

Feasibility of a meshfree post-processing for boundary element methods

A. Buchau & W. M. Rucker

Institut für Theorie der Elektrotechnik, Universität Stuttgart, Germany

Abstract

A substantial feature of boundary element methods is that a surface discretization of examined three-dimensional objects suffices to compute the solution of the underlying field problem. However, an auxiliary volume mesh is often created in practical applications to evaluate the solved problem and to apply classical post-processing methods, for instance to compute and visualize field lines or isosurfaces. In contrast, a meshfree post-processing approach is excellently suitable for boundary element methods but requires adapted and enhanced post-processing algorithms. Instead of a pre-computation of field values in a previously defined grid or mesh, field values are computed on demand only in points that are really needed to evaluate and to visualize the field. Here, two visualization techniques, the computation of field lines and the computation of isosurfaces, are shown to demonstrate the power of meshfree post-processing approaches using pure boundary element methods. In the case of field line computations, a relatively small number of evaluation points are defined during the determination of a field line based on a Runge-Kutta-Fehlberg method. On the other hand, information about data range of field values must be supplied in the complete space for iso-value search methods. In both cases, a hierarchical octree scheme is applied to group all boundary elements and evaluation points spatially and to provide information about field values in the complete space. The computation of field values is significantly accelerated by a coupling of the post-processing algorithms with the fast multipole method, an established compression technique for boundary element methods. Finally, the numerical examples show that the presented meshfree post-processing approaches enable evaluation and visualization of results of boundary element methods comparable to domain based methods like finite element methods.

Keywords: fast multipole methods, field line computations, isosurface computations, meshfree post-processing, visualization.



1 Introduction

In addition to an accurate and efficient solution of an electric or magnetic field problem (Buchau *et al.* [1]), a multifarious post-processing along with a vivid visualization of computed results is very important in practical applications of boundary element methods (BEM). Although a BEM requires a modelling and discretization of domain surfaces only, interesting points for field evaluations are normally located inside domains. Hence, it is necessary to provide a robust and efficient method for field computations at arbitrary points inside domains using boundary values of BEM. An established approach is to discretize all considered domains with volume elements and to pre-compute field values in the nodes of these elements (Hafla *et al.* [2]). Then, field values are further processed to obtain visualization objects, for instance field lines or isosurfaces, which are rendered in a virtual or in an augmented reality environment (Lang and Wössner [3]).

A more advanced approach is to couple directly visualization and post-processing of the field problem (Buendgens *et al.* [4]). Instead of a pre-computation of field values in the nodes of a post-processing mesh, field values are computed on demand and only in points that are actually needed for visualization. Then, the number of expensive BEM computations is significantly reduced (Buchau and Rucker [5]). Furthermore, the accuracy of this meshfree approach depends only on the accuracy of BEM integrals but not on the resolution of a post-processing mesh.

A challenge in the context of a meshfree BEM post-processing is to determine the position of an evaluation point relating to the boundary elements automatically even in the case of complex shaped domain boundaries. An efficient approach for an automatic domain detection method is to combine an adaptive octree based scheme with a ray-tracing algorithm (Buchau and Rucker [6]) to enable a visualization of field values in slides considering correct material values.

An automatic domain detection method is a prerequisite for a meshfree computation of field lines. There, a start point is defined and the next point of the field line is for instance determined using a Runge-Kutta-Fehlberg method (Buchau and Rucker [5]). Both correct material values are needed in each point of the field line and domain boundaries have to be detected reliably to visualize sharp bends of field lines at domain boundaries (Buchau and Rucker [7]). In total, a relatively small number of field points are necessary to obtain accurate field lines.

In contrast, the computation of isosurfaces requires that field values are provided in complete space. To avoid an expensive volume mesh, an octree is applied both for the isovalue search method and for the computation of field values (Buchau and Rucker [8]). Fields are described inside octree cubes using high order Taylor series expansions in spherical coordinates. Then, the computation of intersection points of the isosurface with the octree cube can be calculated accurately along with relatively low computational costs.

