User-defined Superconductivity Formulations in COMSOL - Tipps for Efficient and Reusable Implementations

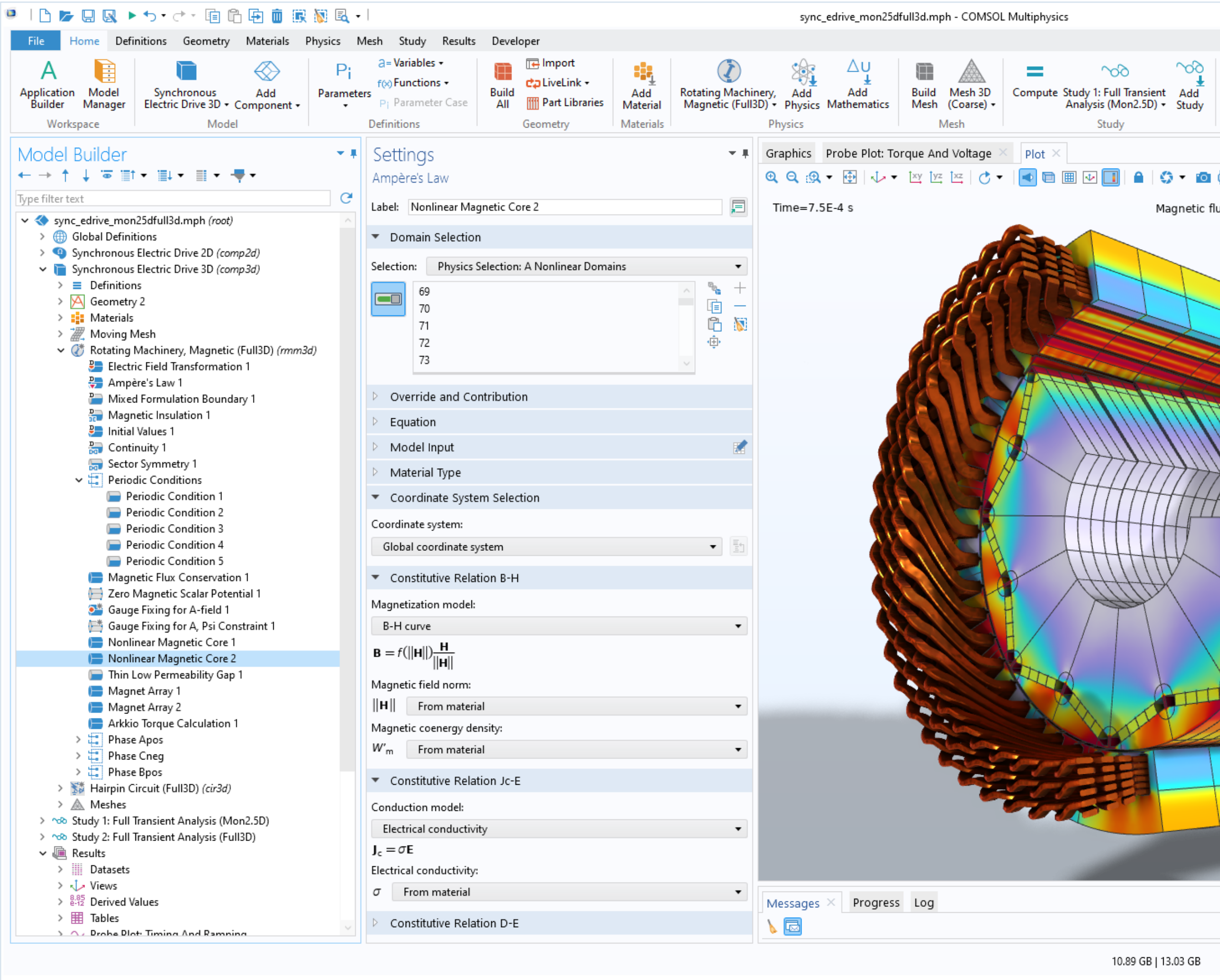
Sven Friedel
COMSOL

Alexandre Arsenault
PSI

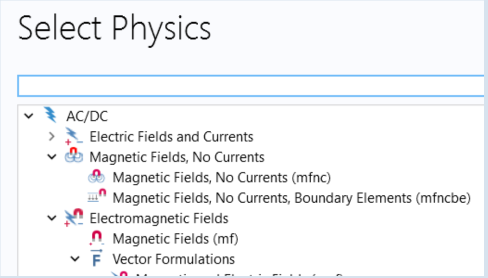
COMSOL

COMSOL Fully Integrated Software Suite

- Multiphysics simulation, from building geometry to results analysis, via a streamlined physics-based modeling workflow
- Application Builder
- Model Manager



Typical COMSOL Usage in the HTS Community



Built-in Physics Interfaces

- H-formulation (*mfh*)
- A-formulation (*mf*)
- ϕ –formulation (*mfnc*)
- Several more…

$$\mathbf{J}_c = \sigma \mathbf{E}$$

$$\mathbf{E} = \rho(\nabla \times \mathbf{H} - \mathbf{J}_e)$$

$$\nabla \times \mathbf{E} = -\frac{\partial \mathbf{B}}{\partial t}$$

$\Leftrightarrow$

$$\mathbf{H} = -\nabla V_m$$

$$\nabla \cdot \mathbf{B} = 0$$

Manual Coupling of Built-in PI

- Use two or more of built-in interfaces and couple in boundaries to obtain H-ϕ, A- ϕ, T-A, etc.
- *“straightforward”*

$$e_a \frac{\partial^2 \mathbf{u}}{\partial t^2} + d_a \frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot \Gamma = f$$

$$0 = \int_\Omega \text{weak } \partial S$$

Equation-based Interfaces

- Coefficient-form
- General form
- Weak form
- *“innovative but nerdy”*

My Topic Today: Further Power Tools Uncovered

1

Modify Built-in Eqns.

Less need to reinvent the wheel, concentrate on your innovation.

2

Record Methods

Design reusable *model methods* to avoid repetitive settings work *in one model*.

