INTEGRATED
ENGINEERING SOFTWARE
www.integratedsoft.com

Electromagnetic Sensor Design:

Key Considerations when selecting CAE Software

Content

Executive Summary

Though sensors based on electromagnetic principles often appear to be simple devices, they can pose challenging analysis problems. This paper will provide guidelines to aid in the proper selection and application of electromagnetic CAE software for simulating and optimizing sensor systems. Example electric, magnetic and eddy current simulations will be used for illustration. In particular, this paper will discuss the approximations and simplifications that are often required to produce practical engineering models.

Characteristics of Electromagnetic Sensor Systems

Electric and magnetic fields can not only propagate through empty space, they can also penetrate various materials. Because of this, electromagnetic sensors are often the ideal choice for situations where direct contact is not possible or not desirable.

However, this “action at a distance” property can also result in a more difficult simulation model than is encountered in more common types of analysis (such as mechanical stress analysis). Problems occur because the model space must be large enough to include all objects which would have a measurable effect on the sensor system.

The designer must therefore have insight into all factors that affect the measurement, and their relative importance. In many cases secondary features must be omitted in order to produce a model which is feasible for solution in a reasonable amount of time.

The remainder of this paper will discuss the factors that influence the choice of appropriate software, and the techniques for generating CAE models.

Basic Selection Criteria

The analysis strategy and resulting choice of software are most heavily influenced by the following factors:

- The type of physical interaction employed in the sensor system. This needs to be determined at the earliest stages of the design.

- The nature of model geometry. Many sensor applications require a full **3D** analysis, but in some cases **2D** or **Rotational Symmetric** (also called **Axisymmetric**) software can be used. In addition, the aspect ratio of objects to be modeled and/or the size of the model space required may be a factor when deciding on the choice of solver method.

- The choice of solver method. Most commonly used for sensor applications are **Finite Element Method (FEM)** and **Boundary Element Method (BEM)** field solvers. (**Finite Difference Time Domain** and **Method of Moments** solvers may be needed in High Frequency applications, but these will not be discussed in this paper.)

We will discuss each of the above points in the following sections.

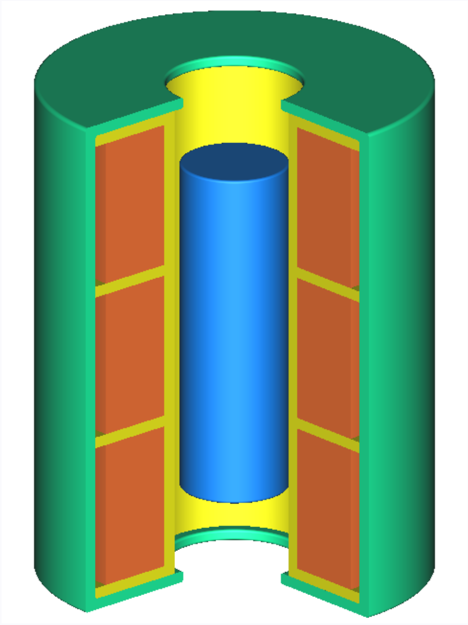
Physics Considerations

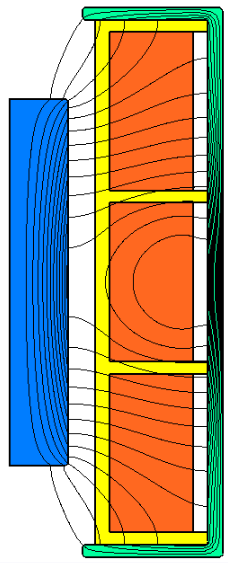
Sensor designers can choose a variety of methods depending on the quantity to be measured. For example, position can be measured using variation in capacitance, variation in flux produced by proximity to a permanent magnet, impedance variation of a coil inducing eddy currents, or inductive coupling of coils to list just four examples.

Obviously simulating a capacitive sensor will require an electric field solver, and simulating flux produced by a permanent magnet will require a magnetic field solver, while the impedance sensor will require an eddy current solver. The inductive coupling sensor may require only a magnetic field solution, or it may require an eddy current solution depending on the physical construction and/or frequency of operation of the sensor. We can illustrate this using the example of a Linear Variable Differential Transformer (LVDT).

The picture at right shows a simplified model of an LVDT. Here we have cut away the outer housing (colored green), coils (colored orange) and bobbin (colored yellow) to show the movable magnetic core (colored blue).

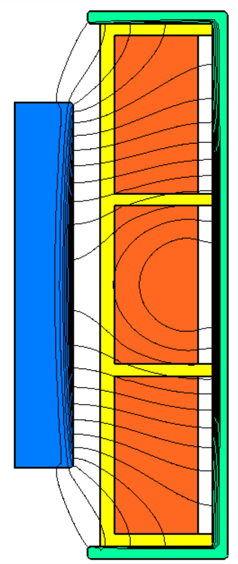
The middle coil is the primary winding of the transformer which is driven by an AC source in order to create a field through the core and housing that couples with the two secondary coils. The two secondary coils are wound with opposite polarities and connected in series. As a result the induced voltages across the secondary coils will cancel when the core is centered. Displacing the core up or down from center will create a differential voltage proportional to the displacement and the direction of displacement will be indicated by the phase shift relative to the primary.





If the core and housing have a low electrical conductivity and/or the operating frequency is low, eddy currents can be ignored and the LVDT can be simulated using a simple magnetic analysis. The field pattern at left shows this situation.

