Interactive Postprocessing in 3D Electromagnetics

Martin Hafner[1], Marc Schöning[1], Marcin Antczak[2], Andrzej Demenko[2], and Kay Hameyer[1]

[1]Institute of Electrical Machines, RWTH Aachen University, D-52056 Aachen, Germany
[2]Institute of Electrical Engineering and Electronics, Poznan University of Technology, 60-965 Poznan, Poland

Nowadays, the postprocessing and visualization of finite element solutions is performed by means of static a posteriori methods. This means that a dynamic interactive exploration of 3D solution data is only possible in a very limited way. In this paper, an interactive postprocessing approach is introduced, allowing for a dynamic interaction with finite element solutions, a simplified mesh cutting and data exploring as well as new ways of exploring complex solutions.

***Index Terms*—Electromagnetic data visualization, interactive postprocessing, scientific visualization, virtual reality, visualization.**

I. INTRODUCTION

THE exploration and interpretation of large amounts of solution data is an important part of an FE design process. Important decisions are made on basis of solution visualizations and further design steps are planned on basis of the ongoing understanding of the device under research. Therefore, effective postprocessing algorithms able of handling large amount of finite element data and the usability of such methods in an interactive way allow a faster design. Today, typical visualizations of finite element solutions are static colorized representations of a field distribution, which map a computed value to a specific color. Additionally, vector fields can be visualized as colorized cones or arrows, indicating the direction and magnitude of the field in every element. In this paper, further postprocessing methods and visualization techniques are introduced to enhance the exploration of finite element solutions. The aim of this work is to enable a bidirectional connection of the finite element package with the visualization framework to give users the ability to interact with a virtual reality (VR) scene. With this technique, the exploration of the visualized solutions can be performed intuitively, e.g., a free placing of cutting plane or spherical surfaces, a dynamic modification of displayed objects as well as a direct access to the FE data that represent those objects. For all mentioned aspects, examples are given to underline the capabilities of the proposed postprocessing methods.

All FE simulations reported in this paper have been done with the finite element package iMOOSE [1].

II. COMPUTER GRAPHICS SOFTWARE

Three-dimensional finite element analysis (FEA) leads to large amounts of solution data. In general, developers of electrical devices need to analyze the electromagnetic behavior in critical machine parts, such as teeth or teeth heads, or identify local magnetic hot spots. An intuitive method for the evaluation of such simulation data is the interactive exploration in virtual reality [2], which provides a direct visual impression of the field. This functionality helps the designer to identify quickly the points of interest, and allows performing further interactions and operations on the solution data. Therefore, in this paper a software methodology is presented to extend [3] by interactive postprocessing abilities.

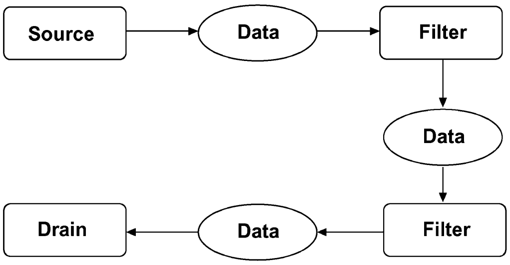


Fig. 1. Visualization pipeline of VTK.

The Visualization Toolkit VTK [4] has been used to visualize (static) 3D finite element solutions. This software package is an open source platform independent software library for 3D computer graphics, image processing and visualization including an interface layer for several interpreted languages, such as Java, Tcl/Tk, Python [5]. The object oriented design of this software is characterized by general, easy to use, data structures, whose versatility encourages a modular use of algorithms acting as filter objects. The working principle of VTK is based on visualization pipelines (see Fig. 1).

These visualization networks are constructed by connecting data objects, representing and providing access to data, and process or filter objects that operate on those data objects [6]. Each pipeline object has an internal state control to detect when a reexecute command is necessary and is also capable of scalable parallel processing [7].