Here, feasibility of BEM post-processing for practical applications in electrical engineering is demonstrated. It is shown, how established visualization methods and efficient BEM computations using the fast multipole method (FMM)



(Greengard and Rokhlin [9]) are combined to provide an efficient meshfree BEM post-processing.

2 Meshfree post-processing approach

The following meshfree post-processing approach is exemplarily shown for electrostatic field problems but it is principally applicable to all kind of BEM formulations. Here, the solution of Poisson's equation for the scalar electric potential $u(\mathbf{r})$ at an arbitrary point \mathbf{r} is considered. The surface A of piecewise homogeneous, linear, and isotropic conductors and dielectrics is discretized using second order Lagrangian quadrilateral elements. The Galerkin method is applied to obtain a linear system of equations for the surface charge density σ on A using the FMM (Buchau *et al.* [1]). Finally, the electric potential is obtained by an evaluation of

$$u(\mathbf{r}) = \frac{1}{\epsilon_0} \int_A \sigma(\mathbf{r}') G(\mathbf{r}, \mathbf{r}') dA'. \quad (1)$$

$G(\mathbf{r}, \mathbf{r}')$ is Green's function of Poisson equation

$$G(\mathbf{r}, \mathbf{r}') = \frac{1}{4\pi |\mathbf{r} - \mathbf{r}'|} \quad (2)$$

and ϵ_0 is free space permittivity. The electric field strength $\mathbf{E}(\mathbf{r})$ is calculated with the help of the analytical derivative of equation (1)

$$\mathbf{E}(\mathbf{r}) = -\frac{1}{\epsilon_0} \int_A \sigma(\mathbf{r}') \text{grad} G(\mathbf{r}, \mathbf{r}') dA'. \quad (3)$$

Both equation (1) and equation (3) are valid in complete space. The correct position of the point \mathbf{r} is needed for the computation of the electric flux density

$$\mathbf{D}(\mathbf{r}) = \epsilon(\text{domain}(\mathbf{r})) \mathbf{E}(\mathbf{r}), \quad (4)$$

where ϵ is the permittivity of the material, in which \mathbf{D} is computed.

An established and easy applicable post-processing approach is to create a volume mesh and to compute the field values in the nodes of this auxiliary mesh. Field values between the nodes are typically interpolated using linear shape functions. A substantial advantage of this approach is that computation of field values and computation of visualization objects are two independent processes. Furthermore, visualization tools and methods, which have been originally developed for finite element methods (FEM), can be used.

Here, a meshfree post-processing is presented. That means field values are computed directly by an evaluation of equations (1) or (3) and not with the help of a volume mesh along with interpolation between the mesh nodes. Of course, such a meshfree approach makes sense only, if field computations are integrated into the visualization tool. Then, the position of the evaluation points is defined dynamically during the computation of the visualization objects. Advantages are that the number of evaluation points is in most cases dramatically reduced in comparison to a mesh-based approach and a high accuracy is achieved. Furthermore, the amount of data, which has to be stored and transferred between the BEM software and the visualization software, is significantly reduced.

In the following section, the main properties of the FMM are shortly summarized. Since both the FMM and most visualization algorithms are based on octree schemes, a flexible octree, which can be used for multiple purposes, is

introduced. Then, a focus is set on an automatic domain detection method, which is a prerequisite for an enhanced post-processing. Two methods are presented afterwards to show that a meshfree BEM post-processing is applicable both for evaluations in a small number of selected points and for evaluations in the complete space.

2.1 Fast multipole method and octree scheme

The FMM is an established method to compress both the matrix of the linear system of equations and the post-processing integrals like equations (1) or (3) (Buchau *et al.* [10]). It is based on a series expansion of Green's function equation (2) into spherical harmonics, which is applied along with a hierarchical octree scheme to reduce the number of computations of originally dense BEM operations significantly with approximately the same accuracy.

