Technische Universität Chemnitz-Zwickau

Sonderforschungsbereich 393

Numerische Simulation auf massiv parallelen Rechnern

Ulrich Elsner Volker Mehrmann

Frank Milde Rudolf A. Römer Michael Schreiber

The Anderson Model of Localization: A Challenge for Modern Eigenvalue Methods

Preprint SFB393/97-31

Preprint-Reihe des Chemnitzer SFB 393

SFB393/97-31 **December 1997**

The Anderson model of localization: a challenge for modern eigenvalue methods*

Ulrich Elsner[‡] Volker Mehrmann[‡] Frank Milde[§] Rudolf A. Römer[§]
Michael Schreiber[§]

Abstract. We present a comparative study of the application of modern eigenvalue algorithms to an eigenvalue problem arising in quantum physics, namely, the computation of a few interior eigenvalues and their associated eigenvectors for the large, sparse, real, symmetric, and indefinite matrices of the Anderson model of localization. We compare the Lanczos algorithm in the 1987 implementation of Cullum and Willoughby with the implicitly restarted Arnoldi method coupled with polynomial and several shift-and-invert convergence accelerators as well as with a sparse hybrid tridiagonalization method. We demonstrate that for our problem the Lanczos implementation is faster and more memory efficient than the other approaches. This seemingly innocuous problem presents a major challenge for all modern eigenvalue algorithms.

Key words. Eigenvalue, eigenvector problems, Lanczos algorithm, Arnoldi algorithm, Anderson model of localization, shift-and-invert, polynomial convergence accelerators

AMS. 65F15, 65F50, 82B44, 65F10

1 Introduction

In this paper we present a comparative study of the application of modern eigenvalue algorithms to an eigenvalue problem arising in quantum physics. The task is to compute a few (5–10) interior eigenvalues and the associated eigenvectors of a family of structured large, sparse, real, symmetric, indefinite matrices. The off-diagonal elements are equal to the off-diagonal elements of the 7–point central difference approximation to the three-dimensional Poisson equation on the unit cube with periodic boundary conditions. The matrices differ from each other only in the diagonal entries, which are suitably chosen random numbers.

Previously this problem was often solved by using the 1987 Cullum and Willoughby implementation of the Lanczos algorithm [6, 7], in the following called CWI. But in the last 10 years several new eigenvalue methods have been developed and implemented as software packages, that seem, at least at first glance, more appropriate than CWI, see, *e.g.*, the recent survey and comparison given in [19]. We apply these new codes to the described family of matrices and check whether they are faster and more memory efficient than CWI. To our surprise, none of the tested codes is consistently better than CWI. As we show below, we find only a single new code which is at least as fast as CWI. But this code needs two orders of magnitude more memory than CWI. We therefore believe that the described family of matrices will present an important new benchmark example and will hopefully lead to modifications and improvements for the current methods.

*This work was supported by the Deutsche Forschungsgemeinschaft within the Sonderforschungsbereich 393 "Numerische Simulation auf massiv parallelen Rechnern".

‡Fakultät für Mathematik, Technische Universität, D-09107 Chemnitz, Germany
§Institut für Physik, Technische Universität, D-09107 Chemnitz, Germany

The paper is organized as follows. In § 2 we describe the underlying quantum physics problem, *i.e.*, the Anderson model of localization, and introduce the parameters used in our study. In § 3 we briefly review the Cullum/Willoughby version of the Lanczos method that has been previously used in the simulations for this model. We then give in § 4 a brief survey of more recent eigenvalue methods. In § 5 we present comparative results for the different methods and show that CWI is faster and needs less memory than all other approaches.

2 The Anderson model of localization

The Anderson model of localization [1] is a convenient model for the investigation of electronic properties of disordered systems. Although it represents a severe simplification of amorphous materials and alloys, it has nevertheless become a paradigmatic model and is currently widely used in the theoretical description of quantum mechanical effects of disorder such as, *e.g.*, spatial localization of electronic wave functions with increasing strength of disorder and the corresponding metal-insulator transitions [18, 21, 35]. The quantum mechanical problem is represented by a Hamilton operator in the form of a real symmetric matrix A and the quantum mechanical wave functions are simply the eigenvectors of A, *i.e.*, finite vectors x with real entries. *E.g.*, for a simple cubic lattice with $M = N \times N \times N$ sites, we have to solve the eigenvalue equation $Ax = \lambda x$, which is given in site representation as

$$x_{i-1,j,k} + x_{i+1,j,k} + x_{i,j-1,k} + x_{i,j+1,k} + x_{i,j,k-1} + x_{i,j,k+1} + \varepsilon_{i,j,k} x_{i,j,k} \\ = \lambda x_{i,j,k}, \tag{1}$$

with i, j, k denoting the cartesian coordinates of a site. The off–diagonal entries of A correspond to hopping probabilities of the electrons from one site to a neighboring site. For simplicity, we have set them all to unity in (1). The disorder is encoded in the random potential site energies $\varepsilon_{i,j,k}$ on the diagonal of the matrix A. We consider only the case of $\varepsilon_{i,j,k}$ being uniformly distributed in the interval $[-w/2, +w/2]$. This is a common simplification, usually used in the studies of the Anderson model of localization with typical values of w ranging from 1 to 30. The boundary conditions are usually taken to be periodic, but hard wall and helical [28] boundary conditions are sometimes also used. According to the Gersgorin circle theorem [13] every such matrix A has eigenvalues in the interval $[-w/2 - 6, +w/2 + 6]$. Possible generalizations of the Anderson model include anisotropic [21] or even random hopping [9] and various choices of the distribution function of the site energies [14]. However, the graph of the matrix remains the same.

Although the above matrix seems to be fairly simple, the intrinsic physics is surprisingly rich. For small disorder ($w \ll 16.5$), the eigenvectors are extended, *i.e.*, $x_{i,j,k}$ is fluctuating from site to site but the envelope $|x|$ is approximately a non-zero constant. For large disorder ($w \gg 16.5$), all eigenvectors are localized, *i.e.*, the envelope $|x_n|$ of the nth eigenstate may be approximately written as $\exp[-|\vec{r} - \vec{r}_n|/l_n(w)]$ with $\vec{r} = (i, j, k)^T$ and $l_n(w)$ denoting the localization length of the eigenstate at the specified strength w of the disorder. In Fig. 1, we show examples of such states for the Anderson model in one spatial dimension. Since extended states can contribute to electron transport, whereas localized states cannot, the Anderson model thus describes a metal-to-insulator transition: In three-dimensional samples at $w = w_c \approx 16.5$, the extended states at $\lambda \approx 0$ vanish and no current can flow. The eigenvector properties are also connected with the statistical properties of the spectrum $\sigma(A)$ of A. In the extended regime one finds level repulsion, while in the localized regime the eigenvalues are uncorrelated resulting in level clustering. These results agree quantitatively with random matrix theory [18]. Directly at w_c there is a so-called critical regime where the eigenvectors are multifractal entities [21, 29] showing characteristic fluctuations of the

