Solutions of TEAM Problems 13 and 20 Using a Volume Integral Formulation

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
9700 South Cass Avenue
Argonne, Illinois 60439, U.S.A.

Abstract: Solutions for TEAM benchmark problems 13 and 20, obtained with an h-type volume integral formulation, are presented. Results computed with an increasing number of unknowns are shown in order to study the convergence of the numerical calculations. Some theoretical questions and aspects of parallelism are also highlighted.

I. Introduction

Nonlinear magnetostatic problems are often solved numerically using formulations based on partial differential equations (PDEs). Although several integral formulations were invented many years ago to solve magnetostatic problems, they have not been as popular as PDEs, probably because of the dense system of equations. Perhaps also the theoretical background of volume integral formulations has not been as well established as in the case of PDEs.

Our purpose is to demonstrate that integral formulations for magnetostatics can be developed using the same approach as with PDE solvers. Adopting the standard procedure of weighted residuals and employing proper finite element spaces, one can show that integral formulations arise from the same well-known theory as PDE formulations. A solution of a well-posed integral formulation minimizes the energy stored in the magnetic field in the same manner as PDEs. In this paper the main principles are discussed. Solutions presented for the TEAM benchmark problems verify the method.

II. An h-type Integral Formulation

For simplicity let us assume that Ω and its boundary Γ are simply connected (without losing the generality of multiply connected regions). If no currents are present in Ω, the magnetic field strength $\mathbf{H}$ is a gradient field. Also, in a multiply connected region, $\mathbf{H}$ is locally a gradient field, but it does not necessarily have a unique scalar potential in Ω.

To set up a discrete problem, we use Whitney elements [2, 3, 4]. $\mathbf{H}$ is approximated in the space $grad\ W^0$. Since $grad\ W^0 = W^1 \cap \ker(curl)$, one has two options for approximating $\mathbf{H}$.

$$\mathbf{H}^1 = \sum_{n=1}^{n_{nodes}} \psi_n grad(w_n), \tag{1}$$

or

$$\mathbf{H}^1 = \sum_{e=1}^{n_{edges}} h_e \mathbf{w}_e \ , \tag{2}$$

where w_n and $\mathbf{w}_e$ are the basis functions of Whitney spaces W^0 and W^1, respectively. In the case of (2) the degrees of freedom h_e are not independent, and further examination is required.

Circulation of a gradient field around any closed path vanishes. Therefore, one of the degrees of freedom in each loop of the mesh can be determined from the others. An independent set can easily be found by forming a tree from the graph of all edges in the mesh. The corresponding basis functions are determined from the incidence relationships of the (oriented) tree and co-tree edges [8, 1]. Let us denote the coefficients associated with the tree edges with $\mathbf{h}^t$. The set of degrees of freedom associated with the graph of all edges is given by

$$\mathbf{h} = \mathbf{R}\mathbf{h}^t, \tag{3}$$

where $\mathbf{R}$ is an $n_{edges} \times n_{tree}$ incidence matrix whose elements are all either -1, 0, or 1 [10]. The basis functions associated with the tree edges are thus

$$\mathbf{v}_i = \sum_{e=1}^{n_{edges}} R_{ei} \mathbf{w}_e, \tag{4}$$

and the approximation of $\mathbf{H}$ is

$$\mathbf{H}^1 = \sum_{e=1}^{n_{tree}} h_e^t \mathbf{v}_e. \tag{5}$$

Since our goal is time-dependent problems, we approximate $\mathbf{H}$ as in (5). In practice, the choice between (1) and (5) affects the generation of the integral equation matrix and the right hand side (RHS). However, assuming exact arithmetic, they both lead to the same solution. Also, the amount of work required to generate the system of equations is about the same in the static case. In eddy-current problems, however, choice (5) seems preferable to (1), even if neither choice excludes generalization from static to time-dependent problems.

