VIRTUAL | LAB

A fully automated, open-source platform for virtual experiments.

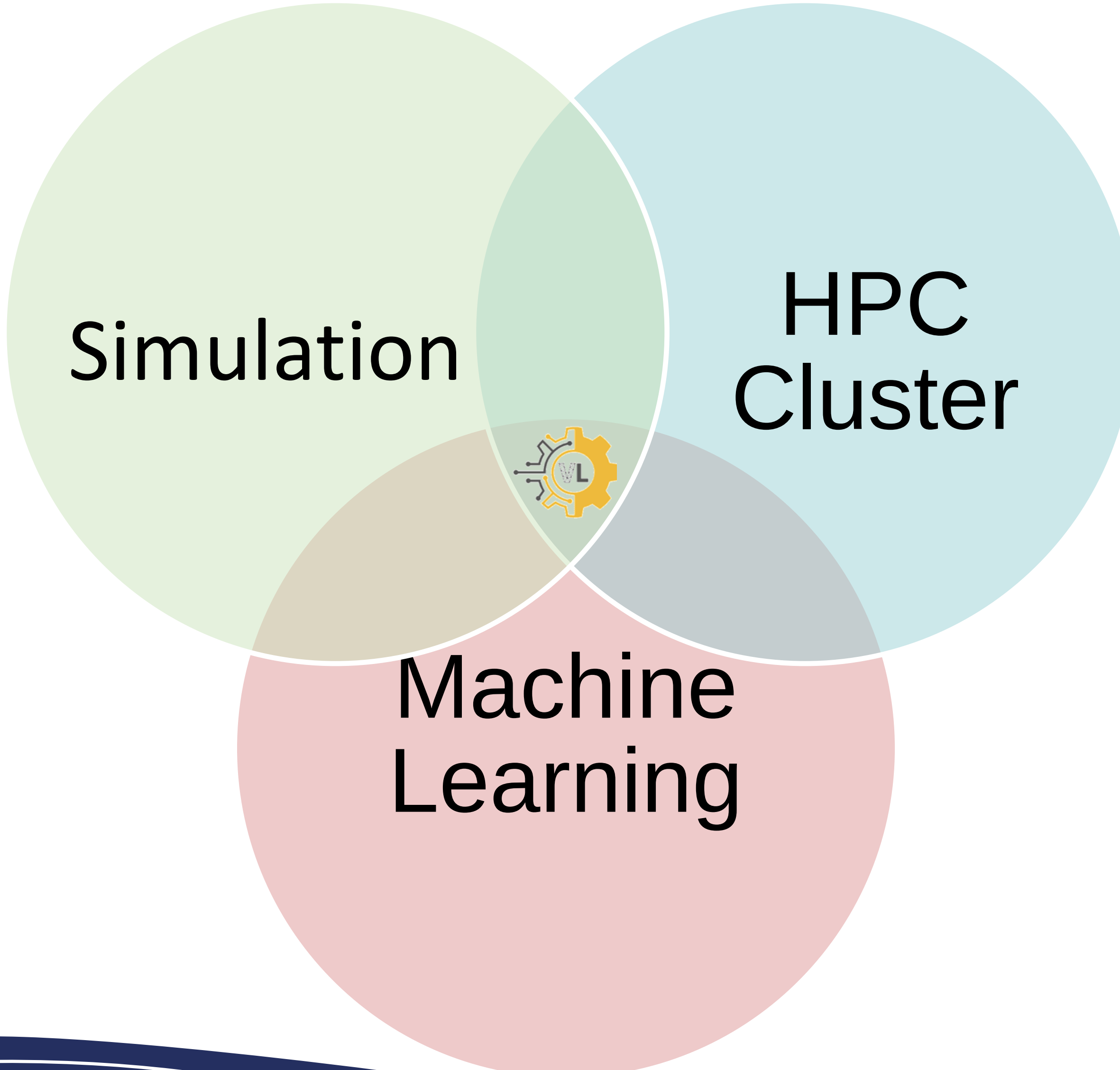
IBSim-4i Training, Institute of Physics, 19th October 2021

Training Format

1. Introduction to VirtualLab.
2. File layout & structure.
3. Tutorial #1: Tensile Test (Mechanical).
4. Tutorial #2: Laser Flash Analysis (Thermal).
5. Tutorial #3: HIVE Experiment (Coupled thermal-electromagnetic).
6. Tutorial #4: Image-based simulation.
7. Tutorial #5: Machine Learning.

Training Format

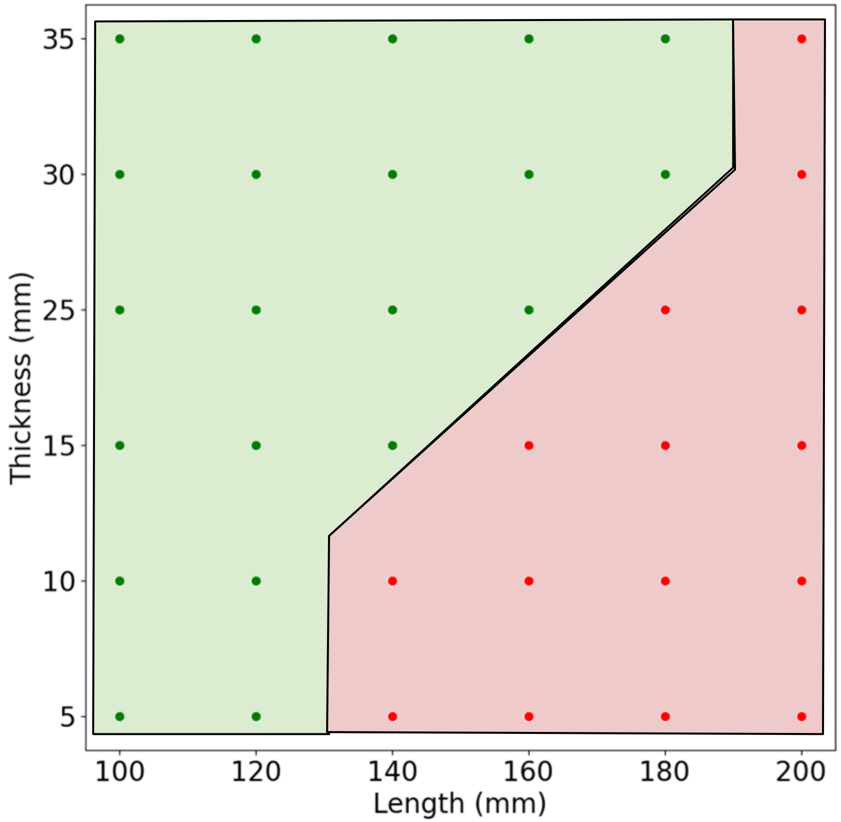
1. Introduction to VirtualLab.
2. File layout & structure.
3. Tutorial #1: Tensile Test (Mechanical).
4. Tutorial #2: Laser Flash Analysis (Thermal).
5. Tutorial #3: HIVE Experiment (Coupled thermal-electromagnetic).
6. Tutorial #4: Image-based simulation.
7. Tutorial #5: Machine Learning.


Simulation
HPC Cluster
Machine Learning

What is VirtualLab?

- A wrapper for software packages.
- Fully automated workflow.
- High-throughput computations.
- Multi-node cluster compatible.
- Open-source & actively developed.
- Written in Python.
- Multiple install options available.

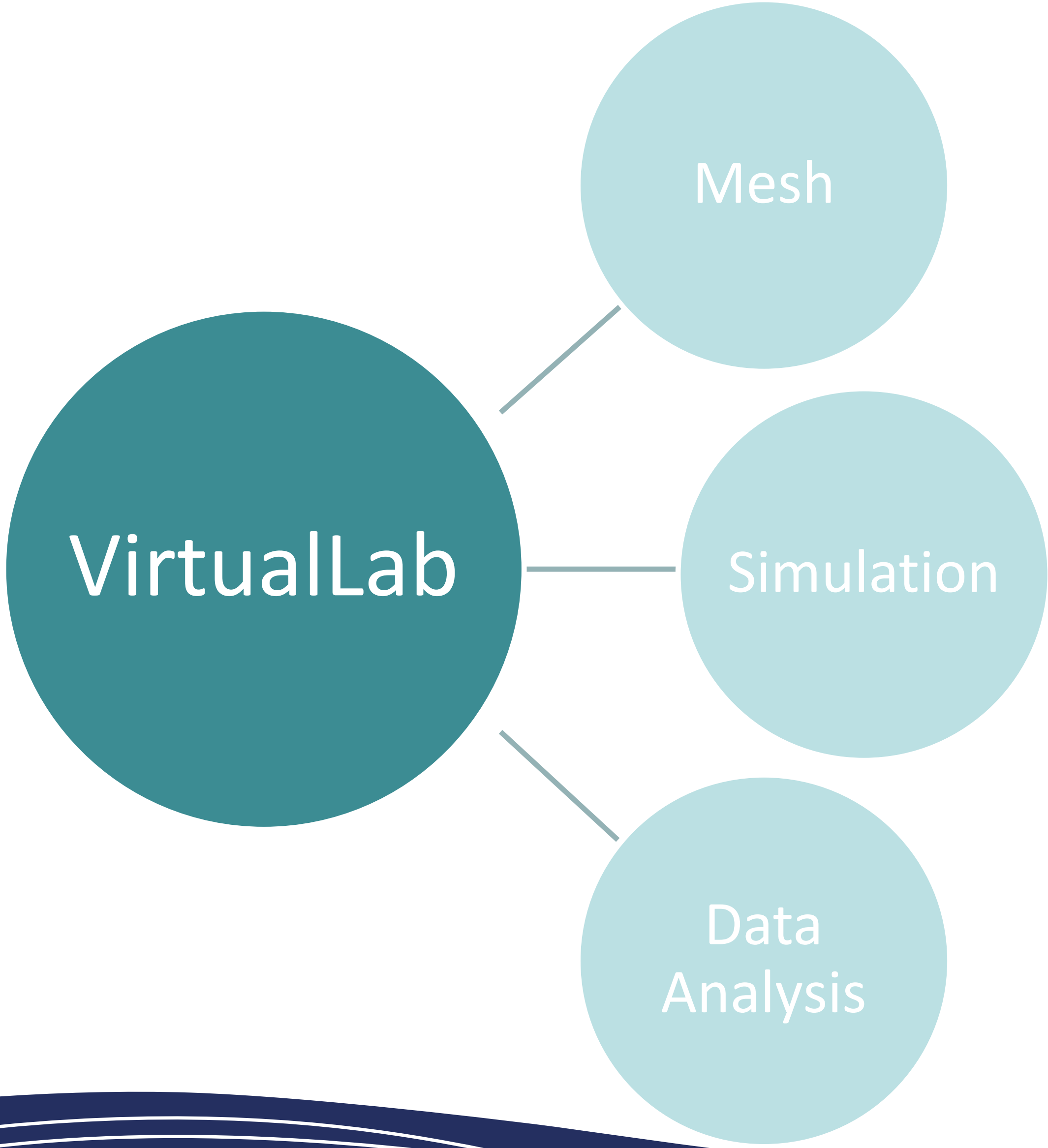
"A parametric analysis is the study of the influence of different geometric and physical parameters on the solution of the problem"



Advantages

- Increased understanding.
- Uncertainty reduction.
- Model simplification.
- Identifying interesting regions.
- Testing robustness.

