Solutions of TEAM Problem #13 Using Integral Equations in a Sequential and Parallel Computing Environment

Lauri Kettunen and Kimmo Forsman
Tampere University of Technology
Laboratory of Electricity and Magnetism
P.O.Box 692, FIN-33101 Tampere
FINLAND

David Levine and William Gropp
Argonne National Laboratory
Mathematics and Computer Science Division
Argonne, IL 60439-4801
U.S.A.

***Abstract* - In this paper a brief discussion of *h*-type volume integral formulations implemented in GFUNET/CORAL code is given and solutions of TEAM benchmark #13 are shown. GFUNET/CORAL is a general purpose code for 2D and 3D magnetostatics. Solutions of TEAM problem #13 are computed using both a sequential and parallel version of GFUNET/CORAL.**

NOTATION

$e = \{i,j\}$ An edge connecting nodes i and j in that order.

w_e A basis function associated with edge e.

$\boldsymbol{W}^1$ Space spanned by 'edge elements', i.e. Whitney elements of degree $p = 1$.

h_e Line integral of magnetic field strength $\boldsymbol{H}$ along edge e, i.e. the coefficient associated with edge e.

$\boldsymbol{H}^1$ Magnetic field strength approximated in space $\boldsymbol{W}^1$, i.e $\boldsymbol{H}^1 = \sum_e h_e \boldsymbol{w}_e$.

$\boldsymbol{M}_w$ Approximation of magnetization $\boldsymbol{M}$ computed with susceptibility and $\boldsymbol{H}^1$, $\boldsymbol{M}_w = \chi(\|\boldsymbol{H}^1\|)\boldsymbol{H}^1$.

BACKGROUND

The integral formulation in GFUNET/CORAL is based on a decomposition of $\boldsymbol{H}$ into the parts $\boldsymbol{H}_s$ and $\boldsymbol{H}_m$ due to source currents and magnetization, respectively

$$\boldsymbol{H}(\boldsymbol{r}) = \boldsymbol{H}_s(\boldsymbol{J},\boldsymbol{r}) + \boldsymbol{H}_m(\boldsymbol{M},\boldsymbol{r}) . \qquad (1)$$

Writing field $\boldsymbol{H}_m$ in terms of magnetic field strength and susceptibility one can setup a system of integral equations and solve for the coefficients describing the magnetic field $\boldsymbol{H}$ [1].

We set up the system of equations in two different ways. In the first case, we approximate $\boldsymbol{H}_m$ and $\boldsymbol{H}_s$ in $\boldsymbol{W}^1$ and solve

$$\int_V \boldsymbol{N}_e \cdot [\boldsymbol{H}^1 - \boldsymbol{H}_m^1(\chi,\boldsymbol{H}^1)]\, dv = \int_V \boldsymbol{N}_e \cdot \boldsymbol{H}_s^1\, dv . \qquad (2)$$

Functions $\boldsymbol{N}_e$ are basis functions of the edges 1,...,n belonging to an independent set. This is discussed in more detail in the next section.

In the other case, $\boldsymbol{H}_m$ due to $\boldsymbol{M}_w$ stands as it is and we solve

$$\int_V \boldsymbol{N}_e \cdot [\boldsymbol{H}^1 - \boldsymbol{H}_m(\chi,\boldsymbol{H}^1)]\, dv = \int_V \boldsymbol{N}_e \cdot \boldsymbol{H}_s\, dv . \qquad (3)$$

Even though eqs. (2) and (3) look similar, there is a significant difference in the results. The integral equation matrix can also be symmetrized, if both sides of eqs. (2) and (3) are multiplied with χ.

INDEPENDENT SET OF EQUATIONS

Combining edge elements (Whitney elements of order $p = 1$) with integral equations in order to solve magnetostatic problems leads to a question of how to define an independent set of equations. The problem is related to the fact that (in a simply-connected region) $\boldsymbol{H}$ is a gradient field and it should be approximated in a subset of $\mathbf{W}^1$, i.e. in $ker(\boldsymbol{W}^1)$ for which all the closed line integrals vanish.