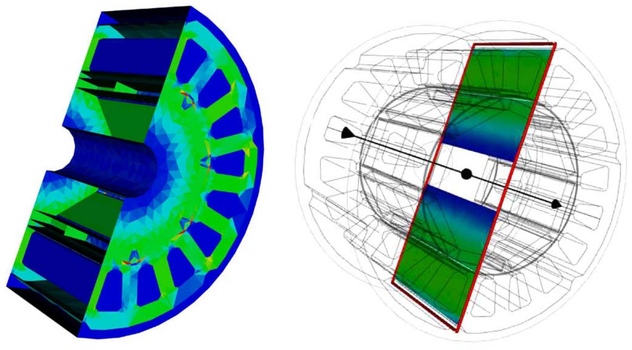
A. VTK Handling

VTK is capable to handle structured as well as unstructured grids—the latter, e.g., triangles or tetrahedrons, being commonly used for electromagnetic finite element analysis. Prior to the application of VTK for visualization purposes, the finite element computations must be converted in so-called unstructured grids, which represent an arbitrary combination of corresponding vtkCells, which is a synonym for finite elements. The shape of these cells is determined by its global coordinate nodes. The solution within each cell is represented either by a scalar or vectorial quantity per cell leading to a discontinuous visualization, or by such a value per node resulting in a smooth (linear) transition of the colorization. The recent publication

Manuscript received December 17, 2009; revised February 12, 2010; accepted February 16, 2010. Current version published July 21, 2010. Corresponding author: M. Hafner (e-mail: Martin.Hafner@IEM.RWTH-Aachen.de).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TMAG.2010.2043821



(a) (b)

Fig. 2. Two cutting types for the visualization of the flux density distribution. (a) Hollow cutting through a flux density distribution of a PMSM. (b) Interactive cutting of the flux density in a PMSM.

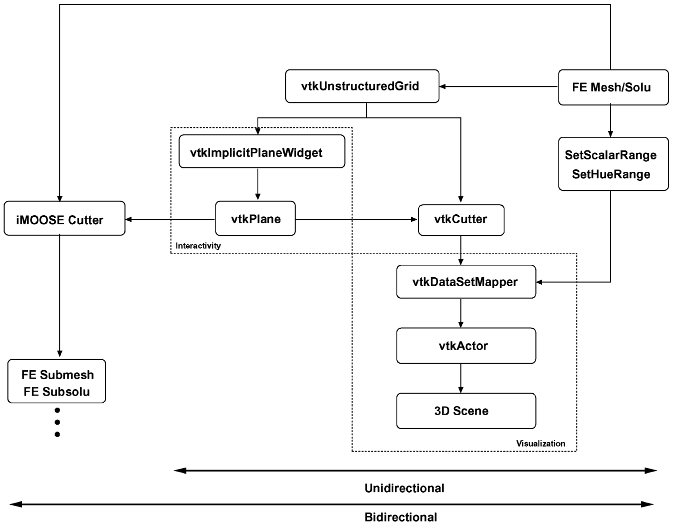


Fig. 3. VTK filter chain for interactive cutting.

[8] showed improvements to represent higher-order node-based shape functions and extensions for isoparametric elements.

The electromagnetic governing equations apply usually 1-form vector potential formulations, giving a continuity of the normal component of the corresponding field quantity. Therefore, all physical characteristics are approximately continuous inside a material domain and discontinuities only occur at elements belonging to different material domains. In consequence, the mesh is partitioned into several submeshes, one for each material domain, when converting the iMOOSE data into the VTK data. Boundary nodes are duplicated so that discontinuous solutions at finite element nodes of the original mesh translate into continuous solutions over the different submeshes. Zero-order solutions that are obtained by derivation, e.g., the curl operation for the magnetic flux density, can discontinuously be visualized by a cell-wise storage. All associations given here can be utilized to combine the data structure of VTK with FE formulations applying node-based shape function. A direct data conversion of edge-based solutions is not feasible at present but can be worked around by a nodal representation which is commonly used in the post-processing of such a type.

III. METHODOLOGY

A. *Interactive Cutting*

In order to accelerate visualization, the solution of 3D problems is often represented on the surface of the geometrical objects only, the solution in the bulk being completely disregarded. Making a plane cut through such a representation gives a hollow shell, as depicted in Fig. 2(a). This is option 1. A better insight into the bulk of the 3D FE solution is obtained by working with an interactive cutting plane, as depicted in Fig. 2(b). This is option 2. With this visualization tool, the complete 3D mesh data and the associated FE solution need to be available in the VTK data structure stored in memory for a fast processing.

