Eigenvalue solvers for electromagnetic fields in cavities

Peter Arbenz and Roman Geus

Swiss Federal Institute of Technology (ETH), Institute of Scientific Computing, CH-8092 Zurich, Switzerland.

Abstract. We report on a comparison of the implicitly restarted Lanczos algorithm as implemented in ARPACK and the Jacobi-Davidson algorithm for solving large sparse generalized symmetric matrix eigenvalue problems. These problems occur in the computation of a few of the lowest frequencies of standing electromagnetic waves in cavity resonators. The computational domain is discretized by a finite element method based on edge elements to avoid spurious modes.

1 Introduction

Most particle accelerators use standing waves in cavities to produce the high voltage RF (radio frequency) fields required for the acceleration of the particles. The mathematical model for these high frequency electromagnetic fields is the eigenvalue problem solving Maxwell equations in a bounded volume. Usually, the eigenfield corresponding to the fundamental mode of the cavity is used as the accelerating field. Higher order modes can be excited due to higher harmonic components contained in the RF (radio frequency) power fed into the cavity and through interactions between the accelerated particles and the electromagnetic field. The RF engineer designing such an accelerating cavity therefore needs a tool to compute the fundamental and about ten to twenty of the following eigenfrequencies together with the corresponding electomagnetic eigenfields.

After separation of variables depending on space and time and after elimination of the magnetic field terms the variational form of the eigenvalue problem for the electric field intensity is given by [1]

$$\begin{aligned} &\textit{Find } (\lambda, \mathbf{u}) \in \mathbb{R} \times W_0 \textit{ such that } \mathbf{u} \neq \mathbf{0} \textit{ and} \\ &(\mathbf{curl}\,\mathbf{u}, \mathbf{curl}\,\mathbf{v}) = \lambda(\mathbf{u}, \mathbf{v}), \qquad \forall \mathbf{v} \in W_0. \end{aligned} \tag{1}$$

Let $L^2(\Omega)$ be the Hilbert space of square-integrable functions over the 3-dimensional domain Ω (the cavity) with inner product $(\mathbf{u}, \mathbf{v}) = \int_\Omega \mathbf{u}(\mathbf{x})\cdot\mathbf{v}(\mathbf{x})\,d\mathbf{x}$ and norm $\|\mathbf{u}\|_{0,\Omega} := (\mathbf{u}, \mathbf{u})^{1/2}$. In (1), $W_0 := \{\mathbf{v} \in W \mid \operatorname{div} \mathbf{v} = 0\}$ with

$$W := \{\mathbf{v} \in L^2(\Omega)^3 \mid \mathbf{curl}\,\mathbf{v} \in L^2(\Omega)^3 \;,\; \operatorname{div} \mathbf{v} \in L^2(\Omega) \;,\; \mathbf{n} \times \mathbf{v} = 0 \text{ on } \partial\Omega\}.$$

This eigenvalue problem has a pure point spectrum [1]. The difficulty with (1) stems from the condition $\operatorname{div} \mathbf{v} = 0$ as it is hard to find divergence-free finite elements. Therefore, ways have been looked for to get around the divergence-free condition. In this process care has to be taken in order not to introduce so-called *spurious modes*, i.e. eigenmodes that have no physical meaning [13]. One approach free of spurious modes is based on a mixed finite element formulation in which the divergence-free condition is enforced by means of Lagrange multipliers [8].

$$\begin{array}{ll} \textit{Find } (\lambda, \mathbf{u}, p) \in \mathrm{I\!R} \times H_0(\mathbf{curl};\Omega) \times H_0^1(\Omega) \textit{ such that } \mathbf{u} \neq \mathbf{0} \textit{ and} & \\ \text{(a) } (\mathbf{curl}\,\mathbf{u}, \mathbf{curl}\,\boldsymbol{\Psi}) + (\mathbf{grad}\,p, \boldsymbol{\Psi}) = \lambda(\mathbf{u}, \boldsymbol{\Psi}), & \forall \boldsymbol{\Psi} \in H_0(\mathbf{curl};\Omega) \\ \text{(b) } (\mathbf{u}, \mathbf{grad}\,q) = 0, & \forall q \in H_0^1(\Omega) \end{array} \tag{2}$$

