CALCULATION OF EIGENMODES IN SUPERCONDUCTING CAVITIES ON APE–SUPERCOMPUTERS USING A SOFTWARE INTERFACE TO MAFIA

F. Neugebauer, DESY Zeuthen, Platanenallee 6, D-15738 Zeuthen, Germany
U. van Rienen, Universität Rostock, Fachbereich Elektrotechnik und Informationstechnik,
Albert–Einstein–Straße 2, D-18051 Rostock, Germany

Abstract

To design modern accelerators a profound knowledge of eigenmodes of RF-cavities is required. For normal conducting as well as for superconducting cavities MAFIA is a well established tool to determine the eigenmodes by numerical means. However, the 3-dimensional treatment of multicell cavities lacks from available computer power on a usual high end workstation. Therefore the present approach uses a parallel SIMD supercomputer (APE-100) to compute the numerical expensive part of the MAFIA-algorithm. The system matrix, incorporating geometry and material information, is transfered to the APE-100 during a normal MAFIA-session using a command provided by the MAFIA toolkit (MTK). Then, on the APE-100 the lowest eigenvalues and their corresponding eigenvectors of the system matrix are determined by means of a conjugate gradient algorithm [3]. The result of the diagonalization procedure is then read back to the MAFIA host where further data analysis and visualization can be done.

1 INTRODUCTION

The construction of modern accelerators is usually supported by the numerical determination of eigenmodes in the accelerating cavities. Often the rotational symmetry of the cavity is used to simplify the numerical simulation. However, in cases where the cavity plus attached rf–components lacks rotational symmetry a fully 3–dimensional treatment of Maxwell's equations is necessary which requires more computer power than is available on a normal high end workstation. In addition the 3–dimensional approach allows for the simulation of fabrication errors and surface roughness which are usually not considered to have rotational symmetry.

In the framework of the Finite Integration Technique (FIT) developed by Weiland and coworkers[1] Maxwell's equations in integral representation are transformed to a set of matrix equations. Using rectangular grids the discretization volume is partitioned in two sets of cells which can be considered dual. In the case of determining the eigenmodes of a cavity the grid voltages along neighboring gridpoints are the degrees of freedom of the resulting eigenvalue problem. It turns out that the so–called system matrix connects grid voltages of a single cell only to grid voltages of adjacent cells. This "next neighbor connection"–property makes the eigenvalue problem especially well suited to be solved on an APE–100 supercomputer for this type of computer is capable of a very fast data exchange between neighboring nodes.

APE–100 supercomputers are mainly used for in QCD theory where a profound experience in solving eigenvalue problems [3] does exist. However, the parallel structure of the computer requires the use of special programming tools and a language (TAO) dedicated to the computer topology which is inefficient in programming advanced file IO, string evaluation and managing pointers.

Therefore the parsing of the geometry input, which is mainly a linear task, is left to MAFIA which is running on a usual workstation. The resulting system matrix incorporating geometry and material information is transfered to the APE–100 by means of the MAFIA toolkit (MTK). Then, on the APE–100 supercomputer the numerical expensive task of finding the lowest eigenvalues and corresponding eigenvectors of a large sparse matrix is performed. The result of the diagonalization procedure is then read back to the MAFIA host where further data analysis and visualization can be done.

The paper is organized as follows: in section 2 a short overview of the Finite Integration Technique and the APE–100 topology is given. In the next section the matrix vector multiplication which is crucial for the used algorithm is considered in detail. In section 4 a performance analysis of the conjugate gradient algorithm used in this approach is given.

2 FINITE INTEGRATION TECHNIQUE AND THE APE–100 TOPOLOGY

The Finite Integration Technique is based on a discretization of Maxwell's equations using a set of two rectangular grids which can be considered dual to each other [1]. The integral representation of Maxwell's equations is transferred to a discrete version by specifying the integration paths as to be along the edges of the discretization cell. For the case of area integrals the 6 bordering rectangles of the cell are chosen as the integration area. The degrees of freedom in the discretized version of Maxwell's equations are not the fields itself, moreover for example the grid voltage along neighboring grid points or the flux over a cell border are used. Therefore the discretized Maxwell's equations remain mathematically equivalent to the continuous case. There is no discretization error and the discretized Maxwell's equations exactly obey the conservation law for charge and current density.

