

On Solving Maxwell Eigenvalue Problems for Accelerating Cavities

Peter Arbenz[†] and Roman Geus[‡]

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

Stefan Adam[§]

Paul-Scherrer-Institute, CH-5232 Villigen, Switzerland

(Dated: September 6, 2000)

We investigate algorithms for computing steady state electromagnetic waves in cavities. The Maxwell equations for the strength of the electric field are solved by a mixed method with quadratic finite edge elements for the field values and corresponding node-based finite elements for the Lagrange multiplier. This approach avoids so-called spurious modes which are introduced if the divergence-free condition for the electric field is not treated properly. To compute a few of the smallest positive eigenvalues and corresponding eigenmodes of the resulting large sparse matrix eigenvalue problems two algorithms have been used, the implicitly restarted Lanczos and Jacobi-Davidson algorithm, both with shift-and-invert spectral transformation. Two-level hierarchical basis preconditioners have been employed for the iterative solution of the resulting systems of equations.

PACS numbers: 29.20.Hm 41.20.Jb 02.60.Dc

I. INTRODUCTION

The common way to produce the accelerating electromagnetic fields in cyclic accelerators is to excite standing waves in accelerating cavities. The mathematical model for these high frequency electromagnetic fields is the eigenvalue problem solving the Maxwell equations in a bounded volume [1].

Usually, the eigenfield corresponding to the fundamental mode of the cavity is used as the accelerating field. A few modes of higher order have to be analyzed as well because these 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 electromagnetic eigenfields. The most interesting quantities besides the eigenfrequencies are local maxima of the eigenmodes as well as the fields on the surface that induce heat in the metallic boundary and determine the power loss from surface currents.

The finite element formulation for the solution of this

problem has been chosen for its high flexibility for the geometric modeling, in particular with strong variations of the scale of structural details.

For the solution of Maxwell eigenvalue problems Kikuchi [2] suggested a mixed formulation method based on Nédélec's edge elements [3, 4] for the electric field and node-based elements for the Lagrange multiplier. In this way, the computed approximate eigenmodes are naturally split into (physically meaningless) curl-free and (in a discrete sense) divergence-free modes. Spurious modes cannot appear. In the engineering literature the advantages and disadvantages of Nédélec's edge elements vs. other finite element formulations are discussed intensely [5, 6].

By numerical experiments reported in [7] we identified Sorensen's implicitly restarted Lanczos algorithm [8] and the Jacobi-Davidson algorithm [9] to be the algorithms best suited to solve the large sparse eigenvalue problems that arise with the above finite element discretizations. They are to be preferred to subspace iteration [10] or the block Lanczos algorithm [11].

In this paper we report on our continuing investigations on eigensolvers and their preconditioning. In contrast to [7] we conduct our experiments on a domain that approximates the new cavity design for the 590 MeV ring cyclotron at the Paul Scherrer Institute (PSI) in Villigen, Switzerland, see Fig. 1 on the right-hand side.

The outline of the paper is as follows. In section II we will give the mathematical formulation of the problem. In section III the Nédélec finite element approach for tetrahedral meshes is looked at from a linear algebra point of view. In sections IV to VI we detail how

[†]Electronic address: arbenz@inf.ethz.ch; URL: <http://www.inf.ethz.ch/~arbenz/>

[‡]Electronic address: geus@inf.ethz.ch; URL: <http://www.inf.ethz.ch/~geus/>

[§]Electronic address: Stefan.Adam@psi.ch

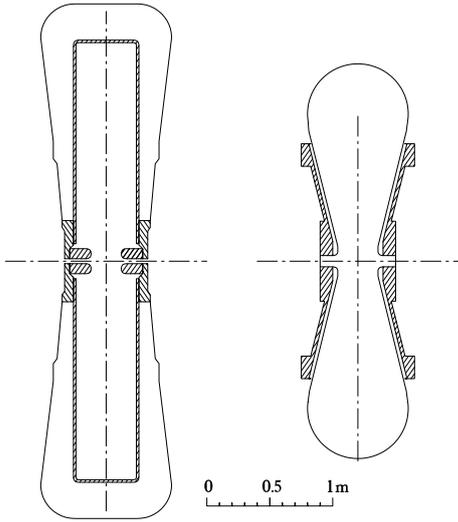


FIG. 1: Cross section of two different designs for accelerating cavities. On the left the original design that is presently in operation; on the right the future design. The plane of these two cross sections is perpendicular to the accelerator mid plane and tangential to the orbit of the accelerated particles. Both cavities extend prismatically in the radial direction over approximately 3.3 meters.

we solve the large eigenvalue problems and in particular solve the systems of equations that are introduced by the spectral transformation. In section VII we compare the performance of several variants in solving the model problem.

II. STATEMENT OF THE PROBLEM

Without changing the basic structure of the problem one can assume that the metallic surfaces are perfectly conducting and that the inside volume of the cavity Ω is all in vacuum. The electromagnetic field in the cavity is described by the Maxwell equations [1]. After separation of time and space and after elimination of the magnetic field intensity the differential equations

$$\mathbf{curl} \mathbf{curl} \mathbf{e}(\mathbf{x}) = \lambda \mathbf{e}(\mathbf{x}), \quad \mathbf{x} \in \Omega, \quad \lambda := \omega^2/c^2, \quad (1a)$$

$$\mathbf{div} \mathbf{e}(\mathbf{x}) = 0, \quad \mathbf{x} \in \Omega, \quad (1b)$$

$$\mathbf{n} \times \mathbf{e} = 0, \quad \mathbf{x} \in \Gamma := \partial\Omega. \quad (1c)$$

are obtained for the electric field intensity \mathbf{e} . Equations (1) have solutions only for certain values ω called *eigenfrequencies* of the electromagnetic oscillations of the resonator. These solutions are called *eigenmodes*.

The eigenvalue problem (1) can be solved analytically for a few particular domains, e.g. when Ω is a rectangular box or a right circular cylinder [1, p.355]. We intend to solve (1) in more complicated domains Ω by the finite

element method. To that end we have to write the eigenvalue problem in variational form. There are a number of ways to do so. The mixed formulation proposed by Kikuchi [2]

Find $(\lambda, \mathbf{e}, p) \in \mathbb{R} \times H_0(\mathbf{curl}; \Omega) \times H_0^1(\Omega)$ such that $\mathbf{e} \neq \mathbf{0}$ and

$$(a) \quad (\mathbf{curl} \mathbf{e}, \mathbf{curl} \Psi) + (\mathbf{grad} p, \Psi) = \lambda(\mathbf{e}, \Psi), \quad \forall \Psi \in H_0(\mathbf{curl}; \Omega) \quad (2)$$

$$(b) \quad (\mathbf{e}, \mathbf{grad} q) = 0, \quad \forall q \in H_0^1(\Omega)$$

is useful as it leads to a mixed finite element formulation that is free of spurious modes [5, 6]. Here, $(\mathbf{u}, \mathbf{v}) = \int_{\Omega} \mathbf{u}(\mathbf{x}) \cdot \mathbf{v}(\mathbf{x}) d\mathbf{x}$ denotes the usual inner product in $L^2(\Omega)^3$, the space of 3-vector valued square integrable functions on Ω . The well-known Sobolev space [12, p.16]

$$H_0^1(\Omega) := \{q \in L^2(\Omega) \mid \mathbf{grad} q \in L^2(\Omega)^3, \quad q = 0 \text{ on } \Gamma\}$$

and the Hilbert space [13, 14]

$$H_0(\mathbf{curl}; \Omega) := \{\mathbf{v} \in L^2(\Omega)^3 \mid \mathbf{curl} \mathbf{v} \in L^2(\Omega)^3, \quad \mathbf{v} \times \mathbf{n} = \mathbf{0} \text{ on } \Gamma\}$$

are used in the formulation (2).