Here, $H_0(\mathbf{curl};\Omega) := \{\mathbf{v} \in L^2(\Omega)^3 \mid \mathbf{curl}\,\mathbf{v} \in L^2(\Omega)^3,\ \mathbf{n} \times \mathbf{v} = 0 \text{ on } \partial\Omega\}$. The finite element discretization of (2) yields a matrix eigenvalue problem of the form

$$\begin{bmatrix} A & C \\ C^T & O \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix} = \lambda \begin{bmatrix} M & O \\ O & O \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix}. \tag{3}$$

where A and M are n-by-n and C is n-by-m. Here, M is positive definite but A is only positive semidefinite. It turns out that (3) becomes most convenient to handle if the finite elements for the vector fields are chosen to be edge elements as proposed by Nédélec [11] and the Lagrange multipliers are represented by corresponding Lagrange (i.e. nodal-based) finite elements [13, p.300]. Then it turns out that the condition $C\mathbf{x} = \mathbf{0}$ is equivalent with requiring $\mathbf{x}$ to be M-orthogonal to the nullspace of A. So, the eigenpairs of (3) are precisely the eigenpairs of

$$A\mathbf{x} = \lambda M\mathbf{x} \tag{4}$$

corresponding to *positive* eigenvalues. To get rid of the high-dimensional eigenspace associated with the eigenvalue zero Bespalov [4] proposed to replace (4) by

$$(A + CHC^T)\mathbf{x} = \lambda M\mathbf{x}, \tag{5}$$

where H is a positive definite matrix chosen such that the zero eigenvalues are shifted to the right of the desired eigenvalues and do not disturb the computations. As suggested by Bespalov we set $H = \alpha I$, a multiple of the identity.

2 Algorithms

In this section we briefly survey the numerical methods that we will apply to the model problem which is a cavity of the form of a rectangular box, $\Omega = (0, a) \times (0, b) \times (0, c)$. We investigate the implicitly restarted Lanczos algorithm as implemented in ARPACK and the Jacobi-Davidson algorithm for computing a few, say p, of the smallest eigenvalues of the matrix eigenvalue problem (5). In the *shift-and-invert* spectral transformation approach [7] the eigenvalue problem (4) is transformed into

$$(A - \sigma M)^{-1} M\mathbf{x} = \mu\mathbf{x}, \qquad \mu = 1/(\lambda - \sigma), \tag{6}$$

with a *shift* σ close to the desired eigenvalues. (When solving (5), A in (6) is replaced by $A + CHC^T$.) Notice that the matrix to the left of (6) is M-symmetric, i.e., it is symmetric with respect to the inner product $\mathbf{x}^T M\mathbf{y}$. The spectral transformation leaves the eigenvectors unchanged. The eigenvalues of (5) close to the shift σ become the largest absolute of (6) and are relatively well-separated.

2.1 The implicitly restarted Lanczos algorithm (IRL)

Because of its large memory consumption it is often impossible to solve large eigenvalue problems to the desired accuracy with the Lanczos algorithm. It is necessary to *restart* the iterative process in some way with as little loss of information as possible. An elegant way to restart has been proposed by Sorensen first for the Arnoldi [16] and later for the symmetric Lanczos algorithm [5]. Software is publicly available in ARPACK [10]. The Lanczos iteration is executed till step $j = p + k$, where k is some positive integer. (In our implementation we set $k = p$ as suggested in [10].) Then by a clever application of the QR algorithm the dimension of the search space is reduced to p in such a way that the p new orthogonal basis vectors still form a Krylov sequence [16,5]. This allows the Lanczos phase to be resumed smoothly. Besides the storage for the matrices A and M, IRL requires space for two $n \times (p + k)$ arrays.

In each step of the Lanczos iteration a symmetric indefinite system of equations of the form $(A + CHC^T - \sigma M)\mathbf{x} = \mathbf{y}$ has to be solved. To this end we used the iterative solver SYMMLQ [12,3]. The accuracy of the solution of the linear system has to be at least as high as the desired accuracy in the eigenvalue calculation in order that the coefficients of the Lanczos three-term recurrence are reasonably accurate [10]. We experimented with various preconditioners. None of them was satisfactory. We obtained the best results with diagonal preconditioning.

