FORSCHUNGSZENTRUM JÜLICH GmbH
Zentralinstitut für Angewandte Mathematik
D-52425 Jülich, Tel. (02461) 61-6402

Interner Bericht

Subspace Methods for Large Sparse Interior Eigenvalue Problems

Bernhard Steffen

FZJ-ZAM-IB-2001-11

August 2001

(letzte Änderung: 16.08.2001)

Preprint: International Journal of Differential Equations and Applications,
Volume 3 No. 3 2001, pp. 339 - 351

SUBSPACE METHODS FOR LARGE SPARSE INTERIOR EIGENVALUE PROBLEMS

BERNHARD STEFFEN
John von Neumann Institute for Computing
Central Institute for Applied Mathematics
Forschungszentrum Jülich
52425 Jülich   Germany
E-mail: b.steffen@fz-juelich.de

The calculation of a few interior eigenvalues of a matrix has not received much attention in the past, most methods being some spin-off of either the complete eigenvalue calculation or a subspace method designed for the extremal part of the spectrum. The reason for this could be the rather chaotic behaviour of most methods tried. Only 'shift and invert' and polynomial iteration seemed to have a predictable behavior. However, polynomial iteration is reasonably fast only for extremal eigenvalues of a matrix where all eigenvalues are close to a known line, and inverting a large sparse indefinite system is tricky, while any inaccuracy in the inverse carries through to the eigenvector.
By now, subspace methods have been developed to a state where they can be applied with benefit to the calculation of inner eigenpairs (eigenvalues and -vectors). This is achieved by using a combination of improved approximate residual correction (Jacobi-Davidson method) with new methods to extract approximations to inner eigenvectors of a large (dimension n) matrix from a low dimensional (dimension $m \ll \sqrt{n}$) subspace. Suited to the needs of practical applications, the selection of eigenpairs requested may be specified by very different means - an eigenvalue range, closeness of the eigenvectors to a given selection of approximate eigenvectors, or special patterns of the eigenvectors like the number of local extrema of the components. Depending on information generated with little overhead relative to standard subspace computations, the extraction method may be switched between standard Ritz projection, inverse Ritz projection and residual minimization. Tests indicate reliable and predictable convergence, while performance depends heavily on the quality of the approximate inverse applied.
Some applications are in theoretical chemistry and from accelerator design. In these cases, the eigenpairs requested are typically from the lower part of the spectrum, but too far from the end to be solved by conventional subspace methods.

AMS CLASSIFICATION: 65F15, 65F50

1 Introduction

The problem of computing some or all eigenvalues and -vectors of a large matrix appears in a wide range of applications from biology (meta stable states of ecosystems) to mechanical engineering (oscillations of suspension bridges). As eigenvalue problems make up a fair share of the supercomputer usage, there is good reason for analyzing and improving the algorithms and implementations as well as for teaching users to make the most of the method chosen.

The classification of eigenvalue problems distinguishes between different matrix structures (general, hermitian, complex symmetric, sparse, ...), different requirements (eigenvalues only, eigenvectors too, all eigenvalues, only extremal eigenvalues, certain part of spectrum, eigenvector similar to excitation vector, ...) and numerical properties (normal, diagonally dominant, ...). While these are extremely important for issues of efficiency and implementation, there are only three basic principles involved in the solution of eigenvalue problems:

Similarity transformations $A = S^{-1}JS$ converting A into some normal form (usually diagonal or Jordan) where eigenvalues can be extracted immediately and eigenvectors are given by columns of S. S is built up iteratively as a product of simple matrices.

Nonlinear equation methods treat the problem directly as an $(n+1)$-dimensional nonlinear equation, searching for a solution in the neighborhood of $\bar{\lambda}$. Methods in use are ’shift and invert’ and polynomial iteration $x^n = P_n(A)x^{n-1}$, where the polynomials P_n are chosen such that the sequence x^n converges, e.g. to the eigenvector of the largest eigenvalue.

Subspace methods where A is projected onto a low (e.g. m) dimensional subspace $\mathbf{V}$. Instead of looking for $x \in \mathbb{R}^n$, $\lambda \in \mathbb{C}$ with $Ax = \lambda x$ for $x \in \mathbf{V}$, $\lambda \in \mathbb{C}$ some kind of relaxed condition is applied to define an approximate eigenvector, e.g. $Ax - \lambda x \perp y \bigvee y \in \mathbf{V}$ or $Ax - \lambda x$ minimal in $\mathbf{V}$. There is a wide range of methods to construct $\mathbf{V}$ (vector updates) as well as a choice of relaxed conditions to apply (eigenpair extraction) which we will look into in detail.

