



LIVIA
Laboratório Interdisciplinar Virtual de
Inteligência Artificial



Faculdade de
Tecnologia
UERJ-Resende

Apostila introdutória ElmerGUI

A fonte de entrada para a descoberta
dos softwares de simulação de fonte aberta.

*Arthur Oliveira
Vahid Nikoofard*

1^a edição
Resende, RJ

Sumário

| | |
|---------------------------------------|-----------|
| 1. Introdução | 3 |
| 2. Método de Elementos Finitos | 4 |
| 3. Instalação dos Softwares | 7 |
| 4. Predefinições | 11 |
| 5. Ex. 1 - Chaminé | 15 |
| 6. Ex. 2 - Campo Magnético | 23 |
| 7. Ex. 3 - Tubo de Metal | 29 |

1. Introdução

Os softwares de código aberto têm desempenhado um papel fundamental na democratização do acesso a ferramentas poderosas e inovadoras em diversas áreas do conhecimento. Essas ferramentas, desenvolvidas e compartilhadas pela comunidade, oferecem uma alternativa viável aos programas comerciais, permitindo que pesquisadores, engenheiros e estudantes tenham acesso a recursos avançados de maneira acessível.

Neste contexto, o ElmerGUI surge como uma ferramenta notável para simulações numéricas e análises de elementos finitos. O Elmer é um software de simulação multifísica, desenvolvido pela comunidade de software livre e disponibilizado sob a licença GPL (General Public License), que permite sua utilização e modificação gratuita.

O ElmerGUI oferece uma interface gráfica intuitiva, facilitando a criação e execução de simulações numéricas complexas. Sua arquitetura modular permite a inclusão de uma ampla gama de modelos físicos e técnicas numéricas, tornando-o uma ferramenta versátil para resolver problemas em diversas áreas, como engenharia mecânica, eletromagnetismo, transferência de calor, entre outras.

Nesta apostila, iremos explorar os recursos e funcionalidades do ElmerGUI e programas que podem nos ajudar nas simulações fornecendo uma visão geral de sua interface e principais recursos. Além disso, iremos abordar exemplos práticos de simulações, fornecendo orientações passo a passo para a resolução de problemas reais.

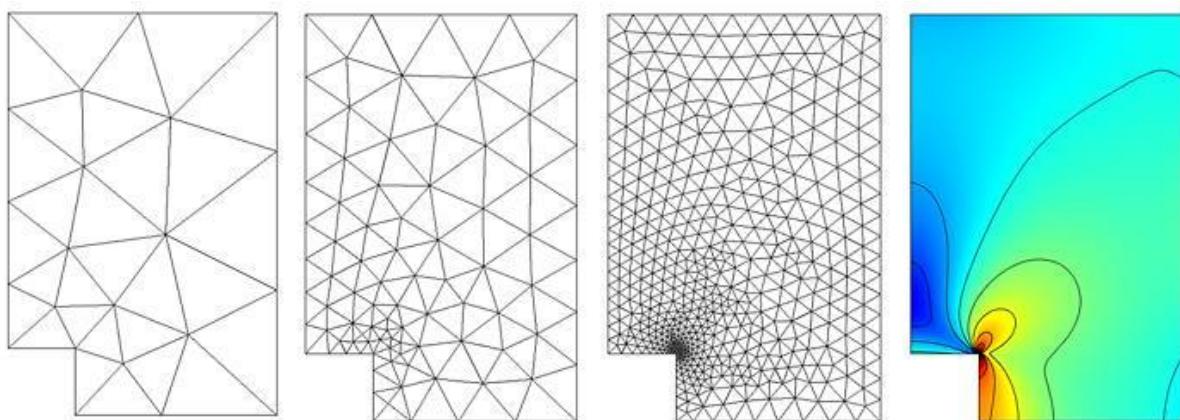
O objetivo deste material é capacitar os usuários a utilizar o ElmerGUI de forma eficiente e obter resultados precisos em suas simulações. Convidamos você a mergulhar no mundo da simulação numérica de código aberto e explorar todo o potencial que o Elmer tem a oferecer.

Vamos começar essa jornada rumo ao conhecimento e à descoberta das possibilidades que o ElmerGUI e os softwares de código aberto proporcionam.

2. Método de Elementos Finitos

Um passo fundamental para a compreensão da funcionalidade dos softwares de simulação é o método que os torna possíveis. A base de resolução desses programas são as chamadas Equações Diferenciais parciais, e o método numérico utilizado para a resolução delas é o Método de Elementos Finitos.

Aplicado ao problema de análise dos fenômenos físicos, o MEF divide o domínio do problema em elementos finitos ao criar uma malha com diversos pontos conectados em nós, com as soluções sendo calculadas dentro desses pontos. Os nós são o ponto principal do MEF, quanto maior sua quantidade sobre uma certa geometria, mais exata a simulação se torna, o conjunto desses nós se chama malha. Nota-se que é possível resolver uma EDP do método a mão, mas é praticamente impossível criar uma malha extensa e resolvê-la por si só.



(Amostra de uma malha menos precisa a uma cada vez mais exata e o resultado de uma simulação genérica por fim)

Ao ser estabelecida uma malha no domínio, uma EDP de formulação forte pode ser gerada, esta equação é muito utilizada por softwares sem interface gráfica como o Code-Aster, o próprio ElmerFEM e a biblioteca do Python FEniCS, nestes casos a Equação Forte do problema deve ser transformado em sua formulação fraca para ser utilizado.

O problema das formulações fortes e fracas pode ser algo de difícil entendimento a princípio, mas é a base do MEF. Na formulação forte, um problema

apresenta uma equação de governo e uma ou mais condições de contorno, esta não pode ser utilizada quando falamos de simulações pois afeta as condições de contorno utilizadas, além de apresentar possíveis erros de maneira elevada.

No exemplo abaixo vemos uma equação em sua formulação forte, com equação de governo e condições de contorno abaixo.

- Ex: Barra elástica submetida a esforços axiais

- Equação de governo: $\frac{d}{dx} (EA \frac{du}{dx}) + b = 0$, para $0 < x < L$
- Condição de contorno 1: $\sigma(x = 0) = (E \frac{du}{dx})_{x=0} = -\bar{t}$
- Condição de contorno 2: $u(x = L) = \bar{u}$

Em que:

- E : Módulo de elasticidade
- $A(x)$: Área da seção transversal:
- $b(x)$: Carregamento distribuído
- $\sigma(x)$: tensão na barra
- $u(x)$: deslocamento
- \bar{t} : Tensão aplicada na extremidade livre
- \bar{u} : Deslocamento prescrito na extremidade

Através do chamado Método de Resíduos Ponderados, podemos integrar isto e transformar em sua formulação fraca, deixando claro que é matematicamente equivalente à equação inicial.

$$\int_{\Omega} W_l \left(\frac{d}{dx} (EA \frac{du}{dx}) + b \right) d\Omega = \int_{\Omega} W_l \left(\frac{d}{dx} (EA \frac{du}{dx}) \right) d\Omega + \int_{\Omega} W_l b d\Omega = 0$$

- Integrando por partes:

$$\int u dv = uv - \int v du \rightarrow \int_0^L f(x)g'(x) dx = f(x)g(x)|_0^L - \int_0^L f'(x)g(x) dx$$

- Se $f(x) = W_l$ e $g(x) = EA \frac{du}{dx}$

$$\int_0^L W_l \left(\frac{d}{dx} (EA \frac{du}{dx}) \right) d\Omega = W_l EA \frac{du}{dx}|_0^L - \int_0^L \frac{dW_l}{dx} EA \frac{du}{dx} dx$$

- Formulação fraca

$$W_l EA \frac{du}{dx} \Big|_0^L - \int_0^L \frac{dWl}{dx} EA \frac{du}{dx} dx + \int_{\Omega} W_l b d\Omega = 0$$

Neste método, integra-se a equação de governo multiplicado a W_1 que é a chamada função peso, a responsável por facilitar a forma como tratamos as condições de contorno na formulação fraca. Se não fosse por esse tratamento matemático, a forma de utilizar os softwares seria extremamente mais complexa, senão impossível.

No ElmerGUI, foco de nossa apostila, essas formulações não entram em consideração, uma vez que um dos pontos mais fortes do software é que ele mesmo as define com base nos pré-requisitos dados pelo operador, logo as aplicações destas formulações não serão abordadas, por mais que sejam extremamente interessantes para se operar os softwares sem interface gráfica já citados.

3. Instalação dos Softwares

A princípio trabalharemos com apenas dois softwares: O ElmerGUI e o Paraview. Com o Elmer resolveremos a simulação em geral, aplicando predefinições, condições de contorno, condições iniciais e as equações gerais a serem utilizadas, uma vez que o solver nos der o resultado numérico e os arquivos referentes à simulação feita, usaremos o Paraview apenas para que possamos ver de forma gráfica como a malha definida influenciou a geometria.

3-1. Elmer

Primeiramente vamos trabalhar com a instalação do Elmer em Windows, para instalação em Linux é indicado entrar em contato com o autor. Acessando o [site oficial do Elmer FEM](#) é possível identificar o link para a página de arquivos.

Windows

There is a nightly build for Windows both with and without mpi updated rather frequently at.

<https://sourceforge.net/projects/elmerfem/>
<http://www.nic.funet.fi/pub/sci/physics/elmer/bin/windows/>

Only 64-bit version is supported. Note that AMD there refers to the instruction set also applicable on Intel processors. When running the installer Windows may fail to run it and report "Windows protected your PC". Then choose "More info" and "Run anyway".

If you use the zip files, then you must manually set the environment variables. Search for "path" in your Windows system and choose "Edit environment variables for your account" and set "Variable" to "Value" as follows:

```
ELMER_HOME = c:\ElmerFEM-gui-nompi-Windows-AMD64  
ELMERSOLVER_HOME = %ELMER_HOME%\share\ElmerGUI  
ELMER_LIB = %ELMER_HOME%\share\elmersolver\lib  
PATH = %PATH%;%ELMER_HOME%\bin;%ELMER_HOME%\lib
```

Como utilizaremos um arquivo .exe executável, os avisos extras não entram em condireção. Clicando no segundo link disponível, uma lista de arquivos são apresentados, todos disponíveis para download, utilizaremos a versão GUI nompi.



[ElmerFEM-gui-nompi-Windows-AMD64.exe](#) 2023-05-30 02:23 151M

Uma vez a pasta instalada, serão disponibilizados três arquivos, no entanto, apenas o segundo é executável, deve-se dar dois cliques neste.

 ElmerFEM-gui-nompi-Windows-AMD64 09/03/2021 23.01 Application 152 604 KB

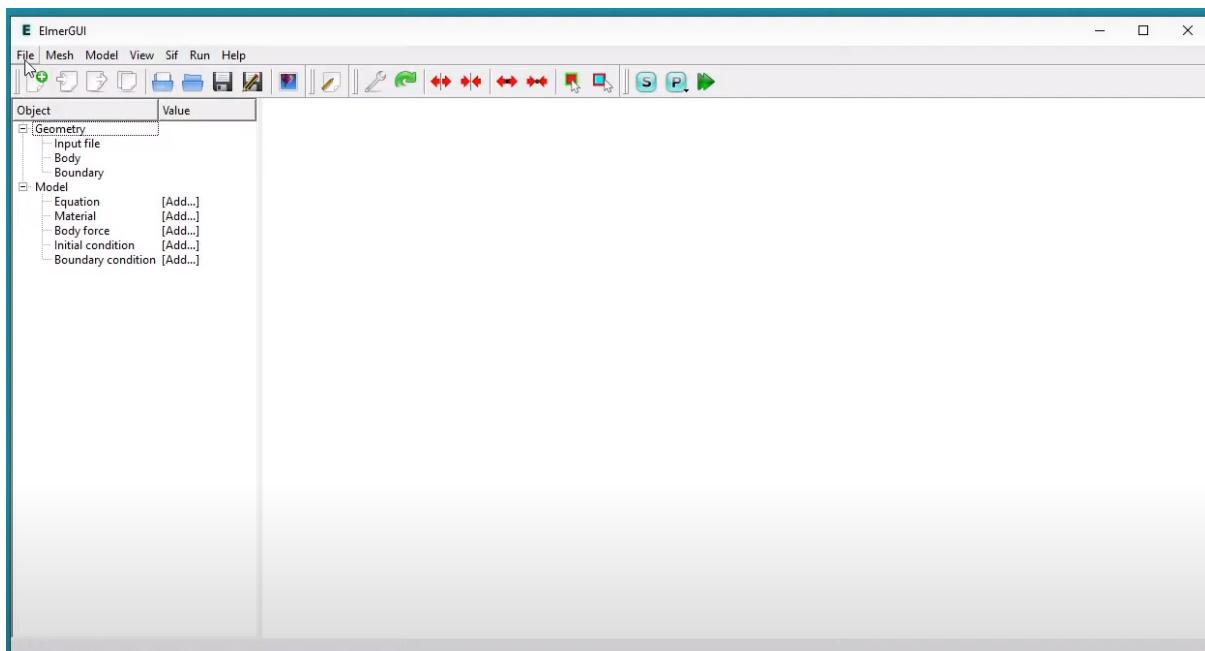
Todos os arquivos usados estão disponíveis no [Repositório do Github](#)

Laboratório Interdisciplinar Virtual de Inteligência Artificial

Após isso, a instalação irá começar, concorde com os termos de privacidade e siga a seguinte configuração de instalação:



O local de armazenamento é de escolha do usuário, desde que tenha espaço suficiente disponível para a instalação. Aguarde a instalação ser finalizada e conclua-a, a seguinte interface deve estar disponível quando abrir o programa:



3-2. Paraview

Passando para o programa que nos ajudará a ver de forma mais clara o resultado das simulações, temos o Paraview com a instalação tão simples quanto a do Elmer. O primeiro passo é acessar o [site oficial](#) do programa e a partir daí seguir para a [aba de downloads](#). Aqui ainda focaremos na versão de windows, usando a

Laboratório Interdisciplinar Virtual de Inteligência Artificial

versão 5.11 do software, usando o arquivo executável .msi, parecido com o tipo .exe, que também é executável e extremamente fácil de ser instalado.

