

Visualizing Edge-Conforming Discrete Field Quantities in Electromagnetic Field Problems with Interfaces

G. K. Karch¹, F. Sadlo¹, H. Songoro², E. Gjonaj², T. Weiland², T. Ertl¹

1: Visualization Research Center, University of Stuttgart, Germany

2: Technische Universitaet Darmstadt, Institut fuer Theorie Elektromagnetischer Felder, Germany

Abstract

Finite element-based electromagnetic field simulation strongly benefits from using edge-conforming representations of the electric field. In this paper we address the visualization of discrete field data resulting from such simulations on 10-node quadratic tetrahedral grids. The use of higher-order grids enables, on the one hand, the accurate approximation of curved interfaces between electromagnetic materials, and on the other hand, it allows for a more accurate computation of derived quantities such as Coulomb forces and related surface tensions. However, a major drawback so far has been the lack of appropriate visualization techniques—common visualization systems do not support this type of data—necessitating a resampling step with all the involved drawbacks, including artifacts in the form of imposed continuity across material boundaries. We introduce a visualization framework implemented as a set of ParaView plugins that evaluates edge-conforming data by means of vector basis functions. Based on this framework we present different visualization approaches for the investigation of the electric field at material boundaries. We demonstrate their utility using electrohydrodynamics simulations of a water droplet on the surface of a high voltage insulator, representing a two-phase flow problem driven by strong electric fields.

Introduction

The investigation of electromagnetic fields on material boundaries is of special interest in the development of structures that are exposed to strong electromagnetic fields. While the electric field at the interfaces is continuous tangential to the material surface, it commonly exhibits discontinuities in normal direction. These discontinuities pose problems in numerical simulation with finite element methods. The traditional approach, based on node-based elements, imposes continuity at material boundaries due to the component-wise interpolation, resulting in an undesired smooth modeling of quantities across the interface. Therefore, edge-based representation of vector quantities was introduced in finite element methods. In the resulting approach, the simulated quantity is interpolated by means of vector shape functions that provide several characteristics important for the representation of the electric field at material boundaries. Since neighboring elements share tangential components on the edges, the tangential field is continuous across those elements. On the other hand, the shape functions cause the tangential component to vanish at the element faces opposite to an edge (see Figures 2a) and 2b) for an illustration), thus allowing for discontinuities in the normal component, which in turn enables correct representation of the electric field at interfaces. Additionally, the introduction of nonlinear elements, such as quadratic tetrahedra, enables a more accurate approximation of curved surfaces, which has particular importance for the computation of the electric field, as the sharp, piecewise linear representation would artificially amplify the electric field magnitude. Consequently, the nonlinear elements significantly reduce the required number of tetrahedra for the approximation of curved boundaries.

Current visualization frameworks, however, do not allow for correct representation of the electric field resulting from edge-based elements—due to their component-wise interpolation they do not provide accurate representation, and more important, they miss field discontinuities across the interfaces. Thus, in this paper we present a framework that is able to correctly visualize the simulated field, also at material boundaries. The field is directly evaluated from the vector shape functions, taking higher-order elements (quadratic tetrahedra) into account, thus providing visualizations consistent with the simulation model. We exemplify the utility of our approach using electrohydrodynamics simulations of a water droplet on the surface of a high voltage insulator.

Related Work

Edge elements were introduced into the finite element method in [10] and [3]. Bossavit [3] identified Whitney elements [20], which are widely used in finite element simulations, as a natural discretization method for eddy currents. The basis functions of edge elements were generalized to arbitrary order for the finite element method, and the resulting convergence was analyzed in [19]. The advantages and disadvantages of edge elements as well as their applications are discussed in [18] and [9]. Isoparametric elements described in [4] represent an alternative that allows for efficient and accurate higher-order computations.

