TECHNISCHE UNIVERSITÄT MÜNCHEN

INSTITUT FÜR INFORMATIK

Sonderforschungsbereich 342:
Methoden und Werkzeuge für die Nutzung
paralleler Rechnerarchitekturen

Matrix Multilevel Methods and Preconditioning

Thomas Huckle

TUM-I9828
SFB-Bericht Nr. 342/11/98 A
November 98

TUM–INFO–11-I9828-150/1.–FI

Anforderungen an: Prof. Dr. A. Bode
Sprecher SFB 342
Institut für Informatik
Technische Universität München
D-80290 München, Germany

Druck: Fakultät für Informatik der
Technischen Universität München

Matrix Multilevel Methods and Preconditioning

Abstract

The Matrix-Multilevel approach is based on a purely matrix dependent description of Multigrid and related methods. The formulation of Multilevel methods as singular matrix extensions leads to the description of the Multilevel method as a preconditioned iterative scheme, and illuminates the significance of the used prolongation, resp. restriction operator for the related preconditioner. As matrix dependent black box restriction C we introduce a shifted form of the original matrix A, namely $C = B(:, 1 : 2 : n)$ with $B = \alpha I - A$, α a good upper bound for the largest eigenvalue of A. This mapping is chosen in such a way that via the related preconditioner the small eigenvalues are enlarged while the maximum eigenvalue remains nearly unchanged. If the components of each eigenvector of A can be seen as a discretization of a continuous function, then we derive estimates on the improved condition number after one step. We mainly consider symmetric positive definite matrices related to 1D-problems, but the results can be directly generalized to nonsymmetric and higher dimensional problems.

1. The Twolevel Method

We consider a linear equation $Ax = b$ with a sparse ill-conditioned $n \times n$ matrix A. The aim is to design a purely algebraic multilevel method that can be applied to any matrix in order to reduce the condition number. First we restrict ourself to the symmetric positive definite case.

The Multigrid method allows the fast $O(n)$ solution of linear equations arising from elliptic PDE (see [1,3,9,10]). The method uses a sequence of grids, and the restrictions and prolongations between the original problem formulated on the different grids. We can consider such methods purely algebraicly based on the matrix A without any geometrical information [13,8].

In the symmetric case the multilevel method is based on a mapping C for the restriction and prolongation of the original linear system on a coarser problem. Then we get $C^T AC$, e.g. as an $n/2 \times n/2$ matrix related to the original problem formulated on a coarse grid. Following [6] and the idea of generating systems we can write the sequence of matrices on different levels also as a sequence of matrix extensions of the form

$$A^{(1)} = A \ , \ A^{(2)} = \begin{pmatrix} A & AC \\ C^T A & C^T AC \end{pmatrix} = \begin{pmatrix} I \\ C^T \end{pmatrix} A \begin{pmatrix} I & C \end{pmatrix} . \qquad (1)$$

Let us first analyse the relation between the original equation $Ax = b$ and the extended matrix $A^{(2)}$. If $(y^T\ z^T)^T$ is a solution of the extended system

$$\begin{pmatrix} A & AC \\ C^TA & C^TAC \end{pmatrix} \begin{pmatrix} y \\ z \end{pmatrix} = \begin{pmatrix} b \\ a \end{pmatrix} ,$$

it is obvious that we have to set $a = C^Tb$, and then $x = y + Cz$ gives the solution of the original problem $Ax = b$. Furthermore, in view of (1), the kernel of $A^{(2)}$ is spanned by the vectors that fulfill $y = -Cz$, and hence the kernel is given by all vectors of the form $\begin{pmatrix} -C \\ I \end{pmatrix} z$. Similarly with (1) the range of $A^{(2)}$ is of the form $\begin{pmatrix} I \\ C^T \end{pmatrix} x$.

To derive the nonzero eigenvalues of $A^{(2)}$ we consider the Rayleigh Quotient relative to the space that is orthogonal to the null space $y = \begin{pmatrix} I \\ C^T \end{pmatrix} x$. With

$$\frac{y^T \begin{pmatrix} I \\ C^T \end{pmatrix} A \begin{pmatrix} I & C \end{pmatrix} y}{y^Ty} = \frac{x^T \begin{pmatrix} I & C \end{pmatrix} \begin{pmatrix} I \\ C^T \end{pmatrix} A \begin{pmatrix} I & C \end{pmatrix} \begin{pmatrix} I \\ C^T \end{pmatrix} x}{x^T \begin{pmatrix} I & C \end{pmatrix} \begin{pmatrix} I \\ C^T \end{pmatrix} x} =$$

$$\frac{x^T(I + CC^T)A(I + CC^T)x}{x^T(I + CC^T)x} = \frac{z^T(I + CC^T)^{1/2}A(I + CC^T)^{1/2}z}{z^Tz} ,$$

we see that the nontrivial spectrum of $A^{(2)}$ is given by the eigenvalues of

$$(I + CC^T)A , \tag{2}$$

and furtherhmore the nonzero eigenvalues of $A^{(2)}$ are related to the eigenvalues of A by $\lambda(A^{(2)}) = \lambda(A)(1 + \epsilon)$ with $0 \le \epsilon \le \lambda_{max}(CC^T)$.

Now we can think of the restriction C also as a preconditioner of the form $I + CC^T$ applied on the original matrix A (see also [2,14]). To be efficient the preconditioner should enlarge the small eigenvalues of A without generating larger than $\lambda_{max}(A)$. Then the condition number of the preconditioned system would be improved. Hence, the main task is to find a sparse matrix C that is a good approximation on a subspace related to the small eigenvalues. Similar problems are considered and solved in [4]; but to obtain the exact solution of such kind of problems is to expensive and can not be used here.

As an example we restrict ourself to the special case that the size of C is reduced to one column. Then an optimal preconditioner should enlarge $\lambda_1 = \lambda_{min}$ without changing $\lambda_n = \lambda_{max}$. Based on the eigensystem for A of the form $A = Q^T\Lambda Q$ the problem can be written as: Find a vector w such that the matrix $\tilde{\Lambda} = \begin{pmatrix} I & w \end{pmatrix}^T \Lambda \begin{pmatrix} I & w \end{pmatrix}$ has minimum condition number (neglecting the zero eigenvalue $\tilde{\lambda}_1 = 0$). In view of the interlace property (see e.g. [12]) we get

$$0 = \tilde{\lambda}_1 \le \lambda_1 \le \tilde{\lambda}_2 \le \lambda_2 \ , \quad \tilde{\lambda}_n \le \lambda_n \le \tilde{\lambda}_{n+1} \ ,$$

and the new condition number is bounded by

$$cond((I + ww^T)\Lambda) = \frac{\tilde{\lambda}_{n+1}}{\tilde{\lambda}_2} \geq \frac{\lambda_n}{\lambda_2} \; .$$

An optimal solution is therefore given by $C = \rho u_{\min}$ - the eigenvector related to the smallest eigenvalue. Then ρ has to be chosen such that $\lambda_2 \leq (1+\rho^2)\lambda_1 \leq \lambda_n$, and the condition number is improved by a factor λ_1/λ_2.