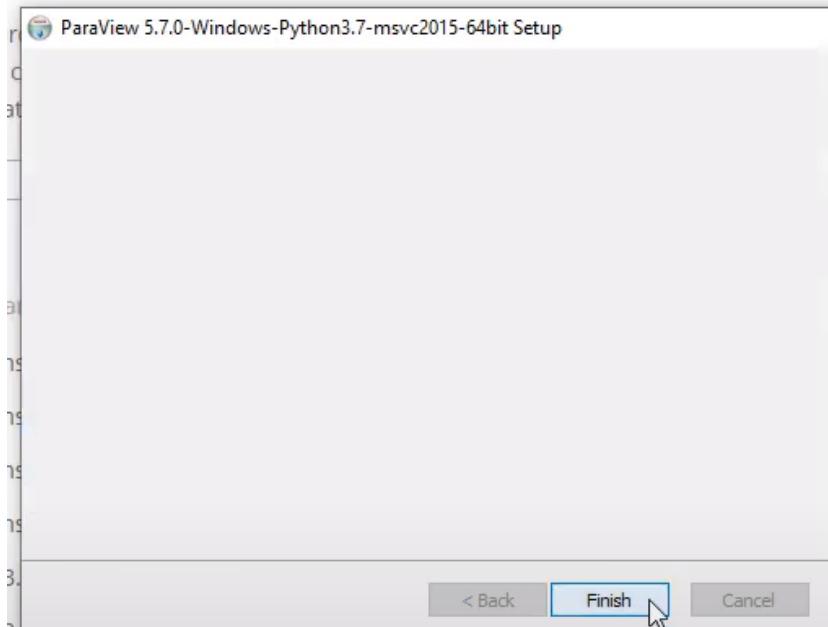
Version v5.11

ParaView

Full suite of ParaView tools, including the ParaView GUI client, ppython, pserver, and pbatch. Versions with MPI in the name require [MS-MPI](#). Unzip .zip packages in a directory with a very short path if you encounter an error message about a path being too long for some files.

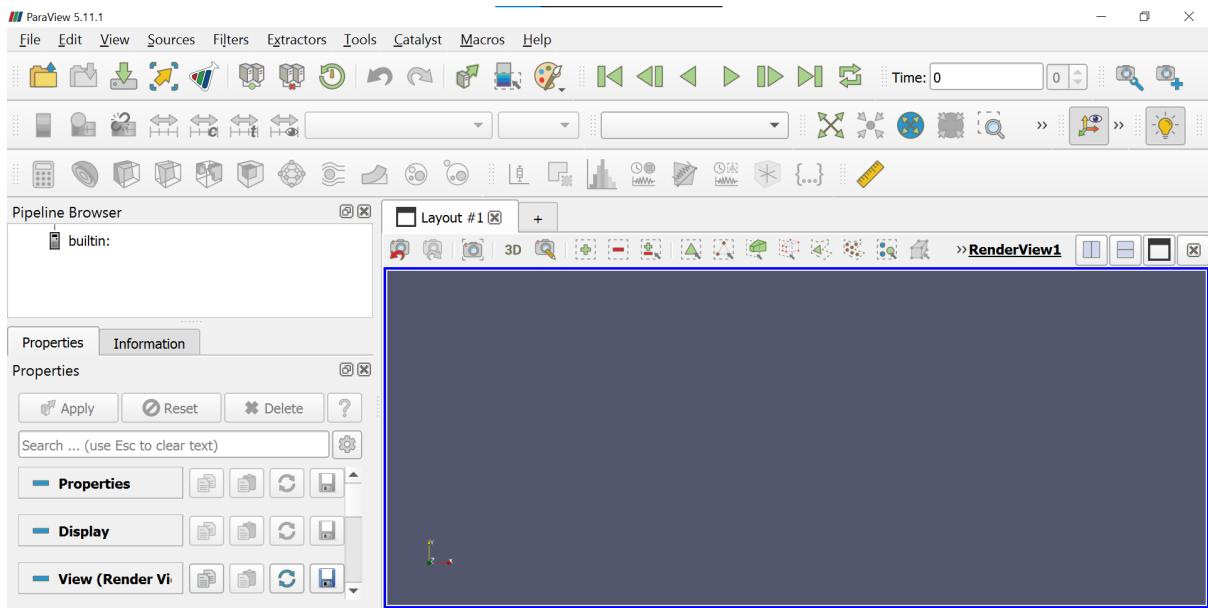
| ParaView-5.11.1-Windows-Python3.9-msvc2017-AMD64.msi | 2023-04-03 09:15 | 342.5M | |
|--|------------------|--------|--|
| ParaView-5.11.1-MPI-Windows-Python3.9-msvc2017-AMD64.msi | 2023-04-03 09:15 | 343.9M | |
| ParaView-5.11.1-Windows-Python3.9-msvc2017-AMD64.zip | 2023-03-30 14:37 | 812.5M | |
| ParaView-5.11.1-MPI-Windows-Python3.9-msvc2017-AMD64.zip | 2023-03-30 14:11 | 817.4M | |
| ParaView-5.11.0-Windows-Python3.9-msvc2017-AMD64.msi | 2022-11-15 11:18 | 342.7M | |
| ParaView-5.11.0-MPI-Windows-Python3.9-msvc2017-AMD64.msi | 2022-11-15 11:18 | 344.2M | |
| ParaView-5.11.0-Windows-Python3.9-msvc2017-AMD64.zip | 2022-11-14 16:50 | 812.8M | |
| ParaView-5.11.0-MPI-Windows-Python3.9-msvc2017-AMD64.zip | 2022-11-14 16:49 | 817.7M | |

Uma vez instalado e executado, escolha o local de instalação e aguarde até que seja finalizado, clique em “**finish**” para que o processo seja concluído.



Laboratório Interdisciplinar Virtual de Inteligência Artificial

Quando concluir o processo de instalação, pesquise pelo programa na ferramenta de busca do windows e o abra, ele deve ter a seguinte aparência:



Assim, já se obtém as ferramentas necessárias para as simulações que nos interessam.

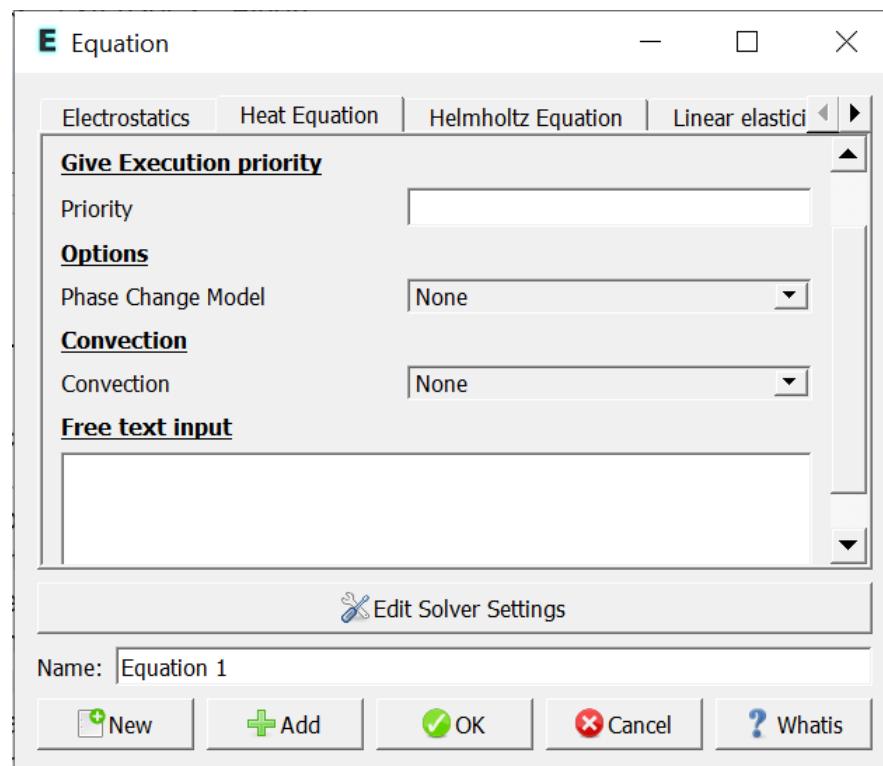
4. Predefinições

Antes de partirmos para exemplos práticos, vale conhecer um pouco mais sobre como funciona o Elmer, que é o verdadeiro foco desta apostila. Primeiro, deve-se ter em mente qual é o processo de simulação que se quer trabalhar com um pensamento baseado puramente na física, em outras palavras, deve-se ter em mente em qual equação você quer trabalhar.

4-1. Equation

Primeiro, deve-se saber qual é o processo de simulação que se quer trabalhar com um pensamento baseado puramente na física, em outras palavras, deve-se ter em mente em qual equação você quer trabalhar.

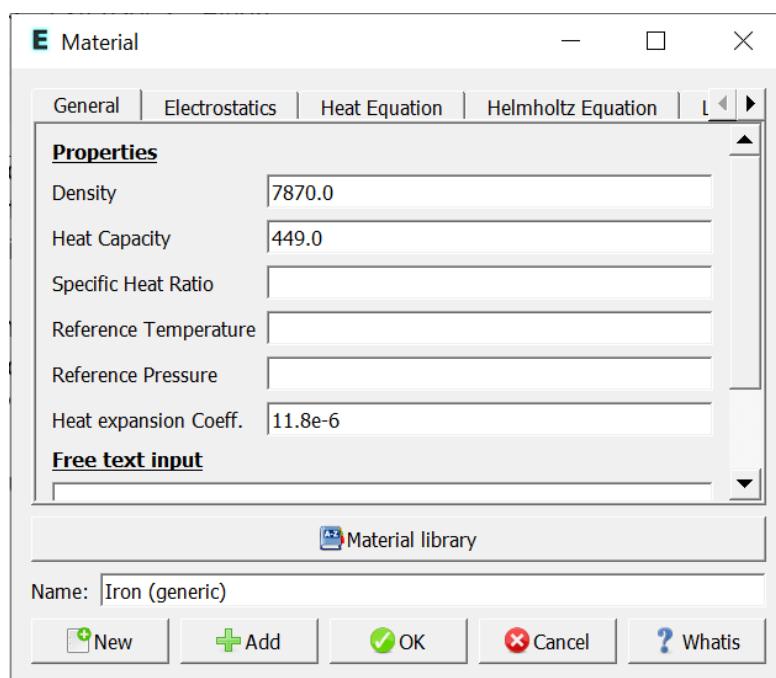
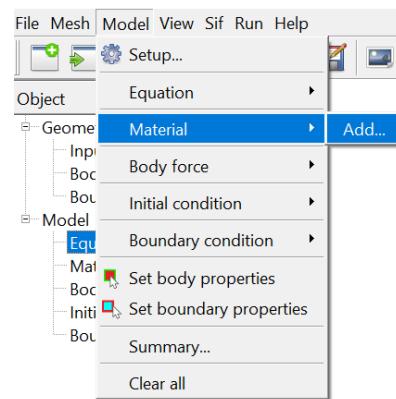
Neste software nós temos as equações de calor, de eletricidade, de elasticidade, de helmholtz e algumas outras que são menos usadas. Quanto mais abrangente for sua simulação, mais equações se usa. Para acessá-las basta acessar o menu superior e selecionar “**models**”, dentro da seção, selecionar “**equations**” e preencher “**active**” na equação escolhida.



4-2. Material

Ainda na seção “models” nós temos a definição dos materiais. Aqui há uma diferença interessante que o Elmer apresenta, há uma biblioteca já pronta com predefinições de materiais já muito utilizados em simulações e no próprio dia-a-dia como alumínio, aço, ferro, silício e até mesmo gases.