The first step of the FMM is to create an octree. The centre of the root cube of the octree coincides with the centre of the bounding box of all boundary elements. The edge length of the root cube is the largest dimension of that bounding box. Then, a cube is subdivided into eight smaller cubes and boundary elements are assigned to child cubes. The subdivision of cubes is continued until only one boundary element is assigned to a cube. Normally, only non-empty cubes are stored. Note it is very important for the following post-processing extensions that the implementation of the octree scheme is very flexible and well suited for additional features.

The sources on the boundary elements, here the surface charge density σ , are replaced by a set of multipoles in the centre of the corresponding octree cube

$$M_n^m = \frac{1}{\epsilon_0} \int_A \sigma(\mathbf{r}') r'^m Y_n^{-m}(\theta', \varphi') dA'. \quad (5)$$

M_n^m are the multipole coefficients of the cube and Y_n^m are spherical harmonics. r , θ , φ are spherical coordinates with origin in the centre of the considered cube. The potential outside the cube is obtained by an evaluation of the multipole expansion

$$u(\mathbf{r}) = \frac{1}{4\pi} \sum_{n=0}^L \sum_{m=-n}^n M_n^m \frac{Y_n^m(\theta, \varphi)}{r^{n+1}}. \quad (6)$$

The order L is set to $L = 9$ or $L = 10$ in practice to keep the computational costs low and to achieve a high accuracy.

The multipole coefficients of a parent cube are computed via a transformation of the multipole coefficients of its child cubes.

The potential inside a cube is calculated with the so-called local expansion

$$u(\mathbf{r}) = \frac{1}{4\pi} \sum_{n=0}^L \sum_{m=-n}^n L_n^m r^n Y_n^m(\theta, \varphi). \quad (7)$$

It is a Taylor series expansion of $u(\mathbf{r})$ in spherical coordinates.

The local coefficients L_n^m are obtained from transformation of L_n^m of its parent cube and from transformations of properly chosen M_n^m of far-field cubes.

2.2 Automatic domain detection method

An efficient automatic domain detection method (Buchau and Rucker [6]) combines the concept of classical ray-tracing algorithms based on an octree scheme (Goldsmith and Salmon [11]) with characteristic properties of BEM.



Therefore an additional octree is created, which is initialized in the same way as for the FMM. The first step of the domain detection of an evaluation point \mathbf{r}_{ep} is to add \mathbf{r}_{ep} to the existing octree. New cubes have to be created and have to be added to the octree. The cube C_{ep} of \mathbf{r}_{ep} is only subdivided, if boundary elements are within the cube or if boundary elements of neighbouring cubes intersect with C_{ep} . This guaranties that C_{ep} is as large as possible to reuse domain data of \mathbf{r}_{ep} for other points in C_{ep} .

In classical ray-tracing algorithms, an octree is especially used to filter relevant elements for intersection tests of the ray with the elements (Goldsmith and Salmon [11]). Here, a filtering is applied, too. Instead of a determination of all cubes, which are intersected by the ray, C_{ep} and some of its neighbour cubes are preselected. If elements are inside C_{ep} or intersecting C_{ep} , these elements are chosen. Otherwise, one neighbour cube with its elements is chosen. The result is a very short list of relevant elements, which are exemplarily shown in figure 1.

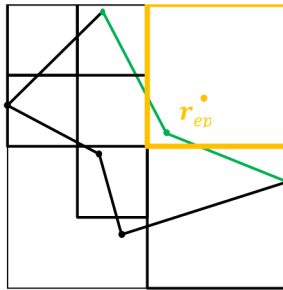


Figure 1: Filtering of relevant boundary elements for automatic domain detection.