The weighted residual form is developed by multiplying equation $\nabla \cdot \mathbf{B} = 0$ with a test function ψ, integrating over Ω, and applying integral relationships analogous to Green's first identity to get

$$\int_\Omega \mu \mathbf{H} \cdot \mathbf{v}_i dv - \int_\Omega \mu \mathbf{H}^m(\chi, \mathbf{H}) \cdot \mathbf{v}_i dv = \int_\Omega \mu \mathbf{H}^s \cdot \mathbf{v}_i dv, \forall \mathbf{v}_i, \tag{6}$$

where $\mathbf{H}^m$ and $\mathbf{H}^s$ are the magnetic fields from magnetization and source currents, respectively. Moreover, it can be shown that the solution of (6) minimizes

$$W = \int_V dv \int_0^{\mathrm{B}} \mathbf{H} \cdot d\mathbf{B}, \tag{7}$$

which is the energy associated with the establishment of a current distribution in the presence of magnetic materials [10, 15].

Finally, we notice that if also $\mathbf{H}^m$ and $\mathbf{H}^s$ are approximated in $W^1 \cap \ker(curl)$, the solution of (6) can be shown to be equivalent to that of

$$h_e - h_e^m(\mathbf{h}^t) = h_e^s, \ \forall e \in T, \tag{8}$$

where T is the set of tree edges [10].

III. Implementation

The code based on the formulation described in this paper is called GFUNET. The current version runs on both sequential and parallel machines. The system of equations is solved with LU factorization or with GMRES, and a nonlinear problem is iterated using the standard Newton-Raphson method.

The most time-consuming parts are the matrix generation, which is a $O(n^2)$ process, and the solver, which scales $O(n^3)$ if LU factorization is used, and $mO(n^2)$ if GMRES is adopted (m is the number of iterations). In practice, we have found that the GMRES either with block diagonal (BDD) or with incomplete LU decomposition (ILU) preconditioner is significantly faster than direct LU factorization with back substitution. For instance, in solving a system of 3,692 equations, LU factorization requires about 400 CPU-seconds whereas GMRES takes only about 65 CPU-seconds to reach the tolerence of 10^{-8} on a DEC Alpha 3000-600 AXP workstation. However, since we are still testing various preconditioners and other iterative solvers (e.g., BiCG and Bi-CGSTAB) especially in a parallel computing environment, all timing results presented in this paper are those obtained with LU factorization. A paper about iterative solvers will be published later on.

TABLE I
Results of TEAM Problem 13

Case	1	2	3	4
Nodes	184	803	3694	12316
Elements	487	2503	14625	55107
Equations	182	801	3692	12314
Energy (J)	0.03288	0.02984	0.02909	0.02867
CPU-time (s)	16.0	207	8809	25105

Recent developments of GFUNET [7] have made the code more than an order of magnitude faster than before [11, 9, 5], and therefore results of this paper update all previous TEAM problem results we have presented. No attempts have been made to minimize the solution times by adjusting the parameters of the code. All settings were those we have found reasonable in general.

IV. Team Problem 13

TEAM benchmark problem 13 [12] was solved using several meshes with an increasing number of nodes and tetrahedra, in order to study the convergence of the results. Data of four different meshes are shown in Table I. In all cases, elements were distributed in the same manner: that is, the size of the tetrahedra varies, but the ratio between the number of elements in different regions remains about the same [6]. The mesh of case 3, consisting of 14,625 tetrahedra and 3,694 nodes is shown in Fig. 1. GFUNET can also take full advantage of the symmetry, and a solution computed with a coarse mesh can be inserted as an initial guess to a new solution.

The first three cases reported were solved on a DEC Alpha 3000-600 AXP workstation, and the fourth case on an IBM SP1 parallel computer with 58 RS/6000 model 370 processors. In addition to the total CPU-time, the energy stored in the magnetic field inside the magnetic materials is printed in Table I. In the parallel case the CPU-time presented is the maximum CPU-time taken by one processor. Results clearly show the energy minimization property of the formulation. The average flux density in the steel plates is plotted in Fig. 2, and magnetic flux density along the given line in Fig. 3. The computed results seem to converge close to, but in some points slightly above, the measured values.

