Modelling Eddy Currents Induced by Rotating Systems

C R I Emson, C P Riley and D A Walsh
Vector Fields Ltd, 24 Bankside, Kidlington, Oxford, OX5 1JE, UK.

Abstract —**As part of a Japanese funded project, a transient eddy current code is being extended to allow for rotational systems. By allowing the system to rotate, induced eddy currents must be modelled. The system being developed uses the "lock-step" approach to model the rotation. As other facilities are included (non-linear materials for example), the solution time of the final code becomes sufficiently large that special care must be taken, and investigations into using parallel hardware becomes necessary.**

I. Introduction

Modelling motional problems has long been considered in 3D eddy current computations, and usually follows the procedure of changing the co-ordinate system to that of the moving source. A modified Ohms Law is used in the moving conductors to include eddy currents due to the motion directly:

$$\mathbf{J} = \sigma\mathbf{E} + \sigma\mathbf{u} \times \mathbf{B} \qquad (1)$$

where **u** is the velocity of the conducting media. This however has the well known restriction that the geometry of the moving parts must be invariant in the direction of motion, allowing a single "snap-shot" solution to be sufficient. This is commonly referred to as a "velocity" solution, and will be used later in this paper as a comparison for benchmarking.

For many problems, including rotating motors and generators, this limitation is too severe, and alternative modelling methods must be used. This has been the basis of a Japanese funded project. The aim of the project, the initial results of which are presented here, are as follows:

- to include eddy currents induced by rotational effects, including rotating geometries, rotating coils, and rotating coil fields,
- to allow various groups of coils, each with varying transient behaviours,
- to allow non-linear materials,
- to use parallel computer systems to reduce the computation time

In addition, the project in the future is also to include coupling to external circuit modellers, and the effects of high temperature superconductors. In this present report, consideration is only given to the bulleted items however.

II. Including Motion

The starting point is a transient three dimensional eddy current software package, that is extended to include the effects of solid body rotation. This requires that the rotating part of the model is physically moved at each time step - it is this rotation that generates eddy currents in the conductors. The extra term in Ohms law, eqn (1), is not then required. It is necessary however to devise a method of taking into account the motion of part of the model. This can be achieved in a number of ways, each having advantages or disadvantages.

A. *Regenerate the Mesh*

The simplest method is to re-mesh the problem at each time step. In general this is a slow procedure, even if automatic mesh generators are used. There is also the difficulty of using the previous time step solution, as the mesh is no longer the same, and the previous solution must be interpolated onto the current mesh. Although feasible, this is far from satisfactory.

It is possible in general to structure the model so that only part of the mesh needs to be built each time step - the other parts of the model having fixed meshes. This saves some of the effort, but is still not recommended.

B. *Boundary Elements*

The boundary element method seems ideal for use in rotating systems, as only the active parts of the problem need to be modelled, and there is no mesh required in between. Therefore, the active parts are free to move with respect to each other in space, and no special attention is required.

There are drawbacks to this method, as follows. As the active parts of the model move, so the matrix associated with the coupling between the various parts changes, and have to be re-computed each time step. Those parts of the fully populated matrix associated with the individual parts do not change however. For very narrow air gap problems, care must also be taken over the matrix generation, especially regarding the evaluation of the near singular integrals associated with close but disconnected surfaces.

C. *Fourier Series*

Global finite elements can be used, where a single element is placed around the slip-surface, using, for example, linear interpolation radially, and a sequence of Fourier terms in the circumferential direction (many other interpolation functions are possible). For many cases this appears attractive, but when the rotor or stator are not smooth, a large number of Fourier terms may be required to achieve sufficient accuracy near the corners.

D. *Lagrange Multipliers*

A method that has been developed, involves the use of a general slip surface between two disconnected meshes. Continuity of the field components is ensured using Lagrange multipliers [1].

Email: Cris.Emson@vector.demon.co.uk
This work has been partly funded by CRIEPI, Japan.

A major advantage is that the mesh on either side of the slip surface can be completely different - both position of nodes, and even size of elements.

Difficulties arising here are largely due to the elements on either side of the slip surface no longer matching, and so the integration of various quantities along the slip surface must be handled correctly. The matrix arising is also singular, but with care a conjugate gradient type of solver can find a solution.

E. Lock-Step

Probably the easiest of the methods is the one adopted here, and uses the technique known as "lock-step"[2]. The elements on the slip surface have a constant subtended angle, $\Delta\theta$, and the rotating part of the mesh is moved $\Delta\theta$ at a time. The mesh is always contiguous, although the time step is then fixed, determined by the element size and the rotational speed.

As the transient solution progresses, the connectivity of the elements adjacent to the slip surface change, and require re-calculation. However, elements not adjacent to the slip surface remain unchanged, and the element matrices associated with these need only be computed once. With this in place, only a minimal amount of computation is required at each time step.