It is easy to see, that $p = 0$ for all solutions of (2), see [2], and that there is a one-to-one correspondence between the solutions (λ, \mathbf{e}) of (1) and the solutions $(\lambda, \mathbf{e}, 0)$ of (2).

In the next section we investigate the finite element discretization of $H_0(\mathbf{curl}; \Omega)$ and $H_0^1(\Omega)$ by Nédélec edge elements and Lagrange node elements, respectively. If also $H_0(\mathbf{curl}; \Omega)$ is approximated by Lagrange elements the eigenvalue zero will not be confined in the neighborhood of zero and can cause spectral pollution.

III. FINITE ELEMENT DISCRETIZATIONS

In this section we discuss finite element approximations of the mixed formulation (2). We assume that Ω is an open, bounded domain in \mathbb{R}^3 with a polyhedral boundary Γ . The domain is triangulated by tetrahedrons.

We replace $H_0(\mathbf{curl}; \Omega)$ and $H_0^1(\Omega)$ by finite dimensional subspaces V_h and W_h , respectively, to obtain

Find $(\lambda_h, \mathbf{e}_h, p_h) \in \mathbb{R} \times V_h \times W_h$ such

that $\mathbf{e}_h \neq \mathbf{0}$ and

$$(a) \quad (\mathbf{curl} \mathbf{e}_h, \mathbf{curl} \Psi_h) + (\mathbf{grad} p_h, \Psi_h) = \lambda_h(\mathbf{e}_h, \Psi_h), \quad \forall \Psi_h \in V_h, \quad (3)$$

$$(b) \quad (\mathbf{e}_h, \mathbf{grad} q_h) = 0, \quad \forall q_h \in W_h.$$

The space $W_h \subset H_0^1(\Omega)$ is the well-known Lagrange (or node-based) finite element space [12],

$$W_h^{(k)} := \{\mathbf{v} \in W \mid \mathbf{v}|_{\Omega_e} \in (P_k(\Omega_e))^3\}, \quad (4)$$

where $P_k(\Omega_e)$ is the space of polynomials in Ω_e of degree $\leq k$. In this paper we exclusively deal with $W_h^{(2)}$. We work with a hierarchical basis and write

$$W_h^{(2)} = W_h^{(1)} \oplus \bar{W}_h^{(2)}. \quad (5)$$

The natural basis of $W_h^{(1)}$ is formed by the piecewise linear polynomials that are 1 in one node (vertex) k and 0 in all others. Thus, $m_1 := \dim W_h^{(1)}$ equals the number of interior vertices of the triangulation. The basis of $\bar{W}_h^{(2)}$ consists of those piecewise quadratic polynomials that are 1 in the midpoint of one edge and 0 in the midpoints of all other edges and in all vertices. So, $m_2 = m - m_1 = \dim \bar{W}_h^{(2)}$ is equal to the number of interior edges of the triangulation.

The space V_h is made up of the $H(\mathbf{curl}; \Omega)$ -conforming finite elements proposed by Nédélec [4, 14, 15]. The basis of $V_h^{(1)}$ consists of piecewise linear vector valued functions with tangential component equal 1 on one interior edge and equal zero on all other edges. Therefore, $n_1 := \dim V_h^{(1)} = m_2$. The basis of $\bar{V}_h^{(2)}$ consists of two sets of functions. The first set is made up of piecewise quadratic polynomials that have a linearly varying tangential component along one interior edge and are equal to zero on all other edges. Each of these basis functions is curl-free and equals the gradient of the basis functions of $\bar{W}_h^{(2)}$ corresponding to the same edge! The second set of basis functions of $\bar{V}_h^{(2)}$ is made up of piecewise quadratic polynomials that have their degrees of freedom on the element faces. There are two basis functions per face. They are defined by the two zeroth moments of the tangential components on the face. So, $n_2 = n - n_1 = \dim \bar{V}_h^{(2)}$ is equal to the number of interior edges plus twice the number of interior faces of the triangulation.

Let $\{\Phi_i\}_{i=1}^n$ be a basis of $V_h^{(2)}$ and $\{\varphi_l\}_{l=1}^m$ be a basis of $W_h^{(2)}$. Then (3) defines the matrix eigenvalue problem

$$A\mathbf{x} = \lambda M\mathbf{x}, \quad C^T\mathbf{x} = \mathbf{0}, \quad (6)$$

where A and M are n -by- n and C is n -by- m . We partition the matrices A , M , and C according to the hierarchical bases,

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}, \quad M = \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix}, \quad C = \begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix}, \quad (7)$$

with $A_{ij}, B_{ij} \in \mathbb{R}^{n_i \times n_j}$ and $C_{ij} \in \mathbb{R}^{n_i \times m_j}$. The (1,1)-blocks correspond to the bilinear forms involving basis functions in $V_h^{(1)}$ and $W_h^{(1)}$, respectively.

Formally, spaces consisting of common Lagrange finite elements could be chosen for V_h and W_h in (3). However, Nédélec's edge elements have the crucial advantage over Lagrange finite elements that [14, §5.3]

$$\mathbf{grad} W_h^{(k)} = \left\{ \mathbf{v}_h \in V_h^{(k)} \mid \mathbf{curl} \mathbf{v}_h = \mathbf{0} \right\}. \quad (8)$$

In a way similar to the continuous case [16], we can decompose any $\mathbf{v}_h \in V_h^{(k)}$ in the form $\mathbf{v}_h = (\mathbf{v}_h - \mathbf{grad} q_h) + \mathbf{grad} q_h =: \mathbf{w}_h + \mathbf{grad} q_h$, where $q_h \in W_h^{(k)}$ is defined by

$$(\mathbf{grad} q_h, \mathbf{grad} \varphi) = (\mathbf{v}_h, \mathbf{grad} \varphi), \quad \forall \varphi \in W_h^{(k)}.$$

Then, $\mathbf{v}_h = \mathbf{w}_h + \mathbf{grad} q_h$ splits $V_h^{(k)}$ – in a *discrete* sense – into a divergence-free and a curl-free part. We write this decomposition as

$$V_h^{(k)} = W_{0,h}^{(k)} \oplus \mathbf{grad} W_h^{(k)}. \quad (9)$$

For $\mathbf{w}_h \in W_{0,h}^{(k)} \setminus \{\mathbf{0}\}$ we have $\|\mathbf{curl} \mathbf{w}_h\|_{0,\Omega} > 0$. Notice, that $W_{0,h}^{(k)}$ is not contained in W_0 . We therefore cannot expect monotonic convergence of the eigenvalues. Equation (8) corresponds to the ‘non-pollution condition’ for the conforming finite element approximation of the model problem by Gruber and Rappaz [17, p.20]. Therefore, the infinitely multiple eigenvalue zero of $\mathbf{curl} \mathbf{curl}$ is not spread over the real axis but remains confined in the origin.