In general we are interested in a larger rank of C which is also necessary to lead to a notable improvement of the condition number for ill-conditioned matrices. Therefore we now write C in the form

$$C \; = \; B\,P \; = \; B(:,J) \tag{3}$$

with an $n \times n$ matrix B and an elemtary projection P which is only related to picking certain columns out of B, given by the index set J.

To make things easier we first describe the case that $C = B$ is a full $n \times n$ matrix. Then we can choose $B = \beta(\alpha I - A)$ with $\alpha = \lambda_{max}(A)$. This matrix has the desired property: λ_{min} becomes large in (2) and λ_{max} remains the same. For this special case we can fully analyse the resulting preconditioned system in order to find an optimal value for β.

Let u be any eigenvector of A with length 1 related to an eigenvalue λ, $\lambda_1 \leq \lambda \leq \alpha$. Then β has to be chosen as large as possible with

$$u^T\Big(I + \beta^2(\alpha I - A)(\alpha I - A)^T\Big)Au \; = \; \lambda\Big(1 + \beta^2(\alpha - \lambda)^2\Big) \leq \alpha \; . \tag{4}$$

Hence, $\beta^2 \leq \frac{1}{\lambda(\alpha-\lambda)}$. The function on the right hand side takes its minimum value for $\lambda = \alpha/2$, which leads to the optimal value $\beta = 2/\alpha$. The change of the eigenvalues of A under the transformation (2) is described by

$$\lambda \;\longrightarrow\; f(\lambda) = \lambda(1 + \frac{4}{\alpha^2}(\alpha - \lambda)^2) \; .$$

In the interval $[\lambda_1, \alpha]$ the function f has a relative maximum at $\lambda = \alpha/2$ of size $f(\alpha/2) = \alpha$, a relative minimum for $\lambda = 5\alpha/6$ with $f(5\alpha/6) = 50\alpha/54$, a global maximum for $\lambda = \alpha$ with $f(\alpha) = \alpha$, and a global minimum for $\lambda = \lambda_1$ with $f(\lambda_1) = \lambda_1(1+\frac{4}{\alpha^2}(\alpha-\lambda_1)^2) \approx 5\lambda_1$. Hence, by applying $I+BB^T$ as preconditioner the condition number is improved by a factor of 5. Note, that not only the smallest eigenvalue is enlarged, but the whole spectrum is compressed. For example all eigenvalues of A in the interval $[\alpha/8, \alpha]$ are mapped into the interval $[65\alpha/128, \alpha] \approx [\alpha/2, \alpha]$, and the interval $[\alpha/3, \alpha]$ into $[25/27\alpha, \alpha]$.

Now we return to the twolevel approach with nontrivial P. Then the above relation (4) translates into

$$u^T(I + \beta^2(\alpha I - A)PP^T(\alpha I - A)^T)Au \; = \; \lambda(1 + \beta^2(\alpha - \lambda)^2\|P^Tu\|^2) \leq \alpha \; . \tag{5}$$

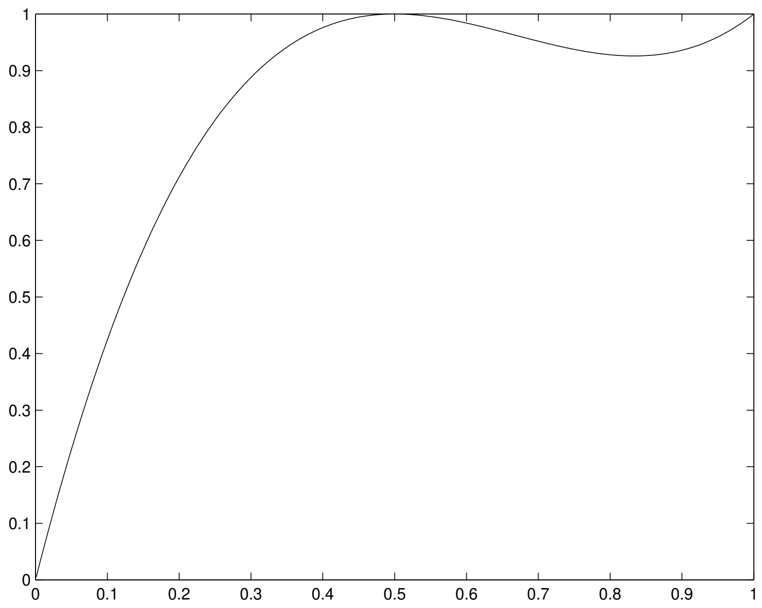

Figure 1: Function $f(\lambda)$ for $\alpha = 1$

Therefore P has to be chosen carefully in such a way that for every small eigenvalue $P^T u$ does not become to small. If the eigenvectors are continuous in the sense that they can be seen as values $g(j/n)$ for a continuous function g, then e.g. $P = I(:,1:2:n)$ gives $||P^T u||^2 \approx 1/2$ for all eigenvectors. This leads to an optimal value of $\beta^2 = 8/\alpha^2$ and we can expect that the smallest eigenvalue is improved by a factor of ≈ 5; but now the related mapping C has only half the number of entries. Note, that for $A = (1/4) * tridiag(-1, 2, -1)$ this optimal factor 8 also appears by diagonal (Jacobi) scaling of the extended system (1) (compare the BPX- or MDS-method [6]). In this case the spectrum of $A^{(1)}$ is no more contained in the interval $[5\lambda_{min}, \alpha]$, but the eigenvalues of A_2 are again smaller than α.
Remark: The eigenvalues of $A_2 = C^T AC$ are closely related to the function $g(\lambda) = \beta^2(\alpha - \lambda)^2\lambda$, and therefore the spectrum of A_2 is contained in the interval

$$[\beta^2(\alpha - \lambda_{min})^2\lambda_{min}(A), (\beta\alpha)^2(4/27)\alpha] \ .$$

For $P = I(:, 1 : 2 : n)$, an eigenvector u to a small eigenvalue of A leads to a small value of the Rayleigh Quotient related to the matrix A_2 and the vector $P^T u$. Therefore $P^T u$ is mainly contained in a subspace spanned by eigenvectors of A_2 that belong to small eigenvalues.

For many examples there is another easy way to derive a matrix B with eigenvalues in reversed order. Let us choose $B = |A|$ the matrix with the entries $B_{i,j} = |A_{i,j}|$ (see [10,5] for such matrix dependent prolongation/restriction operators). For $A = tridiag(-1, 2, -1)$ this leads to the standard prolongation with $B = tridiag(1, 2, 1)$. Hence, for many matrices after diagonal scaling of A we can expect a similar behaviour of the Multilevel approach related to $B = |A|$ and to $B = \alpha I - A$, because the diagonal scaling transforms A nearly to $tridiag(-1, 2, -1)$.

