Coil Modeling with ElmerFEM

using Elmer Circuit Builder

Jonathan Velasco
ElmerTeam
CSC – IT Center for Science, Finland

CSC

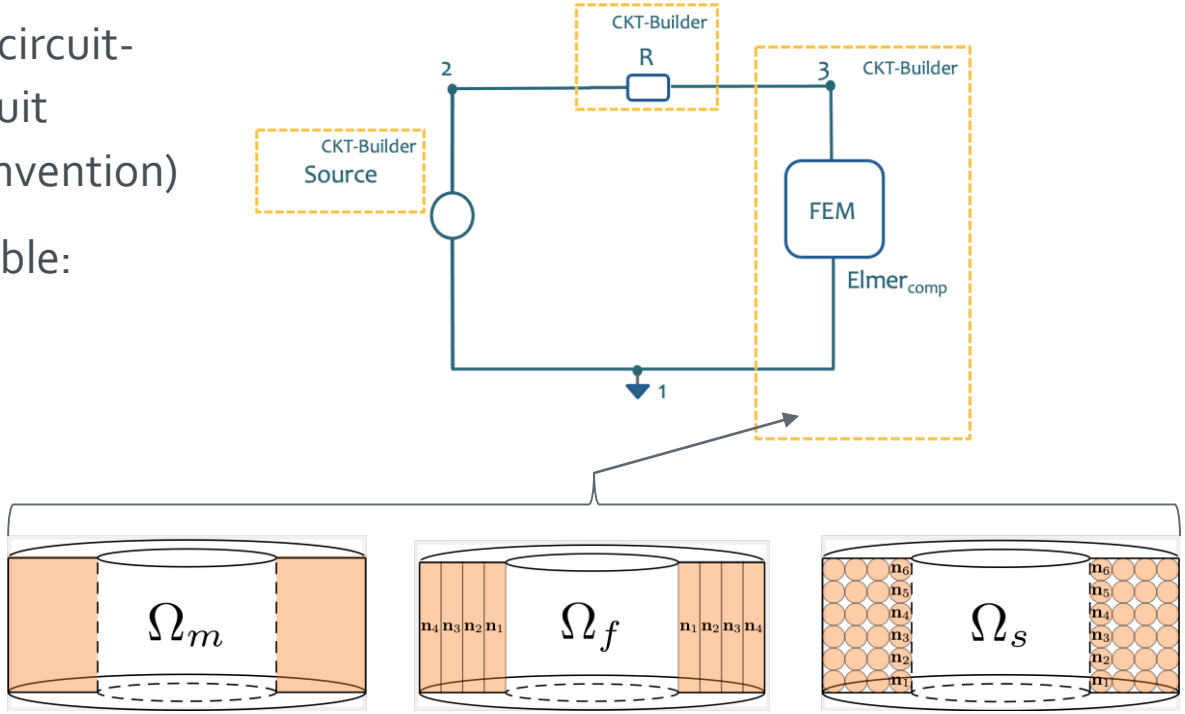
8 Dec 2021

Circuit modeling background

- Elmer has the ability to solve magnetoquasistatic equations strongly coupled with electrical circuits
- Elmer Circuits module was was developed as in collaboration between Finnish universities research labs and industrial partners in SEMTEC funded by TEKES (Business Finland).
- ElmerCircuit Builder is a complementary tool for the Circuit module making the writing of the circuit defeniotions easier

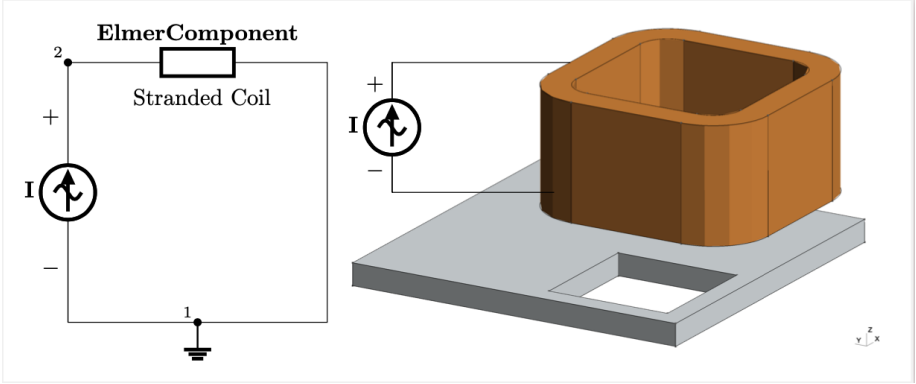
What is the Elmer Circuit Builder?