An independent set of equations is found by forming a tree from the graph of all edges in the mesh. A tree connects all the nodes in the mesh (in each distinct region) without forming any loops. Hence, coefficients of the co-tree edges can be uniquely defined once a tree is set, because the sum of the coefficients must be zero around any closed loop in the mesh. (The co-tree/tree separation was introduced by Albanese and Rubinacci [2] in order to define flux across surfaces co-tree edges enclose.)

Once a tree is chosen, one can define an incidence matrix $\boldsymbol{G}$, elements of which are all -1, 0, or 1 [3]. $\boldsymbol{G}$ is a k x n rectangular matrix, where k is the number of edges in the mesh and n is the number of edges in a tree. Basis functions $\boldsymbol{N}$, associated with the tree edges, are defined with the incidence matrix

$$\boldsymbol{N}_e = \sum_{i=1}^{k} \boldsymbol{G}_{i,e} \boldsymbol{w}_i \ . \qquad (4)$$

Hence $\boldsymbol{N}_\mathrm{e}$ is a linear combination of the $\boldsymbol{w}$'s. The sum of $\boldsymbol{N}_\mathrm{e}$'s is a gradient field and $\boldsymbol{H}$ expressed in terms of independent edges is

$$\boldsymbol{H}^1 = \sum_{e=1}^{n} \boldsymbol{N}_e h_e \ . \qquad (5)$$

Currently we choose a tree rather arbitrarily. However, the choice affects the condition number of the integral equation matrix and it becomes meaningful when using iterative solvers to solve the resulting linear systems.

GENERATION OF EQUATION MATRIX

The magnetic parts of a problem are split into tetrahedra, and for practical reasons we set up a local tree for each element [4], [5]. Local trees simplify the integration of the $\boldsymbol{H}_\mathrm{m}$ field due to $\boldsymbol{M}_\mathrm{w}$. It also causes a significant decrease in the amount of work required to set up the integral equation matrix. The tree structure causes some complexity in the generation of the matrix, but it still can be done in-place without allocating extra memory.

In order to avoid recomputing some of the geometric data during nonlinear iterations we store in a scratch file either the magnetic scalar potential at the nodes, or $\boldsymbol{H}_\mathrm{m}(\boldsymbol{M}_\mathrm{w})$ at the Gaussian integration points of the tetrahedra depending on the approximation of $\boldsymbol{H}_\mathrm{m}$. The size of these files often becomes the limiting factor preventing us from running very large problems on a sequential computer [5]. (So far we have used double precision variables and have not tried to take the advantage of single precision data where possible.) On a parallel machine, or in a cluster of workstations, the size of the scratch files is not as big problem because there is a larger amount of disk space available overall.

In the parallel version the integral equation matrix is decomposed rowwise among the processors and as a result no data broadcast is required during matrix generation. The parallel version of the matrix generation routine, when run on one processor, is about 2-4% slower than the sequential version.

SOLVER

Currently, we solve a system of linear equations using LU decomposition (Crout's algorithm) and backsubstitution in the sequential version or , alternatively, Generalized Minimal Residual (GMRES) iterative solver and block diagonal preconditioning in the parallel version [6].

Nonlinear problems are iterated with simple updating of susceptibility. The GMRES solver allows us to pick the solution of the previous nonlinear cycle as an initial guess to the new one.

PARALLEL IMPLEMENTATION

GFUNET/CORAL is implemented such that the sequential and parallel versions share much of the same source code. The *Chameleon Parallel Programming Tools* [7] we use provides a low overhead interface to many vendors message-passing libraries. Chameleon also provides a uniform interface for program startup, and simplifies the use of clusters of workstations or massively parallel computers.