2. The Multilevel method

The analysis of the previous section describes a twolevel method. Now we have to generalize the approach that it works in a multilevel fashion. Until now we have arrived at the representation $A = A_1 = A^{(1)}$, $A_2 = C_1^T A_1 C_1$,

$$A^{(2)} \;=\; \begin{pmatrix} A_1 & A_1 C_1 \\ C_1^T A_1 & C_1^T A_1 C_1 \end{pmatrix} = (\, I \quad C_1\,)^T A_1 \,(\, I \quad C_1\,) \;,$$

or in preconditioned form

$$(I + C_1 C_1^T) A_1 \;.$$

Let us assume that the projection C_1 is chosen properly such that it strongly contains the eigenvectors related to small eigenvalues. Then, on the next level we can restrict ourself to projections of the form $C = C_1 C_2$. Now we can apply (2) a second time and arrive at a preconditioner

$$(I{+}C_1C_2C_2^TC_1^T)(I{+}C_1C_1^T)A_1 \;=\; (I{+}C_1C_1^T{+}C_1C_2C_2^TC_1^T{+}C_1C_2C_2^TC_1^TC_1C_1^T)A_1 \;.$$

To make the preconditioner symmetric positive definite we delete the last nonsymmetric term and use only

$$I + C_1 C_1^T + C_1 C_2 C_2^T C_1^T \;=\; I + C_1 (I + C_2 C_2^T) C_1^T \;.$$

Then we have different formulas for the extended system:

$$A^{(3)} = \begin{pmatrix} A & AC_1 & AC_1C_2 \\ C_1^TA & C_1^TAC_1 & C_1^TAC_1C_2 \\ C_2^TC_1^TA & C_2^TC_1^TAC_1 & C_2^TC_1^TAC_1C_2 \end{pmatrix} = \begin{pmatrix} I \\ C_1^T \\ C_2^TC_1^T \end{pmatrix} A\,(\, I \quad C_1 \quad C_1C_2\,)$$

$$= (\, I \quad C_1\,(\, I \quad C_2\,)\,)^T A\,(\, I \quad C_1\,(\, I \quad C_2\,)\,) =$$

$$= \begin{pmatrix} I & 0 & 0 \\ 0 & I & C_2 \end{pmatrix}^T \begin{pmatrix} A & AC_1 \\ C_1^TA & C_1^TAC_1 \end{pmatrix} \begin{pmatrix} I & 0 & 0 \\ 0 & I & C_2 \end{pmatrix} =$$

$$= \begin{pmatrix} I & 0 & 0 \\ 0 & I & C_2 \end{pmatrix}^T (\, I \quad C_1\,)^T A\,(\, I \quad C_1\,) \begin{pmatrix} I & 0 & 0 \\ 0 & I & C_2 \end{pmatrix} \;, \tag{6}$$

and in preconditioned form

$$(I + C_1C_1^T + C_1C_2C_2^TC_1^T)A = (I + C_1(I + C_2C_2^T)C_1^T)A$$

or

$$\begin{pmatrix} I & 0 \\ 0 & I + C_2C_2^T \end{pmatrix} \begin{pmatrix} A & AC_1 \\ C_1^TA & C_1^TAC_1 \end{pmatrix} = \begin{pmatrix} A & AC_1 \\ (I + C_2C_2^T)C_1^TA & (I + C_2C_2^T)C_1^TAC_1 \end{pmatrix} . \tag{7}$$

This leads to different heuristics for choosing C_2. In view of (7) we can think of C_2 as a second preconditioning step related to $A_2 = C_1^T A C_1$ and therefore we can set

$$C_2 = \beta_2(\alpha_2 I - A_2)\, P_2 \; . \tag{8}$$

We can derive (8) also based on another approach. The new projection C_2 defines the preconditioned system

$$(I + C_1 C_1^T + C_1 C_2 C_2^T C_1^T) A$$

and thus in the sense of (4) and (5) we get

$$u^T C_1 C_2 C_2^T C_1^T u = \beta_1^2 u^T (\alpha I - A) P C_2 C_2^T P^T (\alpha I - A)^T u$$

$$= \beta_1^2 (\alpha - \lambda_{min})^2 (u_{min}^T P) C_2 C_2^T (P^T u_{min}) =$$

$$= \beta_1^2 (\alpha - \lambda_{min})^2 (u_{min}^T P_1) B_2 P_2 P_2^T B_2 (P_1^T u_{min}) \; .$$

Now, B_2 should be chosen in such a way that it gets large for the vectors $P_1^T u$ related to small eigenvalues of A. In view of the previous remark at the end of Section 1 we can expect that the vectors $P_1^T u$ are related to small eigenvalues of A_2 which again suggests to define C_2 via (8).

We can also formulate another way for choosing C_2. Note, that the eigenvalues of A_2 are closely related to the function $g(\lambda) = \beta^2(\alpha - \lambda)^2 \lambda$. This shows that the large eigenvalues of A are also translated into very small eigenvalues of A_2. If we define C_2 with (8), then in this second step we try to enlarge these originally large eigenvalues together with the small eigenvalues of A. This suggests another way to define C_2, namely again as a projection of the first level-matrix $\alpha I - A$. If for example A is a Toeplitz matrix then we can consider the Toeplitz matrix $\alpha I - A = T$ and choose C_j as a submatrix $T(1 : 2^l, 2 : 2 : 2^l)$ (for similar multigrid methods for Toeplitz matrices see [7]).

In order to obtain similar improvements on the condition number on every level it is necessary that all the derived smaller systems have similar porperties as the original matrix A. If for example A_2 is well conditioned then obviously another projection will lead to no improvement of the spectrum. Hence, we have to choose C and P in such a way that the matrix $\tilde{A} = P^T B^T A B P$ inherits important properties of A. In many cases the behaviour of A on the vector $e_n = (1, ..., 1)^T$ is very important - this is related to the property that the rowsum of entries is often zero. Hence we may ask that $e_{n/2}^T \tilde{A} e_{n/2} = e_n(J)^T B^T A B e_n(J) \approx e_n^T A e_n / 2$. We obtain this property by choosing B such that $B e_n(J) = e_{n/2}/\sqrt{2}$. For $B = \sqrt{2} * tridiag(1/4, 1/2, 1/4)$ and $J = (2, 4, 6, ..., n)$ (the usual Multigrid prolongation) this is obviously fullfilled. In many cases after diagonal scaling this is also nearly satisfied for both mappings $B = \lambda_{max} I - A$ and $B = |A|$.

Now we have defined a multilevel method based only on the original matrix A and the maximum eigenvalues of the resulting systems A_j. It is necessary to include also some kind of smoothing operation on every level to get fast convergence. Here we will mainly consider the Jacobi method for smoothing. In (1) or (6) the Jacobi smoothing is nothing else then diagonal preconditioning. Note, that in the same way one can use Gauss-Seidel or any other levelwise method.

To include Jacobi smoothing let us consider the enlarged problem

$$A^{(k)} = \begin{pmatrix} A & AC_1 & AC_1C_2 & \cdots & AC_1..C_k \\ C_1^TA & C_1^TAC_1 & C_1^TAC_1C_2 & \cdots & C_1^TAC_1..C_k \\ \vdots & & \cdot & & \vdots \\ \vdots & & & \cdot & \vdots \\ C_k^T..C_1^TA & \cdots & \cdots & & C_k^T..C_1^TAC_1..C_k \end{pmatrix} =$$

$$= (\, I \quad C_1 \quad C_1C_2 \quad \cdots \quad C_1..C_k\,)^T A\,(\, I \quad C_1 \quad C_1C_2 \quad \cdots \quad C_1..C_k\,)$$

$$= (\, I \quad C_1(I\; C_2(I\; \cdots\; (I\; C_k)\cdots))\,)^T A\,(\, I \quad C_1(I\; C_2(I\; \cdots\; (I\; C_k)\cdots))\,)\ , \tag{9}$$

and in preconditioned form

$$\begin{aligned}(I + C_1C_1^T + C_1C_2C_2^TC_1^T + \;\cdots\; + C_1..C_kC_k^T..C_1^T)A = \\ (I + C_1(I + C_2(I + \;\cdots\; (I + C_kC_k^T)\cdots)C_2^T)C_1^T)A = M^{(k)}A\ .\end{aligned} \tag{10}$$

In the form (9) we can comprise any preconditioner on the matrix $A^{(k)}$, for example Jacobi, Gauss-Seidel or ILU preconditioner, and employ the conjugate gradient method with zero starting vector. But usually we want to compute only the small matrices A_j and the projections C_j on every level and not the whole system $A^{(k)}$ much less the - nearly dense - preconditioner $M^{(k)}$. Therefore, we will use only levelwise block-diagonal preconditioners based on the level matrices A_j.