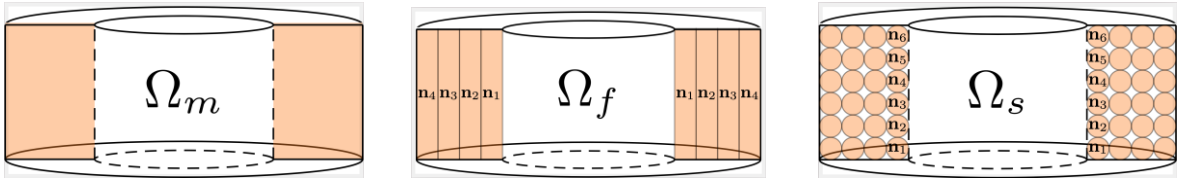
- Python module to automate circuit-matrix creation based on circuit connections (terminal/pin convention)
- Electrical Components Available:
  - Voltage Sources (V)
  - Current Sources (I)
  - Resistors (R)
  - Inductors (L)
  - Capacitors (C)
- FEM Coil types
  - Massive, Stranded and Foil



What kind of problems are we solving with the CircuitBuilder?

- Current conducting coils/wires (2D/3D) coupled to electrical circuit networks
- Coil Models (CircuitsAndDynamics.F90)
  - Massive (Solid conducting material)
  - Stranded (Homogenized)
  - Foil winding (Homogenized)
- Magnetodynamics av Formulation





ElmerSolver: Basic Structure of SIF file

- Basic idea: **Sections** + **Keywords**
- Each section starts with **SectionName** and ends with with "**End**"
  - Alternative for one keyword
    SectionName :: Keyword
- In each section we may have an arbitrary number of keywords
- Keywords are of type
  - **Real** : real valued number
  - **Integer** : integer number
  - **Logical** : True or False
  - **String**: not case-sensitive text
  - **File**: case-sensitive text

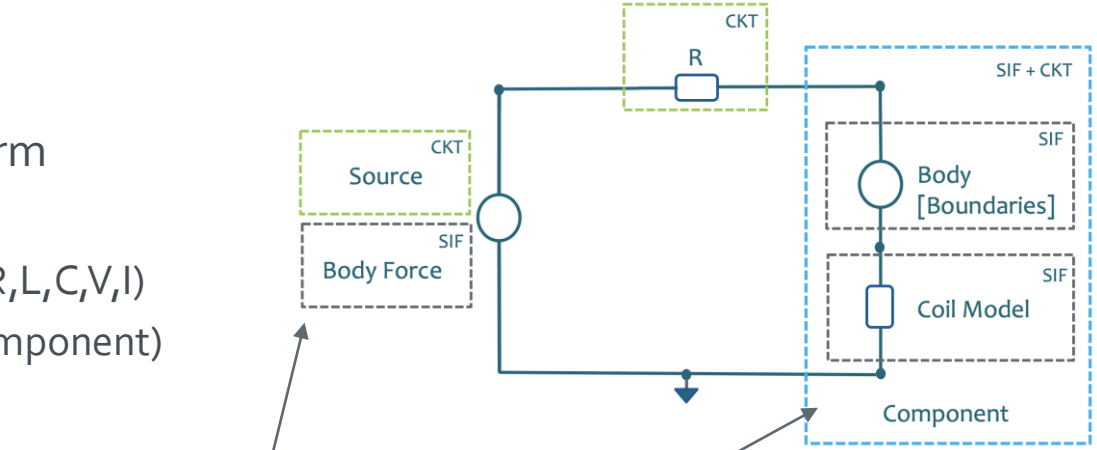
- Sections are
  - **Header**
  - **Constants**
  - **Simulation**
  - **Solver i**
  - **Body i**
  - **Equation i**
  - **Body Force i**
  - **Material i**
  - **Initial Condition i**
  - **Boundary Condition i**
  - **Component i**
  - **Run Control**
- Not all sections are needed