2.2 The Jacobi-Davidson algorithm (JDQR).

The Jacobi-Davidson algorithm has been introduced by Sleijpen and van der Vorst [15] for solving arbitrary generalized eigenvalue problems. Here, we present a modification of their algorithm adapted to the eigenvalue problem $A\mathbf{x} = \lambda M\mathbf{x}$ with symmetric A and symmetric positive definite M.

In the Jacobi-Davidson algorithm the eigenpairs of $A\mathbf{x} = \lambda M\mathbf{x}$ are computed one by one. Let $\mathbf{q}_1, \ldots, \mathbf{q}_k$ be eigenvectors that have already been

Initialization:

Set $Q_0 = [\,]; W_0 = [\,]$; $k = 0$. Choose $\mathbf{v}_1$ with $\|\mathbf{v}_1\|_M = 1$.
$\mathbf{v}_A = A\mathbf{v}_1$; $H_1 = [\mathbf{v}_A^T\mathbf{v}_1]$; $V_1 = [\mathbf{v}_1]$;
$\tilde{\mathbf{q}} = \mathbf{v}_1$; $\tilde{\mathbf{w}} = M\tilde{\mathbf{q}}$; $\mathbf{r} = (A - \tilde{\lambda}M)\tilde{\mathbf{q}}$; $j = 1$;

Iteration: While $k < p$ do

(i) $j = j + 1$;
(ii) **Correction equation:** Solve approximately for $\mathbf{t}$
$\left\{(I - \tilde{Q}_k\tilde{W}_k^T)(A - \tilde{\lambda}M)\mathbf{t} = -(I - \tilde{Q}_k\tilde{W}_k^T)\mathbf{r}, \quad \tilde{W}_k^T\mathbf{t} = 0\right\}$
where $\tilde{Q}_k = [Q_k, \tilde{\mathbf{q}}]$ and $\tilde{W}_k = [W_k, \tilde{\mathbf{w}}]$ $(= M\tilde{Q}_k)$.
(iii) $\mathbf{v}_j = (I - V_{j-1}V_{j-1}^T M)\mathbf{t}/\|(I - V_{j-1}V_{j-1}^T M)\mathbf{t}\|_M$; $\quad V_j = [V_{j-1}, \mathbf{v}_j]$
(iv) $\mathbf{v}_A = A\mathbf{v}_j$; $H_j = \begin{bmatrix} H_{j-1} & V_{j-1}^T\mathbf{v}_A \\ \mathbf{v}_A^T V_{j-1} & \mathbf{v}_A^T\mathbf{v}_j \end{bmatrix}$
(v) **Rayleigh-Ritz step:** Decompose $H_j = S_j\Lambda_j S_j^{-1}$ by QR algorithm
where $S_j = [\mathbf{s}_1, \ldots, \mathbf{s}_j]$ and $\Lambda_j = \mathrm{diag}(\lambda_1^{(j)}, \ldots, \lambda_j^{(j)})$
with $|\lambda_l^{(j)} - \sigma| \le |\lambda_{l+1}^{(j)} - \sigma|$, $l = 2, \ldots, j$
(vi) **Convergence test**
repeat
(a) $\tilde{\lambda} = \lambda_1^{(j)}$; $\quad \tilde{\mathbf{q}} = V_j\mathbf{s}_1$; $\quad \tilde{\mathbf{w}} = M\tilde{\mathbf{q}}$; $\quad \mathbf{r} = (A - \tilde{\lambda}M)\tilde{\mathbf{q}} = A\tilde{\mathbf{q}} - \tilde{\lambda}\tilde{\mathbf{w}}$;
(b) found $= (\|\mathbf{r}\|_2 < \varepsilon$ and $(j>1$ or $k=p-1)$ and $k<p)$;
(c) if found,
$Q_{k+1} = [Q_k, \tilde{\mathbf{q}}]$; $\quad W_{k+1} = [W_k, \tilde{\mathbf{w}}]$; $\quad k = k + 1$;
$V_{j-1} = V_j[\mathbf{s}_2, \ldots, \mathbf{s}_j]$; $\quad \Lambda_{j-1} = \mathrm{diag}(\lambda_2^{(j)}, \ldots, \lambda_j^{(j)})$;
$H_{j-1} = \Lambda_{j-1}$; $\quad S_{j-1} = I_{j-1}$; $\quad j = j - 1$;
until not(found)
(vii) **Implicit restart:** if $j = j_{\max}$
$V_{j_{\min}} = V_j[\mathbf{s}_1, \ldots, \mathbf{s}_{j_{\min}}]$; $\quad \Lambda_{j_{\min}} = \mathrm{diag}(\lambda_1^{(j)}, \ldots, \lambda_{j_{\min}}^{(j)})$;
$H_{j_{\min}} = \Lambda_{j_{\min}}$; $\quad S_{j_{\min}} = I_{j_{\min}}$; $\quad j = j_{\min}$;