From (9) we can translate preconditioners very easily to the form (10). In the Jacobi case for example we have $D_j = diag(A_j) = diag(C_j^T \cdots C_1^TAC_1 \cdots C_j)$ and for every matrix A_j we can use $D_j^{-1/2}$ as left and right preconditioner. Then, with

$$D = diag(D_1^{-1/2}, \cdots, D_k^{-1/2})\ ,$$

(9) translates into

$$D \begin{pmatrix} A & AC_1 & AC_1C_2 & \cdots & AC_1..C_k \\ C_1^TA & C_1^TAC_1 & C_1^TAC_1C_2 & \cdots & C_1^TAC_1..C_k \\ \vdots & & \cdot & & \vdots \\ \vdots & & & \cdot & \vdots \\ C_k^T..C_1^TA & \cdots & \cdots & & C_k^T..C_1^TAC_1..C_k \end{pmatrix} D \;=$$

$$\begin{pmatrix} I & D_1^{1/2}C_1D_2^{-1/2} & D_2^{1/2}C_2D_3^{-1/2} & \cdots \end{pmatrix}^T D_1^{-1/2}AD_1^{-1/2} \begin{pmatrix} I & D_1^{1/2}C_1D_2^{-1/2} & \cdots \end{pmatrix} .$$

Hence, we only have to replace A by $\tilde{A} = D_1^{-1/2} A D_1^{-1/2}$, and each C_j by $\tilde{C}_j = D_j^{1/2} C_j D_{j+1}^{-1/2}$. This leads to the new preconditioned form

$$(I + \tilde{C}_1(I + \tilde{C}_2(I + \ \cdots (I + \tilde{C}_k \tilde{C}_k^T) \cdots) \tilde{C}_2^T) \tilde{C}_1^T) \tilde{A} \ . \tag{11}$$

Note that here the Jacobi scaling is also necessary to obtain that the matrices $\tilde{A}_j$ share the same properties (in the sense that the rowsums are nearly zero) .

3. Numerical Examples

For practical implementation of the method we will consider different variations:

- We will always include the diagonal preconditioner in the form (11).
- We will construct C_j and $\tilde{C}_j$ based on A_j and an approximate maximum eigenvalue α_j of $\tilde{A}_j$ in the form $C_j = (\alpha_j I - A_j)P$.
- We can compute α_j - the approximation on the largest eigenvalue - on every level, or always use $\alpha_j = \alpha = \lambda_{max}(A)$. Note that if α is chosen properly then $\lambda_{max}(A_j) \leq \lambda_{max}(A)$ for every j.

As numerical examples we consider the finite difference discretization of the 1D elliptic PDE $(a(x)_x u(x))_x = f(x)$ with Dirichlet boundary conditions for

(1) $a(x)$ constant,
(2) $a(x) = 1 + sin(8\pi x)^2$,
(3) $a(x) = 1 + sin(16\pi x)^2$,
(4) $a(x) = 1 + sin(32\pi x)^2$,
(5) $a(x) = 1 + exp(\pi x) sin(8\pi x)^2$,
(6) $a(x) = 1 + exp(2\pi x) sin(8\pi x)^2$,
(7) $a(x) = 1 + exp(8\pi x) sin(8\pi x)^2$,
(8) $a(x) = sin(\pi x)^2$.
(9) $a(x)$ is piecewise constant with ten different values between 0.1 and 2.1.
(10) $a(x)$ is piecewise constant with ten different values between 0.1 and $2.1E + 8$.

In the first examples we always choose α as the exact eigenvalue of A. The next tables display the condition numbers for the matrix multilevel method based on
- estimating α_j on every level (MML),
- only on the finest level (MML0),
- for the multigrid method written in the MDS-form that is strongly related to the BPX-preconditioner [2,6],
- and for the Jacobi-preconditioned original problem.

$n=2^l$	MML	MML0	$\lvert A\rvert$	MDS	$D^{-1}A$
5	4.55	6.32	5.46	4.60	414.3
6	5.43	7.26	6.35	5.12	1.7E3
7	6.34	8.19	7.27	5.62	6.6E3
8	7.26	9.14	8.20	6.11	2.7E4

Table 1. Condition number example 1, $a(x) = const.$

$n=2^l$	MML	MML0	$\lvert A\rvert$	MDS	$D^{-1}A$
5	4.41	6.11	5.28	11.60	440.6
6	5.38	7.15	6.27	10.74	1.8E3
7	6.32	8.15	7.24	10.19	7.05E3
8	7.26	9.13	8.19	10.42	2.8E4

Table 2. Condition number example 2, $a(x) = 1 + sin(8\pi x)^2$.

$n=2^l$	MML	MML0	$\lvert A\rvert$	MDS	$D^{-1}A$
5	4.31	6.04	5.23	5.75	466
6	5.25	7.02	6.14	29.2	1.8E3
7	6.25	8.07	7.16	25.3	7.0E3
8	7.23	9.09	8.16	22.6	2.8E4

Table 3. Condition number example 3, $a(x) = 1 + sin(16\pi x)^2$.

$n=2^l$	MML	MML0	$\lvert A\rvert$	MDS	$D^{-1}A$
5	4.55	6.32	5.46	4.60	414
6	5.191	7.98	6.10	6.30	1.9E3
7	6.12	7.93	7.03	98.8	7.1E3
8	7.15	9.00	8.07	80.6	2.8E4

Table 4. Condition number example 4, $a(x) = 1 + sin(32\pi x)^2$.

$n=2^l$	MML	MML0	$\lvert A\rvert$	MDS	$D^{-1}A$
5	4.37	6.08	5.28	36.4	605.4
6	5.37	7.16	6.28	32.00	2.2E3
7	6.32	8.16	7.25	29.9	8.8E3
8	7.26	9.13	8.20	31.2	3.6E4

Table 5. Condition number example 5, $a(x) = 1 + exp(\pi x) sin(8\pi x)^2$.

$n = 2^l$	MML	MML0	$\lvert A \rvert$	MDS	$D^{-1}A$
5	4.37	6.10	5.26	250.5	2.6E3
6	5.37	7.17	6.28	149.4	6.2E3
7	6.32	8.18	7.25	97.5	1.7E4
8	7.26	9.14	8.20	86.3	6.2E4

Table 6. Condition number example 6, $a(x) = 1 + exp(2\pi x)sin(8\pi x)^2$.

