Structural Analysis: Going Beyond Standard Load Cases

Dr. Ivar KJELBERG[1]
CSEM sa, Rue Jaquet-Droz 1, CH-2002 Switzerland, ivar.kjelberg@csem.ch

Abstract: Having worked with classical FEM tools for years, for structural and thermo-elastic analysis of complex systems, my first experience with COMSOL Multiphysics gave me some frustrations:
What had happened to the gravity loads, torque loads, the rotations, and the rigid body elements (RBE), how to add a punctual mass load to a sub-system in 3D-solid; and so on?
In fact, everything is *in* COMSOL Multiphysics, but it is how-to access these equations that is so different from my previous tools. To keep the global multi-physics approach of COMSOL Multiphysics you will now have to "THINK PHYSICS", to understand again your PDE's and to apply the physics on them; while before, this was all cooked-down by pressing on click-ready check-boxes.
The advantages are clearly that with COMSOL Multiphysics you link in whatever other Physics you want, while this has never been possible in such a way before; with the drawback that you must, with COMSOL Multiphysics, change your way to do things.
If it might feel confusing in the beginning, once understood, applying your Physics becomes rather obvious.
I hope that my presentation will answer some of the questions that you might have too, and that I also often see appearing on the COMSOL Multiphysics Community Forum.

Keywords: Structural analysis, Boundary Conditions, Rotations, Rigid Body Loads.

1. Introduction

As an engineer working at a high-tech industrial R&D company CSEM sa, with many years of structural type modeling experience using traditional FEM tools, my switching to – and learning of – COMSOL Multiphysics took some time and resulted in quite some frustrations. Today, from the FORUM, I see that others are having often similar feelings. I notice too that Structural Physics in COMSOL Multiphysics is still somewhat poorly equipped with cooked-up engineering boundary condition. We are used to rapid “clickable” engineering, from older FEM tools; COMSOL Multiphysics is proposing “clickable Physics” but we are not at the same level. In fact everything is in there. As an engineer I’m requested to propose very rapidly several models to map a solution domain of a new design, the time to write out and to verify long equations is simply not available.

Furthermore, as more and more engineers will adopt COMSOL Multiphysics, because of its unique access to the multiple physics under the same environment, this lack of immediate productivity in their traditional field of structural physics have a negative impact on the selling process to our managers, which is a pity.

The present paper summarizes some of my encountered “hard-points”, and concepts that I was not taught by the COMSOL Multiphysics documentation, nor in their otherwise excellent training courses. Thereafter, I give some examples of load cases I use regularly for my modeling, and I end with a few suggestions for further “nice-to-have” features. From my many and interesting discussions with "support", I understand that structural physics is on their priority list, several new features have come with v4, such as the Rigid Connector, and the eigenfrequency normalization for mass participation factors. I’m looking forward to the newer versions to come, with even more functionalities.

2. COMSOL versus older FEM tools

For those of you with a long experience with more classical FEM tools, doing engineering analysis, you might have some surprises in the beginning when approaching COMSOL Multiphysics. The environment is close to classical tools, but still different in subtle ways. In short, you are used to a three step approach:

- Import a well behaved CAD model
- Define the mesh, and its related material properties, and boundary conditions on the mesh nodes
- Solve and post-process

But you will soon understand that COMSOL Multiphysics have some extra steps.

2.1 Objects and Entities

COMSOL Multiphysics has added a major step on the geometry defining the "analyzed" FEM geometrical ***Entities*** (my explanations hereafter might differ slightly from COMSOL intentions, but here is how I understand and navigate with these items. Basically, this means that many of us need a little brain wash and a restart to work efficiently with COMSOL Multiphysics).

It is onto these highest level ***Entities*** (and not onto the mesh) that one link the material properties, the physics, and the shape functions. All ***Boundary Conditions*** (BC) are linked to the ***Entity Boundaries*** (second level entities). The meshing operation comes only thereafter, followed by the solving and post-processing.

This extra step on the geometry allows separating the meshing from the rest of the model buildup and eases any changes to – and adaption of – the meshing process.

Furthermore, the geometry does not need to be as well behaved as for other FEM tools. It remains though essential to understand the geometrical analysis method of COMSOL Multiphysics while it is transforming ***Geometrical Objects*** (i.e. CAD points, lines, surfaces or faces, volumes; and pre-defined shapes such as spheres, blocks, etc.; or any CAD type assembly of such objects called here ***Composite Objects***) into the analyzed ***Entities***.