Fig. 1. JDQR: M-orthogonal Jacobi-Davidson algorithm without preconditioning. The Rayleigh-Ritz problem is solved by the symmetric QR algorithm.

computed and let $(\tilde{\lambda}, \tilde{\mathbf{q}})$ be an approximation to a further eigenpair $(\lambda, \mathbf{u})$ with $\mathbf{u}^T M Q_k = \mathbf{0}$ where $Q_k = [\mathbf{q}_1, \ldots, \mathbf{q}_k]$. The pair $(\tilde{\lambda}, \tilde{\mathbf{q}})$ is obtained from a Ritz-Galerkin projection of (4) onto the search space $\mathcal{R}(V_j)$ spanned by the columns of $V_j \in \mathbb{R}^{n\times j}$, $V_j^T M V_j = I_j$, $V_j^T M Q_k = O$,

$$H_j\mathbf{x} = V_j^T A V_j\mathbf{x} = \mu V_j^T M V_j\mathbf{x} = \mu\mathbf{x}, \qquad H_j := V_j^T A V_j. \tag{7}$$

As with the implicitly restarted Lanczos procedure the search space is expanded up to a certain dimension, say $j_{\max}$. In the expansion phase $\mathcal{R}(V_j)$ is extended by a vector $\mathbf{t}$ M-orthogonal to the current eigenvector approximation $\tilde{\mathbf{q}}$ and M-orthogonal to the already computed eigenvectors by means of the correction equation [14] in step (ii) of Fig. 1,

$$(I-\tilde{Q}_k\tilde{W}_k^T)(A - \tilde{\lambda}M)\mathbf{t} = -(I-\tilde{Q}_k\tilde{W}_k^T)\mathbf{r}, \quad \tilde{W}_k^T\mathbf{t} := M\tilde{Q}_k^T\mathbf{t} = 0. \tag{8}$$

Notice that this equation is consistent by construction as the residual $\mathbf{r} := (A - \tilde{\lambda}M)\tilde{\mathbf{q}}$ is orthogonal (in the Euclidean sense) to $\mathrm{span}\{V_k\}$.

Equation (8) needs to be solved only approximately. Therefore, an iterative solver can be used. Fokkema et al. [6] proposed to use a preconditioner of the form $(I - \tilde{Q}_k\tilde{W}_k^T)K(I - \tilde{Q}_k\tilde{W}_k^T)$ where K is a good and easily invertible approximation of $A - \tilde{\lambda}M$. Following [6] one finds that

$$(I - \tilde{Y}_k\tilde{H}_k^{-1}\tilde{W}_k^T)K^{-1}, \qquad \tilde{Y}_k = K^{-1}\tilde{Q}_k, \quad \tilde{H}_k = \tilde{Y}_k^T M\tilde{Q}_k, \tag{9}$$

is the Moore-Penrose pseudo-inverse of this preconditioner [1]. Multiplying the matrix in (9) with the first equation in (8) yields the preconditioned correction equation

$$(I - \tilde{Y}_k\tilde{H}_k^{-1}\tilde{W}_k^T)K^{-1}(A - \tilde{\lambda}M)\mathbf{t} = -(I - \tilde{Y}_k\tilde{H}_k^{-1}\tilde{W}_k^T)K^{-1}\mathbf{r}, \quad \tilde{W}_k^T\mathbf{t} = \mathbf{0}, \tag{10}$$

which in the actual algorithm replaces equation (8).