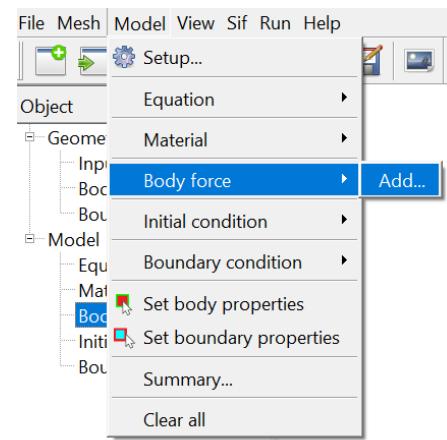
Essas predefinições incluem dados usados nas equações como Módulo de Young, Coeficiente de Poisson, densidade, Condutividade Térmica e etc. E ainda assim, caso o material não esteja na biblioteca do programa, é possível importá-lo, ou simplesmente definir os coeficientes que sejam de gosto do operador, acrescentando-o à malha.

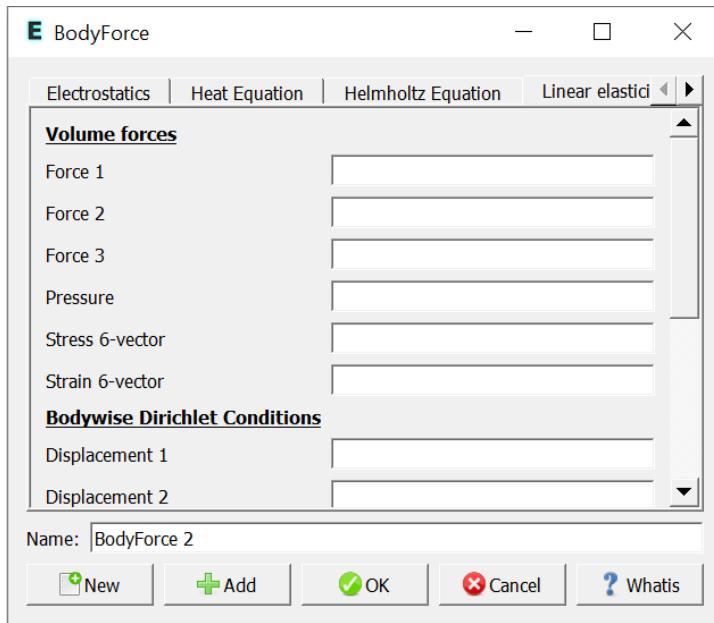


4-3. Body Force

A seção se refere a uma carga ou força aplicada à malha de forma geral, onde todos os nós são atingidos. Essa força atua em todo o volume do objeto.

O body force pode representar diversas fontes de carga, como gravidade, forças eletromagnéticas, forças de inércia, entre outras. Qualquer força externa que atinja a peça como um todo deve ser levado em consideração aqui.

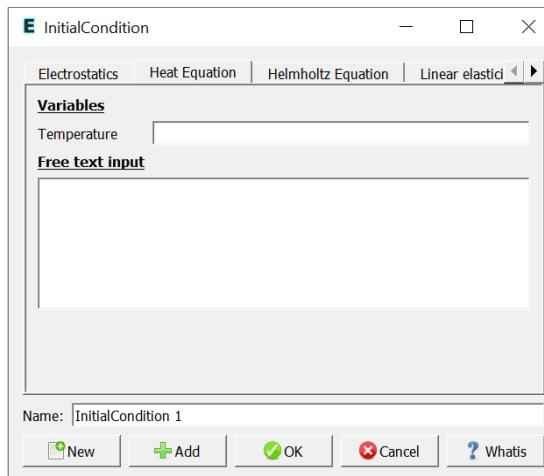
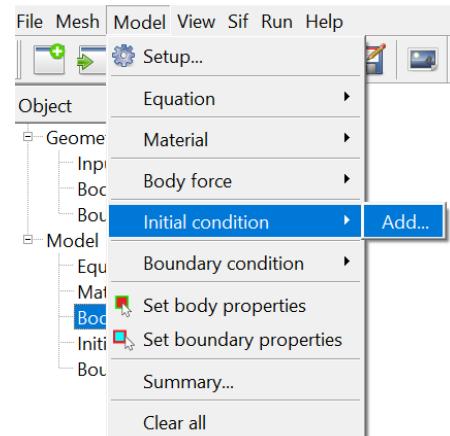




4-4. Initial Condition

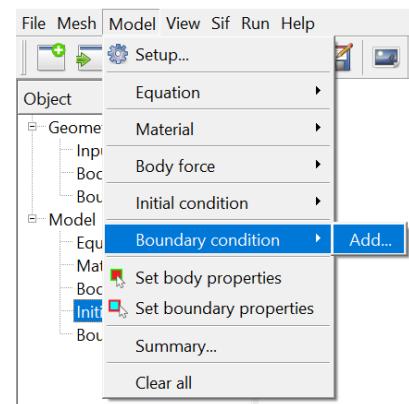
Podendo ser traduzido como “condição inicial”, nada mais é do que as características que a geometria possui no início ou antes da simulação ser iniciada. Aqui devem ser levados em consideração fatores como a temperatura, velocidade, concentrações, pressão e outros fenômenos físicos.

Em resumo, a condição inicial é a especificação dos valores dos campos físicos no início do tempo ou do processo de simulação, fornecendo a base para a análise da evolução desses campos ao longo do tempo.



4-5. Boundary Condition

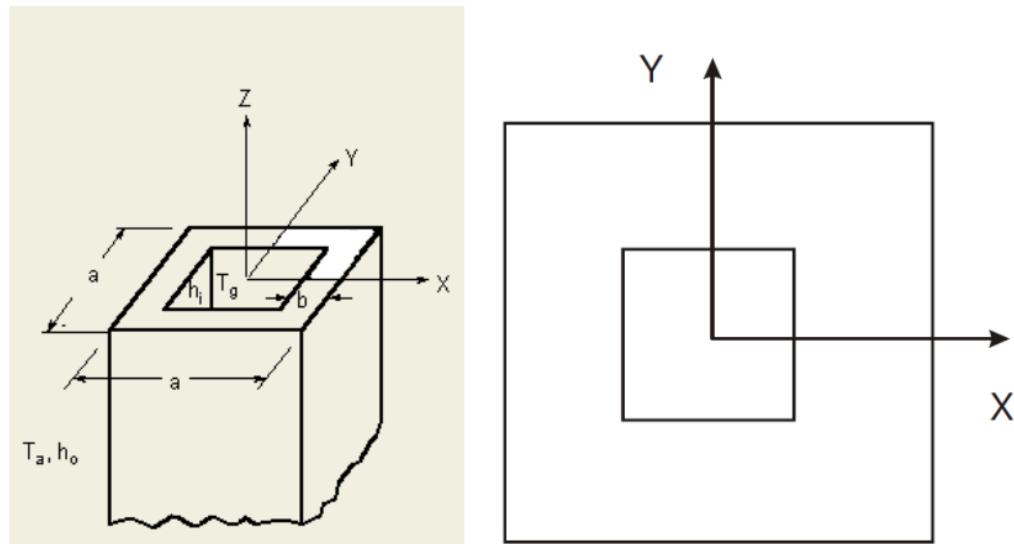
Essa é talvez a ferramenta e predefinição mais importante a ser utilizada. Também chamada de “condição de contorno”, as boundary condition exercem papéis parecidos com o do body force, mas agem apenas sob locais específicos da estrutura, elas são essenciais para que o solver do Emer possa desenvolver e resolver as equações diferenciais parciais (EDPs) e garantir uma solução única e coerente.



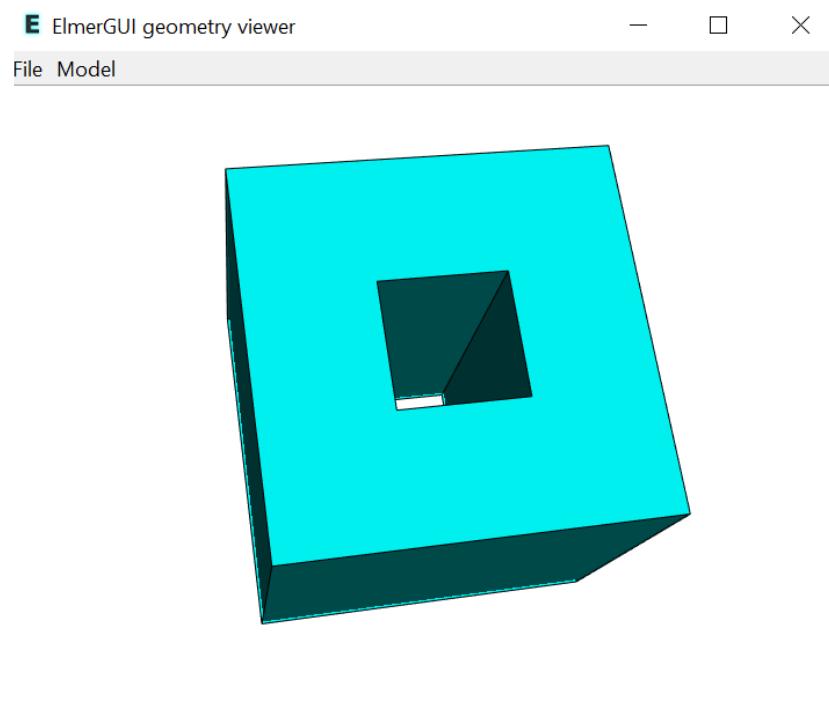
As condições de contorno podem especificar valores fixos de por exemplo, temperatura, pressão, velocidade, concentração, entre outros, em pontos ou regiões específicas das fronteiras. Por exemplo, pode-se impor uma temperatura constante em uma determinada superfície ou uma taxa de fluxo de massa fixa em uma saída de um sistema. Além disso, as condições de contorno podem estabelecer restrições de fluxo, como impedir a passagem de calor ou a transferência de massa através de uma superfície específica.

5. Ex. 1 - Chaminé

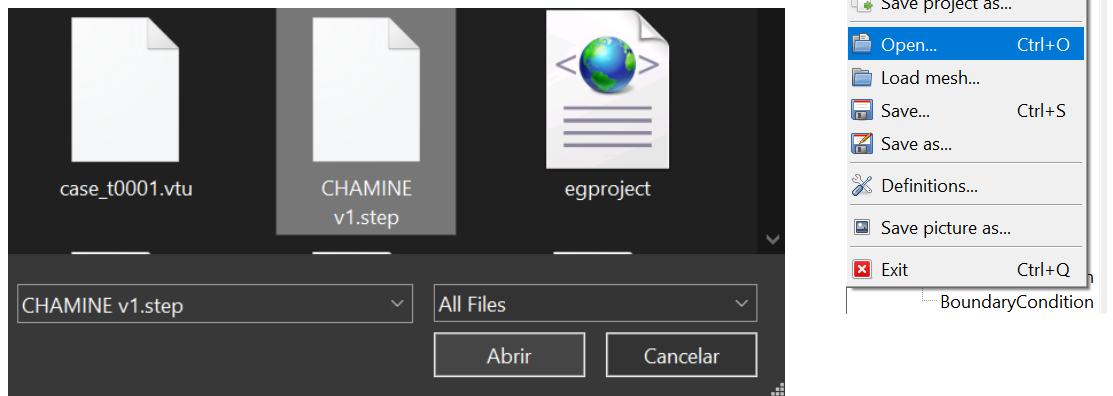
Uma vez entendido o funcionamento do software, vamos passar para um exemplo. Neste caso, vamos simular a variação de temperatura do topo de uma chaminé industrial que possui condutividade térmica de $K = 1,4 \text{ W/m}^2 \text{ K}$. A peça possui temperatura interna de $T_g = 100^\circ\text{C}$ e externa de $T_a = 30^\circ\text{C}$, sendo $a = 60 \text{ cm}$ e $b = 20 \text{ cm}$.



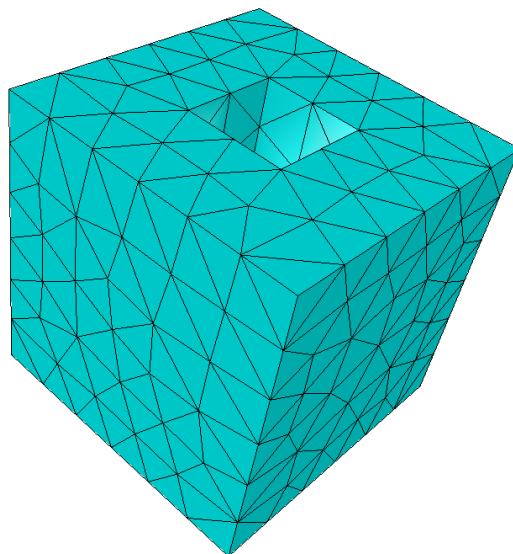
Montando a geometria que está com as medidas dadas:



Para importar o arquivo para o projeto no Elmer, seleciona-se a seção “file” e clica em “open”. O arquivo deve obrigatoriamente estar em .step ou qualquer outro formato aceito pelo Elmer, mas esse é o mais aceito em outros softwares de desenho.