VirtualLab Components



VirtualLab Packages

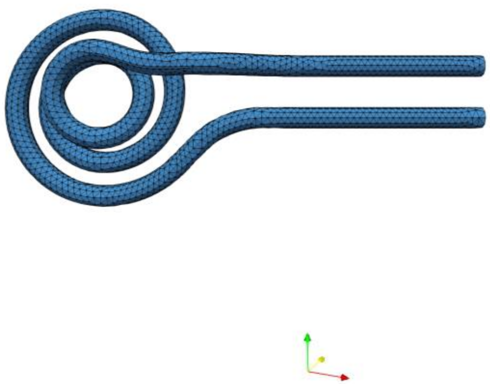
SALOME

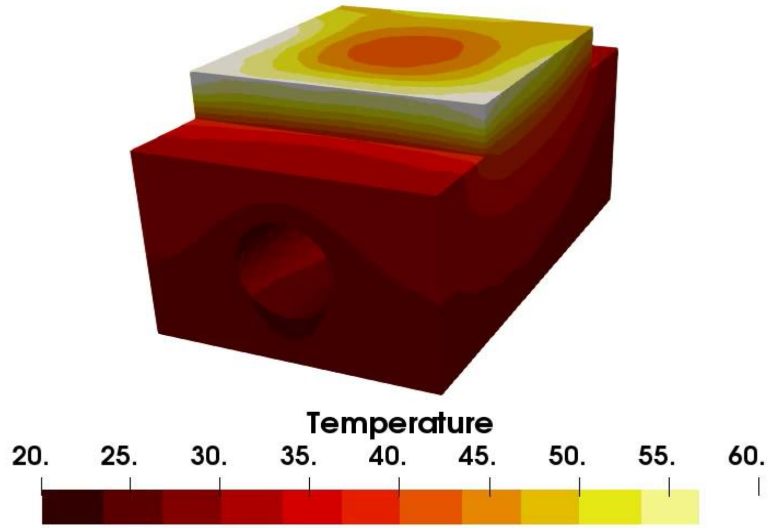
code_aster

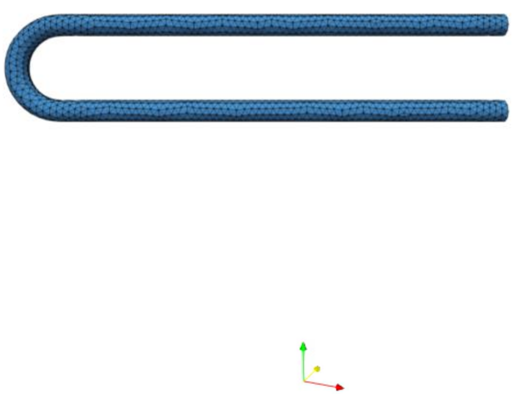
ermes

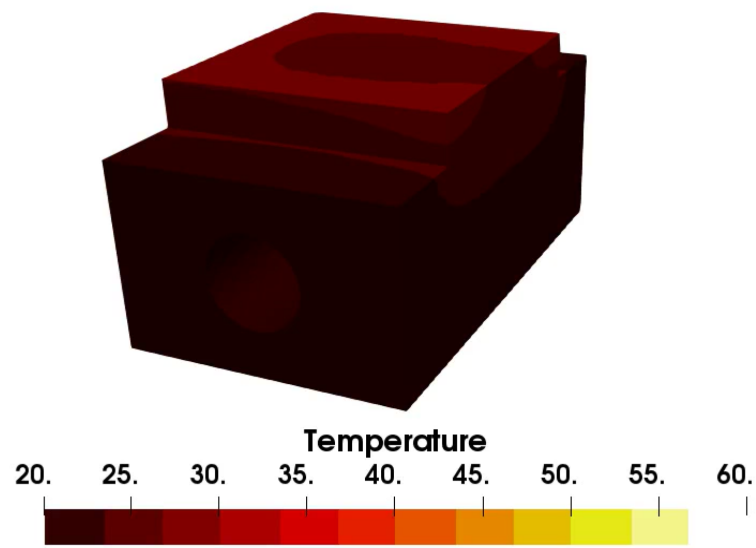
ParaView

VirtualLab Use: Modelling HIVE

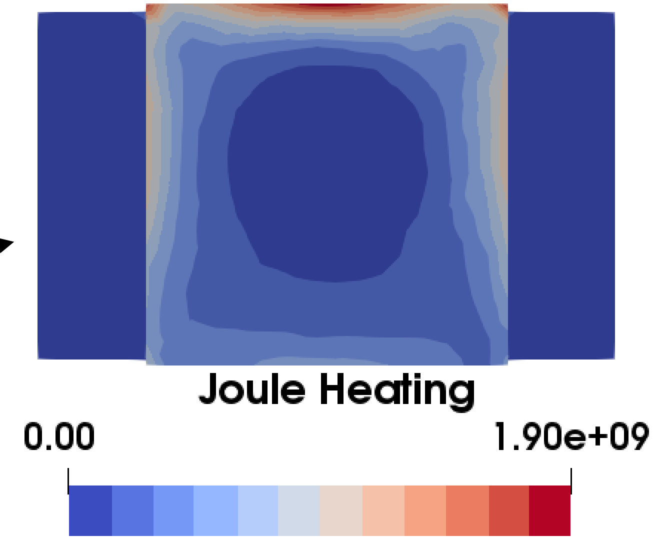


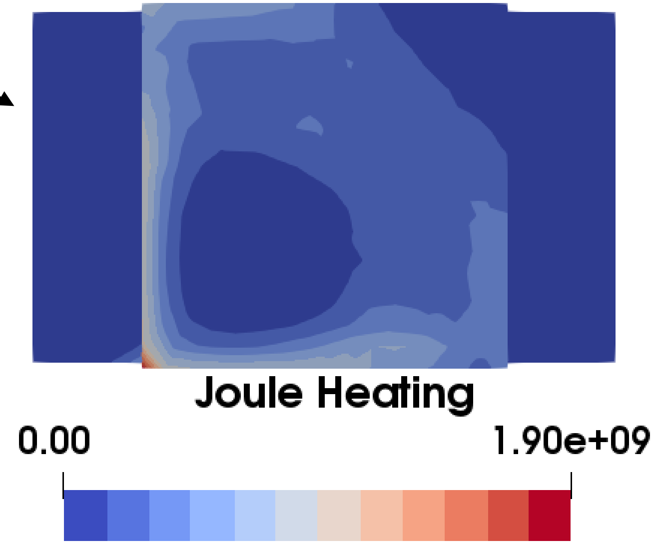






- Identify experimental parameters which produce a desired result.
- Maximum power.
- Minimum variation with power above certain value.

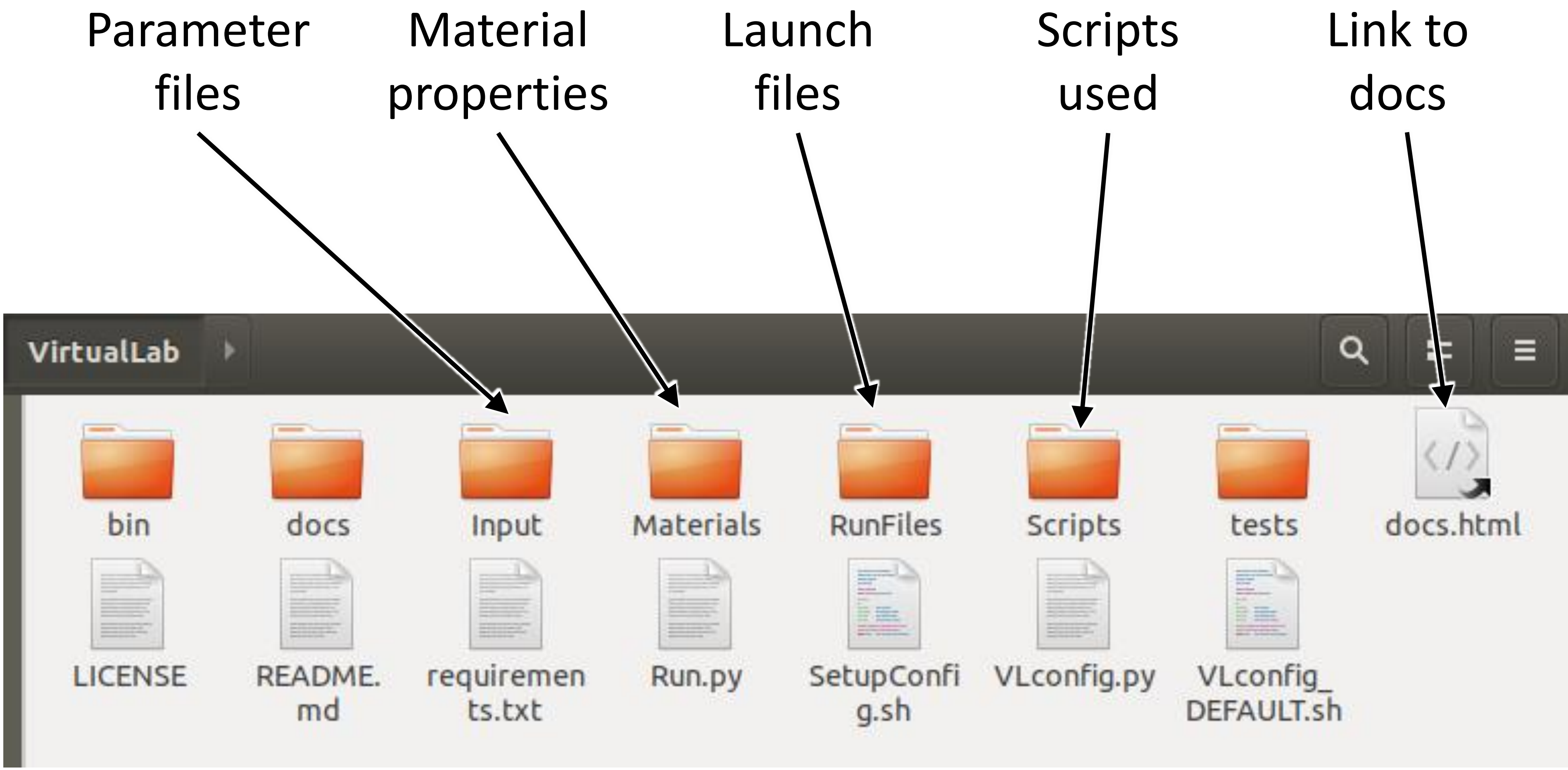




Training Format

1. Introduction to VirtualLab.
2. File layout & structure.
3. Tutorial #1: Tensile Test (Mechanical).
4. Tutorial #2: Laser Flash Analysis (Thermal).
5. Tutorial #3: HIVE Experiment (Coupled thermal-electromagnetic).
6. Tutorial #4: Image-based simulation.
7. Tutorial #5: Machine Learning.

Top Level Directory Structure



Running VirtualLab: The RunFile

```
#=============================================
# Header
import sys
sys.dont_write_bytecode=True
from Scripts.Common.VirtualLab import VLSetup

#=============================================
# Setup
Simulation='Tensile'
Project='Tutorials'
Parameters_Master='TrainingParameters'
Parameters_Var=None

#=============================================
# Environment
VirtualLab=VLSetup(
           Simulation,
           Project)

VirtualLab.Settings(
           Mode='Headless',
           Launcher='Process',
           NbThreads=1)

VirtualLab.Parameters(
           Parameters_Master,
           Parameters_Var,
           RunMesh=True,
           RunSim=True,
           RunDA=True)

VirtualLab.Mesh(
           ShowMesh=False,
           MeshCheck=None)

VirtualLab.Sim(
           RunPreAster=True,
           RunAster=True,
           RunPostAster=True,
           ShowRes=False)

VirtualLab.DA()

VirtualLab.Cleanup()
```

- Python file to launch VirtualLab.
- Header: Import modules required.
- Setup: Variables required by VirtualLab.
- Environment: How VirtualLab is run.

Running VirtualLab: The RunFile - Setup

```
#=======================================
# Setup
Simulation='Tensile'
Project='Tutorials'
Parameters_Master='TrainingParameters'
Parameters_Var=None
```

- Type of 'virtual' experiment.
- Name of project being worked on.
- Master parameter files used for analysis.
- Parametric/Var parameter file used for analysis.

Running VirtualLab: The RunFile - Environment

```
#================================
# Environment
VirtualLab=VLSetup(
           Simulation,
           Project)

VirtualLab.Settings(
           Mode='Headless',
           Launcher='Process',
           NbThreads=1)

VirtualLab.Parameters(
           Parameters_Master,
           Parameters_Var,
           RunMesh=True,
           RunSim=True,
           RunDA=True)

VirtualLab.Mesh(
           ShowMesh=False,
           MeshCheck=None)

VirtualLab.Sim(
           RunPreAster=True,
           RunAster=True,
           RunPostAster=True,
           ShowRes=False)

VirtualLab.DA()

VirtualLab.Cleanup()
```

- Initiate VirtualLab.
- Update VirtualLab settings (optional).
- Setup analysis using parameters & create directories.
- Run ‘Mesh’ routine.
- Run ‘Simulation’ routine
- Run ‘Data Analysis’ routine.
- Delete temporary directories created.

Timetable

Tue 19 Oct 2021

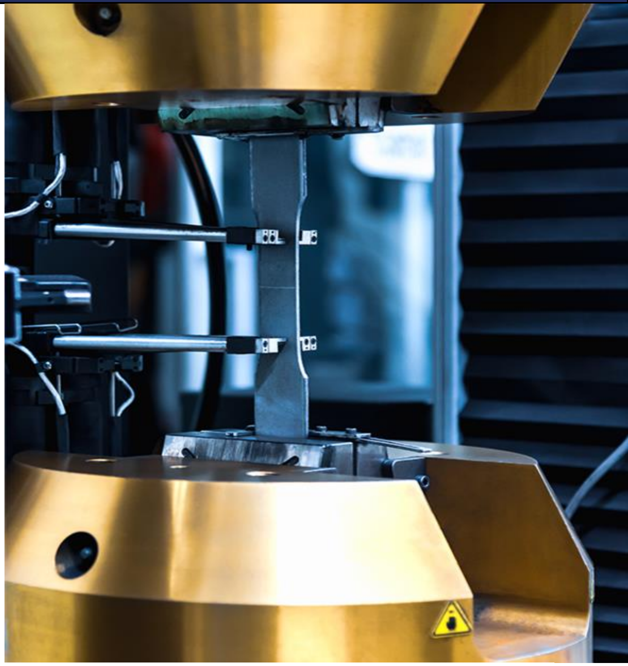
13:30	Introduction
13:45	Tutorials 1-2
14:45	Coffee
15:15	Tutorials 3-5
17:00	End

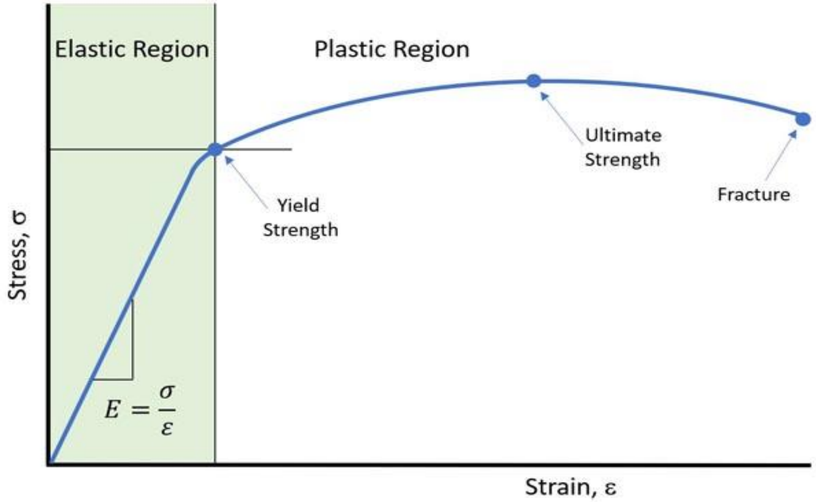
Training Format

1. Introduction to VirtualLab.
2. File layout & structure.
3. Tutorial #1: Tensile Test (Mechanical).
4. Tutorial #2: Laser Flash Analysis (Thermal).
5. Tutorial #3: HIVE Experiment (Coupled thermal-electromagnetic).
6. Tutorial #4: Image-based simulation.
7. Tutorial #5: Machine Learning.