3

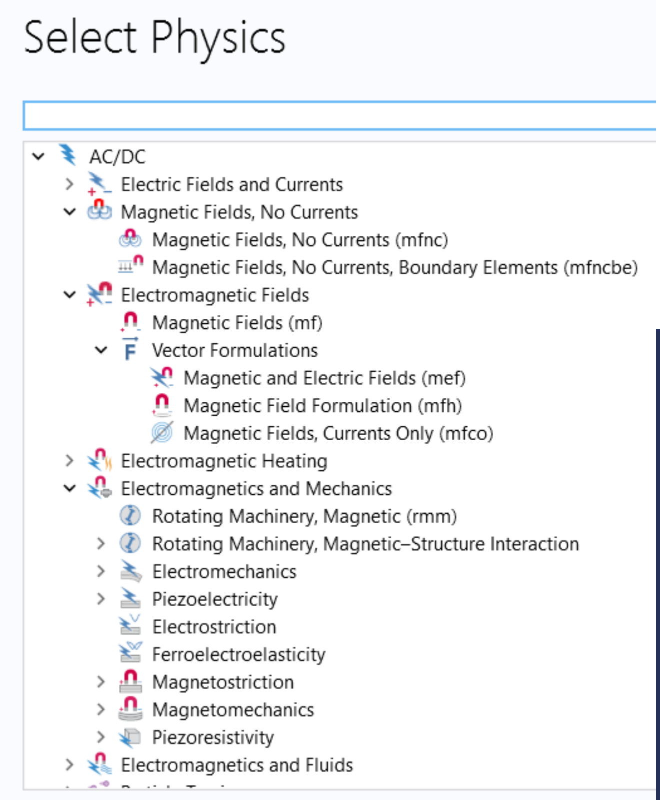
Add-ins

Publish code snips to be re-used in *several models.*

4

Physics Builder

Design your own tailor-made physics interfaces that everybody can use and plug them into COMSOL.



Built-in Interfaces

Advantages

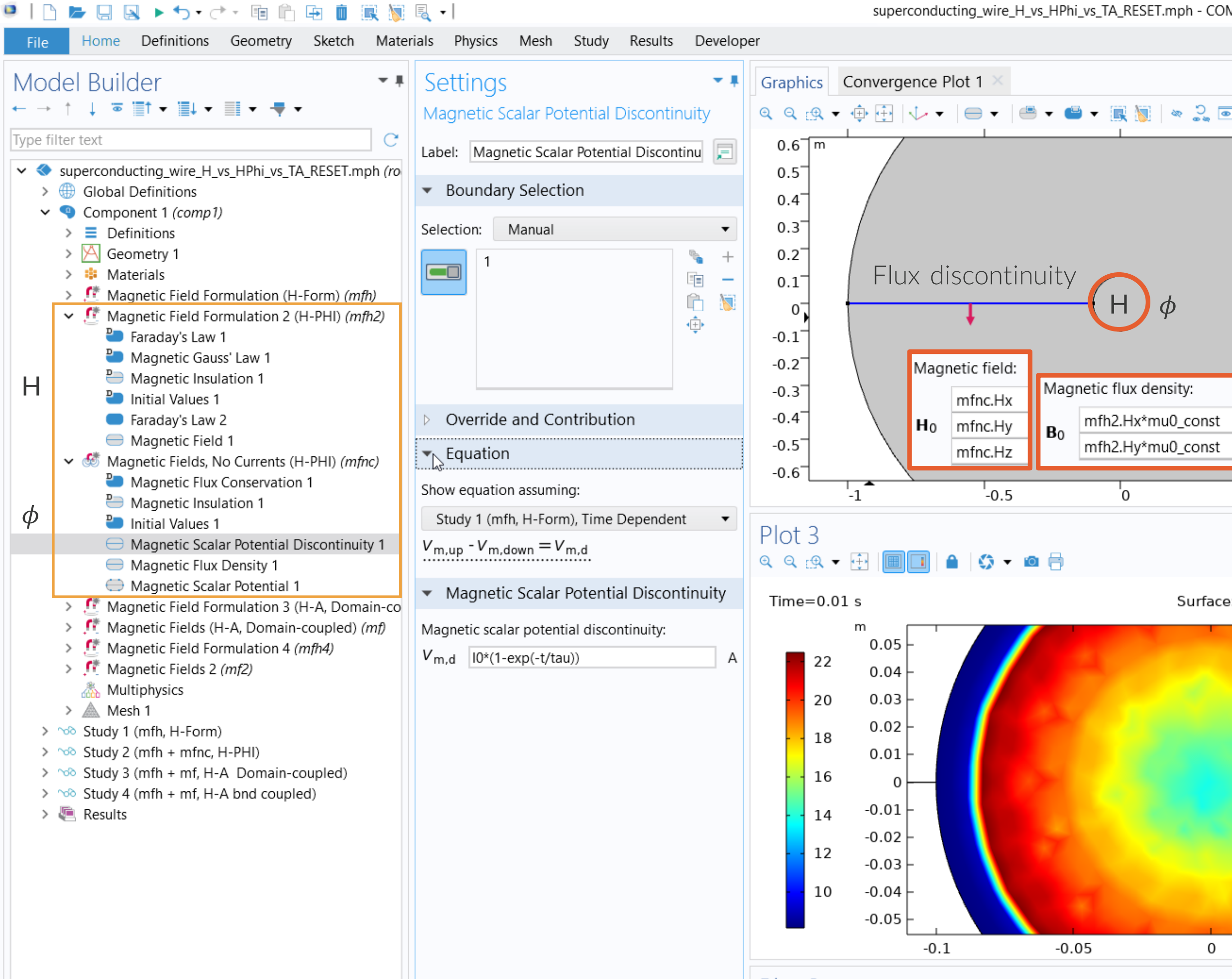
- Well-established and maintained technology
- Constantly extended by new features
- Easy coupling with any other existing technology

Recent News

- *(mfco)* Biot-Savart formulation
- *(mfncbe)* Boundary Elements formulation
- Time-periodic FEM solver
- Etc…