All relevant boundary elements are sorted by their distance to \mathbf{r}_{ep} and the closest element is chosen as candidate for domain detection. The direction \mathbf{d}_r is determined using the closest point on the candidate element \mathbf{r}_{be} to \mathbf{r}_{ep}

$$\mathbf{d}_r = \mathbf{r}_{be} - \mathbf{r}_{ep}. \quad (8)$$

The remaining boundary elements are tested for intersection of the ray

$$\mathbf{r}_0 = \mathbf{r}_{ep} + \alpha \mathbf{d}_r \quad (9)$$

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

$$f(\xi, \eta, \alpha) = |\mathbf{r}_0(\alpha) - \mathbf{r}_{bet}(\xi, \eta)|. \quad (10)$$

$\mathbf{r}_{bet}(\xi, \eta)$ 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 domain 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 of

$$(\mathbf{r}_{ep} - \mathbf{r}_{be}) \cdot \mathbf{n}_{be} \begin{cases} > 0, & D_{ep} = D_{be+} \\ < 0, & D_{ep} = D_{be-} \end{cases} \quad (11)$$

\mathbf{n}_{be} is the normal vector in \mathbf{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 equation (9) is equal or nearly zero, another element is searched.

Since this search method is expensive for a huge number of evaluation points, it is combined with the octree scheme. The domain is only searched, if elements are inside C_{ep} or intersect C_{ep} . Otherwise, the domain is determined once for the complete cube C_{ep} . This enables a reuse of the already determined domain for other points in C_{ep} along with a remarkable reduction of computational costs.

The basic concept of domain detection is summarized depicted in figure 2.

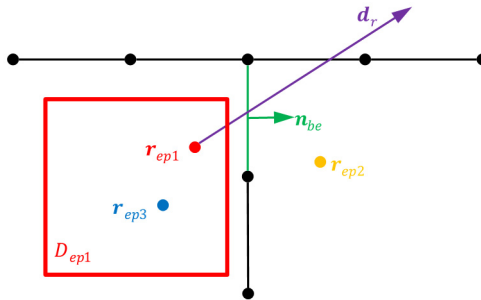


Figure 2: Basic concept of domain detection method.

2.3 Field line computation

Field lines are an excellent example for visualization objects, which require field values in only a few points and which give insight into properties of the field in complete space. Hence, it is very advantageous not to compute field values in complete space and afterwards the field line but to couple the field line computation with the field computation. Then, the number of expensive BEM evaluations is dramatically reduced and accuracy is normally higher.

A field line is defined by the differential equation

$$\frac{\partial \mathbf{r}(t)}{\partial t} = \mathbf{v}(\mathbf{r}(t)). \tag{12}$$

$\mathbf{v}(\mathbf{r}(t))$ is a vector field, for instance \mathbf{E} . The solution of eqn. (12) is

$$\mathbf{r}(t) = \mathbf{r}_0 + \int_0^t \mathbf{v}(\mathbf{r}(\tau)) d\tau. \tag{13}$$

The numerical solution of equation (13) is computed using a Runge-Kutta-Fehlberg (RKF) method (Fehlberg [12]). There, the error and the step size is automatically controlled by an evaluation of a 5th order method with an embedded 4th order method. Since both 4th and 5th order method use the same coefficients and the same BEM evaluation points, RKF method is very efficient.

In practice, a start point of a field line is defined and the other points of the field line are determined using a RKF method. That means, the position of evaluation points is not known in advance of the post-processing but is defined during the runtime of the post-processing. Hence, a meshfree post-processing is well suited.

Here, a modified FMM algorithm is advisable. Evaluation points are sequentially added to the octree. First, position of each evaluation point is

determined using the position detection method described in the previous section. Then a reversed FMM algorithm is executed. A reversed algorithm is more efficient than the classical algorithm. The reason is that the next point of a field line is usually close to the previous one. In many cases, the new evaluation point is in a cube of already existing evaluation points. That means, the local expansion is available and only some near-field computations must be evaluated additionally. If a new cube is created, its local expansion can be efficiently calculated using multipole and local expansions of its neighbour cubes. In contrast to the classical algorithm, all computations start at the evaluation point instead of the sources. Hence, the number of cubes, which must be passed during the algorithm, is much smaller.