V. Team problem 20

The main purpose of TEAM problem 20 [13] is to com-

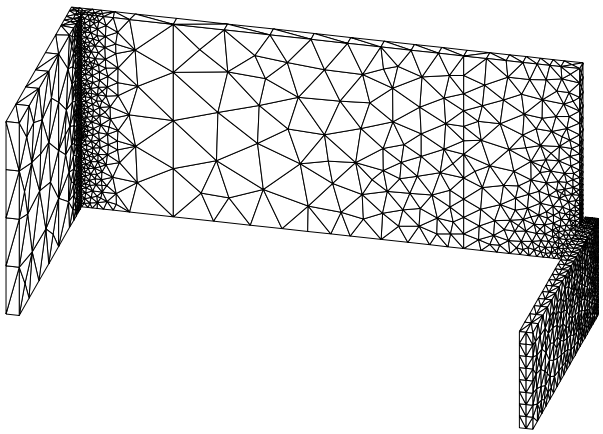


Fig. 1. Mesh of TEAM problem 13: case 3: 3694 nodes and 14,625 elements

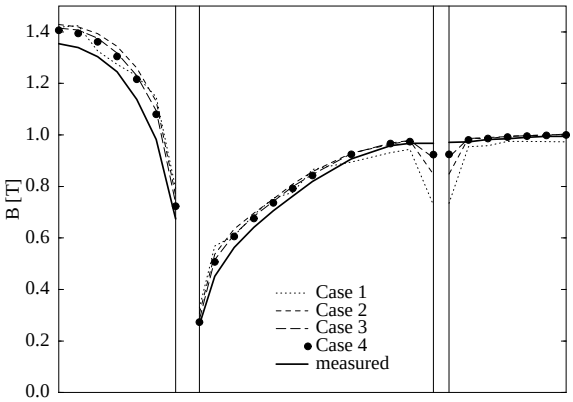


Fig. 2. TEAM problem 13: average flux density in the steel plates

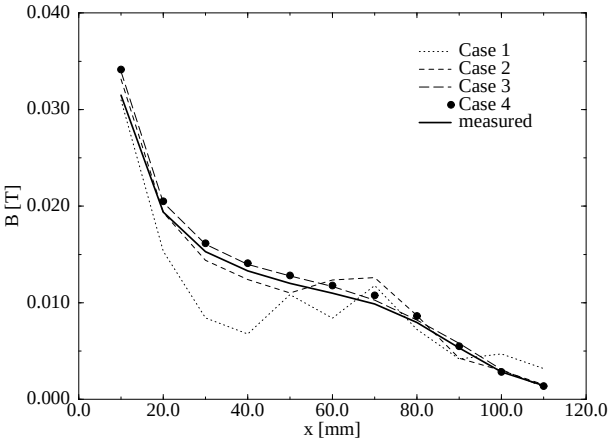


Fig. 3. TEAM problem 13: magnetic flux density in air

pare computed forces with measured ones [14]. While neither Maxwell's stress tensor nor the method of virtual work is attractive with integral equations, forces are computed with equivalent magnetization currents and the Lorentz force. Equivalent magnetization currents, defined by $\mathbf{J}_m = \nabla \times \mathbf{M}$ and $\mathbf{K}_m = \mathbf{M} \times \mathbf{n}$, produce the same magnetic field as $\mathbf{M}$ itself, and therefore they can be used in force calculations. Moreover, in the discrete case, $\mathbf{J}_m = 0$ everywhere in space as a result of the approximation of $\mathbf{M}$. Thus the force $\mathbf{F}$ acting on the bar of TEAM problem 20 can be expressed as

$$\mathbf{F} = \int_S \mathbf{K}_m \times \mathbf{B} da, \tag{9}$$

where S contains all facets of the tetrahedra in the bar. The Lorentz force approach has also another advantage: it is a straightforward method and does not require special expertise.