In the present implementation, it is assumed that the slip surface extends the entire length of the model in the z-direction - forming a cylindrical slip-surface.

III. Use of Implied Local Co-ordinates

The formulation used is based on the magnetic vector potential (here using the Coulomb or Lorentz gauges [3]), in conjunction with total and reduced magnetic scalar potentials for non-conducting media. The use of reduced scalar potentials allows coils to be included, without them having to be part of the finite element mesh. In order to couple to vector or total scalar potentials, the coil field is required on the interface.

A. Scalar Potential Volumes

For a volume of reduced potential containing a set of coils, the coil fields due to the contained coils only are required on the interface. Any coils outside this volume can be disregarded. This implies that a volume of reduced scalar potential must either be wholly in the stationary part of the mesh, or within the rotating part of the mesh, and must not span the slip-surface.

For a rotating reduced potential volume, where the included coils are moving synchronously with the mesh, the coil field quantities required on the interface are the normal field component, and the scalar jump across the interface - both scalar quantities.

B. Vector Potential Volumes

Within a rotating vector potential volume, by assuming the element matrices are unchanged from one time step to another, there is an implication of a rotating co-ordinate system within the vector region. This is not a difficulty, but it is necessary to 'un-rotate' the co-ordinate in order to obtain the global vector potential values.

When a reduced potential volume abuts a vector volume, an additional interface term is required due to the coil fields. This term is aligned with the rotating co-ordinate system used in the vector volume, and must be taken into account when recovering the fields. Again the assumption is that a rotating reduced potential volume does not touch a stationary vector potential volume, and *vice versa*, otherwise the coil fields must be transformed to the correct co-ordinate system at each time step, which is to be avoided in the interests of efficiency.

IV. Time Stepping Algorithm

Various options present themselves when devising the time integration method. Since the matrix is changing from one time step to another (due to rotation of the mesh), this too must be taken into account.

The simplest method to adopt is backward time differencing. This is a stable technique, which is essential when there is little or no choice in the time step being used.

In the backward time stepping scheme, it is only necessary to compute the right hand side vectors and the matrix corresponding to the next time step. If a Crank Nicholson method were used in comparison, then the right hand side and matrix needs to be evaluated for the next time step, and the current time step, in order to generate the time stepping procedure. This tends to increase the storage required, and also has the added complication that the matrices at the two time steps have different sparsity patterns (due to the mesh having moved).

A further advantage of backward time stepping is that the results obtained are then valid at the time step in question (unlike Crank Nicholson for example, where the solution is valid at a point midway between time steps). This implies all solutions and boundary conditions are all valid at a specified time step, without the need for further interpolation over a time step.

A. Storing 'Static' and 'Dynamic' Matrices

In order to make the time stepping as efficient as possible, the amount of computation at each time step needs to be minimised. This can be achieved if those parts of the model that are not varying with time are stored separately to the moving parts.

The 'static' part of the model would be the stationary and the rotating meshes. As discussed above, the element matrices do not vary as the model rotates. The 'dynamic' part of the model includes those elements on the slip surface. These need recomputing at each time step, mainly because their connectivity is altered, and hence they contribute to different areas of the system matrix.

This same technique of splitting the model into 'static' and 'dynamic' parts can also be applied to the modelling of non-linear material properties. In this case, the 'static' part of the model includes the linear materials, whereas the 'dynamic' parts involve the non-linear materials.

B. Multiple coil drives

In general, it may be necessary for different coils to have differing time variations. For example, to generate a rotating field, it is necessary to divide the conductors into groups, each group being driven with a sinusoidal variation with time, but differing by a constant phase.

To make the analysis as efficient as possible, the field due to each group of coils is calculated at the outset, along with its contribution to the system right hand side vector, and is stored in the database. At each time step, these are used to construct the overall system right hand side, having been scaled by the appropriate drive functions.

V. Testing

The initial tests were to confirm that the rotating mesh does not corrupt what is otherwise a trivial solution - namely free space with a uniform field imposed.

The next stage is to confirm that a rotating iron brick gives identical solutions to a series of static solutions (since no eddy current are present, the problem is not time dependant).

A. *Eddy Currents*

A circular disc rotating in uniform field is then modelled. After the initial transients associated with suddenly starting the rotation, the solution tends to a steady state solution, which should be the same as a "velocity" solution (since the geometry is invariant in the direction of motion, a single snap-shot can be computed, where the velocity term is included explicitly). The results from the two solution techniques are shown in Fig 1, where the two current patterns are indeed the same.

A difference in the quality of the current potentials arises between the two methods. The velocity solver assigns element velocities (constant per element), giving a piecewise constant component to the current density. The present method however implies a nodal velocity, and the current densities are interpolated from these nodal values. The results in the present method are therefore much smoother, as expected.