An important feature of the matrix equations is its locality. Actually this is due to the fact that the chosen integration space is restricted to the neighboring cells of the selected degree of freedom. As a consequence the resulting system matrix which eigensystem has to be determined is sparse with a priori known pattern of entries. A detailed treatment of the theory yields that 13 elements of the system matrix are non zero for each degree of freedom. These elements connect to degrees of freedom belonging to neighboring cells.

The locality of the matrix equations can be exploited on APE–100 in a quite natural way. The nodes of the APE–100 supercomputer are arranged on a three–dimensional rectangular grid (see Fig.1) as is the grid used for the discretization of Maxwell's equations. The cells of the discretization volume are distributed to the nodes so that each processor is responsible for its own segment of real space.

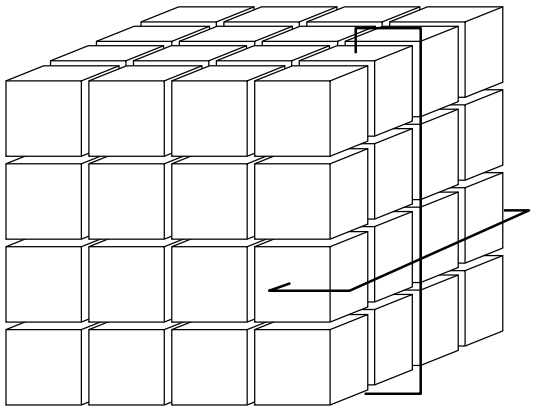

Figure 1: Topology of the APE–100 (4x4x4 nodes). Between adjacent processors there is a fast data transfer which is about 4 times slower than local memory access. The whole cube is subject to periodic boundary conditions in x–,y– and z–direction resulting in a hyper torus. Due to the SIMD character of the APE–100 no latency time occurs on data transmission.

The matrix–vector multiplication is then mainly a local operation on each node. Only in cases where the cell lies on the segment boundary data exchange with neighboring nodes will occur. The APE–100 is perfectly suited to such a situation because the SIMD character of the supercomputer accounts for a very fast data transfer with neighboring nodes without latency.

One drawback of numerical determination of eigenmodes is that the solution space is composed of two solution spaces – one holding the eigenmodes which are searched for and one holding the so–called ghost modes. For FIT there is a workaround excellently described in [2]. Another difficulty arises from the periodic boundary conditions which are built in to the APE–100 topology but are not implemented in the MAFIA package for all coordinate directions. Therefore boundary flags are necessary in the present approach.

3 DATA DISTRIBUTION STRATEGY OF THE SYSTEM MATRIX

Heart of the conjugate gradient algorithm described in[3] is the Ritz method applied to the functional μ

$$\mu(\vec{z}) = \frac{\langle \vec{z}, \hat{A}\vec{z} \rangle}{\langle \vec{z}, \vec{z} \rangle}, \tag{1}$$

where A denotes the system matrix and z is the vector of grid voltages. Given a random initial vector the algorithm searches for the minimum of μ in the orthogonal subspace of all previously determined eigenvectors. Acceleration of the algorithm is achieved by using exact diagonalization in the subspace spanned by the numerically computed eigenvectors.

As usual for algorithms determining the eigensystem of large spares matrices the efficient coding of the matrix vector multiplication routine is crucial for a high performance of the algorithm. Therefore the matrix–vector multiplication routine was coded using the extract–replace method of TAO allowing for efficient use of all of the 128 registers of the FPU.

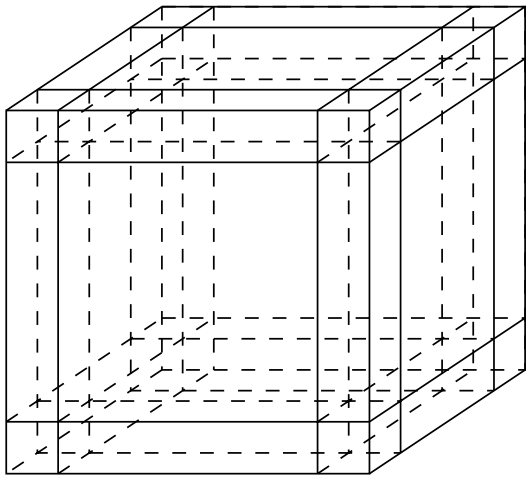

Figure 2: To avoid "IF" statements and flags in the TAO source code the total number of cells residing on one node is divided into 27 parts with definite neighboring relations. For the bulk region (fully inside the box) all neighboring cells are in the local memory of the node. For cells in other pieces it is known at programming time which part of the matrix–vector multiplication requires remote data access.