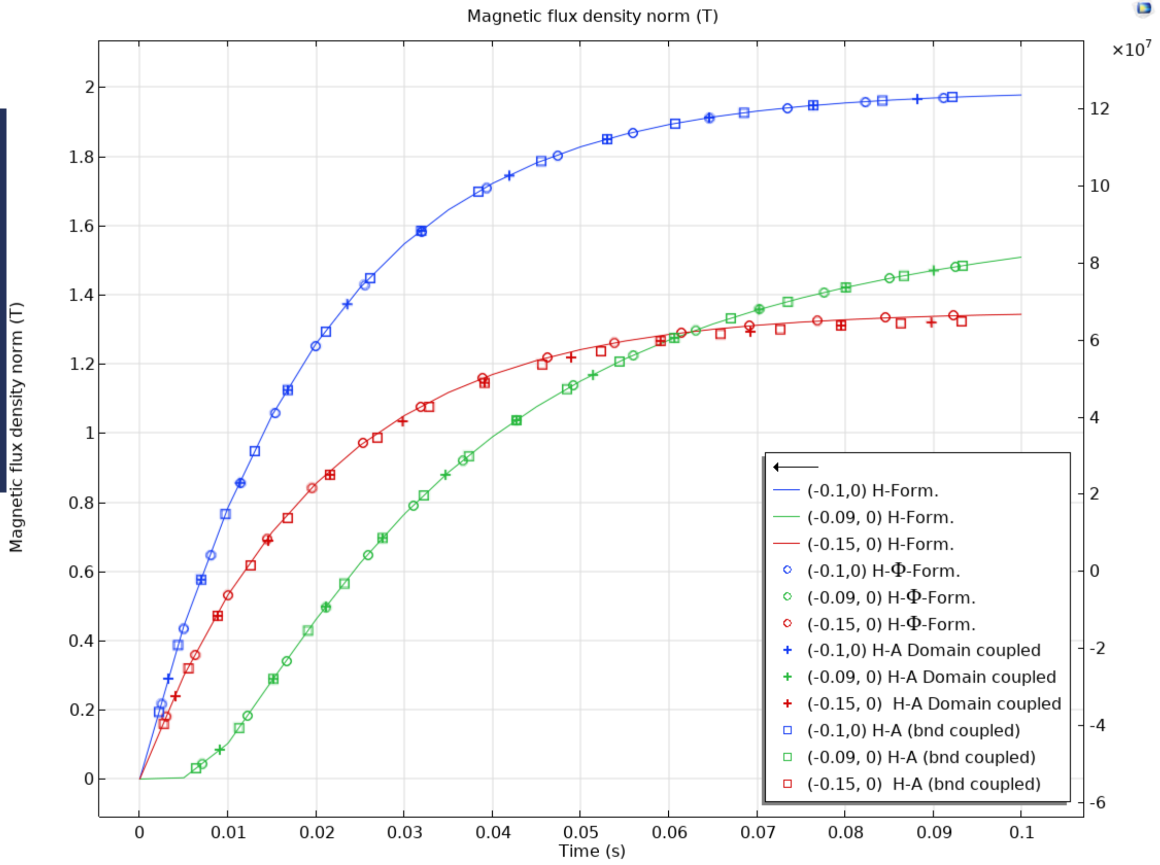
Manual Coupling

- Activate 2 or more physics
- Define domains of validity
- Couple at interfaces
- Enjoy many built-in features, e.g. discontinuities, constitutive relations, solver defaults



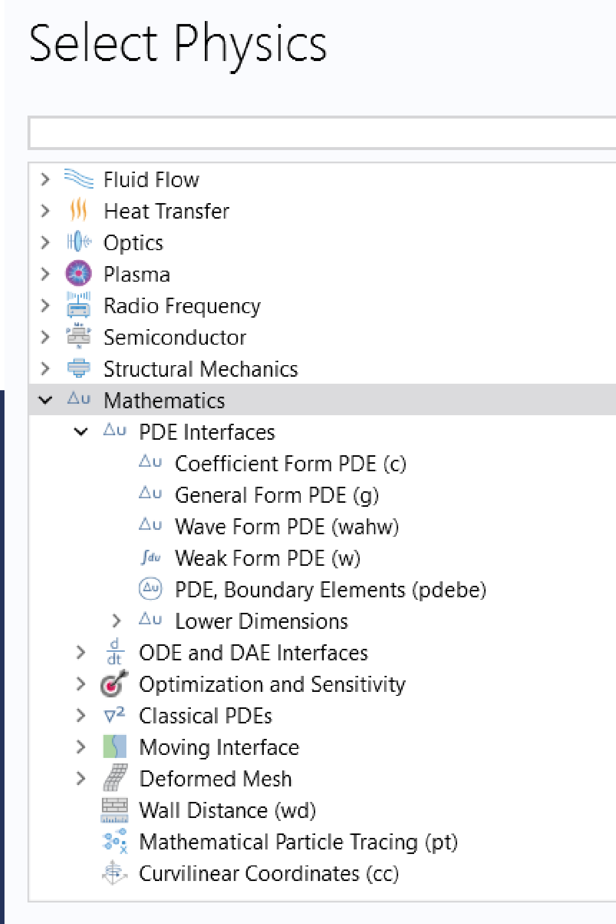
Manual Coupling Results

- Proven success in many publications
- Research topics are:
  - Order of shape function
  - Accuracy vs. speed
  - Domain- or boundary coupling



Why would one use anything else? 🧐

COMSOL



Equation-Based Interfaces

Pros

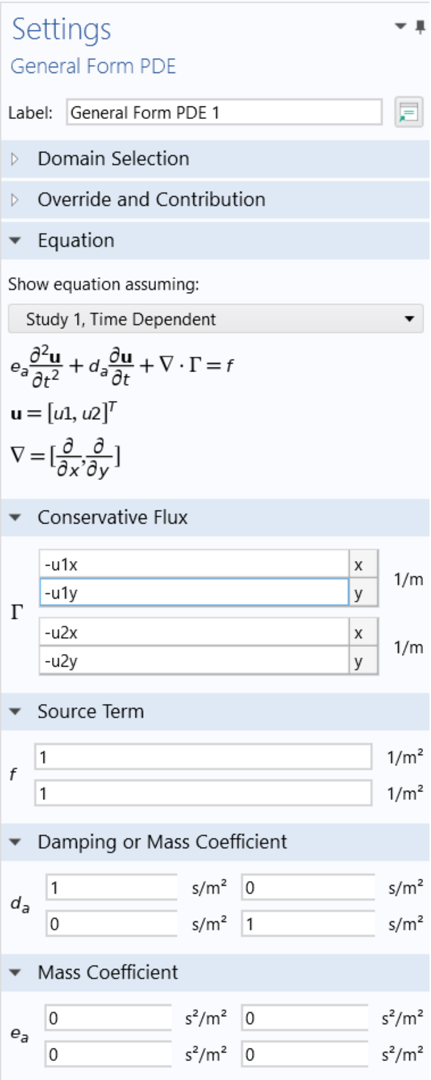
- Full control for testing
- Textbook notation

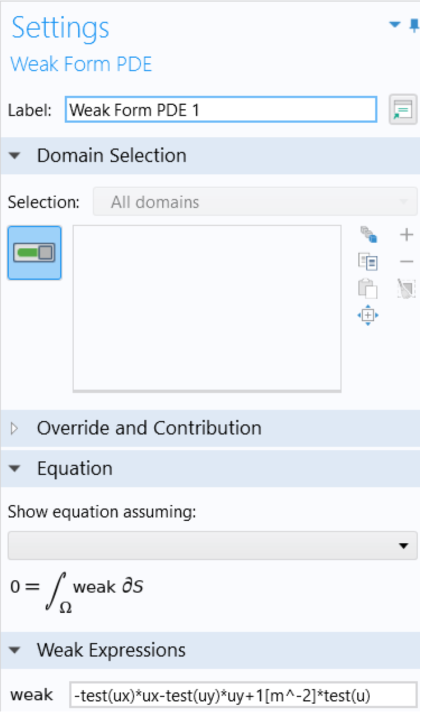
Cons

- No default units
- Simple solver defaults
- No constitutive relations
- Basic boundary conditions
- Reuse is “nerdy”

Equation-Based Modeling

- Used in many publications
- Access to weak form
- 3D, 2D, 1D entities





Example: Magnetodynamic H- ϕ Formulation [1]

Second, we implemented the H–ϕ formulation with Faraday's law in nonconducting domains, such that the equations are given by [5]

$$\text{In } \Omega_c^C: \quad \frac{\partial \mu \nabla \phi}{\partial t} = 0$$
$$\text{In } \Omega_c: \quad \nabla \times (\rho \nabla \times \mathbf{H}) = -\frac{\partial \mu \mathbf{H}}{\partial t}. \qquad (5)$$

We refer to this case as the magnetodynamic H–ϕ formulation (H–ϕ/D).