In order that $\mathbf{u}_h = \sum_{j=1}^n \xi_j \Phi_j$ is in $W_{0,h}^{(k)}$ we must have

$$(\mathbf{grad} \varphi_l, \mathbf{u}_h) = \sum_{j=1}^n (\mathbf{grad} \varphi_l, \Phi_j) \xi_j = 0, \quad 1 \leq l \leq m,$$

or, equivalently,

$$C^T \mathbf{x} = \mathbf{0}, \quad \mathbf{x} = (\xi_1, \dots, \xi_n)^T,$$

which is the second equation in (6). Because of (8) we can write $\mathbf{grad} \varphi_l = \sum_{j=1}^n \eta_{jl} \Phi_j$. Therefore,

$$(\Phi_i, \mathbf{grad} \varphi_l) = \sum_{j=1}^n (\Phi_i, \Phi_j) \eta_{jk}, \quad 1 \leq l \leq m,$$

or

$$C = M [\mathbf{y}_1, \dots, \mathbf{y}_m] =: MY, \quad \mathbf{y}_l^T = (\eta_{1l}, \dots, \eta_{ml}). \quad (10)$$

In a similar way one obtains

$$H := C^T Y = Y^T M Y, \quad h_{kl} = (\mathbf{grad} \varphi_k, \mathbf{grad} \varphi_l).$$

H is the system matrix that is obtained when solving the Poisson equation with the Lagrange finite elements $W_h^{(k)}$.

If $\mathbf{a} = (\alpha_1, \dots, \alpha_m)^T$ is the vector of coordinates of φ in the basis $\{\varphi_i\} \subset W_h^{(k)}$ then $\mathbf{x} = Y\mathbf{a}$ are the coordinates of $\mathbf{grad} \varphi$ in the basis $\{\Phi_j\} \subset V_h^{(k)}$. On the other hand, if $\Phi = \sum_{j=1}^n \xi_j \Phi_j \subset V_h^{(k)}$ then $\mathbf{grad} \varphi = \sum_{i=1}^m \alpha_i \varphi_i$ with $\mathbf{a} = H^{-1} C^T \mathbf{x}$ is the part of Φ with non-zero divergence. So, $\mathbf{x} = (I - YH^{-1}C^T)\mathbf{x} + YH^{-1}C^T\mathbf{x}$ is the vector form of the Helmholtz decomposition (9).

Notice that Y is very sparse. We have already mentioned that the gradient of a basis function $\varphi_k \in \bar{W}_h^{(2)}$ is an element of the first set of basis functions of $\bar{V}_h^{(2)}$. So, m_1 rows of Y have a single entry 1. The gradient of the piecewise linear basis function corresponding to vertex k , say, is a linear combination (with coefficients ± 1) of the basis functions of $V_h^{(1)}$ whose corresponding edge has vertex k as one of its endpoints.

Equation (10) means that $C^T \mathbf{x} = \mathbf{0}$ is equivalent to requiring \mathbf{x} to be M -orthogonal to the eigenspace $\mathcal{N}(A) = \mathcal{R}(Y)$ corresponding to the eigenvalue 0. Thus, the solutions of (6) are precisely the eigenpairs of

$$A\mathbf{x} = \lambda M\mathbf{x} \quad (11)$$

corresponding to the *positive* eigenvalues. We could therefore compute the desired eigenpairs $(\lambda_j, \mathbf{x}_j)$ of (6) by means of (11) alone. The computed eigenvectors corresponding to positive eigenvalues would automatically satisfy the constraint $C^T \mathbf{x}_j = \mathbf{0}$. This is actually done if the linear systems of the form $(A - \sigma M)\mathbf{x} = M\mathbf{y}$ that appear in the eigensolver can be solved by direct methods. Numerical experiments showed however that the high-dimensional zero eigenspace has a negative effect on the convergence rates if preconditioned *iterative* solvers have to be used.

IV. SOLVING THE MATRIX EIGENVALUE PROBLEM

The classic method for computing a few selected eigenpairs of $A\mathbf{x} = \lambda M\mathbf{x}$ is subspace iteration [10, 18]. Its linear convergence rate is however too low to be competitive with modern algorithms. The Lanczos algorithm in turn converges exponentially [10, 11]. Its main drawback is that it requires memory space proportional to the number of iteration steps that have to be made until convergence. These huge memory requirements can cause the Lanczos algorithm to stop prematurely before convergence has taken place. These well-known facts have been verified for the rectangular cavity [7]. The algorithms that successfully solved all problem sizes were the implicitly restarted Lanczos algorithm [8, 19, 20], and the Jacobi-Davidson algorithm [9, 21, 22]. Like Lanczos these algorithms converge superlinearly but like subspace iteration their memory requirements are bounded by the order of the problem n times a small multiple of the desired eigenvalues p .

In this study we use the implicitly restarted Lanczos algorithm and the Jacobi-Davidson algorithm as described in [7] to solve the matrix eigenvalue problem (6) that we write in the form

$$M^{-1}A\mathbf{x} = \lambda\mathbf{x}, \quad C^T \mathbf{x} = Y^T M\mathbf{x} = \mathbf{0}. \quad (12)$$

Note that $M^{-1}A$ is symmetric (self-adjoint) with respect to the inner product $\langle \mathbf{x}, \mathbf{y} \rangle = \mathbf{x}^T M\mathbf{y}$. Admissible vectors \mathbf{x} are orthogonal to the nullspace $\mathcal{N}(A) = \mathcal{R}(Y)$ of A . Thus, $M^{-1}A$ is positive definite as an operator from $\mathcal{R}(Y)^{\perp M}$ onto itself. The lowest part of its spectrum is desired.

A Spectral transformations

When computing a few, say p , eigenvalues of (12) closest to a number τ it is essential to apply a spec-

tral transformation to get a reasonable speed of convergence [11, 23, 24, 25]. With the shift-and-invert spectral transformation (12) becomes

$$(A - \sigma M)^{-1} M\mathbf{x} = \mu\mathbf{x}, \quad C^T \mathbf{x} = \mathbf{0}, \quad \mu = \frac{1}{\lambda - \sigma}. \quad (13)$$

The shift σ is chosen close to τ . The spectral transformation leaves the eigenvectors unchanged. The eigenvalues of (12) close to the shift σ become the largest absolute of (13). In addition they are relatively well-separated which improves the speed of convergence of the eigensolvers [10]. The cost of the improved convergence rate is solving a system of equations of the form

$$(A - \sigma M)\mathbf{x} = M\mathbf{y}, \quad C^T \mathbf{x} = \mathbf{0}, \quad (14)$$

in each iteration step of the eigensolver. We assume that A and M are so big that they cannot be factored such that (14) has to be solved iteratively. In the subsequent sections we are going to discuss ways how to do this and, in particular, how to precondition (14).

B Implicitly Restarted Lanczos algorithm (IRL)

To overcome the memory consumption of the Lanczos algorithm it has to be restarted in some way. Sorensen [8] suggested an elegant way to do so. He showed how to extract a Krylov subspace with given spectral properties from the actual trial space (also Krylov subspace). These ideas that apply to the Arnoldi algorithm for non-symmetric eigenvalue problems as well are implemented in the Fortran subroutine package ARPACK [26].