If only a few eigenpairs are required, one problem is to define which ones. Mathematical treatment has been restricted to searching for a certain range of eigenvalues. This is fine for the convergence considerations, as in the final stages of calculations the eigenvalue approximations will by necessity be sufficiently precise. At the start, an approximation of an eigenvector may be available that allows a definite identification of the eigenvector, e.g. from a similar problem with different material properties or slightly different geometry or from a low resolution calculation, but the precision of the eigenvalue approximation may not be sufficient to define a window containing only a small number of eigenvectors. So the starting procedure needs some new considerations.

2 Update procedures

The update procedure creates $\mathbf{V^{(k+1)}}$ from $\mathbf{V^{(k)}}$ by adding some directions and possibly reducing the dimensions again. The reduction of dimensions is increasing the number of iteration steps, but as the steps get computationally cheaper, there will usually be some gain in computing time. Even if not, the reduction in memory requirement may be helpful. The reduction - sometimes termed restart - usually retains the approximations to the eigenvectors required plus those to neighboring eigenpairs.

2.1 Starting out

Few problems really have to start from scratch, usually a similar problem has been solved before, so a random start is a matter of convenience, not necessity. For extreme eigenpairs, the gain in starting with a crude approximation to an eigenvector instead of random is only moderate. For inner eigenpairs, the extraction procedure is difficult until the eigenvalue approximations have converged into the proper range, so here starting with an approximation to an eigenvector is really helpful.

Another consideration concerns the dimension of $\mathbf{V^0}$. With most update procedures, the updates to an eigenvector approximation contain large components in the direction of neighboring eigenvectors. Therefore, it is efficient to start out small - just enough directions to get the right multiplicity of eigenvectors, not enough to get all eigenvectors wanted - and increase the dimension of the subspace during the process [16,14].

2.2 Krylow space updates of subspace

The simplest sequence of subspaces $\mathbf{V^{(k)}}$ is given by the Krylow construction $\mathbf{V^{(k)}} = span(x_0, Ax_0, \ldots, A^{(k)}x_0)$. This is the basis of the methods of Lanczos and Arnoldi, which are about the best possible for black box solvers for a few extreme eigenvalues. The Krylow space allows a construction of an orthogonal basis via a three term recurrence which is extremely efficient. There are quite a number of computational shortcuts available with this choice of updates, such that the performance is better than an iteration count would suggest. If only eigenvalues are required, they need little storage, while the computation of eigenvectors is either very memory-consuming or needs a second pass. There are stability problems, but those can be dealt with nicely, and efficient implementations are available [4,6,1].

A related choice is $\mathbf{V^{(k+1)}} = span(\,Aw_1{}^{(k)}, \ldots, Aw_m{}^{(k)})$, $w_i{}^{(k)}$ a Basis of $\mathbf{V^{(k)}}$. This has the advantage of a search subspace with constant (small) dimension which reduces memory requirement and enhances stability, but converges only to the largest eigenvalues[13]. Improvements use polynomial acceleration $\mathbf{V^{(k+1)}} = span(\,P(A)w_1{}^{(k)}, \ldots, P(A)w_m{}^{(k)})$ [13,16,14]. The construction of the optimal polynomial needs some information on the spectrum of A, and will be efficient only when the spectrum is a (possibly curved) line. For eigenvalues alone, polynomial acceleration cannot beat the methods of Lanczos and Arnoldi, for eigenvectors it may be faster.

2.3 Approximate inverse updates of subspace

Approximate inverse updates make use of special features of the matrix and can therefore be very efficient if properly implemented. The idea is to define a linearized correction equation of the eigenvalue approximation and utilize a computationally cheap approximation of this equation. With $\lambda_1 = x_i^T A x_i$ and e_i the correct eigenvector, this equation reads $(A - \lambda_1 I)(e_i + q_i) \approx (A - \lambda_1 I)x_i$, which with $\lambda_1 \approx \lambda, \quad (A - \lambda I)e_i = 0$ seems to give a reasonable way to construct improved subspace updates. Let B_{λ_i} be an (easy to compute) approximation to $(A - \lambda_i I)^{-1}$. Add the approximate inverses applied to the residue to the search space: $\mathbf{V^{(k+1)}} = span(\mathbf{V^{(k)}}, B_{\lambda_1} r_1, \ldots, B_{\lambda_m} r_m)$, where $r_i := (A - x_i^T A x_i\ I)x_i$ with x_i the best eigenvector approximations available. An alternative is using $B_{\lambda_1} x_i$ directly, thus approximating 'shift and invert', but this obviously gives almost parallel basis vectors.