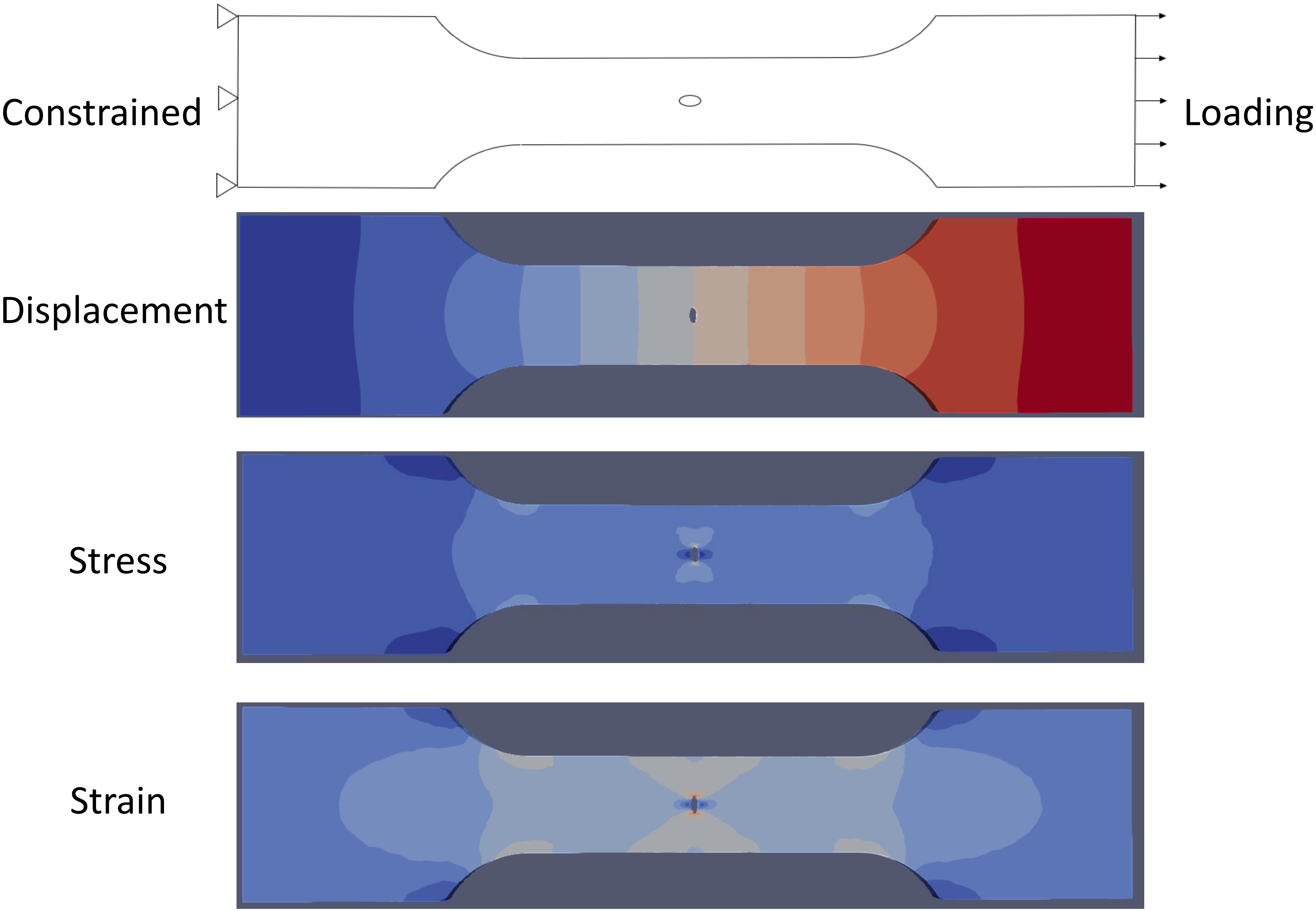
Tutorial #1: Tensile Test

- A 'dog-bone' shaped sample is loaded either through;
  - Force controlled.
  - Displacement controlled.
- Provides the Young's elastic modulus (E) of a material.





Tutorial #1: Virtual Tensile Test



Tutorial #1: Documentation

- Open the documentation in the VirtualLab directory.
- Navigate to Tutorials -> Mechanical -> Tensile Testing.
- Read through the ‘Sample’ & ‘Simulation’ sections

Tutorial #1: RunFile

```
#=======================================
# Setup

Simulation='Tensile'
Project='Tutorials'
Parameters_Master='TrainingParameters'
Parameters_Var=None

#=======================================
# Environment

VirtualLab=VLSetup(
           Simulation,
           Project)

VirtualLab.Settings(
           Mode='Interactive',
           Launcher='Process',
           NbThreads=1)

VirtualLab.Parameters(
           Parameters_Master,
           Parameters_Var,
           RunMesh=True,
           RunSim=True,
           RunDA=True)

VirtualLab.Mesh(
           ShowMesh=False,
           MeshCheck=None)

VirtualLab.Sim(
           RunPreAster=True,
           RunAster=True,
           RunPostAster=True,
           ShowRes=True)

VirtualLab.DA()

VirtualLab.Cleanup()
```

- Open ‘*RunFiles/RunTutorials.py*’.
- Results saved to ‘*Output/Tensile/Tutorials*’.
- Parameter files found in ‘*Input/Tensile/Tutorials*’.
- Mode is set to ‘Interactive’.
- ShowRes is set to True.

Please see screen
for instructions

Tutorial #1: Task 1 – Launching VirtualLab

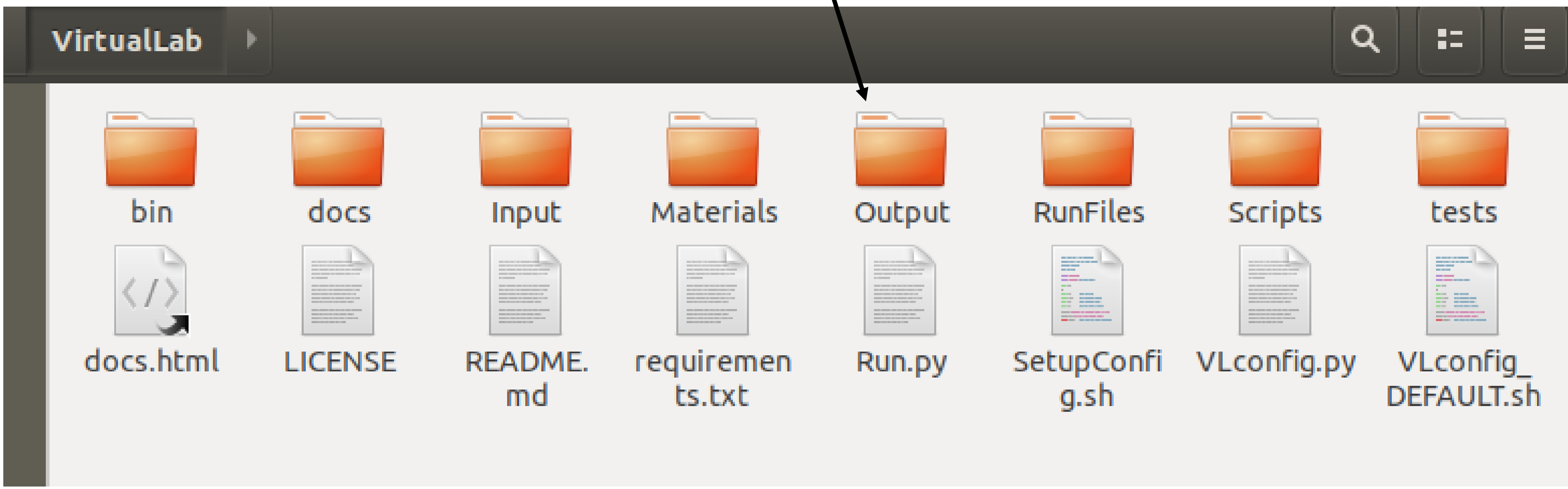
- Everything is already set up for the first task.
- As 'Parameters_Var' is None only a single mesh and simulation will be run.
- Launching VirtualLab:
  - Open a terminal (Ctrl+Alt+t or RightClick>Terminal).
  - Type 'cd VirtualLab' and hit enter (no need for '').
  - Type 'VirtualLab –f RunFiles/RunTutorials.py' and hit enter.

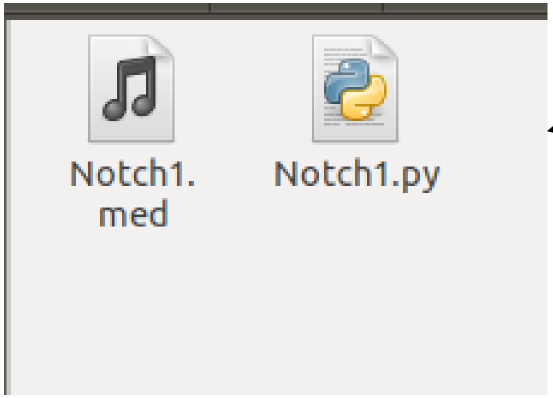
```
(base) ubuntu@ubuntu-VirtualBox:~$ cd VirtualLab/
(base) ubuntu@ubuntu-VirtualBox:~/VirtualLab$ VirtualLab -f RunFiles/RunTutorials.py
```

- Note: You will need to manually close the pop-up window.

VirtualLab Output

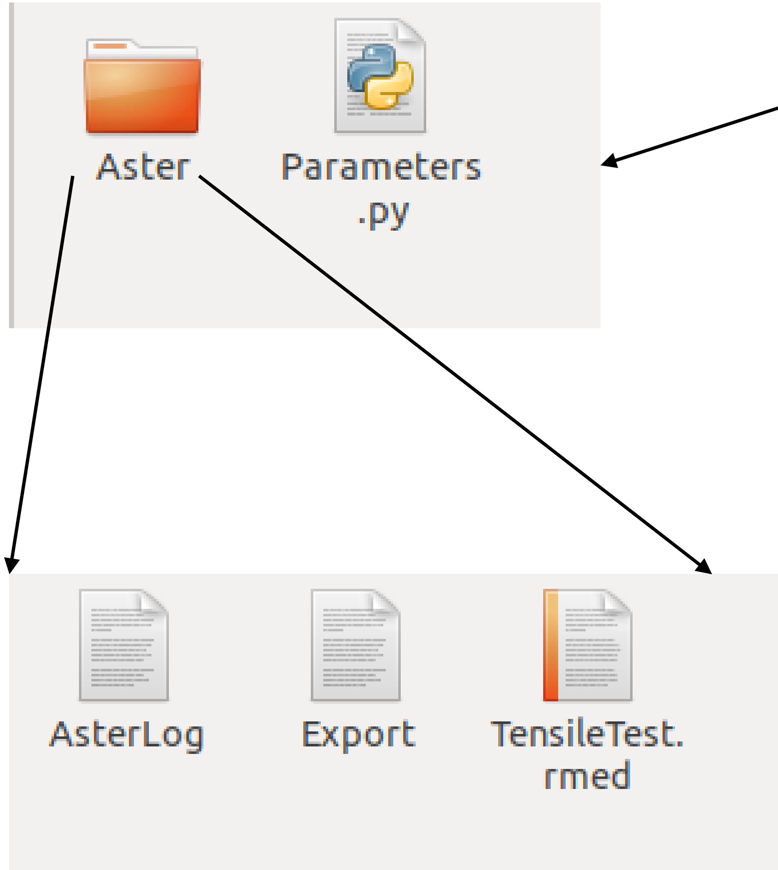
Output
directory





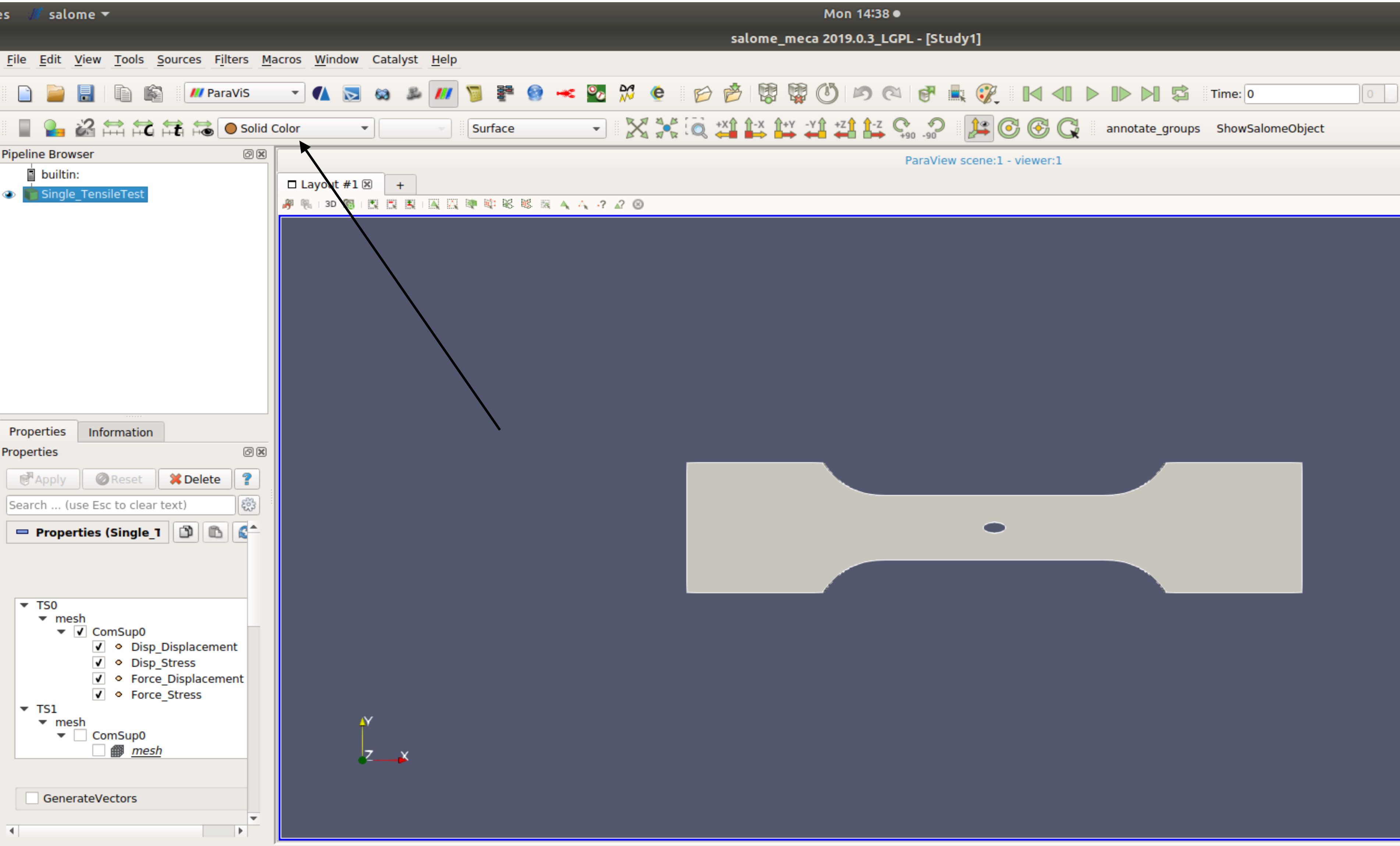
- Open directory *'Output/ Tensile/Tutorials/Meshes'.*
- *'Notch1.med'*: The mesh.
- *'Notch1.py':* Parameter values used to create mesh.