We solved (10) by the conjugate gradient squared (CGS) method [3] with K equal to the diagonal of $A - \tilde{\lambda}M$. The projectors in (10) or step (ii) in Fig. 1 have been implemented in a straightforward way. This amounts to a step of classical Gram-Schmidt. Therefore, the approximate solution $\mathbf{t}$ of the correction equation will not satisfy the orthogonality constraint $\tilde{W}_k^T\mathbf{t} = \mathbf{0}$. This constraint has been enforced by a final step of the modified Gram-Schmidt orthogonalization process. The step (iii) in Fig. 1 is implemented by a further step of the modified Gram-Schmidt.

We stopped the iterative solution of (10) in the ℓ-th step if $\|\tilde{\mathbf{s}}_\ell\| < (\frac{2}{3})^i\|\tilde{\mathbf{r}}_i\|$ where $\tilde{\mathbf{r}}_i$ is the right side and $\tilde{\mathbf{s}}_\ell$ is the residual of (10). We obtained better results using the factor $2/3$ instead of $1/2$ which is proposed in [6]. Here i indicates the number of times the system (10) has been solved to compute a certain eigenvalue. The counter i is reset to zero with each new eigenvalue to compute.

If the search space has dimension $j_{\max}$ it is shrunk to dimension $j_{\min}$ by selecting only the $j_{\min}$ Ritz vectors corresponding to Ritz values closest to the shift σ. We set $j_{\max} = 2j_{\min} = 2p$. In JDQR, we successively updated σ to be as large as the largest eigenvalue previously computed. In this way we prevented the vectors form converging to $\mathcal{N}(A)$.

Besides the storage for the matrices $A - \sigma M$ and M, memory space is needed for a $n \times j_{\max}$ array and for three $n \times p$ arrays.

2.3 Remark on the buckling spectral transformation

Instead of (6) one could solve

$$(A - \sigma M)^{-1}A\mathbf{x} = \nu\mathbf{x}, \qquad \nu = \lambda/(\lambda - \sigma). \tag{11}$$

This is in fact very natural as the eigenvectors corresponding to the positive eigenvalues of (4) are in the range of $(A - \sigma M)^{-1}A$. The matrix $(A - \sigma M)^{-1}A$ is M-symmetric as well as A-symmetric.

In the buckling spectral transformation [7,10] orthogonality is taken with respect to $\mathbf{x}^T A\mathbf{y}$ which is an inner product on $\mathcal{R}((A-\sigma M)^{-1}A)$, see [1]. In our experiments with the buckling mode we did not get correct results with ARPACK. This is probably due to a bug that is fixed only in the most recent (not yet public) version of the package [9]. Replacing the inner product $\mathbf{x}^T A\mathbf{y}$ by $\mathbf{x}^T M\mathbf{y}$ as in the shift-and-invert mode gave satisfactory results. The timings we obtained in this way were about the same as those obtained with the shift-and-invert spectral transformation.

With JDQR it is possible to work in the range of $(A-\sigma M)^{-1}A$ equipped with the inner product $\mathbf{x}^T A\mathbf{y}$, too. The correction equation (8) has to be projected onto $\mathcal{R}((A-\sigma M)^{-1}A)$. This can be accomplished with the M-orthogonal projector $I - M^{-1}C(C^T M^{-1}C)C$ where C is the matrix in (3). Notice that $M^{-1}C$ is a basis of the null space of A. It appears that forming $I - M^{-1}C(C^T M^{-1}C)C$ is possible but would consume too much memory space.