In Table II the data for three different meshes is shown. In addition, timings of the most important routines of GFUNET are presented. The mesh of case 3 is shown in Fig. 4. In the other three cases, elements were distributed in the same manner. Results were computed on a DEC Alpha 3000-600 AXP workstation. In solving large problems, the most time-consuming routines are the matrix generation and the solver. As Table II shows, the matrix generation scales as $O(n_{nodes} \times n_{elements})$. However, most of the time needed in matrix generation is spent in integration of geometry-dependent terms. These terms are computed only once and then are stored into a file. The same data is used while updating the integral equation matrix and in Jacobian matrix generation.

In a parallel computing environment the integral equation matrix is generated without any data broadcast; therefore, the time needed in matrix generation decreases linearly with an increasing number of processors. Efficiency of the parallel LU solver was found to be about 75–80% on an IBM SP1. However, the amount of data broadcast becomes greater with an increasing number of processors, and at some point the global efficiency is overwhelmed by data transfer. In practice, the optimal number of processors depends on the size of the problem and on the properties of the connecting network.

In Table III forces and flux densities at the points given in the problem definition are listed. Results solved with a highly refined mesh consisting of 50,263 tetrahedra and 10,327 nodes are also shown (case 4). This case was solved on the IBM SP1 using 47–64 processors. The **B** fields computed along the given lines are shown in Fig. 5 and Fig. 6. Also in this problem the computed results converge close to the measured values. However, the **B** field in the case of 1000 AT and the forces seem to converge above the measured data.

TABLE II
TIMING OF THE MAIN PARTS OF GFUNET
(TEAM Problem 20 with three meshes)

Case	1	2	3
Number of nodes	155	1013	3986
Number of elements	366	3815	17953
Number of equations	153	1011	3984
Tree generation (s)	0.01	0.3	3.7
Path generation (s)	0.001	0.008	0.03
RHS generation (s)	8.1	52.8	220
Matrix generation (s)	3.2	190	3500
Matrix update (s)	0.06	8.5	207
Jacobian generation (s)	0.1	8.0	322
LU-solver (s)	0.05	13.9	438
Number of iterations	16	12	11
Total CPU-time (s)	13.6	491	11367

TABLE III
RESULTS OF TEAM PROBLEM 20

Case	1	2	3	4	Measured
Nodes	155	1013	3986	10327	
Elements	366	3815	17953	50263	
$B_z(P_1)$ (T)					
1000 A	0.311	0.328	0.323	0.322	0.36
3000 A	0.755	0.835	0.840	0.835	0.84
4500 A	0.878	0.984	0.996	0.989	0.99
5000 A	0.910	1.021	1.035	1.028	1.03
$B_z(P_2)$ (T)					
1000 A	0.258	0.272	0.256	0.247	0.24
3000 A	0.563	0.581	0.571	0.575	0.63
4500 A	0.621	0.648	0.648	0.654	0.72
5000 A	0.636	0.664	0.666	0.672	0.74
$(B_z)_{ave}(\alpha - \beta)$ (T)					
1000 A	0.783	0.746	0.734	0.727	0.72
3000 A	1.784	1.784	1.787	1.788	1.75
4500 A	1.984	1.994	1.998	2.000	2.01
5000 A	2.028	2.039	2.042	2.044	2.05
$(B_z)_{ave}(\gamma - \delta)$ (T)					
1000 A	0.153	0.142	0.128	0.137	0.13
3000 A	0.379	0.366	0.362	0.362	0.36
4500 A	0.447	0.444	0.440	0.440	0.43
5000 A	0.466	0.465	0.461	0.462	0.46
F_z (N)					
1000 A	9.31	8.63	8.53	8.50	8.1
3000 A	62.29	57.29	57.00	56.57	54.4
4500 A	82.18	77.73	77.19	76.39	75.0
5000 A	87.19	82.96	82.40	81.54	80.1
CPU-time (s)					
1000 A	13.7	490	11367	19970 (47 proc.)	
3000 A	13.3	278	8005	10207 (64 proc.)	
4500 A	12.4	251	7462	12443 (48 proc.)	
5000 A	12.3	262	7378	12180 (48 proc.)	