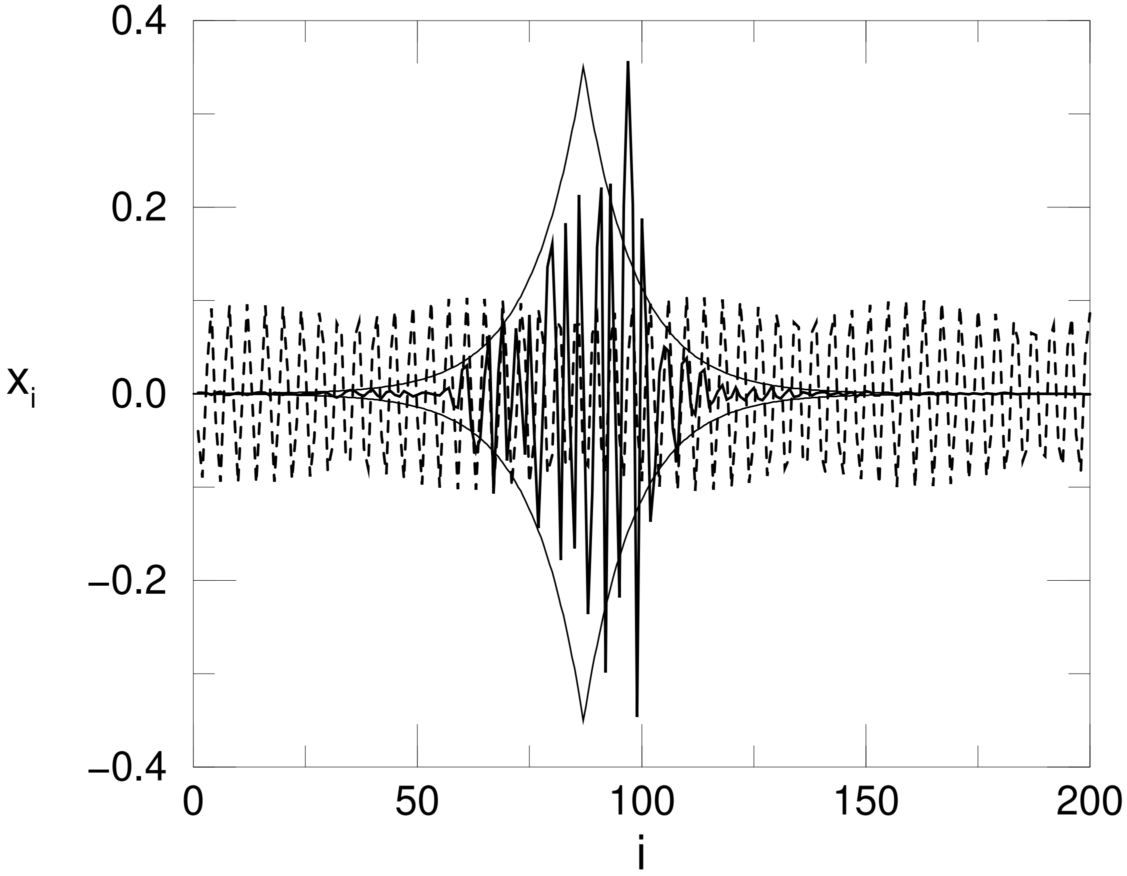

Figure 1: Extended (dashed line) and localized (thick solid line) eigenstate for a single realization of the Anderson model in one spatial dimension with $N = 200$ sites and periodic boundary conditions. For the localized eigenstate, we also show the exponential envelope with localization length $l \approx 12$ (thin lines) according to § 2.

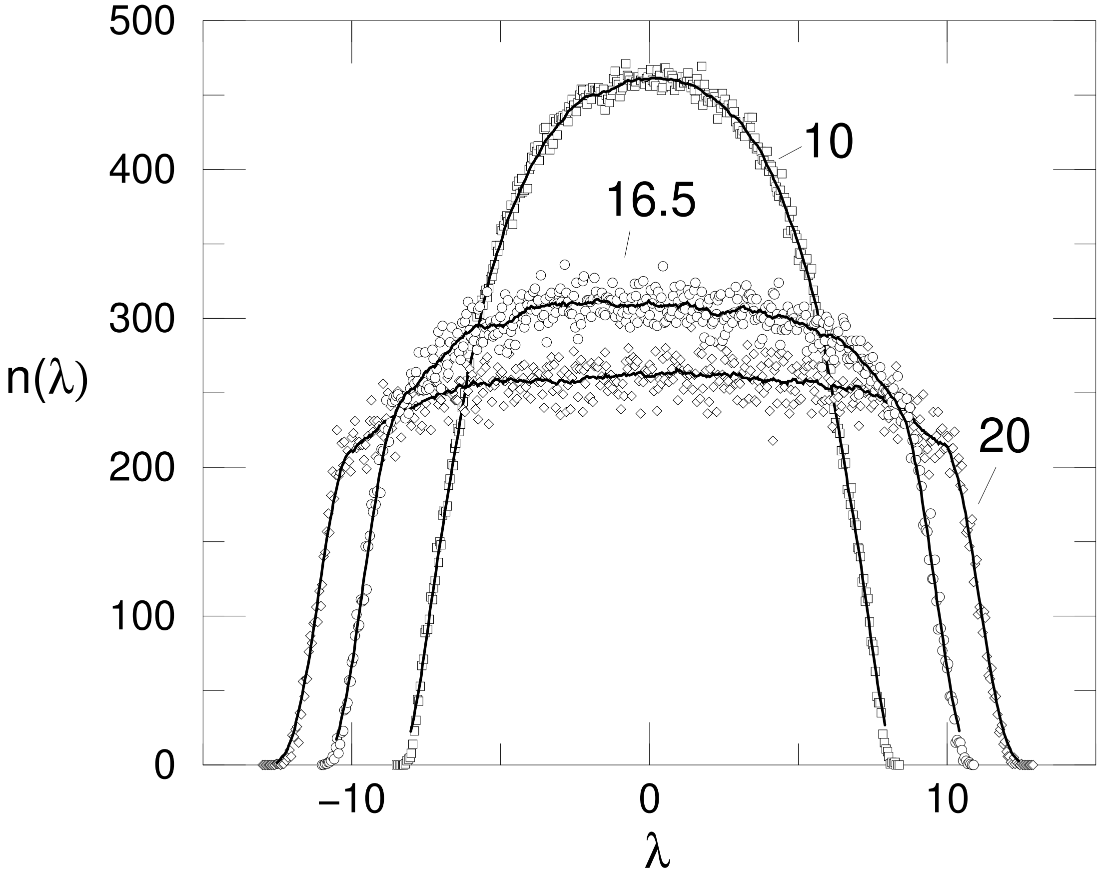

Figure 2: Histogram $n(\lambda)$ of eigenvalues for a single system with $N^3 = 48^3$ sites and $w = 10$ ($\square$), 16.5 ($\circ$), and 20 ($\diamond$). The bin width is 0.05. The lines are obtained by consecutively averaging 20 bins.

amplitude on all length scales. In order to numerically distinguish these three regimes, namely localized, critical and extended behavior, one needs to (i) go to extremely large system sizes and (ii) average over many different realizations of the disorder, *i.e.*, compute eigenvalues or -vectors for many matrices with different diagonals.

In the present paper we concentrate on the computation of a few eigenvalues and corresponding eigenvectors for the physically most interesting case of critical disorder w_c and in the center of $\sigma(A)$, *i.e.*, at $\lambda = 0$, for system sizes as large as possible. In Fig. 2, we show a histogram of $\sigma(A)$ for different disorders. Note the high density of states at $\lambda = 0$ in all cases. Therefore we have the further numerical challenge of distinguishing clearly the eigenstates in this high density region.