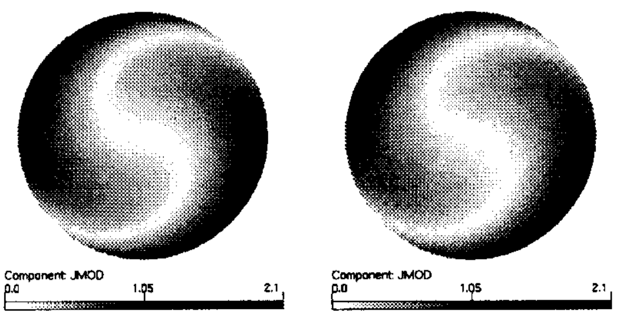

Fig 1: Rotating disk in uniform field showing eddy current pattern using present method (left), and one generated using a "velocity" solver (right)

B. *Rotating Fields*

The final results presented here for a simplified 2-phase induction motor. The driving field is also rotating (produced by coils with sinusoidally varying currents, but with phase differences). The rotor is moving at a different speed to the coil field, hence inducing eddy currents. The element size in this example is based on the most rapidly changing component (either the slip frequency, or the coil field frequency). The field in the iron, and the eddy current densities in the rotor are shown at two time steps in Figs 2 and 3.

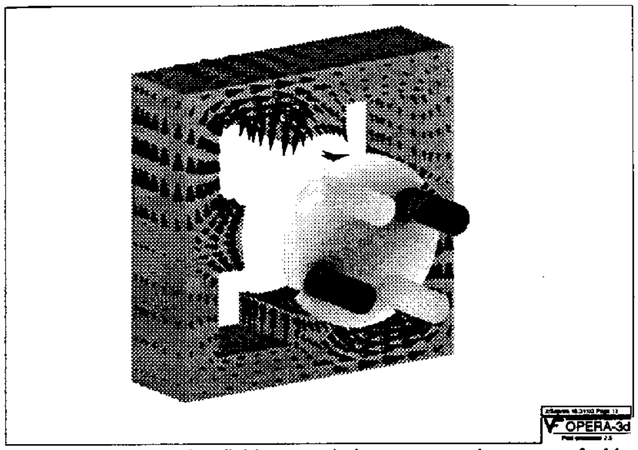

Fig 2: Solution showing field vectors in iron stator, and contours of eddy currents in rotor, in an idealised 2 phase induction motor.

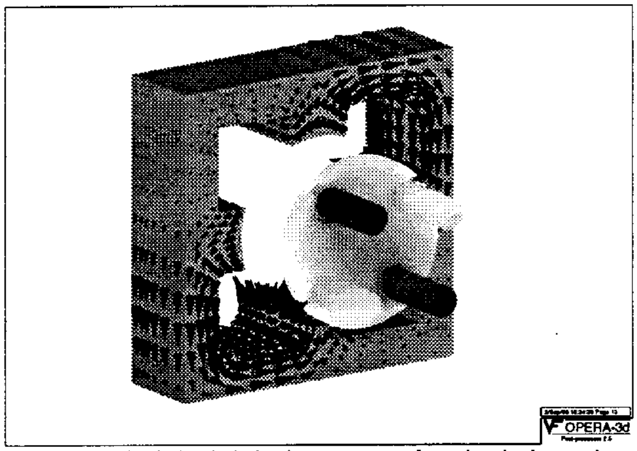

Fig 3: Second solution in induction motor, at a later time in the rotation.

VI. Need for Parallelisation

A feature that quickly becomes apparent when modelling typical problems is that the time required in the simulation becomes very large. This can be shown in the time required to solve a 70MW generator, at a single position only. Table 1 shows the times spent on different parts of the finite element model. Due to the complex coil shapes required, the right hand side computation is the most costly, followed by the matrix solution itself.

These two parts of the model are therefore studied, with a view to writing the algorithms to specifically fit onto parallel architectures.

A. *Parallelisation Of Right Hand Side Generation*

The right hand sides are mainly determined from the

Seconds	%	
119212	100	TOTAL
30	0.0	Sparsity pattern
107744	90.4	RHS calculation
23	0.0	Scalar matrix generation
53	0.0	Vector matrix generation
11321	9.5	Iterative solution of matrix
41	0.0	Field and energy calculation

Table 1: Timing for 70MW generator model, having 26477 nodes, 25536 elements and a total of 48853 equations

coil fields, which themselves are computed by evaluating the Biot Savart integrals given in eqn (1). Adaptive integrals are used to improve the accuracy of these integrals.

$$\mathbf{H}_c(x) = \int_\Omega \frac{\mathbf{J}(x') \times \mathbf{r}}{|\mathbf{r}|^3} dx' \qquad (2)$$

To maintain maximum efficiency of the parallel hardware, it is necessary to ensure that each processor is kept fully loaded. The expressions to be evaluated however vary in the amount of computation required, depending on where the field is required (and hence how far the adaption process in performing the integrals must be extended). The load balancing in this case is achieved by cyclic distribution of integrals over the processors, and as one processor completes its assignment, more integrals are allocated to that processor.