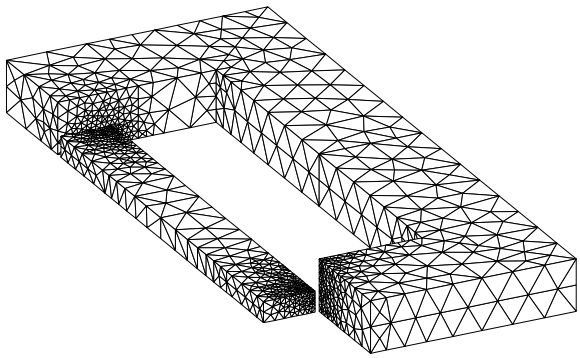


Fig. 4. Mesh of TEAM problem 20: case 3, 3986 nodes and 17,953 elements

VI. Conclusion

The results computed with GFUNET for the TEAM benchmarks seem to converge close to the measured values, but not in all cases. At this point, we cannot say whether this is due to the difficulty in measuring the BH-curve and field values or due to numerical reasons. However, according to tests we have carried out for several other problems, GFUNET has in all cases converged toward the same solution as TOSCA [16]. It would be useful to see the convergence of other methods for the TEAM problems in order to get a better understanding of the numerical methods.

Acknowledgment

We gratefully acknowledge use of the Argonne High-Performance Computing Research Facility. This facility is funded principally by the U.S. Department of Energy Office of Scientific Computing. The work of the third and fourth authors was supported by the Office of Scientific Computing, U.S. Department of Energy, under Contract W-31-109-Eng-38.

References

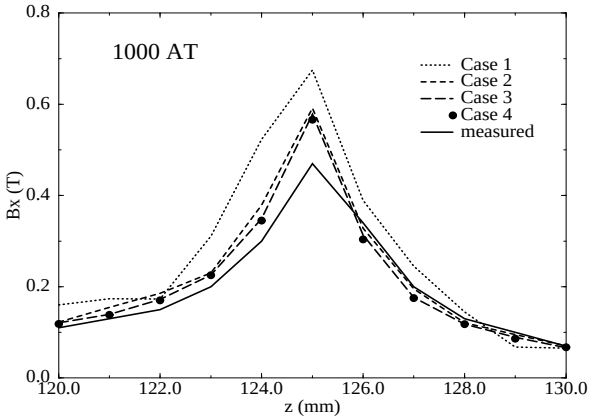
[1] R. Albanese and G. Rubinacci. Integral formulation for 3-D eddy-current computation using edge elements. In *IEE Proceedings*, volume 135, Pt A, No. 7, pages 457–462, 1988.

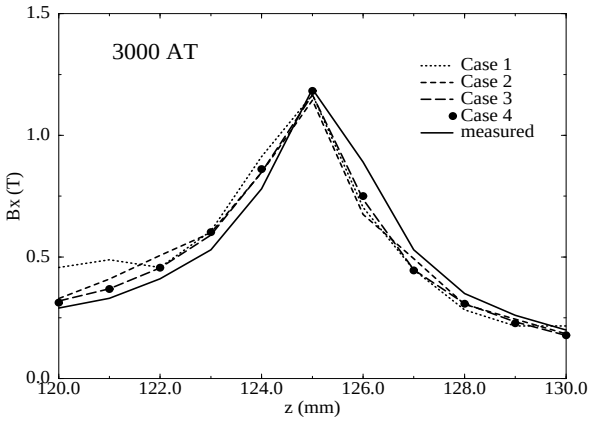
[2] A. Bossavit. Magnetostatic problems in multiply connected regions: Some properties of the curl operator. In *IEE Proceedings*, volume 135, Pt A, pages 179–187, 1988.