In the field of visualization, several techniques have been developed to address appropriate representation of higher-order elements. Wiley et al. [21] developed a technique for direct ray casting of curved

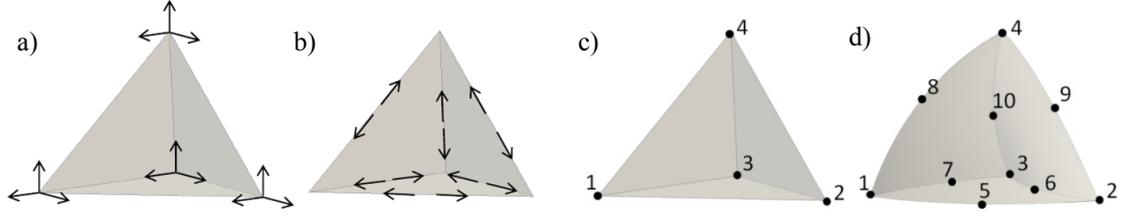


Figure 1. Data representation on tetrahedral elements. a) Node-based data representation, b) edge-based data representation, c) linear tetrahedron, and d) quadratic tetrahedron with additional mid-edge points.

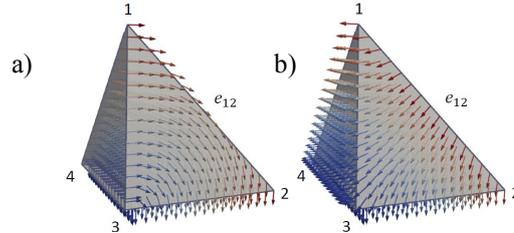


Figure 2. Vector basis functions on tetrahedra. a) First order basis function $\bar{N}_{1,2}^1$ on edge $e_{1,2}$ and b), second-order basis function $\bar{N}_{1,2}^{13}$ on edge $e_{1,2}$.

quadratic elements without prior tessellation into linear elements. For cell-based polynomial fields, isosurface extraction from higher-order finite elements was presented in [13], where adaptive mesh refinement is employed for accurate representation. A technique for isosurface extraction providing a trade-off between rendering speed and quality was suggested by Pagot et al. [12], based on a particle transport along the gradient field, and ray casting in the neighborhood of the final location of the particles. Schroeder et al. [15] addressed the complexity of higher-order basis functions from p- and hp-adaptive methods by employing an automatic tessellation technique with recursive edge-based subdivision. Direct visualization of discontinuous Galerkin simulations was presented in [17], where an adaptive sampling technique is used for high quality volume rendering, and utilization of a GPU cluster allows for interactivity. Feature extraction from discontinuous Galerkin simulations based on the parallel vectors operator was proposed in [11]. A solution for the visualization of non-conforming meshes, based on point-based rendering, was developed by Zhou and Garland [22]. Isosurfaces from higher-order elements can be also visualized in a point-based manner [8], in which case the costly inverse mapping to evaluate the basis functions is avoided. Most recent works in the field of higher-order finite element visualization include a ray casting method with pre-computation of world-element space transformation [2], as well as ray casting with depth peeling [6]. Interface reconstruction for multiphase flow simulation was visualized in [5]. For visualization of electromagnetic fields, examples include the visualization of the field in Tokamak reactors [14], visualization of the coronal field [7], and topological analysis of magnetic fields [1]. It is worth noting that edge elements have not been introduced into visualization so far, and it is thus the aim of this paper to do so.

Quadratic Tetrahedra

Quadratic tetrahedra are particularly suitable for finite element-based simulations involving electromagnetic field problems with interfaces, since they allow for curved edges and faces. Considering that the elements can adapt to relatively strong deformations, mesh refinement can be avoided for many problems. This at the same time allows for a significant reduction of the required number of elements—curved parts of the simulation domain no longer require strong refinement for accurate computation of the electric field. These elements, however, are more difficult to implement, and the mesh formation takes significantly longer than for linear elements [23]. Quadratic tetrahedra have variable metric, i.e., the Jacobian determinant is not constant over tetrahedron. This further complicates computations, e.g., point location inside tetrahedra.