The weak equations for this formulation are given by

$$\frac{\partial}{\partial t}\mu \int_{\Omega_c^C} \nabla\phi \cdot \nabla\tilde{\phi}\, \mathrm{d}\Omega + \int_{\Gamma} \hat{\mathbf{n}} \times \mathbf{E} \cdot \nabla\tilde{\phi}\, \mathrm{d}\Gamma = 0 \qquad (6)$$

$$\int_{\Omega_c} \rho \nabla \times \mathbf{H} \cdot \nabla \times \tilde{\mathbf{H}}\, \mathrm{d}\Omega + \frac{\partial}{\partial t}\mu \int_{\Omega_c} \mathbf{H} \cdot \tilde{\mathbf{H}}\, \mathrm{d}\Omega - \int_{\Gamma} \underbrace{\hat{\mathbf{n}} \times \mathbf{E}}_{\mathbf{E}_t} \cdot \tilde{\mathbf{H}}\, \mathrm{d}\Gamma = 0. \qquad (7)$$

$$\underbrace{\int_{\Omega_C^C} -\mu\left(\frac{\partial \phi_r}{\partial t}\tilde{\phi}_r + \frac{\partial \phi_z}{\partial t}\tilde{\phi}_z\right) d\Omega}_{weak\ term} \qquad \underbrace{\frac{\partial}{\partial t}(\nabla \cdot \mathbf{B}) = 0}_{strong\ form}$$

Weak Contribution

Weak expression:

```
mu0_const*d(-phir,t)*test(-phir)+mu0_const*d(-phiz,t)*test(-phiz)
```

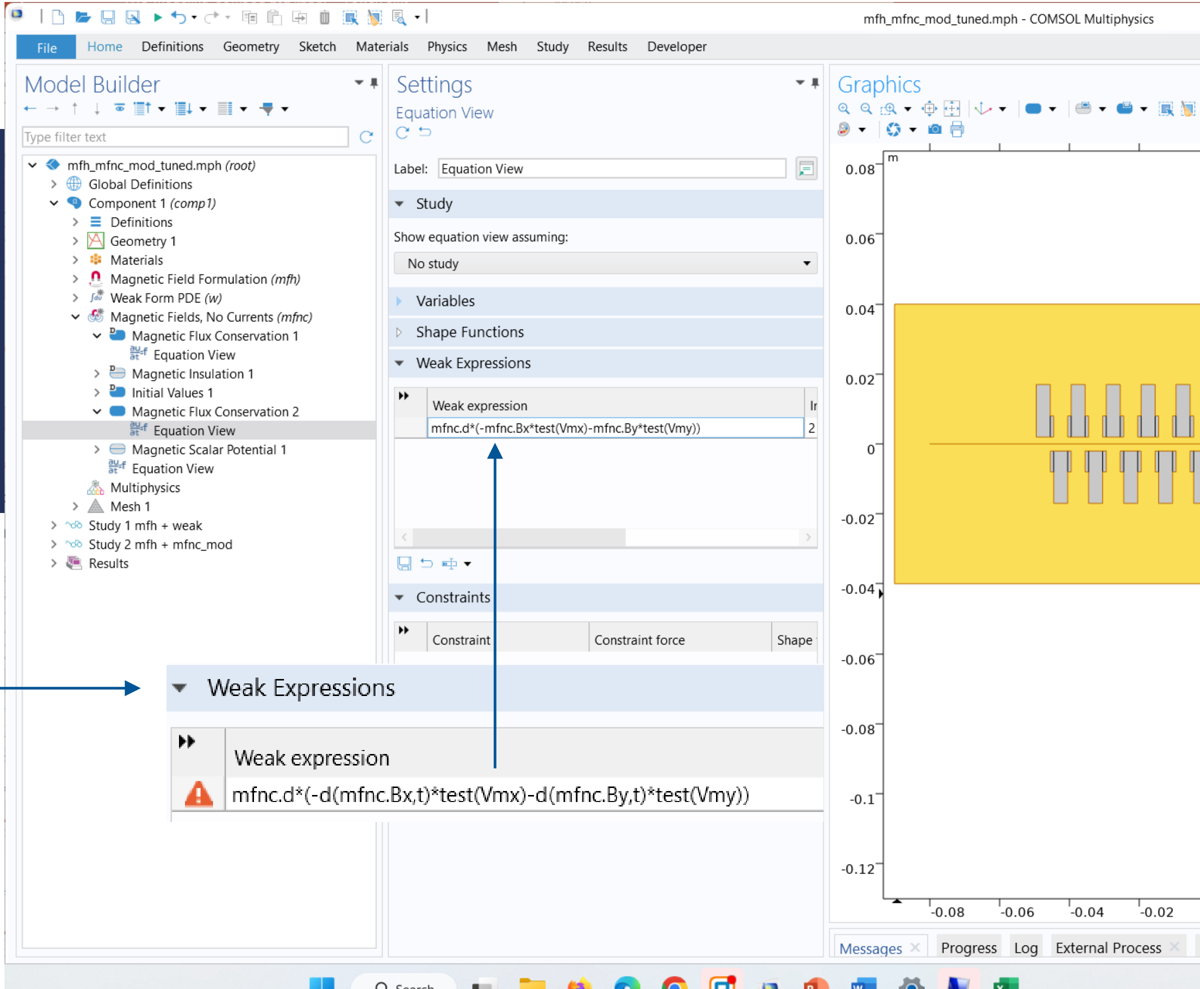

[1] *A. Arsenault , B. de Sousa Alves , G. Giard , and F. Sirois , «Magnetodynamic H–φ Formulation for Improving the Convergence and Speed of Numerical Simulations of Superconducting Materials", IEEE Trans. Appl. Supercond., Vol. 33, No. 7, 2023.* https://doi.org/10.1109/TASC.2023.3293449

COMSOL

1

Modify Built-in Equations

- Use ϕ –formulation (*mfnc*) instead of weak form
- **Change weak term only**
- Enjoy all the goodies of built-in formulation



Modify Built-in Equations

- Compare compute time
  - PDE Version: 36 s
  - Modified mfnc: 56 s (+55%)
- The PDE version uses less robust solver defaults; need to change those, too.