3 Numerical experiments

In this section we report on the comparison of IRL and JDQR. We computed the 15 smallest eigenvalues of (1) with $\Omega = (0,a)\times(0,b)\times(0,c)$ where $a = 0.8421$, $b = 0.5344$, and $c = 0.2187$. The unrealistic box cavity has the advantage that its eigenvalues and eigenmodes are given analytically [1]. It is thus easy to determine approximation errors. The computational results have been obtained with the 32-processor HP Exemplar X-class system at the ETH Zurich. We used the machine for its large memory. Preliminary results on a parallel implementation are reported in [2].

3.1 Accuracy with various types of Nédélec elements

The model domain allows a comparison of box and tetrahedral elements. In Fig. 2 the computing cost is plotted against the relative accuracy of the eigenvalue λ_1 for linear and quadratic box and tetrahedral Nédélec elements. (The higher eigenvalues show a similar behavior but are of course less accurate.) The execution times comprise the solution time of the ARPACK eigensolver for computing 15 eigenvalues but not the time for building the matrices A and M. The box-shaped domain Ω is subdivided in the longest direction in a number of intervals that is increased by steps of 1. The subdivisions in the other directions are made such that the subdomains get cube-shaped as good as possible.

The plots show that it is cheaper to get accurate eigenvalues with box elements than with tetrahedral elements. This is not surprising, as the number of degrees of freedom is considerably higher with the tetrahedral elements than with the box elements while the approximation power is not increased accordingly. (There are 6 tetrahedrons in a box.) The difference is larger with

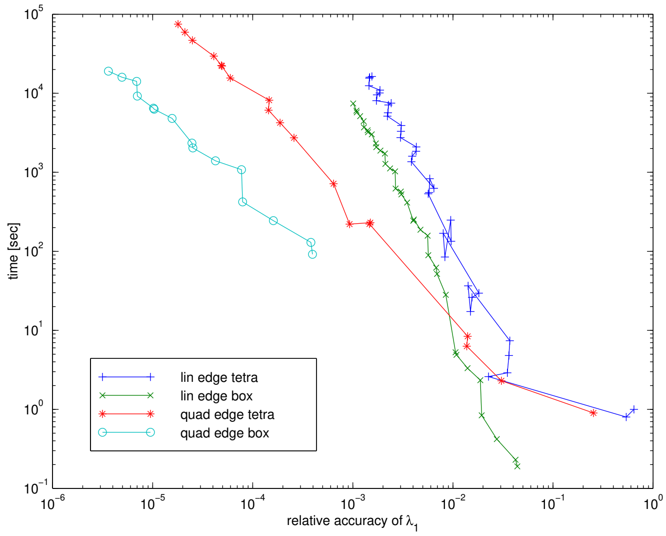

Fig. 2. Comparison of tetrahedral and box Nédélec elements.

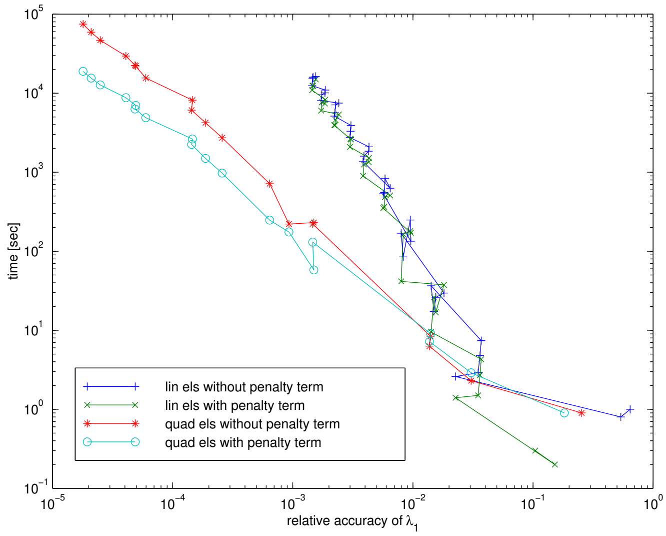

Fig. 3. Comparison of tetrahedral Nédélec elements with and without penalty term.