Special treatment of the next point and of the step-size is required near domain boundaries. A crossing of a domain boundary is detected using the domain data of the evaluation point. If a domain boundary is detected, the step-size is reduced to approach the domain boundary with small steps (figure 3). Steps across a domain boundary are rejected. The last small gap between the field line and the domain boundary is closed by a linear extrapolation of the field line. The intersection point with the domain boundary is the start point of the field line in the next domain. Finally, field lines with sharp bends at domain boundaries are obtained.

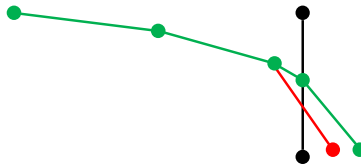


Figure 3: Illustration of reduction of step size near domain boundaries.

2.4 Isosurface computation

In contrast to the above described field line computation, some visualization techniques, for instance isosurfaces, need field values available in complete space. As already discussed, this requirement is slightly contrary to the idea of BEM. Here, an isovalue search method is directly coupled with the BEM post-processing. Volume data is provided with the help of an octree and FMM series expansions.

An isosurface computation consists of two important steps. First, an isovalue search method is applied to determine a cloud of points, which are all lying on the isosurface. The second step is to connect all these points to a meshed surface.

An established isovalue search method is based on an octree scheme. Typically, numerically computed or measured field values are provided in the considered space using a volume mesh. The octree enables a fast and efficient search of octree cubes, which have field values in the range of the searched isovalue (Wilhelms and van Gelder [13]). Intersection points of the cubes or of volume elements with the isosurface are simply obtained by a linear interpolation along cube edges or along volume element edges. Improved algorithms result in high quality isosurfaces, which can be used for further computations [14, 15].

Here, field values inside domains are not explicitly available. The idea is to provide volume data with the help of an octree and the FMM. Therefore, an octree is created and all cubes including empty cubes are stored. To consider the surrounding air domain, too, additional cube layers are added around the root cube (figure 4). In total, two octrees are created. The red cubes in figure 4 belong to the so-called inner octree, which is essentially a dense FMM octree. The green cubes are added to the inner octree to treat fringing fields. All computations inside the inner octree are performed by classical FMM algorithms. The blue cubes are assigned to the so-called outer octree. Here, it suffices to compute all field values with only one multipole expansion with origin in the centre of the outer octree.

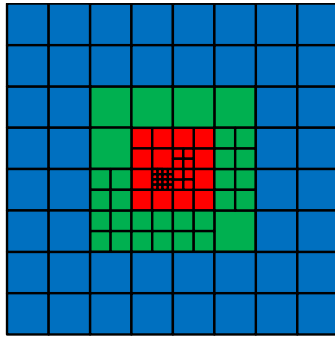


Figure 4: Octree scheme for geometrical subdivision of total space.

The first step of the meshfree isovalue search method is to determine the range of the field values inside each cube of the two dense octrees. This is exemplarily discussed for the scalar electric potential $u(\mathbf{r})$. Since $u(\mathbf{r})$ has extrema only at boundary elements or at the corners of a cube, it suffices to compute $u(\mathbf{r})$ in these points. Fortunately, most cubes are in the far-field of boundary elements and FMM series expansions describe $u(\mathbf{r})$ completely. Afterwards, a filtering of relevant cubes is executed very efficiently.

A significant difference of the meshfree isovalue search method in comparison to classical isovalue search methods is that field values between the corners of a cube or along element edges are not linear interpolated but calculated with high accuracy using high order FMM series expansions. Hence, a high accuracy for relative large domains is obtained. That means the number of cubes is in practice much smaller than the number of elements of a comparable volume mesh.

The intersection points of the isosurface with the cube edges are computed using a line search method along the relevant cube edges. Since the cubes are relatively large, an isosurface mesh based on these intersection points only, is very coarse. Hence, an additional refinement is done. Therefore, further search lines parallel to the cube edges are added and the intersection points of these lines with the isosurface are computed, too (figure 5). Of course, a refinement is also performed inside the cube (figure 6). The number of additional search lines is adjustable can be adjusted in dependency of the viewer position in the rendering software, too.