A quadratic tetrahedron is illustrated in Figure 1d) (cf. Figure 1c) for comparison with a linear tetrahedron). The element is defined by 10 nodes, each carrying three components of the vector field, resulting in 30 degrees of freedom. The shape functions corresponding to the nodes of quadratic tetrahedra are defined as:

$$N_l = L_l(2L_l - 1), \text{ for } l=1, \dots, 4 \quad (1)$$

$$N_5 = 4L_1L_2, \quad N_6 = 4L_2L_3, \quad N_7 = 4L_1L_3, \quad N_8 = 4L_1L_4, \quad N_9 = 4L_2L_4, \quad N_{10} = 4L_3L_4. \quad (2)$$

The world coordinates (x_1, x_2, x_3) of a point inside a quadratic tetrahedron can be obtained from barycentric coordinates (L_1, L_2, L_3, L_4) using:

$$x_i = \sum_{l=1}^{10} x_{i,l} N_l \quad (3)$$

where x_i represents the i -th world coordinate of the sought point, and $x_{i,l}$ represent the i -th coordinate of the l -th node of the tetrahedron. Since the computation of barycentric coordinates from world coordinates is nontrivial for quadratic tetrahedra, iterative methods, such as Newton-Raphson iteration, are commonly used.

Edge-Conforming Data Representation

In the node-based finite element method, the data are stored on the nodes of the elements. As illustrated in Figure 1a), for a linear tetrahedral element there are four nodes, each carrying three components of the vector field. In edge-based representation, on the other hand, the data are attached to the edges of an element, as shown in Figure 1b). Here, each edge stores two tangential components of the vector field. In both representations there are in total twelve degrees of freedom for a linear tetrahedron. It is worth noting, however, that incomplete degrees are widely used in simulations of the electromagnetic field with interfaces in order to handle discontinuities at material boundaries. To model the electric field in the edge-based element, Whitney vector basis functions are employed. The first-order incomplete Whitney shape functions are defined as

$$\vec{N}_{i,j}^l = L_i \nabla L_j - L_j \nabla L_i, \text{ for } l=1,\dots,6, \quad (4)$$

where i and j are the nodes defining edge (i,j) , L_i and L_j are barycentric coordinates of nodes i and j , and ∇L_i and ∇L_j are the constant gradients of the barycentric coordinates. The basis functions are then used to interpolate the electric field \vec{E} from the tangential components c :

$$\vec{E} = \sum_{l=1}^6 c_l \vec{N}^l. \quad (5)$$

To extend the element to the complete first order, additional edge functions are used:

$$\vec{N}_{i,j}^l = L_i \nabla L_j + L_j \nabla L_i, \text{ for } l=7,\dots,12. \quad (6)$$

The second-order basis functions are defined for edges:

$$\vec{N}_{i,j}^l = L_j(2L_i - L_j)\nabla L_i - L_i(L_i - 2L_j)\nabla L_j, \text{ for } l=13,\dots,18 \quad (7)$$

and faces:

$$\vec{N}_{i,j}^l = L_i L_j \nabla L_k - L_i L_k \nabla L_j, \text{ for } l=19,\dots,22 \quad (8)$$

$$\vec{N}_{i,j}^l = L_j L_k \nabla L_i - L_j L_i \nabla L_k, \text{ for } l=23,\dots,26 \quad (9)$$

$$\vec{N}_{i,j}^l = L_j L_k \nabla L_i + L_i L_k \nabla L_j + L_i L_j \nabla L_k, \text{ for } l=27,\dots,30. \quad (10)$$

In Figure 2 two edge functions are shown. Both basis functions have non-zero tangential component along the edge $(1,2)$, and zero tangential component on the faces $(1,3,4)$ and $(2,3,4)$. The edge-conforming representation has thus several characteristics that are particularly useful in electromagnetic simulations. First and foremost, they do not impose continuity more than required by the physics of the given phenomena. This means they can model discontinuities of the normal component at the interfaces while ensuring continuity of the tangential field component. It is worth noting that continuity across the interfaces must be accomplished explicitly. Additionally, they allow for comparably simple local p-adaptation, and are well suited for the solution of partial differential equations for the field intensities derived from Maxwell's equations.

