

Efficient computation of source magnetic scalar potential

W. Hafla, A. Buchau, and W. M. Rucker

Institute for Theory of Electrical Engineering, Stuttgart, Germany

Abstract. Magnetic field problems are often excited by a known source field which itself is caused by free currents. Some formulations to solve the problem require knowledge of the source magnetic scalar potential whose gradient is the source field. In general, it has to be computed as a line integral. An approach for efficient computation of this potential has been developed that is based on an algorithm that gives short integration paths as well as on application of the fast multipole method. Numerical results indicate the efficiency of this approach especially when the number of current-carrying volume elements or the number of observation points are high.

1 Introduction

Magnetic field problems often consist of a known exciting source field due to free currents \mathbf{J} and an unknown induced field. Many formulations to solve the problem have the magnetic scalar potential, its normal derivative or its sources as unknowns. Some of them require only the source field \mathbf{H}_J as known quantity which can be obtained by evaluation of Biot-Savart law,

$$\mathbf{H}_J(\mathbf{r}) = \frac{1}{4\pi} \int_V \mathbf{J}(\mathbf{r}') \times \nabla' G(\mathbf{r}, \mathbf{r}') dV'$$

where $\nabla' G$ is the gradient of Green's function of free space,

$$\nabla' G(\mathbf{r}, \mathbf{r}') = \frac{\mathbf{r} - \mathbf{r}'}{|\mathbf{r} - \mathbf{r}'|^3}. \quad (1)$$

Other formulations require knowledge of the magnetic scalar potential ψ_J whose gradient is the source field \mathbf{H}_J . In gen-

Correspondence to: W. Hafla
(wolfgang.hafla@ite.uni-stuttgart.de)

eral, it can be computed as a line integral, If ψ_J has to be computed at the nodes \mathbf{r}_i of a mesh line integrals

$$\psi_J(\mathbf{r}_i) = \int_{C_i} \mathbf{H}_J(\mathbf{r}') \cdot d\mathbf{r}' \quad (2)$$

have to be carried out where all integration paths C_i begin at the same arbitrary start node \mathbf{r}_S of a given reference potential ψ_S and reach to every mesh node \mathbf{r}_i . Direct evaluation of Eq. (2) suffers from two drawbacks. First, errors accumulate during the numerical integration. Integration paths should therefore be as short as possible. We therefore constructed an algorithm that spans a tree over the mesh nodes and tries to keep integration path lengths short. Second, computation time can become large when the line integral is evaluated. We therefore reduced computation time by applying the fast multipole method (FMM) to evaluate Biot-Savart law.

2 Algorithm for integration path finding and for integration

There are some requirements for good integration paths. Of course, already computed potentials should be reused. Also, the sum over all integration path lengths as well as the maximum length should be as short as possible, i.e. the tree should be balanced. The basic idea we pursued was to regard the considered mesh on whose nodes the potential has to be computed as a graph as described in Gross and Yellen (2004). It is an undirected edge-weighted graph whose nodes equal a set V of vertices. The problem of finding paths C_i is equivalent to finding a spanning tree T whose vertices are V . The weights of the edges of T correspond to the length of the edges.

In order to build T , adjacency relationships between the vertices have to be determined first. Thus, three major steps have to be performed to compute the potential at the nodes:

1. Determine adjacency relationships between the nodes.

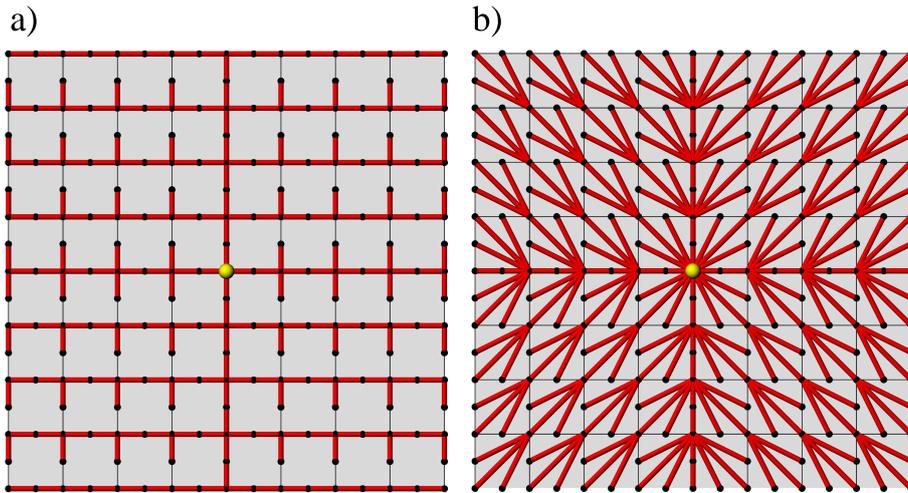


Fig. 1. Quadrature meshed with 64 eight-noded quadrilaterals. The integration tree (red) with its root node at the centre (yellow) connects all nodes of the mesh (black).