However, for greater conductivities and/or higher frequencies, induced eddy currents will alter the field pattern and act as an additional load on the primary circuit. The field pattern shown at right includes the effects of induced eddy currents. Note how the field lines are now concentrated near the boundaries of the core and housing.



Occasionally, permanent magnet systems may involve linear or rotational motion which can in itself induce eddy currents in conductive bodies. Some eddy current software packages can approximate this by assigning constant velocities and performing a steady state solution. However, if time varying current sources are also present (including single frequency AC), a full transient simulation will usually be required.

The key point from the above examples is that a thorough understanding of the physics to be modeled is the primary requirement that must be established at the start of the design project.

Geometry Considerations

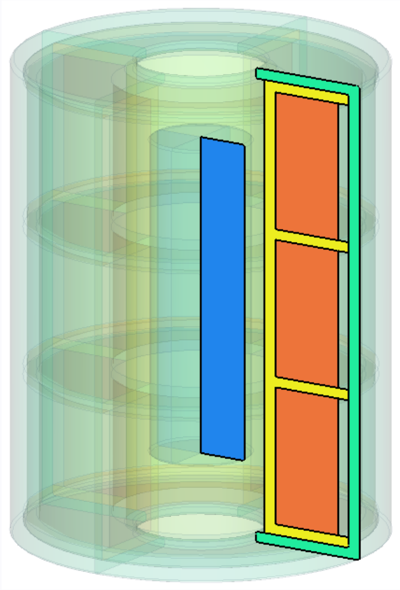
Electromagnetic CAE software can be classified as either fully **3D** or **2D**. Most **2D** software can be set to solve field equations in their cylindrical coordinate formulations. This cylindrical coordinate mode is referred to as **Rotational Symmetric** (the term we will use in this paper) or **Axisymmetric**. We will use the abbreviation **RS** to indicate a Rotational Symmetric solution.

Though **2D** CAE analysis predominates in the areas of motor and generator design, **RS** analysis has more applications in the field of sensors.

The LVDT model used in the previous section is a classic example of an **RS** problem. The picture at right shows that all the components of the LVDT are solids of revolution formed by extruding surfaces in a circular path around a common axis.

Note that both conditions are required. If all components of a model possess cylindrical symmetry, but they do not share the same axis, then the **RS** formulation will not apply.

In the LVDT example, even though the core moves during operation, the movement is constrained along the axis so the **RS** formulation will always be valid.

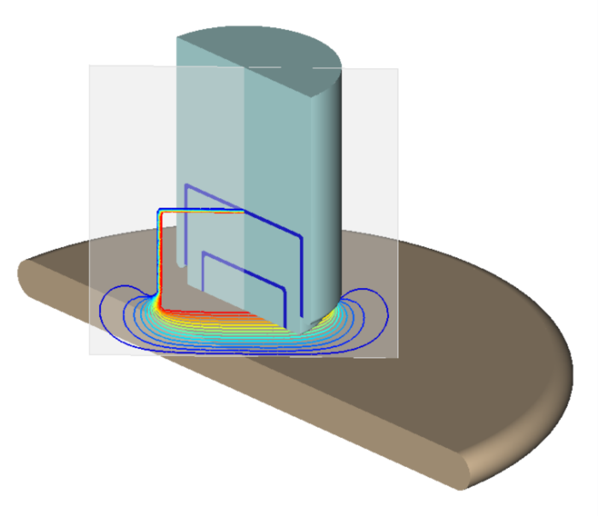


When the **RS** formulation can be applied, it has three significant advantages:

- It will produce ***exactly*** the same results as a full **3D** simulation.
- The **RS** models will be easier to build and modify since all the geometric objects are on a 2D plane.
- **RS** models will solve significantly faster than full **3D** models. This is particularly important when trying to optimize models, and/or in transient simulations since both of these situations require multiple model solutions.

The **RS** formulation is also an excellent approximation in many cases where only the sensor itself has cylindrical symmetry.

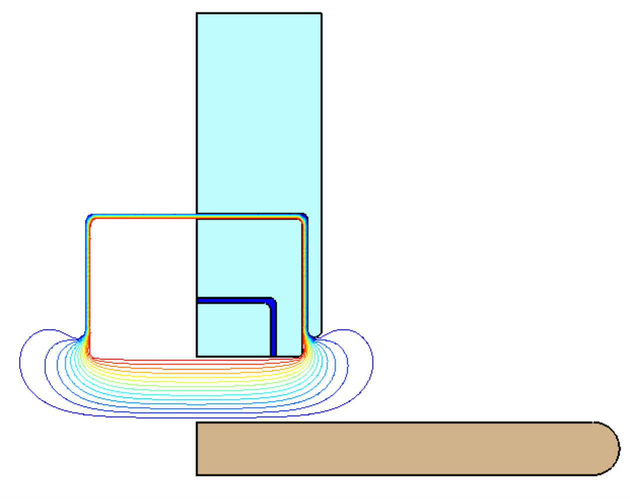
At right we show a half section view of a capacitive sensor above a disk shaped target. In this case the target also possesses cylindrical symmetry, but this is not a strict requirement. Note that only a small part of the target surface interacts with the sensor. Because of this, we can replace a non-symmetric target with one that has cylindrical symmetry provided the target is large enough, and the target surface is parallel to the sensor element.



At right we show the resulting **RS** model of the capacitive sensor.