The ***Entities*** are named: "points", "edges" "boundaries" and "domains" for respectively 0, 1, 2, and 3D dimensional levels. ***Entities*** are unique, ordered, and identified or numbered. Isolated geometrical objects not belonging to the highest level dimensions are mostly treated as orphans and will not appear in the *Entity List*.

Finally, the geometry analysis process has two ways to generate ***Entities***: i) via ***Union*** or ii) ***Assembly*** (the latter has nothing to do with a classical CAD assembly).

Entities defined via a ***Union*** are the most common and means that i) any overlapping highest order ***Objects*** are first split to create separate non-overlapping ***Domains,*** ii) any common 2nd highest ***Object*** surfaces are split and made into unique ***Boundaries*** between two adjacent ***Domains,*** and iii) a ***Continuity*** boundary condition is assumed by default on all common boundaries to adjacent domains.

Entities defined via an ***Assembly*** are there for special cases when continuity is **not** wanted by default on the physics boundary conditions. This is mostly for contact physics or when one wants to define user specific physics between adjacent boundaries. To avoid forcing all boundaries to be doubled, one should use the ***Composite Objects*** to "assemble" different objects to be treated together as for a ***Union***. The ***Assembly*** process will then be performed only between the Composite objects, and not for all contained ***Geometrical Objects***.

There are a few further transformation performed during this geometrical analyzing process that are to be studied and tested, and one should take care to understand the difference of treatment of "highest" level items (3D ***Objects*** and ***Domains*** in a 3D geometry, and the 2nd level ***Objects*** and their corresponding ***Boundaries***.

This approach can also lead to some surprises, if not correctly understood, i.e. the integrated total surface of a set of ***Domains*** in 2D, created via a ***Union*** or an ***Assembly***, might differ significantly if you have overlapping highest level ***Objects***. Hence, erroneous user results are easily obtained if such surface integrations are used in the post-processing.

The implications of the use of a COMSOL Multiphysics ***Assembly*** operation are that the user is responsible for the definition of all ***Identity*** or ***Contact Pairs*** to be defined in the ***Definitions*** node, as well as to add the required physics to link these pairs.

The consequence of the existing of this extra ***Entity*** creation process is that COMSOL Multiphysics applies all physics and boundary conditions to geometrical items, hence isolating them from the mesh operations that get its properties by inheritance. All this allows a far more flexible model processing and changes.

There are still further important issues such as to understanding the use of the different ***Frames*** of COMSOL Multiphysics, the v4.1 documentation is treating this rather well, and I will only remind here that when defining ***Operators*** in the ***Definition node*** you must ensure that you use the correct ***Frame***, but for Postprocessing integrations operations the ***Frame*** is by default defined in the ***Data Set - Solution*** node (higher up in the model tree). Forgetting this might too lead to quite unexpected results!

Frames are important for structural physics too, as in v4, the ***Spatial Frame*** is defined and driven by the deformations, and by default; this was not so in v3.5a.

2.2 What about boundary loads in [Pa] and domain loads in [N/m^3]?

Another point that puzzled me in the beginning with COMSOL Multiphysics was that, as an engineer, I define my models properties based on the specification sheets of the components I use, i.e. my actuators provide forces in Newton or moments in [N*m], and the Earth gravity acceleration is 9.81 [m/s^2]. The links from these values to the BC entries do not seem obvious. This is related to the (many) implicit assumptions in the GUI field notations of COMSOL Multiphysics, and the different level considered for physics definitions: physics equations are defined on finite geometrical domains, but most equation fields are entered as defined for the elementarily mesh element.

We may start with, Newton's law:

Fb = m*a

A load is defined at our macroscopic object level as a force **Fb** in Newton's acting on the mass via the acceleration, but FEM deals with small "finite" volumes (mesh elements) that adds up to form the domains, and COMSOL Multiphysics wants the BC's to be defined at the FEM element level, not in our macroscopic world level.

The main reason is that a unified physical language is needed to allow for multiphysics equation linking.