Transformation from Node-based to Edge-based Representation

In our visualization framework, we need to evaluate Equation 5 to visualize the electric field inside a quadratic tetrahedron. Hence, the tangential components must be either provided directly from the simulation or computed in a pre-processing step from the node-based representation. In the latter case, the electric field must be stored on each of the 10 nodes of a quadratic tetrahedron, such that two tetrahedra may share node coordinates, but their electric field data is defined separately for each node. For example, if there are three elements sharing one node, the node position might be defined once, but the electric field components must be three times, once for each tetrahedron. This allows us to compute the tangential components by solving the following linear system:

$$\mathbf{A} \mathbf{c} = \mathbf{b}. \quad (11)$$

\mathbf{A} is assembled from the local matrices of each tetrahedron, while the local matrix elements are defined as:

$$a_{l,m} = \sum_{q=1}^Q (\vec{N}^l \cdot \vec{N}^m) w_q \det(J_q), \quad (12)$$

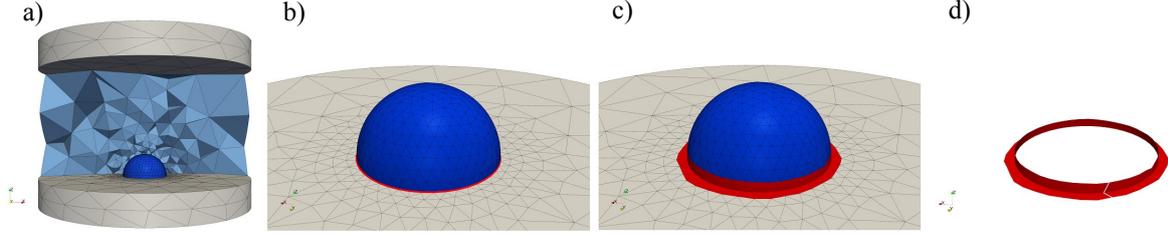


Figure 3. Stages for the visualization of the electric field on the contact line. a) Simulation domain with three materials: air (light blue), water drop (dark blue), and insulator (gray). b) The contact line (red) between drop and insulator is extracted. c) The drop cells and insulator cells adjacent to the contact line are extracted. d) The extracted contact region is cut before the “unrolling” stage illustrated in Figure 4.

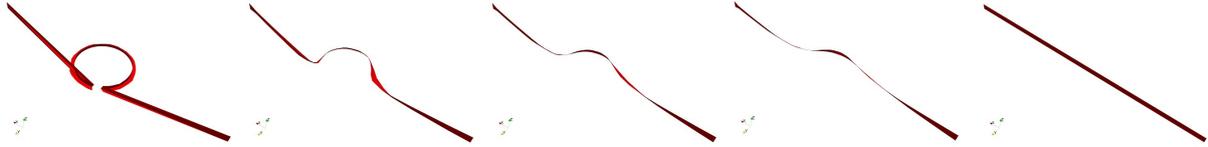


Figure 4. The process of Laplacian smoothing for the contact region (from left to right). The cut points from each side are positioned away from each other (left). Iterative smoothing applied to the rest of the mesh points (middle). Final rectification applied to the points, so that they lie on a plane (right).

where Q is the number of quadrature points, w_q is the weight of quadrature point q , and $\det(J_q)$ is the Jacobian determinant at point q , with respect to barycentric coordinates. The full expression for the Jacobian determinant in quadratic tetrahedra can be found in [23]. The vector \mathbf{b} is assembled from

$$b_l = \sum_{q=1}^Q (\vec{N}^l \cdot \vec{E}_q) w_q \det(J_q), \quad (13)$$

where the electric field \vec{E}_q at point q is computed from Equation 3. l and m are local basis function indices which must be transformed to global indices i and j to assemble the matrix \mathbf{A} and vector \mathbf{b} :

$$A_{i,j} = a_{l,m}, \quad b_i = b_l. \quad (14)$$

After the assembly, the system can be solved for \mathbf{c} using a linear algebra package (e.g., PETSc).

Visualization

We implemented our visualization methods as plugins in the visualization environment ParaView [24]. The data is read either in node-based form or edge-based form using our custom reader plugin. In the former case, we transform the data to edge representation, as described in the previous section. This was done for the simulation results visualized in this paper, as they were converted to node-based representation at simulation output. Note that the mapping between local and global indices is also used later in Equation 5 to find the global tangential components for the local indices of the shape functions.