Until recently, the only practical method making use of approximate inverses was Davidson's method which simply uses the diagonal entries of A to compute B, and it was useful only for matrices from theoretical chemistry. While convergence was demonstrated to be rather fast, no analysis was available, and attempts to improve it by using better approximate inverses failed. In hindsight, the reason for this is quite clear, and some idea was there right from the start. If B is exact, then $B_{\lambda_i} r_i = x_i$, obviously not a good choice. So B must not be too good. On the other hand, if B is a poor approximation, this is not much better than taking r_i itself, which is the Krylow subspace calculation without the computational shortcuts.

The annoying problem that improving B might reduce convergence was understood and overcome in [15], where it was proved that the exact way to define a correction equation is to project the correction problem into the space orthogonal to e_i, and e_i not being available, the space orthogonal to any approximation e_i, notably x_i to will do fine, too. This leads to the improved definition of q_i: $[(I - x_i\, x_i{}^T)\; B_{\lambda_i}\; (I - x_i\, x_i{}^T)]\; q_i \;=\; r_i$

The projection $(I - x_i\, x_i{}^T)$ is not easy to incorporate into the matrix, but there is no need to do so. A rank-1 update of $B_{\lambda_i}^{-1}\; r_i$ is sufficient. The cost of this update is with band matrix inverse one extra right hand side, one scalar product and one vector update per inverse, with CG inverse one multiplication of preconditioner times vector per inverse plus one scalar product and one vector update per CG-step.

There is no need to use the same type of approximate inverses throughout the computation. In some finite element test cases, the best efficiency has been achieved by starting out with a rather crude and simple choice for B (diagonal only) and getting more accurate as the eigenvector approximations improve[3]. This leaves the field of tuning the algorithm wide open.

Approximate inverses are, strictly speaking, not part of the eigenvalue algorithm but only a plug-in, but of course of highest importance for the efficiency. Therefore a few words on the topic seem appropriate. A standard choice is CG methods (QMR, MINRES, ...). They do the job, but for inner eigenvalues - strongly indefinite problems - may need quite a few inner iterations. Multiscale methods promise to be faster, but suffer from differences in the eigenvalues at different resolution, so that an eigenvalue that is slightly bigger than the one required on the scale of A may be smaller on the coarse scale, causing convergence to slow down or even stall. This can be overcome by projecting the equation on the coarse scale onto a space orthogonal to all nearby eigenpairs of the coarse scale, which again is tricky.

3 Eigenvalue extraction

The goal of the extraction part is to find in a given subspace vectors that are good approximations of eigenvectors of A as well as the corresponding approximations of the eigenvalues. The description is independent of the update method, while the implementation shows strong interdependence.

3.1 Ritz projection

The Ritz projection is the most important approach to extract eigenvalue and -vector approximations from a given subspace. The basic idea[13] is: May $\mathbf{V^{(k)}}$ be a subspace of $\mathbb{R}^n$ at iteration step k with an orthonormal basis $\vec{w}_1^{(k)}, \ldots, \vec{w}_m^{(k)}$ and $W^{(k)}$ the matrix with columns $\vec{w}_j^{(k)}$, $S^{(k)} \;:=\; (W^{(k)})^T\, A\, W^{(k)}$, $\bar{\lambda}_j^{(k)}$ the eigenvalues of $S^{(k)}$, and $T^{(k)}$ a matrix with the eigenvectors of $S^{(k)}$ as columns. The columns $\vec{x}_j^{(k)}$ of $W^{(k)}\, T^{(k)}$ (the Ritz vectors) are approximations to eigenvectors of A with the Ritz values $\bar{\lambda}_j^{(k)} = (\vec{x}_j^{(k)})^T\, A\, \vec{x}_j^{(k)}$ approximating eigenvalues of A. If the subspace allows a good approximation of the extremal eigenvectors of A, the corresponding Ritz vectors will be close to optimal approximations[13].