$n = 2^l$	MML	MML0	$\lvert A \rvert$	MDS	$D^{-1}A$
5	4.25	5.94	5.13	8.5E8	6.3E9
6	5.47	7.17	6.31	5.7E8	1.3E10
7	6.42	8.23	7.31	3.7E8	2.7E10
8	7.32	9.19	8.25	2.2E8	5.5E10

Table 7. Condition number example 7, $a(x) = 1 + exp(8\pi x)sin(8\pi x)^2$.

$n = 2^l$	MML	MML0	$\lvert A \rvert$	MDS	$D^{-1}A$
5	12.07	6.08	5.57	42.25	1.1E4
6	13.91	7.02	6.51	81.93	8.6E4
7	15.72	7.96	7.45	164.7	6.9E5
8	17.54	8.92	8.41	340.7	5.6E6

Table 8. Condition number example 8, $a(x) = sin(\pi x)^2$.

Next, we compare the iteration numbers for different methods. In practice it is not possible to use the exact maximum eigenvalue. Therefore we consider also upper bounds or estimates for α. We define α as

- the exact maximum eigenvalue ($\alpha = \lambda_{max}(A)$)
- the 1-norm of A ($\alpha = ||A||_1$)
- as the Lanczos estimate λ_{max} after mi Lanczos steps
- as $\lambda_{max} + \lambda_{min}$ after mi Lanczos steps (denoted by $mi < 0$ in Tables 9).

The right hand side in our numerical examples was chosen to be $(1, ..., 1)^T$. The stopping criterion is fulfilled if the exact relative residual is less then 10^{-6}. We compute α on every level, but we get nearly the same results if we use only A for choosing α. In the following tables '-' denotes that the computation had to be omitted because it would have been to expensive.

α / $n = 2^l$	2^8	2^9	2^{10}	2^{11}
MDS	117	119	121	122
$\|\|A\|\|_1$	122	130	120	124
λ_{max}	15	-	-	-
$mi = 200$	16	17	32	39
$mi = 150$	15	25	32	46
$mi = 100$	17	37	46	58
$mi = 50$	42	58	69	89
$mi = 30$	57	78	98	118
$mi = 10$	108	123	183	256
$mi = -2$	15	16	16	17
$\|A\|$	15	16	16	17

Table 9. Iteration number for example 7, $a(x) = 1 + exp(8\pi x)sin(8\pi x)^2$.

For example 1-8 we get nearly the same condition number for the different matrix multilevel methods which is better than the multigrid condition numbers, and only slightly growing in the size n. From table 9 we see that the Lanczos estimate $\lambda_{max} + \lambda_{min}$ and $B = |A|$ give the best results.

α / $n = 2^l$	2^5	2^6	2^7	2^8	2^9	2^{10}	2^{11}
MDS	16	17	19	21	22	23	24
MML	14	15	15	15	16	17	17
$\|A\|$	13	13	14	14	15	15	15

Table 10. Iteration number for example 8.

α / $n = 2^l$	2^5	2^6	2^7	2^8	2^9	2^{10}	2^{11}
MDS	23	27	33	41	49	56	64
MML	12	13	14	14	15	15	15
$\|A\|$	12	13	14	14	15	15	15

Table 11. Iteration number for example 9.

α / $n = 2^l$	2^5	2^6	2^7	2^8	2^9	2^{10}	2^{11}
MDS	58	82	102	119	103	68	78
MML	11	13	14	15	16	16	16
$\|A\|$	12	14	14	15	16	16	16

Table 12. Iteration number for example 10.

These results show that the convergence in the matrix multilevel approach applied on the different test examples does not depend on the function $a(x)$, and is only slightly increasing with the problem size. The diagonal preconditioning that is used here is very easy to parallelize; therefore the matrix multilevel approach is especially interesting in a parallel environment.

4. Generalizations of the Matrix Multilevel Method

For the Jacobi or Gauss-Seidel iteration we often introduce damping factors. The same is possible here if we replace the preconditioner in (10) by

$$(I + \omega_1 C_1 (I + \omega_2 C_2 (I + ...) C_2^T) C_1^T) = (I + \omega_1 C_1 C_1^T + \omega_2 C_1 C_2 C_2^T C_1^T + ...) \, .$$

In view of (1) and the analysis of Section 1 a factor $\omega < 1$ may be necessary to reduce the maximum eigenvalue to be $\leq \alpha$. A factor $\omega > 1$ can be helpful for faster convergence if it is possible to enlarge the small eigenvalues without changing λ_{max}.

If the matrix $C^T AC = A_2 = LL^T$ can be inverted we define a preconditioner for the extended system (1) as a block diagonal matrix

$$\begin{pmatrix} diag(A)^{-1/2} & 0 \\ 0 & L^{-T} \end{pmatrix} ,$$

or a preconditioner for $diag(A)^{-1/2} \; A \; diag(A)^{-1/2}$ by

$$M = I + diag(A)^{1/2} \, C A_2^{-1} \, C^T \; diag(A)^{1/2} \; .$$

This can be useful if we are not able to apply the full multilevel method and have to stop at a certain level. Then we can include the inverse of the last A_j in the preconditioner.

For A **symmetric indefinite** the eigenvalues are negative and positive. Hence, to find a polynomial that enlarges the small eigenvalues without changing the large ones, we can choose $B = \alpha^2 I - A^2$. Then small eigenvalues λ are replaced by $\lambda(1+\rho)$ for a factor $\rho > 0$. The main disadvantage of this approach is that the subproblems A_j are loosing their sparsity. Hence, we have to stop at a level j, or apply the full multilevel method, but with total costs of at least $O(n \, log(n))$.

If the smallest eigenvalue is negative but near zero we can use the original approach and again set $B = \alpha I - A$ with $\alpha \approx \lambda_{max}(A)$. As example we consider the Finite-Difference dicretization of the one-dimensional Helmholtz equation $-(g(x)u'(x))' + cu(x) = 0$ with different values of c and $g(x) = 1 + exp(2x)sin(8x)$ (see [5]).

c	$n=2^l$	λ_{min}	MML(λ_{max})	mi=-2	MDS	$\|A\|$
1	5	7.5E-4	9	8	30	7
	6	3.2E-4	8	9	33	8
	7	1.1E-4	10	10	26	10
	8	3.2E-5	11	12	33	10
	9	7.7E-6	12	13	37	12
	10	-	-	12	40	12
10	5	5.3E-4	8	8	28	8
	6	2.5E-4	9	9	31	8
	7	9.7E-5	10	10	27	9
	8	2.7E-5	11	10	37	10
	9	6.6E-6	11	10	46	11
	10	-	-	11	39	12
100	5	-6.6E-3	12	13	36	17
	6	-1.3E-3	13	14	34	11
	7	-2.7E-4	16	16	38	13
	8	-6.5E-5	18	16	31	15
	9	-1.6E-5	19	17	36	17
	10	-	-	23	41	14
1000	5	-3.0E-1	34	30	47	24
	6	-5.3E-2	41	28	57	26
	7	-1.2E-2	40	35	51	37
	8	-2.9E-3	47	37	67	32
	9	-7.2E-4	>500	43	64	46
	10	-	-	46	62	41

Table 13. Iteration number for 1D Helmholtz equation

If A is **nonsymmetric** then for the mapping B we can use a low degree polynomial $p(A)$ with $p(0)=1$ and $|p(x)|$ is small for the extreme eigenvalues of A. In the normal case, $p(A, A^T) = |\lambda_{max}|^2 I - AA^T$ should be a good choice. To preserve the sparsity of the restricted linear systems A_j, one can choose the identity for the prolongation and $p(A, A^T)$ for the restriction or vice versa.