We have developed two visualization methods using our framework. In the first one, we evaluate and visualize the electric field on the material interface—in our case, on the interface between air and the droplet in a strong electric field (Figure 3a)). In the second one, we visualize the time-dependent electric field within a region near the contact line between droplet and the insulator on which the droplet is located. To facilitate the analysis of this region, which is important for the investigation of the dynamics, including static discharges [16], we “unroll” the region of the droplet interface and insulator surface near the contact line to create a rectangular strip, stack the resulting strips from consecutive time steps, and interpolate them to a space-time representation to provide an appropriate visualization of the space-time dynamics. The details are given in the next two sections.

Electric Field on the Interface

For the visualization of the electric field on the air-drop interface, the quadratic tetrahedra adjacent to the interface between the two materials M_1 (air) and M_2 (water) must be found from the material information provided with the simulation data. For each tetrahedron, we check if it belongs to M_1 , and if so, we determine if it faces a tetrahedron of material M_2 . If this is the case, we store the indices of both elements inside a list. Subsequently, a triangular mesh that represents the interface is extracted from the shared faces of the quadratic tetrahedra stored in the list. Please note that for the visualization we approximate curved tetrahedral faces from the simulation by linear triangles, i.e., we employ a parametric representation. Each triangle in the mesh is then subdivided into smaller coplanar triangles. For each resulting triangle we compute the middle point, and evaluate

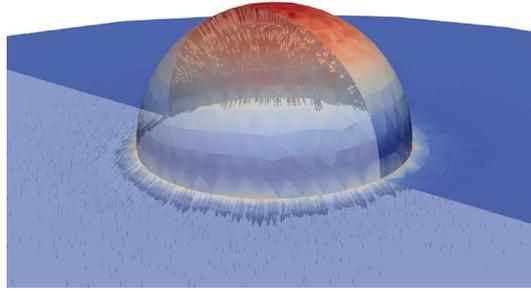


Figure 5. Visualization of a droplet on an insulator in the presence of a strong electric field. The insulator and droplet are made partially transparent to reveal the electric field direction (glyphs with size scaled according to the field strength). The electric field is strongest at the top of the droplet (red) as well as at the interface.

its barycentric coordinates in the original triangle. Since the coordinates are computed on tetrahedral faces (i.e., $L_i = 0$ on the face opposite to node i), it is not necessary to employ computationally expensive point location algorithms. The shape functions are then evaluated using Equations 4 and 6-10, and finally the electric field is evaluated using Equation 5 and stored on the resulting mesh. See Figure 5 for a result. The whole procedure was implemented as a ParaView plugin.

Space-Time Visualization of Electric Field around Contact Line

To extract the contact line between air, water, and insulator (Figure 3b)), we first extract two interfaces: interface I_1 between M_1 (air) and M_2 (water), and interface I_2 between M_1 and M_3 (insulator). From these interfaces we take only those triangles that share at least one vertex with a triangle from the other interface. The resulting triangular mesh is shown in Figure 3c). In the next step we cut the mesh (Figure 3d)) and duplicate the cut points so that on each side of the cut the triangles are topologically disconnected. The cut points from one side are then positioned away from the other cut points (Figure 4, left). An iterative Laplacian smoothing, illustrated in the rest of Figure 4, is performed on the other points of the mesh strip, placing each point on the average of direct neighbors at each iteration step. Only neighbors on the perimeter are taken into account for the points laying on it, otherwise the whole mesh would collapse into one line. After last iteration the strip is rectified, since the Laplacian smoothing does not converge to a perfect plane after reasonable amount of iterations. The extracted contact region, shown in Figure 3d), and the smoothed counterpart (Figure 4, right) are subdivided into fine triangular meshes, such that for each cell in the original geometry there is a corresponding cell in the smoothed one. Finally, we sample the data on the original subdivided mesh, as described in the previous section, and assign the values to the smoothed mesh.

To obtain a space-time representation, a 3D rectilinear grid is created such that the resolution in T direction corresponds to time (i.e., number of simulation time steps), while the x - z plane is aligned with the “unrolled” contact line strips. For the simulation data we used a $1024 \times 51 \times 25$ grid (i.e., with 51 time steps and 1024×25 samples per strip). We then sample the smoothed (triangulated) strip for each time step in the x - z plane of the 3D grid at the respective T (time step) position. As a result, the strips from consecutive time steps are sampled and then stacked on each other, resulting in a 3D grid.