The necessary filter procedure is shown in Fig. 3. First, the finite element mesh and solution are converted into a vtkUnstructuredGrid data set for each material. A vtkImplicitPlaneWidget, an interactively placeable infinite plane, is bounded to these grids. The mathematical representation of the cutting surface, in this case a vtkPlane, can be obtained from the 3D widget, so that the vtkCutter filter can generate a cutting mesh along that surface. The resulting grid is mapped to graphics primitives by the vtkDataSetMapper, which also maps the scalar range of the finite element solution to a given color range, specified by SetHueRange. The next element of the filter chain is the vtkActor representing an entity of the rendering scene. In particular, vtkActor combines object properties (color, shading type, etc.), geometric definition, and orientation in the world coordinate system. Since visualization, model interaction and cut-mesh generation are separated objects in the filter chain, cf. the pipeline principle in Section II, other widgets types, such as point-, line-, plane-, sphere- and spline-widgets, can be applied for further purposes.

The cutting plane of Fig. 2(b), can be moved, resized or rotated within the model boundaries and is computed in real time. The vtkCutter slices through the data set and creates a cutting surface by reducing the order of all intersecting cells. Fig. 2(b) exemplifies this cutting interaction on a permanent magnet synchronous machine (PMSM), with about 500,000 first order tetrahedron elements, to illustrate the flux density distribution inside.

Option 2 realizes a unidirectional communication from iMOOSE to VTK, that enables a data visualization on the mesh and submesh surfaces as well as an interactive data exploration inside the FE domain within the VTK software to individually spot the point of interest. In many typical electromagnetic problems, further postprocessing operations by the finite element package are in this situation required for a further evaluation, e.g., a leakage flux computation requiring FE integration. Therefore, a bidirectional connection between iMOOSE and VTK is required to transfer the interaction data back to the FE package and re-perform the requested action (a space order reducing element cutting). The resulting mesh data are also visualized by the described VTK filter chains and can be modified as well.

B. *Direct Model Interaction*

To improve the interactivity in 3D visualizations of finite element data, possibilities for a direct model interaction are re-

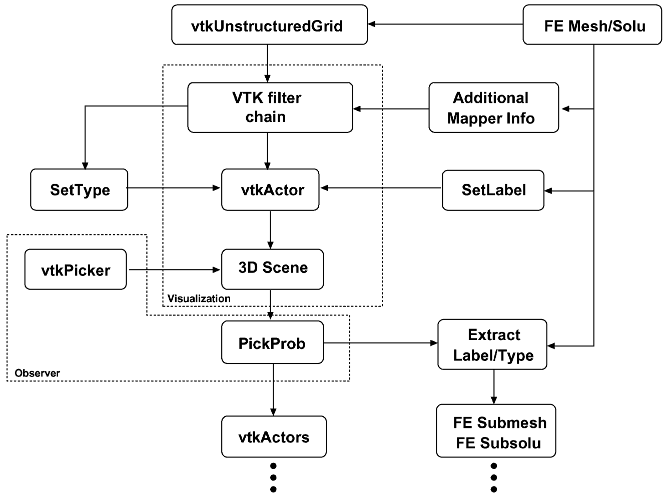


Fig. 4. VTK filter chain for direct model interaction.

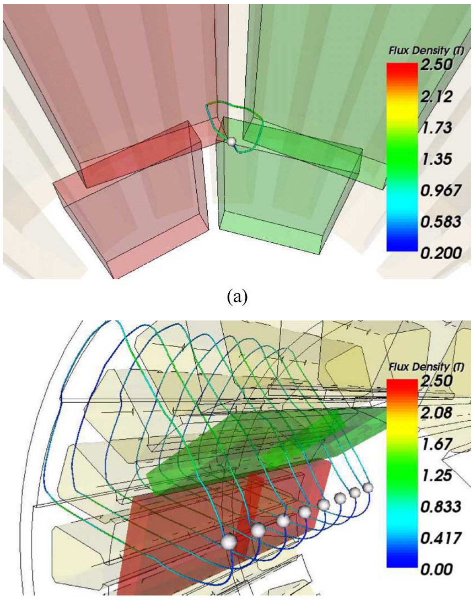
quired. The idea of a direct model interaction is to give users an intuitive direct access to the visualized solution data. By this, any kind of operation, e.g., mathematical integration or multiplication, can be performed on the input, so that the modification of the visualization can directly be observed. Since 3D visualizations are scalable on different display systems (from normal desktop PC up to virtual reality systems like cave style systems [3]), an intuitive model interaction, controllable by various 3D input devices that directly operate in the 3D scenes, is required. To fulfill the mentioned criteria, a software methodology is required, that analyzes the actual 3D scene to distinguish between different visualization types like a geometry, a mesh or a scalar or vectorial field solution plot. The methodology needs to return the corresponding original data sets from the FE meshes and solutions. The generalized VTK filter chain for direct model interaction is shown in Fig. 4. As mentioned before, the finite element mesh and solution are filtered by an arbitrary VTK filter chain and stored in a vtkActor placed in a 3D scene, cf. Section III-A. To identify the vtkActors in the further processing, each object gets additional information about the visualization type (SetType) and the submesh label identification (SetLabel). In the 3D scene, vtkPicker, controlled by a 3D input device, can be applied to grab vtkActor objects (PickProb). Type and label characteristics of the latter class objects can be used to extract the corresponding input data from the FE solution. These data sets are returned to the user interface. The same control pattern enables a direct access to the properties of single visualization objects.

