FEM using projection of physical properties suitable for movement modeling and optimization processes.

B. Ristagno[1,2], D. Giraud[1,2], J. Fontchastagner[1],
D. Netter[1], N. Takorabet[1], G. Devornique[2] and N. Labbe[2]

[1] Université de Lorraine, GREEN, 54500 Vandoeuvre-lès-Nancy, France

[2] Valeo Electrical System, 38070 Saint-Quentin-Fallavier, France

baptiste.ristagno@univ-lorraine.fr

Abstract

Purpose - Optimization processes and movement modeling usually require a high number of simulations. The authors propose to reduce global central processing unit (CPU) time by decreasing each evaluation time.

Design/Methodology/Approach - Remeshing the geometry at each iteration is avoided in the proposed method. The idea consists in using a fixed mesh on which functions are projected in order to represent geometry and supply.

Finding - Results are very promising. Central processing unit (CPU) time is reduced for three dimensional problems by almost a factor two, keeping a low relative deviation from usual methods. CPU time saving is doable by avoiding meshing step, and with a better initialization of iterative resolution. Optimization, movement modeling and transient-state simulation are very efficient and give the same results as the usual FE method.

Research limitations implications - The method is restricted to simple geometry due to the difficulty of finding spatial mathematical function describing the geometry. Moreover, a compromise between imprecison - caused by the boundary evaluation - and time saving must be found.

Originality/Value - The method can be applied to optimize the design of rotating machines. Moreover, movement modeling is performed by shifting functions corresponding to moving parts.

Keywords - Finite Element Method, Mesh, Projection

1 Introduction

Nowadays, Finite Element Method (FEM) is widely used in electromagnetic field modeling which provides very accurate results compared to experimentations. It also allows the description of complex geometries and problems with a very simple implementation. These advantages explain the spread of FEM in many fields. Nevertheless, for complex three dimensional problems, it requires huge computational resources and consequently very long CPU time. Optimization processes usually need a high number of calls of the objective function. Therefore, FEM does not seem suitable in 3D optimization processes. In literature, several modeling methods, such as hybrid models which combine finite element and reluctance network, have been suggested in order to reduce simulation time Devornique et al. (2017). In this article, the authors propose a method allowing meshing step saving in any iterative process such as movement modeling. The mesh being fixed, all physical parameters (properties, supply or armature movement) are implemented by projecting mathematical functions on the mesh grid. This method is applied to a usual permanent magnet synchronous machine (PMSM) in 2D and in 3D. Interesting results are obtained and discussed.

2 Methodology

2.1 A brief overview of the usual way of using FEM

In the FEM method, each elementary part of the initial geometry is implemented in the software. This discretization is then used to build an adapted mesh in which all nodes match with material boundaries. Behavior law and physical properties of all materials are defined based on this mesh. Consequently, we can take the example of the reluctivity ν. It is a piecewise function defined as ν_i with i corresponding to different material regions Ω_i. The same characteristic can be highlighted on the current density $\mathbf{j}$ or the remanent flux density $\mathbf{b_r}$, both often used in magnetic problems, as seen in Figure 1.

The global weak formulation is applied to the whole domain using all the piecewise functions as explained by Bastos & Sadowski (2016). For example, a permanent magnet synchronous machine implies the weak formulation along with vector potential $\mathbf{a}$ seen in (1).

$$\begin{split}\sum_k \int\limits_{\Omega_k} \nu_k(\|\mathbf{curl}\,\mathbf{a}\|)\,\mathbf{curl}\,\mathbf{a}\cdot\mathbf{curl}\,\mathbf{a}' d\Omega_k - \sum_l \int\limits_{\Omega_l} \nu_l(\|\mathbf{curl}\,\mathbf{a}\|)\,\mathbf{b_r}_l\cdot\mathbf{curl}\,\mathbf{a}' d\Omega_l \\ - \sum_m \int\limits_{\Omega_m} \mathbf{j}_m\cdot\mathbf{a}' d\Omega_m = 0, \forall \mathbf{a}' \in \mathbf{H_0}(\mathbf{curl},\Omega)\end{split} \tag{1}$$

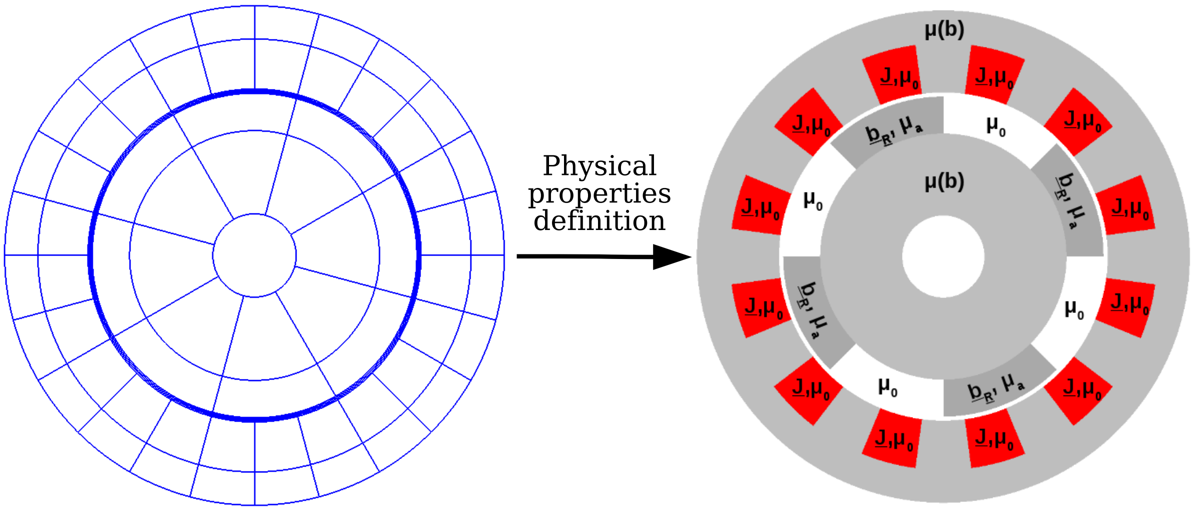

Figure 1: Principle of the usual method applied on a conventional PMSM

where Ω is the whole domain ($\cup\Omega_i$, $i \in [1;\text{number of subdomains}]$, Ω_l magnets subdomains with $l \in [1;\text{number of magnet subdomains}]$ and Ω_m source subdomains with $m \in [1;\text{number of source subdomains}]$. Numerical integration method applied to weak formulation is strongly linked to spatial discretization of geometry. Some iterative processes as optimization problem or movement modeling need to modify the geometry and consequently the weak formulation support. As iterative processes required a huge number of steps, all these pre-processing part reevaluations represent a non negligible waste of time, especially in 3D.