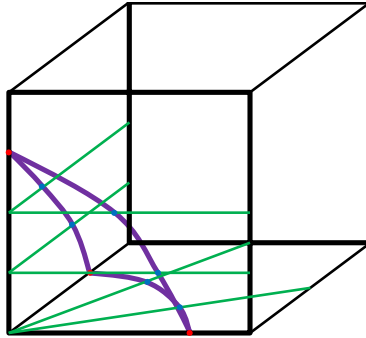


Figure 5: Computation of intersection points at cube edges and cube faces.

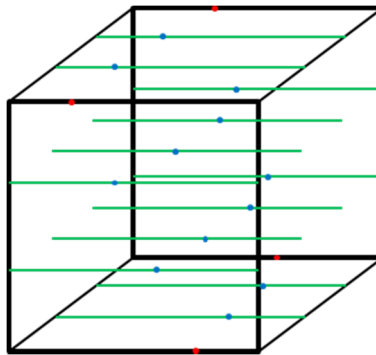


Figure 6: Refined computation of intersection points inside a cube.

Finally, all computed intersection points must be connect by a surface mesh. The points are the nodes of linear triangular elements, which are the standard element in computer graphics. Unfortunately, the creation of a mesh of a point cloud is a challenging task. Advancing front algorithms are robust but expensive [16, 17]. Since the nodes are fixed, an adapted advancing front algorithm is advisable.

3 Numerical results

A high voltage insulator is studied to demonstrate feasibility of meshfree BEM post-processing methods. The electric field between the two electrodes is computed using an indirect BEM formulation based on surface charge densities on the surfaces of the electrodes and on the dielectric insulator. In total, all complex shaped surfaces were discretized with 2352 boundary elements. The corresponding linear system of equations with 7204 degrees of freedom was solved iteratively using GMRES with a Jacobi preconditioner in 37 s on 4 cores of an Intel Core i7-3820QM CPU with 2.7 GHz and 16 GB main memory at 1600 MHz.

The bidirectional communication between the open source visualization tool COVISE (Lang and Wössner [3]) and the BEM post-processing tool, which is implemented in C#, is established using Open Data Protocol (OData) [18].

First, the automatic domain detection method is tested. In figure 7, a visualization of detected domains is presented, where one quarter of the configuration was removed for graphical purposes. Each colour represents a domain. The surface charge density is shown by a colour plot at the surface of the electrodes and at the surface of the insulator. The determination of the domain in 72,561 points took 33 s using a single core.

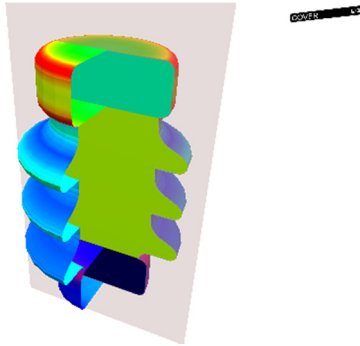


Figure 7: Detected domains visualized in a slice.

In figure 8 computed field lines of the electric field strength are depicted. The computation of 8 field lines took 45 s. It can be clearly seen that sharp bends of field lines at domain boundaries are obtained.

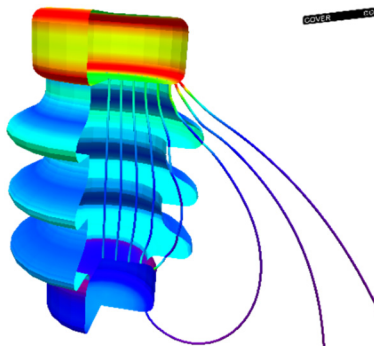


Figure 8: Field lines of the electric field strength.