Tutorial #1: Task 1 – Simulation Output

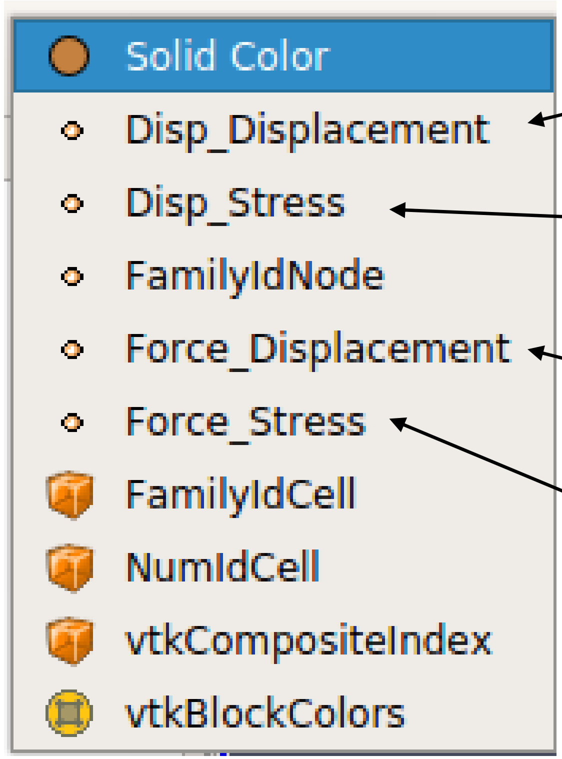


- Open the directory *‘Output/ Tensile/Tutorials/Single’*
- Parameter values used in simulation saved to *‘Parameters.py’*.
- Open *‘Aster’* directory.
- *‘Export’*: information passed to CodeAster.
- *‘AsterLog’*: Log of terminal output from CodeAster.
- *‘TensileTest.rmed’*: CodeAster result file.

Tutorial #1: Task 1 – Visualising Results

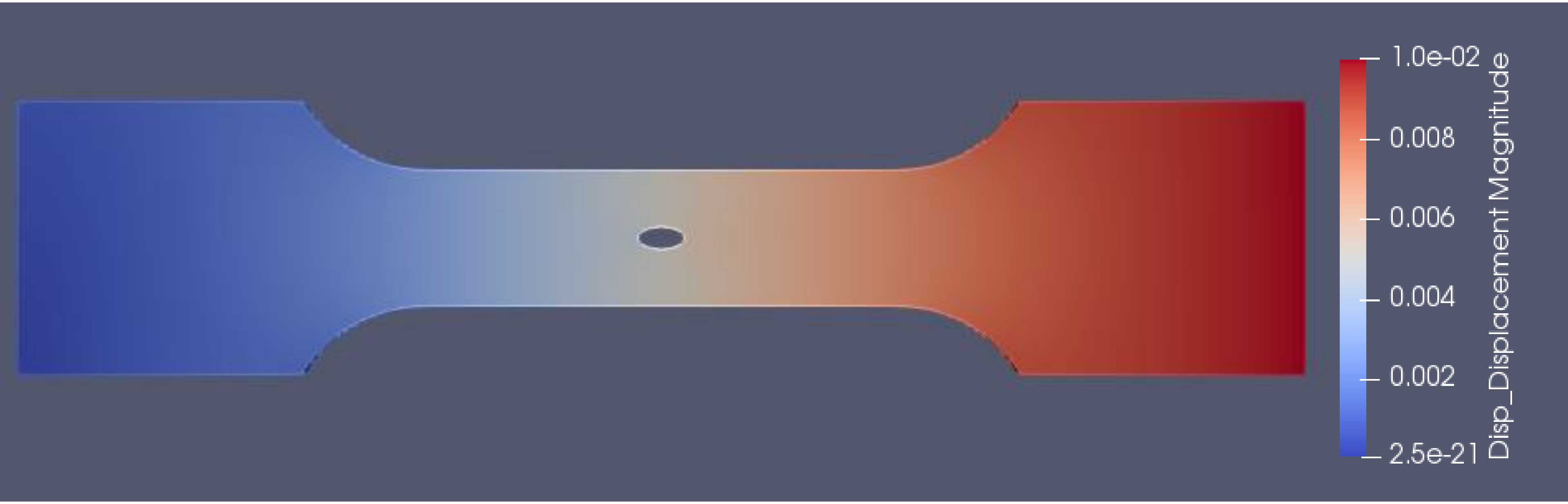


Tutorial #1: Task 1 – Results Available



- Displacement plot for displacement controlled analysis.
- Stress plot for displacement controlled analysis.
- Displacement plot for force controlled analysis.
- Stress plot for force controlled analysis.

Tutorial #1: Task 1 – Displacement Visualisation



Please see screen
for instructions

Tutorial #1: Task 1 – Change Parameters

- Open *'Input/Tensile/Tutorials/TrainingParameters_Change.py'*.
- Change the value for one of the 'Mesh' parameters.
- Launch VirtualLab using the RunFile *'RunFiles/RunTutorials_Change.py'*.
- **Note**: You may receive an error message if the geometry is infeasible.

Summary

- VirtualLab motivation & use.
- Directory structure
- RunFile layout & inputs required.
- Parameter files layout & use.
- Launching VirtualLab.
- Output files explained.
- Visualising results using ParaViS.
- Making changes to parameter files.

Remaining Tasks

- Complete the 3 remaining tasks:
  - Task 2: Running multiple simulations.
  - Task 3: Running in parallel.
  - Task 4: Only run simulations.
- Navigate to Tutorials -> Mechanical -> Tensile Testing -> Task 2.

Summary: Task 2 - 4

- Running multiple studies using Parameters_Var file.
- Running in parallel.
- Running simulations only.

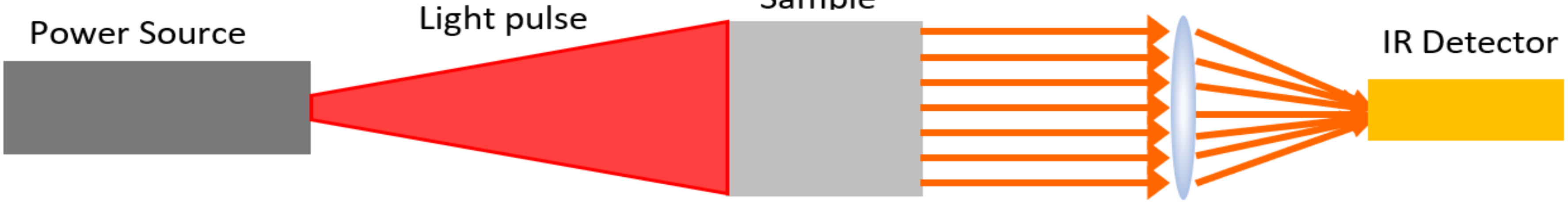
Please see screen
for instructions

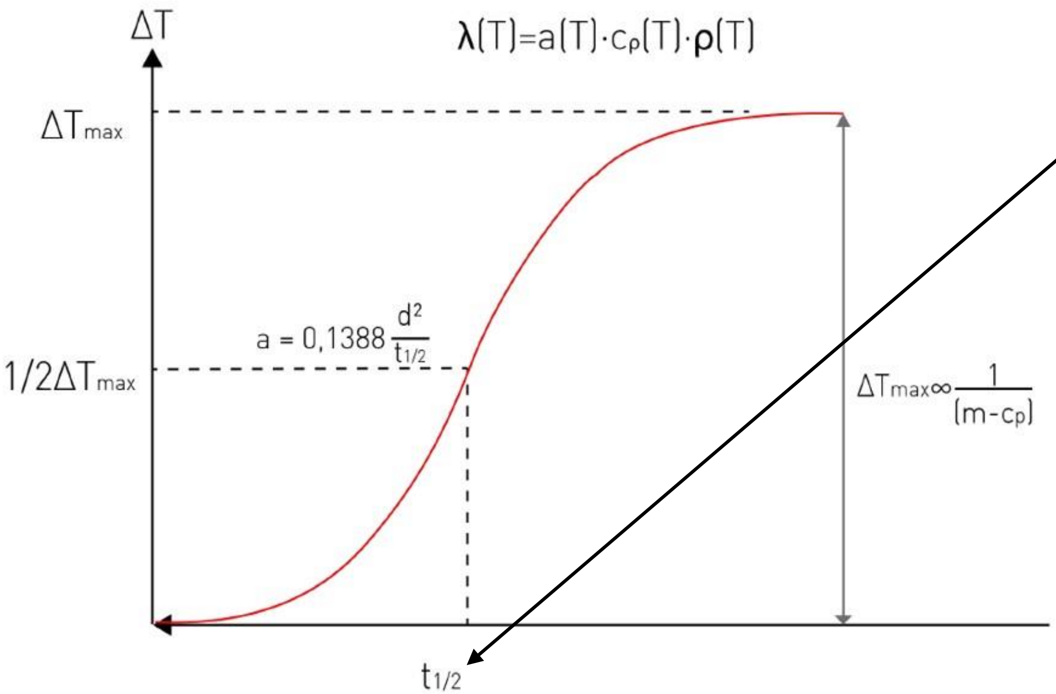
Training Format

1. Introduction to VirtualLab.
2. File layout & structure.
3. Tutorial #1: Tensile Test (Mechanical).
4. Tutorial #2: Laser Flash Analysis (Thermal).
5. Tutorial #3: HIVE Experiment (Coupled thermal-electromagnetic).
6. Tutorial #4: Image-based simulation.
7. Tutorial #5: Machine Learning.

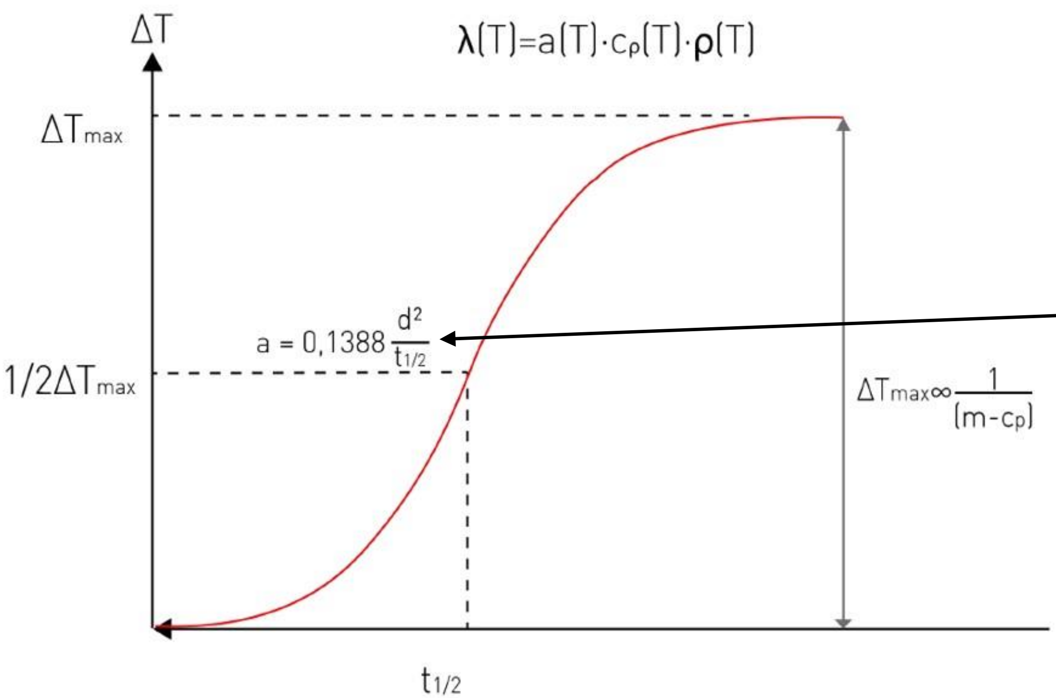
Tutorial #2: Laser Flash Analysis

- Front face heated with a light pulse.
- Temperature rise on rear is measured as a function of time.
- Used to measure the thermal diffusivity of a material.

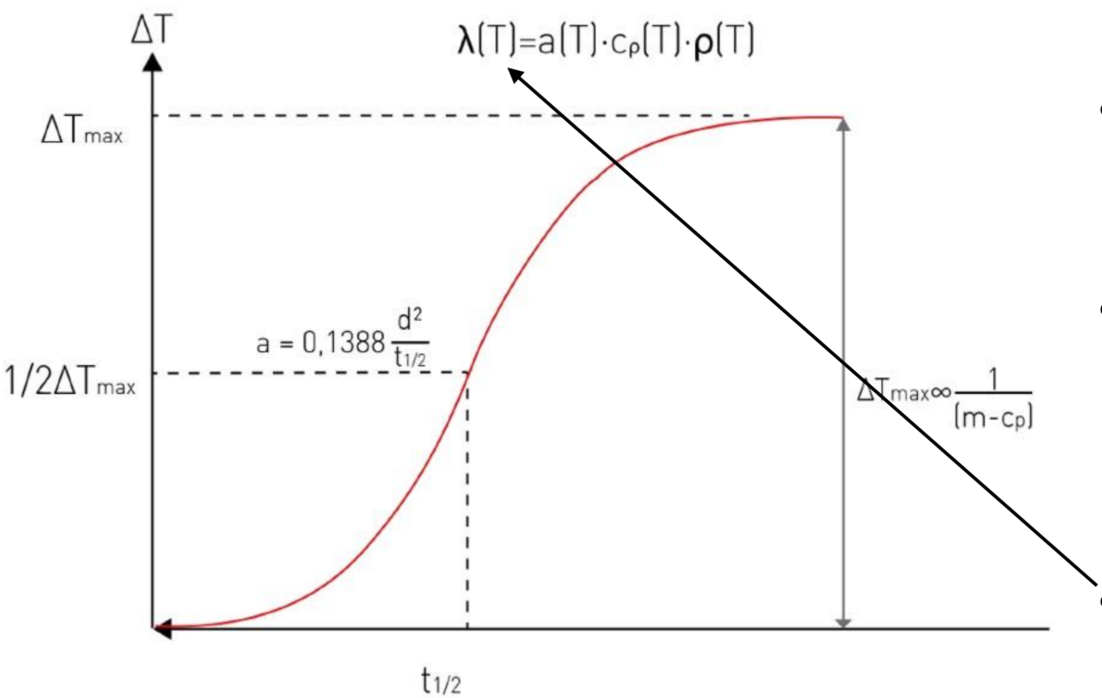




- Temperature increase on sample rear.
- Measure the 'half-rise' time.