Taking a model of a few domains (assume 3D and union mode, to simplify) each domain "**_i**" has a specific material property, hence a density **rho_i**, and each domain is subdivided into many FEM mesh elements of size **dx dy dz**. Integrating the volume over these mesh elements we obtain:

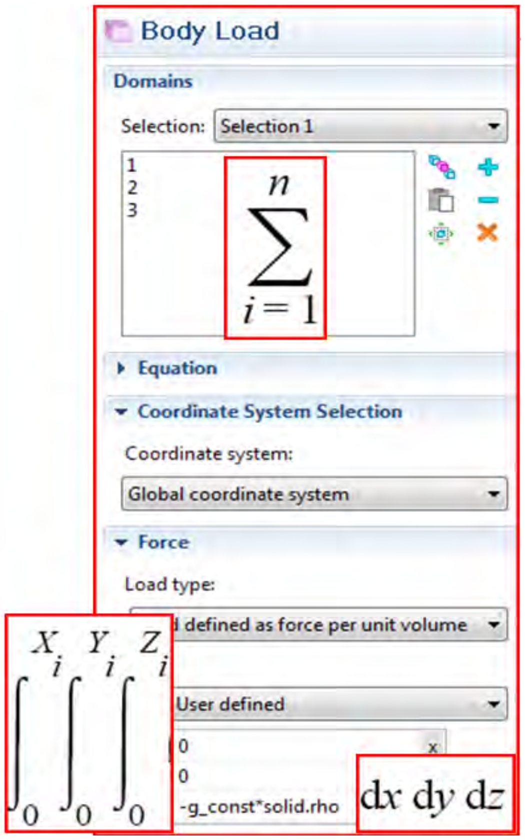
$$Fb = m\,G = \rho\,V\,G$$

$$Fb := \sum_{i=1}^{n}\left(\rho_i\,G\int_0^{X_i}\int_0^{Y_i}\int_0^{Z_i} 1\,\mathrm{d}x\,\mathrm{d}y\,\mathrm{d}z\right)$$

Figure 1. Newton's law for a body force **Fb** and a gravity acceleration **G** = 9.81[m/s^2] for a volume **V**.

But we fill into the GUI field for a structural ***Body Load*** only the "***solid.rho*g_const***" value corresponding to the central part "**ρ_i G**" of the equation above.

The sum over "**i**" as well as the triple integration over the 3D domain is implicit for COMSOL Multiphysics; this is not always well understood or simply forgotten by the new users.



Body Load

Domains

Selection: Selection 1

1
2
3

$$\sum_{i=1}^{n}$$

Equation

Coordinate System Selection

Coordinate system:

Global coordinate system

Force

Load type:

defined as force per unit volume

User defined

0
0
-g_const*solid.rho

x

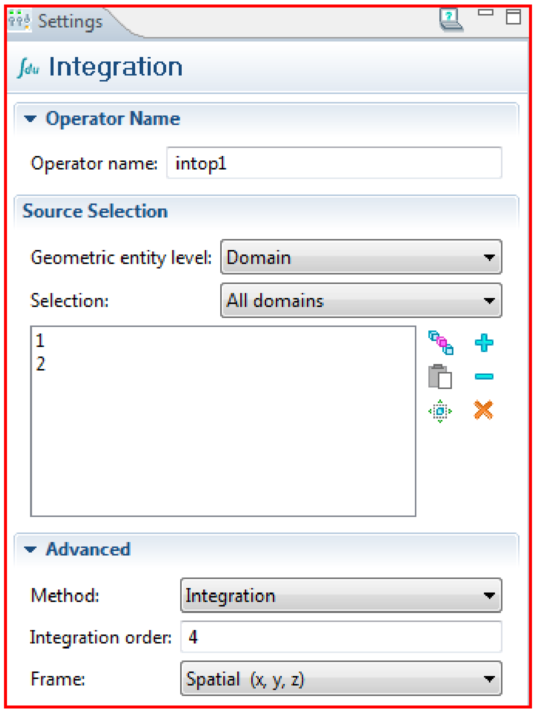
$$\int_0^{X_i}\int_0^{Y_i}\int_0^{Z_i}$$

$$\mathrm{d}x\,\mathrm{d}y\,\mathrm{d}z$$

Figure 2. The sum over "**i**" for the selected domains, and the implicit triple integration for a structural Body Load on a 2D model, note the implicit "**_i**" for rho.

Again not understanding these differences might lead to a cryptical error messages such as for the following case:

Two domains defined by two different material properties are used to get the total volume and total mass via a domain integration operator ***intop1()*** defined at ***Definitions*** node level. The variable **Vol** = ***intop1(1)*** is simply the volume, the variable **MM** = ***intop1(solid.rho)*** results to the total mass of both domains, while **Mm** = ***intop1(1)*solid.rho*** triggers an error message stating that it failed to "evaluate the variable ***mod1.solid.rho*** – Global scope". This is to be understood as that ***solid.rho*** is defined globally, but is not uniquely defined; therefore it must appear inside the operator to be correctly summed up over the domains, the implicit "_i" is not apparent!