Example 1: Let A be a diagonal matrix and let $\mathbf{V^{(k)}}$ contain a good approximation to the largest eigenvector of A:

$$A := \begin{bmatrix} -1000 & 0\,0\,0 \\ 0 & 4\,0\,0 \\ 0 & 0\,5\,0 \\ 0 & 0\,0\,6 \end{bmatrix} \qquad \mathbf{V^{(k)}} := \ span\ (\begin{bmatrix} 0.001 & -0.005 \\ 0.1 & -1.0 \\ 0 & 1.0 \\ 1 & 0.001 \end{bmatrix})$$

The Ritz projection will calculate the approximation $[.00066, .0320, .0677, .9972]$ to $[0, 0, 0, 1]$, near optimal, and the Ritz value will be 5.9929.

For hermitian A, the Ritz vectors are forced to be orthogonal, while the projections of the eigenvectors of A onto $\mathbf{V^{(k)}}$ will not be orthogonal. Now, the Ritz vector $\vec{x}\,_1^{(k)}$ to the smallest (largest) Ritz value may be askew to all eigenvectors of A. $\vec{x}\,_2^{(k)}$ will be orthogonal to $\vec{x}\,_1^{(k)}$, therefore even if a non extremal eigenvector of A has a good approximation in $\mathbf{V^{(k)}}$, this may not be orthogonal to $\vec{x}\,_1^{(k)}$ and therefore not be close to a Ritz vector. Therefore, the second eigenvector of A can be expected to be well approximated only if the extremal one has at least a decent approximation, and inner eigenvectors of A may be poorly approximated even if a good approximation is contained in $\mathbf{V^{(k)}}$. This effect is pronounced when there appear numerically multiple Ritz values.

Example 2: Let A be as before and exchange the last two rows in the basis vectors of $\mathbf{V^{(k)}}$, which now contains a good approximation to an inner eigenvector of A. The approximations to $[0, 0, 1, 0]$ calculated by Ritz projection will be $[-.00184, -.4423, .7338, .5156]$, which is almost 43^0 off the desired eigenvector and much inferior to the starting approximation. The Ritz value of 4.92 is almost correct. The situation is not necessarily improved by improving the subspace. Changing the first column of $\mathbf{V^{(k)}}$ to $[0.0001, 0.01, 1, 0]$ does not help.

3.2 Methods for interior eigenvalues

If Ritz projection performs poorly, inner eigenvalues may be approximated from a subspace containing a good approximation of the eigenvector by either of two methods depending on information available. Instead of calculating the projection of A onto $\mathbf{V^{(k)}}$, an inverse projection of $(A - \bar{\lambda}I)^{-1}$ onto $\mathbf{W^{(k)}} := (A - \bar{\lambda}I)\mathbf{V^{(k)}}$ is calculated with only marginally increased effort. Now the formerly interior eigenvalues transform to extremal ones, and if $\bar{\lambda}$ is chosen properly, the corresponding eigenvector approximations (in $\mathbf{W^{(k)}}$) are good. Applying $(A - \bar{\lambda}I)^{-1}$ to these approximations is easy, just a linear combination of the basis vectors in $\mathbf{V^{(k)}}$, and yields good approximations to eigenvectors in the neighborhood of $\bar{\lambda}$ (harmonic Ritz projection, [15,10]). However, $\bar{\lambda}$ must not be an eigenvalue of A, otherwise the projected problem degenerates. With $\bar{\lambda} = 4.9$, the previous example returns $[0.0017, 0.2390, .9605, -0.1429]$, a much better but not optimal approximation. Changing $\bar{\lambda}$ to 4.995 gives $[0.0013, 0.1630, 0.9845, -0.0646]$, quite good. This will work fine when eigenvalues in a well-known range are looked for.

There are, however, problems where the eigenvalues are not known with sufficient accuracy. The required eigenvalues may be specified e.g. from low accuracy computations, from observations, or from certain geometric patterns of zeroes or extrema. In this case, a residual minimization gives better results:

Calculate $\{x_j^{(k)}, \lambda_j\}$ as the local minima of $||(A - \lambda x\ I)x||$ for all $x \in \mathbf{V^{(k)}}$, $||x|| = 1$. From

these, choose the vectors that match the desired properties.
As this is a (low dimensional) nonlinear problem, some approximation is needed. If an approximate eigenvector $\bar{v}$ is known, a simple but almost always sufficient linearization is minimizing $||(A - \bar{v}^T A \bar{v}\ I)(\bar{v} + x)||$ over $x^T \bar{v} = 0$. This yields $[0.00006, -0.0861, .9791, .1840]^T$ for the above example, only slightly less accurate than harmonic Ritz projection with $\bar{\lambda} = 4.995$. Changing $V[1,2]$ to 0.0001 changes the picture, the harmonic Ritz projection will perform only slightly better than simple Ritz projection, while the residual minimization will be nearly perfect.
If no approximate eigenvector is known, a slightly more elaborate procedure is required.