- Temperature increase on sample rear.
- Measure the 'half-rise' time.
- Use to calculate thermal diffusivity,
  d - thickness.



- Temperature increase on sample rear.
- Measure the 'half-rise' time.
- Use to calculate thermal diffusivity,
  d - thickness.
- Then calculate thermal conductivity,
  C_p - Specific heat capacity,
  ρ - Density.

Tutorial #2: Documentation

- Open the documentation in the VirtualLab directory.
- Navigate to Tutorials -> Thermal -> LFA.
- Read through the ‘Sample’ & ‘Simulation’ sections

Please see screen
for instructions

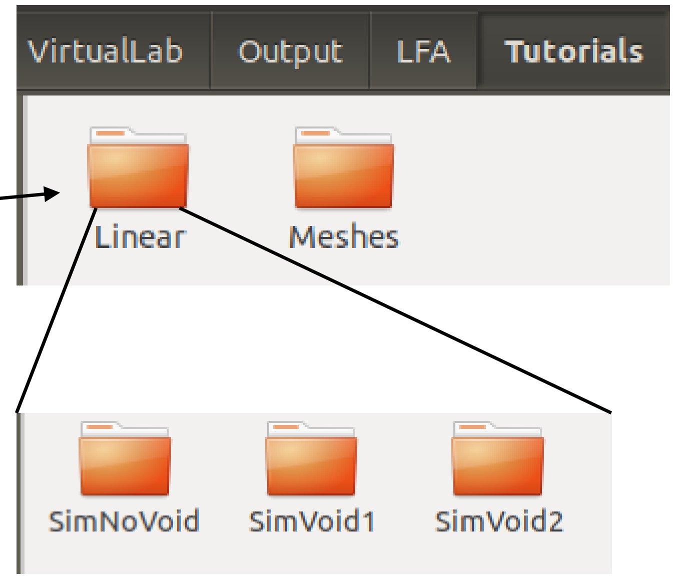
Tutorial #2: Task 1

- Navigate to Tutorials -> Thermal -> LFA -> Task 1.
- Complete the steps for Task 1 & Task 2.

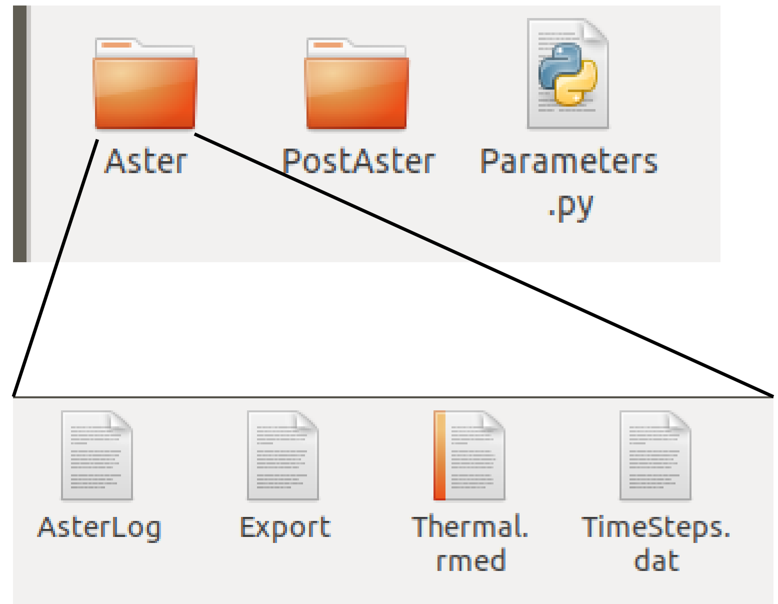
- **Note**: You will need to close the GUI once you have finished viewing the meshes (Task 1).

Tutorial #2: Task 2 – Results

- Go to the project directory (*‘Output/LFA/Tutorials’).*
- Directory names ‘Linear’ created.
- Inside are simulation results.

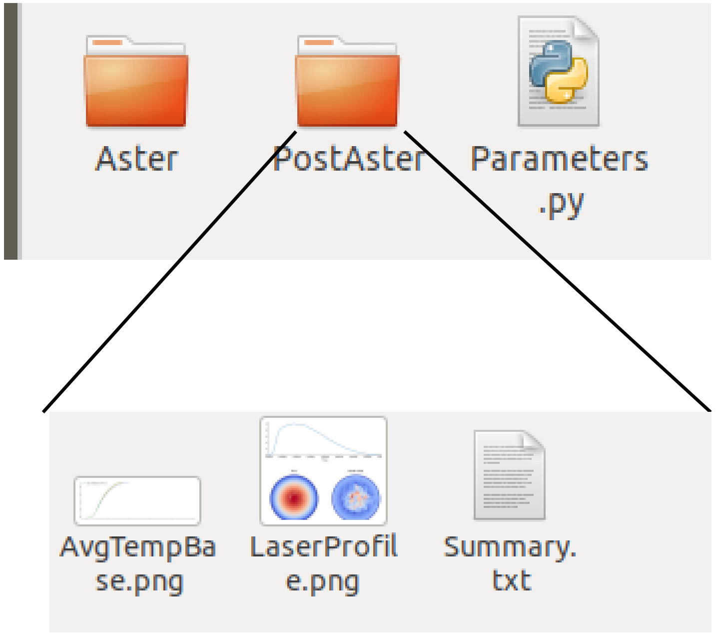


- Open ‘SimNoVoid’.
- ‘Aster’ directory.
- ‘PostAster’ directory.
- Parameters used in simulation.
- ‘TimeSteps.dat’: Time steps used in simulation.

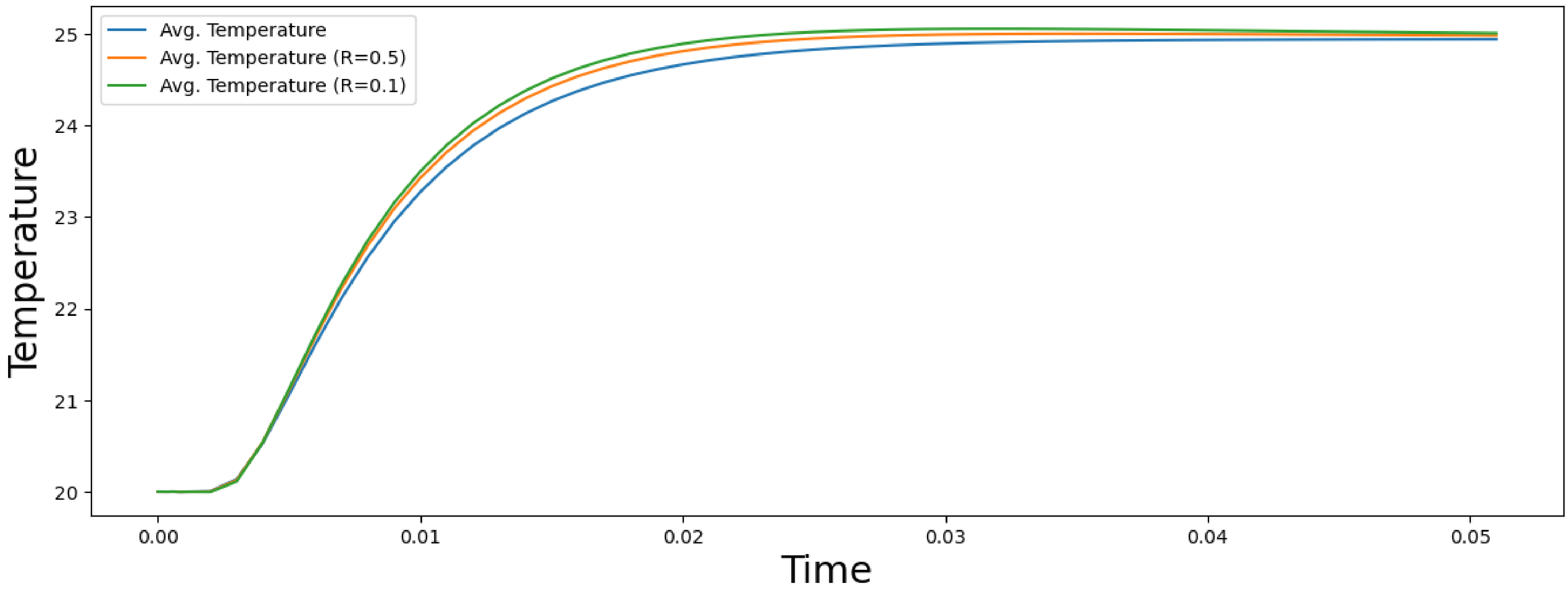



```
0.000000
0.000020
0.000040
0.000060
0.000080
0.000100
0.000120
0.000140
0.000160
0.000180
0.000200
0.000220
0.000240
```

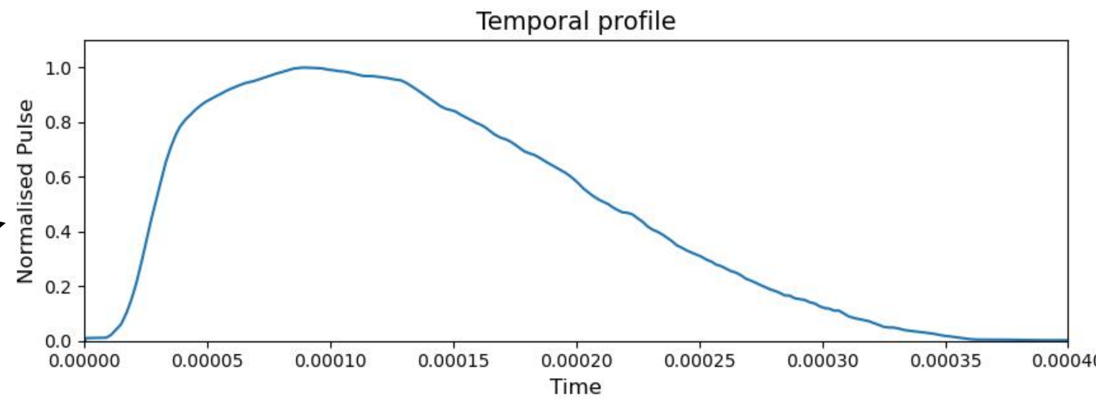
- Open 'SimNoVoid'.
- 'Aster' directory.
- 'PostAster' directory.
- Parameters used in simulation.
- 'TimeSteps.dat': Time steps used in simulation.

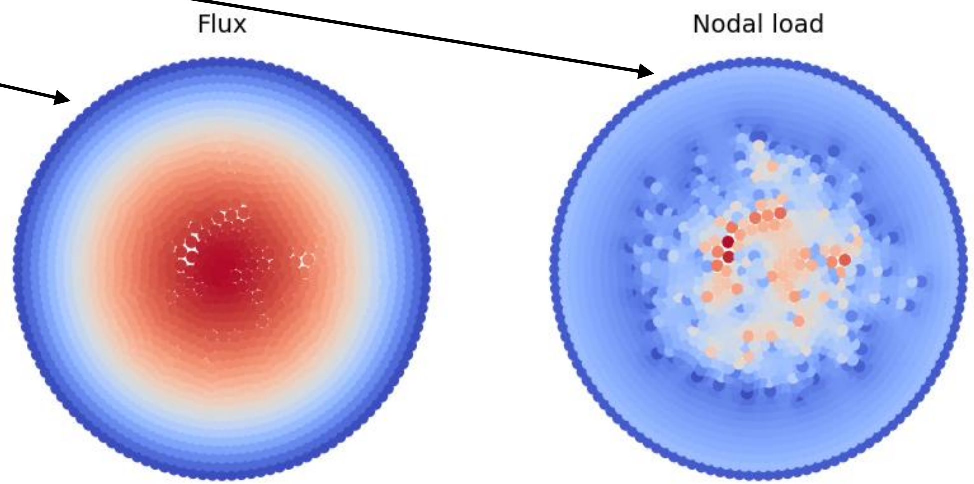


- 'AvgTempBase.png': Temperature profile on rear v time.



- ‘LaserProfile.png’: Laser profile used in analysis.
- Temporal profile.
- Spatial profile (Power).
- Spatial profile (Flux).





Tutorial #2: Task 2 - PostAster

- 'Summary.txt': Information gathered during post-processing.
- Accuracy of time discretisation for laser pulse.
- Accuracy of spatial discretisation for laser pulse.
- Back calculated thermal conductivity.
- Theoretical temperature increase.