Operator name: intop1

Geometric entity level: Domain

Selection: All domains

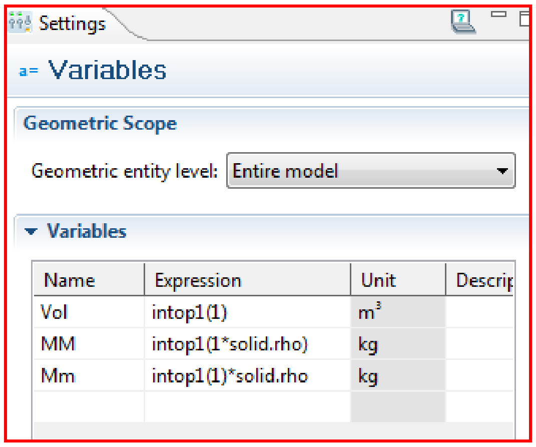
1
2

Method: Integration

Integration order: 4

Frame: Spatial (x, y, z)

Figure 3. Definition of an integration operator ***intop1*** for two domains with different materials Note the spatial frame definition at this node level.



Geometric entity level: Entire model

Name	Expression	Unit	Descrip
Vol	intop1(1)	m^3	
MM	intop1(1*solid.rho)	kg	
Mm	intop1(1)*solid.rho	kg	

Figure 4. Corresponding variables; "**Mm**" is wrong, **solid.rho** has an implicit "_i" that is not resolved.

A last comment on 3D and lower order models: COMSOL always calculates in 3D, it considers a 2D model as a simplified 3D model where the depth "*z*" (the out-of-paper direction) has a default thickness (***solid.d)*** of 1[m] (some physics consider other default values, be aware!) and simplifies the physics accordingly. While for 2D-axi the thickness "z" is replaced by the "loop length" ***2*pi*r*** (here too, there is a normalization change between v3.5a and v4, be aware!).

3. Other structural load cases

3.1 Rotary acceleration load

The Body Load might not only come from gravity, but could also be generated by a rotational velocity, this case is rather obvious and the corresponding body force density acting on each elements can be simply expressed by the acceleration force **Fr = ρ*ω^2*r** where **ρ** is the density, **ω** the angular frequency and **r** the radius measured from the axis of rotation, either via a ***Definitions*** user ***Coordinate System*** or via a variable such as **r = sqrt((x-x0)^2 + (y-y0)^2).**

3.3 Moment load on 3D solid model

Applying a moment to a shaft can be achieved in several manners. The new **Rigid Boundary Connector** (RBC) is ideal for such load cases. But, the boundary you define the RBC onto becomes as said "rigid" and this is not always desirable. If the moment applied is known you might distribute a **Mz[N*m]** load on a boundary with a **Boundary Load** condition as for the following 2D example applying to the **end area** of a cylinder defined in the x-y plane:

Fx[N] = - 2*(*y*-Y0) / R1*Mz / R1
Fy[N] = + 2*(x-X0) / R1*Mz / R1

Where **(X0, Y0)** is a point defining the axis and **R1[m]** is the full cylinder radius. Again by implicitly integrating these force values over the circular section we find back the total moment **Mz[N*m]**.

However, if the moment is applied to the rim (to the edge or to a certain height along the cylinder edge) then the factor "**2**" above should be dropped!

Another case is: you know *where* to apply a force or a moment load, but *not* its final value; however you know the resulting moment **MzLoad[N*m]** you want to see at a given interface location. If this location has a ***Fixed BC*** you can use the weak expression and the resulting Lagrange Multipliers ***lm*** to precisely extract the moment seen and have COMSOL multiphysics doing a means square optimization of the load for you to match the desired output value. The load case can be expressed as above; the resulting measured moment **Mzm[N*m]** on the fixed boundary is obtained by integrating the **lm[Pa]**'s as following (3D, z-axis moment):

Mzm = *intop1(u_lm*y - v_lm*x)*

Less precise results will be achieved by using the surface traction forces ***solid.Tax[Pa]*** instead of ***u_lm[Pa]***.

