



SIMPRO

## **Multi-physics simulations for electrical machine development**

Authors: Janne Keränen, Aino Manninen, Jenni Pippuri

Confidentiality: Public



<b>Report's title</b>		
Multi-physics simulations for electrical machine development		
<b>Customer, contact person, address</b>		<b>Order reference</b>
Tekes, Matti Säynätjoki Kyllikinportti 2, P.O. Box 69, FI-00101 Helsinki, FINLAND		Tekes: 40204/12
<b>Project name</b>		<b>Project number/Short name</b>
Computational methods in mechanical engineering product development		100553/SIMPRO
<b>Authors</b>		<b>Pages</b>
Janne Keränen, Aino Manninen, Jenni Pippuri		18
<b>Keywords</b>		<b>Report identification code</b>
Electrical machines, FEM, open-source, Elmer, parallelisation, computation, simulation		VTT-R-04618-15
<b>Summary</b>		
<p>In electrical machine R&amp;D, like in other fields, different computation methods are needed for different purposes. For detailed analysis and product design, a multi-physical finite element (FEM) analysis is required. However, if advanced multi-objective optimisation of electro-mechanical devices is pursued, the computational burden of FEM analysis is too high, and more simple analytical models are needed. In this study, both alternatives are discussed.</p> <p>One hot topic in FEM computations has lately been the utilisation of open-source FEM tools. The benefits of open source can be found e.g. in massive parallel computation, more vast scripting possibilities, and possibility to tailor the software for special usage. Here, we study open-source FEM tools for three-dimensional (3D) electromagnetic and multi-physical simulation of electrical machines. The two most promising tools are Elmer and GetDP. We present two electrical machine test cases with Elmer. Elmer and GetDP are presently only open-source tools, which can solve 3D electromagnetic problems. However, they lack features needed for rotating electrical machine computations, e.g. they have limitations with the rotation model. The limitations are more severe in GetDP. The parallel performance of Elmer exceeds the results of GetDP, as GetDP utilises parallel computation only in linear solvers, whereas Elmer also in linear system assembly. Furthermore, in the test cases, Elmer shows a good parallel performance with rotating electrical machine models.</p> <p>Additionally, optimisation of electrical machines was studied. An analytical model implemented in MATLAB was used to represent an electrical machine in the optimisation process. Both single and multi-objective algorithms were tested in DAKOTA optimisation tool.</p>		
<b>Confidentiality</b>	Public	
Tampere 14.10.2015		
<b>Written by</b>	<b>Reviewed by</b>	<b>Accepted by</b>
Janne Keränen, Research Scientist	Pavel Ponomarev, Research Scientist	Johannes Hyrynen, Head of Research Area
<b>VTT's contact address</b>		
VTT Technical Research Centre of Finland, P.O. Box 1000, FI-02044 VTT, FINLAND		
<b>Distribution (customer and VTT)</b>		
Tekes/Matti Säynätjoki, 1 copy VTT/archive, 1 copy		
<p><i>The use of the name of the VTT Technical Research Centre of Finland (VTT) in advertising or publication in part of this report is only permissible with written authorisation from the VTT Technical Research Centre of Finland.</i></p>		

## Contents

---

1	Pre-study of multi-physics simulation for electrical machine development.....	3
1.1	Introduction: Modelling of electrical machines.....	3
1.1.1	Numerical modelling of electrical machines.....	3
1.1.2	Analytical model for electrical machines.....	4
1.2	Open-source FEM tools for electrical machine computation .....	5
1.2.1	GetDP .....	6
1.2.2	Elmer .....	7
1.2.3	Other tools .....	8
1.3	Electrical machine optimisation.....	8
2	Experiences with Elmer .....	9
2.1	Elmer usage .....	9
2.2	Elmer case study models.....	10
2.3	Results .....	12
3	Electrical machine optimisation with DAKOTA .....	14
3.1	DAKOTA.....	14
3.2	Results and discussion .....	14
4	Acknowledgements.....	15
	References.....	16

# 1 Pre-study of multi-physics simulation for electrical machine development

---

## 1.1 Introduction: Modelling of electrical machines

Design of electrical machines requires a good command of multi-physical phenomena including electromagnetism, mechanics, thermodynamics, structural dynamics, vibration, noise, fluid dynamics, and fatigue. The functioning of the machines rises from electromagnetic phenomena connected with mechanical rotation and connected electrical circuits. Here, we concentrate on researching the multi-physical simulations combining the latter three fields of physics.

As electrical machines design is inheritably multi-physical, its accurate three-dimensional (3D) modelling has been out-of-question until recently. The computational burden has been too heavy for traditional computation tools and hardware. Another reason is that 3D magnetic eddy-current problems require quite nonstandard modelling structures of numerical mathematics compared to e.g. structural analysis, where Finite Element Method (FEM) is originated from. Other comparative traditional methods, like Finite Volume Method, are not directly applicable either. Hence, traditional methods for electrical machines research and design have been analytical and two-dimensional (2D) FEM approaches. Relatively simple analytical or semi-analytical approaches may be applied for the standard motor topologies, especially in the initial electromagnetic designing or optimisation. Further and more detailed investigations are founded on a numerical discretised solution of the electromagnetic field. However, there is still a need for both analytical and numerical FEM solutions.

### 1.1.1 Numerical modelling of electrical machines

Numerical 2D electromagnetic computation is a standard tool in design and analysis of electromechanical apparatuses. 2D can be used, if only a certain cross section of the machine needs to be modelled to completely represent the machine's behaviour. 2D modelling is most commonly used for radial flux machines, i.e. machines where the magnetic flux flows radially in the air gap with respect to the shaft of the machine (the shaft-directional component of the magnetic flux density can be assumed to be zero) and the electrical current density is assumed to be in the direction of the shaft. The 3D features are neglected or modelled separately. For example, the end effects of radial flux machines are commonly dealt with additional (circuit or analytical) models to 2D FEM.

Let us clarify the differences of the 2D and 3D cases. Generally, both in 2D and 3D, magnetic flux density inside the machine can be found from following partial differential equation, along with proper boundary and initial conditions,

$$\text{curl} \frac{1}{\mu} \text{curl} A = -\sigma \left( \frac{\partial}{\partial t} A + \text{grad} V \right) + J_s, \quad (1.1)$$

where  $A$  is the magnetic vector potential,  $V$  the electrical scalar potential,  $J_s$  the known source current,  $\mu$  the permeability of the material, and  $\sigma$  the electrical conductivity. The above is so-called A-V formulation for eddy current problems.  $V$  is nonzero only in conducting regions, whereas  $A$  exists in the whole domain. After solving (1.1), the magnetic flux density and electrical field strength can be solved from  $B = \text{curl} A$  and  $E = -\partial A / \partial t - \text{grad} V$ .

In 2D, the unknown magnetic vector potentials have only one component (normal to the 2D plane) and the potential can be modelled with standard nodal FEM functions, whereas in 3D the so-called edge element basis functions are required. The change from nodal elements to

edge element causes very heavy changes in simulation code, above the natural dimension changes. For example, the numerics of the curl-curl operator in (1.1) are totally different from those of div-grad in the 2D-version of (1.1). To be precise, the stiffness matrix from curl-curl is singular, if no special treatment (gauging) is done to squeeze a non-singular submatrix.

Further, integral of  $\partial A/\partial t$  along the conductors behaves as a source term for the electrical circuits connected to the machine. In reverse direction, the circuits are coupled to the electromagnetic field problem via the electric scalar potential  $V$ . In 2D, the conductors need to be perpendicular to the 2D plane (i.e.  $z$ -directional). Then,  $V$  is constant in the 2D cross section of each conductor and constantly varies in the  $z$ -direction. This makes circuit connection in 2D quite simple; connection is established directly via variables  $dV/dz$  and  $A_z$ . In 3D,  $V$  does not change linearly along  $z$ -axis, but its dependence on any spatial dimension is free inside a conductor. If connected with electronic circuit equations,  $V$ 's values in so-called connector areas are connected with external circuit equations.

Due to these differences between 2D and 3D models—along with natural size difference of the 2D and 3D discretised models—2D models have had tolerable computational times already from 1980s, but 3D problems still need days to weeks to be solved with commercial software. Especially the computation is intolerably slow, when the machine's core materials are assumed to be conducting, i.e. eddy currents in the core are taken into account in the problem formulation. Parallel computation can be seen as a solution for this, but none of the present commercial electrical machine FEM modelling software tools effectively take maximum advantage from parallelism. Further, due to this multi-physical nature of the computation and the unique features of 3D eddy current computation (i.e. edge elements and rotor mesh rotation), this can be seen as niche market for commercial FEM software vendors. Thus, here we concentrate on researching the status of open-source FEM tools for multi-physical electrical machine simulations.

Availability of electromagnetic computational software is good, and both commercial and open-source providers are well presented. Commercial providers include, for instance, Ansys Maxwell [1], Cedrat Flux [2], COMSOL Multi-physics [3], JSOL JMAG [4], and Cobham (formerly Vector Fields) Opera [5]. Perhaps the most well-known open-source software for FEM numerical modelling are GetDP [6] and Elmer [7]. Though the number of different numerical electromagnetic codes is rather large, only few of them are capable of coping with all the special features that are associated with rotating electrical machines. This is particularly true when a two-dimensional (2D) approach is not sufficient and a three-dimensional (3D) approach must be taken instead.

### 1.1.2 Analytical model for electrical machines

As a part of the optimisation procedure, an analytical dimensioning tool was used for modelling a permanent magnet synchronous generator. The dimensioning tool is based on the analytical equations commonly used in the machine design [8,9], and has been developed inside VTT during previous projects using MATLAB. For this particular task, the dimensioning tool was modified so that it could be used for both single- and multi-objective optimisation.

The flowchart of the analytical design process is presented in Figure 1.

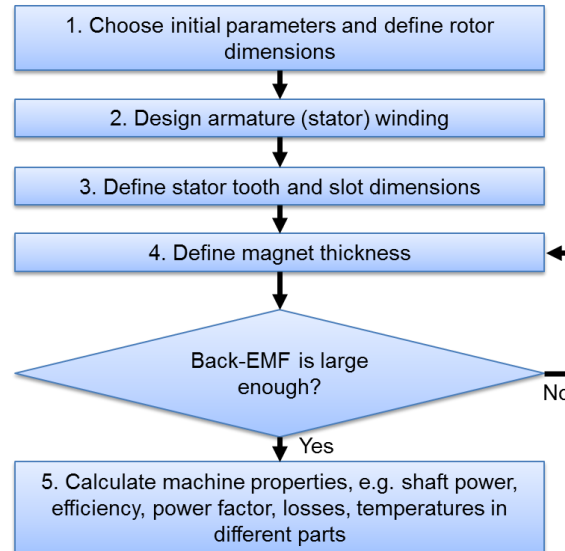


Figure 1. Flow chart of the analytical PMSM design tool.

The computation sequence is the following: First, the initial parameters such as power, rotation speed, magnetic flux density and current density, are chosen, and the basic mechanical dimensions, such as the air gap diameter and machine length, are defined.

Then a suitable stator winding is designed. Currently, the dimensioning tool works for single layer integer slot windings. The number of stator slots and number of conductors in a slot are defined so that the winding produces a suitable linear current density. The dimensions of the stator teeth and slots are defined so that the stator winding can be fitted into the slots. The yoke size is defined in the initial parameters.

For surface mounted magnets, these parameters include the width and thickness of the magnets. First the magnet width is chosen depending on the pole pitch. After that, the thickness is defined iteratively in such a way that the magnets produce enough flux in the air gap. After defining the size of the magnets, it is checked that the back induced electromotive force (back-EMF) is large enough. For a generator, it should be about  $1.05 \times$  phase voltage.

After the dimensioning, the resistance, inductances, shaft power, losses, efficiency and power factor are calculated. Also the heat transfer in the machine is estimated to ensure that especially the permanent magnets are not overheated.

## 1.2 Open-source FEM tools for electrical machine computation

As described in section 1.1.1, 2D electrical machine FEM is in formulation perspective far simpler than in 3D. Also model creation and handling is far more complicated in 3D compared to 2D, i.e. 3D FEM often requires a CAD engine for the geometry. From these reasons, there is ten-fold more open-source or in-house 2D finite element codes for electrical machines compared with 3D alternatives.

In Finland, there is long history of using non-commercial 2D FEM tools both in industry and universities. Antero Arkkio developed 2D tool called FCSMEK in 1980s in Aalto University [10]. FCSMEK have since being used in dozens of doctoral theses in Aalto, and several companies have adopted it. Especially ABB needs to be mentioned, as they have developed and integrated FCSMEK for their standard electrical machine development tool. ABB has own FCSMEK version where they have implemented many features non-existent in commercial alternatives. FCSMEK have been integrated in their own in-house R&D modelling environ-



ment ADEPT. Designer can alternatively use analytical formulas or FCSMEK for computing the machine parameters and characteristics.

There are dozens of other 2D tools (e.g. FEMM, iMOOSE, and FEMAG), in different level and different licencing and ownership. However, here we concentrate more on those tools that have also 3D features and that are released in open-source licencing. Among them, Elmer and GetDP are discussed in more detail, as they are the two most used open-source FEM tools for 3D electrical machine simulations.

### 1.2.1 GetDP

GetDP (*General environment for the treatment of Discrete Problems*) is a software environment for the numerical solution of integro-differential equations, allowing multi-physical problems (e.g. electromagnetic, thermal) utilising different numerical methods (e.g. FEM, integral methods). The problems can be one to three-dimensional and they can be static, transient or harmonic. [6]

The main feature differentiating GetDP from other tools is that the formulation of a discrete problem is given by user in ASCII data files with syntax close to the symbolic mathematical expressions of this problem. GetDP thus gives user advanced developing tools and a large freedom in adding new functionalities only by writing a new ASCII file, without changing the source code. Its aim is to be welcoming and of easy use for both development and application levels: it consists of a working environment in which the definition of any problem makes use of a limited number of objects, which makes the environment structured and concise. [11]

GetDP is usually used through the Gmsh user interface. Gmsh is an open-source graphical pre- and post-processor for 2D and 3D FEM, including a build-in CAD engine and a mesh generator. Gmsh has four modules: geometry, mesh, solver and post-processing, where the solver is by default GetDP. The input to these modules can be done either interactively using the graphical user interface or with ASCII script files using Gmsh's own syntax. The focus is on fast parametrisable meshing tool and on visualisation capabilities. Gmsh uses a boundary representation to describe geometries, i.e. the model is created by successively defining points, lines, surfaces, and volumes. This is a quick and parametrisable way to create simple or repetitive geometries, but can become inconvenient for large models. For complex models Gmsh can be linked with an external CAD kernel and import native files directly. For example, the free CAD kernel Open CASCADE enables native BREP, STEP and IGES import. When linked with appropriate CAD kernel, Gmsh can generate simplicial finite element meshes for CAD models in their native format. [12]

Both GetDP and Gmsh are originated from University of Liège, Belgium. GetDP was created by professors Patrick Dular and Christophe Geuzaine, whereas Gmsh was created by Christophe Geuzaine and Jean-François Remacle from Université catholique de Louvain, Belgium. Both software tools are under GNU General Public License (GPL).

As described above, the electrical machine models need a model for rotor rotation and for the connected electrical circuits. There are two main types of different models allowing the rotor mesh to rotate relative to the stator mesh: sliding surface models and moving band methods. In sliding surface, the two meshes are allowed to move (i.e. slide) relative to each other, which produces a surface where two meshes (of rotor and stator) can be non-conformal or discontinuous, and different methods are used to interpolate the data from one mesh to another. GetDP uses moving band method, where a band of mesh exists in the air-gap between stator and rotor meshes which provides conformity. The mesh in the moving band is allowed to distort slightly during the rotation. When the distortion is too high, the band is re-meshed, to maintain the conformity and quality of the mesh, and new distortion iteration is started. In GetDP the method works in 2D, but currently the efficient 3D moving band is limited to



structured meshes of the air-gap and to a fixed step that depends on the discretisation and the rotation speed. For general meshes GetDP uses a global re-meshing, which is quite slow procedure. Circuits can be described as constraints to global quantities in the input *pro*-file for GetDP.

GetDP can utilise parallel linear solvers, if GetDP is compiled with PETSc. Parallel iterative solvers are said by the developers to scale quite well (with up to 128 cores with ILU(0) preconditioner and GMRES iterative solver). Further, parallel direct solvers are available, too; e.g. typically MUMPS is used with 16 cores. However, even the GetDP developers admit that the GetDP problem definition makes the computation slower. Hence, they are developing new solver inside Gmsh.

### 1.2.2 Elmer

Elmer is an open-source FEM based software package for the solution of partial differential equations, mainly developed by CSC and intended for multi-physics simulations [7]. Elmer development was started in 1995 in collaboration with Finnish Universities, research institutes and industry. It was published as open source in 2005 under GPL license (the ElmerSolver library was further freed in 2012 to use LGPL licensing), which have made the use and development of Elmer more international. Presently, the user base of Elmer is in thousands, with electromagnetism is one of the major applications. Elmer attracts 20000 downloads per year.

Elmer can deal with a great number of different equations, which may be coupled in a generic manner making Elmer a versatile tool for multi-physical simulations. Elmer includes physical models of e.g. fluid dynamics, structural mechanics, electromagnetics, heat transfer and acoustics. In some research intensive fields, Elmer is the globally leading code, e.g. glaciology 3D ice flow simulation. Moreover, Elmer's strengths are in multi-physical problems and in massive parallelisation. In parallelisation, Elmer mainly trusts in on the message passing approach (the MPI standard), a method most suitable for distributor-memory parallelisation. The model mesh is divided into so-called partitions and computations associated with the partitions are performed in parallel by many computational cores simultaneously. In outline, the linear system assembly, the matrix-vector products of the iterative linear system solvers and preconditioning are performed in parallel for each partition. Along with the own partition methods, also Metis library can be used to partition the mesh. Further, Elmer can utilise external libraries for the solution of linear systems, like Hypre [13], Trilinos [14], and MUMPS [15]. In recent years, also multithreading (for shared memory parallelisation, e.g. Intel Xeon Phi type solutions, with OpenMP) of the solvers have been in the works. For some solvers, the parallel performance may scale up to thousands of cores.

The main developer in CSC has been Juha Ruokolainen from the beginning of the development. Recently, Dr. Peter Råback has become the main responsible of the Elmer development and the Elmer team manager. The Elmer team have four full-time workers, and CSC further has several additional Elmer specialists. Within the present decade, there has been effort to develop Elmer towards electromagnetics problems; for example edge elements, indispensable for electrodynamic simulations, were implemented few years ago. The first solver to use edge elements, WhitneyAVSolver, was implemented by Juha Ruokolainen in 2010 with the help of Saku Suuriniemi from Tampere University of Technology. ABB have been the driving force of Elmer development in Electrical machine computations. ABB have already taken Elmer as a tool for R&D of the electrical machines. ABB has long used non-commercial 2D FEM package FCSMEK, and is taking Elmer as a similar solution for 3D simulations. Presently, there are several development projects for Elmer in electrical machine multi-physics perspective, along with vast and growing domestic usage of Elmer in Finnish electromechanical industry.

Since 2013, an all-time developing electrical circuit model has existed in Elmer. Elmer's rotation air-gap model is of sliding air-gap type, based on the so-called mortar method. The method is more suitable for 3D compared to moving band method used in GetDP.

One of the efforts towards accelerating electromagnetics development of Elmer has been the formation of Finnish Elmer electromagnetics and electrical machine community (presently called Elmer/EM) in autumn 2012. Representatives from CSC, ABB Finland, Trafotek, Kone, Konecranes, Sulzer Pumps, Ingersoll-Rand Finland, Skanveir, VISEDO, VTT, Aalto University, Tampere University of Technology, Lappeenranta University of Technology, and Magnet Technology Centre have participated at the community meetings and workshops, and many have joined SEMTEC Tekes project for further development. The role of the community is the change of information on Elmer development, prioritising the development steps, and preparation of projects for Elmer development and usage. VTT have been the coordinator of Elmer/EM from the beginning.

### 1.2.3 Other tools

iMOOSE is or was a project of Department of Electrical Machines of Aachen University (RWTH), Germany, focusing on electromagnetic problems arising from the design of electrical machines [16]. It was released as open source in 2002, but last published version dates back to year 2004. It had static and time-transient solvers for 2D and static and time-harmonic for 3D. iMOOSE had also iMOOSE.trinity tool, a 3D visualisation and post-processing tool. There have been in-house development of iMOOSE also later in RWTH, but no updates to the open-source version have been published.

FEMM (Finite Element Method Magnetics) is a Windows-based finite element solver for 2D magneto- and electrostatic problems, along with heat and current flow problems, with graphical pre- and post-processors. FEMM is licensed under the terms of the Aladdin Free Public License. Its electrical machine modelling is very limited due to lack of a time-dependent solver and a rotation model. [17]

FEMAG is a software tool for the calculation of 2D static or quasi-stationary electromagnetic fields, specialised to electrical machine problems. FEMAG has been developed since 1982 at ETH Zürich. The licensing of FEMAG is unclear. There are projects to develop a 3D counterpart for FEMAG, possibly utilising GetDP and Gmsh. [18,19]

## 1.3 Electrical machine optimisation

The optimisation of an electrical machine is typically a multi-objective optimisation task with several design variables and also constraints. In this optimisation study, a permanent magnet synchronous generator was chosen as an example. In this case, there are totally 14 design variables, 2 are discrete variables and the other 12 are continuous. The variables are shown in Table 1. The objective functions are output power, torque density, mass, efficiency, power factor and cost. There are also 3 constraints that have to be satisfied. The electrical machine model that is used for the optimisation is presented earlier in section 1.1.2. Chapter 3.1 explains the optimisation of a permanent magnet synchronous machine using DAKOTA. In this task, we have co-operated with Aalto University, where Nature Inspired Computing (NIC) methods have been used for optimisation, and with University of Jyväskylä, where Interactive Multi-objective optimisation has been done. However, because this report only covers the results from the VTT subproject, we will only discuss the results obtained from DAKOTA here.

Table 1. Design variables.

Parameter number	Description	Type	Min	Max
1	Pole pair number	integer	20	80
2	Linear current density (A/m)	real	35000	65000
3	Current density (A/m <sup>2</sup> )	real	2000000	6000000
4	Air-gap (m)	real	0.001	0.05
5	Rotor outer diameter/stack length	real	0.8	5
6	Relative magnet width	real	0.6	0.95
7	Tangential stress	real	21000	48000
8	Rotor yoke flux density (T)	real	1.3	1.6
9	Number of slots per pole per phase	integer	1	3
10	Stator outer diameter, relative	real	0.8	0.99
11	Slot opening, relative	real	0.25	0.75
12	Slot width, relative	real	0.75	0.95
13	Slot height 1, relative	real	0.01	0.1
14	Slot height 2, relative	real	0.01	0.1

## 2 Experiences with Elmer

### 2.1 Elmer usage

Elmer is a versatile FEM software tool for multi-physical computation. However, the traditional usability by graphical user interface is not even close to the level of commercial FEM counterparts. The feasibility comes from different possible modelling toolchains and scripting possibilities. Elmer includes graphical user interface ElmerGUI and post-processor ElmerPost, but the usage is not limited to them. Further, there are no own geometry and mesh creation tools, hence the mesh needs to come outside. Elmer can read several different model geometry and mesh file types, or external mesh generators (Tetgen [20] or Netgen [21]) can be used through the ElmerGUI. However, most often the mesh is made with external tools and imported to Elmer. The tool ElmerGrid can be used to convert differing mesh formats into Elmer-suitable meshes. Presently, officially ElmerGrid can read the following external mesh formats<sup>1</sup>:

- \*.ansys: ANSYS input format created by ANSYS pre-processor using a special ANSYS macro
- \*.inp: Abaqus input format created by Ideas pre-processor for Abaqus
- \*.fil: Abaqus output format created by the solver of Abaqus

<sup>1</sup> Unofficially, due to certain license or IPR restrictions, some other input formats are available, but not described publicly.

- \*.FDNEUT: Fidap neutral file created by Gambit, the pre-processor of Fluent
- \*.unv: Universal mesh file format, however only a subset of unv-format can be read
- \*.mphtxt: Comsol Multi-physics mesh format
- \*.dat: Fieldview format, used with Gridgen pre-processor
- \*.node,\*.ele: Triangle 2D mesh format created by Triangle 2D Delaunay mesh generator
- \*.mesh: Mesh format of Medit mesh generation program
- \*.msh: Mesh format of GID mesh generation program
- \*.msh: Mesh format of Gmsh mesh generation program

Hence, above-mentioned Gmsh can be used as a pre-processor for Elmer, and this workflow is quite popular, especially in research sector. Further, ElmerGUI uses Open CASCADE (OCC) library [22] for CAD import features of several CAD formats like IGES, STEP, and BREP. Open CASCADE is an open-source CAD kernel and development platform for 3D modelling applications. Another very popular pre-processor for Elmer is SALOME [23,24]. ElmerGrid can read the unv file format and some other mesh formats written by SALOME. Our SALOME tests have shown that SALOME is very versatile tool for pre-processing; the biggest problem is its instability on certain procedures. However, tailored toolchains for creating the models can be made, too. For example, ABB has integrated Elmer as a part of their modelling environment ADEPT. Presently, they are generating 2D geometry by few parameters; secondly the 2D geometry is meshed, and finally the 2D mesh is extruded to a 3-dimensional mesh by ElmerGrid.

For post-processing, the ElmerPost is usable, but already quite old-fashioned. For internal alternative, ElmerGUI has also an internal postprocessor, based on the Visualization Toolkit (VTK) [25]. However, the most usable alternative is ParaView [26]. ParaView is an open-source, multi-platform data analysis and visualisation application, developed to analyse extremely large datasets using parallel computing. Elmer's ResultOutputSolver can write ParaView's VTU format. All the visualisation in this report and publication [27] are done with ParaView. Further, Elmer can also write Gmsh post-processing format.

## 2.2 Elmer case study models

Elmer has been studied with two typical 3D electrical machine models, i.e. two case studies. These tests were performed during ParallaX and SIMPRO Tekes projects, both networked with companies. In the beginning of testing (early 2013), only 3D models without rotation were available in Elmer. Hence, a model for inductance and force computation of radial flux machine's end-windings was used for initial tests. Later, spring 2014, a working 3D rotation model for electrical machines in Elmer was developed. After that, a rotating induction motor model was tested. Both electrical motor model geometries are made by Mikko Lyly from ABB. The induction motor model is generated by the above-mentioned ABB workflow.

The end-winding model of Figure 2 is representing a situation, where two-phase short-circuit fault occurs. End-windings are the part of radial flux electrical motor windings that are outside motor's core, in this case the stator core. The end-windings connect the conductors located in different stator core slots and thus they need to be quite long and curved. The true interest is in the displacements caused for the end-windings by the magnetic forces and in the inductance estimation of this part of machine. First, Elmer's WhitneyAVSolver is used to solve the magnetic vector potential  $A$  and electric scalar potential  $V$  inside the windings. Second, with the solved electromagnetic fields, the forces caused by them to the windings are solved;

the forces are used as the source for the elasticity solver to simulate the deformations of the windings. All the electromagnetic material properties in the model are linear. The WhitneyAVSolver part of the model's solving procedure represents one linear step of Newton-Raphson iteration of one time-step of transient non-linear electrical machine case. Thus, this problem could be seen as a one basic problem solved quite many times within one typical electrical machine computation. Further, the model avoids complication by not having connected electrical circuit models, or a rotation model. Hence, the WhitneyAVSolver part of the end-winding model represents pure electromagnetic problem included in every electrical machine magnetic model.

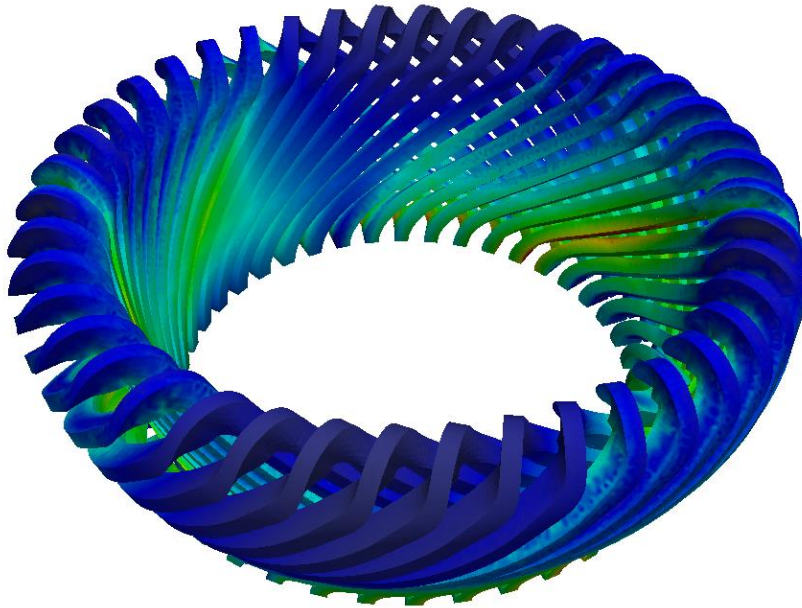


Figure 2. End-winding model, magnitude of magnetic flux density ( $B$ ) shown on the surface of the end-windings. Only the end-windings are included in the model. Figure made with ParaView [26].

The induction motor model of Figure 3 is a three-phase four-pole squirrel-cage induction motor with 48 stator slots and 58 rotor slots. Only half of the motor was modelled taking advantage of the rotational magnetic symmetry of the motor design, which helps to reduce the model size.



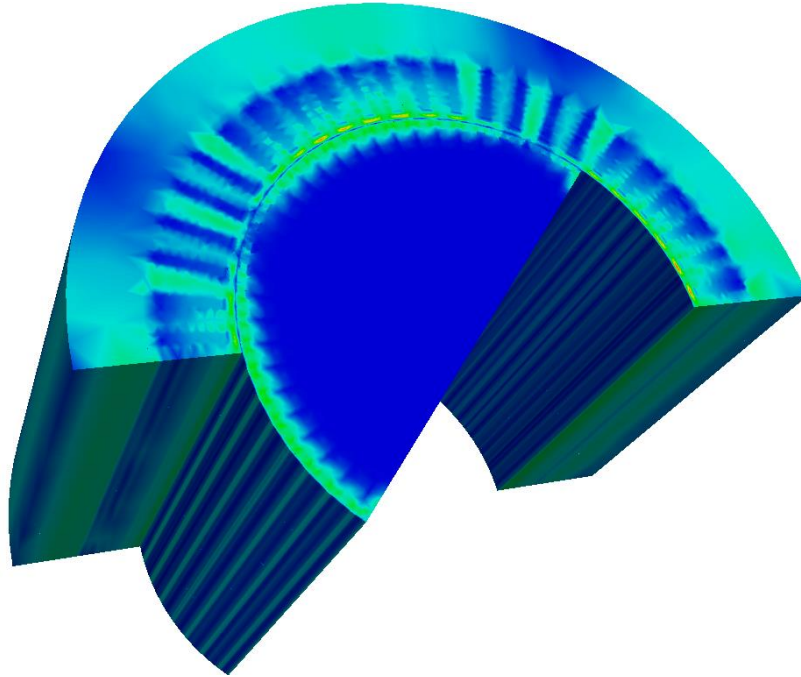


Figure 3. Induction motor model, the magnitude of magnetic flux density on the surface of the motor. Only half of the motor is modelled, the other half is periodically symmetric.

The model is circuit driven, i.e., it is a multi-physical magnetic field-circuit problem. Model's key parameters are tabulated in Table II. The model geometry is invariable along its axis. It contains only the parts of coils that are inside the cores; the end-windings are dealt with circuit parameters that are included in the circuit models. The speed and position of the rotor are solved from the equation of motion. The start-up solution is used for the tests, meaning that the rotation speed and magnetic fields in the motor are zero at the beginning of simulation. The materials of the motor are magnetically nonlinear. All the magnetic materials have real BH-curves, but all except copper parts and axle are non-conducting, i.e. modelled without eddy currents.

Table 2. Key input parameters of the induction motor model and computation.

Time step length [s]	0.001
Total number time steps [-]	50 or 65
Peak value of supply voltage [V]	3000
Stator winding connection [-]	Y
Modeling of rotor speed [-]	Speed and position solved from the equation of motion
Target speed [rpm]	1500
Load torque [Nm]	0

Elmer's air-gap rotation model still has limitations. For efficient parallelisation, special partitioning scheme is needed (see [27] for details). This requires a special mesh. Presently we extrude a 2D mesh to axial direction, i.e. the mesh is prismatic. This type of mesh allows air-gap partitioning in axial direction for parallel computation.

## 2.3 Results

The magnetic field solution of the two test problems are show in Figure 2 and Figure 3. The results in more detail are represented in the publication [27]. Figure 4 depicts the magnetic flux density on the end-winding surfaces. Further, Figure 5 shows the magnetic flux density on the induction motor model at a certain time-step.

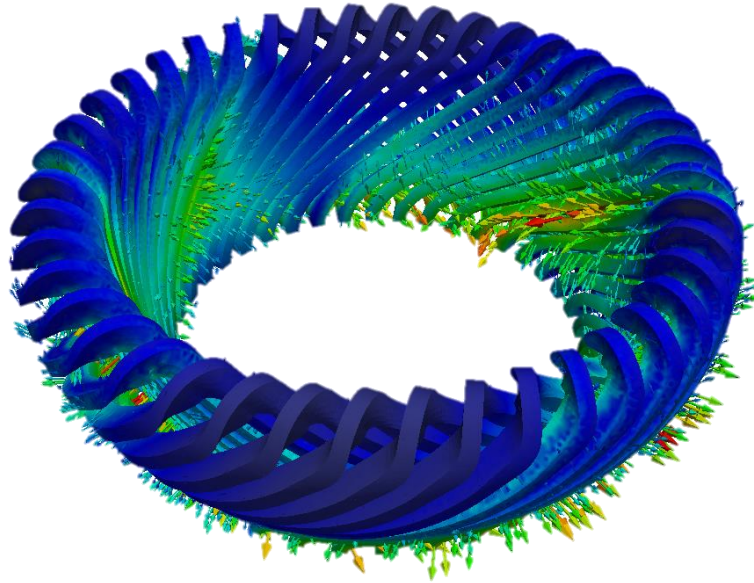


Figure 4. End-winding model, the vectors of magnetic flux density ( $B$ ) and magnitude of  $B$ .

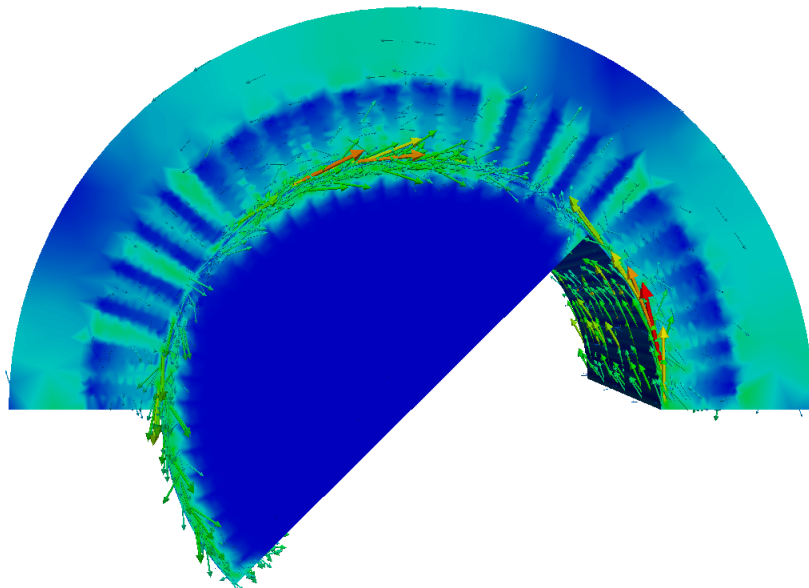


Figure 5. Induction motor model, the magnetic flux density ( $B$ ) vectors and  $B$ 's magnitude on the surface of the motor.

Based on the end-winding results presented in [27], the parallel efficiency of present Elmer version in non-rotating electromagnetic problems is surprisingly good. The non-rotating test case could utilise effectively up to 500 CPU cores. If we include the rotation model along with the electrical circuit models, the parallelisation is more challenging, but our test results show scaling up to 60 CPUs. With the non-rotating model, a commercial alternative could effectively only utilise less than 5 cores. Hence, Elmer is shown to linearly scale up to substantially higher number of cores than a well-known commercial alternative, in detail, Elmer is more than 40-fold faster with the same hardware.



### 3 Electrical machine optimisation with DAKOTA

#### 3.1 DAKOTA

DAKOTA (Design Analysis Kit for Optimization and Terascale Applications) is an open source framework for design optimisation, parameter estimation, uncertainty quantification, and sensitivity analysis. It is developed in Sandia National Laboratories, Livermore, USA, and can be used as a higher level analysis tool that can be connected to external numerical analysis tool. [28]

In this task, DAKOTA was used for electrical machine optimisation. Both single and multi-objective algorithms were tested. It turned out that many algorithms were not able to handle discrete design variables. Only SOGA (Single Objective Genetic Algorithm) and MOGA (Multi Objective Genetic Algorithm) were able to do this, so they were chosen.

The main purpose of using DAKOTA was to provide comparative results for researchers at Aalto University and University of Jyväskylä. At the same time, the idea was to get experience of DAKOTA and to find out how well it is suited in optimisation of electrical machines.

#### 3.2 Results and discussion

In order to test the single objective optimisation, 50 different runs were performed with different starting point using the SOGA algorithm. Additionally, the same test was done for different maximum number of function evaluations: 500, 1000, 5000, 10000, 15000, 20000, and 25000. The torque density was chosen as the objective. Table 3 shows the results of the tests. The columns show results for different number of function evaluations. Because 50 different runs were done for each case, 50 results were obtained. Table 3 shows the best, the worst, the average, the median, and the standard deviation of the 50 obtained optimum values. It can be seen, that 500 function evaluations is not enough to find the global optimal point, because the obtained torque density is much smaller than for the other columns. However, the average of the 50 runs stays almost the same after 5000 function evaluations.

Table 3. SOGA test results, optimal torque density (Nm/m<sup>3</sup>) with different maximum number of function evaluations, selected values and measures from 50 runs.

	500	1000	5000	10000	15000	20000	25000
Best	26,1796166	47,8802905	40,3626063	40,3246745	49,38166	40,3246745	40,3246745
Worst	16,4840982	19,0602415	21,8181246	21,8181246	21,6703281	21,8181246	21,8181246
Average	21,0380483	25,7574942	29,5928555	29,4293924	29,8345144	29,3660253	29,4293924
Median	20,6823971	24,8753533	28,303142	28,303142	28,1441459	28,303142	28,303142
Standard deviation	2,18227281	4,7678476	5,52008933	5,31599109	6,12701161	5,3015773	5,31599109

The results show that DAKOTA can be used together with MATLAB code to optimise electrical machines. However, it remains unclear whether it is the best available tool for this, because in the end not enough data was gathered. However, this data could be used to compare the performance of DAKOTA with some other optimisation tools. For a person not familiar

with DAKOTA it requires a little bit learning and going through the manual, before being able to create a properly working DAKOTA input file. However, the amount of work done was not too excessive, and DAKOTA is clearly an option for electrical machine optimisation.

There are other alternatives for optimization using open-source and free software. E.g. optimization algorithms available from open-source scientific computing python-based environment SciPy can be effectively utilized for optimizations of electrical machines. Particularly, differential evolution (DE) algorithm can be effectively used to find a global optimum of a multivariable multi-objective optimization problem which involves discrete and continuous variables. Being a population-based evolution algorithm, DE is very suitable for parallelization. [29-31]

## 4 Acknowledgements

---

The Elmer related work in this topic was done in co-operation with ParallaX project (first and second years), also funded by Tekes, as a part of ABB-VTT business case. The authors want to thank CSC, ABB, and Trafotek for co-operation. The work still continues in ParallaX and SEMTEC projects (the latter is also a Tekes funded project). In optimisation related research, the work was made in collaboration with University of Jyväskylä (JYU) and Aalto University. A journal publication will be done in collaboration with JYU.

## References

---

- [1] ANSYS Maxwell. Available at: <http://www.ansys.com/Products/Simulation+Technology/Electromagnetics/Electromechanical+&+Power+Electronics+&+Mechatronics/ANSYS+Maxwell>. Accessed Aug. 30, 2013.
- [2] Flux by CEDRAT. Available at: <http://www.cedrat.com/en/software/flux.html>. Accessed Aug. 30, 2013.
- [3] Comsol Multiphysics. Available at: <http://www.comsol.fi/>. Accessed Aug. 30, 2013.
- [4] JMAG - Simulation Technology for Electromagnetic Design. Available at: <http://www.jmag-international.com/>. Accessed Aug. 30, 2013.
- [5] COBHAM - Opera Simulation Software. Available at: <http://www.cobham.com/about-cobham/aerospace-and-security/about-us/antenna-systems/specialist-technical-services-and-software/products-and-services/design-simulation-software/opera.aspx>. Accessed Aug. 30, 2013.
- [6] GetDP: a General Environment for the Treatment of Discrete Problems. Available at: <http://geuz.org/getdp/>. Accessed Aug. 30, 2013.
- [7] Elmer - Open Source Finite Element Software for Multiphysical Problems. Available at: <http://www.csc.fi/english/pages/elmer>. Accessed Aug. 30, 2013.
- [8] Pyrhönen J, Jokinen T, Hrabovcová V. Design of Rotating Electrical Machines. Chichester, UK: John Wiley & Sons, Ltd.; 2008.
- [9] Manninen A. Evaluation of the effects of design choices on surface mounted permanent magnet machines using an analytical dimensioning tool. Aalto University: Master's Thesis; 2012.
- [10] Arkkio A. Analysis of induction motors based on the numerical solution of the magnetic field and circuit equations. : Helsinki University of Technology; 1987.
- [11] Dular P, Geuzaine C. GetDP reference manual: the documentation for GetDP a general environment for the treatment of discrete problems. Available at: <http://www.geuz.org/getdp/>. Accessed Jan 29, 2015.
- [12] Geuzaine C, Remacle J. Gmsh: a three-dimensional finite element mesh generator with built-in pre-and post-processing facilities, 2007. International Journal for Numerical Methods in Engineering 2009;79(11):1309-1331.
- [13] Falgout RD, Yang UM. hypre: A library of high performance preconditioners. Computational Science—ICCS 2002: Springer; 2002. p. 632-641.
- [14] Heroux MA, Bartlett RA, Howle VE, Hoekstra RJ, Hu JJ, Kolda TG, Lehoucq RB, Long KR, Pawlowski RP, Phipps ET. An overview of the Trilinos project. ACM Transactions on Mathematical Software (TOMS) 2005;31(3):397-423.
- [15] MUMPS: a MULTifrontal Massively Parallel sparse direct Solver. Available at: <http://mumps.enseeiht.fr/>. Accessed Oct 6, 2015.

- [16] Van Riesen D, Monzel C, Kaehler C, Schlensok C, Henneberger G. iMOOSE-an open-source environment for finite-element calculations. *Magnetics, IEEE Transactions on* 2004;40(2):1390-1393.
- [17] Finite Element Method Magnetics. Available at: <http://www.femm.info/wiki/HomePage>. Accessed Jan 29, 2015.
- [18] FEMAG. Available at: [http://people.ee.ethz.ch/~femag/englisch/index\\_en.htm](http://people.ee.ethz.ch/~femag/englisch/index_en.htm). Accessed Jan 29, 2015.
- [19] FEMAG - Das FE-Programm für elektrische Maschinen. Available at: <http://www.femag.de/>. Accessed Jan 29, 2015.
- [20] TetGen, A Quality Tetrahedral Mesh Generator and a 3D Delaunay Triangulator. Available at: <http://wias-berlin.de/software/tetgen/>. Accessed Oct 6, 2015.
- [21] NETGEN - automatic mesh generator. Available at: <http://www.hpfem.jku.at/netgen/>. Accessed Oct 6, 2015.
- [22] Open CASCADE. Available at: <http://www.opencascade.com/>. Accessed Oct 6, 2015.
- [23] SALOME: The Open Source Integration Platform for Numerical Simulation. Available at: <http://www.salome-platform.org>. Accessed Aug 9, 2013.
- [24] Salome platform component model for numerical simulation. *Computer Software and Applications Conference, 2007. COMPSAC 2007. 31st Annual International: IEEE; 2007.*
- [25] VTK. Available at: <http://www.vtk.org/>. Accessed Oct 6, 2015.
- [26] Henderson A. *The ParaView guide: a parallel visualization application.* : Kitware; 2007.
- [27] Keränen J, Pippuri J, Malinen M, Ruokolainen J, Raback P, Lyly M, Tammi K. Efficient Parallel 3-D Computation of Electrical Machines With Elmer. *Magnetics, IEEE Transactions on* 2015;51(3):1-4.
- [28] DAKOTA version 5.4 Online Manual. Available at: <https://dakota.sandia.gov>. Accessed May 1, 2015.
- [29] Ponomarev P, Petrov I, Bianchi N, Pyrhonen J. Selection of Geometric Design Variables for Fine Numerical Optimizations of Electrical Machines. *Magnetics, IEEE Transactions on* 2015;PP(99):1-1.
- [30] Stipetic S, Miebach W, Zarko D. Optimization in design of electric machines: Methodology and workflow. *ACEMP – OPTIM – ELECTROMOTION 2015 August 2015:1-8.*
- [31] Lampinen J, Zelinka I. Mixed Integer-Discrete-Continuous Optimization by Differential Evolution - Part 1: The Optimization Method. *5th International Mendel Conference On Soft Computing 1999:71-76.*