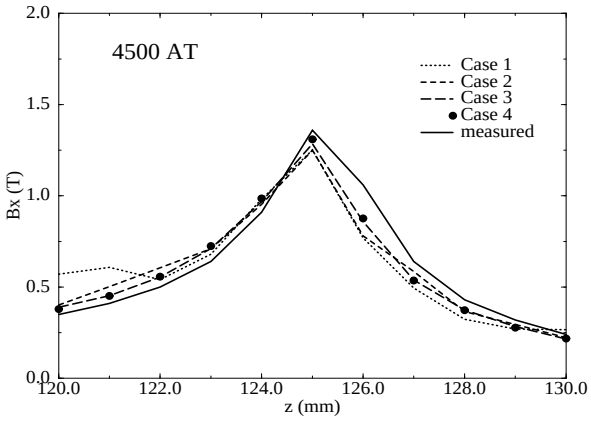
[3] A. Bossavit. Whitney forms: A class of finite elements for three-dimensional computations in electromagnetism. In *IEE Proceedings*, volume 135, Pt A, pages 493–499, 1988.

[4] A. Bossavit. Simplicial finite elements for scattering problems in electromagnetism. *Comp. Meth. in Appl. Mech. and Eng.*, 76:299–316, 1989.

[5] K. Forsman and L. Kettunen. Solutions of TEAM problem #20 using integral equations. In *Proceedings of Miami Int. ACES/TEAM Workshop*, Miami, Florida, 1993.







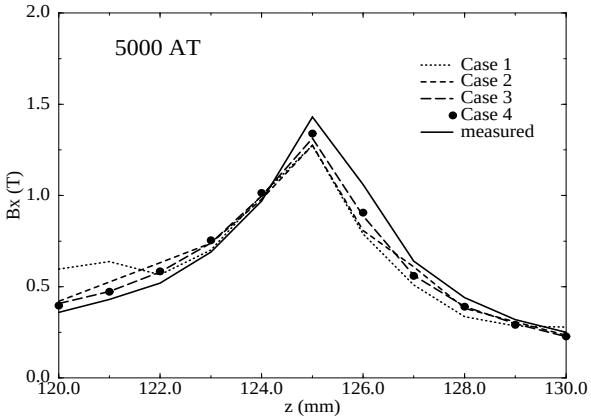
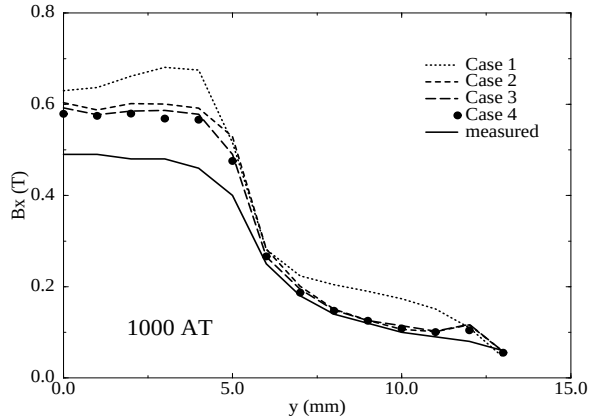
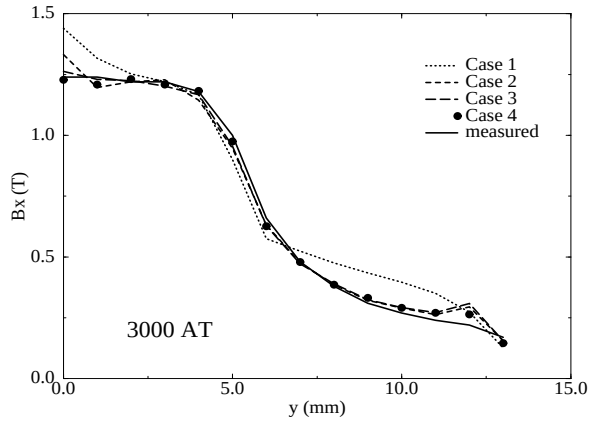
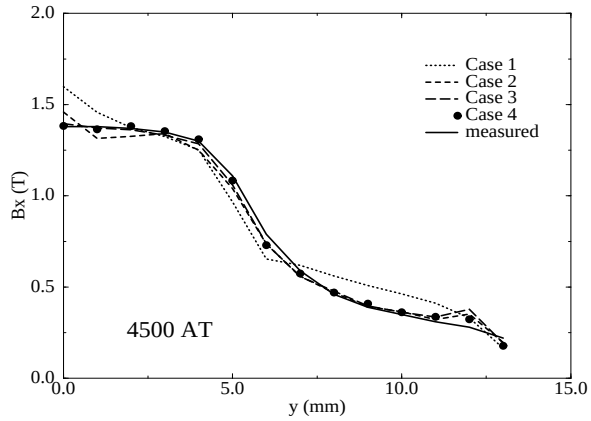