Therewith, a combination of the interactive cutting geometries (cf. Section III-A) with the direct model interaction presented here is possible, to calculate the flux in various positions in an electrical machine for example.

IV. APPLICATION

A. *Enhanced Illustration Facilities*

The filter chain given in Fig. 3 is demonstrated on the interactive cutting exploration on 3D electromagnetic computations, see Fig. 2(b). To show its applicability for further purposes, this method is exemplified as interactive input device for the start point and start surface definition for the computation and visualization of closed flux lines [9]. The example chosen here is an internal mounted permanent magnet synchronous machine (IMPMSM) with rotor staggering, since in this rotor configuration stray fluxes with axial flux component are expected which are hard to locate in the solution data, and almost impractical to adequately visualized by standard methods in 3D.



(a)

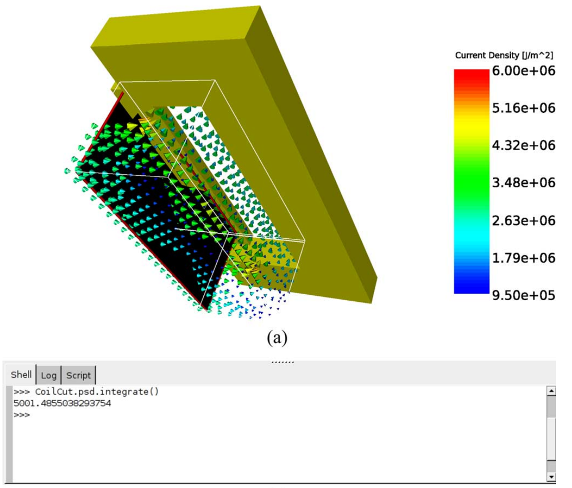
(b)

Fig. 5. Interactive streamline computation and visualization exemplified on a flux density distribution of an IMPMSM with and without rotor skewing. (a) Interactively found stray flux stream line between the skewed permanent magnets. (b) Seeding points along a line below a unskewed rotors pole.

For the visualization given in Fig. 5(a) the interactively placeable vtkPointWidget is used as seed point for the flux line computation. By this, the user has the possibility to explore the regions of interest and locate the corresponding coordinates quickly. In this example, a chosen starting point in the near of the permanent magnet edge, shows a stray flux line which closes through the other magnet angular to the z-axis. The same mechanism can be used to start a number of streams from a placed line, as exemplified in Fig. 5(b) for the same machine configuration without rotor skewing.

B. *Interactive Postprocessing*

The main motivation and effort of this work is to combine the classical methods of electromagnetic FE postprocessing with interactive capabilities which are suitable for virtual reality techniques. To demonstrate the general idea behind the theoretical concept, drawn in Sections III-A and III-B, it is applied to a quarter model of the TEAM 20 workshop problem [10]. Its quasi-static field distribution is evoked by a current injected on the cross section planes of the coil. The field is computed in two steps. Firstly, the current density in all conducting regions is calculated, and secondly the vector potential and the corresponding induction field. Applying the methods given in Figs. 3 and 4, an



(a)

```
>>> CoilCut.psd.integrate()
5001.4855038293754
>>>
```

(b)

Fig. 6. Model interaction to evaluate the coil current by an interactive integration, exemplified on the TEAM 20 test case. (a) Visualization of model interaction to obtain the coil current by the current density distribution. (b) Corresponding console output of integral current computation.

interaction to compute the current from the element-wise current density field distribution could for instance be archieved as follows:

- Interactive placement of a visualized cutting plane [dark rectangular region in Fig. 6(a)].
- A first representation of the field on the plane can be generated from the graphical information contained in VTK data structures [preview, as e.g., shown in Fig. 2(b)].
- For further FE processing, if required, the cutting is repeated through the iMOOSE mesh this time, and the FE solution on the plane is generated. A new graphical object is created and displayed in the VR environment.
- This object, can now be selected in the VR environment (by clicking, grabbing, . . .) in order to be attributed a handle or a symbolic name for further processing.

The last operation can optionally be monitored in an interactive Python console, by manipulating the scalar quantity (PSD) of the symbolic link to the VR slice object (CoilCut), see Fig. 6(b).

This simple procedure can be extended to arbitrary postprocessing operation.

V. CONCLUSION

Efficient methods for the visualization of finite element solutions are essential for the evaluation of electromagnetic devices under research and development. In this paper, an interactive postprocessing technique is introduced that extends the static process to provide dynamic modifications within the visualization and an intuitive 3D data exploration. Generalized techniques for this postprocessing approach are proposed and described by means of visualization patterns for interactive cutting and direct model interaction.

Both techniques are exemplified on realistic 3D electromagnetic problems to demonstrate the benefit of an enhancement of classical finite element postprocessing by an interactive visualization framework. This enables an exploration of finite element solution, which is extended by the ability to interact intuitively on the visual impression. An expendable combination with enhanced illustration facilities for the visualization of stream lines has been presented to locate the designers point of interest. The interactive evaluation procedure has been demonstrated on a standard test case, to show the advantages of the proposed framework.

The best virtual reality systems are CAVE-systems, which create a nearly prefect immersion effect of 360° by projecting the visualization on the inner surface of a hollow cube. In this environment normal GUI windows would disturb the immersion, so that users need an additional portable screen for the interface controlling, e.g., a tablet PC, which requires an integration of a virtual reality library in the presented framework, e.g., VISTA [11]. Since many FE simulations are transient to evaluate time characteristics, transient interactive visualizations could assist to observe occurring phenomena [12]. The implementation of FEA restarting facilities would make it possible to perform simulations as a further reaction within the VR scene.

REFERENCES

[1] D. van Riesen, C. Monzel, C. Kaehler, C. Schlensok, and G. Henneberger, "iMOOSE-an open-source environment for finite-element calculations," *IEEE Trans. Magn.*, vol. 40, no. 2, pp. 1390–1393, 2004.

[2] W. R. Sherman and A. B. Craig*, Understanding Virtual Reality: Interface, Application, and Design*, 1st ed. New York: Morgan Kaufmann, 2003.

[3] M. Schöning and K. Hameyer, "Applying virtual reality techniques to finite element solutions," *IEEE Trans. Magn.*, vol. 44, no. 6, pp. 1422–1425, 2008.

[4] I. Kitware*, VTK User's Guide Version 5*, 5th ed. Clifton Park, NY: Kitware, Inc., Sep. 2006.

[5] G. van Rossum, Python Programming Language 2009 [Online]. Available: http://www.python.org

[6] W. Schroeder, L. Avila, and W. Hoffman, "Visualizing with VTK: A tutorial," *IEEE Comput. Graph. Applicat.*, vol. 20, no. 5, pp. 20–27, 2000.

[7] J. Ahrens, C. Law, W. Schroeder, K. Martin, K. Inc, and M. Papka, "A Parallel approach for efficiently visualizing extremely large, time-varying datasets," 2000 [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.35.8065

[8] W. Schroeder, F. Bertel, M. Malaterre, D. Thompson, P. Pebay, R. O'Bara, and S. Tendulkar, "Methods and framework for visualizing higher-order finite elements," *IEEE Trans. Visual. Comput. Graph.*, vol. 12, no. 4, pp. 446–460, 2006.

[9] M. Hafner, M. Schöning, M. Antczak, A. Demenko, and K. Hameyer, "Methods for computation and visualization of magnetic flux lines in 3D," presented at the 17th Conf. on the Computation of Electromagnetic Fields, Florianopolis, Brazil, 2009.

[10] International Compumag Society (ICS)—TEAM Problems [Online]. Available: http://www.compumag.co.uk/team.html

[11] T. van Reimersdahl, T. Kuhlen, A. Gerndt, J. Henrichs, and C. Bischof, "ViSTA: A multimodal, platform-independent VR-toolkit based on WTK, VTK, and MPI," presented at the 4th Int. Immersive Projection Technology Workshop (IPT2000), Ames, IA, 2000.

[12] J. Biddiscombe, B. Geveci, K. Martin, K. Moreland, and D. Thompson, "Time dependent processing in a parallel pipeline architecture," *IEEE Trans. Visual. Comput. Graph.*, vol. 13, no. 6, pp. 1376–1383, 2007.