To reveal the temporal variation of the electric field, isosurface extraction and volume rendering are employed on the space-time stack. The isosurface extraction (see Figure 6) is done with the Marching Cubes algorithm using the “Contour” plugin from ParaView. Volume rendering (see Figure 7) displays the 3D scalar data in a transparent form, where opacity and color is defined for each scalar value using a so-called transfer function. The final image is achieved by a compositing technique, based on the rendering equation of the emission-absorption light model, using the “Volume” data representation from ParaView. For the space-time stacking and 3D mesh generation we developed a respective ParaView plugin based on our framework.

Results and Discussion

The basic visualization of electro-hydrodynamic simulations of a droplet in the presence of strong electric fields is illustrated in Figure 5. The simulation was obtained by a coupled fluid dynamics and electric field solver: the finite element method step determines the electric field, which is then used to compute the dynamics of the droplet with the finite volume method according to the induced flow. The solution of the fluid dynamics stage is fed back to find the electric field in the domain. The process is repeated for each simulation step. Figure 5 shows one time step from the simulation. The droplet is positioned on an insulator, within a strong, electric field. The electric field lines in the air (not shown in the figure) are vertical. As can be seen, the electric field is strongest at the top of the droplet. What is more interesting, however, is that the field strength on the contact line is amplified due to the sharp corner formed by the droplet-insulator contact region. Since this region is of

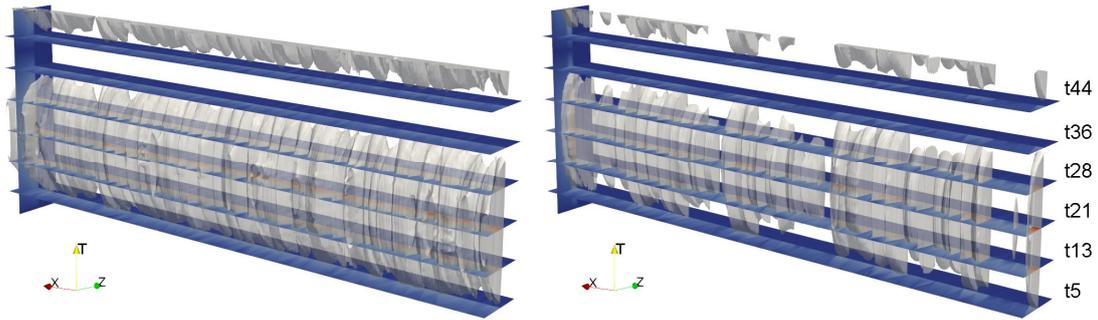


Figure 6. Visualization of the time-dependent electric field a) tangent and b) normal to the interface around the contact line, visualized with cross sections (horizontal sections) of the space-time representation. Time evolution from bottom to top (T -axis, respective time steps on the right). Air-insulator interface is at the front, drop-air interface in the back. The transparent isosurface and the vertical plane show the spatial and temporal variation of the electric field, respectively. Interestingly, the normal component varies stronger than the tangent one.

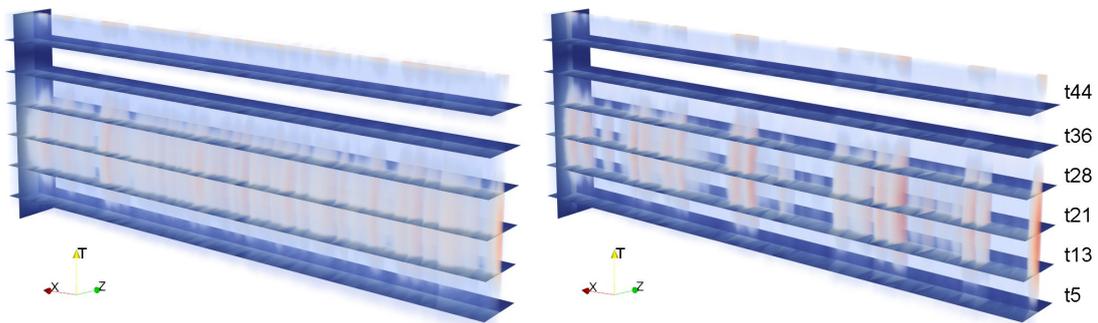


Figure 7. Space-time visualization of the time-dependent electric field, as in Figure 6. Volume rendering is shown instead of isosurfaces. The magnitude of the electric field is considerably lower for the drop-air interface (back side of the planes).

particular interest for the domain experts, the space-time visualization technique facilitates the investigation of the time-dependent electric field near the contact line.