3 The Lanczos algorithm and the Cullum/Willoughby implementation

As outlined in the last section, each of the matrices A is sparse, symmetric and indefinite. Furthermore, the matrix-vector multiplication Ax can be written explicitly as in (1) and is thus easily implemented. An ideal candidate for an algorithm taking advantage of nearly all these properties is the Lanczos algorithm [13]. This algorithm iteratively generates a sequence of orthogonal vectors v_i, $i = 1, \ldots, K$, such that $V_K^T A V_K = T_K$, with $V =$

$\{v_1, v_2, \cdots, v_K\}$ and T_K a symmetric tridiagonal $K \times K$ matrix. One obtains the recursion

$$\beta_{i+1} v_{i+1} = A v_i - \alpha_i v_i - \beta_i v_{i-1}, \tag{2}$$

where $\alpha_i = v_i^T A v_i$ and $\beta_{i+1} = v_{i+1} A v_i$ are the diagonal and subdiagonal entries of T_K, $v_0 = 0$ and v_1 is an arbitrary starting vector. For $K = M$ in exact arithmetic this is an orthogonal transformation to tridiagonal form that needs M matrix-vector multiplications. The eigenvalues of the tridiagonal matrix T_K, also known as Ritz values, are then simply the eigenvalues of the matrix A and the associated Ritz vectors yield the eigenvectors [6, 7, 13, 22, 25].

In finite precision arithmetic, however, the Lanczos vectors v_i loose their orthogonality after a small number of Lanczos iterations. Consequently, there appear so called "spurious" or "ghost" eigenvalues in $\sigma(T_K)$, which do not belong to $\sigma(A)$.

There are several solutions to this problem: total reorthogonalization of all Lanczos vectors against each other, selective reorthogonalization [23], or distinguishing between good and spurious eigenvalues. While the reorthogonalization leads to an increase in memory requirements and computing time, since all or several of the v_i need to be stored and reorthogonalized, the solution implemented in CWI [7] uses a simple and highly successful procedure to identify the spurious eigenvalues, thereby avoiding reorthogonalization. It thus only uses two Lanczos vectors in each iteration step and consequently the memory requirements are very small. An eigenvalue of T_K is identified as being spurious if it is also an eigenvalue of the matrix T'_K which is constructed by deleting the first row and column of T_K. Still, the good eigenvalues produced may not yet have converged properly for a given K. So we further use the fact that good eigenvalues will be replicated in $\sigma(T_K)$ if K is large enough. We only accept eigenvalues as being good eigenvalues after they have been replicated at least once in $\sigma(T_K)$. Hence we usually need at least $K \geq 2M$. Finally, in order to obtain the eigenvectors corresponding to these good eigenvalues of A, all Lanczos vectors must be computed a second time, again doubling the computational effort.

The convergence of the Lanczos algorithm is very fast for the eigenvalues close to $\min \sigma(A)$ and $\max \sigma(A)$. This is especially true if these eigenvalues are well separated. However, for eigenvalues in the interior of $\sigma(A)$ and for eigenvalues which are not well separated the convergence is slow. Furthermore, the tridiagonal matrix T_K becomes very large for an iteration in the interior of $\sigma(A)$. Nevertheless, the CWI has been used to study the Anderson model of localization even at $\lambda = 0$ successfully for years [27, 29, 18, 14, 21, 35] and eigenstates for matrices with $N = 100$ can be obtained within a few weeks of computing time [34].

We also remark that most of the computational effort in the Lanczos algorithm is spent on the iteration of (2), *i.e.*, on matrix-vector multiplications and vector additions. These can be easily parallelized and thus the CWI is well suited for parallel architectures. For example, the eigenspectra presented in Fig. 2 have been obtained by such a parallel version of CWI running for about 60 hours for each realization using 16 processors of a Parsytec GCC Power Plus.

4 Modern approaches

Lately there has been much progress in eigenvalue methods mostly concentrating on non-symmetric matrices. We refer to [19] for a recent survey. The symmetric problem is usually assumed to be taken care of implicitly. But although our symmetric eigenvalue problem is well-conditioned [13], the fact that the eigenvalues are clustered in the neighborhood of $\lambda = 0$, our region of interest, creates difficulties for all numerical methods. Promising choices for possible replacements of the Cullum and Willoughby approach are the implicitly restarted Arnoldi method [19] and the hybrid tridiagonalization (HTD) algorithm of Cavers [4]. Another new approach is the Jacobi-Davidson method [3]. In the following we will pay special attention to the Arnoldi approach, since it allows the easy use of the shift-and-invert technique. We expect this to overcome the above mentioned clustering problem at $\lambda = 0$.

4.1 Modifying the eigenproblem

The problem of slow convergence in the interior of $\sigma(A)$ can be overcome by computing eigenvalues and eigenvectors for a modified eigenvalue problem $f(A)x = f(\lambda)x$. The function f is chosen such that the desired point λ in σ is mapped onto or close to the minimum or maximum of $\sigma(f)$. Furthermore, one should choose f, such that $\sigma(f)$ has well separated eigenvalues at min $\sigma(f)$ and max $\sigma(f)$.

Among the many possible choices for $f(A)$, we shall consider in the following: (i) *polynomial convergence accelerators*, where $f(A)$ is chosen as a polynomial which has its maximum (or minimum) at λ. This moves λ to max $\sigma(f(A))$ (or min $\sigma(f(A))$ resp.). We remark that occasionally these convergence accelerators are somewhat misleadingly called preconditioners. (ii) *shift-and-invert* with $f(A) = (A - \lambda I)^{-1}$ and I the $M \times M$ identity matrix. This choice of f requires the additional solution of a linear system with $A - \lambda I$ in each step of the eigenvalue iteration [13, 25]. For the solution of this linear system there are again two alternatives: (ii.a) *direct sparse solvers* for $A - \lambda I$. Unfortunately, not many direct solvers exist which can make efficient use of the sparseness for indefinite problems. (ii.b) *iterative solvers* using only matrix-vector multiplications with $A - \lambda I$. The iterative methods promise to make large matrix sizes possible, since they benefit in an optimal way from sparsity. Memory requirements and computational cost of a matrix-vector multiplication are proportional to the number of non-zeros and therefore proportional to M for our present problem. However, we note that in our case $(A - \lambda I)$ is indefinite which will lead to slow convergence for most iterative solvers. Since the convergence of the iterative solver is dominated by the condition number of the linear system, one may employ preconditioners to accelerate its convergence [13].

All these approaches result in a competition between smaller numbers of Lanczos or Arnoldi iterations and increased costs for each such iteration step. For this reason it is not a priori clear whether they will indeed give a net reduction in computation time.

4.2 The implicitly restarted Arnoldi method

In a recent comparison [19] of different Arnoldi based packages ArPack [20] was found to be the fastest and most reliable of the compared codes.

When applied to symmetric eigenvalue problems, the main difference between the techniques in ArPack based on the Arnoldi iteration and the Lanczos iteration is an implicit restart technique. The Arnoldi method stores a number of Ritz vectors produced by the iterations and after a small number of steps initiates a restart which uses an implicit QR–algorithm for the small eigenvalue problem to create a new starting vector and to maintain orthogonality among the Ritz vectors. In contrast to the Lanczos algorithm more vectors have to be stored but spurious eigenvalues are avoided.

The ArPack implementation further allows the easy use of additional acceleration methods such as polynomial convergence acceleration and shift-and-invert as outlined above. So ArPack is probably the best choice for a replacement of CWI.