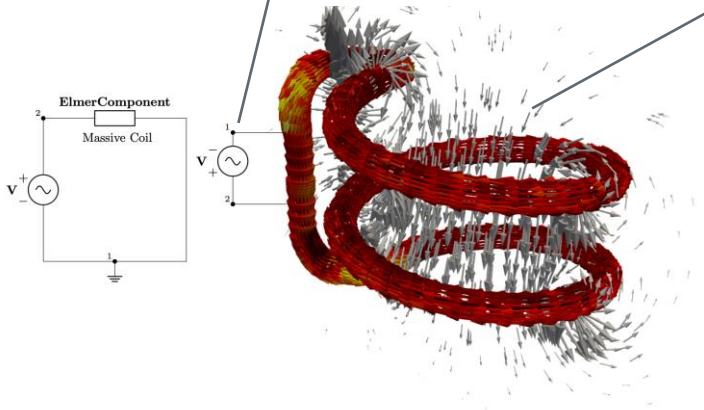
Example of minimal sif file

```
! Minimal sif file example
Check Keywords "Warn"

Header :: Mesh DB "." "square"

Simulation
  Max Output Level = 5
  Coordinate System = Cartesian
  Simulation Type = Steady
  Output Intervals(1) = 1
  Steady State Max Iterations = 1
  Post File = "case.vtu"
End

Body 1
  Equation = 1
  Material = 1
End

Equation 1
  Active Solvers(1) = 1
End

Solver 1
  Equation = "ModelPDE"
  Variable = "Field"
  Procedure = "ModelPDE" "AdvDiffSolver"
  Linear System Solver = Direct
End

Material 1
  diffusion coefficient = 1.0
End

Boundary Condition 1
  Name = ”Fixed”
  Target Boundaries(1) = 1
  Field = 0.0
End

Boundary Condition 2
  Name = ”Flux”
  Target Boundaries(1) = 2
  Field Flux = 1.0
End
```

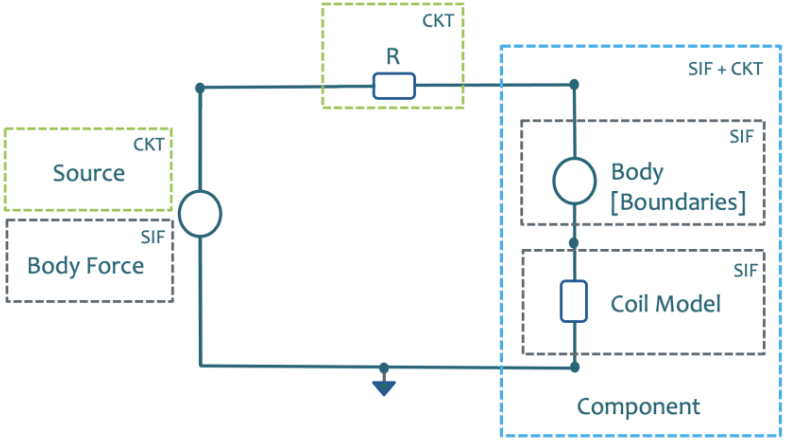
Elmer Circuits: Association between Source Input File (SIF) and Circuit Definition (CKT) Matrix Form Ax'+Bx = f

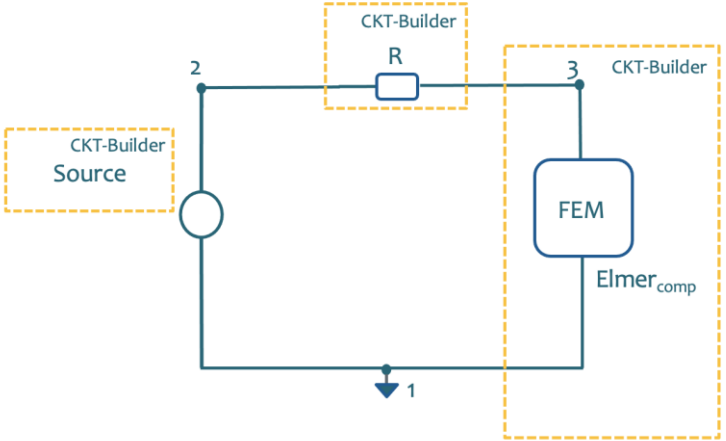
- Under Circuit File in Matrix Form
  - Source
  - Electrical Components (e.g: R,L,C,V,I)
  - [FEM] Component (ElmerComponent)
- Define in SIF
  - [FEM] Component
    - Coil Model
  - Body Force 1
    - Voltage or Current (V,I)