For many examples we can again set $B = \alpha I - A$ with $\alpha \approx \lambda_{max}((A+A^T)/2)$. As numerical example we consider the Finite-difference discretization of the one-dimensional Convection-Diffusion equation $-(g(x)u'(x))' + cu'(x) = f(x)$ with Dirichlet conditions for different values of c and $g(x) = 1 + exp(2x)sin(8x)$ (see [5]).

c	$n = 2^l$	MML(λ_{max})	MML(mi=-2)	MDS	$\|A\|$
1	5	9	9	35	8
	6	10	10	29	9
	7	10	10	33	10
	8	11	10	29	10
	9	-	11	34	11
	10	-	13	55	12
10	5	10	10	25	10
	6	12	11	29	11
	7	12	12	27	13
	8	15	14	36	14
	9	-	14	32	17
	10	-	14	39	15
100	5	24	24	33	33
	6	21	21	34	33
	7	20	20	33	20
	8	24	24	40	24
	9	-	28	39	27
	10	-	33	42	32
1000	5	73	72	66	72
	6	156	136	126	192
	7	> 500	222	214	> 500
	8	> 500	129	229	> 500
	9	-	91	254	> 500
	10	-	73	> 500	71

Table 14. Iteration number 1D Convection-Diffusion equation

The last example $A = tridiag(1, 2, 1)$ shows that the matrix multilevel approach can be applied to more general problems as the usual multigrid method.

$n = 2^l$	MML(mi=-2)	MDS
5	12	36
6	13	89
7	13	207
8	14	475
9	14	> 500
10	14	> 500

Table 15. Iteration number tridiag(1,2,1)

If the linear system is related to a **higher dimensional** problem we can introduce a modified technique in order to capture the structure of A in a better way. Let us consider the 2D case and a separable PDE. Then, the matrix A can e.g. be written as a Kronecker sum (see e.g. [11])

$$A = (A_1 \otimes I) \, + \, (I \otimes A_2) \; .$$

The matrix B that is applied in the Multigrid approach is given not by a Kronecker sum but by a Kronecker product $B_1 \otimes B_2$ with $B_i = tridiag(1, 2, 1)$. Hence, we will also choose a matrix B as a Kronecker product $B = B_1 \otimes B_2$ and then use B to define the matrix C in the form $C = BP$.

Now the eigenvalues and eigenvectors of A are given by the sum of the eigenvalues of A_1 and A_2, resp. the Kronecker product of the eigenvectors. Hence, every eigenvector of A is of the form $u = u_1 \otimes u_2$. In view of the product rules for the Kronecker product we get

$$Au = ((A_1 \otimes I) + (I \otimes A_2))(u_1 \otimes u_2) = (A_1 u_1 \otimes u_2) + (u_1 \otimes A_2 u_2) =$$

$$= (\lambda_1 + \lambda_2) u \; .$$

To find an efficient matrix B we again want to enlarge the small eigenvalues of A without changing the large ones. Hence, with

$$Bu = (B_1 \otimes B_2)(u_1 \otimes u_2) = (B_1 u_1 \otimes B_2 u_2)$$

we can choose

$$B_1 := (\alpha_1 I - A_1)\beta_1 \quad , \quad \text{and} \quad B_2 = (\alpha_2 I - A_2)\beta_2$$

with α_1 and α_2 the maximum eigenvalues of A_1, resp. A_2. Then the minimum eigenvalues are approximately enlarged by a factor $1 + (\beta_1\beta_2\alpha_1\alpha_2)^2$. Furthermore, the elementary projection matrix P can be chosen as $P = P_1 \otimes P_2$, and then on the second level we get

$$C^T A C \;=\; (C_1^T A_1 C_1 \otimes C_2^T C_2) + (C_1^T C_1 \otimes C_2^T A_2 C_2) \; .$$

In the next step we again can define the restriction matrix via the Kronecker product where the first factor is designed to improve on $C_1^T A_1 C_1$, and the second factor is related to $C_2^T A_2 C_2$. This is the direct generalization of the 1D approach to higher dimensions via the Kronecker product. We can think of the prolongations also in the form

$$C = BP = (B_1 \otimes B_2)(P_1 \otimes P_2) = (B_1 P_1 \otimes B_2 P_2) = (B_1 P_1 \otimes I)\,(I \otimes B_2 P_2) = F_1 F_2 \;\; .$$

Hence in general we can write A in the form $A = A_1 + \cdots + A_d$, where each term is related to the i-th direction. Then for each A_i we can define a matrix F_i, and in total we can set $C = F_1 F_2 \cdots F_d$.

The advantage of this approach is that it better captures the structure of A. If we apply the original method on the 2D case, the twolevel method again leads to an improved condition number, but in general one of the next matrices $A^{(k)}$ is more dense and/or well-conditioned, and then the multilevel approach will give no improvement in the following steps. Hence, sometimes it is necessary to modify the matrix multilevel approach in order to generate subproblems with similar properties as the original matrix A; only then a full multilevel method can be efficient. We will analyse the general 2D case in a forthcoming paper.

5. Conclusions

We have developed a purely matrix-dependent multilevel method for solving linear equations. The prolongation/restriction operator is defined by a shift $B = \lambda_{max}(A)I - A$, $B(:, 1:2:n)$, of the given matrix in order to enlarge $\lambda_{min}(A)$. This idea can be applied on every level of the method. The numerical results show an improvement over the Multigrid approach in the preconditioned form (BPX,MDS). The method can be generalized to indefinite, nonsymmetric, higher-dimensional problems and is a promising approach to derive further new matrix-dependent Multigrid algorithms. The derived approach can be seen as a modified polynomial preconditioner, modified by elementary projections and the included Jacobi-scaling.

References

[1] A. Brandt: *Multi-level adaptive solutions to boundary value problems*, Math. Comp. 31 (1977), pp. 333-390.

[2] J. Bramble, J. Pasciak, and J. Xu: *Parallel multilevel preconditioners*, Math. Comp. 55(191) (1990), pp. 1-22.

[3] W. Briggs: *A Multigrid Tutorial*, SIAM, Philadelphia, 1987.

[4] L. Elsner, C. He, V. Mehrmann: *Minimizing the condition number of a positive definite matrix by completion*, preprint September 1993.

[5] M. Griebel: *Zur Lösung von Finite-Differenzen- und Finite-Element-Gleichungen mittels der Hierarchischen-Transformations-Mehrgitter-Methode*, TU München, Institut f. Informatik, TUM-I9007: SFB-Report 342/4/90 A, 1990.

[6] M. Griebel: *Multilevel algorithms considered as iterative methods on semidefinte systems*, SIAM J. Sci. Comput. 15 (1994).

[7] G. Fiorentino, S. Serra: *Multigrid methods for Toeplitz matrices*, Calcolo 28 (1992), pp. 283-305.