4.3 Other approaches

We have also studied the use of the HTD method [4] and the Jacobi/Davidson method [3]. The HTD method is a direct tridiagonalization method specifically designed for sparse matrices. This makes it an interesting approach for our purposes. However, it is not explicitly designed to compute only few interior eigenvalues and associated eigenvectors.

The Jacobi/Davidson method appears to be another promising future direction if it can be properly accelerated. The current version is designed for complex unsymmetric problems and to report comparative results would not be fair to this interesting new development. We intend to further study this method in the future.

5 Results

After a short discussion of the specific implementations and parameters, we now present the results of our comparison. The tests are performed on Hewlett-Packard HP9000 735/125 workstations for $N^3 \leq 24^3$ and on a HP9000 K460 with the fast PA8000 processor for $N^3 \geq 24^3$. The latter machine allows us to use up to 1.9 GB RAM and is about 3.5 times faster. In order to obtain a fair comparison we always require that the eigenresidual of the computed eigenvalue/eigenvector pair satisfies $|Ax - \lambda x| \leq 10^{-8}$. The CPU times have been measured using the UNIX `time` command of the `tcsh` shell. Even for the largest system sizes considered, we have usually taken at least 5 different realizations of disorder and averaged the resulting CPU times. Sometimes, when the CPU times for a given algorithm fluctuate widely, we report the range of times instead of a simple average. We remark that the use of `time` introduces a further uncertainty into the results such that we always have an error of about 10%. The random number generators used are `ran2` from [24] and the `rand` command from MATLAB.

5.1 The standard approach

For our particular problem we can reach $M = 80^3 = 512000$ with CPU times of about two weeks on the K460 machine using CWI. However, keeping in mind the configurational averaging necessitated by the underlying physical problem, a reasonable upper limit for the matrix size is $M = 50^3 = 125000$.

In Table 1 we show the CPU times obtained for CWI in the center and at the edge of $\sigma(A)$. Note that the computing times are nearly independent of the disorder parameter W, but, as expected, CWI is much faster at the edges of $\sigma(A)$ than at $\lambda = 0$. In Table 2, we show the results for CWI at $\lambda = 0$ in dependence on the matrix size M.

In the ARPACK implementation of Lehoucq et al. [20] one has to set the parameter NCV which is the largest number of basis vectors that will be used in the implicitly restarted Arnoldi process. In normal mode and in the interior of $\sigma(A)$ we find that the actual value of NCV heavily influences computing time as shown in Table 3. This dependence on NCV becomes less pronounced for eigenvalues close to min $\sigma(A)$ and max $\sigma(A)$ as shown in Table 4. However, since we do not know of any strategy to choose NCV optimally, this is a severe restriction of ARPACK. Furthermore, as shown in Table 1, ARPACK, working in normal mode, is much slower than CWI both in the center and at the edge of $\sigma(A)$. It becomes too slow for practical use already at $M = 12^3 = 1728$ as shown in Table 2.

In Tables 1 and 2 we also include CPU times for the HTD method. Note that we only show the CPU times needed to transform A to tridiagonal form. Nevertheless, we find that HTD is much slower than CWI. We remark that when we use CWI to compute

M	W	CWI	ArPack	HTD
1000	10.0	2.4	240 – 2200	22
1000	16.5	2.5	230 – 1400	22
1000	20.0	2.4	140 – 1300	23
1728	10.0	7.9	1700 – 12000	170
1728	16.5	7.8	410 – 12000	170
1728	20.0	7.6	1100 – 20000	160
4096	10.0	43		2600
4096	16.5	40		2500
4096	20.0	40		2500
1000	10.0	0.71	0.78	22
1000	16.5	0.77	0.85	22
1000	20.0	0.80	0.89	23
1728	10.0	0.94	1.5	170
1728	16.5	1.0	1.7	170
1728	20.0	1.1	1.8	160
13824	10.0	9.4	57	
13824	16.5	9.2	57	
13824	20.0	9.3	71	

Table 1: CPU times in seconds to compute 5 eigenvectors with CWI, ArPack, and HTD. The upper part of the table corresponds to eigenvalues in the interior of $\sigma(A)$ at $\lambda \approx 0$, the lower part corresponds to the 5 largest eigenvalues.

M	CWI	CWI+ conv. acc.	ArPack	ArPack+ conv. acc.	HTD
1000	2.5	5.6	230 – 1400	4.9	22
1728	7.8	11	410 –12000	10	170
4096	40	59		66	2500
13824	770	1700		1700	
13824	220			550	
27000	1000			3500	
91125	20000				
110592	35000				

Table 2: CPU times in seconds to compute at $w = 16.5$ the eigenvectors corresponding to the 5 eigenvalues closest to $\lambda = 0$ with CWI, CWI with Chebyshev-polynomial acceleration, ArPack in normal mode, ArPack with Chebyshev-polynomial acceleration and HTD for various matrix sizes M. The CPU times in the upper (lower) part of the table have been measured on the HP 735 (HP K460).

	NCV		
	90	110	130
1	2577	4337	9455
2	2011	2267	3455
3	4190	1935	3635
4	411	755	1204
5	2435	1811	11620
6	8188	4704	2506

Table 3: CPU times in seconds to compute 5 eigenvectors with ArPack for 6 different diagonals and 3 choices of NCV for $M = 1728$ and $w = 16.5$ at $\lambda = 0$.

	NCV					
	$w = 10$		$w = 16.5$		$w = 20$	
	15	20	15	20	15	20
1	41	44	63	62	56	58
2	65	65	43	47	49	55
3	57	60	54	64	44	47
4	62	62	44	49	49	55
5	72	76	72	68	76	71
6	44	45	56	62	87	85
7	81	70	72	88	108	90
8	52	60	49	51	70	66
9	42	48	58	68	99	75

Table 4: CPU times in seconds to compute the eigenvectors corresponding to the 5 largest eigenvalues of A with ArPack for 9 different diagonals and 2 choices of NCV with $M = 13824$.

the full spectrum as in Fig. 2, it is still faster than HTD except for small system sizes $M \leq 12^3 = 1728$.

5.2 Polynomial convergence acceleration

As outlined above, polynomial convergence acceleration is usually a convenient choice to speed up the computation of eigenvalues and -vectors corresponding to a small region of $\sigma(A)$. Here, we test a polynomial provided by D. Sorensen, one of the authors of ArPack, and C. Sun [31]. It is based on a Chebyshev-type polynomial given by the following recursion:

$$\begin{aligned} p_1(x) &= 1 \\ p_2(x) &= a + bx^2 \\ p_{n+1}(x) &= 2(a + bx^2)p_n - p_{n-1} \end{aligned} \tag{3}$$

where $a = (x_1^2 + x_2^2)/(x_1^2 - x_2^2)$ and $b = 2/(x_2^2 - x_1^2)$. Also, p_n is symmetric with a local maximum $p_n(0) > 1$ at zero, $|p_n(x)| \leq 1$ in the intervals $[x_1, x_2]$ and $[-x_2, -x_1]$, and p_n grows rapidly for $|x| > x_2$ as shown, *e.g.*, in Fig. 3 for $n = 20$. In general, one would like to have x_1 and x_2 chosen automatically in order to obtain a suitable function f as described in § 4. In all our present calculations we use $n = 50$ with $(x_1)^2 = 0.005$ and $(x_2)^2 = 1.1\,[\max\,\sigma(A)]^2$ according to [31]. This polynomial convergence acceleration speeds up ArPack immensely as one can see in Table 2. Since $\lambda = 0$ is now mapped to $\max\,\sigma(p_{50}(A))$, the actual value of NCV is less important. We found NCV $= 50$ to be a good choice to make the execution times faster, although this requires more memory. Still CPU times are about a factor of two larger than for CWI without any convergence acceleration. Unfortunately, CWI itself is not made faster by the use of this accelerator as also shown in Table 2. Although the number of Lanczos vectors needed to achieve convergence is reduced remarkably, the additional computational effort now required for every Lanczos step becomes very large. At the end one needs even slightly more matrix-vector multiplications than without convergence acceleration.