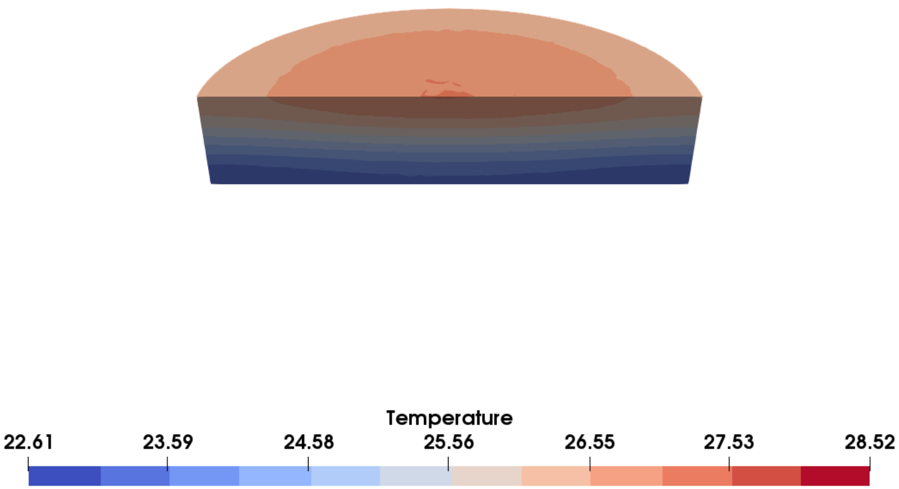
```
### Laser pulse discretisation###

Exact area under laser temporal profile: 0.000186
Area due to temporal discretisation: 0.000186
Error: 0.171708%

Exact volume under laser spatial profile (MGD): 0.154029
Volume due to the spatial discretisation: 0.153964
Error: 0.042595%

### Thermal Conductivity ###

Thermal conductivity of Copper: 394.000W/mK
Calculated thermal conductivity from results: 384.001 W/mK
Error: 2.537704%

### Anticipated temperature ###

Measured temperature increase from simulation: 4.941439°C
Anticipated temperature increase from energy input: 4.951701°C
Error: 0.207235%
```

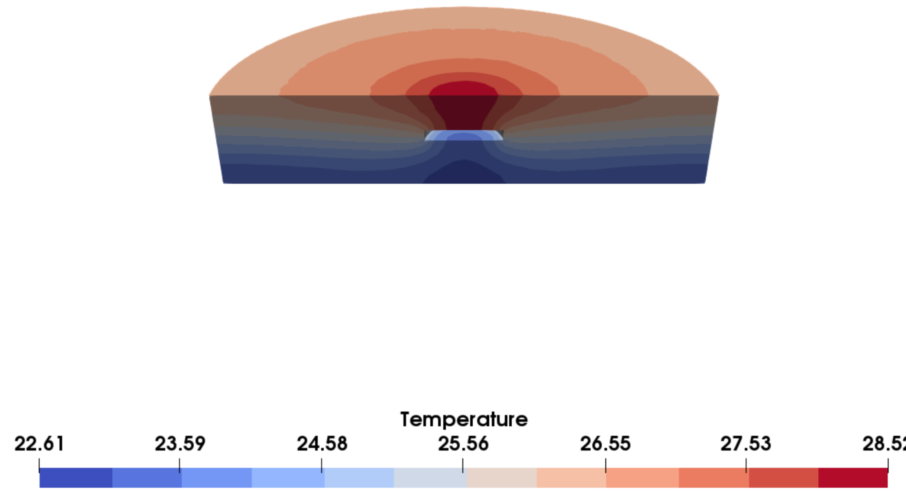
Please see screen
for instructions

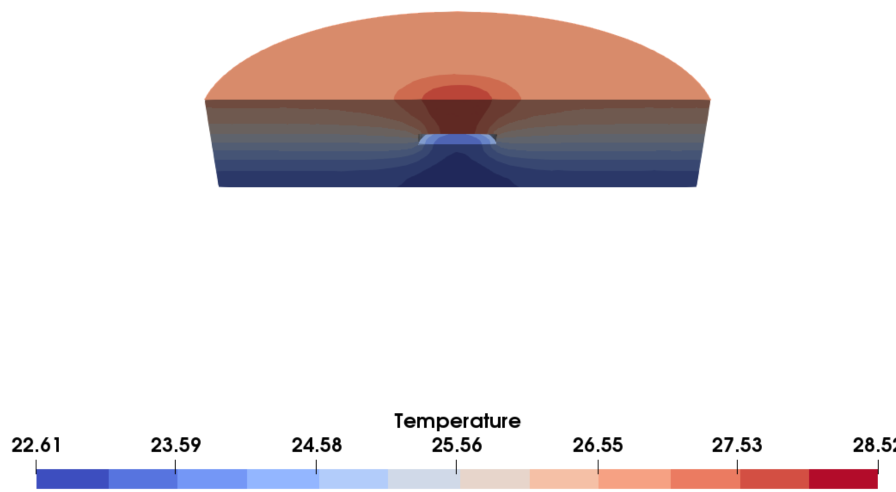
Tutorial #2: Task 3 & 4.

- Open documentation from VirtualLab directory.
- Navigate to Tutorials -> Thermal -> LFA-> Task 3.
- Read through Task 3, but you do not need to run it.
- Complete Task 4.
- **Note:** In *'Input/LFA/Tutorials/TrainingParameters.py'* you will need to uncomment DA.PVGUI=True at the bottom of the file.

Tutorial #2: Task 4 – Collective Post-Processing







Tutorial #2: Task 5

- Navigate to Tutorials -> Thermal -> LFA-> Task 5.
- Follow the steps to complete task 5.
- **Optional Extra**:
  - Change *Mode* to ‘Headless’ in the RunFile (Output written to file instead).
  - Change *ShowRes* to False.
- **Once VirtualLab is running, grab a coffee and let the computer do the work.**

Please see screen
for instructions

Summary: Tutorial #2

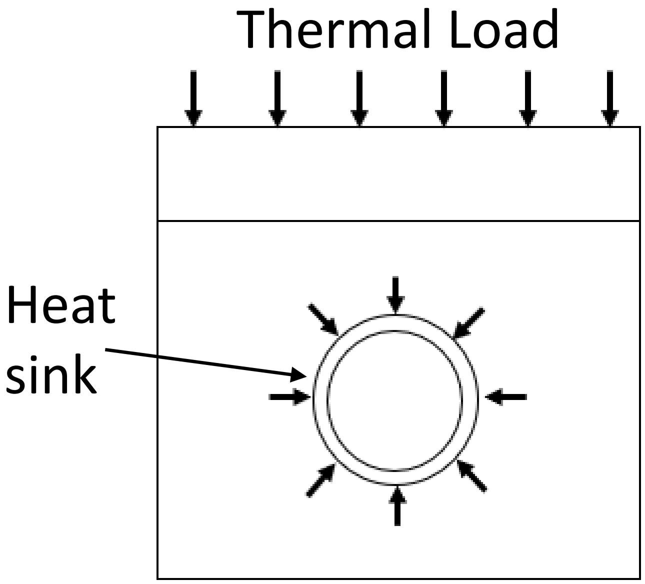
- View meshes created.
- Running transient simulations.
- Organising simulations into sub-directories.
- Running individual post-processing with Sim.PostAsterFile.
- Running collective post-processing with DA.
- Running in Headless mode.

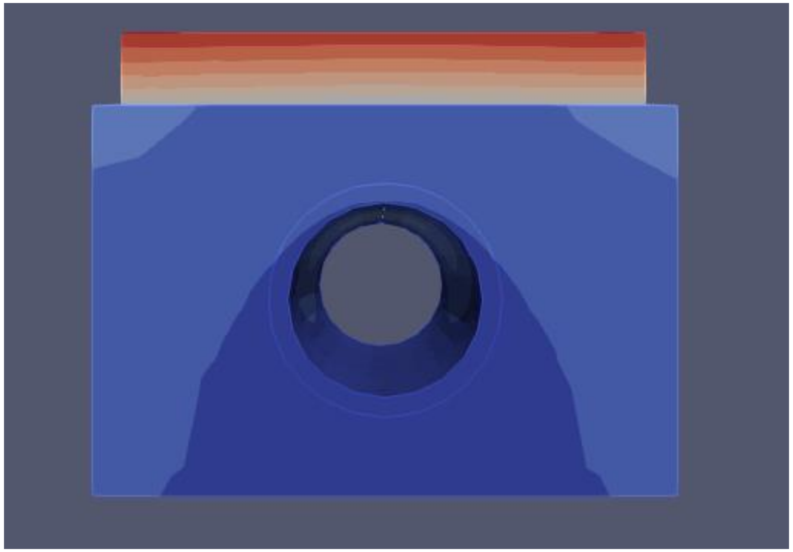
Training Format

1. Introduction to VirtualLab.
2. File layout & structure.
3. Tutorial #1: Tensile Test (Mechanical).
4. Tutorial #2: Laser Flash Analysis (Thermal).
5. Tutorial #3: HIVE Experiment (Coupled thermal-electromagnetic).
6. Tutorial #4: Image-based simulation.
7. Tutorial #5: Machine Learning.

Tutorial #3: HIVE Experiment

- Heat by Induction to Verify Extremes (HIVE) facility at the UK Atomic Energy Authority (UKAEA).
- Designed to research & develop plasma facing components (PFCs).
- Components thermally loaded using induction heating and actively cooled using pressurised coolant.


Thermal Load
Heat sink



Tutorial #3: Documentation

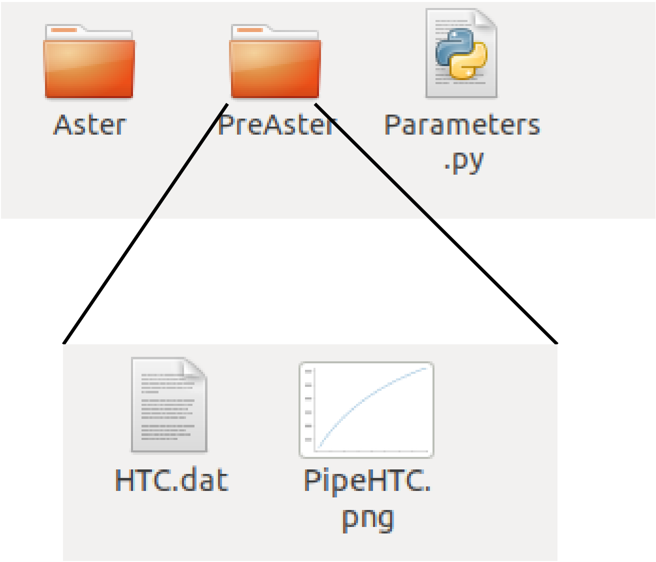
- Open the documentation in the VirtualLab directory.
- Navigate to Tutorials -> Multi-Physics -> HIVE.
- Read through the ‘Sample’ & ‘Simulation’ sections

Please see screen
for instructions

Tutorial #3: Task 1

- Navigate to Tutorials -> Multi-Physics -> HIVE -> Task 1.
- Complete Task 1.

- Go to *'Output/HIVE/Tutorials/Examples/Uniform'*.
- HTC.dat: Heat transfer data between pipe and sample.
- PipeHTC.png: Image of heat transfer data.

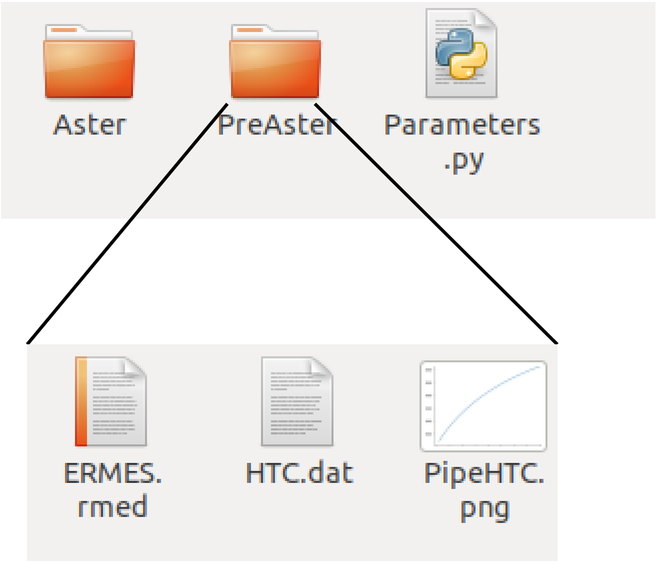


Please see screen
for instructions

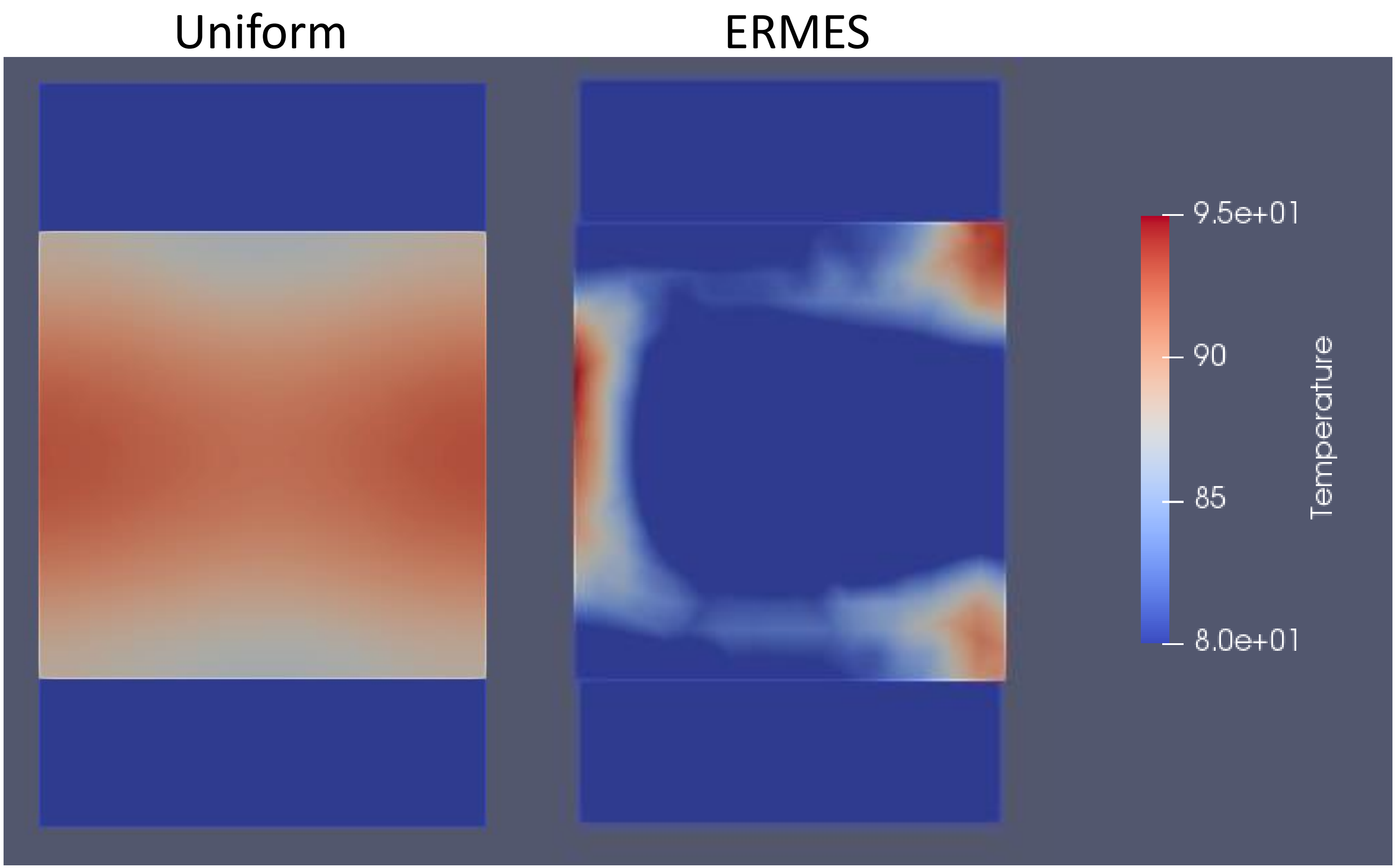
Tutorial #3: Task 2 & 3

- Navigate to Tutorials -> Multi-Physics -> HIVE -> Task 2.
- Complete Task 2 & 3.