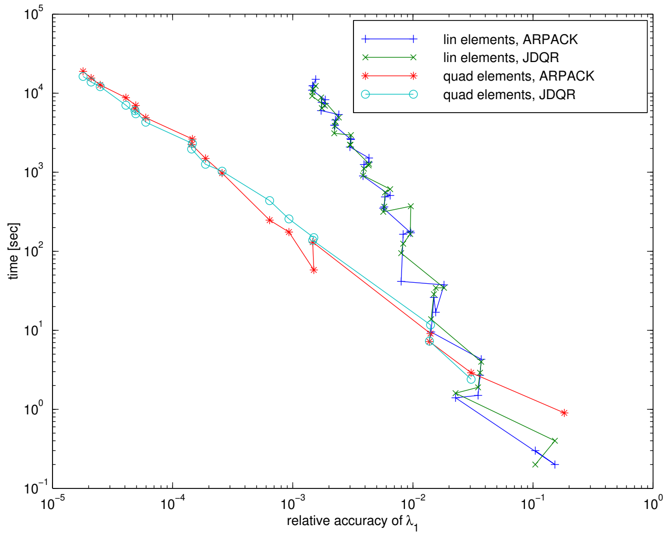

Fig. 4. Performance comparison of IRL and JDQR with penalty term.

quadratic elements that have numerous degrees of freedom on the element interfaces. These are shared by only two finite elements.

3.2 Bespalov's penalty term

With the results plotted in Fig. 3 we checked if introducing Bespalov's penalty term CHC^T in (5) is advantageous. We chose $H = \alpha I$ as suggested by Bespalov [4] and set $\alpha = 50/h$. The comparison of execution times again obtained with IRL and accuracy reveals that convergence is improved considerably.

3.3 IRL vs. JDQR

In Fig. 4 the implicitly restarted Lanczos algorithm is compared with the Jacobi-Davidson algorithm.

The times for the *linear* elements show quite an erratic behaviour. The grid is quite fine in this case. It should be noted that the refinement of the grid in one direction increases the problem size (and thus solution time) but does not improve the accuracy of eigenfunctions that do not vary in this direction. While IRL is ahead of JDQR for most of the small problem sizes, JDQR is 10 to 20% ahead of IRL for most of the larger problem sizes.

With the *quadratic* elements the mesh is much coarser. The approximation power originates in the higher polynomial degrees of the finite elements. The timing curves are relatively 'smooth'. Here, JDQR is consistently faster than IRL by about 10 to 20% for the large problem sizes.

We now discuss the behavior of the two algorithms in the case of the quadratic elements where the problem size grows from $n = 1408$ to 45226. (In the latter case A and M have each about 1'730'000 nonzero elements.) With both algorithms the number of outer iteration steps varies only little. It is ~ 65 with IRL and ~ 210 with JDQR. So, the number of restarts does not depend on the problem size. Each outer iteration step requires the solution of one system of equations. We used the solver SYMMLQ with IRL and CGS for JDQR. One (inner) iteration step of CGS counts for about two iteration steps of CGS. We applied diagonal preconditioning in all cases. The superiority of JDQR over IRL for large problem sizes can be explained by the number of inner iteration steps. The average number of inner iteration steps per outer iteration step grows from 64 to 91 with JDQR but from 423 to 1140 with IRL. In ARPACK each system of equations is solved to high accuracy. In JDQR the accuracy requirement is not so stringent and actually varies form (outer) step to step, cf. §2.2. This explains the higher iteration numbers for IRL. The projections in (10) improve the condition number of the system matrix. Further, the shift σ is updated in JDQR while it stays constant with ARPACK. These may be the reasons for the less pronounced increase of the number of inner iteration steps with JDQR. Notice that the projections made in each iteration step account for less than 10% of the execution time of JDQR.

4 Conclusions

In [1] we have observed that the block Lanczos algorithm is the fastest algorithm for solving (4). Its memory requirements however limit its applicability. The largest of our problems could not have been solved with Lanczos. The algorithms that work for very large problems are those considered in this study: the implicitly restarted Lanczos and the Jacobi-Davidson algorithm. In our experiments the IRL algorithm combined with SYMMLQ for solving the symmetric-indefinite linear system of equations in the shift-and-invert approach was about 10% faster than Jacobi-Davidson combined with CGS.