5.3 Shift-and-invert with direct solvers

We now discuss the use of the shift-and-invert mode of ArPack together with a direct solver for the linear system $(A - \lambda I)y = b$. We first note that although our matrix A is

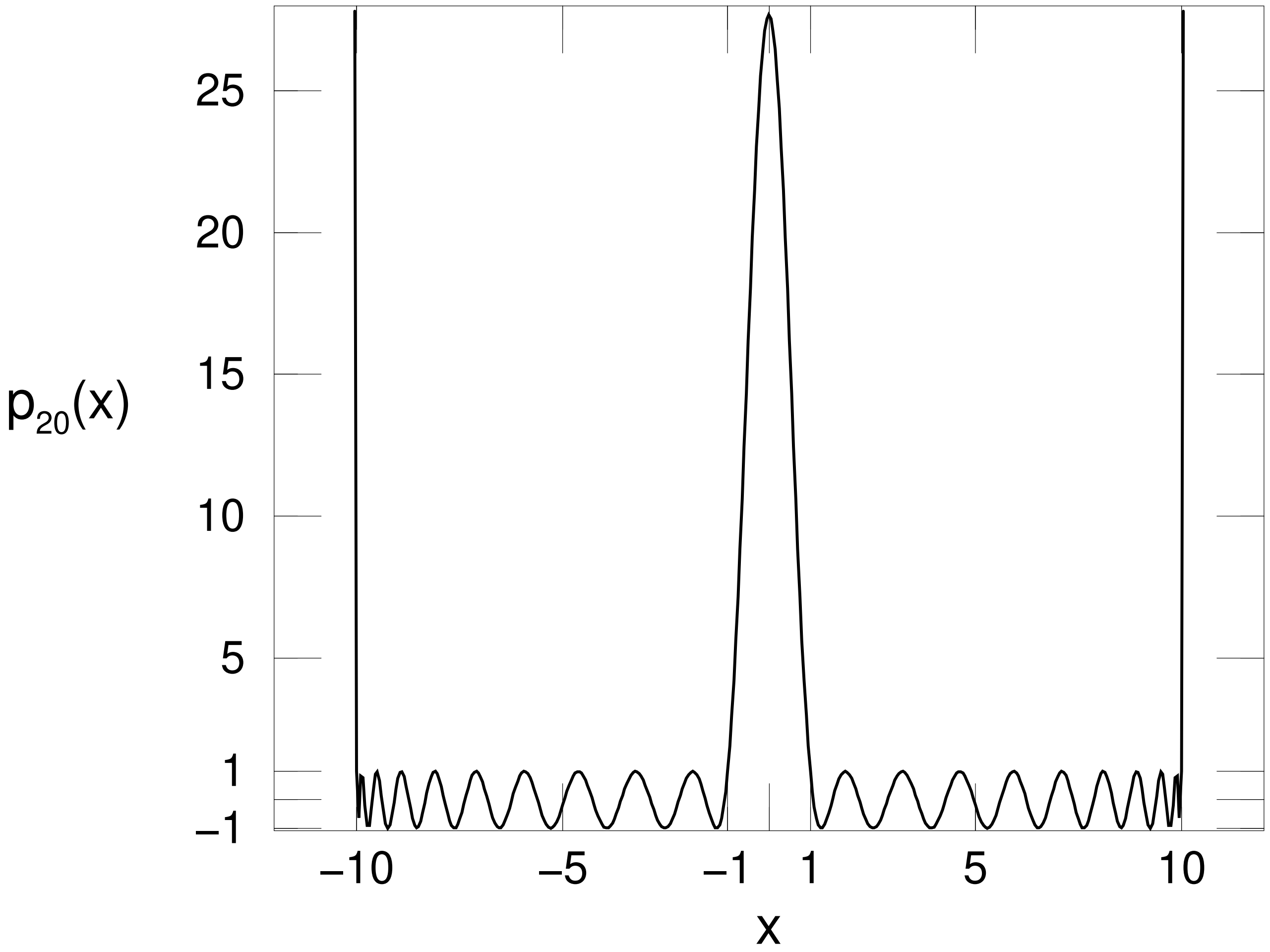

Figure 3: Chebyshev polynomial $p_{20}(x)$ with $x_1 = 1$ and $x_2 = 10$.

M	CWI	ArPack+ conv. acc.	HTD	ArPack+ LU	BKP	SuperLU	MA27	MA27+HB
1000	0.24	0.6	1.4	7.4	10	7.8	0.8	0.4
1728	0.43	1.2	6.7	18	22	19	2.2	0.9
4096	1.0	2.3	13	80	81	92	8.6	2.9
13824	3.4	6.6					70	22
27000	6.5	15					200	68
91125	22						1300	500
110592	27						2600	600

Table 5: Memory requirements in MB to compute at $w = 16.5$ the eigenvectors corresponding to the 5 eigenvalues closest to $\lambda = 0$ for the different diagonalizers (Names as in the text, HB indicates hard wall boundary conditions)

M	CWI	ArPack+ LU	BKP	SuperLU	MA27	MA27+HB
1000	2.5	39	74	8.8	1.3	0.88
1728	7.8	150	300	28	5.0	2.0
4096	40	1200	1900	220	39	9.8
13824	770				740	140
13824	220				260	58
27000	1000				1300	250
91125	20000				19000	4900

Table 6: CPU times in seconds to compute at $w = 16.5$ the eigenvectors corresponding to the 5 eigenvalues closest to $\lambda = 0$ with shift-and-invert ArPack and different direct solvers. For easier comparison, we also include CWI. The CPU times in the upper (lower) part of the table have been measured on the HP 735 (HP K460).

symmetric, it is not positive definite and thus we cannot use a sparse Cholesky decomposition. Unfortunately, there are only few packages available for sparse symmetric indefinite problems [8]. Therefore we also investigated several packages for general sparse matrices.

Meschach [32] is a freely available mathematical package written in C. There are three sparse factorization methods implemented in Meschach: Cholesky, LU, and Bunch-Kaufmann-Parlett (BKP). Cholesky factorization does not work due to the indefiniteness of A. LU and BKP are supposed to take advantage of the sparseness of our problem. However, we find that they have huge memory requirements of the order of M^2 as shown in Table 5. So they are inapplicable for large system sizes. And even for small systems they turn out to be much too slow as shown in Table 6.

The Harwell Subroutine Library [16] contains the sparse symmetric indefinite solver MA27. As shown in Table 6, ArPack with MA27 is about as fast as CWI. In fact, MA27 seems to become faster than CWI for $M \geq 45^3 = 91125$. Unfortunately we could not test this because of the huge memory requirements of MA27 as shown in Table 5.