Elmer Circuits and Elmer Circuit Builder:

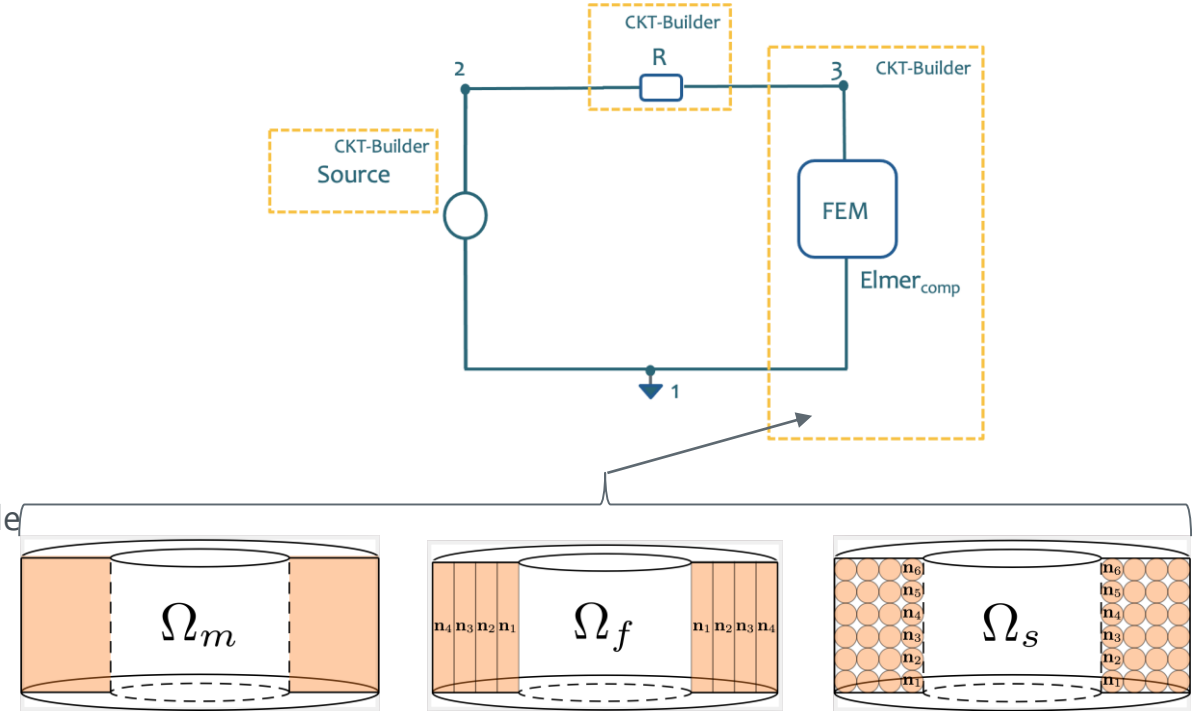




Elmer Circuit Builder: How to use it?

https://github.com/ElmerCSC/elmer-elmag/tree/main/CircuitBuilder

- What do you need?
  - Python 3 and appropiate editor
  - IDE not needed but encouraged
- How to install it?
  - $ pip install elmer-circuitbuilder
- How to set it up:
  - Import elmer_circuitbuilder library
  - Setup output file name
  - Set number of circuits
  - Populate circuit with ground/ref node
  - and needed electrical components
  - Add components to circuit
  - Generate Elmer Circuit
  - Add .definition to .sif file