For computing the lowest frequencies of a box-shaped cavity resonator quadratic finite elements are to be preferred over linear finite elements and structured grids are to be preferred over non-structured grids. However, realistic cavities will probably not admit structured grids anymore. Also, nodal-based finite elements that outperform edge elements in the case the box-shaped cavities [1] will hardly be successful in realistic domains.

So far we solved the sparse systems using diagonal preconditioning. In subsequent work we will try to apply more sophisticated preconditioners.

References

1. St. Adam, P. Arbenz, and R. Geus, *Eigenvalue solvers for electromagnetic fields in cavities*, Tech. Report 275, ETH Zürich, Computer Science Department, October 1997, (Available at URL `http://www.inf.ethz.ch/publications/tr.html`).
2. P. Arbenz and R. Geus, *Parallel solvers for large eigenvalue problems originating from Maxwell's equations*, Euro-Par '98, Springer-Verlag, 1998, (Lecture Notes in Computer Science).
3. R. Barret, M. Berry, T. F. Chan, J. Demmel, J. Donato, J. Dongarra, V. Eijkhout, R. Pozo, Ch. Romine, and H. van der Vorst, *Templates for the solution of linear systems: Building blocks for iterative methods*, Society for Industrial and Applied Mathematics, Philadelphia, PA, 1994, (Available from Netlib at URL `http://www.netlib.org/templates/index.html`).
4. A. N. Bespalov, *Finite element method for the eigenmode problem of a RF cavity resonator*, Soviet Journal of Numerical Analysis and Mathematical Modelling **3** (1988), 163–178.
5. D. Calvetti, L. Reichel, and D. C. Sorensen, *An implicitely restarted Lanczos method for large symmetric eigenvalue problems*, Electronic Transmissions on Numerical Analysis **2** (1994), 1–21.
6. D. R. Fokkema, G. L. G. Sleijpen, and H. A. van der Vorst, *Jacobi-Davidson style QR and QZ algorithms for the partial reduction of matrix pencils*, Preprint 941, revised version, Utrecht University, Department of Mathematics, Utrecht, The Netherlands, January 1997.
7. R. Grimes, J. G. Lewis, and H. Simon, *A shifted block Lanczos algorithm for solving sparse symmetric generalized eigenproblems*, SIAM J. Matrix Anal. Appl. **15** (1994), 228–272.
8. F. Kikuchi, *Mixed and penalty formulations for finite element analysis of an eigenvalue problem in electromagnetism*, Computer Methods in Applied Mechanics and Engineering **64** (1987), 509–521.
9. R. B. Lehoucq, Private communication, May 1998.
10. R. B. Lehoucq, D. C. Sorensen, and C. Yang, *ARPACK users' guide: Solution of large scale eigenvalue problems by implicitely restarted Arnoldi methods*, Department of Mathematical Sciences, Rice University, Houston TX, October 1997, (The software and this guide are available at URL `ftp://ftp.caam.rice.edu/pub/software/ARPACK/`).
11. J. C. Nédélec, *Mixed finite elements in* $\mathbb{R}^3$, Numerische Mathematik **35** (1980), 315–341.
12. C. C. Paige and M. A. Saunders, *Solution of sparse indefinite systems of linear equations*, SIAM J. Numer. Anal. **12** (1975), 617–629.
13. P. P. Silvester and R. L. Ferrari, *Finite elements for electrical engineers*, 3rd ed., Cambridge University Press, Cambridge, 1996.
14. G. L. G. Sleijpen, A. G. L. Booten, D. R. Fokkema, and H. A. van der Vorst, *Jacobi-Davidson type methods for generalized eigenproblems and polynomial eigenproblems*, BIT **36** (1996), 595–633.
15. G. L. G. Sleijpen and H. A. van der Vorst, *A Jacobi-Davidson iteration method for linear eigenvalue problems*, SIAM J. Matrix Anal. Appl. **17** (1995), 401–425.
16. D. Sorensen, *Implicite application of polynomial filters in a* k*-step Arnoldi method*, SIAM J. Matrix Anal. Appl. **13** (1992), 357–385.