[8] A. Greenbaum: *Analysis of a multigrid method as an iterative technique for solving linear systems*, SIAM J. Num. Anal. 21 (1984), pp. 473-485.

[9] W. Hackbusch, U. Trottenberg: *Multigrid Methods*, Springer-Verlag,Berlin,1982.

[10] W. Hackbusch: *Multigrid Methods and Applications*, Springer-Verlag, Berlin, 1985.

[11] R. Horn, C. Johnson: *Topics in Matrix Analysis*, Cambridge University Press, Cambridge, 1989.

[12] B. Parlett: *The Symmetric Eigenvalue Problem*, Prentice Hall, Englewood Cliffs, N.J., 1980.

[13] J.W. Ruge, K. Stüben: *Algebraic Multigrid.* Multigrid Methods, McCormick (ed.), SIAM, Philadelphia, 1987.

[14] H. Yserentant: *Old and new convergence proofs for multigrid methods*, Acta Numerica, Cambridge Univ. Press, N.Y., (1993), pp. 285-326.

SFB 342: Methoden und Werkzeuge für die Nutzung paralleler Rechnerarchitekturen

bisher erschienen :

Reihe A

Liste aller erschienenen Berichte von 1990-1994
auf besondere Anforderung

342/01/95 A Hans-Joachim Bungartz: Higher Order Finite Elements on Sparse Grids
342/02/95 A Tao Zhang, Seonglim Kang, Lester R. Lipsky: The Performance of Parallel Computers: Order Statistics and Amdahl's Law
342/03/95 A Lester R. Lipsky, Appie van de Liefvoort: Transformation of the Kronecker Product of Identical Servers to a Reduced Product Space
342/04/95 A Pierre Fiorini, Lester R. Lipsky, Wen-Jung Hsin, Appie van de Liefvoort: Auto-Correlation of Lag-k For Customers Departing From Semi-Markov Processes
342/05/95 A Sascha Hilgenfeldt, Robert Balder, Christoph Zenger: Sparse Grids: Applications to Multi-dimensional Schrödinger Problems
342/06/95 A Maximilian Fuchs: Formal Design of a Model-N Counter
342/07/95 A Hans-Joachim Bungartz, Stefan Schulte: Coupled Problems in Microsystem Technology
342/08/95 A Alexander Pfaffinger: Parallel Communication on Workstation Networks with Complex Topologies
342/09/95 A Ketil Stølen: Assumption/Commitment Rules for Data-flow Networks - with an Emphasis on Completeness
342/10/95 A Ketil Stølen, Max Fuchs: A Formal Method for Hardware/Software Co-Design
342/11/95 A Thomas Schnekenburger: The ALDY Load Distribution System
342/12/95 A Javier Esparza, Stefan Römer, Walter Vogler: An Improvement of McMillan's Unfolding Algorithm
342/13/95 A Stephan Melzer, Javier Esparza: Checking System Properties via Integer Programming
342/14/95 A Radu Grosu, Ketil Stølen: A Denotational Model for Mobile Point-to-Point Dataflow Networks
342/15/95 A Andrei Kovalyov, Javier Esparza: A Polynomial Algorithm to Compute the Concurrency Relation of Free-Choice Signal Transition Graphs
342/16/95 A Bernhard Schätz, Katharina Spies: Formale Syntax zur logischen Kernsprache der Focus-Entwicklungsmethodik
342/17/95 A Georg Stellner: Using CoCheck on a Network of Workstations
342/18/95 A Arndt Bode, Thomas Ludwig, Vaidy Sunderam, Roland Wismüller: Workshop on PVM, MPI, Tools and Applications
342/19/95 A Thomas Schnekenburger: Integration of Load Distribution into ParMod-C

Reihe A

342/20/95 A	Ketil Stølen: Refinement Principles Supporting the Transition from Asynchronous to Synchronous Communication
342/21/95 A	Andreas Listl, Giannis Bozas: Performance Gains Using Subpages for Cache Coherency Control
342/22/95 A	Volker Heun, Ernst W. Mayr: Embedding Graphs with Bounded Treewidth into Optimal Hypercubes
342/23/95 A	Petr Jančar, Javier Esparza: Deciding Finiteness of Petri Nets up to Bisimulation
342/24/95 A	M. Jung, U. Rüde: Implicit Extrapolation Methods for Variable Coefficient Problems
342/01/96 A	Michael Griebel, Tilman Neunhoeffer, Hans Regler: Algebraic Multigrid Methods for the Solution of the Navier-Stokes Equations in Complicated Geometries
342/02/96 A	Thomas Grauschopf, Michael Griebel, Hans Regler: Additive Multilevel-Preconditioners based on Bilinear Interpolation, Matrix Dependent Geometric Coarsening and Algebraic-Multigrid Coarsening for Second Order Elliptic PDEs
342/03/96 A	Volker Heun, Ernst W. Mayr: Optimal Dynamic Edge-Disjoint Embeddings of Complete Binary Trees into Hypercubes
342/04/96 A	Thomas Huckle: Efficient Computation of Sparse Approximate Inverses
342/05/96 A	Thomas Ludwig, Roland Wismüller, Vaidy Sunderam, Arndt Bode: OMIS — On-line Monitoring Interface Specification
342/06/96 A	Ekkart Kindler: A Compositional Partial Order Semantics for Petri Net Components
342/07/96 A	Richard Mayr: Some Results on Basic Parallel Processes
342/08/96 A	Ralph Radermacher, Frank Weimer: INSEL Syntax-Bericht
342/09/96 A	P.P. Spies, C. Eckert, M. Lange, D. Marek, R. Radermacher, F. Weimer, H.-M. Windisch: Sprachkonzepte zur Konstruktion verteilter Systeme
342/10/96 A	Stefan Lamberts, Thomas Ludwig, Christian Röder, Arndt Bode: PFSLib – A File System for Parallel Programming Environments
342/11/96 A	Manfred Broy, Gheorghe Ştefănescu: The Algebra of Stream Processing Functions
342/12/96 A	Javier Esparza: Reachability in Live and Safe Free-Choice Petri Nets is NP-complete
342/13/96 A	Radu Grosu, Ketil Stølen: A Denotational Model for Mobile Many-to-Many Data-flow Networks
342/14/96 A	Giannis Bozas, Michael Jaedicke, Andreas Listl, Bernhard Mitschang, Angelika Reiser, Stephan Zimmermann: On Transforming a Sequential SQL-DBMS into a Parallel One: First Results and Experiences of the MIDAS Project
342/15/96 A	Richard Mayr: A Tableau System for Model Checking Petri Nets with a Fragment of the Linear Time μ -Calculus