Elmer Circuit Builder: The main template

https://github.com/ElmerCSC/elmer-elmag/tree/main/CircuitBuilder

```
# Imported Libraries:
import sys
from elmer_circuitbuilder import *                                              # STEP 1
# ------------------------------------------------------------------------------

def main(argv=None):

    # name output file - do not remove                                          # STEP 2
    output_file = ""

    # initialize circuits: number of circuits - do not remove                   # STEP 3
    c = number_of_circuits(1)

    # reference/ground node needed - do not remove.                             # STEP 4
    c[1].ref_node = 1

    # ----------------------- Electrical Network Definition ---------------------  # STEP 5

    # Components

    # Define coil type: massive, stranded, foil

    # Define dimension related features if needed (closed, open)

    # store components in array components = [comp1, comp2,...] - do not remove  # STEP 6
    c[1].components.append([])

    # ------------------------------------------------------------------------------

    # generate elmer circuit.definitions - do not remove / do not edit          # STEP 7
    generate_elmer_circuits(c, output_file)

    return 0


if __name__ == "__main__":
    sys.exit(main() or 0)
```

```
Coil = ElmerComponent("Coil", 1, 2, 1, [1])
```

```
elmer_circuitbuilder.ElmerComponent
def __init__(self,
             name: str,
             pin1: int,
             pin2: int,
             component_number: int,
             master_body_list: list[int],
             sector: Any = 1) -> None
```

ElmerComponent is a derived class of the Component class to represent 2D and 3D Coils in Elmer.

```
Vs = V("Vs", 1, 3, 1)
```

```
elmer_circuitbuilder.V
def __init__(self,
             name: str,
             pin1: int,
             pin2: int,
             value: float = None) -> None
```

V is a derived class of the Component class to represent ideal voltage sources in Volt.

Elmer Circuit Builder: output file <circuit_name>.definition

https://github.com/ElmerCSC/elmer-elmag/tree/main/CircuitBuilder

```
! ------------------------------------------------------------------------
! ElmerFEM Circuit Generated: November 02, 2021
! ------------------------------------------------------------------------

! ------------------------------------------------------------------------
! Number of Circuits in Model
! ------------------------------------------------------------------------
$ Circuits = 1

! ------------------------------------------------------------------------
! Parameters                                                          (1)
! ------------------------------------------------------------------------

! General Parameters
$ I1 = 10

! Parameters in Component 1: Coil1
$ N_Coil1 = 10   ! Number of Turns
$ R_Coil1 = 0.1  ! Coil Resistance
$ Ns_Coil1 = 1   ! Sector/Symmetry Coefficient (e.g. 4 is 1/4 of the domain)
```

```
! ------------------------------------------------------------------------
! Matrix Size Declaration and Matrix Initialization                   (2)
! ------------------------------------------------------------------------
$ C.1.variables = 5
$ C.1.perm = zeros(C.1.variables)
$ C.1.A = zeros(C.1.variables,C.1.variables)
$ C.1.B = zeros(C.1.variables,C.1.variables)

! ------------------------------------------------------------------------
! Dof/Unknown Vector Definition
! ------------------------------------------------------------------------
$ C.1.name.1 = "i_I1"
$ C.1.name.2 = "i_component(1)"
$ C.1.name.3 = "v_I1"
$ C.1.name.4 = "v_component(1)"
$ C.1.name.5 = "u_2_circuit_1"

! ------------------------------------------------------------------------
! Source Vector Definition
! ------------------------------------------------------------------------
$ C.1.source.5 = "I1_Source"

! ------------------------------------------------------------------------
! KCL Equations
! ------------------------------------------------------------------------
$ C.1.B(0,0) = -1
$ C.1.B(0,1) = 1

! ------------------------------------------------------------------------
! KVL Equations
! ------------------------------------------------------------------------
$ C.1.B(1,2) = 1
$ C.1.B(1,4) = -1
$ C.1.B(2,3) = -1
$ C.1.B(2,4) = 1

! ------------------------------------------------------------------------
! Component Equations
! ------------------------------------------------------------------------
$ C.1.B(4,0) = 1
```

```
! ------------------------------------------------------------------------
! Additions in SIF file                                               (3)
! ------------------------------------------------------------------------
Component 1
  Name = "Coil1"
  Master Bodies Name = Coil
  Coil Type = "Stranded"
  Number of Turns = Real $ N_Coil1
  Resistance = Real $ R_Coil1

  ! Additions for 3D Coil
  Coil Use W Vector = Logical True
  W Vector Variable Name = String CoilCurrent e
  Electrode Area = Real $ Ae_Coil1
End

! ------------------------------------------------------------------------
! Sources in SIF                                                      (4)
! ------------------------------------------------------------------------

Body Force 1
  I1_Source = Variable "time"
        Real MATC "I1"
End

! ------------------------------------------------------------------------
! End of Circuit
! ------------------------------------------------------------------------
```

Source Time Dependencies (SIF Modifications)