The Lanczos iteration process with shift-and-invert spectral transformation is executed until $j = p + k$, where k is some positive integer. We used $k = p$ as suggested in [26]. Complete reorthogonalization to guarantee stability is feasible as by assumption $p + k$ is not big. If $j = p + k$ then the dimension of the trial space is reduced to p with a procedure related to the implicit shifted QR algorithm. The shifts are chosen to be the k eigenvalues of the tridiagonal matrix produced by the Lanczos iteration process that are furthest away from the target τ . As the p dimensional subspace left after the above procedure is again a Krylov subspace the Lanczos iteration process can be restarted.

In IRL, the system of equations (14) has to be solved to high accuracy as otherwise the Lanczos relation among the basis vectors does not hold true.

Besides the storage for the matrices A and M , the memory requirements of IRL are essentially the space needed to store $2n(p + k)$ floating point variables.

C Jacobi-Davidson algorithm (JD)

The Jacobi-Davidson algorithm has been introduced by Sleijpen and van der Vorst [9] for solving arbitrary

generalized eigenvalue problems. We have adapted their algorithm JDQZ to the generalized eigenvalue problem with symmetric A and symmetric positive definite M [7].

The Jacobi-Davidson algorithm is not a Krylov subspace method. The trial space spanned by the M -orthogonal columns of $V_j \in \mathbb{R}^{n \times j}$ used in JD is extended by means of the solution \mathbf{t} of the *correction equation*

$$\begin{aligned} (I - \tilde{W}_k \tilde{Q}_k^T)(A - \tilde{\lambda}M)(I - \tilde{Q}_k \tilde{W}_k^T)\mathbf{t} &= -\mathbf{r}, \\ (I - \tilde{Q}_k \tilde{W}_k^T)\mathbf{t} &= \mathbf{t}, \end{aligned} \quad (15)$$

where $\tilde{Q}_k = [\mathbf{q}_1, \dots, \mathbf{q}_k, \tilde{\mathbf{q}}]$ and $\tilde{W}_k = M\tilde{Q}_k$. Here, $(\tilde{\lambda}, \tilde{\mathbf{q}})$ is a Ritz pair in the actual trial space and $\mathbf{q}_1, \dots, \mathbf{q}_k$ are the already computed eigenvectors. Initially, $k = 0$. Notice, that $(I - \tilde{W}_k \tilde{Q}_k^T)\mathbf{r} = \mathbf{r}$ such that (15) is consistent if $A - \tilde{\lambda}M$ is invertible. The solution \mathbf{t} of (15) is M -orthogonalized against the columns of V_j and appended to V_j to obtain V_{j+1} .

If $j = j_{\max}$ the dimension of the trial space is reduced to $j = j_{\min} \geq p$ by choosing the j_{\min} most desirable Ritz vectors from $\mathcal{R}(V_{j_{\max}})$.

If (15) is solved exactly, then one step of the algorithm turns out to be one step of the Rayleigh quotient iteration [9] which converges cubically for symmetric eigenvalue problems [10]. If (15) is solved only approximately this high convergence rate gets lost. There is a trade-off between speed of convergence and the amount of work one is willing to spend for solving (15).

If equation (15) is solved approximately by a Krylov subspace method a good preconditioner has to be employed to get a satisfactory convergence rates. Notice that the matrix in (15) maps on the orthogonal complement of \tilde{Q}_k instead of \tilde{W}_k . Therefore, the preconditioner also has the duty to map the solutions in the correct subspace. In a similar way as Fokkema et al. [22] we use a preconditioner of the form

$$(I - \tilde{W}_k \tilde{Q}_k^T)K(I - \tilde{Q}_k \tilde{W}_k^T) : \mathcal{R}(\tilde{W}_k)^\perp \longrightarrow \mathcal{R}(\tilde{Q}_k)^\perp$$

where the symmetric matrix K is a good and easily invertible approximation of $A - \tilde{\lambda}M$. It is straightforward to show that the inverse of this mapping is given by

$$(I - \tilde{Y}_k(\tilde{W}_k^T \tilde{Y}_k)^{-1} \tilde{W}_k^T)K^{-1}(I - \tilde{W}_k \tilde{Q}_k^T) \quad (16)$$

where $\tilde{Y}_k = K^{-1}\tilde{W}_k$ [16]. The preconditioned (15) equation then becomes

$$\begin{aligned} (I - \tilde{Y}_k(\tilde{W}_k^T \tilde{Y}_k)^{-1} \tilde{W}_k^T)K^{-1}(I - \tilde{W}_k \tilde{Q}_k^T)(A - \tilde{\lambda}M)\mathbf{t} \\ = (I - \tilde{Y}_k(\tilde{W}_k^T \tilde{Y}_k)^{-1} \tilde{W}_k^T)K^{-1}(A - \tilde{\lambda}M)\mathbf{t} \\ = -(I - \tilde{Y}_k(\tilde{W}_k^T \tilde{Y}_k)^{-1} \tilde{W}_k^T)K^{-1}\mathbf{r}, \\ \tilde{W}_k^T \mathbf{t} = 0. \end{aligned} \quad (17)$$

Memory space is needed for the blocks \tilde{Q}_k , \tilde{W}_k , \tilde{Y}_k , and V . Thus, besides the storage for the matrices $A - \sigma M$ and M , $(3p + j_{\max})n$ memory locations are needed.

V. SOLVING THE CONSTRAINED SYSTEM OF EQUATION I: THE STRAIGHTFORWARD APPROACH

We investigate two ways how to solve (14). In the straightforward approach of this section we apply the conjugate gradient method forcing the iterates into $\mathcal{R}(C)^\perp = \mathcal{R}(Y)^\perp$. In the second approach of section VI we treat (14) as an augmented system of order $n + m$.

We exploit that

$$M^{-1}(A - \sigma M) : \mathcal{R}(Y)^\perp \longrightarrow \mathcal{R}(Y)^\perp \quad (18)$$

is symmetric positive definite and apply the preconditioned conjugate gradient (cg) method to solve $M^{-1}(A - \sigma M)\mathbf{x} = \mathbf{y}$. Applying the preconditioner M^{-1} amounts to approximately solving

$$(A - \sigma M)\mathbf{x} = M\mathbf{y}. \quad (19)$$

Notice that both M and $A - \sigma M$ map $\mathcal{R}(Y)^\perp$ one-to-one onto $\mathcal{R}(Y)^\perp$. Two problems arise if (19) is solved as it stands. First, $(A - \sigma M)$ as a map of the whole n -space is indefinite and is bound to slow down convergence [27]. Second, if (19) is not solved to high accuracy the approximate solution will fail to satisfy the constraints in (14). By choosing the preconditioner

$$M^{-1}P_{\mathcal{R}(Y)^\perp}^T := M^{-1}(I - CH^{-1}Y^T)$$

the cg iterates stay in the correct space. In the Jacobi-Davidson algorithm the approximate solution \mathbf{t} of (15) is additionally forced to be orthogonal to the Ritz vector $\tilde{\mathbf{q}}$ and to the previously computed eigenvectors.