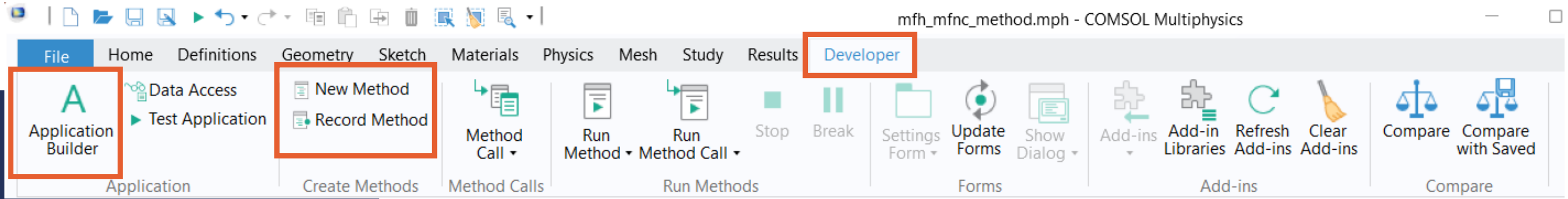
Solver Level	Setting	PDE	mfnc	runtime [s]
Direct solver	Check error estimate	No	Auto	56
Advanced	Reuse sparsity pattern	off	on	61
Fully Coupled	Jacobian Update	minimal	once per time-step	53
Fully Coupled	Max. No of iterations	4	15	38
Fully Coupled	Tolerance factor	1	0.2	38
Time stepping	Steps taken by solver	free	strict	36
Time stepping	Error estimation	Include algebraic	Exclude algebraic	36
Direct solver	Change from MUMPS to	PARDISO	PARDISO	20

Do I really need to change 10 settings manually, again? 🧐

2

Record Method

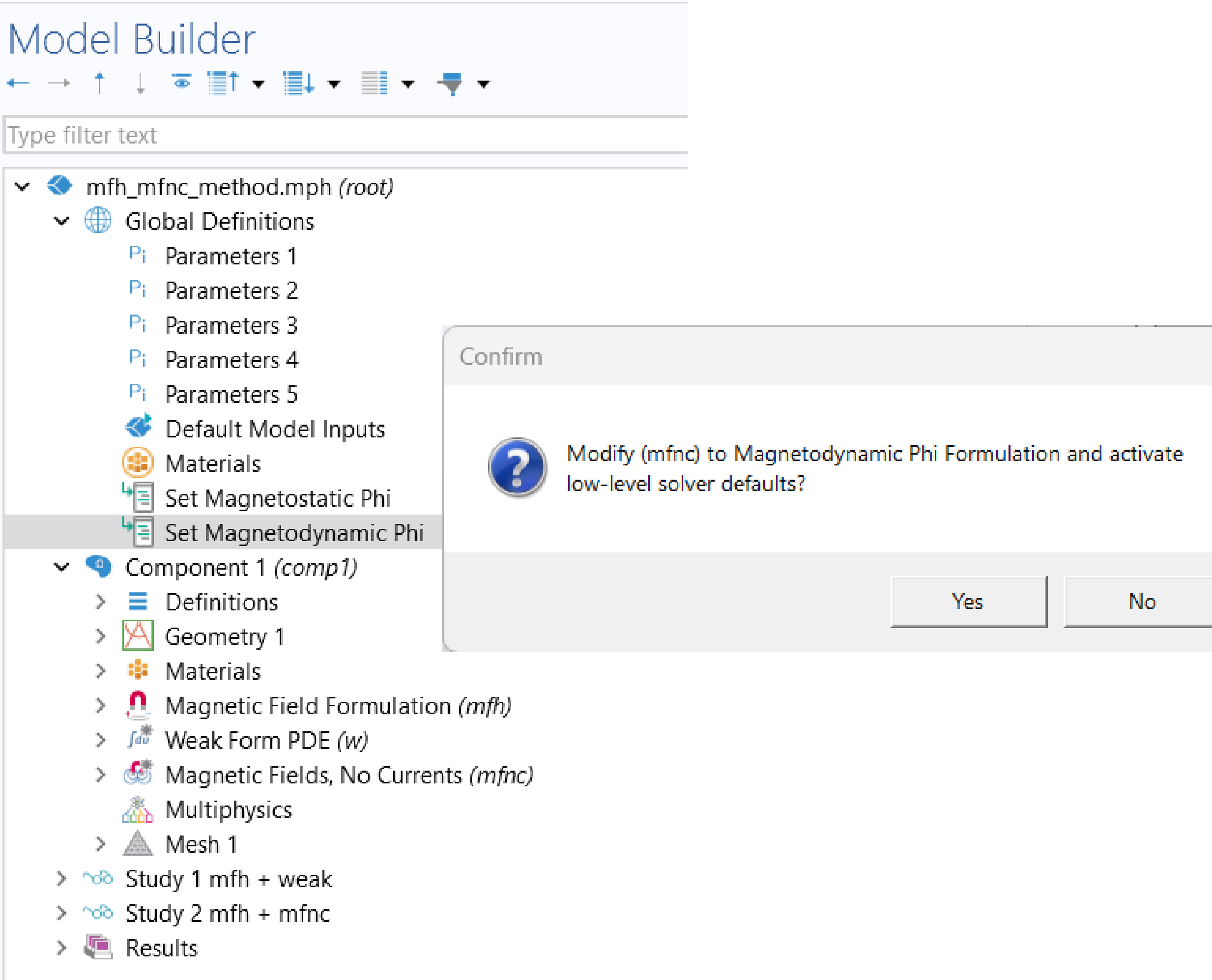
- COMSOL allows to record GUI operations as code from its Java® API




```java
// Modify weak form to Magnetodynamic Phi

model.component("comp1").physics("mfnc").feature("mfc1").featureInfo("info")
  .set("root.comp1.mfnc.mfc1.weak$1", new String[]{"mfnc.d*(-d(mfnc.Bx,t)*test(Vmx)-d(mfnc.By,t)*test(Vmy))", "2"});
model.component("comp1").physics("mfnc").feature("mfc2").featureInfo("info")
  .set("root.comp1.mfnc.mfc2.weak$1", new String[]{"mfnc.d*(-d(mfnc.Bx,t)*test(Vmx)-d(mfnc.By,t)*test(Vmy))", "2"});

// Set Solver to PDE defaults
model.sol("sol3").feature("t1").feature("dDef").set("errorchk", false);
model.sol("sol3").feature("t1").feature("aDef").set("cachepattern", false);
model.sol("sol3").feature("t1").feature("fc1").set("jtech", "minimal");
model.sol("sol3").feature("t1").feature("fc1").set("maxiter", 4);
model.sol("sol3").feature("t1").feature("fc1").set("ntolfact", 1);
model.sol("sol3").feature("t1").set("tstepsbdf", "free");
model.sol("sol3").feature("t1").set("estrat", "include");
model.sol("sol3").feature("t1").feature("dDef").set("linsolver", "pardiso");

// Rename Physics and Study
model.component("comp1").physics("mfnc").label("Magnetic Fields, No Currents, modified");
model.study("std2").label("Study 2 mfh + mfnc_mod");
```

Method Call

- Embed method calls into the Model Builder tree.
- Methods can be refined to detect existing feature nodes.