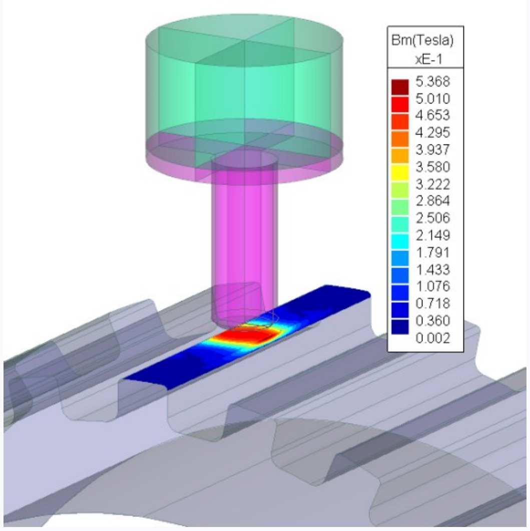
Note that the **RS** formulation cannot account for the effects of tilting the disk so that it is no longer parallel with the sensor element. (However in a later section we will show how planar symmetry conditions can be used for this situation.)

As another example, the **RS** formulation would not strictly apply if the target were a cylinder with axis perpendicular to the sensor element.



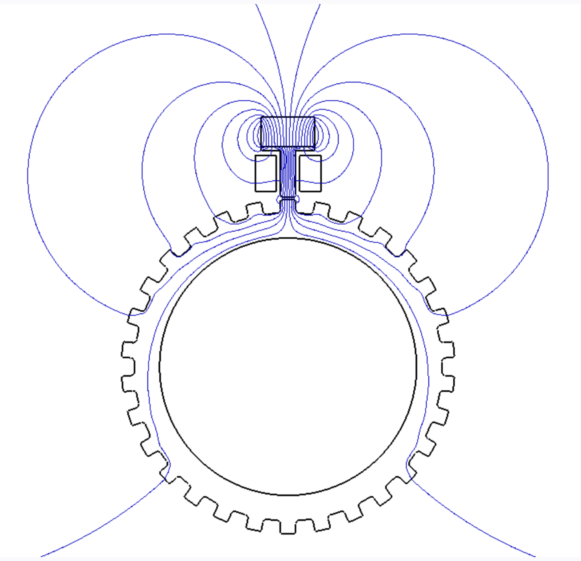
The key components of Magnetic and Eddy Current sensors are also often cylindrical and can be simulated using **RS** models depending on the characteristics of the target.

The picture at right shows an example where the **RS** formulation will ***not*** be appropriate. Here a permanent magnet (colored green) is attached to a pole piece (colored magenta) and is positioned above a rotating gear (colored blue-grey). This is a common arrangement used in variable reluctance and Hall type sensor systems. Though the sensor element possesses cylindrical symmetry, the gear target does not.



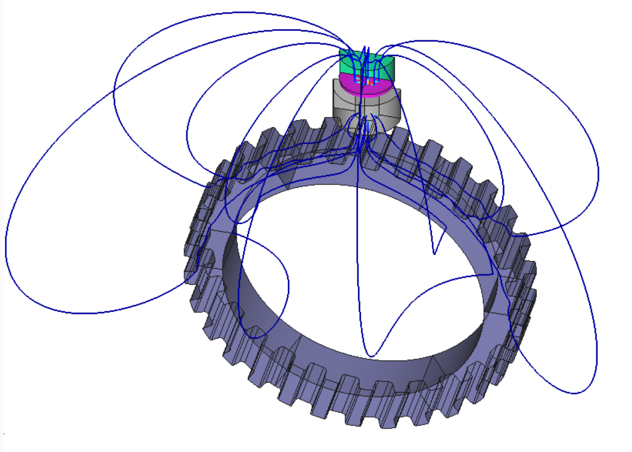
It might seem that some approximate results could be obtained using a **2D** (instead of **RS**) formulation as shown in the picture at right. There are two problems which limit the usefulness of this approach.

First, the **2D** field solution calculates results on a per unit length basis (usually per meter). These results can then be scaled by the length in the third dimension to predict the sensor performance. However, it is not clear what length should be used to scale the field results. For a **2D** formulation to be valid the magnet, pole piece and gear should ideally all be the same length which is clearly not the case.



The second and perhaps more serious problem is that the **2D** formulation imposes restrictions on the possible flux paths that are not realistic for the actual system.

The picture at right shows a partial stream line plot that illustrates flux paths that are impossible to model using **2D** software.



The examples shown in this section illustrate that the choice of analysis software is often more heavily influenced by the geometry and behavior of the target than the geometry of the sensor itself.

In some cases it may be desirable to have both **2D/RS** as well as full **3D** software. Though in theory only **3D** software is required, the speed and ease of use of **2D/RS** software can expedite the initial design phase and may be well worth the additional cost.

Solver Considerations

Electromagnetic sensors function by transforming a quantity to be measured into outputs which are some form of signal (voltage or current), or some type of circuit parameter (which is usually measured by its impedance). Electromagnetic CAE software simulates the sensor outputs by computing the relevant field solution for the sensor system. The process of obtaining an accurate field solution is therefore the key step to accurately simulating a sensor system.