Here, we are mainly interested in two-level preconditioners induced by hierarchical finite element basis. Proceeding as Bank [28] we approximate A , M , and C partitioned as in (7) by the block diagonal

$$\begin{aligned} A_1 &= \begin{bmatrix} A_{11} & O \\ O & A_{22} \end{bmatrix}, & M_1 &= \begin{bmatrix} M_{11} & O \\ O & M_{22} \end{bmatrix}, \\ C_1 &= \begin{bmatrix} C_{11} & O \\ O & C_{22} \end{bmatrix}. \end{aligned} \quad (20)$$

A_1 , M_1 , and C_1 have similar properties as A , M , and C ; in particular, $A_1 Y = O$ and $C_1 = M_1 Y$. Further, $A_1 - \sigma_1 M_1$ maps $\mathcal{R}(Y)^\perp$ onto $\mathcal{R}(Y)^\perp$. Now we define a preconditioner

$$M^{-1}(A_1 - \sigma_1 M_1)P_{\mathcal{R}(Y)^\perp} : \mathcal{R}(Y)^\perp \longrightarrow \mathcal{R}(Y)^\perp \quad (21)$$

with $P_{\mathcal{R}(Y)^\perp} = I - YH_1^{-1}C_1^T$ and $H_1 = C_1^T Y$. The inverse of this preconditioner (as a mapping from $\mathcal{R}(Y)^\perp$ onto itself) is

$$\begin{aligned} P_{\mathcal{R}(Y)^\perp} (A_1 - \sigma_1 M_1)^{-1} M \\ = (I - YH_1^{-1}C_1^T)(A_1 - \sigma_1 M_1)^{-1} M. \end{aligned} \quad (22)$$

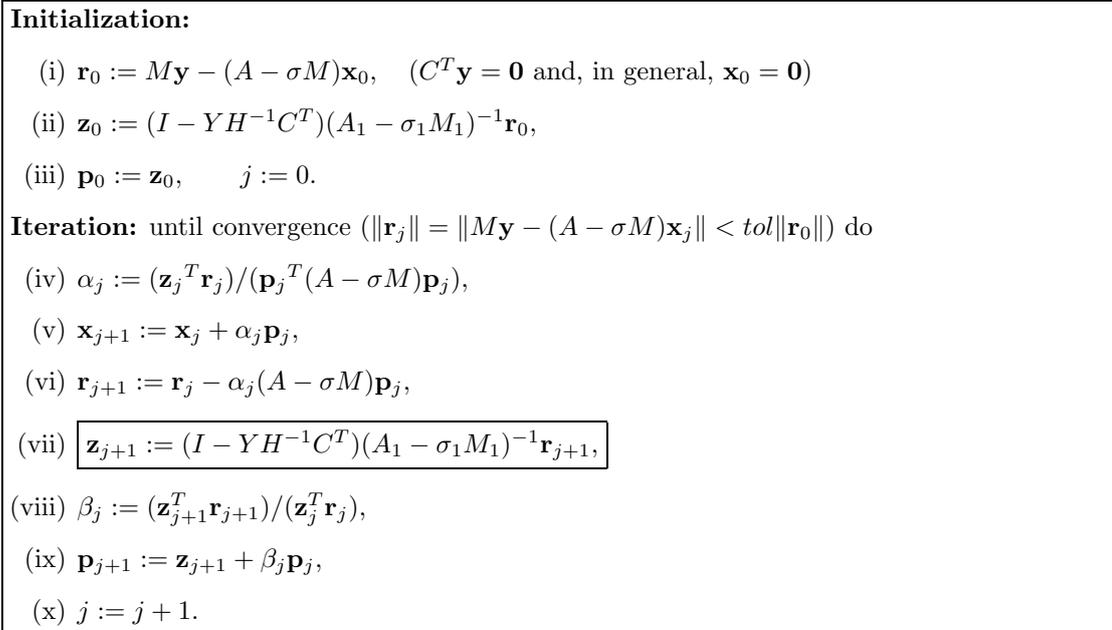


FIG. 2: Preconditioned conjugate gradient method for $(A - \sigma M)\mathbf{x} = M\mathbf{y}$

The preconditioned conjugate gradient method [29, p.247] for solving (14) with preconditioner (21) is displayed in Fig. 2.

Notice that the M -inner products have disappeared. Operations with M occur in the computation of the residuals and implicitly in the M -orthogonal projector. In the algorithm in Fig. 2 the various iterates satisfy

$$Y^T \mathbf{r}_j = C^T \mathbf{z}_j = C^T \mathbf{p}_j = C^T \mathbf{x}_j = \mathbf{0} \quad \text{for all } j,$$

such that the approximation \mathbf{x}_j is in the correct space whenever we stop the iteration. Step (vii) in Fig. 2 consumes most of the execution time. Two systems of equations have to be solved (1) with $A_1 - \sigma_1 M_1$ and (2) with H inside the projector. In fact, we use a direct solver for the (1,1) block $A_{11} - \sigma_1 M_{11}$ and just execute one step of the Jacobi or symmetric Gauss-Seidel iteration for approximating the inverse of $A_{22} - \sigma_1 M_{22}$, see Bank [28]. For solving Poisson's equation with H we use a direct method as well.

VI. SOLVING THE CONSTRAINED SYSTEM OF EQUATION II: THE AUGMENTED SYSTEM APPROACH

An alternative way to write (14) is as an augmented linear system,

$$\begin{bmatrix} A - \sigma M & C \\ C^T & O \end{bmatrix} \begin{bmatrix} \mathbf{x}' \\ \mathbf{x}'' \end{bmatrix} = \begin{bmatrix} M\mathbf{y} \\ \mathbf{0} \end{bmatrix}. \quad (23)$$

If $A - \sigma M$ is nonsingular then the block LU factorization of the 2-by-2 block matrix in (23) immediately shows that

$\mathbf{x}'' = \mathbf{0}$. Therefore, the first component \mathbf{x}' of the solution \mathbf{x} of (23) satisfies $(A - \sigma M)\mathbf{x}' = M\mathbf{y}$ by the first equation and the constraints by the second equation.

The symmetric indefinite matrix in equation (23) has order $n+m$. It has n positive and m negative eigenvalues. Solving (23) instead of (19) has the advantage that there are no (explicit) constraints to be taken into account. There are however more unknowns and quite an accurate solution is needed to satisfy the constraints.

We considered preconditioners of the form

$$\begin{bmatrix} A_1 - \sigma_1 M_1 & C \\ C^T & O \end{bmatrix} \quad (24)$$

when solving (23) by the preconditioned conjugate gradient method [16]. A_1 and M_1 have been defined in (20). It turned out that this preconditioner leads to an algorithm that is very similar to the one of the straightforward approach. The difference is that the projector $I - YH^{-1}C^T$ is replaced by the much more complicated $I - K_1^{-1}C(C^T K_1^{-1}C)^{-1}C^T K_1^{-1}$.