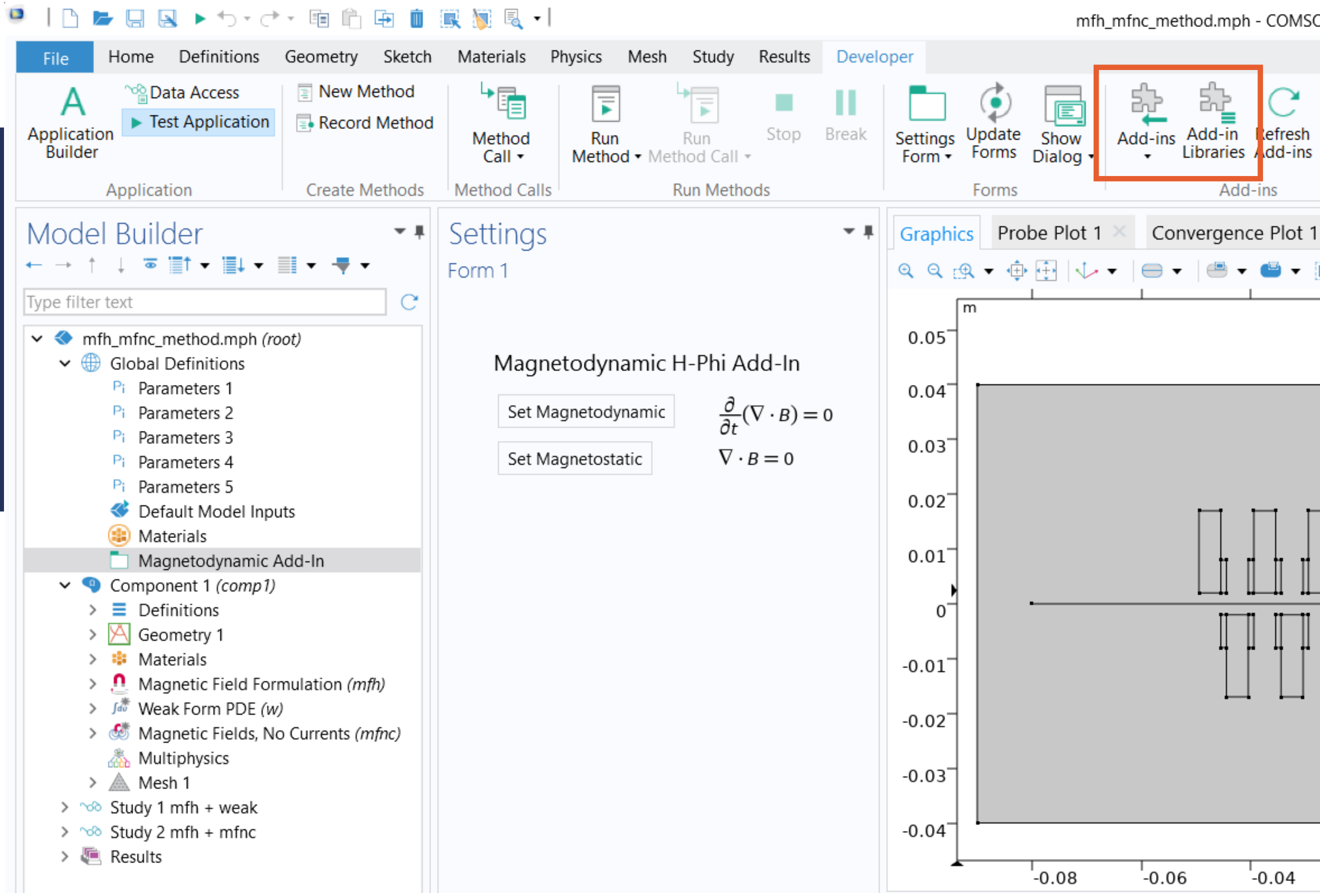
Can I reuse methods in other files? 🧐

COMSOL

3

Add Ins

- Make methods and forms reusable in other models.
- Load Add-in from library.
- Make sure that the Add-in is robust!