An isosurface of the electric potential is shown in figure 9. The mesh of the isosurface is relatively coarse but the nodes of these elements are at the right position. Unfortunately, the implemented mesh generator is not robust enough to connect the nodes in the case of a refinement correctly. Here, 612 cubes of 6778 cubes were selected for the isovalue search. In total, the potential was computed

in 50,648 points to obtain the data range of the cubes and to determine the intersection points with the help of a line search method. The number of evaluation points would be much larger for a comparable volume mesh. Note, that in the current implementation field values are computed separately for each cube. The computational costs can be further reduced, if values at common edges or common corner are reused in neighbour cubes.



Figure 9: Isosurface of the electric potential.

4 Conclusion

Boundary element methods are not only well suited for an efficient solution using surface models of studied problems, but impressive post-processing of BEM solutions is possible, too. Especially a meshfree post-processing approach reduces the number of expensive BEM evaluations significantly and enables a bidirectional coupling of a BEM tool with a visualization tool. A combination of established visualization algorithms, accurate BEM computations, and FMM compression technique results in an efficient post-processing approach both for evaluations in a small number of points, e.g. field lines, and for evaluations in the complete space, e.g. isosurfaces. A robust meshfree domain detection method ensures that correct material values are used and domain boundaries are reliably detected. Summarized, a meshfree BEM post-processing provides comparable features as known for numerical methods, which are based on a domain discretization.

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] Hafla, W., Weinklader, 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.

- [3] Lang, U. and Wössner, U., Virtual and Augmented Reality Developments for Engineering Applications, Proceedings of ECCOMAS 2004, Jyväskylä, July 24–28 2004.
- [4] 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.
- [5] Buchau, A. and Rucker, W. M., Meshfree Visualization of Field Lines in 3D, 14th IGTE Symposium, pp. 172-177, Graz, 2010.
- [6] Buchau, A. and Rucker, W. M., The meshfree computation of stationary electric current densities in complex shaped conductors using 3D boundary element methods, Boundary Elements and Other Mesh Reduction Methods XXXVII, WIT Transactions on Modelling and Simulation, vol. 57. pp. 121-133, 2014.
- [7] Buchau, A. and Rucker, W. M., Meshfree Computation of Field Lines Across Multiple Domains Using Fast Boundary Element Methods, IEEE Transactions on Magnetics, vol. 51, no. 3, 2015.
- [8] Buchau, A. and Rucker, W. M., A meshfree isovalue search method for boundary element methods, COMPUMAG 2015, Montreal, 2015.
- [9] Greengard, L. and Rokhlin, V., A new version of the Fast Multi-pole Method for the Laplace equation in three dimensions, Acta Numerica, pp. 229-269, 1997.
- [10] 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.
- [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] Fehlberg, E., Klassische Runge-Kutta-Formeln fünfter und siebenter Ordnung mit Schrittweiten-Kontrolle, Computing, vol. 4, no. 2, pp. 93-106, 1969.
- [13] Wilhelms, J. and van Gelder, A., Octrees for Faster Isosurface Generation, ACM Transactions on Graphics, vol. 11, no. 3, pp. 201-227, 1992.
- [14] Schreiner, J., Scheidegger, C. E. and Silva, C. T., High-Quality Extraction of Isosurfaces from Regular and Irregular Grids, IEEE Transactions on Visualization and Computer Graphics, vol. 12, no. 5, pp. 1205-1212, 2006.
- [15] Dey, T. K. and Levine, J. A., Delaunay meshing of isosurfaces, The Visual Computer, vol. 24, no. 6, pp. 411-422, 2008.
- [16] Löhner, R., Progress in Grid Generation via the Advancing Front Technique, Engineering with Computers, vol. 12, pp. 186-210, 1996.
- [17] Hornung, A. and Kobbelt, L., Robust Reconstruction of Watertight 3D Models from Non-uniformly Sampled Point Clouds Without Normal Information, Eurographics Symposium on Geometry Processing, 2006.
- [18] Open Data Protocol (OData), OASIS, Advancing open standards for the information society, www.oasis-open.org.