buildTree()

| |
|--|
| Define a mesh node as root vertex and set its potential ψ_0 |
| <i>current level</i> = 0 |
| Do loop over all vertices of <i>current level</i> |
| Do loop over all neighbour vertices of current vertex |
| Add all neighbour vertices that do not exist yet within tree |
| Until (no vertices have been created in <i>current level</i>) |

Fig. 2. Algorithm for tree building.

2. Generate a spanning tree whose root is the start node r_s .
3. Find and integrate C_i along the paths by recursively going from all leaves to the root of the tree.

Step 1 can be processed by use of the elements of the mesh. Dependant which nodes within an element are regarded as adjacent, different realizations of T are obtained. For example, in Fig. 1a only direct neighbour nodes along element edges were defined as adjacent whereas in Fig. 1b all nodes within each element that shares the node were considered so. The adjacency relationships between the N nodes can be stored in a sparse matrix, the so-called adjacency list.

For the construction of an algorithm that generates the spanning tree of step 2 we found it to be crucial to grow the tree level-wise. With this approach realizations of T are mostly good balanced which again can be seen from Fig. 1. For that we solely use the information stored in the adjacency list, i.e. node coordinates are ignored. Starting from the tree that contains at its level 0 only the start node at r_s , its neighbour vertices are determined with the adjacency list. They are added to T 's next level 1. Then all neighbour vertices of

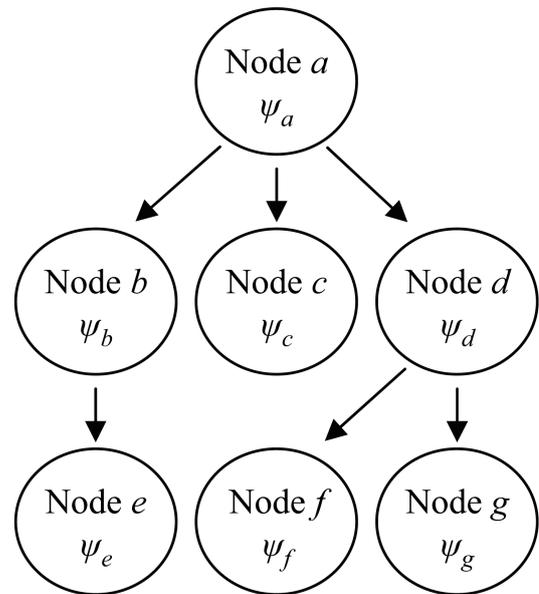


Fig. 3. Example of an integration tree.

level 1 are added at level 2 if they are not part of the tree yet and so on. The complete algorithm is shown in Fig. 2.

All line integrals along C_i in step 3 run from the root of T to its leaves. An efficient way to reuse already computed parts of the integral is to determine paths C_i by recursively starting at the leaves of the tree and moving upwards. For example, consider T displayed in Fig. 3. If the potentials ψ_a , ψ_d and ψ_f at the nodes a , d and f have been computed,

$$\psi_g = \psi_d + \int_{r_d}^{r_g} \mathbf{H}_J(\mathbf{r}'_i) \cdot d\mathbf{r}'_i, \quad (3)$$

i.e. to compute ψ_g only integration between the ancestor of vertex g and vertex d is necessary. If integration is started

at the leafs of T , ancestor vertices can be successively called until a vertex is reached whose potential has already been computed. From there on, integration between all vertices to the regarded leafs are computed and stored. In Fig. 4 the respective diagram that realizes the algorithm is displayed.

As described by Binns, Lawrenson and Trowbridge (1992) it has to be ensured that C_i does not intersect current-carrying regions. Since the exciting field is not irrotational it is also necessary to prevent the potential from being multi-valued. This can occur if the region where the potential is computed is multiply connected as with transformer problems. It therefore has been proposed by Simkin and Trowbridge (1979) to introduce suitable cuts to make the region simply connected. As the adjacency list is built with the aid of mesh elements, cuts can be defined easily by detaching elements that adjoin the cut. This means that nodes that lie on the cut have to be duplicated and assigned to elements of one side of the cut. That way the integration tree does not span over the cut which allows the potential to be discontinuous,

3 Fast multipole method

Volume integrations are generally time-consuming, therefore computation of Eq. (1) is often slow. This may also hold for determination of \mathbf{J} if it is not interpolated but computed at every integration point.