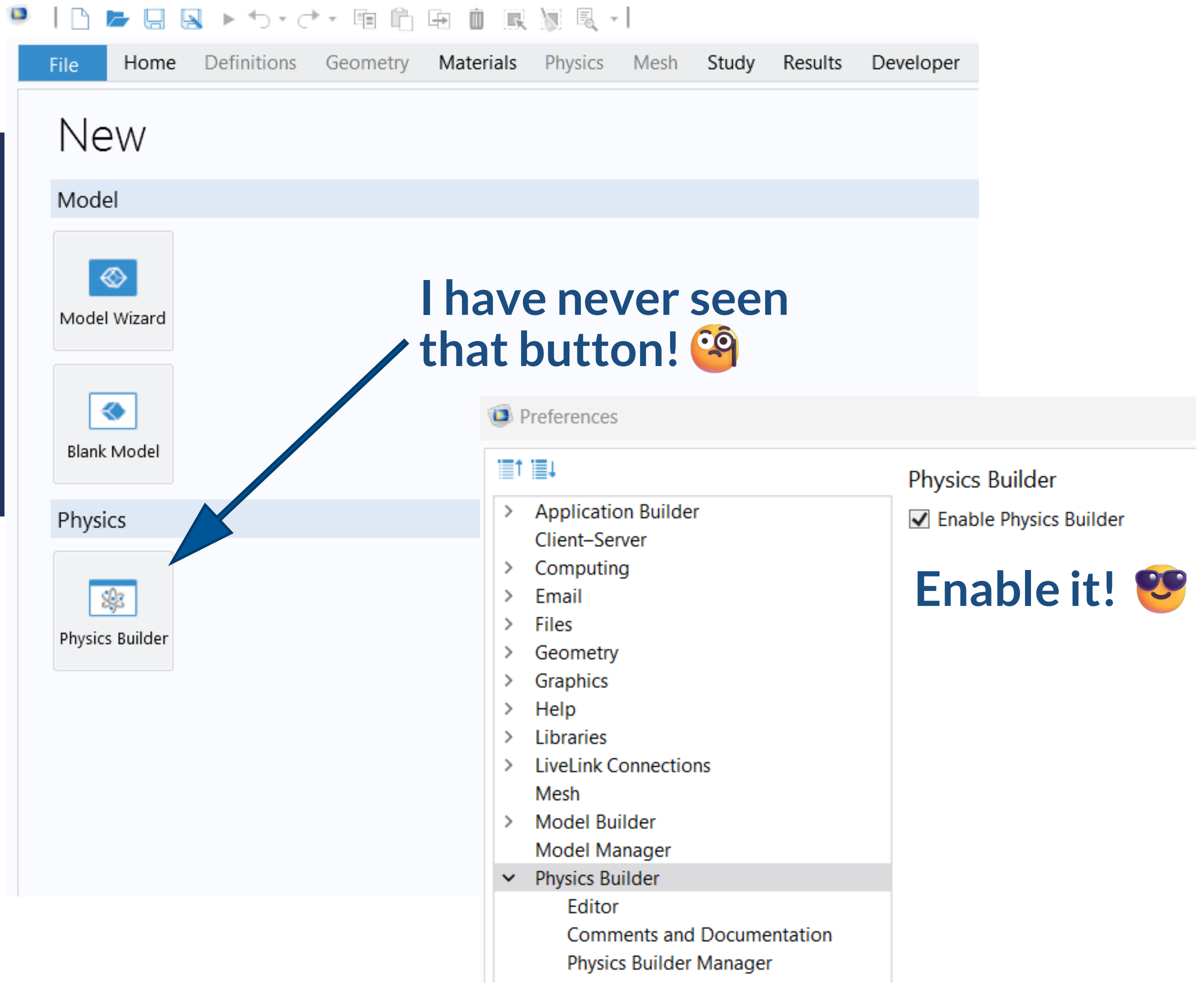


But what if I want to do a 3D model? Rewrite the Add-In? 🧐

4

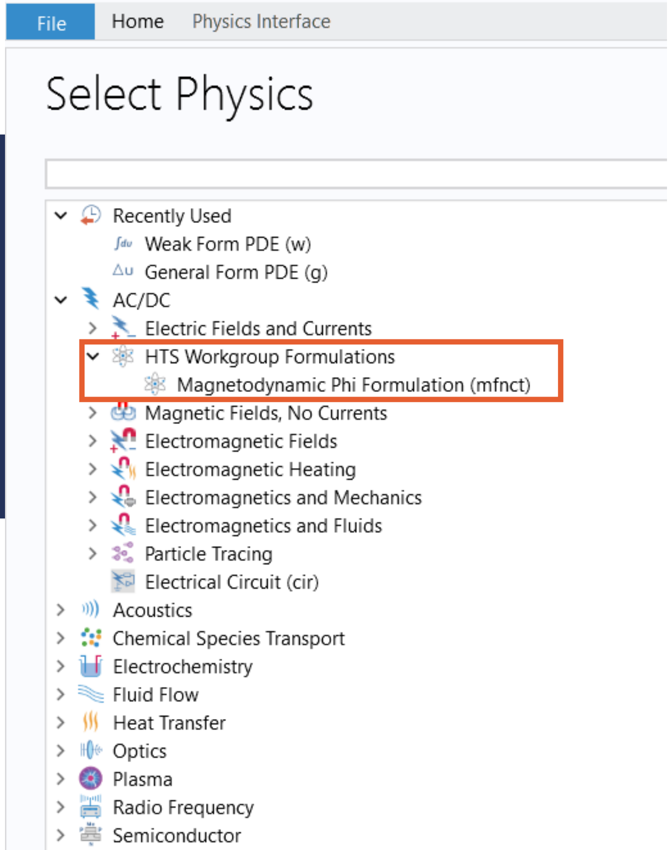
Physics Builder

- Build your own physics interfaces, and use them just like standard COMSOL physics.



Physics Builder

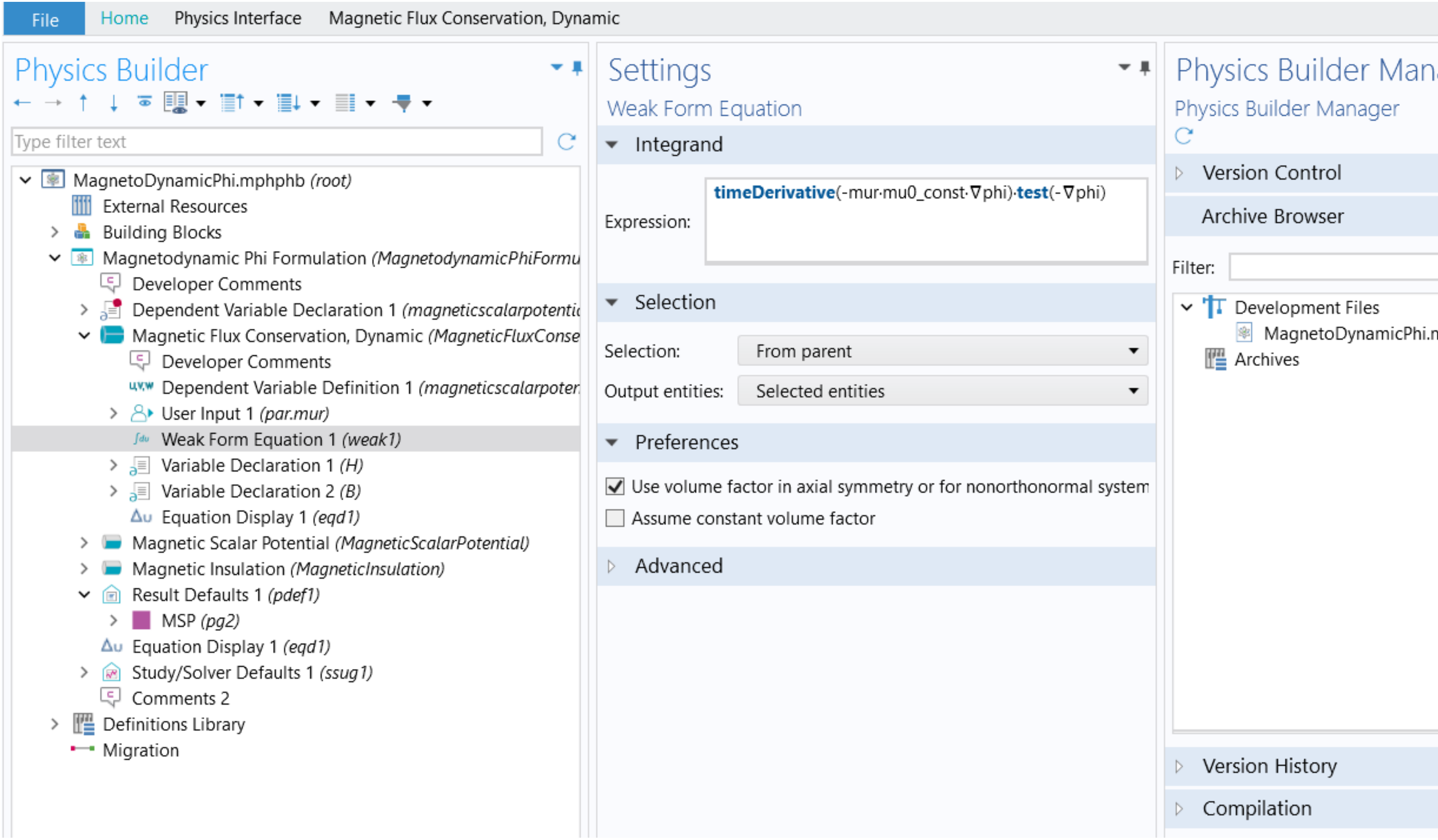
- Physics appear in the regular *Select Physics* dialog



