178.03
6	1.02	3.62	24.37	73.58	183.49
7	1.07	3.78	25.52	76.36	190.79
8	1.08	4.67	22.30	59.38	132.55
$l = 10$					
1	1.08	7.97	70.15	688.04	
2	0.97	4.89	38.64	132.98	1111.04
3	0.95	4.16	32.82	113.15	685.93
4	0.99	4.43	28.75	94.33	248.61
5	1.12	4.13	25.35	76.26	434.87
6	1.18	4.50	27.88	81.52	197.62
7	1.22	4.69	29.99	86.40	205.91
8	1.30	5.49	32.45	92.05	212.42

one processor is less than one second and parallelization is not necessary. For medium size problems the parallel efficiency on two processors is close to 90% but on three and more processors it decreases. The reason is that communication between two processors in one node is much faster than communication between nodes. For the largest problems ($h^{-1} = 256$) the available physical memory was not enough to solve the problem on one processor. The corresponding numbers in boxes show an atypical progression which is due to the usage of swap memory. The numerical results show that the main advantage of the parallel algorithm is that we can easily solve large problems using a parallel system with distributed memory.

Figure 1 shows the speed-up S_p of the full version of the algorithm obtained for $h^{-1} = 128$ and $l = 3, 4, \ldots 10$. As it was expected when the number of equations l is divisible by the number of processors p the parallel efficiency of the parallel algorithm is higher. The reason is the partitioning of the vectors u_k, d_k, r_k, z_k onto the processors described in previous section.

Table 3. Execution time for GCG-LS(0) for $s = 5$

p	h^{-1}			
	32	64	128	256
	$l = 8$			
1	0.84	6.07	57.02	2046.74
2	0.48	3.46	31.01	935.01
3	0.51	3.16	26.69	255.81
4	0.59	2.99	21.45	189.93
5	0.67	3.52	23.86	428.05
6	0.76	3.62	26.81	437.50
7	0.82	4.15	29.04	215.17
8	0.85	5.38	26.00	155.73

p	h^{-1}			
	32	64	128	256
	$l = 10$			
1	1.16	8.51	76.50	
2	0.65	4.87	41.57	1335.88
3	0.67	4.55	36.44	817.74
4	0.71	4.46	32.03	275.20
5	0.86	4.72	29.53	522.18
6	0.96	5.14	32.62	533.91
7	1.06	5.77	35.31	471.83
8	1.09	6.60	38.63	482.45

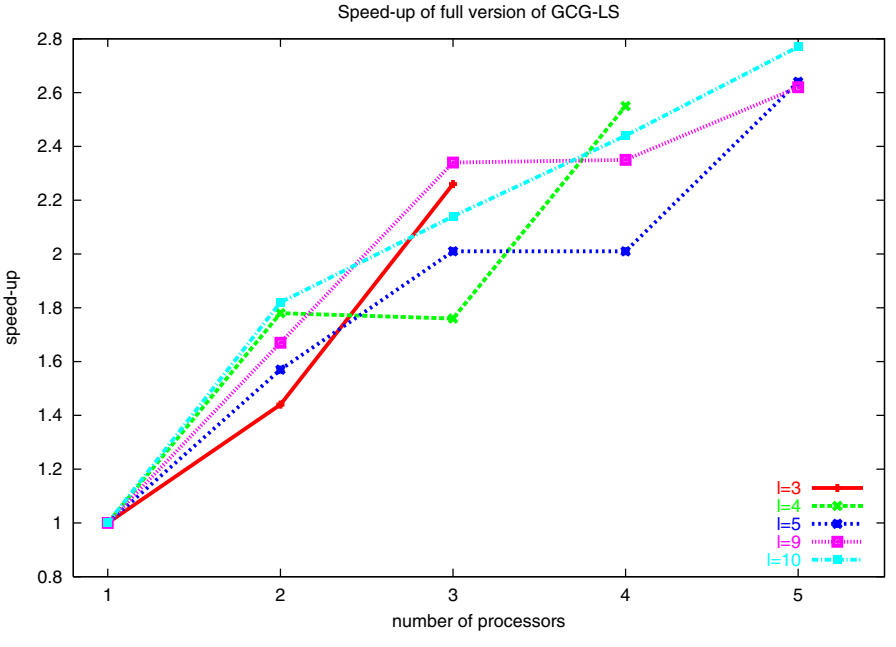

Fig. 1. Speed-up of the full version of GCG-LS algorithm

6 Concluding Remarks and Future Work

In this paper we have reported on the parallel performance of a new preconditioner applied to the generalized conjugate gradient method used to solve a sparse linear system arising from systems of convection-diffusion equations. The proposed preconditioner has inherent parallelism — the preconditioning step is implemented without any communications between processors. We have shown

that the code parallelizes well, resulting in a highly efficient treatment of large-scale problems.

The next step in development of the parallel code will be the implementation of matrix vector products using nonblocking MPI_Isend functions and avoiding communications for zero elements of the matrix V. In this way we can overlap the computation of part of the product and communication between processors. Our future plans include an approximation of the blocks of the preconditioner in order to implement a parallel preconditioning step on multiprocessor systems with more processors.

Acknowledgments

This research was supported by grant I-1402/2004 from the Bulgarian NSF and by the project BIS-21++ funded by FP6 INCO grant 016639/2005, and by Hungrarian Research Grant OTKA No. T 043765.

References

1. Anderson, E., Bai, Z., Bischof, C., Blackford, S., Demmel, J., Dongarra, J., Du Croz, J., Greenbaum, A., Hammarling, S., McKenney, A., Sorensen, D.: *LAPACK Users' Guide*, third edition, SIAM, Philadelphia, 1999, http://www.netlib.org/lapack/.
2. Axelsson, O.: A generalized conjugate gradient least square method. Numer. Math. **51** (1987), 209–227.
3. Axelsson, O.: Iterative Solution Methods. Cambridge University Press, 1994.
4. Axelsson, O., Karátson J.: Symmetric part preconditioning for the conjugate gradient method in Hilbert space. Numer. Funct. Anal. **24** (2003), No. 5-6, 455–474.
5. Axelsson, O., Karátson J.: Superlinearly convergent CG methods via equivalent preconditioning for nonsymmetric elliptic operators. Numer. Math. **99** (2004), No. 2, 197-223.
6. Faber, V., Manteuffel, T., Parter, S.V.: On the theory of equivalent operators and application to the numerical solution of uniformly elliptic partial differential equations. Adv. in Appl. Math., **11** (1990), 109–163.
7. Karátson J., Kurics T.: Superlinearly convergent PCG algorithms for some non-symmetric elliptic systems. J. Comp. Appl. Math., to appear
Preprint: http://www.cs.elte.hu/applanal/eng/preprint_eng.html **2006-09**.
8. Manteuffel, T., Otto, J.: Optimal equivalent preconditioners. SIAM J. Numer. Anal. **30** (1993), 790–812.
9. Manteuffel, T., Parter, S. V.: Preconditioning and boundary conditions. SIAM J. Numer. Anal. **27** (1990), no. 3, 656–694.
10. Snir, M., Otto, St., Huss-Lederman, St., Walker, D., Dongara, J.: *MPI: The Complete Reference*. Scientific and engineering computation series (The MIT Press, Cambridge, Massachusetts, 1997) Second printing.
11. Walker, D., Dongara, J.: MPI: a standard Message Passing Interface, *Supercomputer* **63** (1996) 56–68.
12. Zlatev, Z.: Computer treatment of large air pollution models. Kluwer Academic Publishers, Dordrecht-Boston-London, 1995.