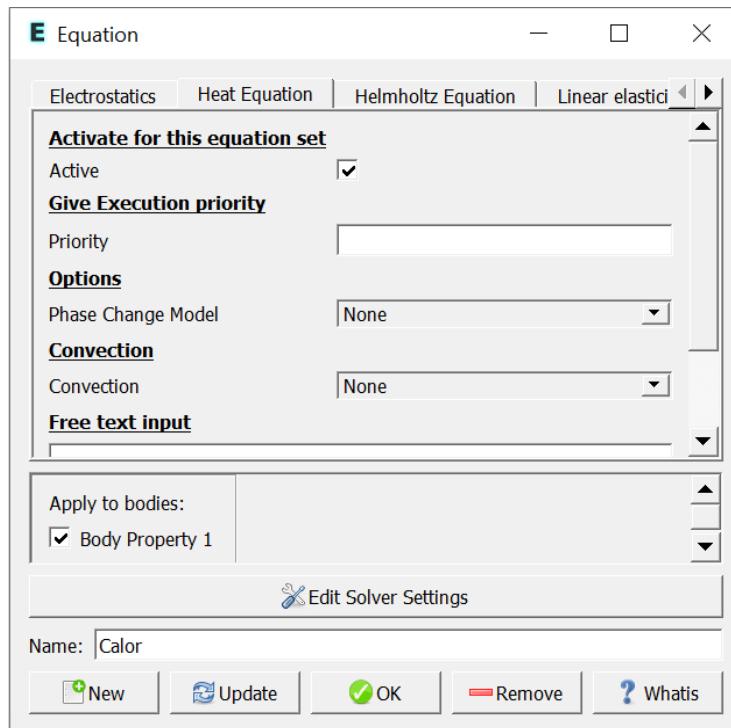
Com a geometria dentro do Elmer já é possível ver a malha criada automaticamente sobre ele, unindo os incontáveis nós sob a peça.



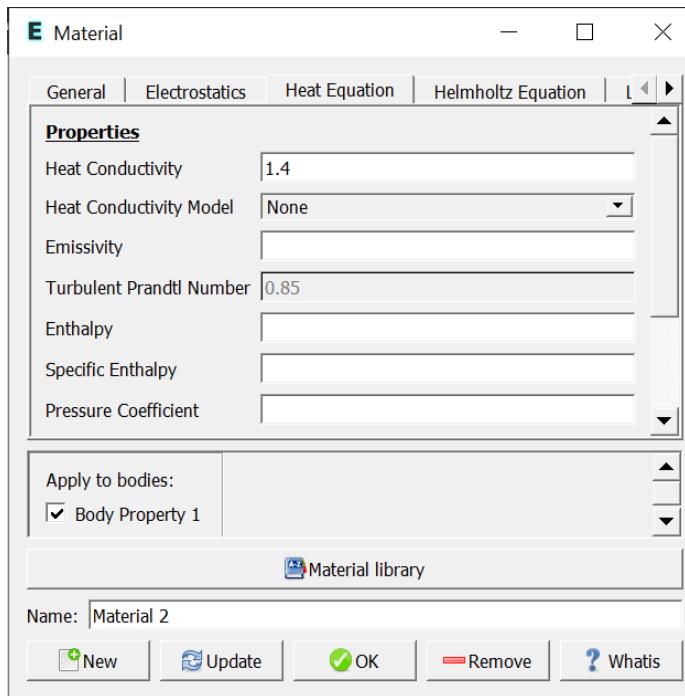
Com a geometria a postos, já podemos começar a preparar as definições. O primeiro passo é pensar no problema que foi dado, como estamos trabalhando com variação de temperatura, a única equação a ser utilizada é a de calor, uma vez que não temos nenhuma outra variável de alteração aqui.

Laboratório Interdisciplinar Virtual de Inteligência Artificial

Abrindo o meu de equações, podemos ativar a de calor e preencher a opção “Body Property 1”:



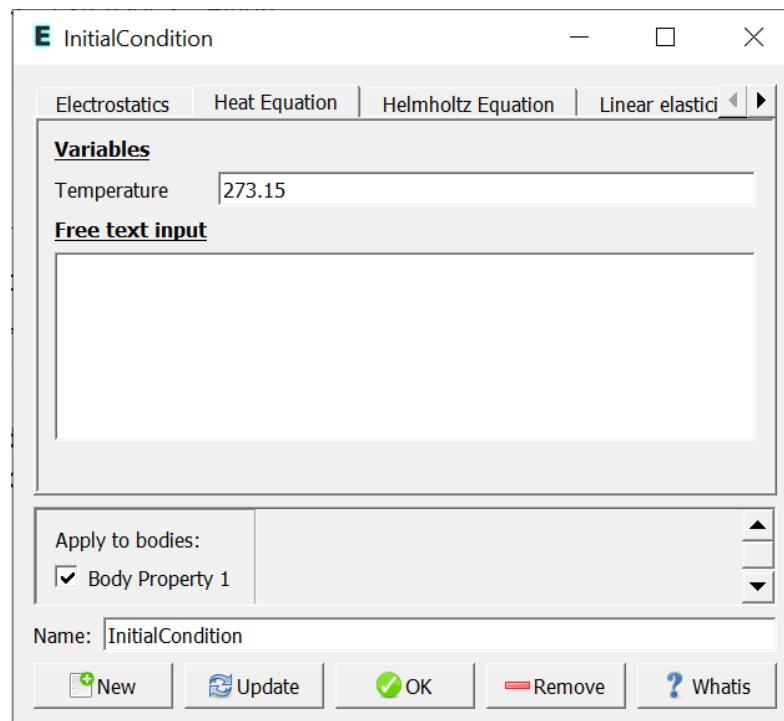
Seguindo a ordem, podemos já definir as propriedades do material, que aqui acabam sendo poucas, uma vez que o que está sendo avaliado é apenas a equação de calor. Como foi dado a condutividade térmica com $K = 1,4 \text{ W/m}^2 \text{ K}$ já podemos usar aqui. O valor de turbulent prandtl number já é dado, após isso, dê um nome ao material e clique em “**Apply to bodies**”.



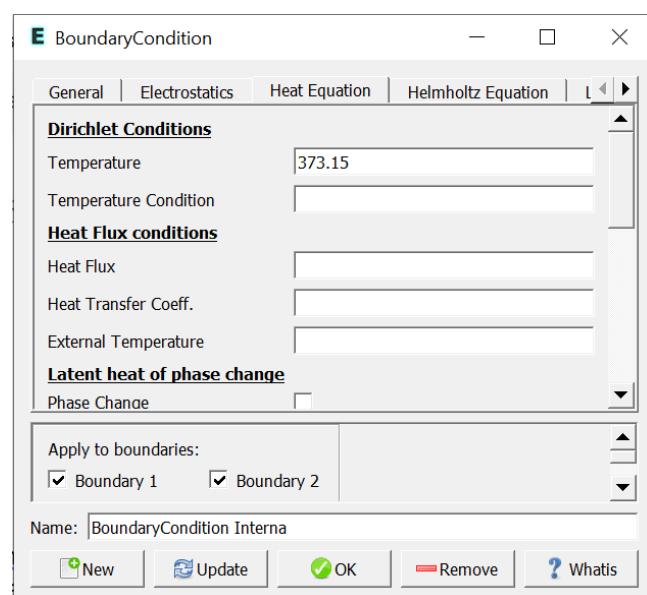
Todos os arquivos usados estão disponíveis no [Repositório do Github](#)

Com a equação e o material já definidos, podemos pensar no body force, aqui, não há nenhuma propriedade geral do sistema que deva ser levado em consideração como força externa além das condições de contorno. Uma vez tendo isso em mente, podemos partir para a Initial Condition.

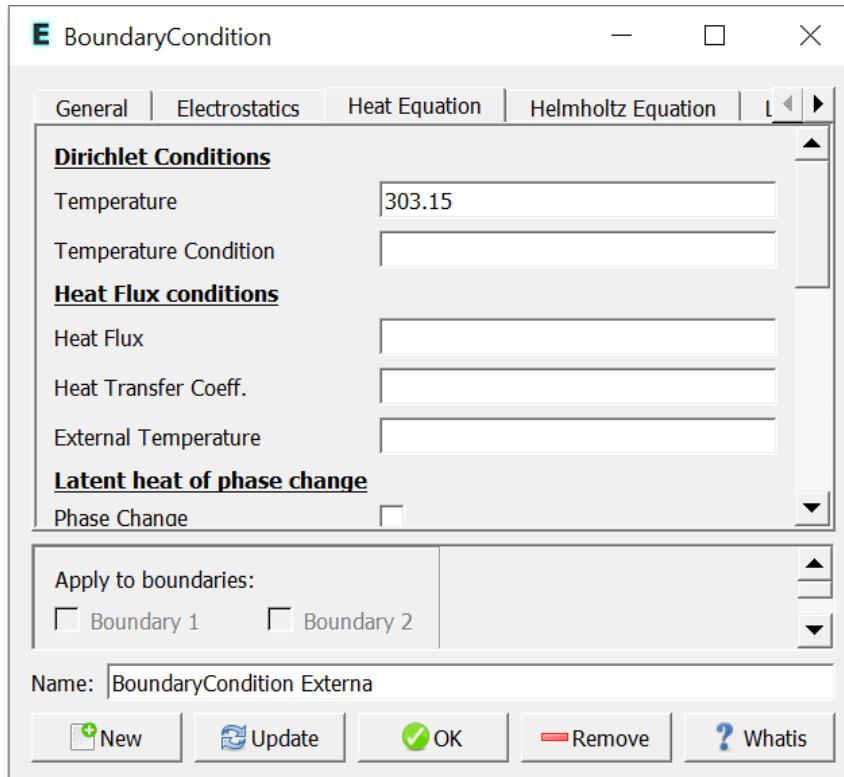
Como não foi especificado no problema a temperatura externa do ambiente, vamos colocar como condição inicial do sistema uma temperatura de 0°C. Todas as temperaturas devem estar convertidas em kelvin, sendo assim fica 273.15K, depois de determinar o valor basta preencher “**Apply to bodies**” e dar ok.



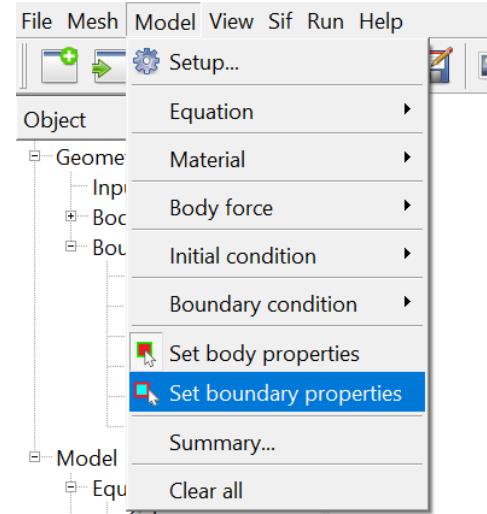
Por último vamos definir as duas condições de contorno existentes na peça e logo após aplicar a elas. Abrindo o menu de Bondery Conditions, vamos primeiro definir as condições internas, deixando a temperatura como 373.15K, que equivale a 100°C, defina um nome para diferenciar da próxima condição de contorno e selecione “**new**”.