In addition care had to be taken in the multiplication routine where boundaries of the segments were involved. To simplify the coding of the multiplication loop by avoiding IF statements and flags definite neighbor relations of the cells were introduced by splitting up the whole segment into 27 pieces (see Fig.2). As a consequence the source code of the matrix–vector multiplication is somewhat blown up since every single multiplication operation has to be written 27 times for every piece of the segment.

4 PERFORMANCE OF THE ALGORITHM

The APE–100 supercomputer is mainly used for long running simulations in the field of lattice QCD and similar theories in high energy physics. To give easy access to a community familiar with MAFIA a software interface to the APE–100 has been written. After defining the geometry in the M (mesh) module of MAFIA, the E module is started and a new command "MaxqSolve" is issued which writes the system matrix to disk and submits the appropriate script to the APE–100 queue. After the solver on the APE–100 has finished the user can read in the resulting eigensystem by giving the "MaxqGetResults" command to the E module. The whole data transfer to and from the APE–100 is hidden from the user.

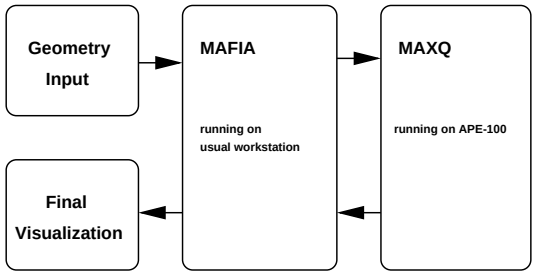

Figure 3: Schematic view of the software interface

The total number of gridpoints which can be used in the discretization is restricted due to memory limitations. Furthermore the used eigensolver needs additional memory for every eigenvector to be found. The memory needed per gridpoint is 28 octets for the system matrix and 12 octets for every eigenvector. This leads to

$$M_{total} = 12 * N * (n + 5) \text{ octets} \qquad (2)$$

for an expression for the total memory used. N is the number of gridpoints and n denotes the number of eigenvectors to be found.

On the QH2 (8x8x4) with 16 MB per node eq. (2) results in:

number of eigenvectors	available gridpoints
1	60.000.000
2	51.000.000
5	36.000.000
10	24.000.000
20	14.000.000
50	6.500.000
100	3.400.000

In principle the last table shows the applicability of the proposed approach. However, investigations considering the role of the single precision floating point arithmetic of the APE–100 are to be done. The scalar products are reported to be sensitive to the single precision / double precision problematics. Therefore the scalar products are coded using a software emulation of double precision arithmetics. The same procedure has already been applied in the original MAFIA package to save memory and keep rounding errors at a minimum.

5 CONCLUSIONS

The calculation of eigenmodes on a three–dimensional basis is crucial for the simulation of accelerating cavities. Only a fully three–dimensional treatment of Maxwell's equations can account for effects connected to fabrication errors and surface roughness and most important to effects which arise from devices such as input couplers or HOM–couplers. These coupling devices inevitably break the rotational symmetry of the cavity cannot be neglected for the determination of eigenmodes of the cavity. However, the lack of computional power on high end workstations normally avoids the inclusion of the three–dimensional effects described above. Therefore in the present paper an approach to a supercomputer solution of the eigenmode problem of superconducting cavities has been made.

Though there is no real example of the approach until now the performance analysis of the algorithm shows that determining the eigenmodes of superconducting cavities on APE–100 supercomputers is a reasonable idea to overcome the limits of high end workstations. Work for the application to parts of the TESLA structure is in progress.

6 ACKNOWLEDGEMENTS

The authors wish to thank Prof. Th. Weiland and his colleagues for many helpful discussions on MAFIA and H. Simma for helping implementing the eigensystem solver and for many hints concerning the APE–100. The support given by DESY in providing computing time for the MAXQ project (Maxwell's Equations on Quadrics) is gratefully acknowledged.

7 REFERENCES

[1] Th. Weiland, 'On the Numerical Solution of Maxwell's Equations and Applications in Accelerator Physics, Part. Acc. , **15**, 245–291 (1984)

[2] Th. Weiland, 'On the Unique Numerical Solution of Maxwellian Eigenvalue Problems in Three Dimensions', Part. Acc. ,**17**, 227–242 (1985)

[3] Th. Kalkreuther, and H. Simma, 'An accelerated conjugate gradient algorithm to compute low-lying eigenvalues – a study for the Dirac operator in SU(2) lattice QCD', DESY 95-137, HUB–IEP–95/10 (1995)