PHYSICS BUILDER – AN INTEGRATED DEVELOPMENT PLATFORM

- Definition of weak form in textbook notation.
- Works in all permitted space dimensions, including extra dimensions.
- Default domain and boundary conditions.
- Default solver settings.
- Default Plot groups.
- Version history.
- Compilation.

Try it! 😎



Take Home Messages

1

Modify Built-in Eqns

Less need to reinvent the wheel, concentrate on your innovation.

2

Record Methods

Design reusable *model methods* to avoid repetitive settings work *in one model*.

3

Add-ins

Publish methods and user interfaces to be re-used in *several models.*

4

Physics Builder

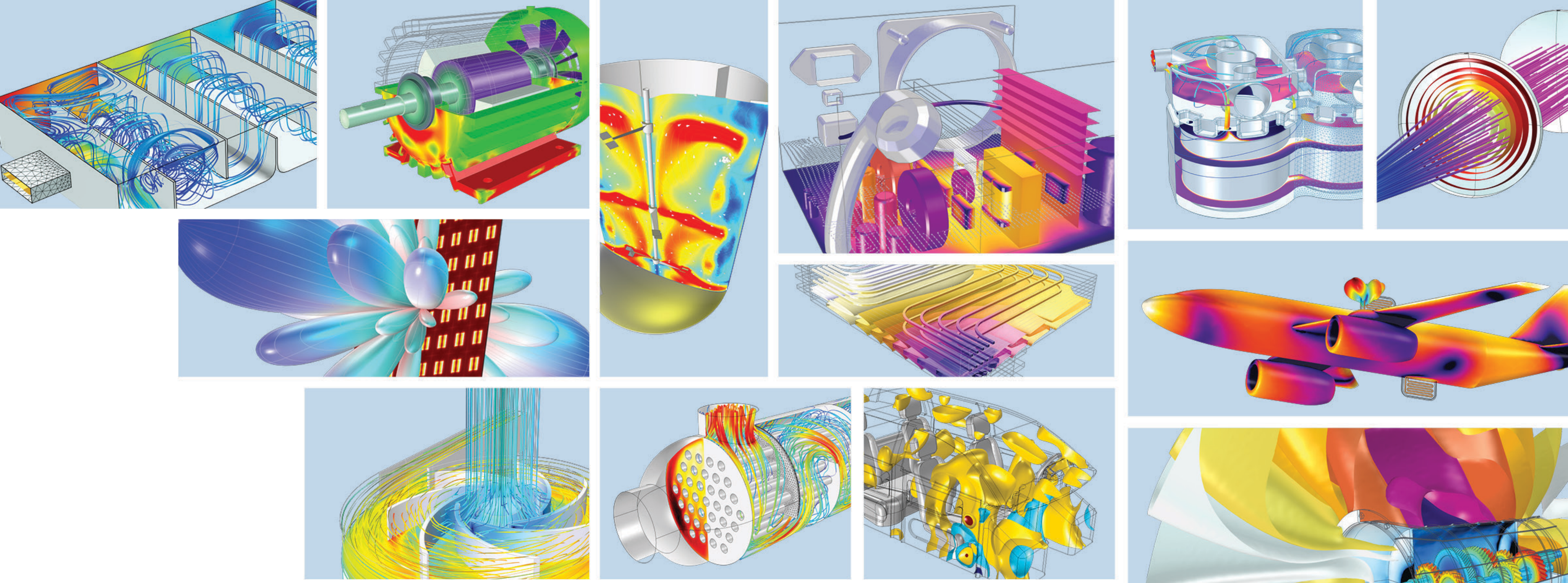
Design your own tailor-made physics interfaces that everybody can use and plug them into COMSOL.

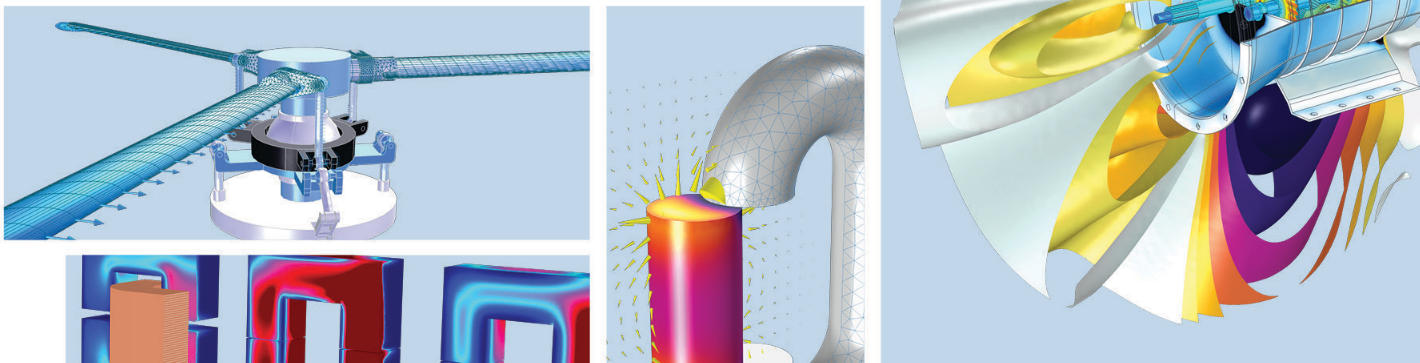
Enjoy the next level of Multiphysics Modeling using ***user-defined*** formulations.

Resources

- 11 h course: Modeling with Partial Differential Equations in COMSOL Multiphysics®
- Model Methods
- Add Ins
- Physics Builder

Who has the first question or comment?





Contact Us

www.comsol.com/contact