The most common field solvers used for sensor applications are based on either the **Finite Element Method (FEM)** or the **Boundary Element Method (BEM)**. Both methods essentially convert the problem of solving the partial differential field equations into the numerical analysis problem of solving large systems of linear equations. Both methods discretize models of physical systems by creating meshes of geometric elements (typically 2D triangles and/or 3D tetrahedra). However, the two methods have fundamental differences in the types of unknowns that are solved for, and in the type of meshing required.

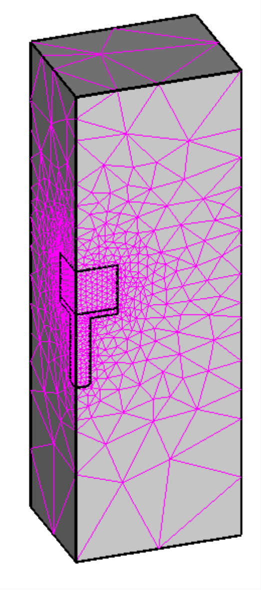
FEM is the older of the two methods and was originally developed to solve structural analysis problems. As applied to electromagnetics, **FEM** formulates a system of linear equations that solves for a potential function, and the field solution is then obtained through a process of numerical differentiation.

BEM uses the approach of solving for equivalent sources (such as charges or currents) and obtains the field solution through a process of numerical integration.

We can use the permanent magnet and pole piece from our previous gear sensor example to illustrate the differences in mesh requirements for the two methods. For simplicity we will omit the gear and model only the magnet and poled piece.

The pictures at right shows a quarter section model of the permanent magnet and pole piece solved using **FEM**. The first consequence of using a **FEM** solver is that some sort of artificial boundary must be created to limit the solution space. Here a rectangular box has been created around the sensor, and the **FEM** formulation forces the field to be zero outside of the box. The size of the box can be specified by the user, and care must be taken that it is sufficiently large so as not to impose an unnatural limitation of the field solution.

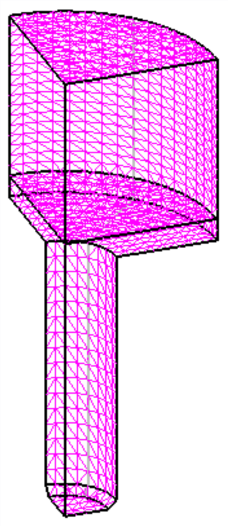
The second consequence is that a continuous mesh must be created throughout the box even in empty space regions.



By way of contrast, the picture at right shows the magnet and pole piece solved using **BEM**. Note that **BEM** does not require the creation of a boundary box, or meshing of empty space regions.

As a general rule, the more mesh elements required, the larger the resulting system of linear equations. This would appear to give **BEM** the clear advantage over **FEM**, and this is generally the case for models which have large regions of empty space. Counteracting this is that **BEM** tends to produce a "dense" matrix while **FEM** produces a "sparse" matrix which may lead to a faster numerical solution.

Another factor to be considered is that the time to create and refine the element mesh may in itself constitute a significant portion of the total solution time. Here again **BEM** may have the advantage particularly if the model contains a number of objects that are separated by small gaps, since the **FEM** formulation requires continuous meshing throughout the model space.



Because of the large variety of sensor applications, it is usually not possible to determine in advance which method will be superior for a particular problem. It may even be desirable to use ***both*** types of solvers in order to provide confirmation of results using entirely different formulations. Fortunately, most vendors provide some sort of trial evaluation which is especially useful when actual test data from production models is available for comparison with simulation results.

Some Brief Comments on Meshing

In the previous section we mentioned that numerical field solvers must discretize the model space into meshes of elements and then use these elements to construct a system of linear equations. The construction of a proper element mesh is therefore vital to obtaining an accurate field solution.

As a general rule the greater the number of elements the more accurate the solution, though usually fewer elements are required where the field gradient is small (a more or less constant field) and/or the field is weak. The naïve approach of creating a dense mesh everywhere in the model space is not only unnecessary, it is inefficient as it results in extremely long solution times.

A further complication is that it is usually not possible to predict in advance exactly where the most elements are required, or how many will be required for an accurate solution. An experienced user can often make an excellent first guess, but usually a trial solution followed by refinement of the initial mesh is desirable to ensure accuracy requirements are met. If the solution with the refined mesh produces results that differ greatly from the initial mesh results, additional refinement steps may be needed.

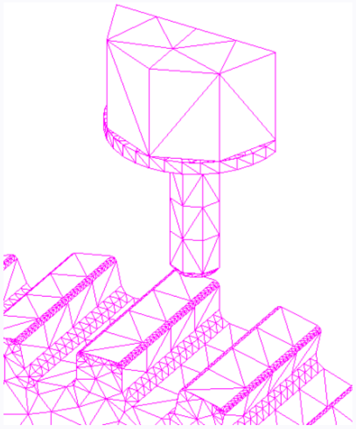
We've also mentioned that sensor design projects typically involve a large number of trial solutions. This provides a further incentive to reduce solution times where possible, and it also suggests that some form of automatic meshing would be desirable for the CAE software selected.

Some CAE software packages employ self-adaptive meshing algorithms. These algorithms automatically refine the mesh using results obtained from a series of trial field solutions. This is particularly desirable for sensor systems which contain moving components since different meshes will be required over the range of operating positions.

We can illustrate how this approach works in practice using our gear sensor example. We will use models solved using **BEM**.

At right we show the magnet, pole piece and gear teeth with a coarse initial mesh.