3.3 Combining the methods for interior eigenvalues

The optimal way to handle interior eigenvalues is a combination of the above methods. This can be done at the cost of a some extra calculations in $\mathbb{R}^m$ only. All three methods require similar calculations in $\mathbb{R}^n$ for different basis sets:
Ritz projection needs an orthogonal basis of $\mathbf{V^k}$, harmonic Ritz projection an orthogonal basis of $(A - \bar{\lambda} I)\mathbf{V^k}$, and the residual evaluation requires a combined orthogonal basis of $\mathbf{V^k}$ and $A\mathbf{V^k}$. Instead of actually building these bases in $\mathbb{R}^n$, it is sufficient just to calculate the necessary transformation from a Cholesky decomposition of the scalar product matrix of a basis. Let $V := [v_1, \ldots, v_m]$ be a non-orthogonal basis of $\mathbf{V^k}$, $C^T C = V^T V$, $R = C^{-1}$, then $W = V\, R$ and $R^T\, ((AV)^T\, V)\, R = W^T A W$.
A similar transformation of $(A - \lambda I)v_i$ gives the matrix for the harmonic Ritz projection, and similarly for residual evaluation. What is needed in $\mathbb{R}^n$ are the calculations of $V^T V$, $(AV)^T\, V$ and $(AV)^T A V$. With these, the coefficients of the approximation vectors to the basis $[v_1, \ldots, v_m]$ can be evaluated using matrices and vectors in $\mathbb{R}^m$ only, along with the eigenvalue approximations and the norm of the residual.
The residual minimization needs some further considerations, as standard minimization methods are not as efficient as required. The problem is the starting heuristic; with a good starting point, a single linear search will give the approximation vector with sufficient accuracy.
Obviously, a small residue is possible only if the vector is close to a Ritz vector or to the subspace spanned by a cluster of Ritz vectors. Therefore, Ritz vectors are calculated first. A cluster of Ritz vectors may either be an approximation to a cluster of eigenvectors of A or a combination of an approximation to a subspace spanned by eigenvectors plus some spurious vectors giving the same Ritz value but forming a linear combination from eigenvectors of A to larger and smaller values as in the second example above. In this case, neither vector gives a good starting point for the residual minimization, but a harmonic Ritz projection applied only to the Ritz vectors in the cluster will show which case is present and isolate all those directions in the cluster that can be used as starting point for residual minimization. By this way, all useful local minima of the residue can be found with reasonable effort and then screened for the conditions describing the desired eigenvalues. In tests even with an eigenvalue range as selection criterion this procedure gave better performance than Sharmonic Ritz projection alone[7].

4 Problems of implementation and parallelization

The eigenvalue computation can be separated into actions in the n-dimensional space and those in the projected space. . The n-dimensional operations consist of calculation of Ax, the solution of $[(I - x_i\, x_i{}^T)\ B_{\lambda_i}\ (I - x_i\, x_i{}^T)]\ q_i = r_i$, calculation of scalar products and linear combinations of vectors. Except for very peculiar data structure of A, all this is done best by using the existing efficient implementations of linear algebra, BLAS and LAPACK for the sequential and PBLAS and ScaLAPACK for parallel computing. The ARPACK and PARPACK packages [1,11] give careful implementations of the Lanczos and Arnoldi method and may serve as a guideline and provide building blocks for other implementations. On workstations, cache optimized linear algebra operations may perform up to tenfold better than naively written code [17].

The implicit orthogonalization saves about half the computations of the explicit one, but the stability problems that explicit orthogonalization is meant to solve remain. If the pivot elements of the Cholesky matrix C indicate a near degenerate basis, the culprit vectors in $\mathbb{R}^n$ have to be transformed to give truly independent directions. This can be organized best by using an incremental Cholesky decomposition.

The computations in low dimensional space that may be parallelized with benefit are the solution of the eigenvalue problem of $\hat{A}^{(k)}$, the residual minimization where vectors are treated independently and the process to choose the approximations that will be put to further use. All the n-dimensional linear algebra calculations can be distributed with benefit over different processors of parallel machines. As they have predictable computational effort, static load balancing will do. The distribution of A is critical[3]. It is also possible to treat groups of vectors on different groups of processors, but this gives more complicated load balancing and larger volumes of data transport for scalar products. The computations in $\mathbb{R}^m$ may be parallelized independent of those in $\mathbb{R}^n$.