Greengard and Rokhlin (1987) already stated that the FMM can in principle be used for efficient computation of Eq. (1) but as far as it is known to the authors it has never been realized for three dimensional problems. We therefore developed an approach that allows application of the FMM to Eq. (1). It can be easily integrated in an existing FMM implementation.

Whilst in the following all derivations are valid for free currents in general, only magnetostatic problems are considered where free currents occur only in exciting coils. For simplicity volume elements that carry free currents are therefore denoted as coil elements.

In the FMM algorithm, field computation is split into a near-field part $\{\mathbf{H}_{\text{near}}\}$ due to observation points that are close to coil elements and a far-field part $\{\mathbf{H}_{\text{far}}\}$ for the remaining elements,

$$\{\mathbf{H}_J\} = \{\mathbf{H}_{\text{near}}\} + \{\mathbf{H}_{\text{far}}\} . \quad (4)$$

The division of element interactions into a near-field and a far-field part is carried out by means of an hierarchical grouping scheme that is based on cubes, the so-called octree. The near-field part has to be computed directly using Eq. (1). Coil elements with large distances from the considered observation point are considered group-wise in $\{\mathbf{H}_{\text{far}}\}$, where the size of the group depends on the distance. Therefore field sources are approximated by a truncated series expansion into spherical harmonics Y_n^{-m} . The coefficients of the series expansion are called multipole coefficients M_n^m which

Vertex::computePotential()

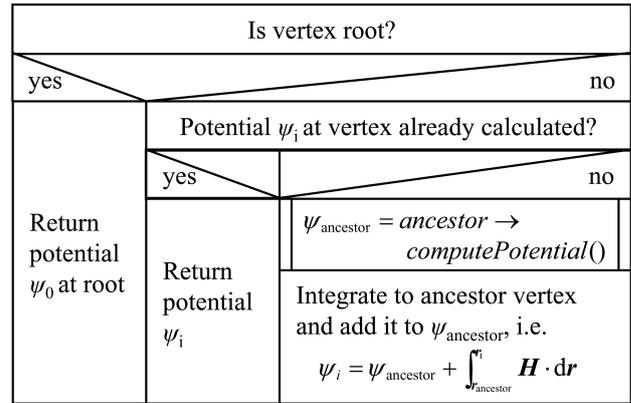


Fig. 4. Algorithm for line integration along edges of the tree.

have been derived by Greengard and Rokhlin (1987) for single layer integral operators,

$$M_n^m = \int_A \sigma(\mathbf{r}') r'^m Y_n^{-m}(\theta', \varphi') dA' . \quad (5)$$

It was shown by Buchau et al. (2004) that multipole expansions are also possible for the double layer integral operator whose kernel is the gradient of Green's Function. Contrary to these operators the integrand of Eq. (1) has vector character. Also the differential operator cannot be put in front of the integral as in the case of field calculation with the FMM. However, the usual substitution of Green's function of its multipole expansion is still possible. Starting from this approach insertion of

$$G(\mathbf{r}, \mathbf{r}') = \sum_{n=0}^L \sum_{m=-n}^n \frac{r'^m}{r^{n+1}} Y_n^m(\theta, \varphi) Y_n^{-m}(\theta', \varphi') \quad (6)$$

into Eq. (1) gives

$$\mathbf{H}_J(\mathbf{r}) = \frac{1}{4\pi} \sum_{n=0}^L \sum_{m=-n}^n M_n^m \frac{1}{r^{n+1}} Y_n^m(\theta, \varphi) \quad (7)$$

with the coefficients

$$M_n^m = \int_V \mathbf{J}(\mathbf{r}') \times \nabla_{\mathbf{r}'} [r'^m Y_n^{-m}(\theta', \varphi')] dV' . \quad (8)$$

While Eq. (6) is exact if the order of the series expansion L is infinite, for numerical calculations it has to be chosen $L < \infty$, e.g. we chose $L = 9$. With the FMM algorithm the multipole coefficients of each cube of the octree are transformed to the so-called local coefficients L_n^m from which the magnetic field can be computed,

$$\mathbf{H}_{\text{far}}(\mathbf{r}) = \frac{1}{4\pi} \sum_{n=0}^{\infty} \sum_{m=-n}^n L_n^m r^n Y_n^m(\theta, \varphi) . \quad (9)$$