In Figures 6 and 7, the space-time visualization of the electric field in the contact region is provided. The images on the left show the electric field tangent to the drop (insulator) surface, while the images on the right show the field normal to the interface. Six selected simulation time steps are additionally displayed (see previous section) and a vertical color-coded cross section shows the time evolution of the field at one position of the contact region. The air-insulator interface is oriented to the front. Figure 6 additionally shows a transparent isosurface of the magnitude of the tangent (normal) field component, while Figure 7 displays the magnitude by volume rendering. The visualization shows interesting characteristics of the time-dependent vector field. It exhibits periodic intervals, as indicated by the isosurface: the high magnitude regions disappear before time step 36 to appear again after time step 44. Volume rendering reveals that the tangential field has uniform distribution along the contact line, while the normal field is characterized by stronger spatial variation. It is also evident that the electric magnitude is considerably weaker on the drop boundary. The static representation of the temporal process enables insights into the overall behavior of the electric field along the drop-insulator contact region.

The most demanding part of our framework is the computation of tangential components from the node-based representation which took more than 7 minutes for about 11000 simulation cells. The computation of the droplet interface took 19 seconds, while one time step of the space-time representation took about 7 seconds.

Conclusion

We presented methods for the visualization of electromagnetic field simulations with interfaces. Our framework is tailored towards the edge-based representation for correct visualization of the vector quantities at the material boundaries. We exemplified the benefits of our methods using data from electro-hydrodynamic simulations of a droplet in the presence of strong electric fields. Since the contact line between the droplet and the insulator is of particular importance for the investigation of the dynamics, including static discharges, we have provided a space-time visualization of the electric field along the circular contact region which avoids occlusion and provides a static representation of the time-dependent process. As future work, we would like to address the performance of our method—our prototype is using only one processor for the evaluation of the tangential components. Using multithreading would significantly speed up the computation. GPU parallelization of the

most demanding stages would also be a promising approach. Additionally, we would like to generalize our framework to other electromagnetic field problems with interfaces.

Acknowledgements

The authors would like to thank the Deutsche Forschungsgemeinschaft (DFG) for support within the Collaborative Research Center SFB-TRR 75.