Remains to define a ***Global Equation*** in the solid physics to allow COMSOL Multiphysics to generate and to control the load moment **Mz[N*m]** (as used for the load case above) and defined by the equation **Mzm – MzLoad = 0 .**

3.4 Moment applied for a given rotation

Another case encountered is if we do not know the true moment to be applied, but we have the desire to achieve a given rotation. This load case is similar to the previous one. The Lagrange Multipliers are not required but a metrology of the desired angle must be performed: via the curl of the deformation field, or an arctan2(x,y), or even a RBC would do to measure the desired angle. Then the ***Global Equation*** defining the applied Moment **Mz** is obtained by an equation of the type as:

ThetaMeasured - ThetaDesired = 0

3.4 Point load in 2D solid model

Many of my structural models are of the Compliant Structure types, where thin metallic shapes allow for large deformations and flexing of solid bodies. It is not really required to model all the domains, the interest is in the thin flexing items, but only the loads coming from the solid parts are of great importance to the final behavior. Such items are used extensively for opto-mechanical guiding i.e. for precision alignment of interferometers where the resolutions desired are easily below the nanometer scale. Today, COMSOL Multiphysics does not allow "simply" saying:

"Select these ***Domains*** and reduce them to a point mass and inertia at the centre of gravity (CoG) and link rigidly, or softly, to the interface ***Boundaries***."

Such a case must be set up by user-written equations, or by linking in a 2D beam model, where point mass and loads are defined and in addition rotations, such to link the beam model rotational degree of freedom to the solid body model rotations.

One way is to add a point on an existing boundary. If the point is well defined on a surface in the ***Geometry*** node, it will appear as a "hard node point" on the ***Boundary Entity***. Thereafter one can apply a ***Boundary Point Load*** corresponding to the point mass **m[kg]** times the desired acceleration.

Even better, to avoid the singularities of a ***Point Load***, the load can be distributed over a surface via a ***Boundary Load*.**

However, one must not forget that a load will not, be considered in an eigenfrequency, nor for a frequency sweep, analysis. Because then the physics equations are transformed and external forces, not related to the eigenfrequency variable ***lambda[1/s]*** respectively frequency ***freq[Hz],*** are ignored.

In 3D solid physics this can be the easiest achieved by adding a ***Boundary Load*** depending on the eigenfrequency ***lambda*** such as:

-lambda^2*m*u

and respectively ***v** , ***w** ; this works fine also for stationary cases as then COMSOL Multi-physics defines internally ***lambda = 0*** (just as for the time ***t*** variable). Nevertheless, this might lead to eigenfrequencies with small residual complex values in V4.1.

Continuing in this way, one can further add some damping (**damp[m/s]**) and any solid spring constant (**stiffx[N/m]**) to the above equation, and defining the full boundary load along "***x***" as:

Fx+(stiffx+*lambdadamp-*lambda^2**m)**u***

Equations for **y** and **z** apply by similarity. To adapt the above equations to a frequency sweep, one can add a simple global variable:

lambda = -i*2*pi*freq

To do it in a cleaner way, one could replace ***lambda*** by ***MyLambda*** and have a *Variable* section redefining MyLambda to *0[1/s]*, *lambda*, *-1*2*pi*freq* for a Stationary, Eigenfrequency respectively Frequency Sweep Study.

Note that the above equations do not consider any inertia components; these must be coupled in via rigid body rotations which are not defined, by default, for solid physics models. This exercise is left to the readers.

Now, what if the point of the CoG is not situated on any 3D boundary, but is outside of any domain?

This requires some additional steps; here too, things have changed from v3.5a. Any single point in space not belonging to any domain will be considered as an orphan and will not be included in the entities list, so one cannot add any equations to these orphan points. If we try adding a new ***General*** or ***Coefficient Form PDE*** physics we will be restricted to ***Domains***, not allowing us to select single points. To select points we can use the ***Weak Form Point PDE (wp)*** physics (for each point). However ***wp*** does not accept any eigenfrequency analysis, which means we are restricted to stationary cases.

My way around, while waiting for a general "soft" boundary conditions, hopefully to come in one of the future releases, is to couple beam physics to my solid physics. This requires defining the rigid body rotations to the beam rotations. These rotations can be expressed, for small deformations, from the curl of the solid physics displacement field (see also the equations for the internal variable ***solid.curlUZ***).

4. Future, nice-to-have features

My wish list is long. "Soft link" BC's, as complement to the Rigid Boundary Connector, are high up on my priority list for structural load cases. As well as automatic domain reduction to their corresponding Point Mass and Point Load singular items, both positioned at the selected domain's CoG.