Currently, the parallel version of GFUNET/CORAL runs on an IBM SP-1 parallel computer, and on a cluster of SUN4/SPARC 2 or IBM RS/6000 workstations connected via Ethernet.

RESULTS

TEAM problem #13 [8] (with the new BH-curve released in the Claremont TEAM workshop) has been solved with several meshes varying the element size, but keeping the refinement ratio the same. Smaller tetrahedra were used close to the air gap and near the bend of the steel plates, Fig. 1. In practice it was found that in all cases the approach in eq. (2) gives more accurate results with a less dense mesh than the system of eq. (3).

Results for six different meshes computed with a DEC Alpha 3000-400 AXP workstation are shown in Table 2 and Figs 2. and 3.

TEAM problem #13 was also solved on 1-4 SUN4/SPARC 2, model 4c/50 workstations, and on an IBM SP-1 parallel computer using 1-8 processors in order to test the parallel version and the iterative GMRES solver (Table 1). However, we would like to emphasize that the timing results of the parallel version are preliminary. The iteration tolerance for the GMRES was set to 10^{-9}, and the SUN version was compiled without any optimization.

CONCLUSION

Using integral equations and a very coarse mesh reasonable results for TEAM problem #13 are available quickly. However, precise results require powerful computing resources.

ACKNOWLEDGMENT

We thank Hania Aazami, Sean Pratt and Diana Tabor for their help developing parallel version of GFUNET/CORAL.

TABLE 1
TIMING OF THE PARALLEL VERSION

Processors	SPARC2[a]	IBM SP1[a]	DEC Alpha[b]
1674 elements, 581 nodes, 579 equations			
1	7515 sec.	951 sec.	2180 sec.
2	3704 sec.	507 sec.	-
4	2180 sec.	279 sec.	-
6	-	235 sec.	-
7854 elements, 2280 nodes, 2278 equations			
1	-	-	90700 sec.
8	-	3416 sec.	-

[a] GMRES iterative solver
[b] LU-decomposition with backsubstitution

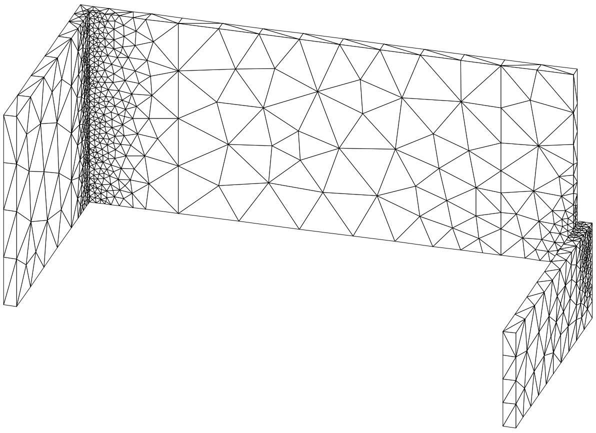

Fig. 1. Tetrahedral mesh for TEAM problem #13, case 3.

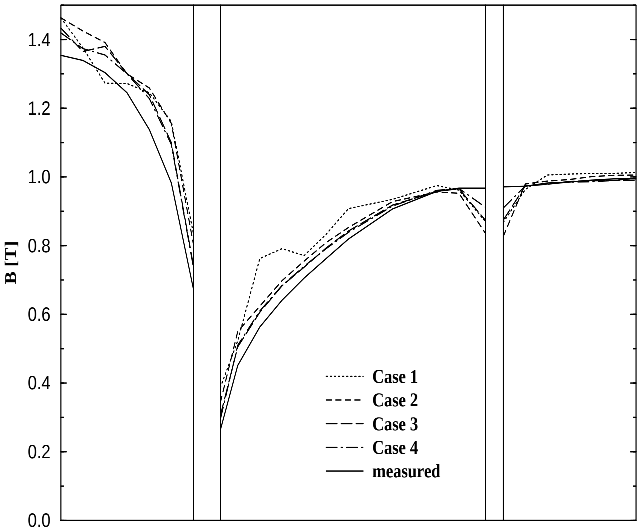

Fig. 2. Average flux density within the steel plates.