Reihe A

342/16/96 A	Ursula Hinkel, Katharina Spies: Anleitung zur Spezifikation von mobilen, dynamischen Focus-Netzen
342/17/96 A	Richard Mayr: Model Checking PA-Processes
342/18/96 A	Michaela Huhn, Peter Niebert, Frank Wallner: Put your Model Checker on Diet: Verification on Local States
342/01/97 A	Tobias Müller, Stefan Lamberts, Ursula Maier, Georg Stellner: Evaluierung der Leistungsf"ahigkeit eines ATM-Netzes mit parallelen Programmierbibliotheken
342/02/97 A	Hans-Joachim Bungartz and Thomas Dornseifer: Sparse Grids: Recent Developments for Elliptic Partial Differential Equations
342/03/97 A	Bernhard Mitschang: Technologie f"ur Parallele Datenbanken - Bericht zum Workshop
342/04/97 A	nicht erschienen
342/05/97 A	Hans-Joachim Bungartz, Ralf Ebner, Stefan Schulte: Hierarchische Basen zur effizienten Kopplung substrukturierter Probleme der Strukturmechanik
342/06/97 A	Hans-Joachim Bungartz, Anton Frank, Florian Meier, Tilman Neunhoeffer, Stefan Schulte: Fluid Structure Interaction: 3D Numerical Simulation and Visualization of a Micropump
342/07/97 A	Javier Esparza, Stephan Melzer: Model Checking LTL using Constraint Programming
342/08/97 A	Niels Reimer: Untersuchung von Strategien für verteiltes Last- und Ressourcenmanagement
342/09/97 A	Markus Pizka: Design and Implementation of the GNU INSEL-Compiler gic
342/10/97 A	Manfred Broy, Franz Regensburger, Bernhard Schätz, Katharina Spies: The Steamboiler Specification - A Case Study in Focus
342/11/97 A	Christine Röckl: How to Make Substitution Preserve Strong Bisimilarity
342/12/97 A	Christian B. Czech: Architektur und Konzept des Dycos-Kerns
342/13/97 A	Jan Philipps, Alexander Schmidt: Traffic Flow by Data Flow
342/14/97 A	Norbert Fröhlich, Rolf Schlagenhaft, Josef Fleischmann: Partitioning VLSI-Circuits for Parallel Simulation on Transistor Level
342/15/97 A	Frank Weimer: DaViT: Ein System zur interaktiven Ausführung und zur Visualisierung von INSEL-Programmen
342/16/97 A	Niels Reimer, Jürgen Rudolph, Katharina Spies: Von FOCUS nach INSEL - Eine Aufzugssteuerung
342/17/97 A	Radu Grosu, Ketil Stølen, Manfred Broy: A Denotational Model for Mobile Point-to-Point Data-flow Networks with Channel Sharing
342/18/97 A	Christian Röder, Georg Stellner: Design of Load Management for Parallel Applications in Networks of Heterogenous Workstations
342/19/97 A	Frank Wallner: Model Checking LTL Using Net Unfoldings

Reihe A

342/20/97 A	Andreas Wolf, Andreas Kmoch: Einsatz eines automatischen Theorembeweisers in einer taktikgesteuerten Beweisumgebung zur Lösung eines Beispiels aus der Hardware-Verifikation – Fallstudie –
342/21/97 A	Andreas Wolf, Marc Fuchs: Cooperative Parallel Automated Theorem Proving
342/22/97 A	T. Ludwig, R. Wismüller, V. Sunderam, A. Bode: OMIS - On-line Monitoring Interface Specification (Version 2.0)
342/23/97 A	Stephan Merkel: Verification of Fault Tolerant Algorithms Using PEP
342/24/97 A	Manfred Broy, Max Breitling, Bernhard Schätz, Katharina Spies: Summary of Case Studies in Focus - Part II
342/25/97 A	Michael Jaedicke, Bernhard Mitschang: A Framework for Parallel Processing of Aggregat and Scalar Functions in Object-Relational DBMS
342/26/97 A	Marc Fuchs: Similarity-Based Lemma Generation with Lemma-Delaying Tableau Enumeration
342/27/97 A	Max Breitling: Formalizing and Verifying TimeWarp with FOCUS
342/28/97 A	Peter Jakobi, Andreas Wolf: DBFW: A Simple DataBase FrameWork for the Evaluation and Maintenance of Automated Theorem Prover Data (incl. Documentation)
342/29/97 A	Radu Grosu, Ketil Stølen: Compositional Specification of Mobile Systems
342/01/98 A	A. Bode, A. Ganz, C. Gold, S. Petri, N. Reimer, B. Schiemann, T. Schnekenburger (Herausgeber): "'Anwendungsbezogene Lastverteilung"', ALV'98
342/02/98 A	Ursula Hinkel: Home Shopping - Die Spezifikation einer Kommunikationsanwendung in FOCUS
342/03/98 A	Katharina Spies: Eine Methode zur formalen Modellierung von Betriebssystemkonzepten
342/04/98 A	Stefan Bischof, Ernst-W. Mayr: On-Line Scheduling of Parallel Jobs with Runtime Restrictions
342/05/98 A	St. Bischof, R. Ebner, Th. Erlebach: Load Balancing for Problems with Good Bisectors and Applications in Finite Element Simulations: Worst-case Analysis and Practical Results
342/06/98 A	Giannis Bozas, Susanne Kober: Logging and Crash Recovery in Shared-Disk Database Systems
342/07/98 A	Markus Pizka: Distributed Virtual Address Space Management in the MoDiS-OS
342/08/98 A	Niels Reimer: Strategien für ein verteiltes Last- und Ressourcenmanagement
342/09/98 A	Javier Esparza, Editor: Proceedings of INFINITY'98
342/10/98 A	Richard Mayr: Lossy Counter Machines

SFB 342 : Methoden und Werkzeuge für die Nutzung paralleler Rechnerarchitekturen

Reihe B

342/1/90 B	Wolfgang Reisig: Petri Nets and Algebraic Specifications
342/2/90 B	Jörg Desel: On Abstraction of Nets
342/3/90 B	Jörg Desel: Reduction and Design of Well-behaved Free-choice Systems
342/4/90 B	Franz Abstreiter, Michael Friedrich, Hans-Jürgen Plewan: Das Werkzeug runtime zur Beobachtung verteilter und paralleler Programme
342/1/91 B	Barbara Paech1: Concurrency as a Modality
342/2/91 B	Birgit Kandler, Markus Pawlowski: SAM: Eine Sortier- Toolbox - Anwenderbeschreibung
342/3/91 B	Erwin Loibl, Hans Obermaier, Markus Pawlowski: 2. Workshop über Parallelisierung von Datenbanksystemen
342/4/91 B	Werner Pohlmann: A Limitation of Distributed Simulation Methods
342/5/91 B	Dominik Gomm, Ekkart Kindler: A Weakly Coherent Virtually Shared Memory Scheme: Formal Specification and Analysis
342/6/91 B	Dominik Gomm, Ekkart Kindler: Causality Based Specification and Correctness Proof of a Virtually Shared Memory Scheme
342/7/91 B	W. Reisig: Concurrent Temporal Logic
342/1/92 B	Malte Grosse, Christian B. Suttner: A Parallel Algorithm for Set-of-Support Christian B. Suttner: Parallel Computation of Multiple Sets-of-Support
342/2/92 B	Arndt Bode, Hartmut Wedekind: Parallelrechner: Theorie, Hardware, Software, Anwendungen
342/1/93 B	Max Fuchs: Funktionale Spezifikation einer Geschwindigkeitsregelung
342/2/93 B	Ekkart Kindler: Sicherheits- und Lebendigkeitseigenschaften: Ein Literaturüberblick
342/1/94 B	Andreas Listl; Thomas Schnekenburger; Michael Friedrich: Zum Entwurf eines Prototypen für MIDAS