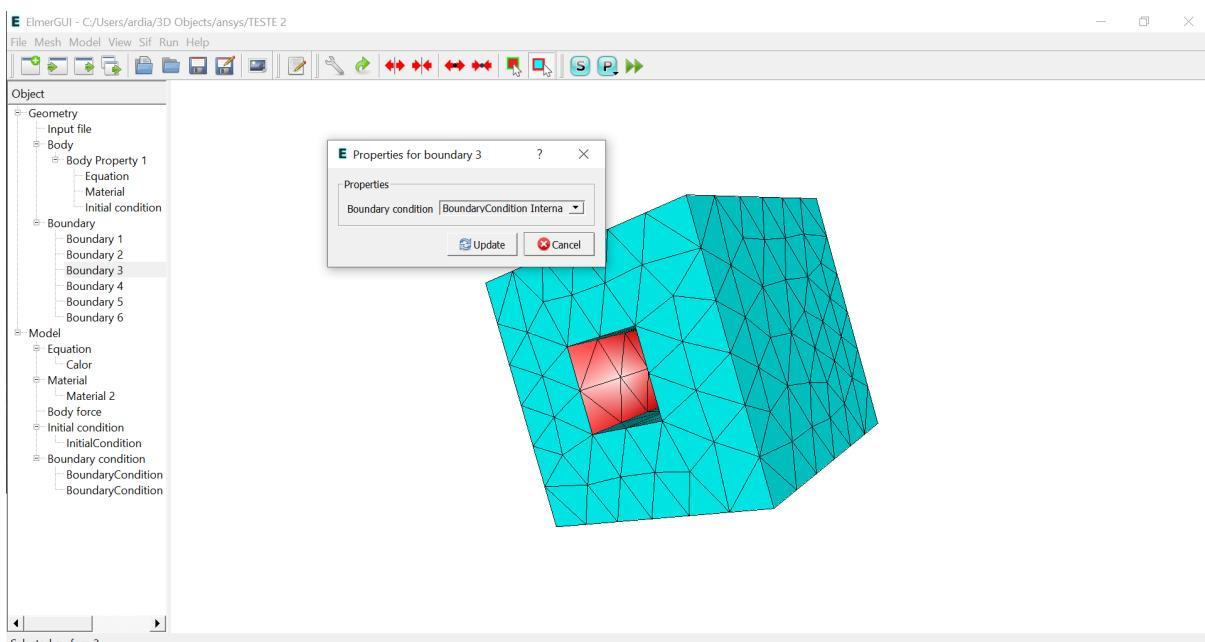
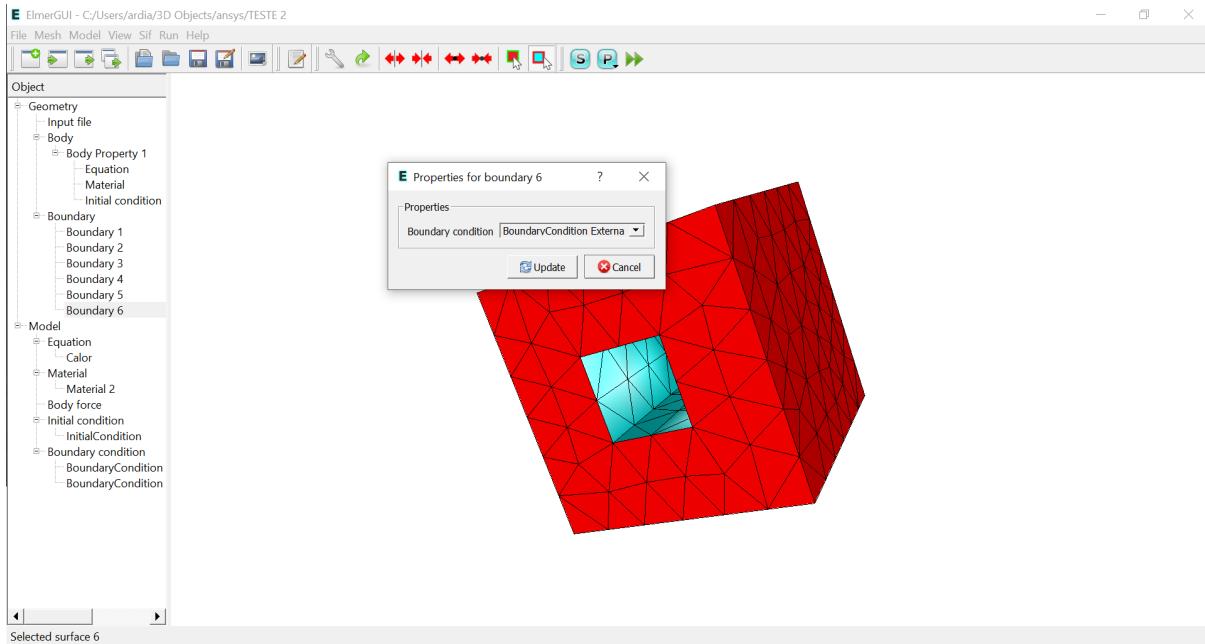
Em seguida, definindo uma nova condição de contorno, altere a temperatura como feito anteriormente para a temperatura externa dada pelo problema de 303.15K, que equivale aos 30°C. Novamente defina um nome para sua condição de contorno e clique em ok.



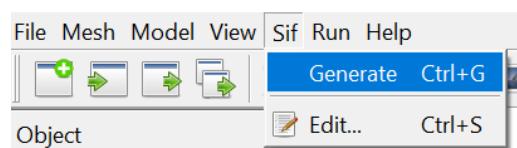
Tendo as condições de contorno definidas, agora vamos aplicá-las. Na seção de “models” temos a opção de “**set boundary condition**” clique ali e logo após dê dois cliques em cada área da geometria, selecionando a condição de contorno respectiva.



Laboratório Interdisciplinar Virtual de Inteligência Artificial



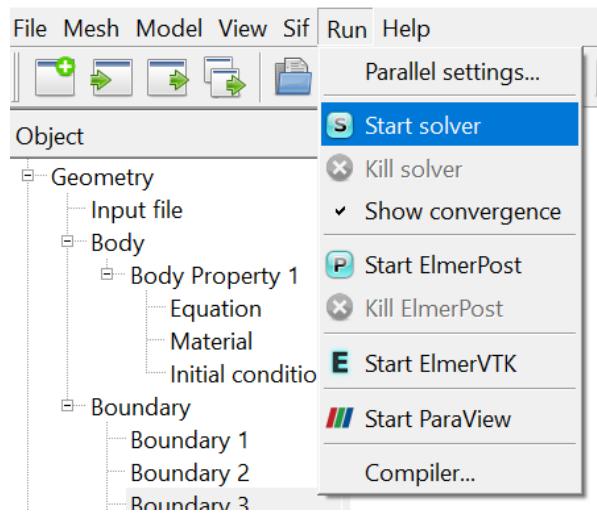
Com as predefinições concluídas, já podemos começar a gerar o código de resolução do solver. Para isso vamos até a seção “**Sif**” e clicamos em “**Generate**”



Tendo o código gerado, ao clicar em “**Edit**” é possível conferir o código com as definições feitas logo antes. É importante salvar o progresso até aqui no próprio menu “**Edit**”.

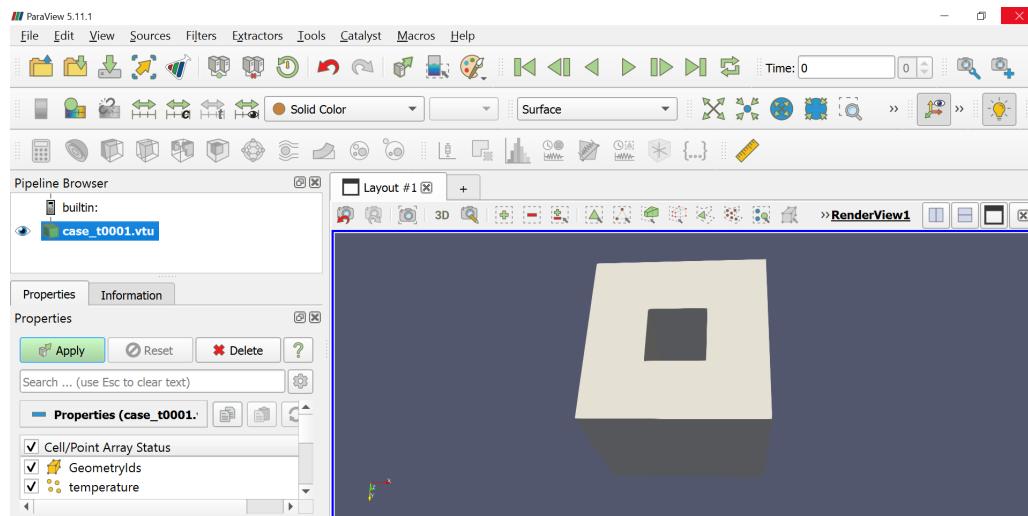
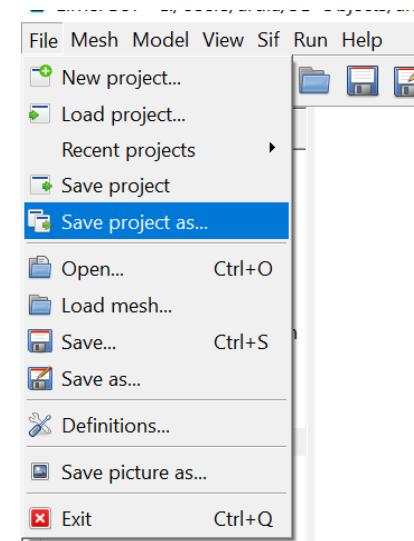
Todos os arquivos usados estão disponíveis no [Repositório do Github](#)

Partindo para os termos do solver, na seção “Run” há uma opção de mesmo nome, basta clicar ali para que a solução comece a ser gerada.



Com o solver correndo bem e o gráfico sendo montado, o que mais resta é salvar o projeto como um todo. Para isso basta ir em “Files” e selecionar a opção “Save project as...” para que o conjunto de arquivos seja salvo na pasta da escolha de quem o faz. Em meio a esses arquivos existe um .vtu, é este que usaremos no paraview para visualização do resultado gráfico.

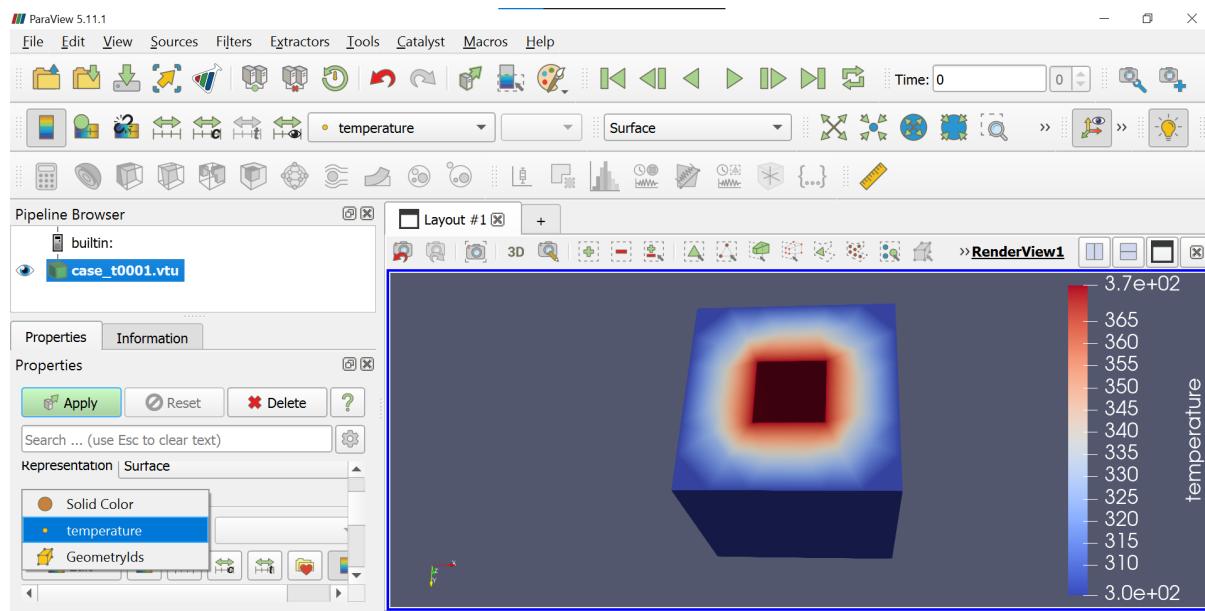
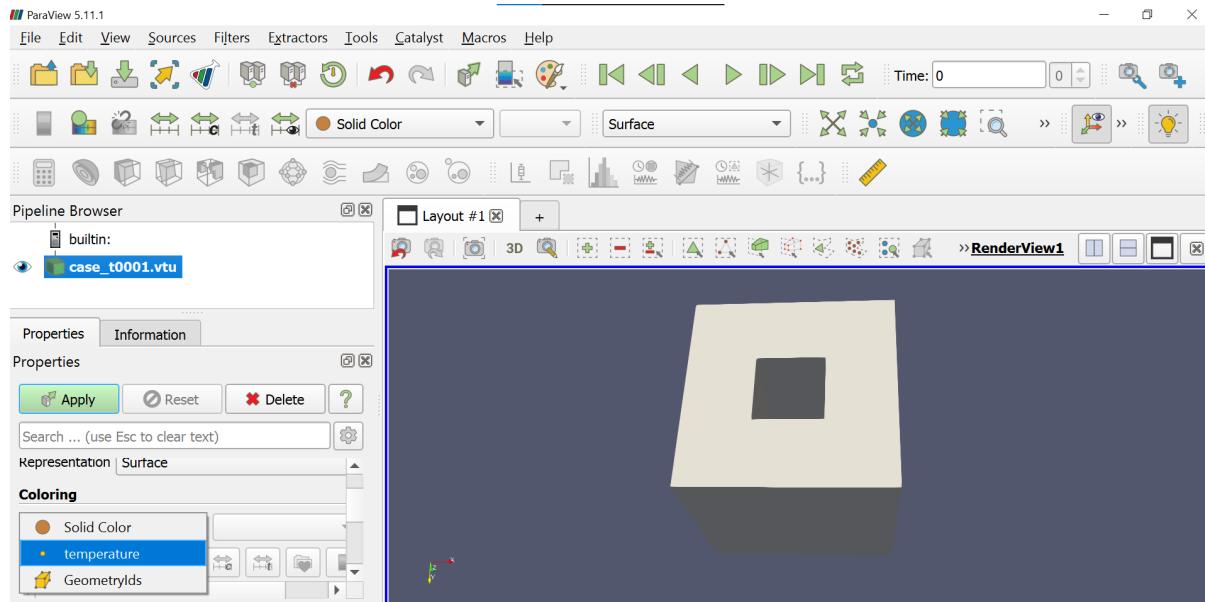
Abrindo o arquivo, primeiro vamos clicar o olho fechado em “Pipeline Browser” no menu do canto esquerdo para possibilitar a visualização do nosso exemplo simulado.