- Go to *'Output/HIVE/Tutorials/ Examples/ERMES'*.
- ERMES.rmed: ERMES results file also saved to PreAster directory.



Please see screen for instructions


Uniform
ERMES
9.5e+01
90
85
8.0e+01
Temperature

Summary: Tutorial #3

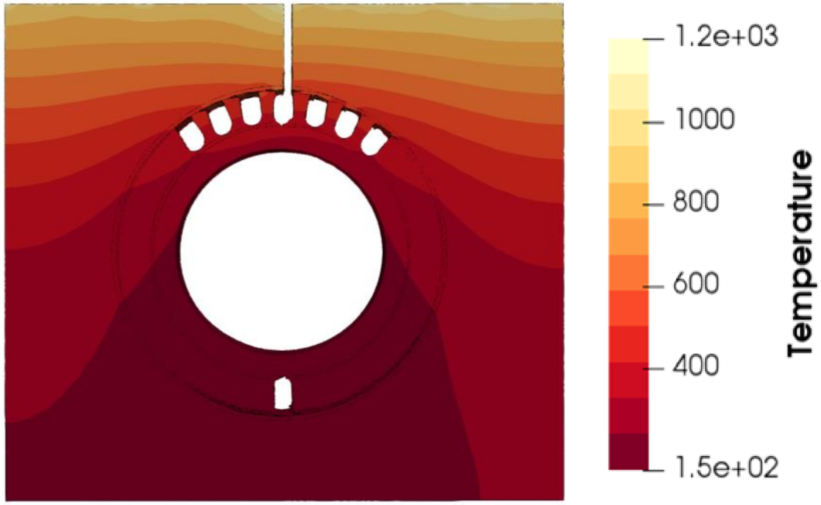
- Running multi-physics simulation.
- Running PreAster routine.
- Switching off CodeAster routine.
- Using ERMES for thermal loads.

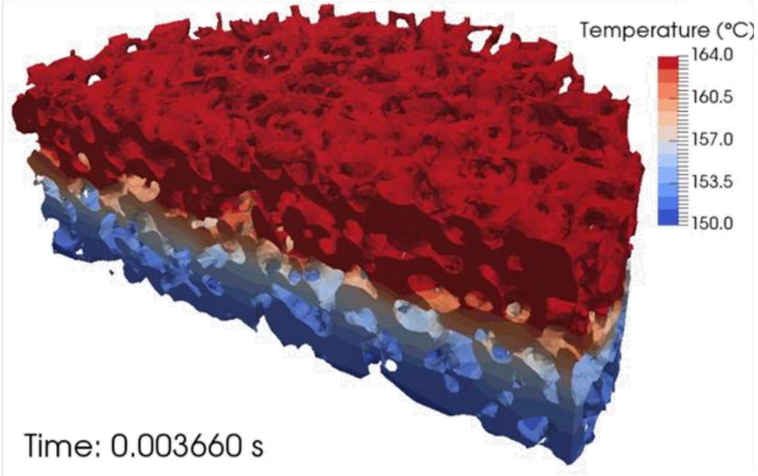
Training Format

1. Introduction to VirtualLab.
2. File layout & structure.
3. Tutorial #1: Tensile Test (Mechanical).
4. Tutorial #2: Laser Flash Analysis (Thermal).
5. Tutorial #3: HIVE Experiment (Coupled thermal-electromagnetic).
6. Tutorial #4: Image-based simulation.
7. Tutorial #5: Machine Learning.

Tutorial #4: Image-based Simulation

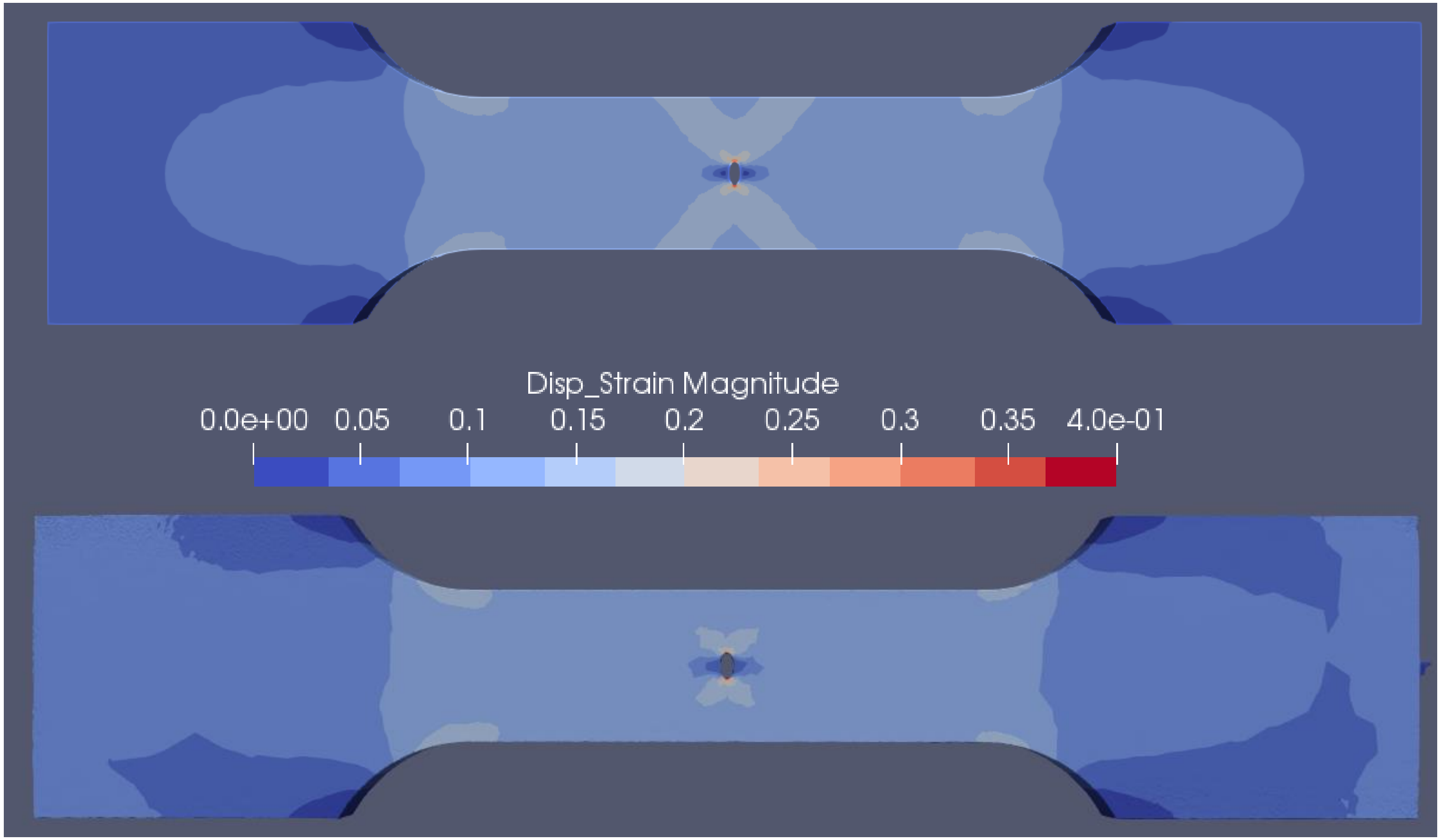
- We are also able to use VirtualLab to perform image-based simulations.
- Meshes generated from scans are used much like CAD meshes have been.





CAD

Image based



Please see screen
for instructions

- Copy the mesh 'Tensile_IBSim.med' from the VirtualLab directory to 'Output/Tensile/Tutorials/Meshes'.
- Navigate to Tutorials -> Image-based Simulation -> Tensile Test in the documentation.
- Follow the steps to complete this task.

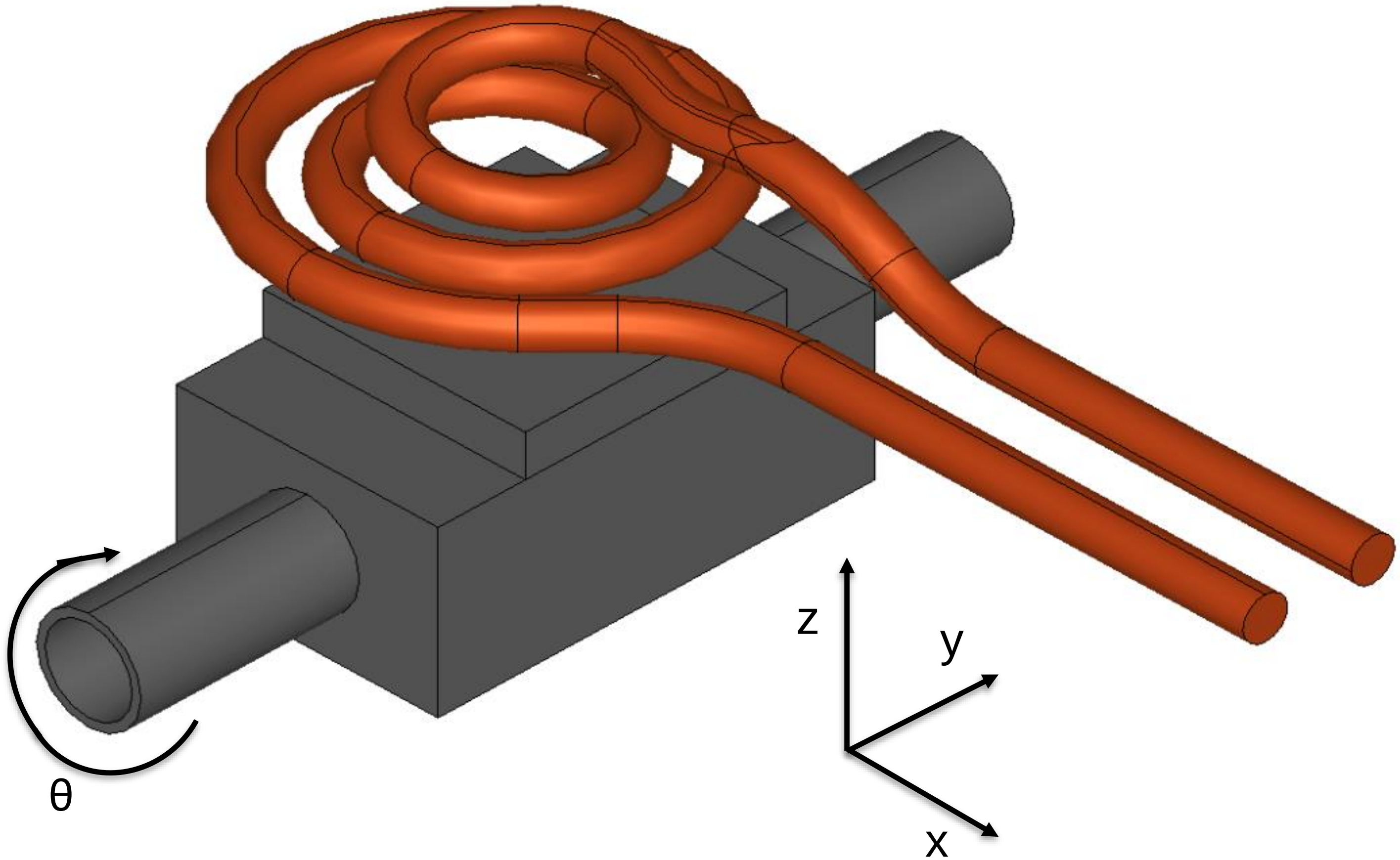
Training Format

1. Introduction to VirtualLab.
2. File layout & structure.
3. Tutorial #1: Tensile Test (Mechanical).
4. Tutorial #2: Laser Flash Analysis (Thermal).
5. Tutorial #3: HIVE Experiment (Coupled thermal-electromagnetic).
6. Tutorial #4: Image-based simulation
7. Tutorial #5: Machine Learning

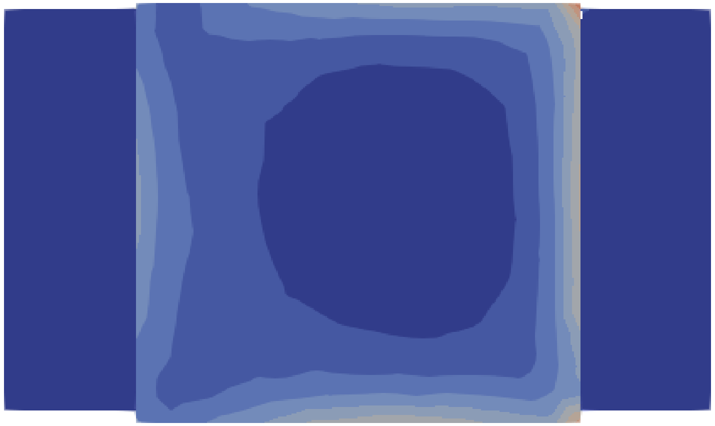
Tutorial #5: Machine Learning

- Machine learning (ML) is the science of getting computers to learn without being explicitly programmed.
- Interact with it multiple times daily.
- ML requires substantial amounts of data.
- VirtualLab is able to collect this data in a fraction of the time.