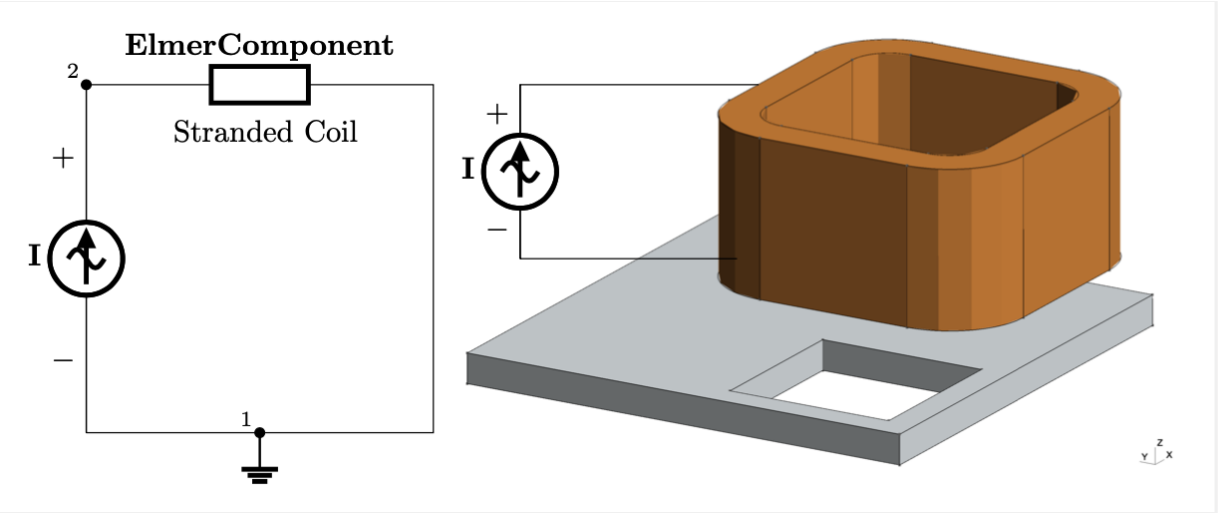
- The Circuit Builder is NOT a simulation tool. All simulation tweaks must be done within the .sif
- Harmonic
  - The Angular Frequency is required in the Simulation Block

```
Simulation
...
  Angular Frequency = Real $ 2*pi*f
...
End
```

- Transient
  - Time dependent functions can be implemented within the Body Force 1 definition. An example of a sinusoidal source:

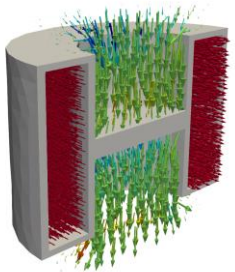
```
Body Force 1
  Source = Variable "time"
      Real MATC "amplitude*sin(omega*tx(0))"
End
```

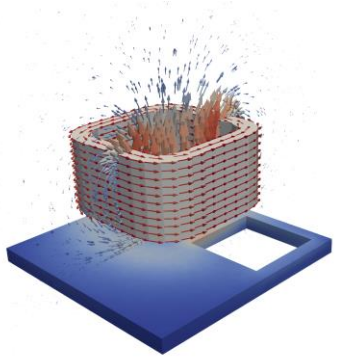
Elmer Circuit Builder Demo

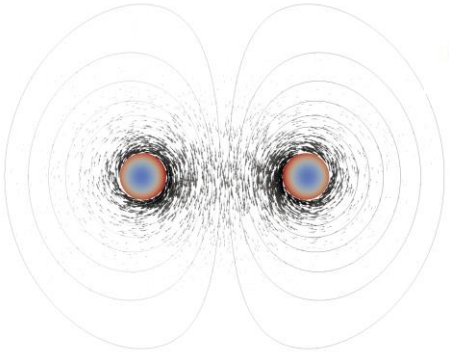


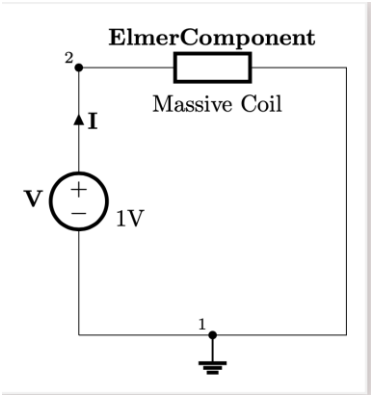
Modeling Coils using the Circuit Builder Summary