Todos os arquivos usados estão disponíveis no [Repositório do Github](#)

Laboratório Interdisciplinar Virtual de Inteligência Artificial

Em seguida como último passo deve-se dirigir ao menu abaixo intitulado “Properties”. Rolando a pequena tela para baixo é visível uma opção de “Coloring” escrito “Solid Color” logo abaixo; clique ali e selecione “temperature” para que assim termos o resultado correto.

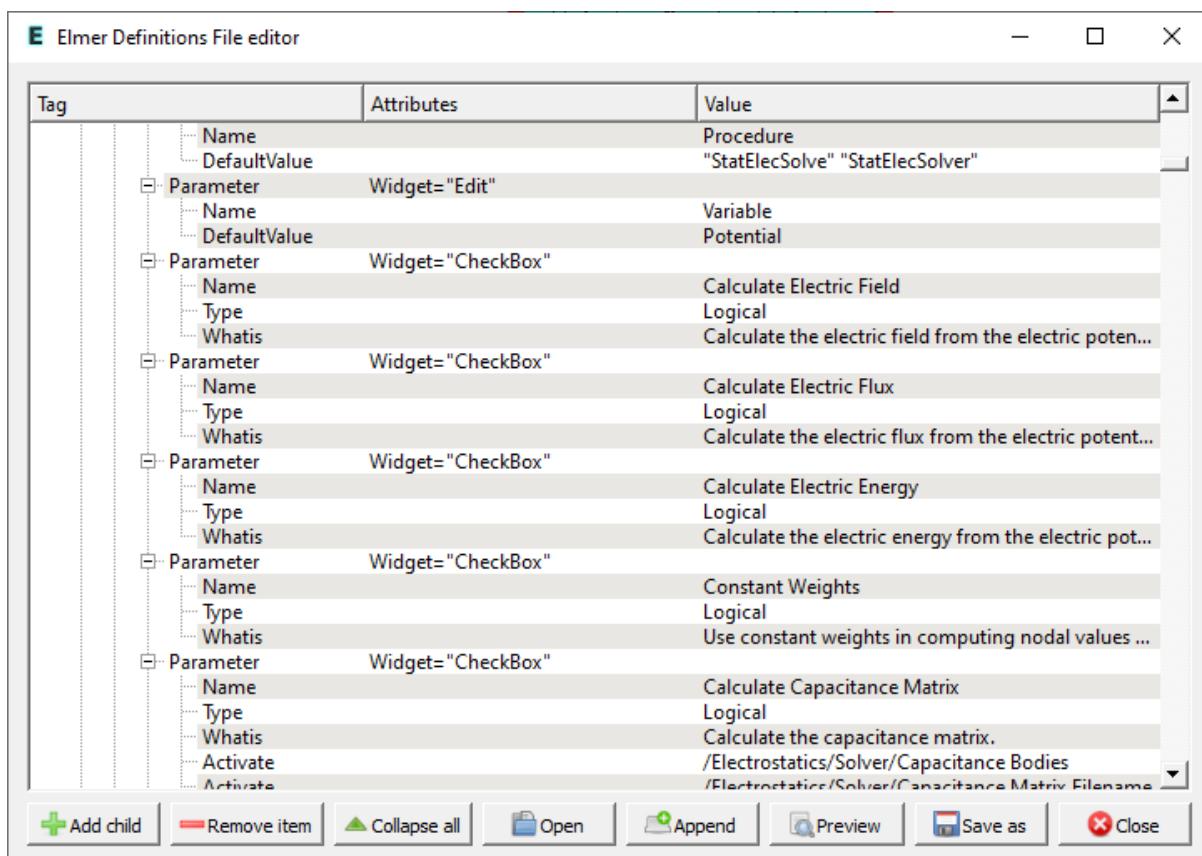


É perceptível que assim como definido na simulação, a temperatura interna da chaminé é muito maior que a externa, e devido à equação de calor vai se alastrando pela parte externa.

6. Ex. 2 - Campo Magnético

Mudando um pouco o assunto retratado, iremos fazer agora uma simulação a respeito do campo magnético de um ímã simples, entretanto, o software básico do Elmer não nos oferece uma equação que satisfaça essa simulação e por isso vamos importá-la. No repositório do github está disponível o arquivo magnetodynamics2d.xml, baixe-o e siga o caminho abaixo.

File
Definitions
Append -> magnetodynamics2d.xml



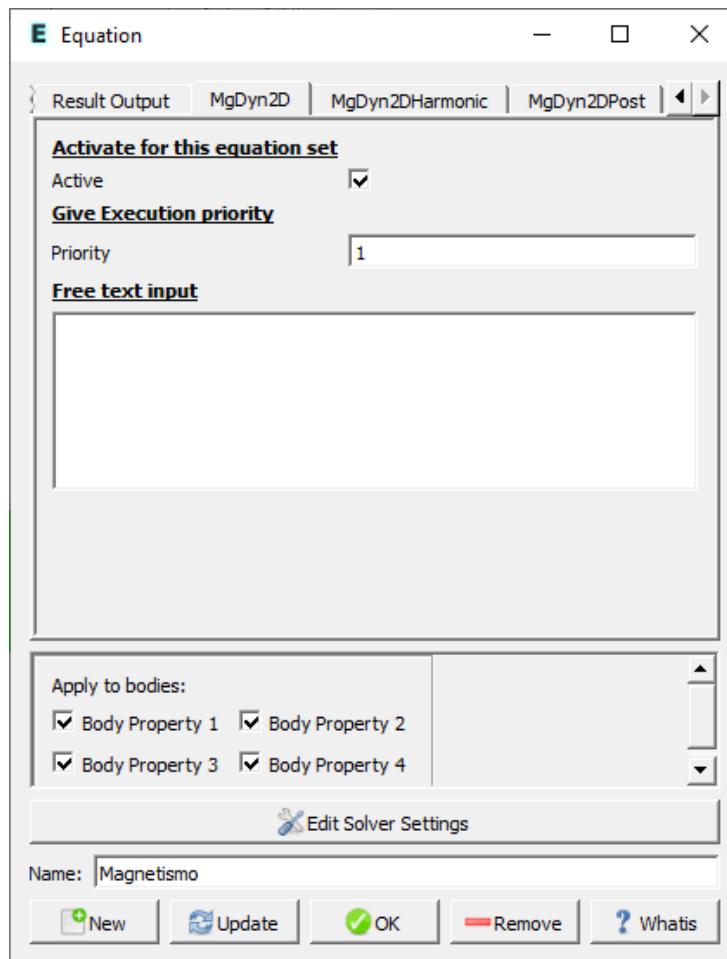
Já tendo a geometria, vamos seguir o mesmo passo do exemplo anterior, importando-a.

File
Open -> horseshoe.msh

Seguindo, vamos definir as configurações da malha. O primeiro passo é ativar a equação a ser utilizada, para isso selecionamos:

Model
Equation
Add

E utilizando a equação MgDyn2D, dê “**Active**” e defina o nome como “**Magnetismo**” ou qualquer outro de sua preferência, selecionando “**Apply to bodies**” em todos os disponíveis



Em seguida vamos definir os materiais utilizados, dessa vez serão quatro diferentes. O material com nome de Ar deve ser o definido como o de espaço vazio, o de Ferro é a parte do imã que faz curva, a parte estreita superior seria o Ferro Maior e a parte inferior, Ferro menor.

Model

Material

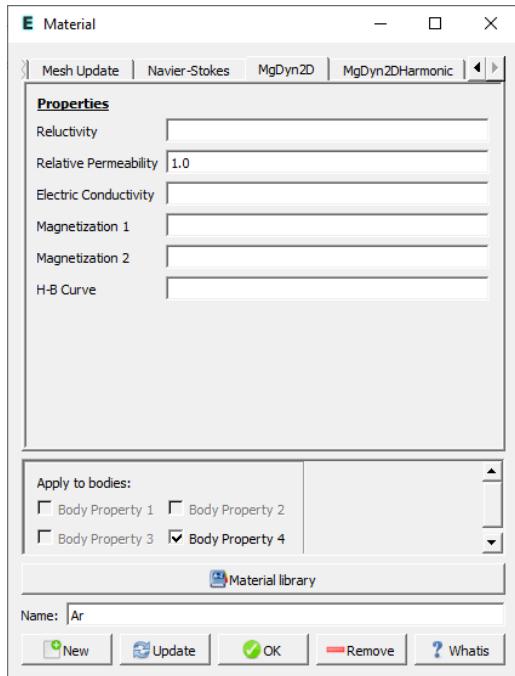
Name = Ar

MgDyn2d

Relative Permeability = 1.0

Add

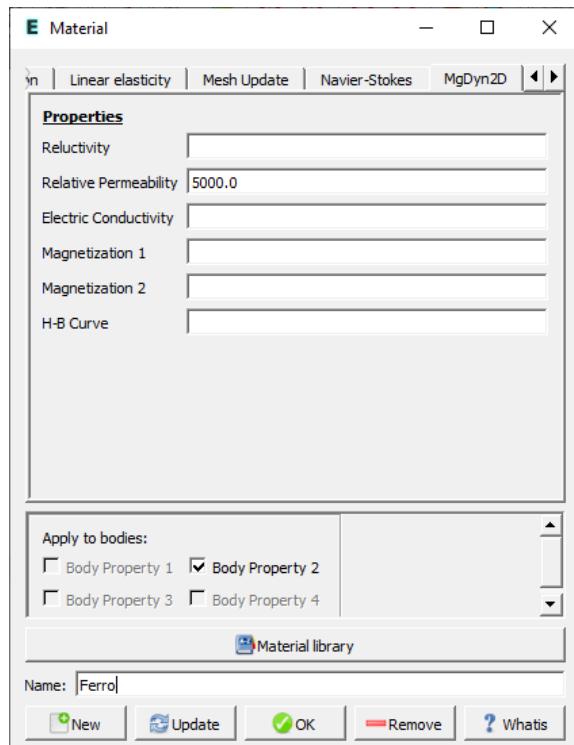
New



Model
Material
Name = Ferro
MgDyn2d
Relative Permeability = 5000.0

Add

New



Model

Laboratório Interdisciplinar Virtual de Inteligência Artificial

Material

Name = Ferro Maior

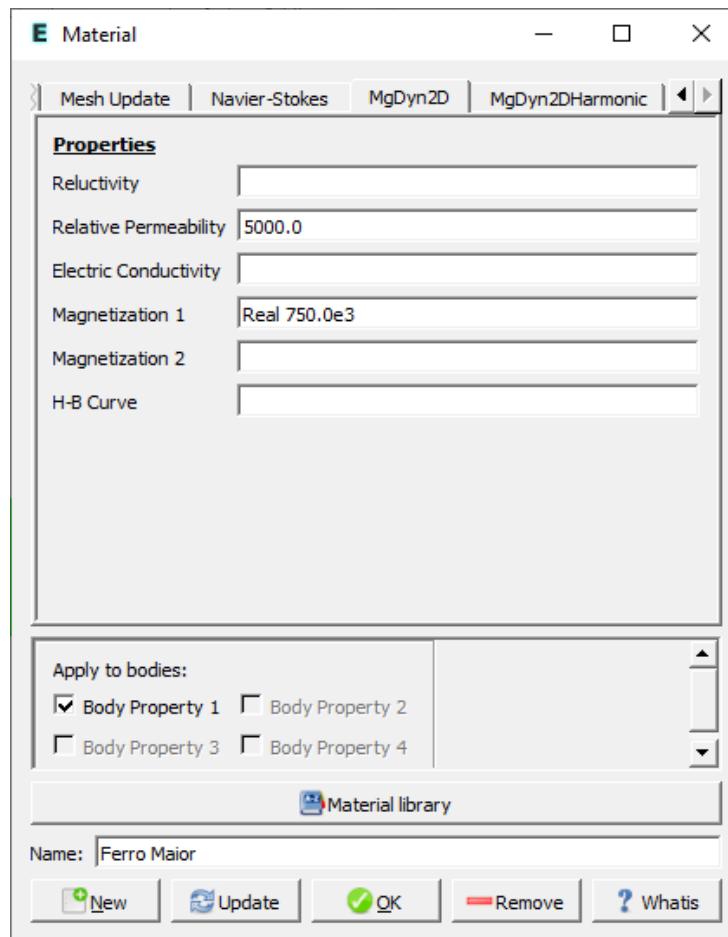
MgDyn2d

Relative Permeability = 5000.0

Magnetization 1 = Real 750.0e3

Add

New



Model

Material

Name = Ferro Menor

MgDyn2d

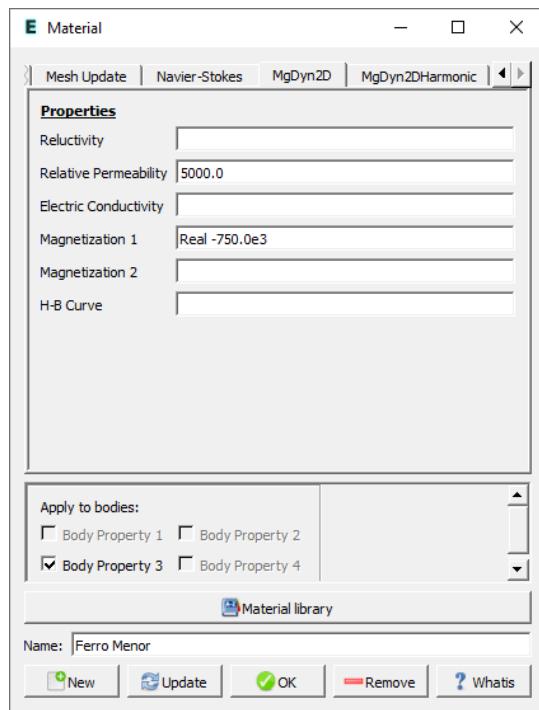
Relative Permeability = 5000.0

Magnetization 1 = Real -750.0e3

Add

New

Laboratório Interdisciplinar Virtual de Inteligência Artificial



Tendo os materiais definidos e aplicados a suas respectivas superfícies, vamos definir a condição de contorno.