This approach allows a standard FMM implementation to be used when each component of Eq. (9) is treated as a scalar

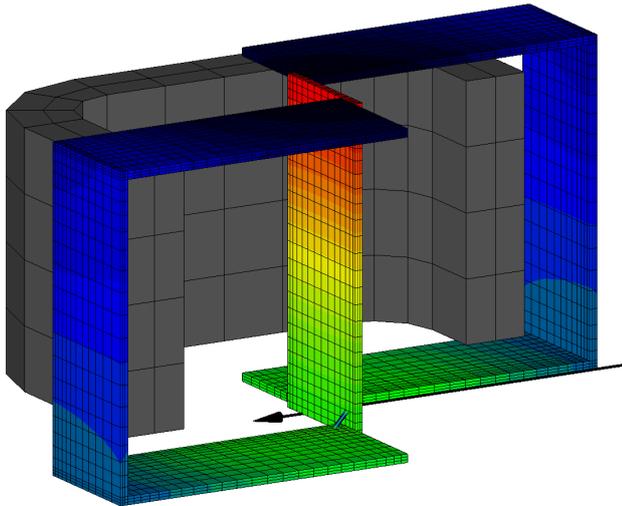


Fig. 5. Investigated mesh of TEAM Workshop Problem 13. Some elements of the coil are omitted for better graphical representation. On the right-hand side obtained integration paths are shown from the dashed section on the left.

potential. Threefold execution of the algorithm, one for each component gives the three components of \mathbf{H}_{far} . The contribution of the near-field \mathbf{H}_{near} needs to be computed only once and can be easily performed during one of the steps to compute a far-field component.

4 Numerical results

4.1 TEAM workshop problem 13

A mesh of the model from TEAM Workshop Problem 13 as published by Nakata et al. (1990) has been used to investigate the integration paths obtained by the presented algorithm. In Fig. 5 a contour plot of the computed magnetic scalar potential is displayed. As there is an air gap that separates the middle plate from the outer ones, auxiliary meshes that connect the parts were introduced. To keep the potential single-valued this was only done at the bottom of the middle plate but not on the top. As a result the potential is discontinuous between the upper air gaps. An enlargement of the auxiliary mesh within the dashed box is shown on the right in Fig. 5. For generation of the adjacency list only nodes lying beneath element edges were considered adjacent as in Fig. 1a. An approach as in Fig. 1b gives approximately the same accuracy but takes more computation time. Statistics of the lengths of integration paths are shown in the histogram plot in Fig. 6. The theoretical maximum integration length is the direct line within the mesh that connects the start node with the node of farthest distance. It is drawn as red line within Fig. 6. This theoretical maximal length is in practice always exceeded as integration paths have to run between mesh nodes.

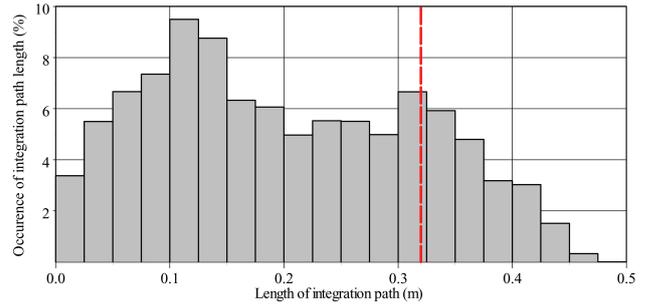


Fig. 6. Histogram plot of occurring integration path lengths.

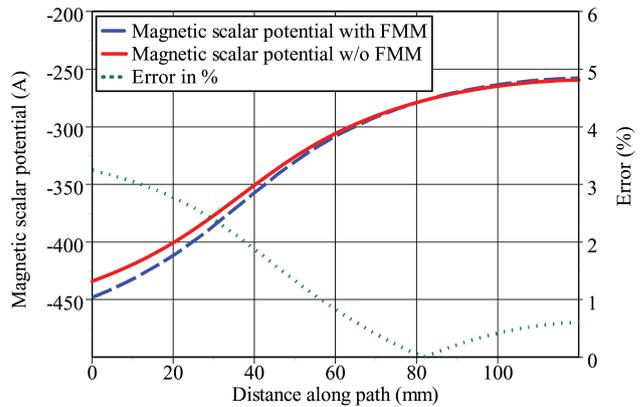


Fig. 7. Comparison of magnetic scalar potential obtained by direct evaluation of Biot-Savart law and by application of FMM. The evaluation path is shown as arrow in Fig. 5.

To compare the accuracy between direct evaluation of Biot-Savart law and with application of the FMM the potential has been computed along the arrow shown in Fig. 5. Results are shown in Fig. 7. The order of the series expansions was chosen to $L = 9$ and second order hexahedrons were used as coil elements. As results obtained by direct evaluation can be considered accurate, the error with FMM is about 3.5%. We believe this is because different to direct evaluation, we interpolated the magnetic field between mesh nodes when applying the FMM.

4.2 Toroidal coils

In Fig. 8 different meshes of toroidal coils are shown that were used to investigate the efficiency of the FMM with respect to computation time and memory requirements. The source magnetic scalar potential has been computed on the toroidal cores of the coils. In Fig. 8 mesh A, the computed potential and on the other meshes the magnetic field magnitude is shown. As the cores are multiply connected cuts had to be introduced to make the potential discontinuous which can be seen in Fig. 8 mesh A.

In Table 1 details of the computational costs by application of the FMM as well as without its application are shown.

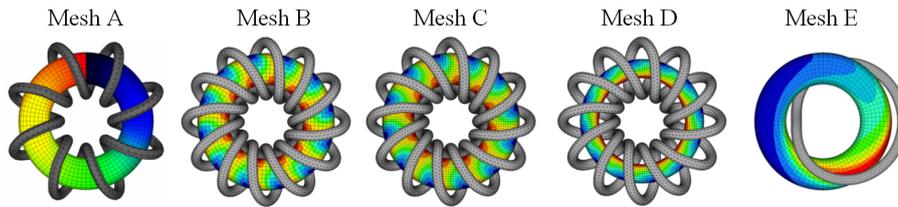


Fig. 8. Investigated meshes of toroidal coils. The contour plot on Mesh A represents scalar magnetic potential, the other meshes the magnitude of magnetic field.

Table 1. Comparison between application of FMM and classical approach regarding memory requirements and computation time for source magnetic scalar potential computation.

| Mesh | Coil Elements | Observation points | Memory with FMM/MBytes | Time with FMM | Time w/o FMM |
|------|---------------|--------------------|------------------------|---------------|--------------|
| A | 4 393 | 30 197 | 196 | 9 min 23 s | 1 h 43 min |
| B | 24 024 | 30 197 | 642 | 42 min 48 s | 9 h 25 min |
| C | 24 024 | 128 077 | 952 | 46 min 20 s | 39 h 56 min |
| D | 12 012 | 52 637 | 396 | 36 min 21 s | 8 h 12 min |
| E | 8 605 | 30 197 | 307 | 12 min 55 s | 3 h 22 min |

Observation points refer to the mesh nodes of the toroidal core. There is a large reduction of computation time especially when the number of coil elements or the number of observation points is large. The calculations were run on a 3.4 GHz Intel Pentium IV computer.

5 Conclusions

The calculation of source magnetic scalar potential is realized by performing line integrations over the magnetic field. Integration paths can be represented as edges of a spanning tree over the nodes of the mesh. Results indicate that the presented algorithm for tree generation gives balanced trees with short integration paths. Numerical errors that accumulate during line integration are therefore low. To accelerate magnetic field computation the FMM has been applied to Biot-Savart law. Computation time is thereby significantly reduced by the cost of extra memory requirements.

References

- Binns, K. J., Lawrenson, P. J., and Trowbridge, C. W.: *The Analytical and Numerical Solution of Electric and Magnetic Fields*, John Wiley and Sons, 1992.
- Buchau, A., Hafla, W., and Rucker, W. M.: Fast and efficient 3-D boundary element method for closed domains, *COMPEL*, 23, 859–865, 2004.
- Greengard, L. and Rokhlin, V.: *The rapid evaluation of potential fields in three dimensions*, Lecture Notes in Mathematics, 1360, edited by Anderson, C. and Greengard, C., Berlin, Germany: Springer-Verlag, 121–141, 1987.
- Gross, J. L. and Yellen, Y.: *Handbook of Graph Theory*, CRC Press LLC, 2004.
- Nakata, T., Takahashi, N., Fujiwara, K., Olszewski, P., Muramatsu, K., Imai, T., and Shiraki, Y.: Numerical analysis and experiments of 3-d non-linear magnetostatic model, *Proc. Int. Symp. and TEAM Workshop, Supplement A*, Okayama, Japan, 308–310, 1990.
- Simkin, J. and Trowbridge, C.: On The use of the Total Scalar Potential in the Numerical Solution of Field Problems in Electromagnetics, *Intern. J. for Num. Methods in Engineering*, 14, 423–440, 1979.