A noteworthy fact is that MA27 is much better for hard wall boundary conditions (HB). This can be explained by the fact that the bandwidth of the matrix is $\mathcal{O}(N^2)$ instead of $\mathcal{O}(N^3)$ as for periodic boundary conditions. However, on physical grounds, a calculation with HB is expected to be much more influenced by the finite size of the cubes considered. So although we can obtain larger system sizes here, the results for the interesting physical quantities may not be as reliable. Nevertheless, ArPack with MA27 for matrices with HB is faster than CWI for matrices with periodic boundary conditions. MA27 with HB

M	CWI	ArPack+		
		QMRX	QMRL	CPX
1000	2.5	68	93	85
1728	7.8	239	320	330
4096	40	1600		2300
13824	770	35000		
13824	220	12000		
27000	1000			
91125	20000			

Table 7: CPU times in seconds to compute at $w = 16.5$ the eigenvectors corresponding to the eigenvalues closest to $\lambda = 0$ with shift-and-invert ArPack and the iterative solvers from QMRPack. For easier comparison, we also include CWI. The CPU times in the upper (lower) part of the table have been measured on the HP 735 (HP K460).

is also faster than CWI with HB, since for CWI there is only a negligible difference in computing time between HB and periodic boundary conditions coming from the matrix-vector multiplication.

SuperLU is a package by Demmel *et al.* [8] doing a sparse LU decomposition. Compared with CWI and MA27, SuperLU is much slower as shown in Table 6. Furthermore, it needs about one order of magnitude more memory than MA27 as shown in Table 5. SuperLU allows the input of different preorderings in addition to the default minimum-degree ordering. We have tested a symmetric minimum degree ordering from the Matlab program and a nested dissection ordering computed by the Chaco package [17]. For some choices of diagonals we derive small savings in run time and/or memory but these are not consistent, *i.e.*, the same kind of ordering speeds up the program for one choice of N and slows it down for $N + 1$.

5.4 Shift-and-invert with iterative solvers

Considering the recent advances in iterative solvers, we initially hoped that ArPack in shift-and-invert mode coupled with a modern iterative method for the solution of linear systems would be quite efficient. As we will show below, this is not the case.

The quasi-minimal-residual (QMR) technique should be one of the best iterative solvers for symmetric matrices that works using only matrix-vector multiplications if no preconditioning is used [12]. However, as shown in Fig. 2, our matrices are indefinite with a nearly symmetric eigenvalue distribution around zero. This results in a very bad iteration count of about $2M$ for the solution of a single linear system of size M. The times and iteration numbers from three variants implemented in QMRPack [11], namely, QMR based on three-term Lanczos with and without look-ahead (QMRL/QMRX) and QMR based on coupled two-term Lanczos without look-ahead (CPX) are not very different as shown in Table 7. For all three methods the iteration count is rather high. Consequently, we find that ArPack in shift-and-invert mode coupled with QMRPack as iterative solver is about 20 times slower than CWI. The ArPack input parameter NCV was set to 15. We also find that the implementation of QMR based on coupled two-term Lanczos with look-ahead does not converge for larger systems within 50000 iterations.

In order to check if other iterative methods are perhaps more efficient than QMR for our family of matrices, we have also tried several such iterative solvers using the Matlab programming environment. In addition to QMR, we have considered the conjugate-gradient-squared method (CGS) [30], the BiConjugate-Gradient method (BiCG) [10], its stabilized

M	QMR	CGS	Bi-CG	Bi-CGSTAB
512	796 – 999	843 – $n.c.$	809 – 980	1006 – $n.c.$
1000	1705 – $n.c.$	1762 – $n.c.$	1701 – $n.c.$	$n.c.$
1728	2918 – $n.c.$	$n.c.$	2932 – $n.c.$	$n.c.$
2744	4809 – $n.c.$	$n.c.$	4775 – $n.c.$	$n.c.$
4096	7270 – $n.c.$	$n.c.$	7401 – $n.c.$	$n.c.$

M	GMRES	GMRES($5N$)	QMR+jac	QMR+tri
512	511– 512	$n.c.$	813 – $n.c.$	295 – 569
1000	995–1000	$n.c.$	$n.c.$	1389 – $n.c.$
1728	1723–1728	$n.c.$	$n.c.$	$n.c.$
2744	2736–2744	$n.c.$	$n.c.$	$n.c.$
4096		$n.c.$	$n.c.$	$n.c.$

Table 8: Number of iterations needed in Matlab in order to solve the linear system $Ay = b$. The abbreviations for the different algorithms are explained in the text. The runs are aborted when the number of iterations is more than $2M$. This case of no convergence is indicated by "n.c.".

variant (Bi-CGSTAB) [33] and the generalized-minimal-residual (GMRES(k)) method [26]. Furthermore, several general purpose preconditioners [2], *i.e.*, the Jacobi (jac) preconditioner, the ILU(0) preconditioner and also the three main diagonals as the preconditioning matrix (tri) have been tested. Since the performance of Matlab programs cannot directly be compared to compiled programs, we only give the iteration count of each algorithm. One such iteration requires at least one matrix-vector multiplication and two inner products and is thus at least as expensive as one Lanczos step. We always use the built-in implementations of these algorithms as in Matlab v5.1.

In Table 8, we show results obtained for various matrix sizes M. The ranges reflect the variations corresponding to 12 different realizations of disorder on the diagonal of the matrices. Note that for the same M we use the same 12 diagonals for all algorithms. We always choose $x_0 = 0$ as initial vector. The iteration count represents the number of iterations needed to solve the matrix equation $Ax = y$ up to a relative accuracy of 10^{-8}. We always stop the algorithms if after $2M$ iterations this accuracy has not been achieved. For practical restart values $k \leq 200$, GMRES(k) does not converge at all within our iteration limit. With no restarts, GMRES needed M or slightly less iterations. But note that both memory and computing-time requirements for M steps of pure GMRES exceed those of a non-sparse direct solver.

None of the tested preconditioners is consistently effective. The Jacobi preconditioner in fact increases the iteration count most of the time. The ILU(0) preconditioner returns a singular matrix and consequently appears inapplicable. The tridiagonal preconditioner is more effective, in some cases reducing the iteration count by up to 50%. But again there are examples where it fails to do anything. We remark that in general the iteration count is consistent with the results from QMRPack. To sum up, we find that all of these iterative algorithms do not perform better that the QMR algorithm and consequently are no real alternative.

Another idea is to work with the matrix A^2 instead of A. Since it is symmetric and positive definite, we can now use the conjugate gradient method. But this squares the condition number of the linear system, which is already usually very large for A [15]. Hence more effort has to be invested into the development of a good preconditioner. We find for our matrices that while the iteration count is in general a bit less than for the methods mentioned above and the preconditioners are more consistently effective we still need of the order of M steps with at least two matrix-vector multiplications for the solution of one linear system. And since the shift-and-invert ArPack still needs to solve several linear systems, all the iterative methods working on A^2 were no match for CWI.

6 Summary

We have tested several modern methods to compute a few inner eigenvectors of a very large sparse matrix corresponding to the Anderson model of localization motivated within theoretical physics. Particularly the implicitly restarted Arnoldi method in connection with polynomial convergence acceleration and in shift-and-invert mode with several direct and iterative solvers for systems of linear equations is compared to the Cullum/Willoughby implementation of the Lanczos method. Despite the recent progress in linear system solvers we find all considered modern methods to be inapplicable for very large system sizes, because either the computation times or the memory requirements are much to large. To sum up, we find that CWI Lanczos is currently still the most efficient method for the matrix type we are interested in. We emphasize that the CWI Lanczos, with our slight modifications as outlined in § 3, is a reliable tool for our problem. In particular, the problem of spurious eigenvalues which plague the original Lanczos algorithm, can be handled safely.