Model

Boundary Condition

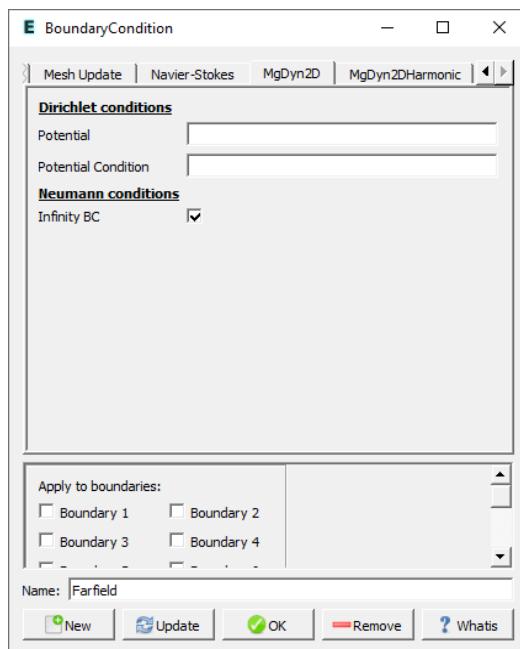
Name = Farfield

MgDyn2d

Infinity BC = True

Add

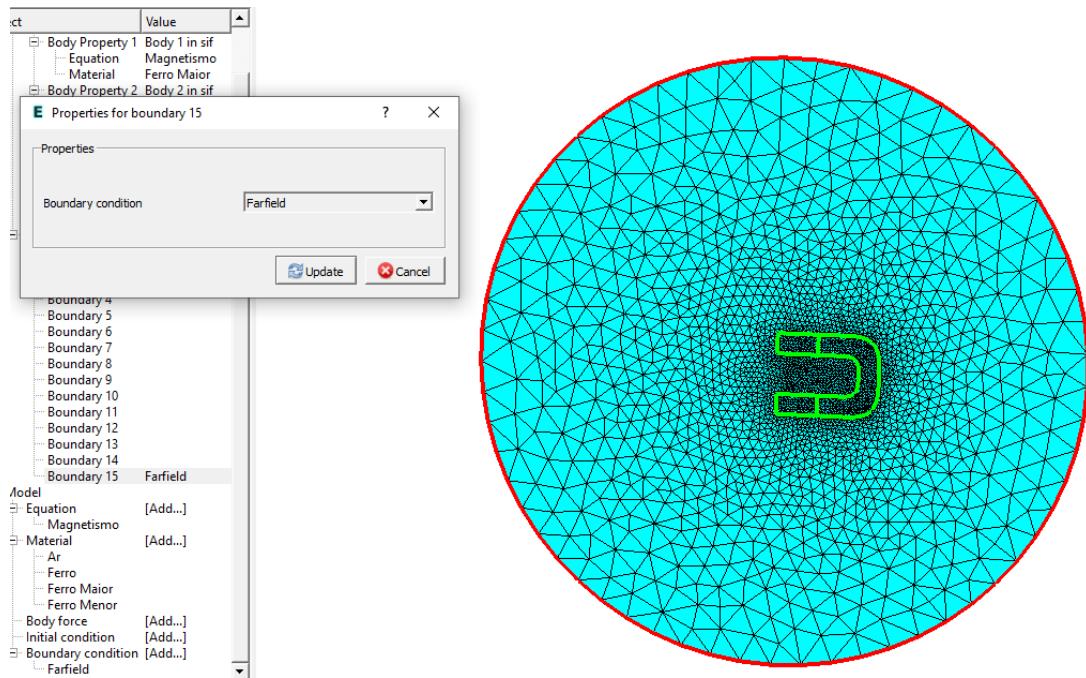
New



Todos os arquivos usados estão disponíveis no [Repositório do Github](#)

Laboratório Interdisciplinar Virtual de Inteligência Artificial

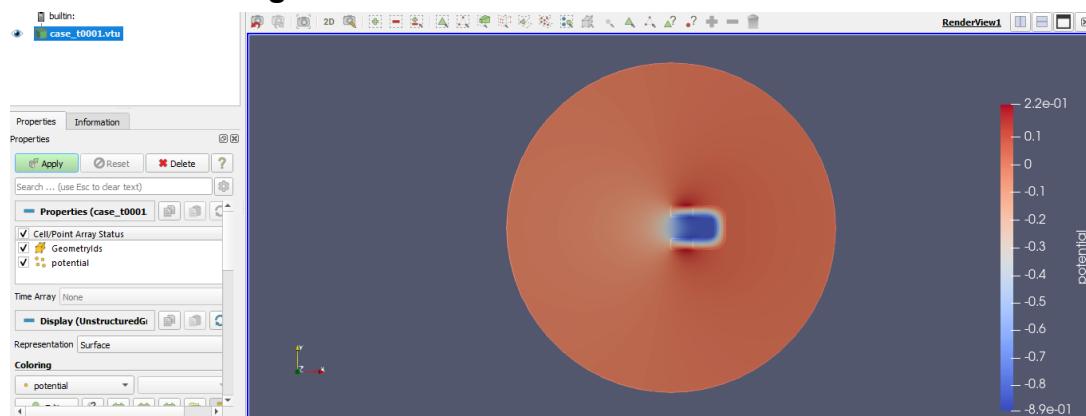
Essa condição de contorno nós aplicaremos aos quatro cantos de nosso campo magnético através da ferramenta “**Set boundary properties**”.



Com nossa malha definida, já podemos gerar o código para o solver com “**Generate**” na seção “**Sif**”.



Assim, salvamos o projeto na seção “**File**” e através da seção “**Run**” pode-se selecionar o solver para que ele rode dando os resultados necessários. Obtendo nosso arquivo .vtu usaremos o mesmo processo que o anterior no Paraview, abrindo o arquivo obtido para esta simulação, selecionando “**Potential**” no menu “**Coloring**”



Todos os arquivos usados estão disponíveis no [Repositório do Github](#)

7. Ex. 3 - Tubo de Metal

Avançando ainda mais no software, agora vamos incluir duas equações diferentes na mesma simulação. Imagine um tubo em que passa água pelo seu interior; aqui, vamos simular a energia de calor causada por essa interação e a velocidade em que esse fluido passa por cada extremidade do tubo.

Para começar, vamos abrir a geometria referente a esse tubo curvo no elmer.

File

Open -> curved_pipe.grd

Como já dito anteriormente, vamos trabalhar com dois tipos de equações base dessa vez. Para a parte interna do tubo, usaremos ambas as equações, sendo selecionada essa em “Apply to bodies”.

Model

Equation

Add

Name = Calor e Fluido

Apply to Bodies = 1

Heat Equation

Active = on

Convection = Computed

Navier-Stokes

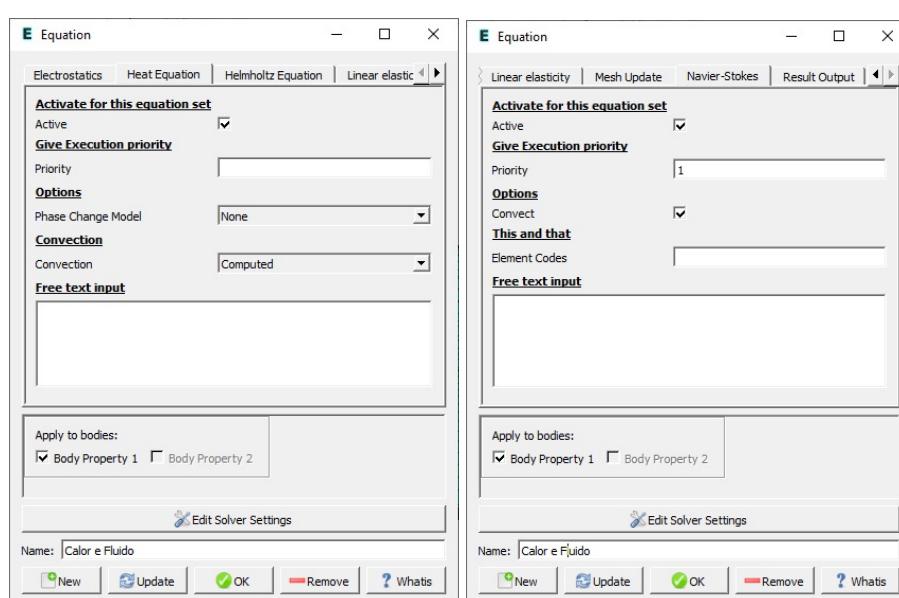
Active = on

Priority = 1

Edit Solver Setting

Linear System

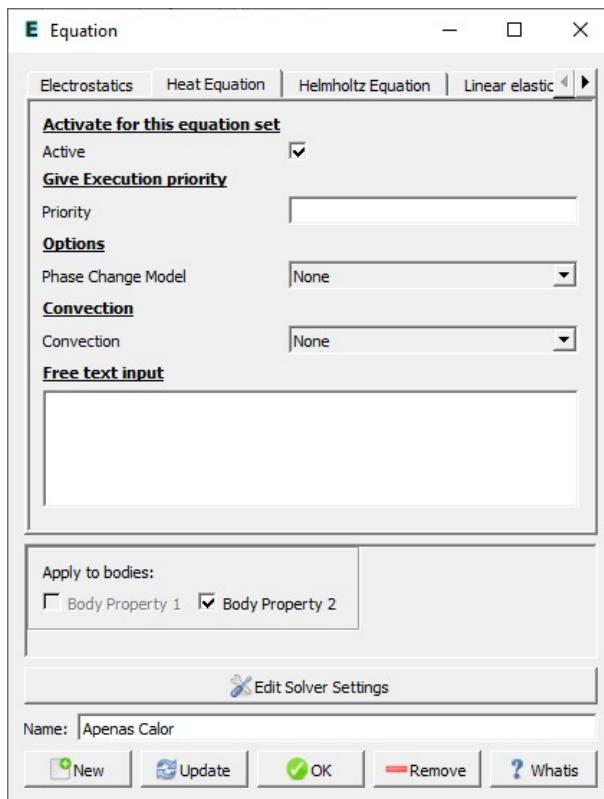
Preconditioning = ILU1



Laboratório Interdisciplinar Virtual de Inteligência Artificial

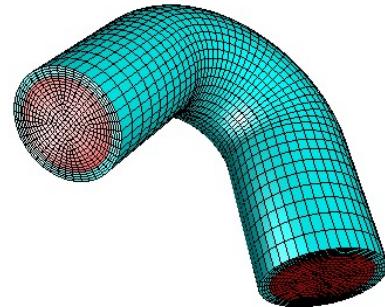
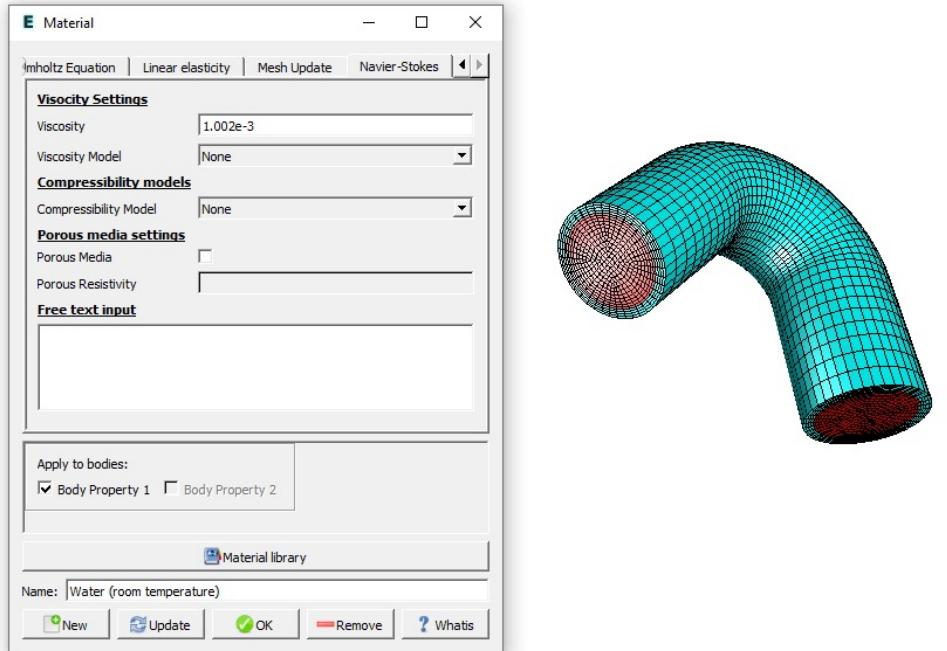
Para a parte externa, trabalharemos apenas com a equação de calor, utilizando:

```
Model
  Equation
    Add
      Name = Just Heat
      Apply to Bodies = 2
    Heat Equation
      Active = on
      Convection = None
```