2.2 Proposed approach

The authors propose to develop a simple method to avoid the pre-processing part reevaluations in an iterative process, similar to the moving band method in the 2D rotating movement modeling. In the case of optimization processes only few methods exist to reduce the time of each evaluation as in Devornique et al. (2018) in which efficient FE models are developed to find a balance between precision and CPU time. That is why the authors proposed, by dint of a trick, a different approach in the use of FEM which mathematically respects the standard weak form. Two crucial points have to be pointed out concerning the traditional way of using FEM :

- The discretization of material boundaries is a good basis for the mesh. Indeed, all nodes and edges perfectly match with boundaries, allowing a good evaluation of behavior law and physical properties at each node, thus at each boundaries. In conclusion, discretization is useful to correctly define material boundaries.

- According to the previous spatial discretization, material properties are implemented in each spatial area, defining a piecewise function of the global domain.

The implementation of piecewise functions (an example is given in (2)) has to be respected. On the other hand, the fact that nodes do not match with boundaries only implies some imprecisions. Considering these facts, the authors chose to define material properties by projection of mathematical functions on a mesh grid to represent the geometry. In other words, the first step is to only define the external boundary of the problem. Using these boundaries, a fixed mesh can be created and used as a support for physical property functions. Indeed, physical properties are no more implemented in discrete regions, as seen in Figure 1, but projected on the whole meshed domain by using space dependent piecewise functions as seen in Figure 2. The whole geometry and supply can be defined thanks to this method.

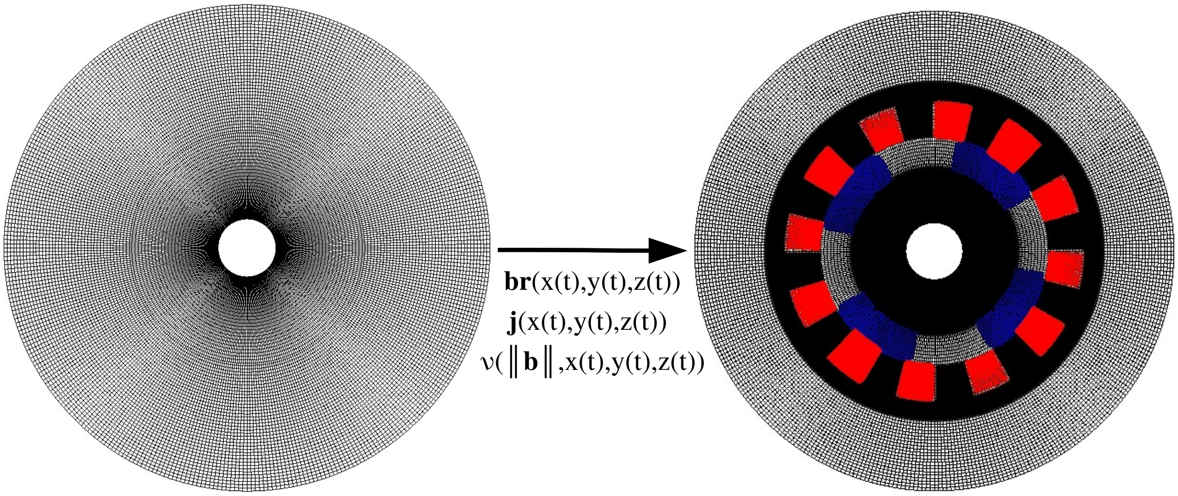

Figure 2: Principle of the proposed method applied on a conventional PMSM

In the case of the previous PMSM, presented in Figure 1 and Figure 2, two physical property functions, the reluctivity ν(x, y, z, t, $\|\mathbf{b}\|$) and the remanent induction $\mathbf{b_r}$(x, y, z, t), along with a source, the current density $\mathbf{j}$(x, y, z, t), are required to describe the complete problem. In the example of the current density, it can be described as in (2), where R_1 and R_2 are inner and outer radii of slots, Θ the angles origin, τ_r the slot pitch, ω the pulsation of slot current, I_{max} the maximum slot current and S_{slot} the slot surface.

$$\mathbf{j}(x,y,z,t)=\left\{\begin{array}{l}
\text{if}\left(\sqrt{x^2+y^2}\geq R_1\ \ \&\ \ \sqrt{x^2+y^2}\leq R_2\right)\\
\quad\text{for}\ \ k=1:1:\text{Number of slots}\\
\qquad\quad\text{if}\bigl(atan\left(\frac{y}{x}\right)>\Theta+2k\tau_r\ \ \&\ \ \bigl(atan\left(\frac{y}{x}\right)<\Theta+\tau_r+2k\tau_r\bigr)\\
\qquad\qquad\frac{I_{max}}{S_{slot}}\cos(\omega t+k\frac{2\pi}{3})\,\mathbf{e_z}\\
\qquad\quad\text{else}\\
\qquad\qquad 0\,\mathbf{e_z}\\
\qquad\quad\text{endif}\\
\quad\text{endfor}\\
\text{else}\\
\quad 0\,\mathbf{e_z}\\
\text{endif}
\end{array}\right. \tag{2}$$

With this approach, defining the external boundary and an adapted fine mesh (discussed in 2.3) is enough to build the model. The mesh is kept unchanged for all iterations of any iterative process, which saves CPU time by avoiding pre-processing evaluatation step. Geometry is defined by projection of physical property functions mentioned above, as seen in Figure 2. Mathematically, the weak formulation seen in (3) is kept unchanged as in (1). Obviously, both formulations are the same, there are only seen from a different point of view.

$$\begin{aligned}
&\int\limits_{\Omega}\nu(x(t),y(t),z(t),\|\mathbf{curl}\,\mathbf{a}\|)\,\mathbf{curl}\,\mathbf{a}\cdot\mathbf{curl}\,\mathbf{a}'d\Omega\\
&\qquad\qquad-\int\limits_{\Omega}\nu(x(t),y(t),z(t),\|\mathbf{curl}\,\mathbf{a}\|)\,\mathbf{b_r}(x(t),y(t),z(t))\cdot\mathbf{curl}\,\mathbf{a}'d\Omega\\
&\qquad\qquad\qquad\qquad-\int\limits_{\Omega}\mathbf{j}(x(t),y(t),z(t))\cdot\mathbf{a}'d\Omega=0,\quad\forall\mathbf{a}'\in\mathbf{H_0}(\mathbf{curl},\Omega)
\end{aligned}\tag{3}$$

Basically, the aforementioned method has some similarities with the mesh-less method presented by Belytschko et al. (1994) although finite elements are still used to lead to well conditioned matrices. Furthermore, the geometry of finite elements in the proposed method is standard and completely decoupled from the geometry (computer aided design (CAD)) of the simulated device. Whereas in the meshless method, the geometry of the elements is arbitrary (no elements, only nodes) but node placements are strongly linked to the geometry of the device.