Since large scale diagonalizations are widely used in theoretical physics — and also theoretical chemistry [5] — we would be happy to learn about any algorithm that does better than CWI for our matrices. We are especially interested in a preconditioner for the iterative methods which is suitably adapted to our problem. Certainly improved direct methods for our matrix type are also of great importance. We hope to have convinced the reader that it may be worthwhile to rethink seemingly easy problems like the present eigenproblem for real and symmetric matrices.

Acknowledgments

We thank Iain Duff for supplying us with Harwell's MA27 routine and Dan Sorensen and Chao Sun for their polynomial convergence accelerator.

References

[1] P. W. Anderson, *Absence of diffusion in certain random lattices*, Phys. Rev., 109 (1958), pp. 1492–1505.

[2] R. Barrett, M. Berry, T. Chan, J. Demmel, J. Donato, J. Dongarra, V. Eijkhout, R. Pozo, C. Romine, and H. van der Vorst, *Templates for the solution of linear systems*, SIAM Publications, Philadelphia, 1994.

[3] J. G. L. Booten, H. A. Van der Vorst, P. M. Meijer, and H. J. J. te Riele, *A preconditioned Jacobi-Davidson method for solving large generalized eigenvalue problems*, Tech. Report CWI-NM-R9414, Centrum voor Wiskunde en Informatica, Amsterdam, July 1994.

[4] I. A. Cavers, *A hybrid tridiagonalization algorithm for symmetric sparse matrices*, SIAM J. Matrix Anal. Appl., 15 (1994), pp. 1363–1380.

[5] R. Chen and H. Guo, *Benchmark calculations of bound states of HO_2 via basis Lanczos algorithm*, Chem. Phys. Lett., 277 (1997), pp. 191–198.

[6] J. Cullum and R. A. Willoughby, *Lanczos Algorithms for Large Symmetric Eigenvalue Computations, Volume 1: Theory*, Birkhäuser, Boston, 1985.

[7] ——, *Lanczos Algorithms for Large Symmetric Eigenvalue Computations, Volume 2: Programs*, Birkhäuser, Boston, 1985. http://www.netlib.org/lanczos/.

[8] J. Demmel, J. Gilbert, and X. Li, *SuperLU Users' Guide*, 1997. http://www.netlib.org/scalapack/prototype/superlu_ug.ps.

[9] A. Eilmes, R. A. Römer, and M. Schreiber, *The Anderson model of localization with random hopping*, Eur. Phys. J. B, 1 (1998). accepted for publication.

[10] R. Fletcher, *Conjugate gradient methods for indefinite systems*, in Proc. Dundee Biennal Conf. Numerical Analysis 1974, G. Watson, ed., Springer, Heidelberg, 1975, pp. 73–89.

[11] R. Freund and N. Nachtigal, *QMRPACK*. http://www.netlib.org/linalg/.

[12] ——, *QMR: A quasi-minimal residual method for non-hermitian linear systems*, Numer. Math., 60 (1991), pp. 315–339.

[13] G. H. Golub and C. F. v. Loan, *Matrix Computations*, Johns Hopkins University Press, Baltimore and London, 3rd ed., 1996.

[14] H. Grussbach and M. Schreiber, *Determination of the mobility edge in the Anderson model of localization in three dimensions by multifractal analysis*, Phys. Rev. B, 57 (1995), pp. 663–666.

[15] W. Hackbusch, *Iterative Solution of Large Sparse Systems of Equations*, vol. 95 of Applied Mathematical Sciences, Springer, Berlin, 1994.

[16] *Harwell subroutine library*. http://www.dci.clrc.ac.uk/Activity.asp?HSL.

[17] B. Hendrickson and R. Leland, *The Chaco User's Guide Version 2*, Sandia National Laboratories, Albuquerque, 1995. ftp://ftp.cs.sandia.gov/pub/papers/bahendr/guide.ps.Z.

[18] E. Hofstetter and M. Schreiber, *Statistical properties of the eigenvalue spectrum of the three-dimensional Anderson hamiltonian*, Phys. Rev. B, 48 (1993), pp. 16979–16985.

[19] R. B. Lehoucq and J. A. Scott, *An evaluation of software for computing eigenvalues of sparse nonsymmetric matrices*, Tech. Report MCS-P547-1195, Argonne National Laboratory, Argonne, 1996. ftp://info.mcs.anl.gov/pub/tech_reports/reports/P547.ps.Z. submitted to ACM Trans. Math. Software.

[20] R. B. Lehoucq, D. Sorensen, and C. Yang, ArPack *users guide: Solution of large scale eigenvalue problems by implicitly restarted Arnoldi methods*, tech. report, Rice University, Department of Computational and Applied Mathematics, Houston, 1996. http://www.caam.rice.edu/software/ARPACK/ug.ps.

[21] F. Milde, R. A. Römer, and M. Schreiber, *Multifractal analysis of the metal-insulator transition in anisotropic systems*, Phys. Rev. B, 55 (1997), pp. 9463–9469.

[22] B. N. Parlett, *The Symmetric Eigenvalue Problem*, Prentice-Hall, Englewood Cliffs, 1980.

[23] B. N. Parlett and D. S. Scott, *The Lanczos algorithm with selective orthogonalization*, Math. Comp., 33 (1979), pp. 217–238.

[24] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, *Numerical Recipes in FORTRAN*, Cambridge University Press, Cambridge, 2nd ed., 1992.

[25] Y. Saad, *Numerical Methods for Large Eigenvalue Problems: Theory and Algorithms*, Wiley, New York, 1992.

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

[27] M. Schreiber, *Fractal eigenstates in disordered systems*, Physica A, 167 (1990), pp. 188–198.

[28] M. Schreiber, *Multifractal characteristics of electronic wave functions in disordered systems*, in Computational Physics, K. H. Hoffmann and M. Schreiber, eds., Springer, Berlin, 1996, pp. 147–165.

[29] M. Schreiber and H. Grussbach, *Multifractal wave functions at the Anderson transition*, Phys. Rev. Lett., 67 (1991), pp. 607–610.

[30] P. Sonneveld, *CGS, a fast Lanczos-type solver for nonsymmetric problems*, SIAM J. Sci. Statist. Comput., 12 (1989), pp. 36–52.

[31] D. Sorensen and C. Sun, *private communications.*

[32] D. E. Stewart and Z. Leyk, *Meschach.* `http://www.netlib.org/c/meschach/`.

[33] H. A. Van der Vorst, *Bi-CGSTAB: A fast and smooth converging variant of Bi-CG for the solution of non-symmetric linear systems*, SIAM J. Sci. Statist. Comput., 12 (1992), pp. 613–644.

[34] I. K. Zarekeshev and B. Kramer, *private communications.*

[35] ——, *Asymptotics of universal probability of neighboring level spacings at the Anderson transition*, Phys. Rev. Lett., 79 (1997), pp. 717–720.

Other titles in the SFB393 series:

96-01 V. Mehrmann, H. Xu. Chosing poles so that the single-input pole placement problem is well-conditioned. Januar 1996.

96-02 T. Penzl. Numerical solution of generalized Lyapunov equations. January 1996.