Then to couple in inertia loads, one need a better access to the rigid body rotations and their derivatives. Today the closest one can get to rotations is to take the curl of the displacement field, which for small angles corresponds nicely to the rotations:

Thx = 0.5[rad]*(*wY-vZ*)
Thy = 0.5[rad]*(*uZ-wX*)
Thz = 0.5[rad]*(*vX-uY*)

Note the uppercase ***X, Y, Z*** required if the ***Spatial Frame*** is active as it is by default in V4; replace by lower case letter if not, as well as for V3.5a. These formulas are **not** valid for large deformation analysis. These values can also be averaged over a given interface ***Boundary***.

In V4 the participation mass factors have appeared for eigenfrequency analysis (see ***Study - Solver Configuration – Eigenvalue Solver - Output*** sub-node), this is an essential tool to analyze and to hierarchies eigenfrequencies, in particular for automation and control applications. However with solid physics this can also be applied to the rotations such to sort eigenmodes that are of importance to rotational degrees of freedom, something not covered today.

Further essential elements for model validation and verification are to have a direct access (and printout) of the model partial and total mass, centre of gravity, and full inertia tensor for different coordinate systems; the minimum being the CoG and the default coordinate systems. Systematic comparison between the CAD values and the meshed values of these variables ensure that the FEM model has correct material definitions and scale. Today, all these values must be defined as user variables and post-processed separately which is time consuming and error prone, i.e.:

M=*intop1(solid.rho)*
Jzz=*intop1(((x-X0)^2+(y-Y0)^2)*solid.rho)*

Today, COMSOL Multiphysics can handle complex models as this one comprising more than 725 domains assembling more than 700 small magnets of different shapes, all made within COMSOL GUI environment, this ACDC model solves within a few minutes, but moving up and down in the model tree can take twice as long, turning off the automatic refresh option seem also to help.

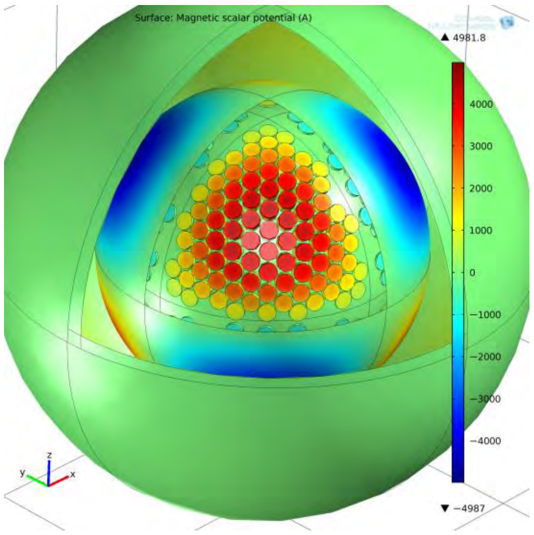


Figure 5. 725 magnet domains, COMSOL model solved for the Magnetic Scalar potential ***A***.

5. Conclusions

COMSOL Multiphysics is today the unique tool allowing mixing many types of different physicals models under the same software environment.

Based on its large palette of PDE solvers, the most complex cases of physics simulations, as those needed today, can be solved efficiently and rapidly. The new v4 environment is particularly powerful and allows for an efficient model building.

Nevertheless, COMSOL Multiphysics still appear as a tool for physicist and academicians, many engineers, particularly those coming with a long background in older FEM tools, need some time to find their marks, and have to learn back how to write out the physics equations for many load cases, even in the classical domain of structural FEM physics.

COMSOL Multiphysics latest release v4.1 has incorporated several useful new features, also for the structural domain, and I have been told that more is in preparation. This is a must when COMSOL Multiphysics will really start to takes shares from the classical FEM market, as most engineers have a long-term background in structural analysis, but need to go beyond.

I hope that the few examples presented here illustrates in fact how easy it is to complete your models with custom physics, but I agree it takes some time to type it in, and specially to validate all this typing, and time is unfortunately one of the scarcest resources, for an engineer.

6. References

1. COMSOL V4.1, *Structural mechanics Module Users Guide*. COMSOL AB, (2010)

2. COMSOL V4.1, *Structural mechanics Module Verification Models*. COMSOL AB, (2010)

3. *Spherical magnetic rotor, COMSOL Multiphysics v4.1*. Model by Leopoldo ROSSINI, CSEM sa (2010)

IKj, November 2010