2.3 Drawbacks and discussions

In this section, the principal drawbacks of the method will be discussed and some advantages will be highlighted. The major drawback is directly linked to the spatial discretization of the usual way of using FEM. As previously mentioned, the interest of the spatial geometry discretization is every node of mesh corresponds to material boundaries. The fact that the material properties are evaluated each node of mesh allows for very well-defined material boundaries. For instance, a crucial boundary for rotating machines is the air-gap which small variation of thickness deeply impacts both global and local variables.

To avoid re-meshing step, the spatial discretization is replaced by the projection of physical properties on a mesh grid. In this condition, mesh nodes do not match anymore with material boundaries. It gets harder to implement clear and perfect boundaries. Indeed, between two nodes, the mathematical piecewise function is defined by inducing an evaluatation of physical properties thanks to a linear interpolation along the edge Bossavit (1992). This situation leads to irrregular boundaries as seen in Figure 3. The boundary is defined between two nodes, so that it can lead to a deviation of thickness of one element. It is interesting to point out the fact that a ruled mesh provides more regular boundaries as seen in Figure 4, but there is still a deviation of thickness.

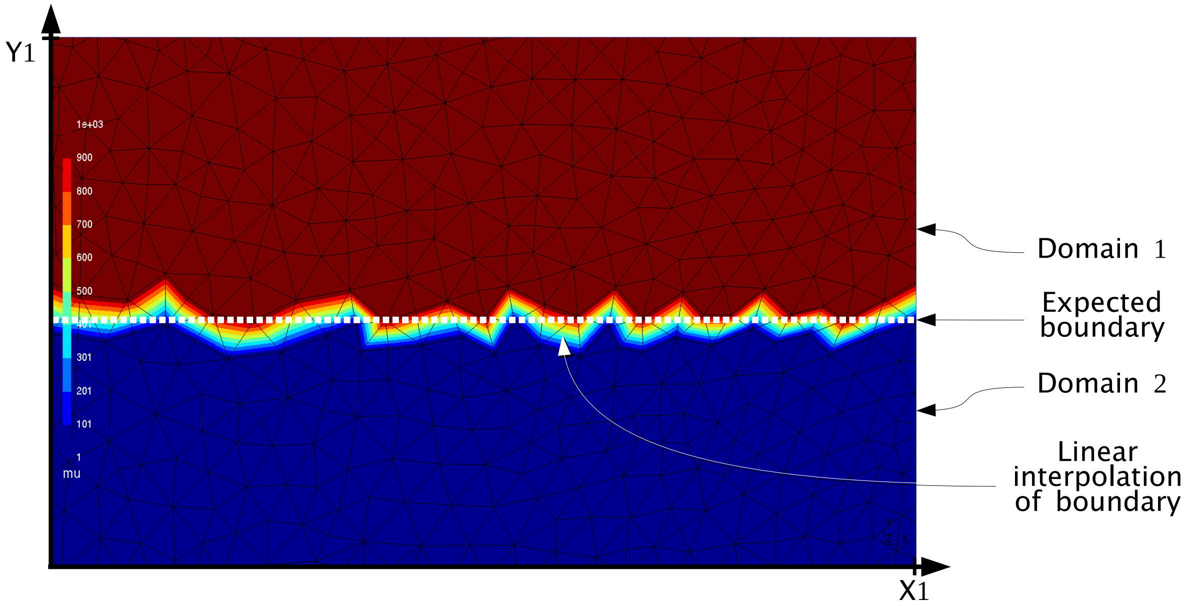

Figure 3: Nodal linear interpolation of physical properties implying boundary deviation with projection method

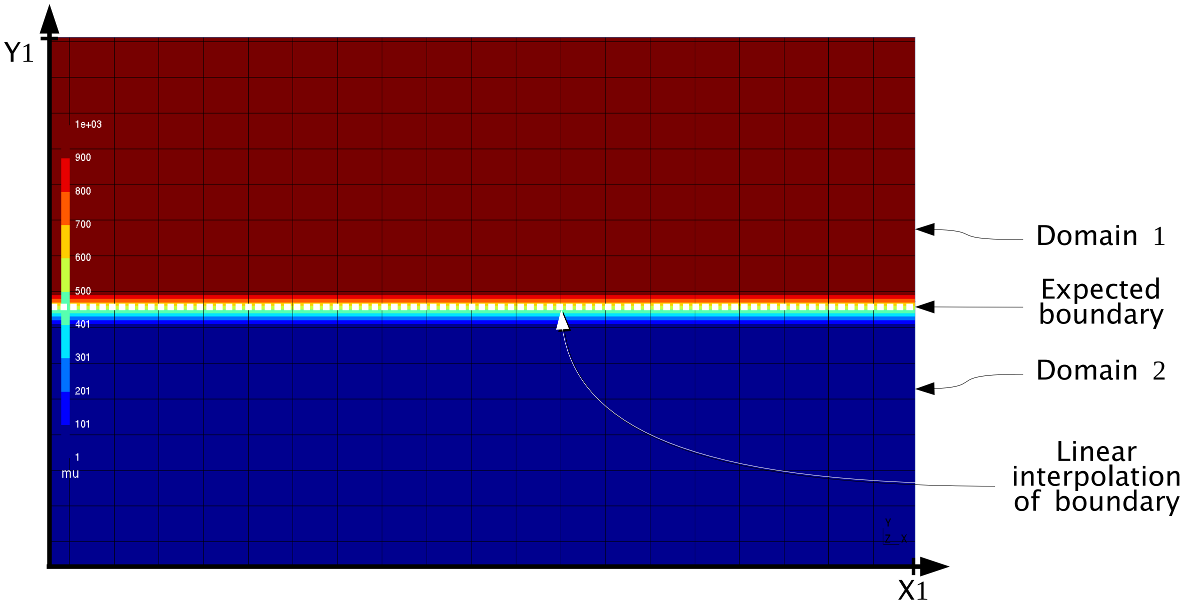

Figure 4: Nodal linear interpolation of physical properties implying boundary deviation with projection method using quadrangle elements

A first solution to reduce the deviation is to reduce the size of the elements to converge to the real boundary. That is to say, an adapted fine mesh can be used to limit the impact of the linear interpolation deviation. Nevertheless, adapting the mesh grid to the smaller mesh element size is a nonsense and will raise the CPU time compared with the usual method because of the high number of Dofs.

Thus, the authors propose to adapt the mesh grid defining some spatial discretizations whose only purpose is to allow a control on the mesh. These geometrical elements are not correlated with real geometry but can be placed wherever the users want to set a fine mesh. Indeed, in the pre-processing part in which external boundaries of the problem are defined (so before the projection step), some non physical points or lines can be placed in interesting areas. Those geometrical elements allow for implementing mesh size near points or number of nodes on lines. Taking the example of Figure 3, the size of the mesh element of the whole domain is about 1 p.u., leading to a deviation of the boundary of 1 p.u. compared to the expected boundary. Defining a line near the boundary allows for reducing the mesh element size only close to the boundary while letting it wide elsewhere. For instance, the mesh element size is reduced by a factor 3 on Figure 5, leading to a more precise evaluation of the boundary, with a deviation of 0.33 p.u. only. Obviously, it is more time-consuming and the meshing time between Figure 3 and Figure 5 is increased by a factor 2. This technique can be illustrated on an electrical machine: the boundary at the air-gap of a PMSM is clearly important, which is why the mesh has to be controlled, but a low deviation at the external boundary of the yoke will not have a deep impact on performance.

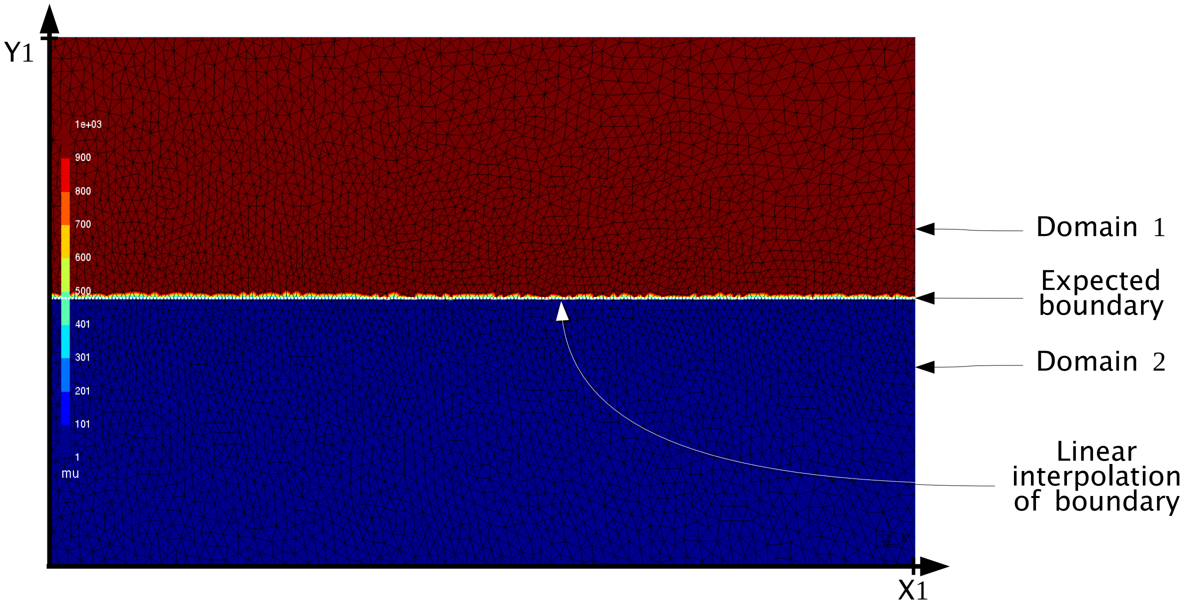

Figure 5: Nodal linear interpolation of physical properties implying boundary deviation with projection method using non physical line to control mesh element size

According to the previous paragraph, there is no mathematical compensation to the nodal linear interpolation. Resulting irregular boundaries can be seen as rough surfaces. This lies in the fact that the used software evaluates pre-processing properties and sources only on the mesh nodes. The linear interpolation between two nodes is quite unavoidable in this case, nevertheless it is imaginable to evaluate properties and sources along mesh edges with a higher order of interpolation of pre-processing functions using maybe another software.

In addition to the pre-processing time saving, the proposed method is mathematically interesting. The iterative process of resolution (the Newton-Raphson method for example) is better initialized than in a multi-static resolution. Indeed, since the mesh is kept unchanged, the solution of non-linear timestep n can be easily used as initialization of non-linear timestep $n+1$. So the CPU time saving is expected to be higher than the saving of pre-processing part.

3 Tests on an application

In order to compare both methods, the same software is used : Onelab interface based on the finite element mesh generator Gmsh Geuzaine & Remacle (2009) and the general finite element solver GetDP Dular & Geuzaine (2019). It was chosen for its flexibility with weak formulations and physical property definitions. All simulations are run with vector magnetic unknowns and magnetostatic formulation.

3.1 Permanent Magnet Synchronous Machine

The study of a usual permanent magnet synchronous machine, 12 slots and 4 poles, is carried out. The first method lies in drawing geometry by CAD and then implement all physical properties in every subdomain created, as seen in Figure 1. The same machine is then implemented by projection of physical functions, as seen in Figure 2. In both cases, particular attention is paid to implement a fine mesh near the airgap. Indeed, the rotation step and mesh element size are equalized and three nodes are placed in radial length of the airgap. In the first method, 15 nodes are implemented on a half slot pitch whereas in the second one, 360 nodes are implemented on the circumference. That is to say, both models have 360 nodes on their circumference with a one-degree rotation step. But the second model implies a constant mesh element size, as in the first one, the number of nodes on a half slot pitch is constant.

Flux density maps of both machines, in magnetostatics 2D and 3D, are compared first. Qualitative correspondence are observed on Figure 6 and Figure 7, which represents an encouraging first step for the validation of the proposed method. Nevertheless, an iterative process has to be carried out to check the expected advantages of projection method and a complete statistical study is required to try to avoid any confirmation bias. Comparisons are made between the proposed method and a basic approach without any special technique (as a moving band or a sliding surface) which can be considered as a "naive" method. In the case of basic movement modeling, a pre-processing part (geometry and mesh) is performed at each position and the system is solved. For a number of positions $Nb_{positions}$, the system is meshed and solved $Nb_{positions}$ times, so:

$$CPU_{Time} = Nb_{positions} * (Meshing_{Time} + Resolution_{Time})$$

In the case of the proposed movement modeling, the meshing step is avoided at each iteration. Moreover, due to the low complexity of the fixed mesh grid, the meshing step is faster than the usual method for same Dofs number. This saved time is hard to evaluate because it depends of the problem geometry but it can be represented by the ε variable ($\in]0;1]$) in the following. The non-linear convergence algorithm being also better initialized in the case of the proposed method, the resolution time is reduced by a factor depending on geometry and supply. CPU time with expected time saving are sum up in Table 1.

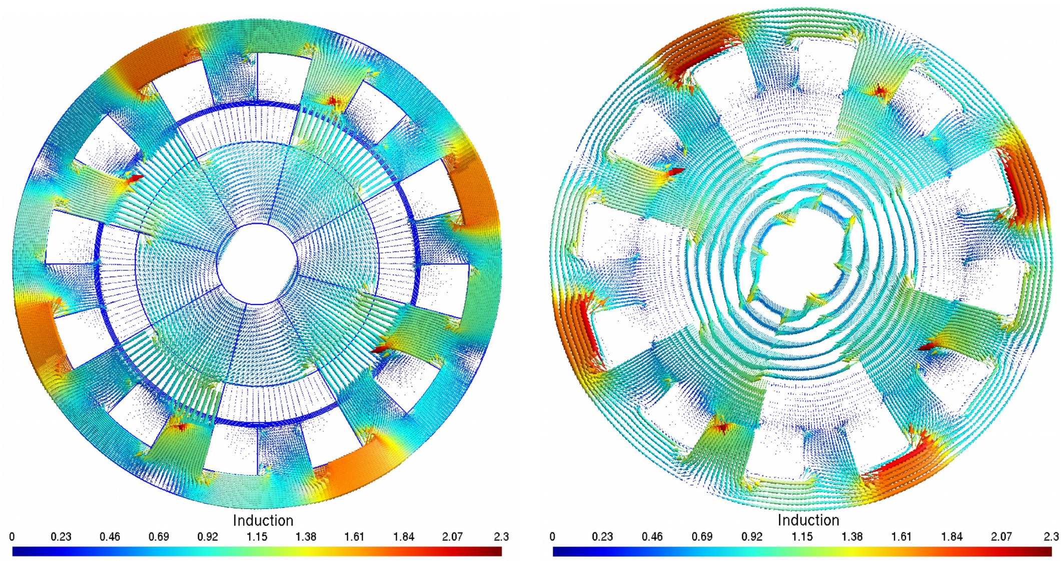

Figure 6: 2D flux density map for usual method (left) and proposed method (right)

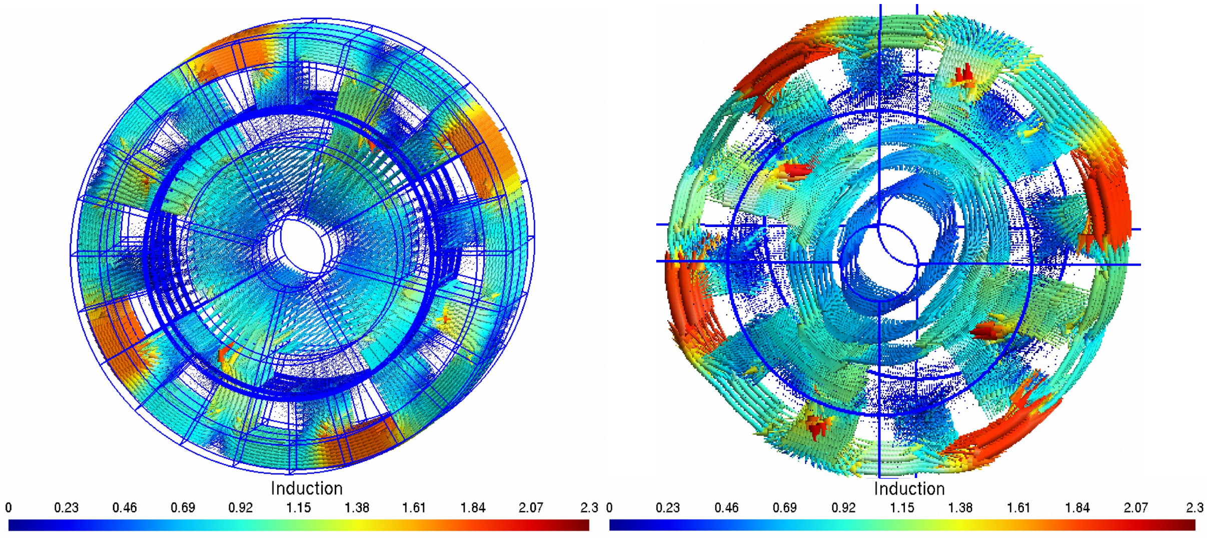

Figure 7: 3D flux density map for usual method (left) and proposed method (right)

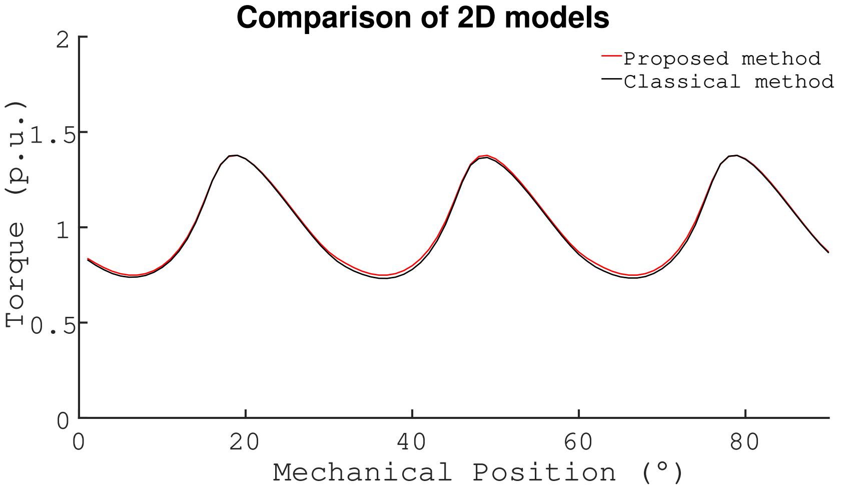

Figure 8: Torque comparison for 2D models

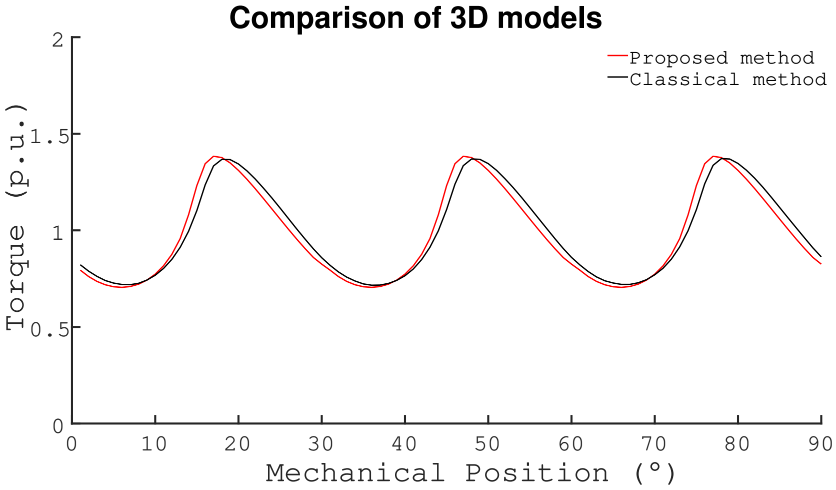

Figure 9: Torque comparison for 3D models

$$CPU_{Time} = \varepsilon * Meshing_{Time} + Nb_{positions} * Resolution_{Time}$$

Table 1: Time saving

	Usual method	Proposed method
Linear problem		
Meshing time	1	$\varepsilon{*}1/Nb_{position}$
Resolution time	1	1
Non-Linear problem		
Meshing time	1	$\varepsilon{*}1/Nb_{position}$
Resolution time	1	1/2

Multistatic torque is computed for both models in 2D and 3D as it can be seen on Figure 8 and Figure 9. The relative deviation between black and red curves is kept under 1% for 2D case. But as expected, the observed saved time is very low because of the unsignificant meshing time. 2D results being encouraging, 3D models are developed in order to have same number of Dofs (around 130,000). The CPU time is divided by a factor 1.7 for our proposed method, keeping a relative deviation under 5% using a Intel Core i7 2.70GHz with 32Gb of RAM.

To implement the proposed model in an optimization process, a statistical validation is needed. 500 randomized 3D machines are generated. 4 geometrical parameters vary in a limited range given in Table 2. Airgap, external radius and bore radius remain constants.

Table 2: Geometrical variation

Variable parameters	Variation range
Slot opening (%)	[15 ; 85]% of slot pitch
Magnet opening (%)	[15 ; 85]% of pole pitch
Armature radius (mm)	[72...108]
Yoke radius (mm) (without magnet)	[140 ... 172]

For each machine, the average torque of both methods is computed and compared. Results are shown in Figure 10 for 130,000 Dofs first.

The average relative deviation is around 13% with a standard deviation of 17%. It is interesting to highlight the fact that deviations are centered on a positive value, which

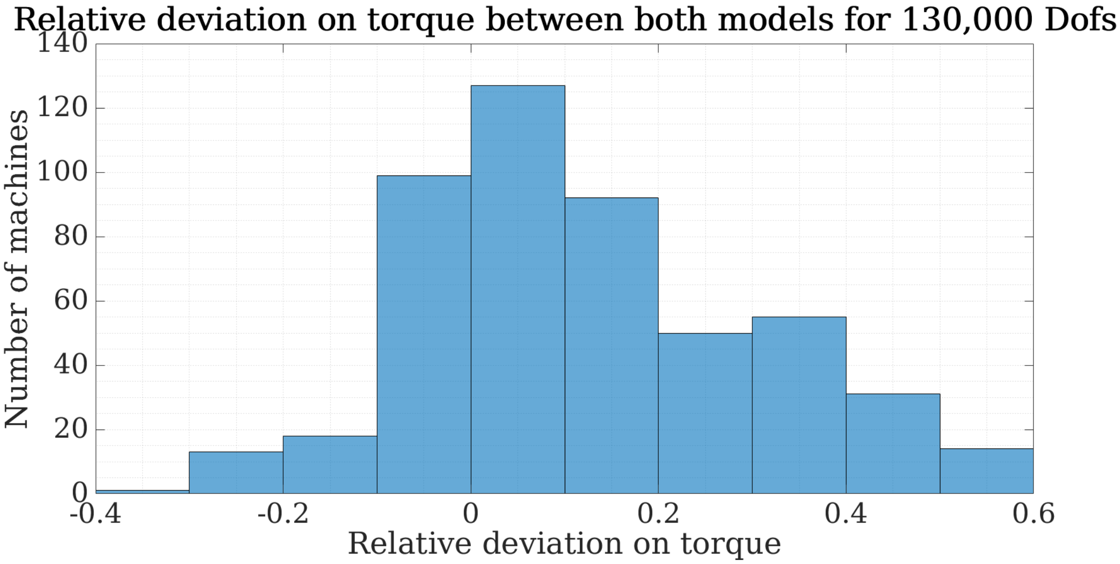

Figure 10: Robustness of proposed method for 130,000 Dofs

means that the proposed method tends to underestimate the ”usual torque”. Considering average and standard deviation, 75% of the simulated machines have a relative deviation below 25% in absolute value. Moreover, it can be observed that refining mesh tends to change the histogram as it can be seen in Figure 11.

Knowing the mean value, the standard deviation and the population size, the confidence interval at 95% on mean value can be estimated. This interval is represented on Figure 12. As expected, refining mesh seems to reduce the relative deviation between both methods by improving the boundary implementation.

Concerning the ε value mentioned in Table 1, the same statistical analysis can be carried out on the 500 machines sample both for 130,000 Dofs and 260,000 Dofs. As a reminder, ε is defined in (4).

$$\varepsilon = \frac{Meshing_{Time,projection}}{Meshing_{Time,usual}} \tag{4}$$

It turns out that the average is 0.132 with a standard deviation of 0.003 for 130,000 Dofs and the average is 0.239 with a standard deviation of 0.005 for 260,000 Dofs. That is to say, meshing time can almost be 10 times faster with projection method than with traditional method but it seems to decrease with Dofs number. However, saved time represented by ε is strongly linked to the geometry discretization (CAD). The analysis of ε only make sense for this particular geometry with our particular choice of discretization.

Nevertheless, according to previous results, the proposed model seems to be suitable for 3D optimization processes. According to Räisänen et al. (2013), there are some additional advantages using a fixed mesh in optimization processes, such as:

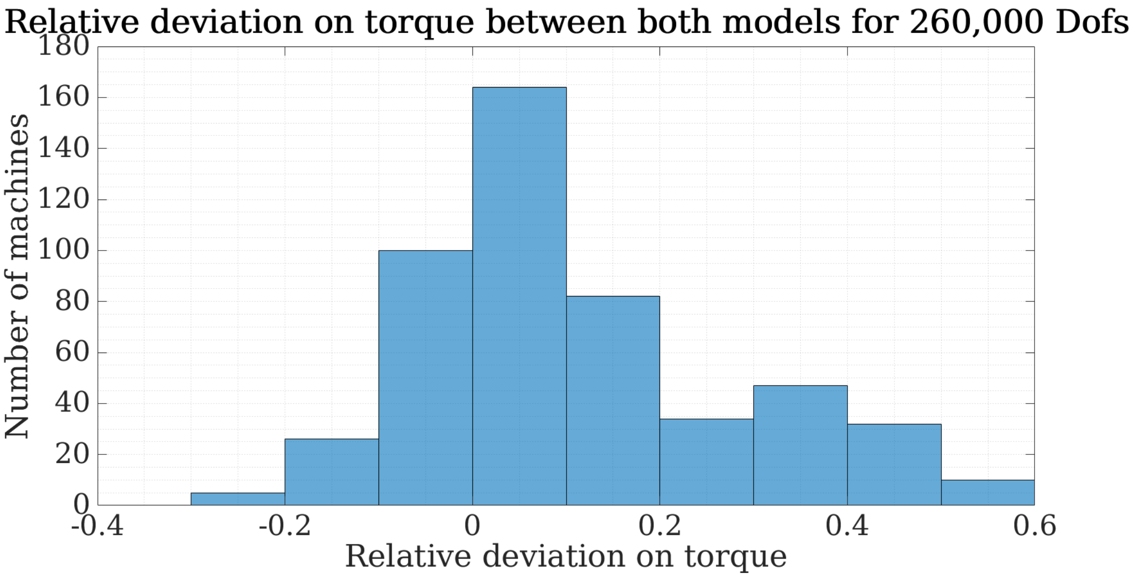

Figure 11: Robustness of proposed method for 260,000 Dofs

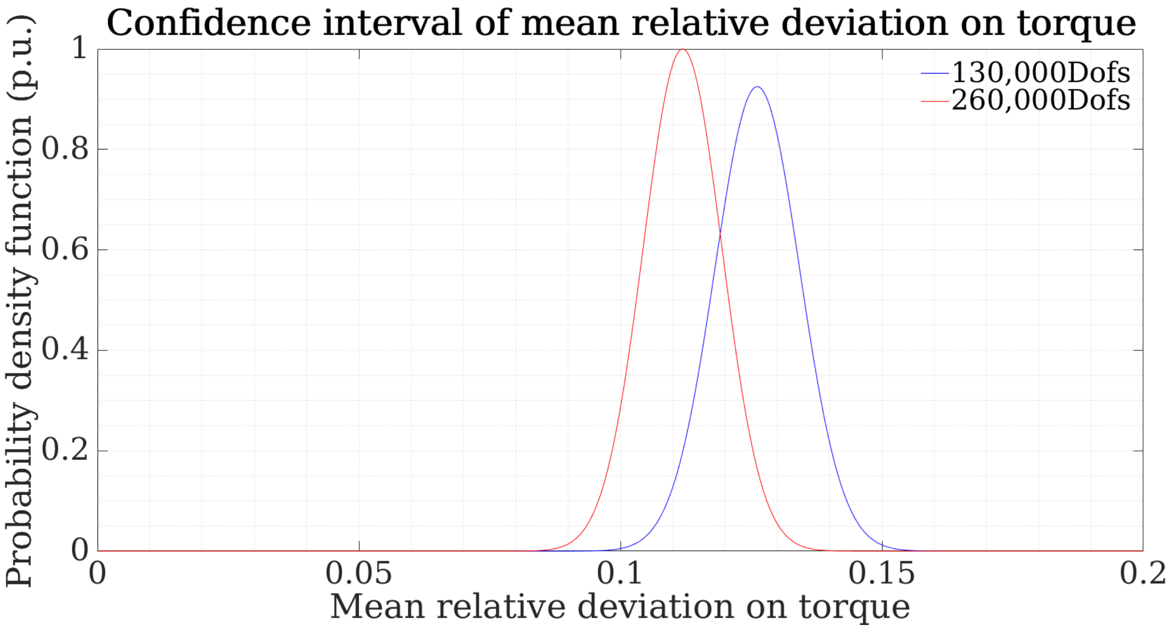

Figure 12: Evolution of confidence interval of mean relative deviation on torque

- Mesh generation might be difficult or even fail for certain configurations;
- Changing the mesh influences the discretization error. For instance, if optimization algorithms are applied, the target function will be polluted with re-meshing noise.

3.2 Optimization problem

Previous results show that multistatic movement of rotating machine is well-modeled by the proposed method. The authors wanted to study the efficiency of the proposed method in some optimization processes. As 2D projected models do not present a real interest, a 3D PMSM geometry optimization will be carried out. The same models as robustness study are used in this section, that is to say the 130,000 Dofs models for time saving purpose. Geometry optimization is applied to the vector solution presented in Table 2. The process consists of a torque ripple Δ_Γ minimisation with minimum mean torque reference $\Gamma_{mean,ref}$ and maximum current density reference $\mathbf{J_{max,ref}}$.

$$\left\{ \min_{X\in\mathbb{R}} \ (\Delta_\Gamma) \ \text{ under } \left\{ \begin{array}{l} c_1 = \Gamma_{mean,ref} - \Gamma_{mean}(X) \leq 0 \\ c_2 = \mathbf{J_{max}} - \mathbf{J_{max,ref}} \leq 0 \end{array} \right. \right.$$

Torque ripple Δ_Γ is defined according to (5) in which Γ_{max} is the maximum of instaneaous torque, Γ_{min} is the minimum of instaneaous torque and Γ_{mean} is the mean value of instaneaous torque.

$$\Delta_\Gamma = \frac{\Gamma_{max} - \Gamma_{min}}{\Gamma_{mean}} \tag{5}$$

This optimization problem is applied to the usual model and to the proposed model as seen in the previous section. NOMAD software Audet et al. (2009) is used to run these optimizations linked to the Onelab interface to solve the magnetic problem. NOMAD is based on a mesh adaptive direct search (MADS) algorithm and is useful for a few variables and a time consuming blackbox evaluation. Both optimization models have been identically parametrized with the same algorithm, objective, constraints and solution vector. Models are investigated by means of three problems, presented in Table 3, in which convergence and stop criteria are modified. Indeed, Problem I is a usual case of optimization process which stops when the relative deviation on the objective function is lower than a fixed limit Deviation$_{min}$. Problems II and III represent two optimization processes in a given maximum time and in a given maximum number of blackbox evaluations.

Results of optimization processes are presented in Table 4. It is very interesting to point out some observations. Problem I shows that both methods converge to the same solution. Compared to the usual method, time saving is observed and the amount of blackbox evaluations increased in the proposed method. Conclusion is that one blackbox evaluation is made twice as fast as the usual method. Thus, the domain is more widely explored in the same period of time with the projection method. Problem II and III

Table 3: Summary of optimization processes

	Problem I	Problem II	Problem III
Objective	Δ_Γ	Δ_Γ	Δ_Γ
Constraints	c_1, c_2	c_1, c_2	c_1, c_2
Convergence	Deviation_{Min}	-	-
Stop criterion	-	Time_{Max}	$\text{BB Evaluation}_{Max}$

confirm this fact, the proposed method can be a little more efficient in a time limited or a blackbox evaluation limited optimization. Both methods converge to a different vector solution and results are re-calculated with the complete model.