We therefore tried out preconditioners of the form

$$\begin{bmatrix} K_1 & C_1 \\ C_1^T & O \end{bmatrix}, \quad K_1 = A_1 - \sigma_1 M_1, \quad C_1 = M_1 Y. \quad (25)$$

We can easily transcribe the algorithm of Fig. 2 into a solver for (23). Each vector in the algorithm has now two components, the first being in \mathbb{R}^n the second in \mathbb{R}^m , see Fig. 3. If \mathbf{x}_0 is chosen such that $C^T \mathbf{x}'_0 = \mathbf{0}$ then traversing stepwise through the algorithm in Fig. 3 one sees that

$$\mathbf{z}''_j = \mathbf{p}''_j = \mathbf{x}''_j = \mathbf{0} \quad \text{for all } j \quad (26)$$

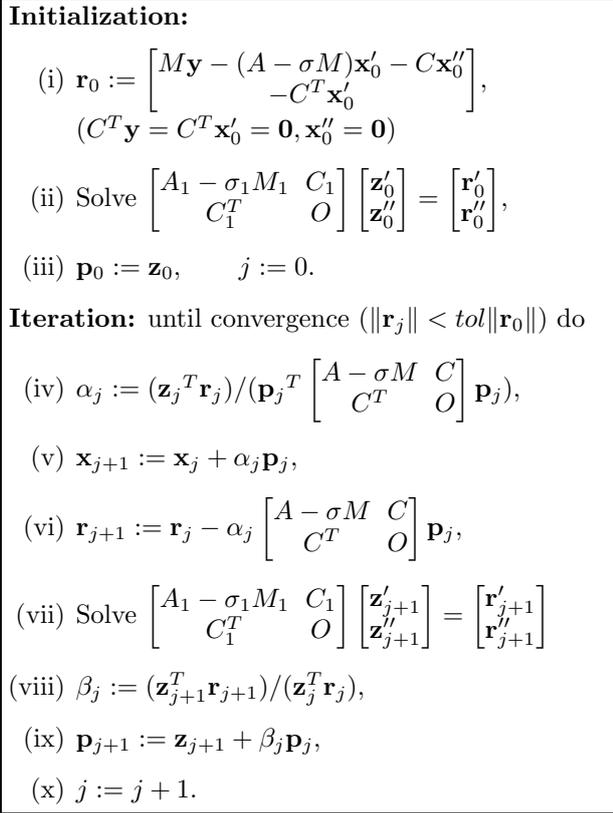


FIG. 3: Preconditioned conjugate gradient method for the augmented linear system

provided that $\sigma = \sigma_1$ which we assume henceforth. To verify (26) we observe that initially $C^T\mathbf{y} = C^T\mathbf{x}'_0 = \mathbf{0}$ such that $Y^T\mathbf{r}'_0 = \mathbf{0}$ and $\mathbf{r}''_0 = \mathbf{0}$. Let us assume that (26) holds for all indices up to j . From step (v) in Fig. 3 we see that $\mathbf{x}''_{j+1} = \mathbf{0}$. From step (vi) we have

$$\begin{aligned} Y^T \mathbf{r}'_{j+1} &= Y^T \mathbf{r}'_j - \alpha_j (-\sigma Y^T M) \mathbf{p}'_j \\ &= Y^T \mathbf{r}'_j + \alpha_j \sigma C^T \mathbf{p}'_j, \\ \mathbf{r}''_{j+1} &= \mathbf{r}''_j - \alpha_j C^T \mathbf{p}'_j. \end{aligned}$$

The second component of the solution of the equation

$$\begin{aligned} \begin{bmatrix} \mathbf{z}'_{j+1} \\ \mathbf{z}''_{j+1} \end{bmatrix} &= \begin{bmatrix} K_1 & C_1 \\ C_1^T & O \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{r}'_{j+1} \\ \mathbf{r}''_{j+1} \end{bmatrix} \\ &= \begin{bmatrix} K_1^{-1} + (1/\sigma)YH_1^{-1}Y^T & YH_1^{-1} \\ H_1^{-1}Y^T & \sigma H_1^{-1} \end{bmatrix} \begin{bmatrix} \mathbf{r}'_{j+1} \\ \mathbf{r}''_{j+1} \end{bmatrix} \end{aligned} \quad (27)$$

in step (vii) of the algorithm is

$$\begin{aligned} \mathbf{z}''_{j+1} &= H_1^{-1} (Y^T \mathbf{r}'_{j+1} + \sigma \mathbf{r}''_{j+1}) \\ &= H_1^{-1} (Y^T \mathbf{r}'_j + \alpha_j \sigma C^T \mathbf{p}'_j + \sigma \mathbf{r}''_j - \alpha_j \sigma C^T \mathbf{p}'_j) \\ &= H_1^{-1} (Y^T \mathbf{r}'_j + \sigma \mathbf{r}''_j) \\ &= \dots = H_1^{-1} (Y^T \mathbf{r}'_0 + \sigma \mathbf{r}''_0) = \mathbf{0}. \end{aligned}$$

Here, we have used that $K_1 = A_1 - \sigma M_1$ is nonsingular and that $K_1 Y = -\sigma C_1$. $\mathbf{p}''_{j+1} = \mathbf{0}$ follows from step (ix) in Fig. 3.

The relations in (26) simplify the algorithm considerably. Only the \mathbf{r}_j have a nonzero second component. It is used in step (vii) of the algorithm to compute \mathbf{z}_j , in fact, \mathbf{z}'_j as $\mathbf{z}''_j = \mathbf{0}$ as we have just seen. From (27) we get

$$\begin{aligned} \mathbf{z}'_{j+1} &= K_1^{-1} \mathbf{r}'_{j+1} + \frac{1}{\sigma} Y H_1^{-1} (Y^T \mathbf{r}'_{j+1} + \sigma \mathbf{r}''_{j+1}) \\ &= K_1^{-1} \mathbf{r}'_{j+1}. \end{aligned}$$

So, actually, \mathbf{r}''_j is never used and thus does not have to be computed at all. It can be recovered at any time as $\mathbf{r}''_j = (-1/\sigma)Y^T \mathbf{r}'_j$.

Taking all the above simplifications into account the algorithm in Fig. 3 with preconditioner (25) becomes Algorithm 4.

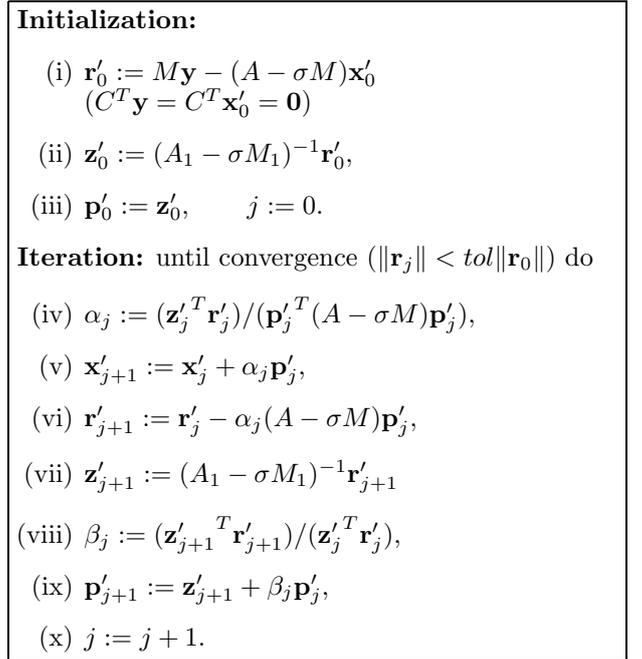


FIG. 4: Preconditioned conjugate gradient method for the augmented linear system after simplifications

Notice that the algorithm in Fig. 4 differs from the one in Fig. 2 in that the projectors $I - YH^{-1}C^T$ are omitted. In fact, in Fig. 4 plain pcg is applied to the consistent but indefinite system of equations $(A - \sigma M)\mathbf{x} = M\mathbf{y}$. It is not yet clear to us if this algorithm is stable at all. Certainly, the constraints $C^T\mathbf{x}'_{j+1} = \mathbf{0}$ are not satisfied before convergence. If we solve systems as in the Jacobi-Davidson algorithm with only low accuracy then we enforce the constraints explicitly at the end of the iteration.