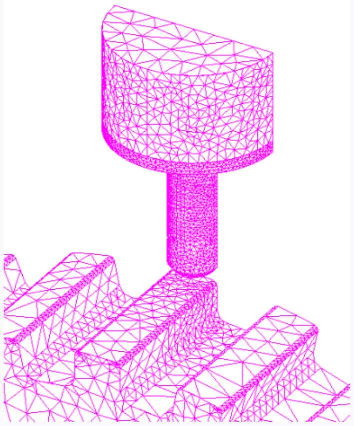
In this case all of the gear teeth have the same type of mesh even though it is obvious that the gear tooth directly aligned with the pole piece will experience the highest field strength.



Next we show the system after several refinement steps.

The magnet and pole piece are now densely meshed which is appropriate since the magnetic flux density will be greatest in these components.

But in particular, note that the gear tooth in alignment with the pole piece is now more densely meshed than the other teeth. Also the mesh is not uniform over the surface of the tooth, but instead is concentrated in the region near the tip of the pole piece.



The meshing capabilities of a CAE field solver program can be a key factor in determining its suitability for a particular sensor application. Here again, a trial evaluation may be the best way to decide among competing alternatives.

Reducing, Simplifying and Approximating Models for Faster Solutions

The advent of parallel processing to utilize multi-core processors, combined with 64-bit operating systems and ever more affordable RAM has dramatically increased the speed of PC based CAE simulation software. Even so, most design projects typically entail the solution of a large number of models – often 3D models – which provides incentive to reduce solution times where possible.

The three most common methods to reduce solution times are:

- Exploiting patterns that naturally occur in the physical system to reduce the model size using symmetry and/or periodicity conditions. When these apply, the reduced model solution will still provide all the information of a full model.

- Simplifying the model by eliminating features of secondary importance and/or truncating the model space. This involves a tradeoff between speed and accuracy, since the simplified model cannot provide an exact solution.

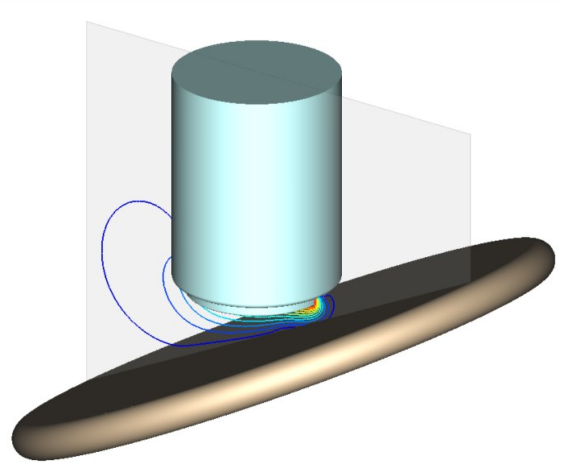
- Approximating **3D** features as being **2D** or **RS**. Again, there will be a tradeoff between speed and accuracy, but in some cases the full **3D** model may be so complex that it exceeds the capacity of available computer resources.

We will illustrate each of these approaches in the following sections.

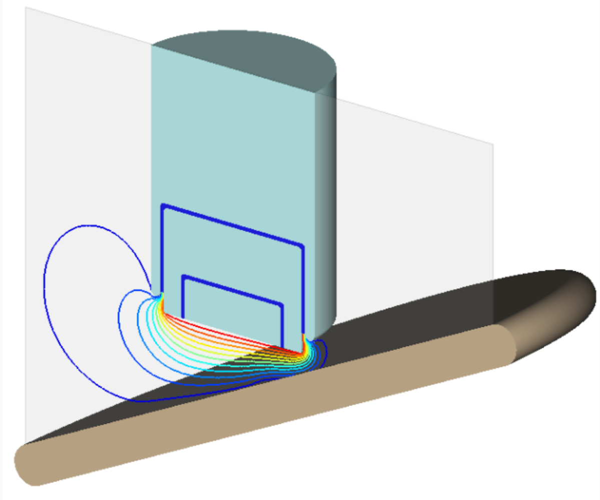
Reducing Model Size using Symmetry and Periodicity

At right we show a full model view of the capacitive sensor we had previously used as one of our examples for **RS** simulations. In this case the disk has been tilted so that the **RS** formulation would no longer apply

Though this situation can no longer be solved as **RS**, it is possible to reduce the size of the model (and reduce solution time) by noting that there is still symmetry across a plane.

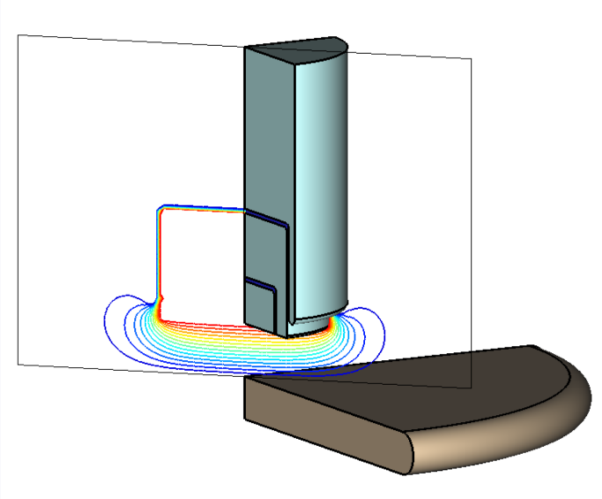


At right we show the resulting model that has been reduced by half using the symmetry condition. Note that we have not sacrificed any accuracy by this method, but the solution time will be greatly reduced.