Table 4: Results of optimization processes

	Problem I	
	Usual Model	Proposed Model
Δ_Γ	1p.u.	1.01p.u.
CPU Time	52h	38h
Iterations	115	145
	Problem II	
	Usual Model	Proposed Model
Δ_Γ	1p.u.	0.99p.u.
CPU Time	10h	10h
Iterations	20	40
	Problem III	
	Usual Model	Proposed Model
Δ_Γ	1p.u.	0.98p.u.
CPU Time	7.5h	3.8h
Iterations	15	15

3.3 Coupled-circuit simulation

Optimizations on magnetostatic resolution show the efficiency of the proposed method. Nevertheless, transient-state simulation are often useful for machine design, which is why the authors wanted to include in the proposed model the possibility of taking back-electromotive force (EMF) into account. Usually, the magnetic flux is computed by integration on the correct surface, from spatial discretization, using vector magnetic potential $\mathbf{a}$:

$$\varphi_i = \iint_{Wire} \mathbf{b} \cdot \mathbf{dS} = \iint_{Wire} \mathbf{rot}\ \mathbf{a} \cdot \mathbf{dS} = \oint_{slot_i} \mathbf{a} \cdot \mathbf{dl} = \frac{1}{S} \iint_{slot_i} (\oint \mathbf{a} \cdot \mathbf{dl}) dS$$

$$, \text{with } i \in [1, .., N_{phase}] \quad (6)$$

As the proposed model has no surface defined, the normal vector to the wire cross-section **dl** (tangential to current density) is implemented as a piecewise function through a scalar field t_i. The flux term is kept unchanged :

$$\varphi_i = \oint_{\Omega} \mathbf{a} \cdot t_i\, \mathbf{dl}, \ \ \text{where} \begin{cases} t_i = 1. \ \text{in slot} \\ t_i = 0. \ \text{elsewhere} \end{cases} \quad (7)$$

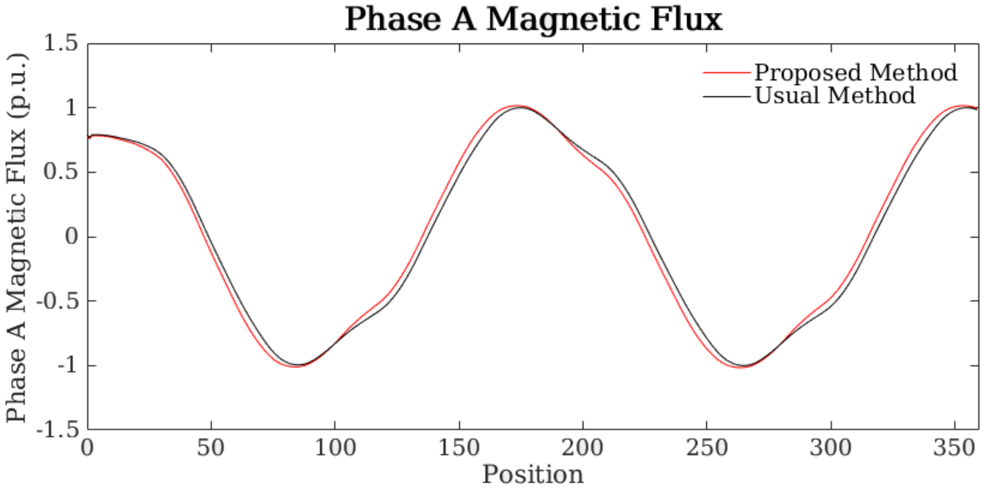

Figure 13: Transient-state magnetic flux computation comparison.

Numerically and mathematically, there is no difference in flux computation between both methods, thus it is not suprising that transient-state fluxes are correctly evaluated, as seen in Figure 13. The low relative deviation observed in flux computation is due to the boundary issue applied to the normal vector $\mathbf{dl_i}$. The back-EMF can be calculated using some numerical derivation method.

4 Conclusion & Future Work

This paper presents the first step of an original method which accurately models magnetic devices by avoiding iterative meshing step. Results are very promising as CPU time is reduced in 3D keeping low relative deviation. CPU time saving is performed by avoiding meshing step, and also by a better initialization of iterative resolution. Optimization, movement modeling and transient-state simulation are very efficient and give same results as the usual FE method. Nevertheless, the use of a fixed mesh grid leads to boundaries issues, therefore to some imprecision in very specific cases. Several advices and solutions are given to overcome this problem. The real interest lies in a kind of mixed model using the usual method for static parts and the projection method for moving and deforming parts. In other words, the purpose of future works is to a develop model in which static parts of the system are discretized using the usual method and moving parts are projected on a fine structured mesh, allowing time saving in iterative processes. In conclusion, this paper represents the first step of a promising method which can be used in several domains of finite element method.

References

Audet, C., Le Digabel, S. & Tribes, C. (2009), Nomad user guide, Technical Report G-2009-37, Les cahiers du GERAD.
URL: *https://www.gerad.ca/nomad/Downloads/user_guide.pdf*

Bastos, J. P. A. & Sadowski, N. (2016), Magnetic Materials and 3D Finite Element Modeling, 1st edition edn, CRC Press.

Belytschko, T., Lu, Y. & Gu, L. (1994), ‘Element-free galerkin methods’, International Journal for Numerical Methods in Engineering **37**(2), 229–256.

Bossavit, A. (1992), ‘Edge-element computation of the force field in deformable bodies’, Transactions on Magnetics **28**(2), 1263–1266.

Devornique, G., Fontchastagner, J., Netter, D. & Takorabet, N. (2017), ‘Hybrid model: Permance network and 3-d finite element for modeling claw-pole synchronous machines’, Transactions on Magnetics **53**(6).

Devornique, G., Fontchastagner, J., Netter, D. & Takorabet, N. (2018), ‘Three dimensional pole shape optimization of claw pole machines based on a hybrid model’, International Journal of Applied Electromagnetics and Mechanics **57**(S1), 73–81.

Dular, P. & Geuzaine, C. (2019), ‘Getdp reference manual: the documentation for getdp, a general environment for the treatment of discrete problems’.
URL: *http://getdp.info*

Geuzaine, C. & Remacle, J.-F. (2009), ‘Gmsh: A three-dimensional finite element mesh generator with built-in pre-and post-processing facilities’, International Journal for Numerical Methods in Engineering **79**, 1309–1331.

Räisänen, V., Kurz, S., Suuriniemi, S., Tarhasaari, T. & Kettunen, L. (2013), ‘How can we deal with moving objects on a fixed mesh?’, Journal of Computational and Applied Mathematics **246**, 260–268.