- Circuit Builder Library:
  - https://pypi.org/project/elmer-circuitbuilder/
  - Main template
  - Transient sources
  - Include in .sif file
- Coil modelling examples:
  - https://github.com/ElmerCSC/elmer-elmag/tree/main/CircuitBuilder



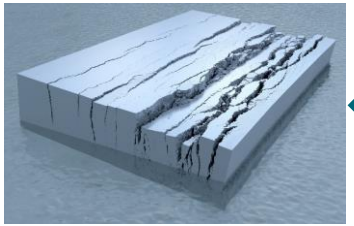


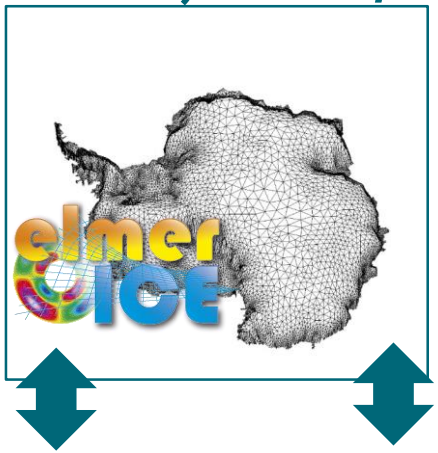




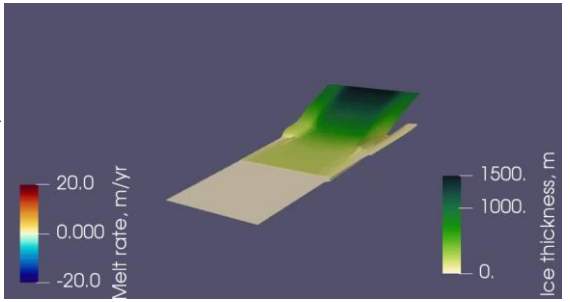
Upcoming webinar: 15.12. Geophysical applications beyond (but in connection to) Elmer/Ice

Coupler to DEM (HiDEM)

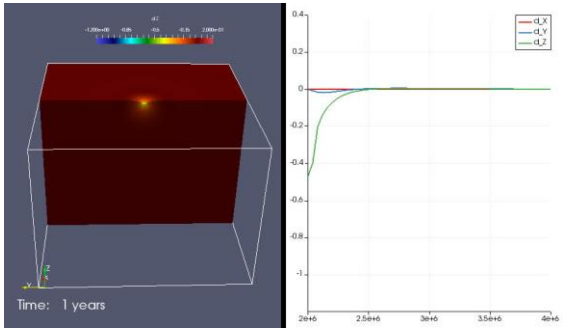




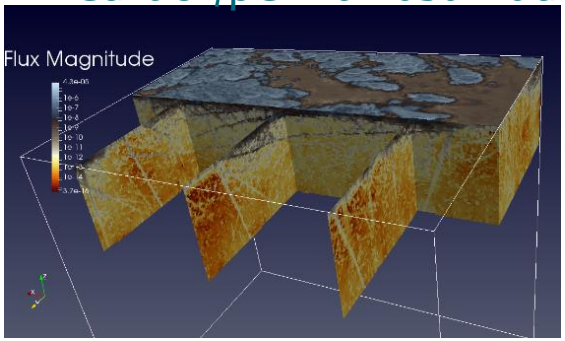
Ice-ocean coupler (FISOC)



Visco-elastic earth model



Bedrock/permafrost model



Registration Link: https://ssl.eventilla.com/event/xdDlJ

Most important Elmer resources

- http://www.csc.fi/elmer
  - Official Homepage of Elmer
- http://www.elmerfem.org
  - Discussion forum, wiki, elmerice community
- https://github.com/elmercsc/elmerfem
  - GIT version control (the future)
- http://youtube.com/elmerfem
  - Youtube channel for Elmer animations
- http://www.nic.funet.fi/pub/sci/physics/elmer/
  - Download repository
- Further information: jonathan.velasco@csc.fi

Thank you for your attention!


facebook.com/CSCfi


twitter.com/CSCfi


linkedin.com/company/csc---it-center-for-science


github.com/CSCfi