If the disk and sensor are maintained parallel and centered, even further reductions are possible. The picture at right shows a one quarter model. Here we can use either two symmetry planes, or we can set the model as angular periodic with four sections in the full model.

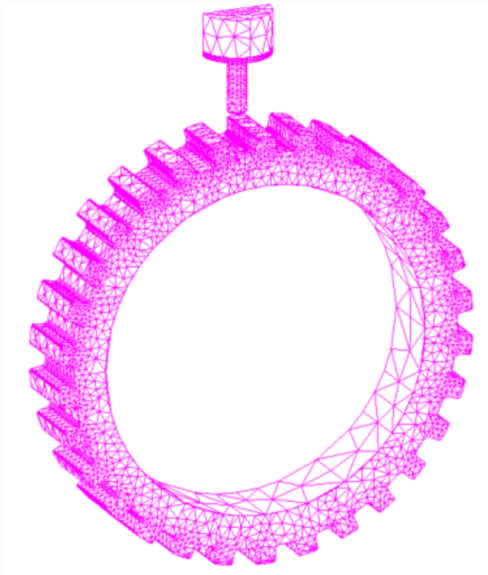
Because of the uniformity of the sensor and disk, we could use the angular periodic method to reduce the model even further, but at some point we could begin to lose accuracy due to badly shaped elements.



Simplifying Models by Eliminating Secondary Features

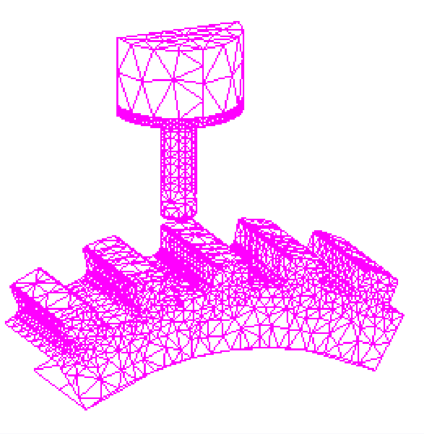
The picture at right shows a permanent magnet sensor aligned with the tooth of a gear. Note that a symmetry plane condition has been used so only half of the gear and sensor have been modeled.

Because the flux density is greater for the parts of the gear close to the sensor we can reduce the model by omitting sections of the gear that are far from the sensor.



The next picture shows a reduced model where only one sixth of the gear is simulated. This model solves considerably faster, but still produces results which are within 5% of the full gear model.

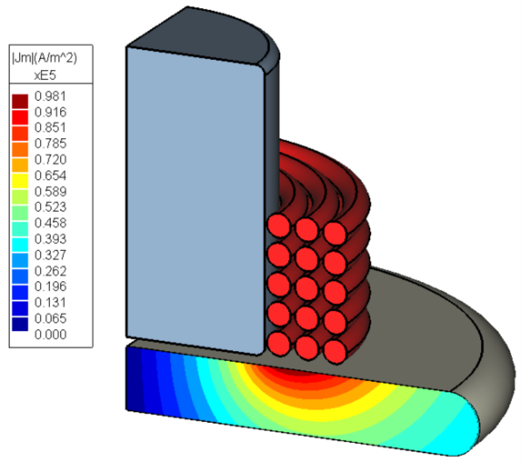
Typically this type of sensor requires solutions at a number of angular positions to model the full variation of flux as the gear rotates. This is compounded when several sensor variations are simulated in order to optimize performance.



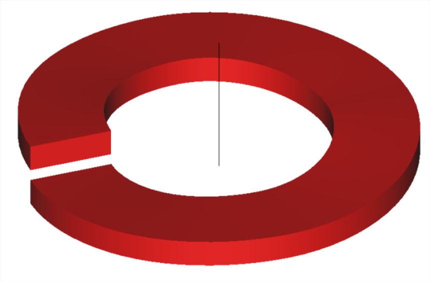
Approximating 3D feature as RS or 2D

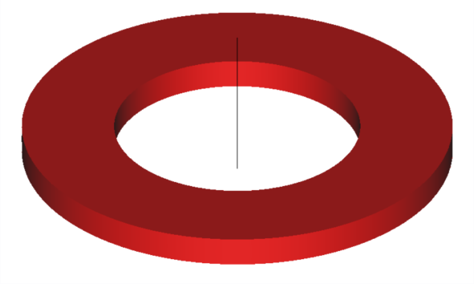
The eddy current sensor shown at right consists of an AC coil wound around a cylindrical ferrite pole piece (here we show a quarter model). The coil current induces eddy currents in the conductive target (shown by the current density contour plot). The coil impedance will change depending on the gap between the sensor and target, or the presence of flaws in the target.

Depending on the operating frequency, the impedance of the coil may be greatly affected by skin effects and proximity effects. Simulating these factors requires modeling all of the individual coil turns, and this may not be feasible in **3D** if there are a large number of turns. Fortunately it is often possible to use **2D** and **RS** approximations. We will illustrate this first with an **RS** example.