VII. NUMERICAL EXPERIMENTS

We performed a few numerical experiments computing the 10 lowest eigenvalues and corresponding eigenvectors of a model of the future design of the copper

TABLE I: Implicitly Restarted Lanczos (ARPACK) results for the copper cavity. Times are in seconds.

Small problem size: $n = 1880, n_1 = 300, m_1 = 27$					
method, preconditioner	it_{int}	it_{out}	t_{tot}	t_{eig}	t_{11}
projection, none	1666	92	709	707	
projection, diagonal	256	92	114	112	
projection, hierarchical/damped Jacobi	231	92	146	144	35
projection, hierarchical/symmetric Gauss-Seidel	113	92	101	99	17
augmented, hierarchical/damped Jacobi	238	92	149	147	54
augmented, hierarchical/symmetric Gauss-Seidel	133	92	112	110	50
Medium problem size: $n = 27088, n_1 = 4688, m_1 = 507$					
method, preconditioner	it_{int}	it_{out}	t_{tot}	t_{eig}	t_{11}
projection, none	1504	43	17834	17300	
projection, diagonal	329	43	4416	3869	
projection, hierarchical/damped Jacobi	167	43	3340	2803	807
projection, hierarchical/symmetric Gauss-Seidel	81	43	2089	1551	394
augmented, hierarchical/damped Jacobi	160	43	2578	2034	1156
augmented, hierarchical/symmetric Gauss-Seidel	89	43	1982	1460	639
Large problem size: $n = 45040, n_1 = 7824, m_1 = 867$					
method, preconditioner	it_{int}	it_{out}	t_{tot}	t_{eig}	t_{11}
projection, none	1209	43	23880	22284	
projection, diagonal	370	43	8566	6968	
projection, hierarchical/damped Jacobi	135	43	5600	4011	1373
projection, hierarchical/symmetric Gauss-Seidel	68	43	3914	2274	686
augmented, hierarchical/damped Jacobi	129	43	4660	2998	1708
augmented, hierarchical/symmetric Gauss-Seidel	74	43	3797	2229	966

cavity, cf. Fig. 1. We used three different discretizations, a coarse discretization with only $n = 1880$ degrees of freedom, a fine discretization with 27'088 and an even finer discretization with 45'040 degrees of freedom. The corresponding values for n_1 and m_1 , cf. (7), are given in Tables I and II. We combined the implicitly restarted Lanczos algorithm (IRL) and Jacobi-Davidson algorithm (JD) with the straightforward ('projection') or modified augmented approach for solving (14). With IRL we solved the positive definite system of equations in the straightforward approach with the preconditioned conjugate gradient method. With JD the shifts vary dynamically and can get inside the convex hull of the spectrum. That's why we used MINRES as our system solver [29]. The system of equations in the augmented system where the preconditioner is indefinite were solved with a special variant of QMR [30]. In either case, we set $\sigma_1 = \sigma$ to be a value slightly below the lowest positive eigenvalue. When solving (14), we employed no, diagonal or hierarchical basis preconditioning. The last was combined with two different treatments of the A_{22} block. Either we performed one step of damped Jacobi (damping factor 2/3) or one step of symmetric Gauss-Seidel. The results for IRL are summarized in Tab. I those for JD in Tab. II. The number it_{int} provides the average number of inner iterations, i.e., the steps until convergence of the linear solver. As systems are solved to much higher accuracy with IRL (to establish the Lanczos three-term recurrence) the iteration count it_{int} is much bigger in Tab. I than in Tab. II.

In fact, with JD, the convergence tolerance of the inner iteration is initially very loose and becomes more stringent as the outer iteration converges [7]. In contrast, the number of outer iterations it_{out} , i.e., the number of systems that are solved is smaller with IRL. In both IRL and JD we let the dimension of the trial space vary between 10 and 25.

t_{tot} gives the overall solution time, including grid handling, matrix assembly, eigensolver, etc., while t_{eig} is the time spent for solving the eigensystem only. Finally, t_{11} gives the time that is spent for solving (forward/backward substitution) the upper diagonal (1,1) block by a direct solver. t_{eig} includes t_{11} . These two numbers are the times of interest. Our discussion is based on them. Notice that in the JD timings the overall solution time for the augmented approach does not contain the building of matrix C and H while it is still included in the IRL times.

The numerical experiments were run in sequential mode on a Sun Enterprise 3500 with six 336 Mhz UltraSparc processors and 3 GB main memory running the Solaris 2.6 operating system. All times are user time.

In all examples, the Jacobi-Davidson algorithm is the faster eigensolver than the implicitly restarted Lanczos algorithm. IRL executes many more steps in the inner iteration as the Lanczos vectors have to be computed to quite a high precision [26].

Higher sophistication in the preconditioner decreases not only the iteration counts but also the execution times.

TABLE II: Jacobi-Davidson results for the copper cavity. Times are in seconds.

Small problem size: $n = 1880, n_1 = 300, m_1 = 27$					
method, preconditioner	it_{int}	it_{out}	t_{tot}	t_{eig}	t_{11}
projection, none	290	142	423	421	
projection, diagonal	49	118	66	64	
projection, hierarchical/Jacobi	42	121	61	59	9
projection, hierarchical/symmetric Gauss-Seidel	20	115	35	33	4
augmented, hierarchical/Jacobi	—	—	—	—	—
augmented, hierarchical/symmetric Gauss-Seidel	39	114	37	36	7
Medium problem size: $n = 27088, n_1 = 4688, m_1 = 507$					
method, preconditioner	it_{int}	it_{out}	t_{tot}	t_{eig}	t_{11}
projection, none	103	119	5333	4805	
projection, diagonal	26	106	1673	1155	
projection, hierarchical/Jacobi	14	109	1451	928	204
projection, hierarchical/symmetric Gauss-Seidel	7	104	1092	564	112
augmented, hierarchical/Jacobi	24	101	868	785	317
augmented, hierarchical/symmetric Gauss-Seidel	14	101	683	600	198
Large problem size: $n = 45040, n_1 = 7824, m_1 = 867$					
method, preconditioner	it_{int}	it_{out}	t_{tot}	t_{eig}	t_{11}
projection, none	77	114	7070	5513	
projection, diagonal	27	107	3566	1987	
projection, hierarchical/Jacobi	12	107	2934	1354	368
projection, hierarchical/symmetric Gauss-Seidel	6	102	2453	866	212
augmented, hierarchical/Jacobi	20	97	1537	1300	562
augmented, hierarchical/symmetric Gauss-Seidel	12	98	1269	1034	366

This becomes more prominent as the problem size increases. This is in contrast to the experiments that we conducted with node elements in the box shaped cavity [31]. The analysis of the hierarchical basis preconditioner states that it_{int} should not increase with the problem size. Here it even decreases substantially.