References

- [1] Bachthaler, S., Sadlo, F., Weeber, R., Kantorovich, S., Holm, C., Weiskopf, D.: Magnetic Flux Topology of 2D Point Dipoles. *Computer Graphics Forum* 31, 3 (2012), pp. 751-760
- [2] Bock, A., Sunden, E., Liu B., Wunsche, B., Ropinski, T.: Coherency-Based Curve Compression for High-Order Finite Element Model Visualization. *IEEE Trans. Vis. Comput. Graphics* 18, 12 (2012), pp. 2315-2324
- [3] Bossavit, A.: A rationale for 'edge-elements' in 3-D fields computations. *IEEE Trans. Magn.* 24, 1 (1988), pp. 74-79
- [4] Irons, B.M., Zienkiewicz, O.C.: The Isoparametric Finite Element System: A New Concept in Finite Element Analysis. Royal Aeronautical Society, (1969)
- [5] Karch, G.K., Sadlo, F., Rauschenberger, P., Meister, C., Eisenschmidt, K., Weigand, B., Ertl, T.: Visualization of Piecewise Linear Interface Calculation. In *Proceedings of IEEE Pacific Visualization Symposium*, (2013), pp. 121-128
- [6] Liu, B., Bock, A., Ropinski, T., Nash, M., Nielsen, P., Wüsche, B. C.: GPU-accelerated direct volume rendering of finite element data sets. In *Proceedings of the 27th Conference on Image and Vision Computing New Zealand*, (2012), pp. 109-114
- [7] Machado, G., Sadlo, F., Müller, T., Müller, D., Ertl, T.: Visualizing Solar Dynamics Data. In *Proceedings of International Workshop on Vision, Modeling and Visualization*, (2012), pp. 95-102
- [8] Meyer, M., Nelson, B., Kirby, R. M., Whitaker, R.: Particle Systems for Efficient and Accurate High-Order Finite Element Visualization. *IEEE Trans. Vis. Comput. Graphics* 13, 5 (2007), pp. 1015-1026
- [9] Mur, G.: Edge elements, their advantages and their disadvantages. *IEEE Trans. Magn.* 30, 5 (1994), pp. 3552-3557
- [10] Nedelec, J. C.: Mixed finite elements in R3. *Numerische Mathematik* 35, 3 (1980), pp. 315-341
- [11] Pagot, C., Osmari, D., Sadlo, F., Weiskopf, D., Ertl, T., Comba, J.: Efficient Parallel Vectors Feature Extraction from Higher-Order Data. *Computer Graphics Forum* 30, 3 (2011), pp. 751-760
- [12] Pagot, C., Vollrath, J., Sadlo, F., Weiskopf, D., Ertl, T., Comba, J. L. D.: Interactive Isocontouring of High-Order Surfaces. *Scientific Visualization: Interactions, Features, Metaphors, Dagstuhl Follow-Ups 2*, (2011), pp. 276-291
- [13] Remacle, J.-F., Chevaugeon, N., Marchandise, É., Geuzaine, C.: Efficient visualization of high-order finite elements. *International Journal for Numerical Methods in Engineering* 69, 4 (2007), pp. 750-771
- [14] Sanderson, A., Chen, G., Tricoche, X., Pugmire, D., Kruger, S., Breslau, J.: Analysis of Recurrent Patterns in Toroidal Magnetic Fields. *IEEE Trans. Vis. Comput. Graphics* 16, 6 (2010), pp. 1431-1440
- [15] Schroeder, W. J., Bertel, F., Malaterre, M., Thompson, D., Pebay, P. P., O'Bara, R., Tendulkar, S.: Methods and Framework for Visualizing Higher-Order Finite Elements. *IEEE Trans. Vis. Comput. Graphics* 12, 4 (2006), pp. 446-460
- [16] Songoro, H., Gjonaj, E., Weiland, T.: Computational modeling of water droplet deformation in strong electric fields. In *Proceedings of International Conference on Electromagnetics in Advanced Applications (ICEAA)*, (2012), pp. 333-336
- [17] Üffinger, M., Frey, S., Ertl, T.: Interactive High-Quality Visualization of Higher-Order Finite Elements. *Computer Graphics Forum* 29, 2 (2010), pp. 337-346
- [18] Webb, J.P.: Edge elements and what they can do for you. *IEEE Trans. Magn.* 29, 2 (1993), pp. 1460-1465
- [19] Webb, J.P.: Hierarchical vector basis functions of arbitrary order for triangular and tetrahedral finite elements. *IEEE Trans. Antennas Propag.* 47, 8 (1999), pp. 1244-1253
- [20] Whitney, H.: Geometric integration theory. *Dover Books on Mathematics Series*, Dover Publications, (2005)
- [21] Wiley, D. F., Childs, H., Harmann, B., Joy, K.: Ray Casting Curved-Quadratic Elements. In *Proceedings of the Sixth Joint Eurographics - IEEE TCVG conference on Visualization*, (2004), pp. 201-210.
- [22] Zhou, Y., Garland, M.: Interactive Point-Based Rendering of Higher-Order Tetrahedral Data. *IEEE Trans. Vis. Comput. Graphics* 12, 5 (2006), pp. 1229-1236
- [23] Advanced Finite Element Methods (ASEN 6367), Department of Aerospace Engineering Sciences, University of Colorado at Boulder (<http://www.colorado.edu/engineering/CAS/courses.d/AFEM.d/AFEM.Ch10.d/AFEM.Ch10.pdf>)
- [24] ParaView - Open Source Scientific Visualization. (<http://www.paraview.org/>)