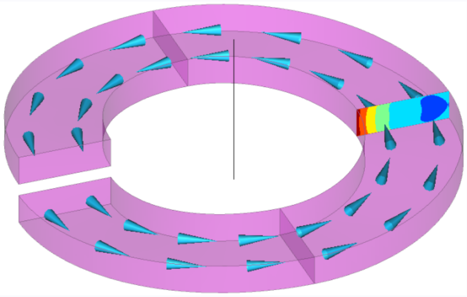


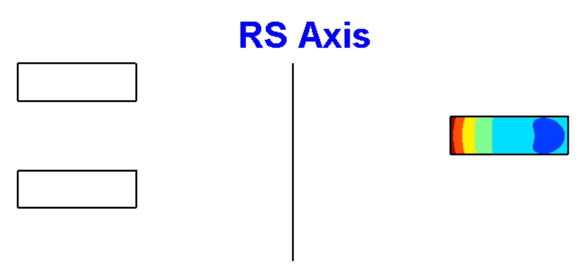
Below left we show a single turn of a helical coil which has a rectangular cross section. At right we show the single turn approximated as a toroid. While a helix requires a full **3D** model, a toroid is a solid of revolution and can be modeled exactly using an **RS** formulation.



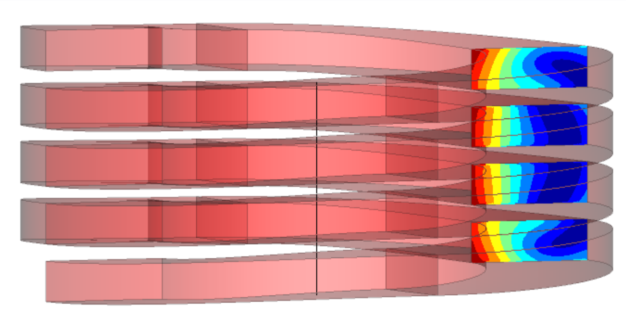


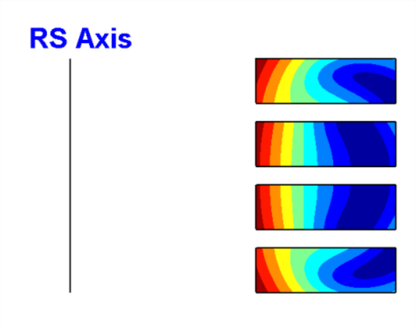
This is in fact a very good approximation in practice. Below left we show the current flow and current density in the **3D** helical turn compared to the **RS** toroid approximation shown at right. In the **RS** model we show the start and end surfaces of the helix solely for visualization purposes; they are not part of the **RS** model.



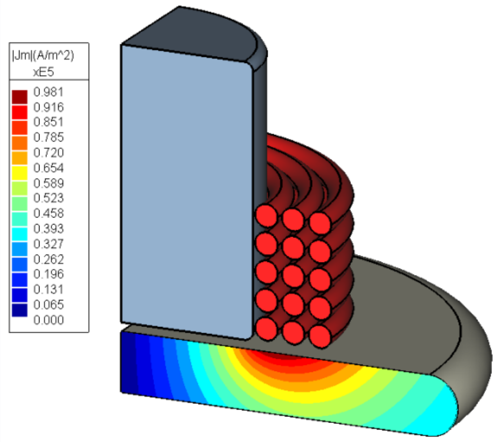


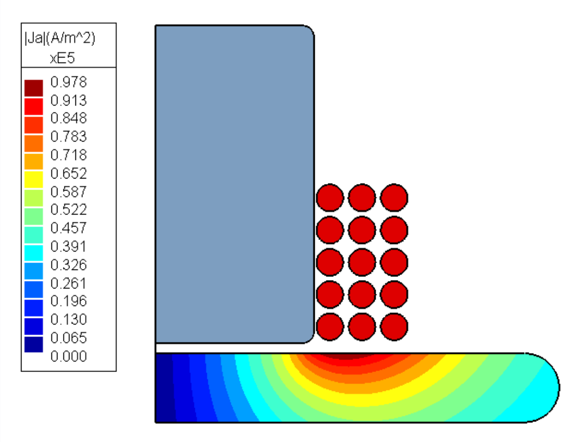
Similarly, a multi-turn helical coil would be approximated as a stack of concentric toroids. In the pictures below we show the **3D** model of a 4-turn coil and its equivalent **RS** representation.



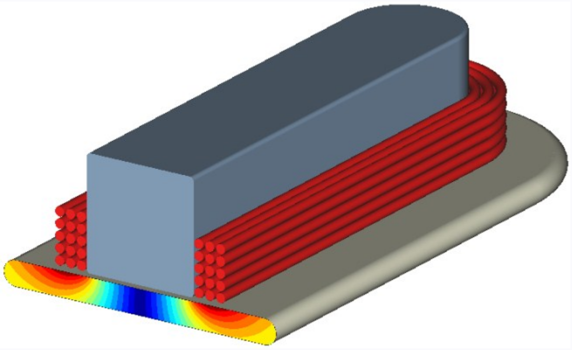


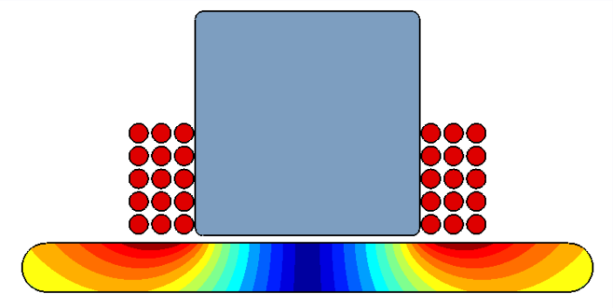
Below left we again show the original **3D** sensor, and at right the equivalent **RS** model.