TABLE 2
RESULTS COMPUTED WITH DEC ALPHA 3000-400 AXP
USING LU-DECOMPOSITION WITH BACKSUBSTITUTION

Case	Elements	Nodes	Eqs.	CPU-time
1	393	164	162	62 sec.
2	1422	496	494	1285 sec.
3	6213	1722	1720	39197 sec.
4	10577	2742	2740	179200 sec.

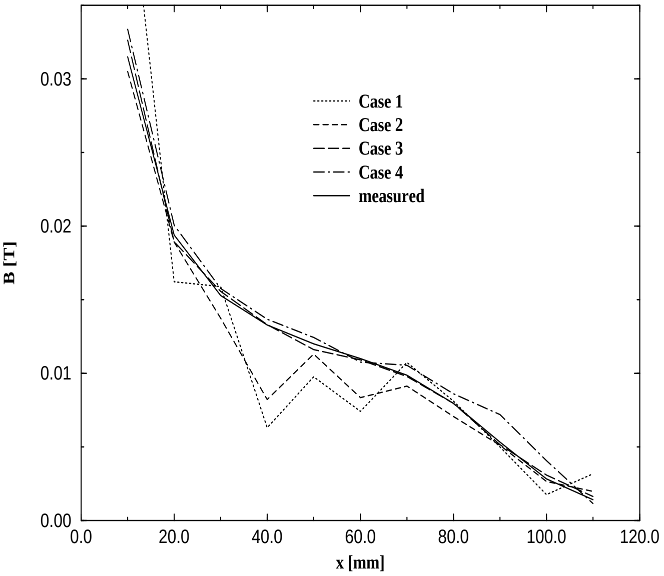

Fig. 3. Magnetic field strenght under the steel plate.

REFERENCES

[1] M.J. Newman, C.W. Trowbridge, L.R. Turner: " GFUN: An Interactive Program as an Aid to Magnet Design," *Proc. 4th Int. Conf. Magn. Tech.*, Brookhaven, New York, September 1972.

[2] R. Albanese, G. Rubinacci: "Integral Formulation for 3D Eddy-current Computation Using Edge Elements," *IEE Proc.*, Vol. 135, Pt. A, No. 7, September 1988, pp. 457-462

[3] A. Bossavit: "Simplicial Finite Elements for Scattering Problems in Electromagnetism," *Comp. Meth. in Appl. Mech. and Eng.*, 76, North-Holland, 1989, pp. 299-316.

[4] L. Kettunen, L.R. Turner: "How to define the minimum set of equations for edge elements," *Int. J. Appl. Elec. in Mat.*, 3, Elsevier, 1992, pp. 47-53

[5] L. Kettunen, L.R. Turner: "Solution of TEAM Benchmark Problem #13, 3-D nonlinear magnetostatic model," *Appl. Comp. Elec. Soc. J.*, vol. 8, no. 2, 1993, pp. 226-231

[6] W. Gropp and B. Smith: "Simplified Linear Equation Solvers Users Manual, " *Mathematics and Computer Science Division, Argonne National Laboratory*, ANL-93/8, March 1993

[7] W. Gropp and B. Smith: "Users Manual for the Chameleon Parallel Programming Tools, " *Mathematics and Computer Science Division, Argonne National Laboratory*, " ANL-93/23, June 1993

[8] T. Nakata, T. Takahashi, K. Fujiwara and P. Olszewski: " Analysis of Magnetic Fields of 3-D Non-linear Magnetostatic Model (Problem 13), " *Proc. of the European TEAM Workshop and Int. Sem. on Elec.magn. Field Anal.* Oxford, England, April 1990. *Comm. of the European Communities, J. Res. C.*, EUR 12988 EN