Fig. 5. TEAM problem 20. B_x along line $a-b$ at different current densities.







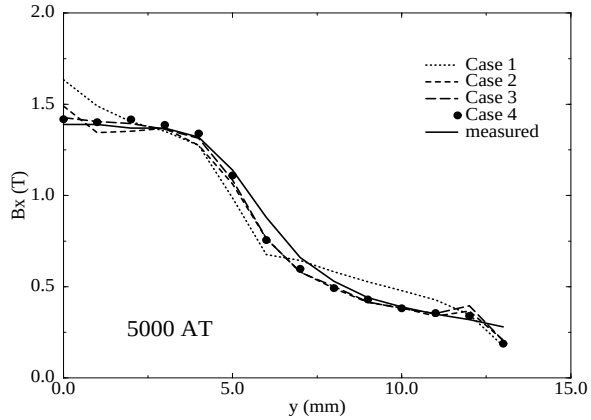


Fig. 6. TEAM problem 20. B_x along line $c-d$ at different current densities.

[6] K. Forsman and L. Kettunen. Tetrahedral mesh generation in convex primitives by maximizing solid angles. *IEEE Transaction on Magnetism*, 1994.

[7] K. Forsman, L. Kettunen, and J.-P. Nuutinen. Performance of a nonlinear 2d and 3d magnetostatic volume integral formulation. In *CEFC 94 Conference*, Aix-les-Bains, France, 1994.

[8] L. Kettunen. *Volume Integral Formulations for Three Dimensional Electromagnetic Field Computation.* PhD thesis, Tampere University of Technology, 1992. Publication 86, Tampere, Finland.

[9] L. Kettunen, K. Forsman, D. Levine, and W. Gropp. Solutions of team problem #13 using integral equations in a sequential and parallel computing environment. In *Proceedings of Miami Int. ACES/TEAM Workshop*, Miami, Florida, 1993.

[10] L. Kettunen, K. Forsman, D. Levine, and W. Gropp. Volume integral equations in nonlinear 3d magnetostatics. Technical Report MCS-P460-0894, Argonne National Laboratory, 1994.

[11] L. Kettunen and L. R. Turner. A Volume integral formulation for nonlinear magnetostatics and eddy currents using edge elements. *IEEE Transactions on Magnetics*, 28(2):1639–1642, 1992.

[12] T. Nakata and K. Fujiwara. Summary of results for benchmark problem 13 (3-d nonlinear magnetostatic model). In *Proceedings of Third International TEAM Workshop*, pages 223–249, Sorrento, Italy, 1991.

[13] T. Nakata, N. Takahashi, and H. Morishige. Analysis of 3-d static force problem. In *Proceedings of TEAM Workshop on Computation of Applied Electromagnetics in Materials*, Sapporo, Japan, 1993.

[14] T. Nakata, N. Takahashi, M. Nakano, H. Morishige, K. Matsubara, J. Coulomb, and J. Sabonnadiere. Improvement of measurement of 3-d static force problem (problem 20). In *Proceedings of of Miami Int. ACES/TEAM Workshop*, Miami, Florida, 1993.

[15] S. J. Polak, A. Wachters, and A. de Beer. An account for the use of the FEM for magnetostatic problems. In *Proceedings of COMPUMAG Conference on the Computation of Electromagnetic Fields*, Oxford, England, 1976.

[16] Vector Fields Ltd., Oxford, U.K. *TOSCA Reference Manual.*