For models which are long in one dimension a **2D** approximation would be more appropriate. The **3D** and **2D** equivalent models for this situation are shown below.





The coil examples we have shown have relatively few turns, and could be modeled directly in **3D** software, though simulation times would become significant if multiple solutions for different gaps were required (as would usually be the case). Typical sensors may have windings consisting of hundreds of turns, which would at best lead to extremely long solution times, or at worst might even exceed available computer capacity.

Model Performance Testing and Design Optimization

Having selected the appropriate software based on physical considerations, and simplified the model where possible, the sensor designer can proceed to the actual design and optimization phases.

Since sensors are often specified by their maximum and minimum operating ranges, the testing of trial designs will involve at least two model simulations. In practice the variation of the sensor output over its range of operation is usually of interest and this will require multiple solutions for every trial design.

Most software packages provide some type of utility to automatically generate multiple solutions. Depending on the vendor, this functionality may be referred to as batch, parametrics or scripting (or in some cases more than one utility may be provided). For this paper we will use the following terminology:

- **Batch** will refer to the simple solution of a series of models. The models may in fact be completely unrelated.
- **Parametrics** will refer to the solution of a series of models which are variations of a single basic design.
- **Scripting** will refer to the use of an external program to control the operation of the CAE simulation software.

These concepts are somewhat related, and may occur in combinations of each other (for example, a batch file may be set to run a series of parametrics).

Batch runs on their own are of limited usefulness since they can only solve pre-existing models. For example, studying the variation of a sensor performance over its operating range would require manually creating each of the individual models for the batch solution.

Parametrics are often the most useful feature for sensor design problems. If the parametrics can be programmed using a graphical interface they are particularly easy to set up. Depending on the ingenuity of the designer, parametric runs can be programmed to produce multiple design variations, and then run each variation through its range of operation.

Scripting can provide even greater latitude in design variations, however this comes at a cost of greater investment in time involved in creating the initial script. Scripting can provide an automatic method for producing multiple custom designs based on end user requirements. Using our previous gear sensor example, a script could be written to create gears of various sizes and with various numbers of teeth; something which could not be done in parametrics. Scripting can also allow external optimization routines to control the CAE analysis software.

Using these tools, optimum designs can be achieved much faster than would be possible by building and testing actual physical prototypes. However, here again the designer must have a complete grasp on the factors affecting the performance of the sensor system. Without this understanding, the optimization process will deteriorate into a brute force random search where all possible parameters are varied in hope of eventually finding the best design.

Summary

The selection and application of CAE software for sensor design projects depends on a thorough understanding of the physics and expected operating conditions of the sensor system.

When evaluating software alternatives, the following points should be considered:

- A sensor design project will almost always involve a large number of simulations. This is a key factor to keep in mind when selecting software.
- Consider whether a **2D/RS** field solver would be appropriate.
- Look for CAE software that automates the construction and solution of models.
- Remember that proper meshing is a crucial requirement in obtaining accurate solutions.
- Whenever possible, have design engineers test the software against current production models to confirm that they will be able to produce the required results for any new projects.

The last point is often neglected, though ultimately it may be crucial in determining whether or not the selected software will be suitable for its intended use. Basing the final purchase decision on vendor demonstrations alone may lead to unpleasant surprises when the software is turned over to the end users. Ideally, the designers themselves should be able to demonstrate that they are able to use the software to produce accurate simulations.

About INTEGRATED Engineering Software

Since its inception in 1984, INTEGRATED Engineering Software has created simulation tools that reflect the inspiration of our customers: thousands of engineers and scientists who, everyday, push the boundaries to envision what is possible. They take their ideas from a realm that is almost science fiction and bring them to reality.

As the name of our company suggests, all our programs are seamlessly integrated, starting from a concept, through entry of the geometry and physics of the problem, to the selection of type of solver and the problem's solution. Once the problem has been solved, a vast number of parameters can be calculated or the field quantities displayed.

INTEGRATED Engineering Software is a leading developer of hybrid simulation tools for electromagnetic and particle trajectory analysis. We provide a complete line of fully integrated 2 and 3 dimensional simulation software.

Since the creation of our company, our focus has always been here and our experience has grown hand-by-hand with a great recognition in our market.

INTEGRATED is staffed with leading R&D engineers in areas such as electrical engineering, magnetics, and high frequency applications. Our tools are used in a wide variety of industries, including manufacturing, automotive, medical, telecommunications, power, health care and aerospace markets, as well as universities and research laboratories.

INTEGRATED products allow engineers and scientists to reduce design cycles, save time and money and deliver more efficient products to the market faster than ever before.

INTEGRATED empowers engineers and scientists with many options to choose from:
The best solvers for each specific application: Boundary Element Method (BEM), Finite Element Method (FEM) or Finite Difference Time Domain (FDTD) solvers. The best optimization tool for each particular design: parametric analysis, scripting or application programming interface (API)

INTEGRATED’s commitment is to provide designers with the most sophisticated analysis tools to assist them in the creation of the future.

Contact us for an evaluation

Send us your model, whatever the level of complexity. **We will show you how to get results from your exact design** – no canned demos.

Contact us for an evaluation and start improving productivity today. A live demo is also available.

Phone: +1 204 632 5636
Fax: +1 204 633 7780
Email: info@integratedsoft.com
Website: www.integratedsoft.com

INTEGRATED
ENGINEERING SOFTWARE