There are full codes, building blocks and development tools available for almost any architecture, but writing efficient parallel programs still requires skill and insight. The easy-to-use methods like HPF or virtual shared memory are considerably less efficient than explicit message passing, which is not an easy-to-use method. The speedup available depends on problem size, but there are examples of a speedup of 500 on a 512 processor machine.

5 Conclusion

With the combination of Jacobi-Davidson iteration with harmonic Ritz projection and residual minimization, the last open problem of eigenvalue calculation - finding specified inner eigenpairs when the eigenvalue range is not know accurately - can be treated with subspace methods, and the question of efficiency is reduced primarily to finding an efficient approximate inverse of the shifted matrix. Implementations are not quite up to the state of theory yet, but it seems they are going to catch up within a few years.

References

1. R.B. Lehoucq, D.C. Sorensen, C. Yang, ARPACK Users' Guide: Solution of Large Scale Eigenvalue Problems with Implicitly Restarted Arnoldi Methods. *http://www.netlib.org/sclapack/arpack_ug.ps.gz*
2. A. Basermann and B. Steffen, New Preconditioned Solvers for Large Sparse Eigenvalue Problems on Massively Parallel Computers, *Proceedings of the Eighth SIAM Conference on Parallel Processing for Scientific Computing,* CD-ROM, SIAM , Philadelphia, (1997).
3. A. Basermann and B. Steffen, Preconditioned Solvers for Large Eigenvalue Problems on Massively Parallel Computers and Workstation Clusters, *Parallel Computing: Fundamentals, Applications and New Directions*, eds. E.H. D'Hollander, G.R. Joubert, F.J. Peters, and U. Trottenberg, Elsevier Science B. V., 565 (1998).

4. J.K. Cullum and R.A. Willoughby, *Lanczos Algorithms for Large Symmetric Eigenvalue Computations*, Volume I: Theory, Birkhäuser, Boston Basel Stuttgart, 1985.
5. E.R. Davidson, The iterative calculation of a few of the lowest eigenvalues and corresponding eigenvectors of large real symmetric matrices, *J. Comp. Phys.* **17**, 87-94 (1975).
6. R.W. Freund and N.M. Nachtigal, QMR, A quasi minimal residual method for non-hermitian linear systems, Numer. Math. **60**, 315 (1991).
7. A. Goeke, Parallele Teilraumverfahren zur Bestimmung von Eigenwerten im Innern des Spektrums, *Jül 3794*, Jlich, (2000) .
8. N. Kosugi, Modification of the Liu-Davidson method for obtaining one or simultaneously several eigensolutions of a large real symmetric matrix, *Comput. Phys.* **55**, 426-436 (1984).
9. B. Lang, Direct solvers for symmetric eigenvalue problems, *in: Modern Methods and Algorithms of Quantum chemistry, J. Grotendorst(Ed.)* NIC, Jlich, 203 (2000).
10. R.B. Morgan, Computing interior eigenvalues of large matrices Linear Algebra Appl. **154-156**, 289-309 (1991) .
11. K.J. Maschhoff, D.C. Sorensen, A Portable Implementation of ARPACK for Distributed Memory Parallel Architectures, *http://www.caam.rice.edu/ kistyn/parpack_home.html*
12. B.N. Parlett *The Symmetric Eigenvalue Problem*, SIAM, Philadelphia, PA (1998) Updated reprint of the 1980 Prentice Hall edition .
13. H. Rutishauser, Computational aspects of F. L. Bauer's simultaneous iteration method, Numer. Math. **13**, 4-13 (1969).
14. B. Steffen, *An Improved Version of the Semi-analytical Eigenvector Processor* KFA-ZAM-IB-9105, (1991).
15. G.L.G. Sleijpen and H.A. van der Vorst,A Jacobi-Davidson iteration for linear eigenvalue problems SIAM J. Matrix Anal. Appl. **17**, 401-425 (1996).
16. J. Tückmantel, *An Improved Version of the Semi-analytical Eigenvector Processor SAP*, CERN/EF/RF 85-4, (1985).
17. S. Turek, *Trends in processor technology and their impact on Numerics for PDE's*, http://www.mathematik.uni-dortmund.de/htmldata1/featflow/ture/, To appear in Numerische Mathematik (2000).