When comparing the straightforward with the (modified) augmented system approach, IRL and JD behave quite differently. With IRL we observe a big gain of more than 25% when employing the augmented system approach combined with damped Jacobi smoothing. The gain is almost negligible if Jacobi is replaced by symmetric Gauss-Seidel. The average number of inner iterations differs only slightly in the two approaches. With Jacobi-Davidson there is a minor gain with damped Jacobi smoothing and a considerable loss with symmetric Gauss-Seidel smoothing. The average number of inner iterations is twice as high for the augmented system approach. The reduced cost per iteration by saving the projector is compensated by a higher number of iterations.

It is worth noting that the time t_{11} for solving the (1,1) block takes a considerable fraction of the solution of the eigenvalue problem. t_{11} is about 20% of t_{eig} in the medium sized problems and about 25% in the large problems. With the augmented system of equation this portion is even above 50%. In this case, the direct solver appears to not yet having been implemented optimally. In any case, the present two-level preconditioner will not

suffice for larger problem sizes. A multi-level method will be necessary for solving the system of equations corresponding to the (1,1) block $A_{11} - \sigma M_{11}$.

VIII. CONCLUSIONS

We have investigated algorithms for solving the eigenvalue problem occurring in the design of resonant cavities. The governing Maxwell equations for the electric field intensity have been discretized by quadratic Nédélec edge elements. The finite element spaces had dimensions up to 45'040. The resulting symmetric matrix eigenvalue problems have been solved by two methods, the Jacobi-Davidson (JD) and the implicitly restarted Lanczos (IRL) algorithm. In both methods, a shift-and-invert spectral transformation has been applied. The such introduced symmetric indefinite systems of equations have been solved iteratively by a Krylov subspace method. The most effective preconditioners with respect to time and iteration numbers were hierarchical basis preconditioners. The average number of iterations to solve these systems not only stayed constant but decreased with increasing problem size.

The JD algorithm clearly outperformed IRL. The execution times of the latter are about twice as long due to the high accuracy requirement of the construction of a correct three-term recurrence.

The symmetric indefinite systems of equations have

been solved in two ways. First, by projecting in the subspace where the system is positive definite, second, by augmenting the system, i.e., by introducing constraints and Lagrange multipliers. With the first we obtained very satisfactory results. The latter gave good results with IRL but lagged clearly behind with JD. The augmented systems approach is not yet completely understood with regard to its convergence behavior and stability. Further investigations in this direction are required.

REFERENCES

- [1] J. D. Jackson, *Classical Electrodynamics* (Wiley, New York, 1975), 2nd ed.
- [2] F. Kikuchi, *Comput. Methods Appl. Mech. Eng.* **64**, 509 (1987).
- [3] J. Jin, *The Finite Element Method in Electromagnetics* (Wiley, New York, 1993).
- [4] J. C. Nédélec, *Numer. Math.* **35**, 315 (1980).
- [5] J. P. Webb, *IEEE Transactions on Magnetics* **29**(2), 1460 (1993).
- [6] G. Mur, *IEEE Transactions on Magnetics* **30**(5), 3552 (1994).
- [7] P. Arbenz and R. Geus, *Numer. Lin. Alg. Appl.* **6**, 3 (1999).
- [8] D. C. Sorensen, *SIAM J. Matrix Anal. Appl.* **13**, 357 (1992).
- [9] G. L. G. Sleijpen and H. A. van der Vorst, *SIAM J. Matrix Anal. Appl.* **17**, 401 (1995).
- [10] B. N. Parlett, *The Symmetric Eigenvalue Problem* (Prentice Hall, Englewood Cliffs, NJ, 1980), (Republished by SIAM, Philadelphia, 1998.).
- [11] R. Grimes, J. G. Lewis, and H. Simon, *SIAM J. Matrix Anal. Appl.* **15**, 228 (1994).
- [12] P. G. Ciarlet, *The Finite Element Method for Elliptic Problems* (North-Holland, Amsterdam, 1978), (Studies in Mathematics and its Applications, 4).
- [13] A. Bossavit, *Computational electromagnetism* (Academic Press, San Diego, 1998).
- [14] V. Girault and P.-A. Raviart, *Finite Element Methods for the Navier-Stokes Equations* (Springer-Verlag, Berlin, 1986), (Springer Series in Computational Mathematics, 5).
- [15] P. P. Silvester and R. L. Ferrari, *Finite Elements for Electrical Engineers* (Cambridge University Press, Cambridge, 1996), 3rd ed.
- [16] P. Arbenz, *On computing some of the principal eigenfrequencies of cavity resonators* (2000), unpublished manuscript.
- [17] R. Gruber and J. Rappaz, *Finite Element Methods in Linear Ideal Magnetohydrodynamics* (Springer-Verlag, Berlin, 1985).
- [18] H. Rutishauser, *Numer. Math.* **16**, 205 (1970), reprinted in: *Linear Algebra*, J.H. Wilkinson, C. Reinsch (eds.), pp. 284–301, Springer, Berlin, 1971.
- [19] D. Calvetti, L. Reichel, and D. C. Sorensen, *Electronic Trans. Numer. Anal.* **2**, 1 (1994).
- [20] J. Baglama, D. Calvetti, and L. Reichel, *BIT* **36**, 400 (1996).
- [21] G. L. G. Sleijpen, A. G. L. Booten, D. R. Fokkema, and H. A. van der Vorst, *BIT* **36**, 595 (1996).
- [22] D. R. Fokkema, G. L. G. Sleijpen, and H. A. van der Vorst, *SIAM J. Sci. Comput.* **20**, 94 (1998).
- [23] D. S. Scott, *SIAM J. Sci. Stat. Comput.* **3**, 68 (1982).
- [24] T. Ericsson, in *Matrix Pencils*, edited by B. Kågström and A. Ruhe (Springer, Berlin, 1983), pp. 177–188, (Lecture Notes in Mathematics, 973).
- [25] B. Nour-Omid, B. N. Parlett, T. Ericsson, and P. S. Jensen, *Math. Comp.* **48**(178), 663 (1987).
- [26] R. B. Lehoucq, D. C. Sorensen, and C. Yang, *ARPACK Users' Guide: Solution of Large-Scale Eigenvalue Problems by Implicitly Restarted Arnoldi Methods* (SIAM, Philadelphia, PA, 1998), (The software and this manual are available at URL <http://www.caam.rice.edu/software/ARPACK/>).
- [27] C. C. Paige, B. N. Parlett, and H. A. V. der Vorst, *Numerical Linear Algebra with Applications* **2**, 115 (1995).
- [28] R. E. Bank, *Acta Numerica* **5**, 1 (1996).
- [29] Y. Saad, *Iterative Methods for Sparse Linear Systems* (PWS Publishing Company, Boston, MA, 1996).
- [30] R. Freund and N. M. Nachtigal, *Appl. Numer. Math.* **19**, 319 (1995).
- [31] P. Arbenz and S. Adam, in *Proceedings of the International Computational Accelerator Physics Conference (ICAP'98)*, edited by K. Ko and R. Ryne (1999), 5 pages (Compact Disc). Available from <http://www.inf.ethz.ch/~arbenz/ICAP98.ps.gz>.