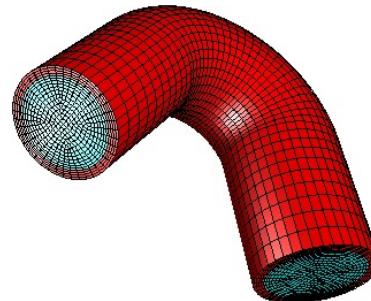
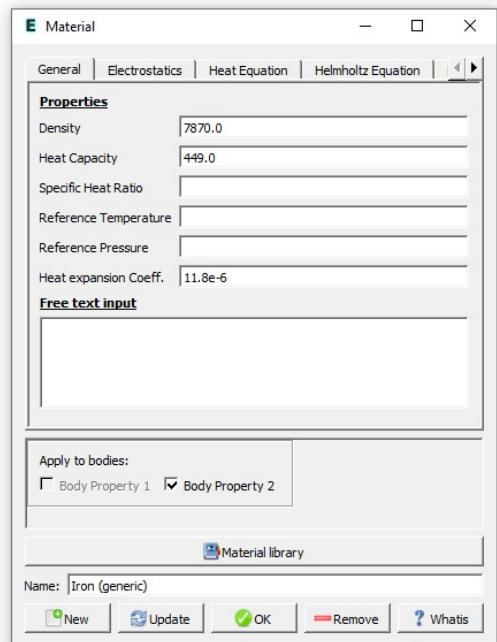


Diferente dos outros exemplos, usaremos da biblioteca disponível no Elmer para definir os materiais que compõem nossa peça. Primeiro a parte interna, que é água, e depois a externa que é feita de Ferro.

```
Model
  Material
    Add
      Material library
        Water (room temperature)
      Apply to Bodies = 1
OK
```



Add
Material library
Iron (generic)
Apply to Bodies = 2
OK



Neste exemplo não usaremos nenhuma das Initial Conditions ou Body Force, uma vez que o movimento do fluido já pode ser considerado nossa condição inicial, e não há nenhum Body Force a ser aplicada.

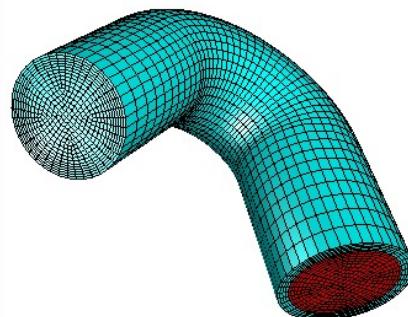
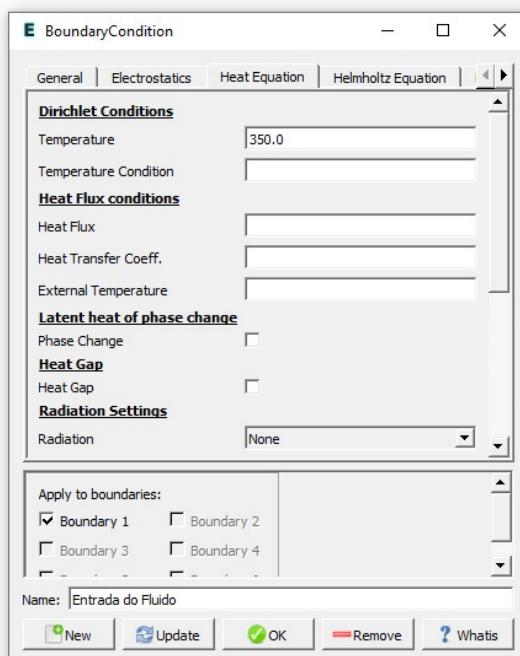
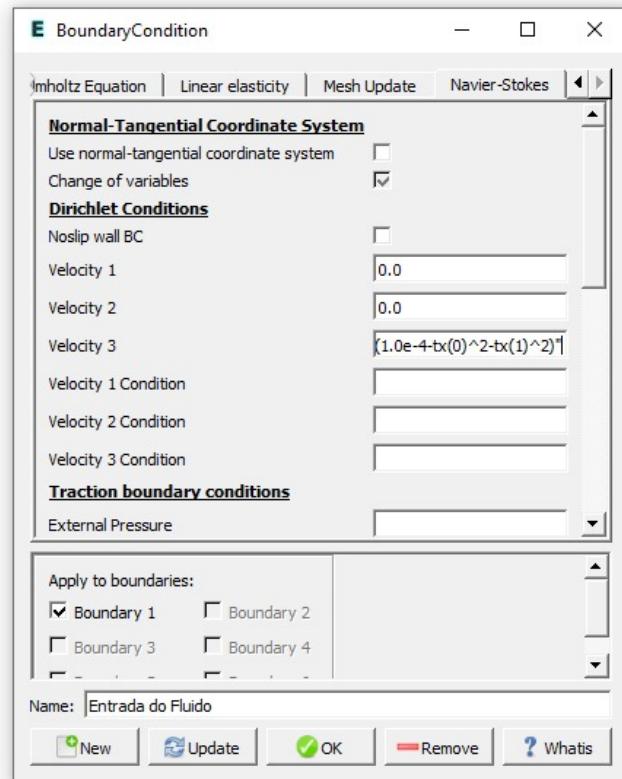
Laboratório Interdisciplinar Virtual de Inteligência Artificial

Já no quesito de condição inicial, temos quatro diferentes a serem aplicadas. Uma para a parte externa, feita de ferro, onde é aplicado apenas calor; a entrada do fluido, a saída dele e o que percorre o interior do tubo. A definição da “Velocity 3” pode ser melhor definida com o atalho ctrl+ENTER.

```

Model
BoundaryCondition
    Name = Entrada do Fluido
    Heat Equation
        Temperature = 350.0
    Navier-Stokes
        Velocity 1 = 0.0
        Velocity 2 = 0.0
        Velocity 3 = " Variable
                      Coordinate
Real MATC
"100.0*(1.0e-4-tx(0)^2-tx(1)^2) "
Add
New

```



Name = Saída do Fluido

Navier-Stokes

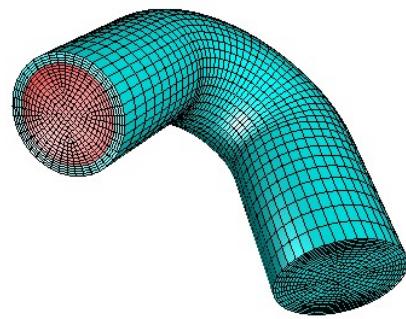
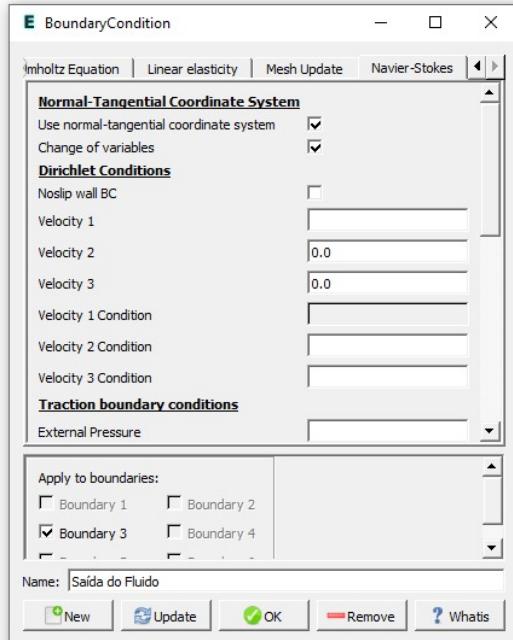
Use normal-tangential coordinate system = on

Velocity 2 = 0.0

Velocity 3 = 0.0

Add

New



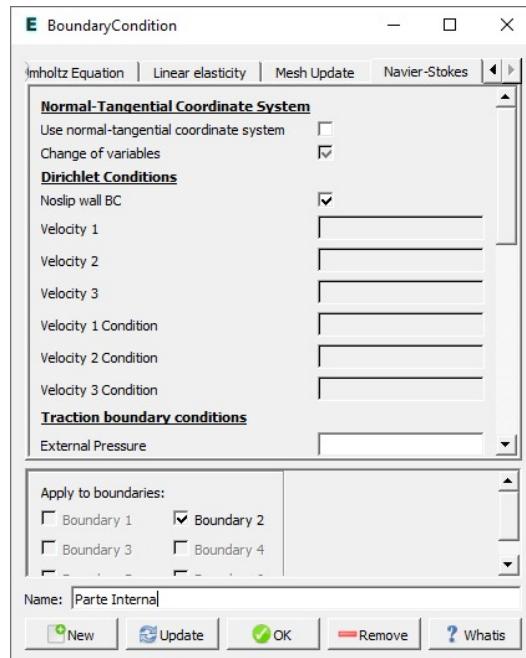
Name = Parte Interna

Navier-Stokes

NoSlip Wall BC = on

Add

New

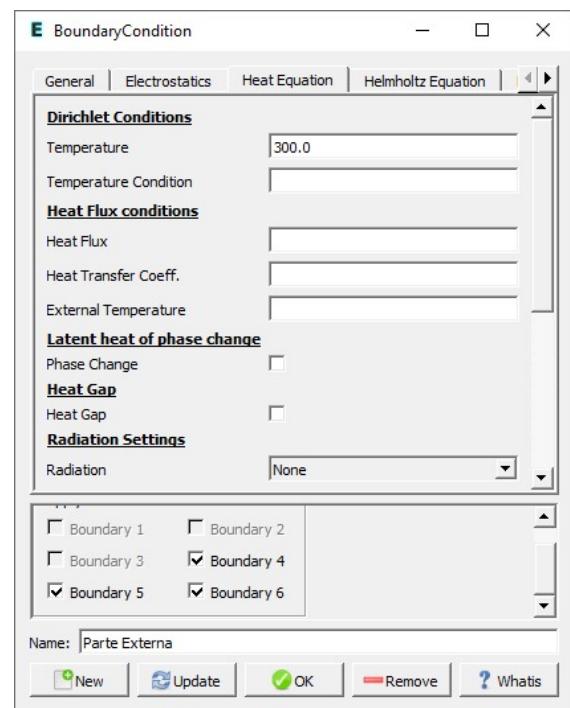


Name = Parte Externa

Heat Equation

Temperature = 300.0

Add



Agora vamos adicionar as Condições de Contorno a seus devidos lugares. Para isso, usa-se a ferramenta que já conhecemos.

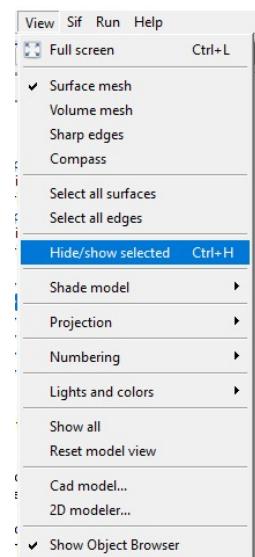
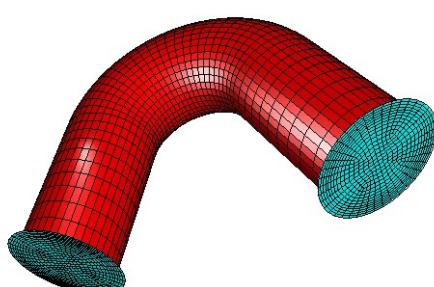
Model

Set boundary properties

Selecione a parte de entrada -> set boundary condition Entrada do Fluido

Selecione a parte de saída -> set boundary condition Saída do Fluido
Choose área externa -> set boundary condition Parte Externa

Para configurarmos a área interna se torna mais complicado, uma vez que precisamos visualizar esta área. Para isso, usamos a ferramenta “Hide/show selected” com a área externa selecionada. Uma vez que a área interna esteja à vista, basta selecioná-la e aplicar sua respectiva condição de contorno.



Com isso, basta gerar o arquivo no menu “**Sif**”, salvar o projeto em “**Files**” e rodar o solver em “**Run**”. Abrindo o arquivo no Paraview, temos os seguintes resultados para velocidade e temperatura respectivamente:

