The meshfree computation of stationary electric current densities in complex shaped conductors using 3D boundary element methods

A. Buchau & W. M. Rucker
Institut für Theorie der Elektrotechnik, Universität Stuttgart, Germany

Abstract

Stationary electric current problems are based on the solution of Laplace equation for the scalar electric potential within each domain of piecewise homogeneous media. The scalar electric potential is continuous at domain interfaces along with the normal component of the electric current density, which is obtained from the normal derivative of the scalar electric potential. An indirect boundary element method is applied for the numerical solution of the three-dimensional problem. The matrix of the corresponding system of linear equations is compressed using the fast multipole method. Here, the focus is on a flexible and meshfree post-processing of the solved problem. The computation of the electric current density requires the electric conductivity of material at the position of the arbitrary chosen evaluation point. A completely automatic domain detection is necessary, if the post-processing is performed without an auxiliary volume mesh. The position of evaluation point is obtained directly from the boundary element mesh, which is used for the solution of the problem. Relevant boundary elements are filtered by an application of a flexible adaptive octree scheme, which is similar to the one of the fast multipole method. An algorithm comparable to ray-tracing is applied to detect the position of the evaluation point with respect to the set of filtered boundary elements. Field values at the evaluation point are obtained based on a reversed flexible fast multipole method scheme. In total, the presented method enables a fast, efficient, and robust post-processing in arbitrary points even in the case of complex shaped boundaries. Evaluation points can be chosen, for instance during the computation of streamlines or in planes, with an almost arbitrary spatial resolution without expensive pre-computations.
Keywords: boundary element methods, fast multipole method, Laplace equations, meshfree post-processing, visualization.


WIT Transactions on Modelling and Simulation, Vol 57, © 2014 WIT Press
www.witpress.com, ISSN 1743-355X (on-line)
doi:10.2495/BE370111

1 Introduction

Fast boundary element methods (BEM) are very attractive for the computation of stationary electric currents in piece-wise homogeneous media (Buchau *et al.* [1]). Modelling and disctretization of three-dimensional problems are significantly simplified by boundary element meshes instead of volume element meshes especially in the case of complex shaped geometrical configurations. Compression techniques like the fast multipole method (FMM) reduce the computational costs and memory requirements and therefore enable the solution of very large problems along with acceptable costs [2, 3]. Furthermore, the FMM accelerates the post-processing in previously defined points, too (Buchau *et al.* [4]).

A powerful post-processing including visualization of results is very important in practical applications of numerical field computations. An established approach is to visualize three-dimensional fields with the help of arrows, streamlines, or coloured surfaces in virtual reality or augmented reality environments [5, 6]. Typically, field values are pre-computed in the nodes of a volume mesh and the visualization tool extracts visualization objects, e.g. streamlines, based on these pre-computed values. That means in the case of boundary element methods that an auxiliary volume mesh is required and the accuracy of post-processing is limited by the resolution of this mesh Hafla *et al.* [7]. A more efficient and more flexible approach is to couple the visualization environment directly with the field computation kernel [8, 9]. Then, field values are computed only in points, which are absolutely necessary for the computation of visualization objects. Hence, the amount of data, which must be stored and transferred, is dramatically reduced. In the case of boundary element methods, this approach results in a meshfree post-processing with straight evaluations of boundary integrals.

A meshfree post-processing requires an automatic domain detection method only using the boundary element mesh of the solution of the problem (Buchau *et al.* [10]). The domain of the arbitrarily chosen evaluation point is needed both for the correct selection of relevant boundary elements in dependency of the used formulation and for the determination of material properties in the evaluation point. The number of boundary elements, which have to be considered for domain detection, is reduced by an octree scheme similar to the one of the FMM. Then, a ray-based search comparable to ray-tracing in computer graphics is applied (Goldsmith and Salmon [11]). The field values are efficiently computed using a modified reversed FMM algorithm (Buchau and Rucker [12]).

Here, an application of a meshfree post-processing for a complex three-dimensional problem is presented. Due to the geometrical configuration of the studied stationary electric current problem, the distance between appropriate evaluation points is extremely varying. The creation of an auxiliary post-processing mesh would be very laborious and the pre-computation of field values would be very expensive. Hence, only a meshfree post-processing is efficient and accurate. Special care must be taken on the correct treatment of the highly varying element size of the problem-oriented mesh both for domain detection and for evaluation of strong singular and nearly strong singular integrals [13, 14].

2 Numerical formulation

2.1 Fast boundary element method

Stationary electric current problems are based on the solution of Laplace equation for the scalar electric potential u in piece-wise homogeneous, linear, isotropic media. Here, an indirect BEM formulation using surface charge densities σ is applied

$$u(\boldsymbol{r}) = \frac{1}{\epsilon_0}\int_A \sigma(\boldsymbol{r}')G(\boldsymbol{r},\boldsymbol{r}')\mathrm{d}A'. \tag{1}$$

G is Green's function of Laplace equation

$$G(\boldsymbol{r},\boldsymbol{r}') = \frac{1}{4\pi}\frac{1}{|\boldsymbol{r}-\boldsymbol{r}'|}. \tag{2}$$

The constant ϵ_0 is the permittivity of free space and A is the surface of all conductors and dielectrics. Note, that eqn. (1) is valid in total space.

The electric potential at the ports corresponds to a Dirichlet boundary condition. The continuity of the normal component of the electric current density $\boldsymbol{J}$ at interfaces between two conductors is written as Neumann boundary condition

$$\kappa_1\frac{\partial u_1}{\partial n} = \kappa_2\frac{\partial u_2}{\partial n}. \tag{3}$$

κ_1 and κ_2 are the electric conductivities of the two conductors at the interface. The Neumann boundary condition between two non-conducting dielectrics is

$$\epsilon_1\frac{\partial u_1}{\partial n} = \epsilon_2\frac{\partial u_2}{\partial n}, \tag{4}$$

where ϵ_1 and ϵ_2 are the permittivities of the two adjacent dielectrics.

The electric field strength $\boldsymbol{E}$ is obtained from the derivative of eqn. (1)

$$\boldsymbol{E}(\boldsymbol{r}) = -\frac{1}{\epsilon_0}\int_A \sigma(\boldsymbol{r}')\,\mathrm{grad}\,G(\boldsymbol{r},\boldsymbol{r}')\,\mathrm{d}A'. \tag{5}$$

Eqn. (5) can be evaluated in total space. On the other hand, a computation of the electric current density

$$\boldsymbol{J} = \kappa\boldsymbol{E} \tag{6}$$

makes only sense for evaluation points $\boldsymbol{r}$ inside a conductor ($\kappa \neq 0$).

All surfaces are discretized using eight-noded, quadrilateral, second-order boundary elements. The resultant linear system of equations is obtained by an application of the Galerkin method. The system matrix is compressed using the fast multipole method. The linear system of equations is solved iteratively with generalized minimal residual method (GMRES) and a Jacobi preconditioner.

Eqns. (1), (5), and (6) are the basis of a post-processing, too. The evaluation point $\boldsymbol{r}_{ep}$ is arbitrarily chosen, e. g. in a plane or during the computation of a streamline.

Since the evaluation points are not known in advance, a modified reversed FMM algorithm is applied to compress eqns. (1) and (5) of the post-processing. All boundary elements are grouped by an octree scheme in the same way as the classical FMM approach. The main difference is that a very flexible octree scheme is needed. The evaluation points are sequentially added to the octree scheme during the runtime of the post-processing. Since the octree structure changes from one evaluation of the FMM to the next evaluation, all relationships between octree cubes, for instance the determination of neighbour cubes, must be

performed on demand. Therefore, the complete octree is organized and stored with the help of linked lists, which enable even significant changes of the structure during runtime.

The first step of the field computation of the post-processing is to add an evaluation point $\boldsymbol{r}_{ep}$ to an octree cube C_{ep}. The search of an appropriate cube C_{ep} is oriented at the existing octree structure defined by the boundary elements of the solution of the problem. One aim is to define C_{ep} as large as possible. Another aim is to get C_{ep} without boundary elements. Furthermore, the minimal size of C_{ep} is limited by its neighbour cubes. Then, the number of so-called near-field interactions is reduced and the series expansions of the FMM can be reused for further evaluation points in the neighbourhood.

All boundary elements, which are assigned to C_{ep} or one of its neighbour cubes, belong to the near-field of $\boldsymbol{r}_{ep}$. Note, boundary elements of other cubes, which intersect C_{ep} or one of its neighbour cubes, have to be considered, too. The near-field is obtained by an evaluation of eqn. (1) or eqn. (5), where A is the surface of all boundary elements in the near-field of $\boldsymbol{r}_{ep}$.

The far-field is taken into account by an evaluation of the local expansion of the FMM, which is a Taylor series expansion of the electric potential in C_{ep} in spherical coordinates r, θ, φ with origin in the centre of C_{ep}

$$u_f(\boldsymbol{r}_{ep}) = \frac{1}{4\pi}\sum_{n=0}^{L}\sum_{m=-n}^{n} L_n^m r^n Y_n^m(\theta,\varphi). \tag{7}$$

L_n^m are the local coefficients in C_{ep}, L is the order of series expansions, and Y_n^m are spherical harmonics.

Here, a modified reversed FMM algorithm is used to compute the local coefficients L_n^m. If the local coefficients L_n^m are needed, their computation starts at the cube C_{ep} and not at the sources as in the original FMM algorithm. The local coefficients L_n^m are obtained from the local coefficients of the parent cube of C_{ep} and the multipole coefficients M_n^m of some other cubes. These cubes are at the same octree level as the cube C_{ep}, are child cubes of the neighbour cubes of the parent of C_{ep}, and are not neighbour cubes of C_{ep}. The local coefficients of the parent cube of C_{ep} are computed in the same way. The multipole coefficients are computed from multipole coefficients of the child cubes of a cube or directly from the surface charge density of boundary elements in the cube

$$M_n^m = \frac{1}{\epsilon_0}\int_A \sigma(\boldsymbol{r}')r'^n Y_n^{-m}(\theta',\varphi')\mathrm{d}A'. \tag{8}$$

The modified reversed FMM algorithm reduces the number of computations, since coefficients of series expansions are only calculated, if they are further processed. Furthermore, unnecessary runs of the FMM algorithm are avoided. If another evaluation point is near a previous evaluation point, all or almost all coefficients of series expansions can be immediately reused without an execution of the whole FMM algorithm. Only the near-field must be evaluated again.

The electric field strength or the electric current density are obtained by an analytic differentiation of eqn. (7) using Cartesian coordinates.

Since evaluation points can be very close to boundary elements or on boundary elements, nearly singular, singular, nearly strong singular, and strong singular

integrals have to be efficiently and precisely computed using special integration techniques based on polar coordinates with origin in the singular point on the boundary element and Taylor series expansions of the integral kernel.

2.2 Automatic domain detection method

An automatic and efficient domain detection method is obligatory to determine the electric conductivity at an arbitrary evaluation point in the case of complex shaped geometrical configurations and a meshfree post-processing.

The basic idea of the domain detection method is to define a ray starting at the evaluation point $\boldsymbol{r}_{ep}$ in direction $\boldsymbol{d}_r$. The first boundary element, which is intersected by this ray is used for position detection (fig. 1). However, the direction $\boldsymbol{d}_r$ is not defined in contrast to well-known ray-tracing methods in computer graphics.

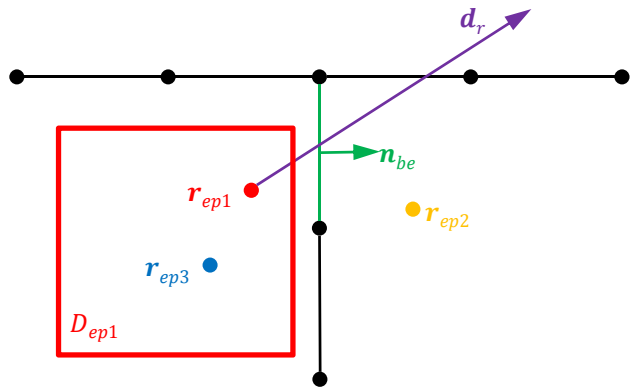

Figure 1: Basic idea of automatic domain detection method.

The closest point on the relevant boundary element to the evaluation point $\boldsymbol{r}_{be}$ is then computed using a gradient search algorithm. The domain of $\boldsymbol{r}_{ep}$ is determined by an evaluation of

$$\left(\boldsymbol{r}_{ep}-\boldsymbol{r}_{be}\right)\cdot\boldsymbol{n}_{be}\begin{cases}>0, & D_{ep}=D_{be+}\\ <0, & D_{ep}=D_{be-}\end{cases}. \tag{9}$$

$\boldsymbol{n}_{be}$ is the normal vector in $\boldsymbol{r}_{be}$. D_{be+} is the domain in direction of the normal vector and D_{be-} is the domain in opposite direction of the normal vector. If the scalar product in eqn. (9) is equal or nearly zero, another element is searched.

Another basic idea is to reuse the determined domain data for as many evaluation points as possible. In the example, which is depicted in fig. 1, the evaluation point $\boldsymbol{r}_{ep1}$ is assigned to a cuboid domain D_{ep1} without boundary elements. Hence, all points inside D_{ep1} belong to the same domain.

Here, the main challenge is to find the first boundary element in direction of the ray. The direction can be arbitrary chosen and the computational costs for domain detection should be as small as possible. Furthermore, the domain

detection method must be very precise to obtain correct field values. Wrong domain detections, which correspond to artefacts in computer graphics, results in completely wrong numerical field values. Wrong values are very problematic for instance during the computation of streamlines. Problem oriented meshes with varying size and shape of boundary elements result in high demands on evaluation of limits in intersection tests.

A common approach in ray-tracing is to filter relevant elements with the help of a tree-based scheme. Here, an octree is used, since an octree is already needed for the FMM. The FMM octree and the domain detection octree are based on the same code but adapted to their special task using modern software techniques.

The first step of position detection is to add the evaluation point $\boldsymbol{r}_{ep}$ to the domain detection octree. If boundary elements are assigned to the cube C_{ep} of $\boldsymbol{r}_{ep}$ or if boundary elements intersect the cube C_{ep}, these boundary elements are used for domain detection (fig. 2). If no boundary elements are assigned to the cube C_{ep} or intersect the cube C_{ep}, the position for the whole cube C_{ep} is determined. Then the position is reused from its parent cube or it is determined from boundary elements in the neighbour cubes of the cube C_{ep}. Finally, a relatively short list of relevant boundary elements is obtained.

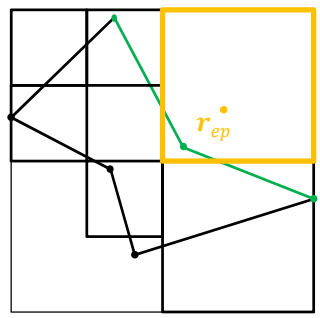

Figure 2: Filtering of relevant boundary elements for automatic domain detection.

The filtered relevant boundary elements are tested for domain detection. First, all filtered boundary elements are sorted by their distance to the evaluation point. The closest filtered boundary element is considered as candidate for position detection. A ray is defined with start point at the evaluation point $\boldsymbol{r}_{ep}$ and the direction from the evaluation point to the closest point on the candidate element $\boldsymbol{r}_{be}$

$$\boldsymbol{d}_r = \boldsymbol{r}_{be} - \boldsymbol{r}_{ep}. \tag{10}$$

The remaining boundary elements are tested for intersection of this ray

$$\boldsymbol{r}_0 = \boldsymbol{r}_{ep} + \alpha \boldsymbol{d}_r \tag{11}$$

with $0 \leq \alpha \leq 1$. The intersection of the ray with another boundary element is tested by minimizing the cost function

$$f(\xi, \eta, \alpha) = |\boldsymbol{r}_0(\alpha) - \boldsymbol{r}_{bet}(\xi, \eta)|. \tag{12}$$

$\boldsymbol{r}_{bet}(\xi, \eta)$ is a point on the boundary element, which is tested for intersection.

If $f(\xi,\eta,\alpha)$ is zero or nearly zero, the ray intersects the tested boundary element and that element is considered as new candidate for position detection. Then, the remaining boundary elements are tested for intersection with this candidate. Finally, a boundary element is found, which is used for the evaluation in eqn. (9).

A gradient search optimization method is used to find the minimum of eqn. (12). The search direction is

$$\boldsymbol{d} = -\operatorname{grad} f(\xi,\eta,\alpha). \tag{13}$$

It can be calculated analytically

$$\boldsymbol{d} = \begin{pmatrix} \frac{\boldsymbol{r}_0-\boldsymbol{r}_{bet}}{|\boldsymbol{r}_0-\boldsymbol{r}_{bet}|}\cdot\frac{\partial}{\partial\xi}\boldsymbol{r}_{bet} \\ \frac{\boldsymbol{r}_0-\boldsymbol{r}_{bet}}{|\boldsymbol{r}_0-\boldsymbol{r}_{bet}|}\cdot\frac{\partial}{\partial\eta}\boldsymbol{r}_{bet} \\ -\frac{\boldsymbol{r}_0-\boldsymbol{r}_{bet}}{|\boldsymbol{r}_0-\boldsymbol{r}_{bet}|}\cdot\boldsymbol{d}_r \end{pmatrix}. \tag{14}$$

The inequality constraints of this optimization are the limits of the local coordinates on the boundary element and the limits of the parameter α.

Along the search direction of eqn. (14) a bisection is applied for parameter β to compute the next values of the iterative gradient search method

$$\begin{pmatrix}\xi\\ \eta\\ \alpha\end{pmatrix}_{j+1} = \begin{pmatrix}\xi\\ \eta\\ \alpha\end{pmatrix}_{j} + \beta\boldsymbol{d}. \tag{15}$$

A fast and simple test for determination of the distance of the evaluation point to a boundary element is to use the centroid of the boundary elements. If only linear triangular elements are used, this approach works well. However, in the case of second order elements, the curvature of the boundary elements is not negligible. In most cases, it suffices to consider the closest node of the boundary element to the evaluation point. This is more precise than the use of the centroid but fails for very adaptive meshes and two close layers of boundary elements. The most precise and reliable approach is to determine the closest point to the given evaluation point for each filtered boundary element. This point is determined using a gradient search algorithm with inequality constraints similar to the above described search method for the computation of an intersection of a ray with a boundary element. The main difference is, that only the local coordinates on the boundary element are used as optimization variables.

3 Numerical results

A model of a printed circuit board for very high electric currents, which is used in agricultural machines, is examined as numerical example (fig. 3). Its mechanical assembly is very robust due to vibrations and high temperatures during operation. Therefore, the rectangular pins of the connectors are pressed into cylindrical holes, which are coated with copper (fig. 4). Hence, a cold welded connection between the brass pins and the copper conductor of the printed circuit board is obtained.

Here, the electric current density inside the connectors, pins, and the two conducting layers of the printed circuit board is investigated. The size of the corresponding linear system of equations is kept as small as possible by an

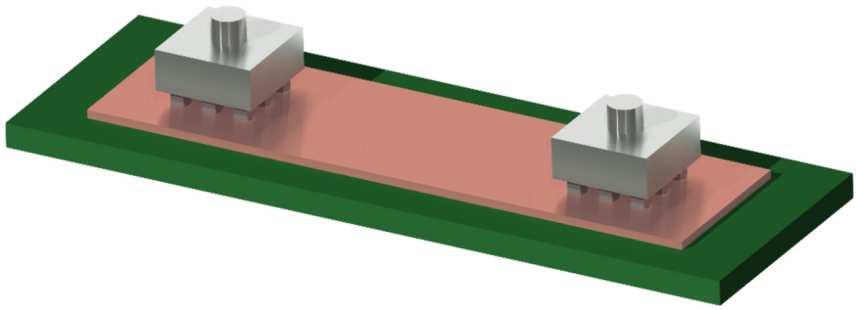

Figure 3: A model of a printed circuit board.

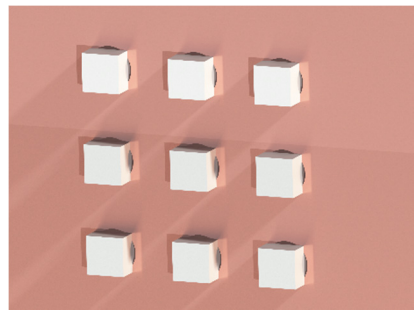

Figure 4: Detail of fig. 3.

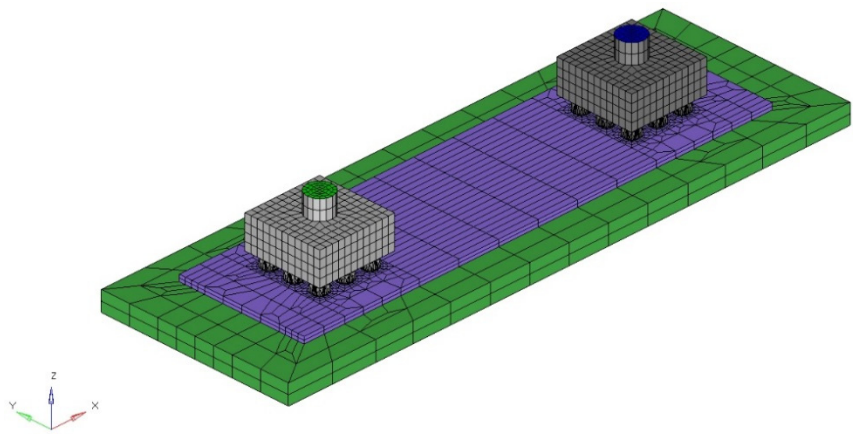

Figure 5: Adaptive boundary element mesh of the printed circuit board of fig. 3.

adaptive, problem-oriented boundary element mesh. In total, all surfaces are discretized with 14752 second order quadrilateral boundary elements (fig. 5). A mixture of thin long elements and elements of significantly different sizes is obtained especially near the pins of the connectors (fig. 6). Furthermore, the small

air-gap between the pins and the conducting cylindrical holes of the printed circuit board results in two layers of large boundary elements with a very small distance between the two layers.

The problem with in total 48000 degrees of freedom has been solved in parallel using 8 cores of two Intel Xeon E5 2643 processors running at 3.3 GHz. The clock speed of random access memory (RAM) has been 1066 MHz. GMRES converged after 112 iteration steps in 12 minutes and 42 seconds.

The post-processing has been executed on an Intel Core i7-3820QM processor at 2.7 GHz and 1600 MHz clock speed of RAM. The presented meshfree post-processing method has been implemented in C# and the software HLRS COVISE of the high performance computing centre of the University of Stuttgart has been used for visualization. The bidirectional coupling between the visualization environment and the field computation software is realized with the help of open data protocol (OData) [15]. Post-processing requests and computed field values are exchanged using extensible markup language (XML) data structures over hypertext transfer protocol (http) between the running processes.

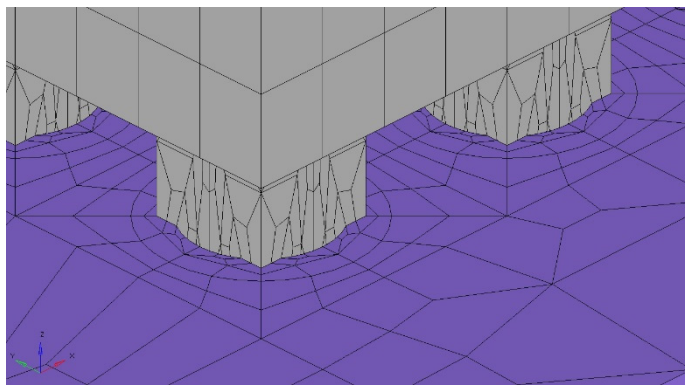

Figure 6: Detail of fig. 5.

First, the correct detection of domains in arbitrary chosen evaluation points has been tested. A slice, which cuts the connector including its pins, the printed circuit board, and the very small air gaps near the pins, has been defined. A second plane with a smaller distance between the evaluation points is defined in the neighbourhood of a pin. The domains are detected with the help of the presented automatic domain detection method. Each domain is depicted by a different colour in fig. 7 and in fig. 8. Both figures show clearly that the domain is correctly detected even very close to large distorted boundary elements and even near close layers of boundary elements. The CPU time for 50451 and 15276 domain detections in the two slices was 1 minute and 47 seconds and 2 minutes and 16 seconds respectively.

The electric current density has been computed in 35391 and in 656 evaluation points. The CPU time was 5 minutes and 49 seconds and 28 seconds. Note, due to

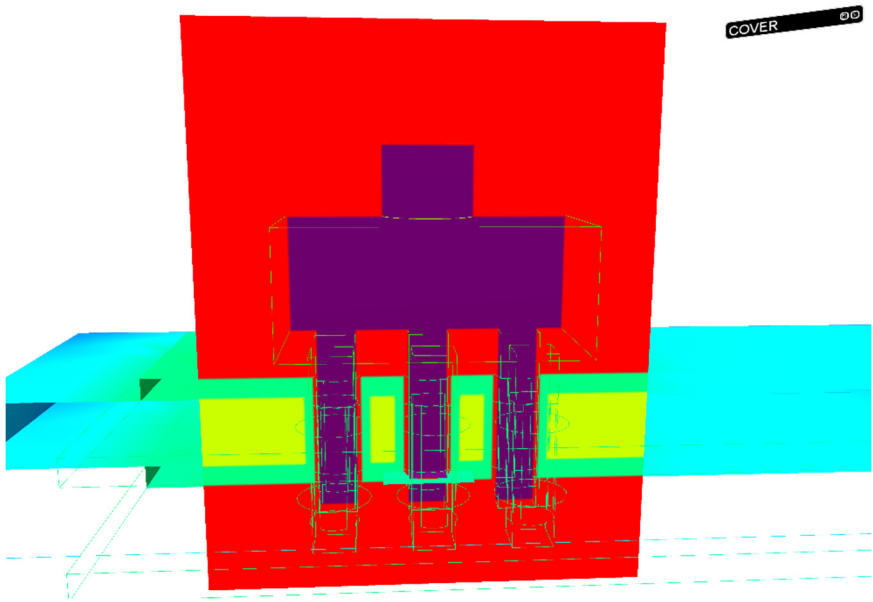

Figure 7: Detected domain number in a slice.

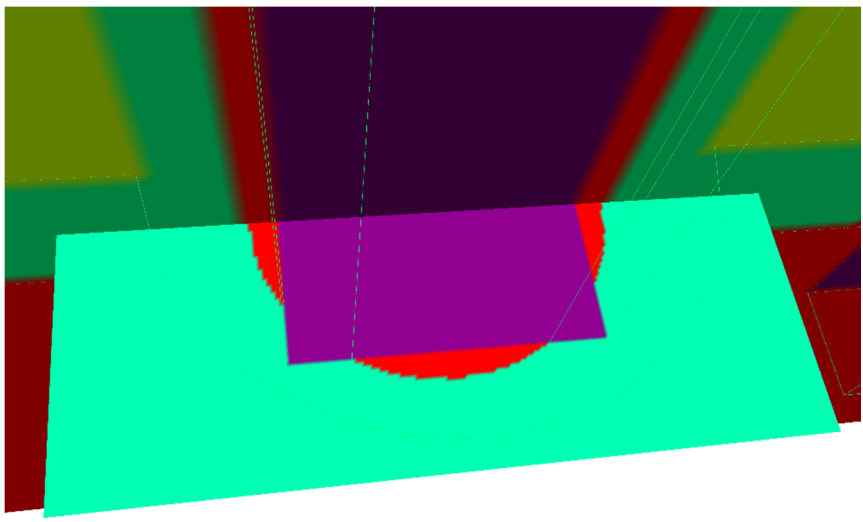

Figure 8: Detail of fig. 7.

the large and distorted boundary elements and the position of the evaluation points a huge number of nearly strong singular integrals and strong singular integrals must be evaluated. The current density is only computed inside conductors. Outside conductors the electric field strength can be calculated but the electric conductivity is zero. The results are depicted in fig. 9 and in fig. 10.

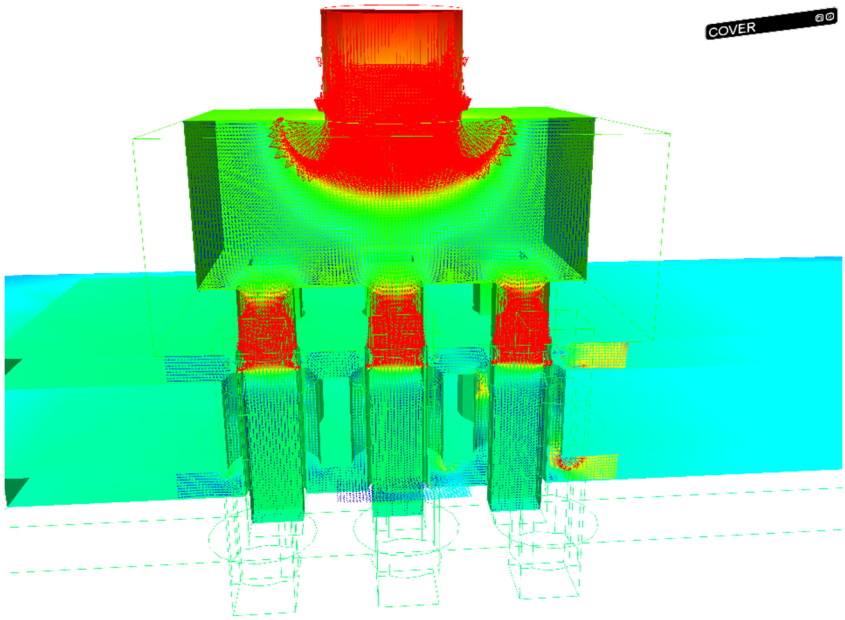

Figure 9: Electric current density.

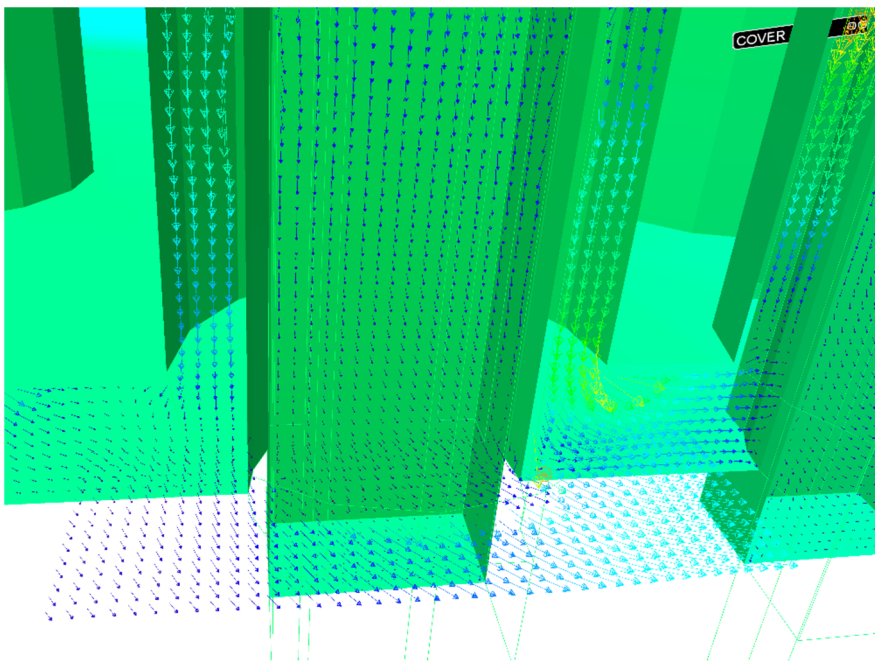

Figure 10: Detail of fig. 9.

4 Conclusions

A computation of electric current densities within conductors using a purely boundary element method approach has been presented. The domain of arbitrarily chosen evaluation points is correctly detected even in the case of complex shaped and adaptively meshed boundaries. The efficiency and robustness of the octree-based approach has been demonstrated with the help of a complex three-dimensional example. The field values are efficiently computed using a modified reversed fast multipole algorithm. A vivid visualization of the obtained results is realized by a bidirectional coupling of the collaborative visualization tool with the field computation software. In total, a very flexible and powerful post-processing in combination with pure boundary element methods is realized.

References

[1] Buchau, A., Hafla, W., Groh, F., and Rucker, W. M., *Fast Multipole Boundary Element Method for the Solution of 3D Electrostatic Field Problems*, Boundary Elements XXVI, WIT Press, pp. 369-379, 2004

[2] Greengard, L. and Rokhlin, V., *A new version of the Fast Multipole Method for the Laplace equation in three dimensions*, Acta Numerica, pp. 229-269, 1997

[3] Buchau, A., Rieger, W., and Rucker, W. M., *BEM Computations Using the Fast Multipole Method in Combination with Higher Order Elements and the Galerkin Method*, IEEE Transactions on Magnetics, vol. 37, no. 5, pp. 3181-3185, 2001

[4] Buchau, A., Rieger, W., and Rucker, W. M., *Fast Field Computations with the Fast Multipole Method*, COMPEL, vol. 20, no. 2, pp. 547-561, 2001

[5] Lang, U. and Wössner, U., *Virtual and Augmented Reality Developments for Engineering Applications*, Proceedings of ECCOMAS 2004, Jyväskylä, July 24-28 2004

[6] Buchau, A., Rucker, W., Wössner, U., and Becker, M., *Augmented reality in teaching of electrodynamics*, COMPEL, vol. 28, no. 4, pp. 948-963, 2009

[7] Hafla, W., Weinläder, A., Bardakcioglu, A., Buchau, A., and Rucker, W. M., *Efficient Post-Processing with the Integral Equation Method*, COMPEL, vol. 26, no. 3, pp. 873-887, 2007

[8] Buchau, A. and Rucker, W. M., *Meshfree Visualization of Field Lines in 3D*, 14th IGTE Symposium, pp. 172-177, Graz, 2010

[9] Buendgens, D., Hamacher, A., Hafner, M., Kuhlen, T., and Hameyer, K., *Bidirectional Coupling Between 3-D Field Simulation and Immersive Visualization Systems*, IEEE Transactions on Magnetics, vol. 48, no. 2, pp. 547-550, 2012

[10] Buchau, A., Jüttner, M., and Rucker, W. M., *Automatic domain detection for a meshfree post-processing in boundary element methods*, 15th International IGTE Symposium, Graz, pp. 386-391, 2012

[11] Goldsmith, J. and Salmon, J., *Automatic Creation of Object Hierarchies for Ray Tracing*, IEEE Computer Graphics and Applications, vol. 7, no. 5, pp. 14-20, 1987
[12] Buchau, A. and Rucker, W. M., *Fast Multipole Method accelerated meshfree Post-Processing in 3D Boundary Element Methods*, COMPUMAG 2013, Budapest, 2013
[13] Huber, Ch. J., Rucker, W. M., Hoschek, R., and Richter, K. R., *A New Method for the Numerical Calculation of Cauchy Principal Value Integrals in BEM applied to Electromagnetics*, IEEE Transactions on Magnetics, vol. 33, pp. 119-123, 1997
[14] Huber, Ch. J., Rieger, W., Haas, M., and Rucker, W. M., *The Numerical Treatment of Singular Integrals in Boundary Element Calculations*, ACES Journal, vol. 12, no. 2, pp. 121-126, 1997
[15] Open Data Protocol (OData), OASIS, Advancing open standards for the information society, www.oasis-open.org