96-03 M. Scherzer, A. Meyer. Zur Berechnung von Spannungs- und Deformationsfeldern an Interface-Ecken im nichtlinearen Deformationsbereich auf Parallelrechnern.
March 1996.

96-04 Th. Frank, E. Wassen. Parallel solution algorithms for Lagrangian simulation of disperse multiphase flows. Proc. of 2nd Int. Symposium on Numerical Methods for Multiphase Flows, ASME Fluids Engineering Division Summer Meeting, July 7-11, 1996, San Diego, CA, USA. June 1996.

96-05 P. Benner, V. Mehrmann, H. Xu. A numerically stable, structure preserving method for computing the eigenvalues of real Hamiltonian or symplectic pencils. April 1996.

96-06 P. Benner, R. Byers, E. Barth. HAMEV and SQRED: Fortran 77 Subroutines for Computing the Eigenvalues of Hamiltonian Matrices Using Van Loans's Square Reduced Method. May 1996.

96-07 W. Rehm (Ed.). Portierbare numerische Simulation auf parallelen Architekturen. April 1996.

96-08 J. Weickert. Navier-Stokes equations as a differential-algebraic system. August 1996.

96-09 R. Byers, C. He, V. Mehrmann. Where is the nearest non-regular pencil? August 1996.

96-10 Th. Apel. A note on anisotropic interpolation error estimates for isoparametric quadrilateral finite elements. November 1996.

96-11 Th. Apel, G. Lube. Anisotropic mesh refinement for singularly perturbed reaction diffusion problems. November 1996.

96-12 B. Heise, M. Jung. Scalability, efficiency, and robustness of parallel multilevel solvers for nonlinear equations. September 1996.

96-13 F. Milde, R. A. Römer, M. Schreiber. Multifractal analysis of the metal-insulator transition in anisotropic systems. October 1996.

96-14 R. Schneider, P. L. Levin, M. Spasojević. Multiscale compression of BEM equations for electrostatic systems. October 1996.

96-15 M. Spasojević, R. Schneider, P. L. Levin. On the creation of sparse Boundary Element matrices for two dimensional electrostatics problems using the orthogonal Haar wavelet. October 1996.

96-16 S. Dahlke, W. Dahmen, R. Hochmuth, R. Schneider. Stable multiscale bases and local error estimation for elliptic problems. October 1996.

96-17 B. H. Kleemann, A. Rathsfeld, R. Schneider. Multiscale methods for Boundary Integral Equations and their application to boundary value problems in scattering theory and geodesy. October 1996.

96-18 U. Reichel. Partitionierung von Finite-Elemente-Netzen. November 1996.

96-19 W. Dahmen, R. Schneider. Composite wavelet bases for operator equations. November 1996.

96-20 R. A. Römer, M. Schreiber. No enhancement of the localization length for two interacting particles in a random potential. December 1996. to appear in: Phys. Rev. Lett., March 1997

96-21 G. Windisch. Two-point boundary value problems with piecewise constant coefficients: weak solution and exact discretization. December 1996.

96-22 M. Jung, S. V. Nepomnyaschikh. Variable preconditioning procedures for elliptic problems. December 1996.

97-01 P. Benner, V. Mehrmann, H. Xu. A new method for computing the stable invariant subspace of a real Hamiltonian matrix or Breaking Van Loan's curse? January 1997.

97-02 B. Benhammouda. Rank-revealing 'top-down' ULV factorizations. January 1997.

97-03 U. Schrader. Convergence of Asynchronous Jacobi-Newton-Iterations. January 1997.

97-04 U.-J. Görke, R. Kreißig. Einflußfaktoren bei der Identifikation von Materialparametern elastisch-plastischer Deformationsgesetze aus inhomogenen Verschiebungsfeldern. March 1997.

97-05 U. Groh. FEM auf irregulären hierarchischen Dreiecksnetzen. March 1997.

97-06 Th. Apel. Interpolation of non-smooth functions on anisotropic finite element meshes. March 1997

97-07 Th. Apel, S. Nicaise. The finite element method with anisotropic mesh grading for elliptic problems in domains with corners and edges.

97-08 L. Grabowsky, Th. Ermer, J. Werner. Nutzung von MPI für parallele FEM-Systeme. March 1997.

97-09 T. Wappler, Th. Vojta, M. Schreiber. Monte-Carlo simulations of the dynamical behavior of the Coulomb glass. March 1997.

97-10 M. Pester. Behandlung gekrümmter Oberflächen in einem 3D-FEM-Programm für Parallelrechner. April 1997.

97-11 G. Globisch, S. V. Nepomnyaschikh. The hierarchical preconditioning having unstructured grids. April 1997.

97-12 R. V. Pai, A. Punnoose, R. A. Römer. The Mott-Anderson transition in the disordered one-dimensional Hubbard model. April 1997.

97-13 M. Thess. Parallel Multilevel Preconditioners for Problems of Thin Smooth Shells. May 1997.

97-14 A. Eilmes, R. A. Römer, M. Schreiber. The two-dimensional Anderson model of localization with random hopping. June 1997.

97-15 M. Jung, J. F. Maitre. Some remarks on the constant in the strengthened C.B.S. inequality: Application to h- and p-hierarchical finite element discretizations of elasticity problems. July 1997.

97-16 G. Kunert. Error estimation for anisotropic tetrahedral and triangular finite element meshes. August 1997.

97-17 L. Grabowsky. MPI-basierte Koppelrandkommunikation und Einfluß der Partitionierung im 3D-Fall. August 1997.

97-18 R. A. Römer, M. Schreiber. Weak delocalization dueto long-range interaction for two electrons in a random potential chain. August 1997.

97-19 A. Eilmes, R. A. Römer, M. Schreiber. Critical behavior in the two-dimensional Anderson model of localization with random hopping. August 1997.

97-20 M. Meisel, A. Meyer. Hierarchically preconditioned parallel CG-solvers with and without coarse-matrix-solvers inside FEAP. September 1997.

97-21 J. X. Zhong, U. Grimm, R. A. Römer, M. Schreiber. Level-Spacing Distributions of Planar Quasiperiodic Tight-Binding Models. October 1997.

97-22 W. Rehm (Ed.). Ausgewählte Beiträge zum 1. Workshop Cluster-Computing. TU Chemnitz, 6./7. November 1997.

97-23 P. Benner, Enrique S. Quintana-Ortí. Solving stable generalized Lyapunov equations with the matrix sign function. October 1997

97-24 T. Penzl. A Multi-Grid Method for Generalized Lyapunow Equations. October 1997

97-25 G. Globisch. The hierarchical preconditioning having unstructured threedimensional grids. December 1997

97-26 G. Ammar, C. Mehl, V. Mehrmann. Schur-like forms for matrix Lie groups, Lie algebras and Jordan algebras. November 1997

97-27 U. Elsner. Graph partitioning - a survey. December 1997.

97-28 W. Dahmen, R. Schneider. Composite Wavelet Bases for Operator Equations. December 1997.

97-29 P. L. Levin, M. Spasojević, R. Schneider. Creation of Sparse Boundary Element Matrices for 2-D and Axi-symmetric Electrostatics Problems Using the Bi-orthogonal Haar Wavelet. December 1997.

97-30 W. Dahmen, R. Schneider. Wavelets on Manifolds I: Construction and Domain Decomposition. December 1997.

The complete list of current and former preprints is available via
`http://www.tu-chemnitz.de/sfb393/preprints.html`.