B. Parallelisation of the Solution Phase

In the software package used, the matrix equations are solved using preconditioned conjugate gradients. The preconditioning is inherently scalar, so was changed to a block form of incomplete elimination. This is then parallel in form.

The conjugate gradient procedure relies heavily on matrix-vector products. This is parallelised by partitioning over the processors. Load balancing again must be considered, and also the exchange of "halo data".

C. Parallelisation Test Cases

To investigate the effectiveness of the parallelisation on various hardware, two models were solved:-

(a) sandwich digital recorder head (Elektra-SS)
(b) superconducting magnet shield (Elektra-TR).

The parallel hardware used is shown below. The speed is given in units of MFlops per processor:

Name	*Speed*	*N° of processors*
SGI PowerChallenge	37	2, 4
IBM SP2	66.7	2,4,8
DEC Alpha Farm	4.3	2,4,8
Meiko CS2	10.3	2,4,8,16

The following figures show the results for the two test cases on the various hardware. As is often reported, the speed up is far from ideal, often hampered by the requirements for data transfer between processors.

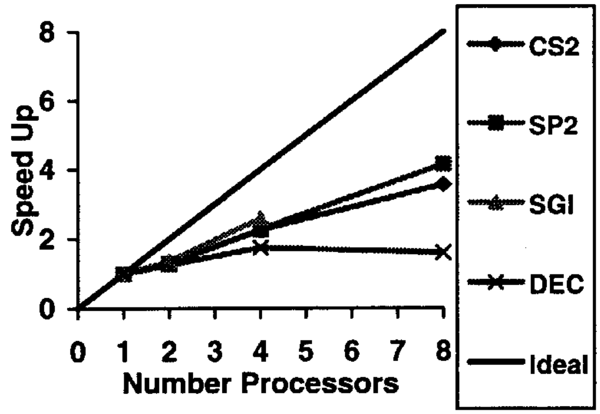

Fig 4: ELEKTRA-TR results

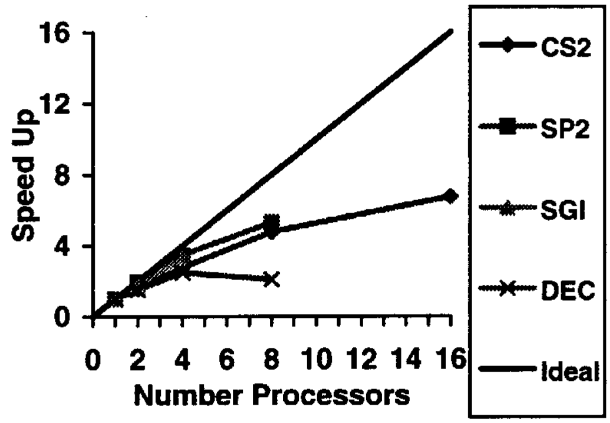

Fig 5: ELEKTRA-SS results

VII. Conclusions

The ability to model rotating systems has been demonstrated, including multiple coil systems (each with its own time variations). Examples were shown verifying the technique and demonstrating its use on a simplified motor. In an attempt to reduce the computation required at each time step, the model is divided into 'static' and 'dynamic' parts, where only elements in the 'dynamic' part of the model are recomputed at each time step. The dynamic part could refer to elements on the slip surface, or non-linear materials.

The main disadvantage of the method discussed is that the time step is defined by the rotational velocity and the element size on the slip surface. To allow full flexibility in the choice of time step, it is necessary to utilise a technique such as Lagrange multipliers on a slip surface.

It has been found that rotating systems require large amounts of computer resources. The need for parallelisation was highlighted, along with some methods that have been used to try to alleviate the problem. The amount of speed up was disappointing, but is similar to that reported by many other authors.

References

1. H C Lai, D Rodger and P J Leonard, "Coupling meshes in 3D problems involving movement", IEEE Trans Mag, Vol 28, No 2, pp1732-4, Mar 1992
2. T Preston *et al*, "Induction motor analysis by time stepping techniques", IEEE Trans Mag, Vol 24, No 1, pp471-4, Jan 1988
3. C F Bryant *et al*, "A comparison of Lorentz and Coulomb gauges in the solution of Maxwell's equations", Presented at 3rd Int Conference on Approximations and Numerical Methods for the Solution of the Maxwell Equations, Oxford, 20-24 March 1995.