Tutorial #5: Coil Configuration

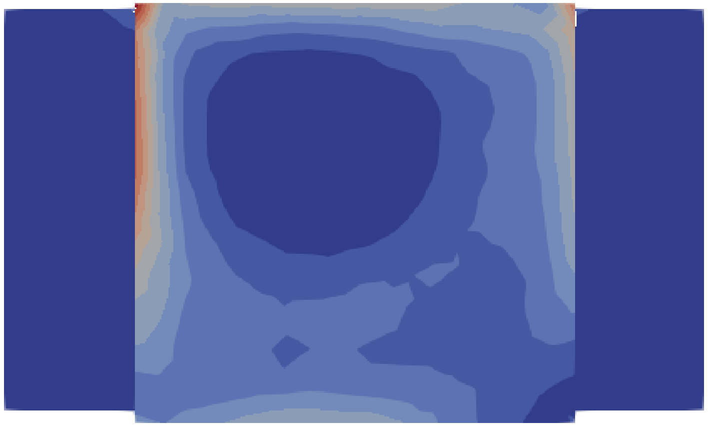


X = 0
Y = 0
Z = 0.00325
θ = 0

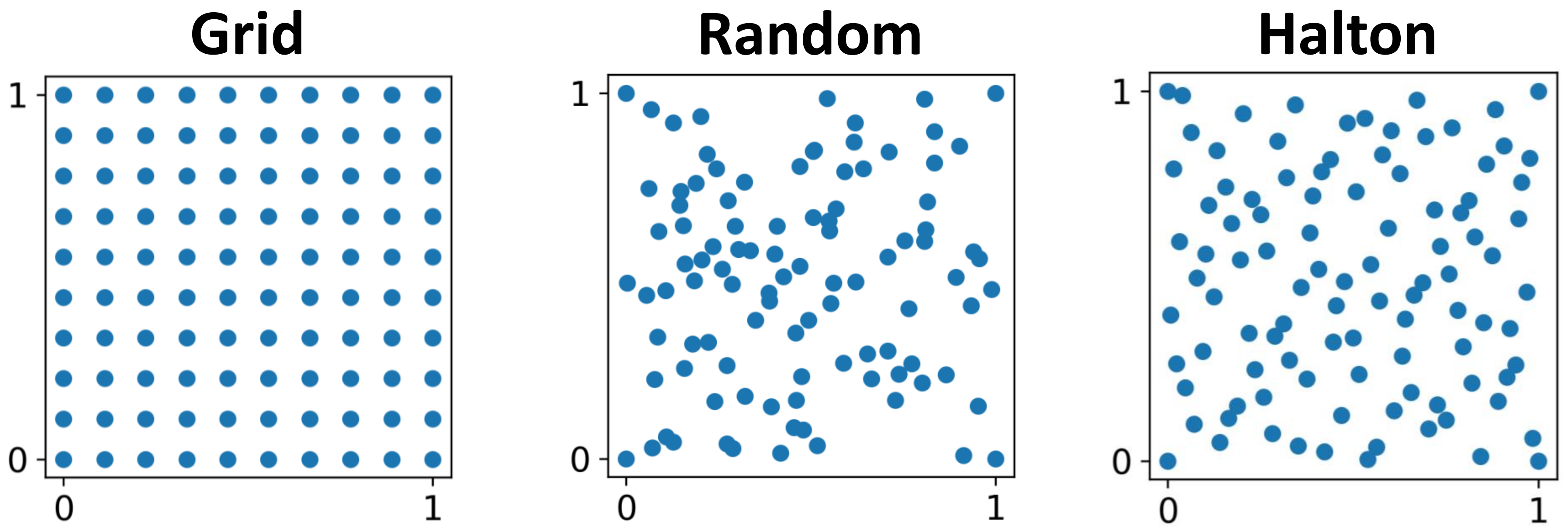


Power = 349.3
Variation = 301.6

X = 0.00625
Y = 0.00625
Z = 0.00161
θ = -1.5625



Power = 446.7
Variation = 342.5


Grid
Random
Halton
1
0
0
1

Please see screen
for instructions

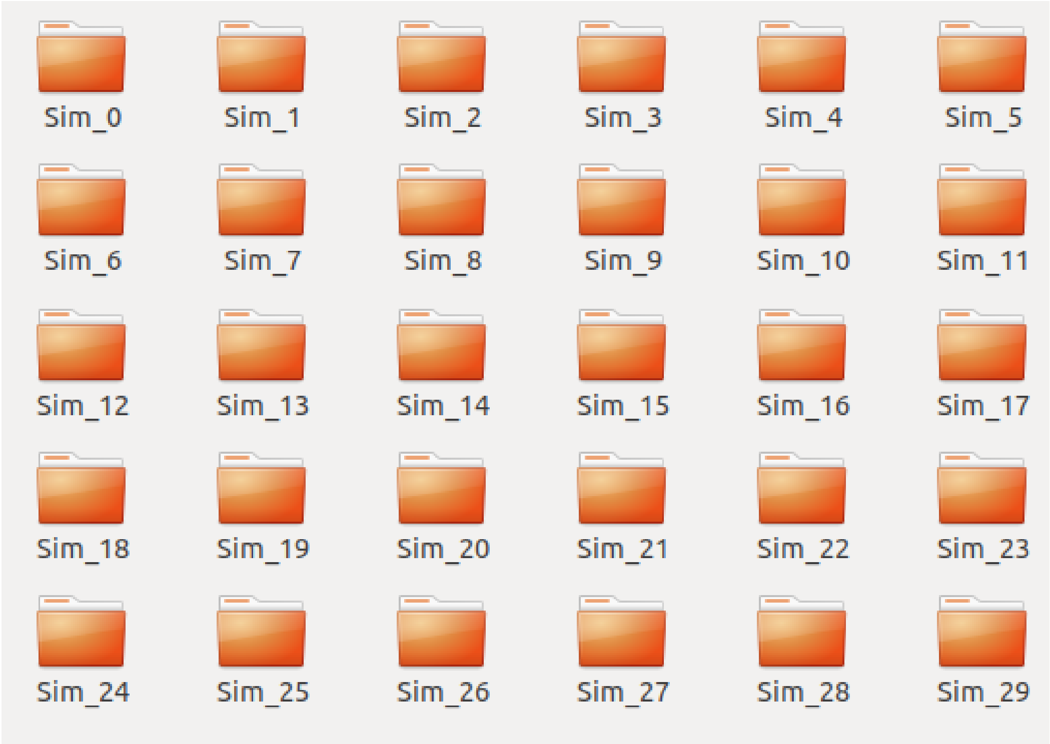
Tutorial #5: Task 1 - Parameter Space Sampling

- Open ‘RunFiles/RunTutorials.py’
- Change *Simulation* to ‘HIVE’.
- Change *Parameters_Var* to ‘PS_Sample’.
- Change *RunMesh* to False.
- Change *PreAster,Aster & PostAster (& ShowRes)* to False.
- Launch VirtualLab.

- **Note:** This will create the simulation directories but not run any of them.

Tutorial #5: Task 1 - Output

- Open ‘Output/HIVE/ Tutorials/Halton’.
- 30 simulation directories created.
- Sim_0 – Sim_15 are values on the bounds.
- Sim_16 – Sim_29 are the first 14 of the Halton sequence.



Tutorial #5: Task 1 - Output

- Open 'Sim_0/Parameters.py'.
- CoilDisplacement & Rotation are on the bounds.

```
Name = 'Halton/Sim_0'
PreAsterFile = 'PreHIVE'
CreateHTC = True
Pipe = {'Type': 'smooth tube', 'Diameter':
Coolant = {'Temperature': 20, 'Pressure':
RunERMES = True
CoilType = 'Test'
CoilDisplacement = [-0.02, -0.02, 0.0015]
Rotation = -5.0
Current = 1000
Frequency = 10000.0
NbProc = 1
Threshold = 1
NbClusters = 100
AsterFile = 'AMAZE_SS'
Mesh = 'AMAZE_Sample'
Model = '3D'
Solver = 'MUMPS'
EMLoad = 'ERMES'
Materials = {'Block': 'Copper_NL', 'Pipe':
```

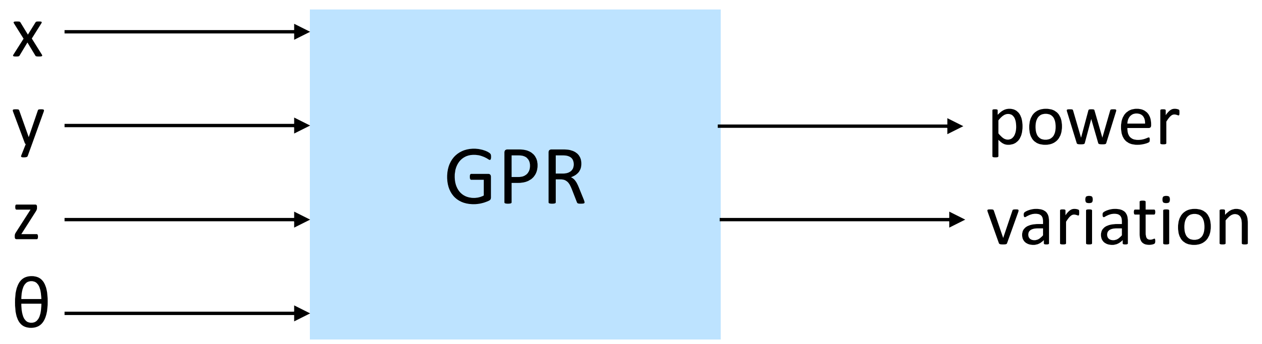
Tutorial #5: Task 1 - Output

- Open 'Sim_16/Parameters.py'.
- CoilDisplacement & Rotation inside the domain.

```
Name = 'Halton/Sim_16'
PreAsterFile = 'PreHIVE'
CreateHTC = True
Pipe = {'Type': 'smooth tube', 'Diameter': 0.01, 'Length
Coolant = {'Temperature': 20, 'Pressure': 2, 'Velocity':
RunERMES = True
CoilType = 'Test'
CoilDisplacement = [0.0, -0.006666666666666668, 0.0022]
Rotation = -3.5714285714285716
Current = 1000
Frequency = 10000.0
NbProc = 1
Threshold = 1
NbClusters = 100
AsterFile = 'AMAZE_SS'
Mesh = 'AMAZE_Sample'
Model = '3D'
Solver = 'MUMPS'
EMLoad = 'ERMES'
Materials = {'Block': 'Copper_NL', 'Pipe': 'Copper_NL',
```

Tutorial #5: Task 2 – Creating a ML Model

- A supervised ML algorithm learns the optimal mapping from the inputs to the outputs.
- Gaussian process regression (GPR) is a type of supervised ML algorithm.
- In this task we will use data collected during my PhD.

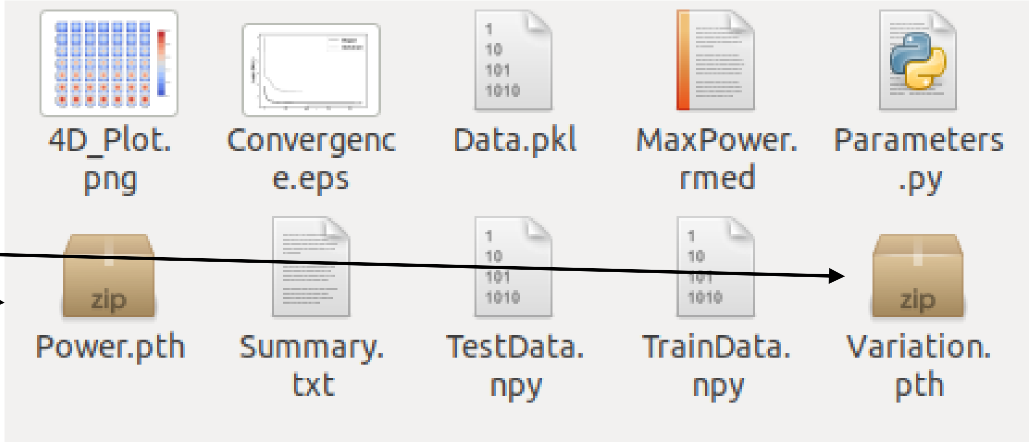


Please see screen
for instructions

Tutorial #5: Task 2 – Creating a Model

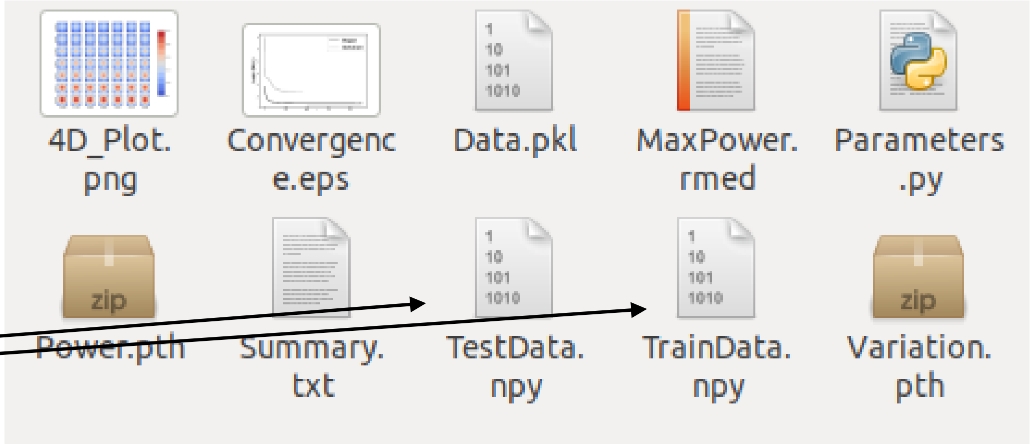
- Copy the file ‘Data.hdf5’ from the VirtualLab directory to ‘Output/HIVE/Tutorials’.
- Open ‘RunFiles/RunTutorials.py’.
- Change Parameters_Master to ‘GPR_Master’.
- Change Parameters_Var to None.

- Go to ‘Output/HIVE/Tutorials/ML/GPR_Halton.
- ML models.

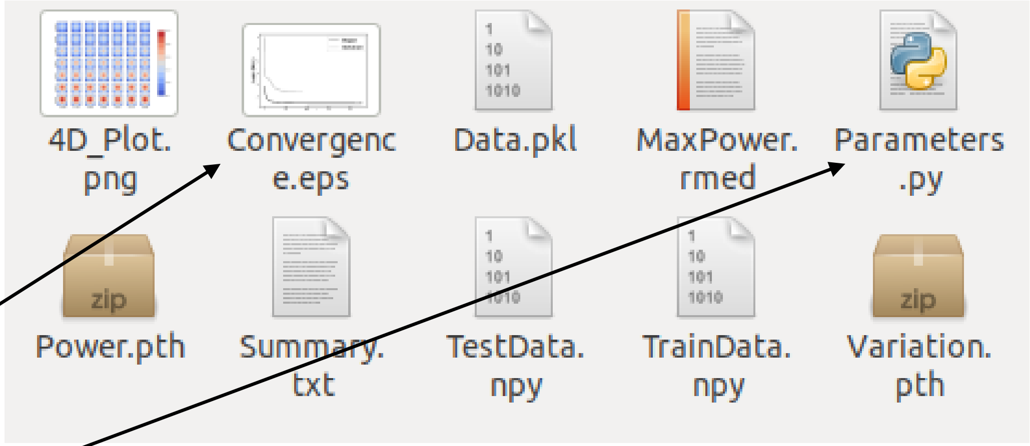


Tutorial #5: Task 2 - Output

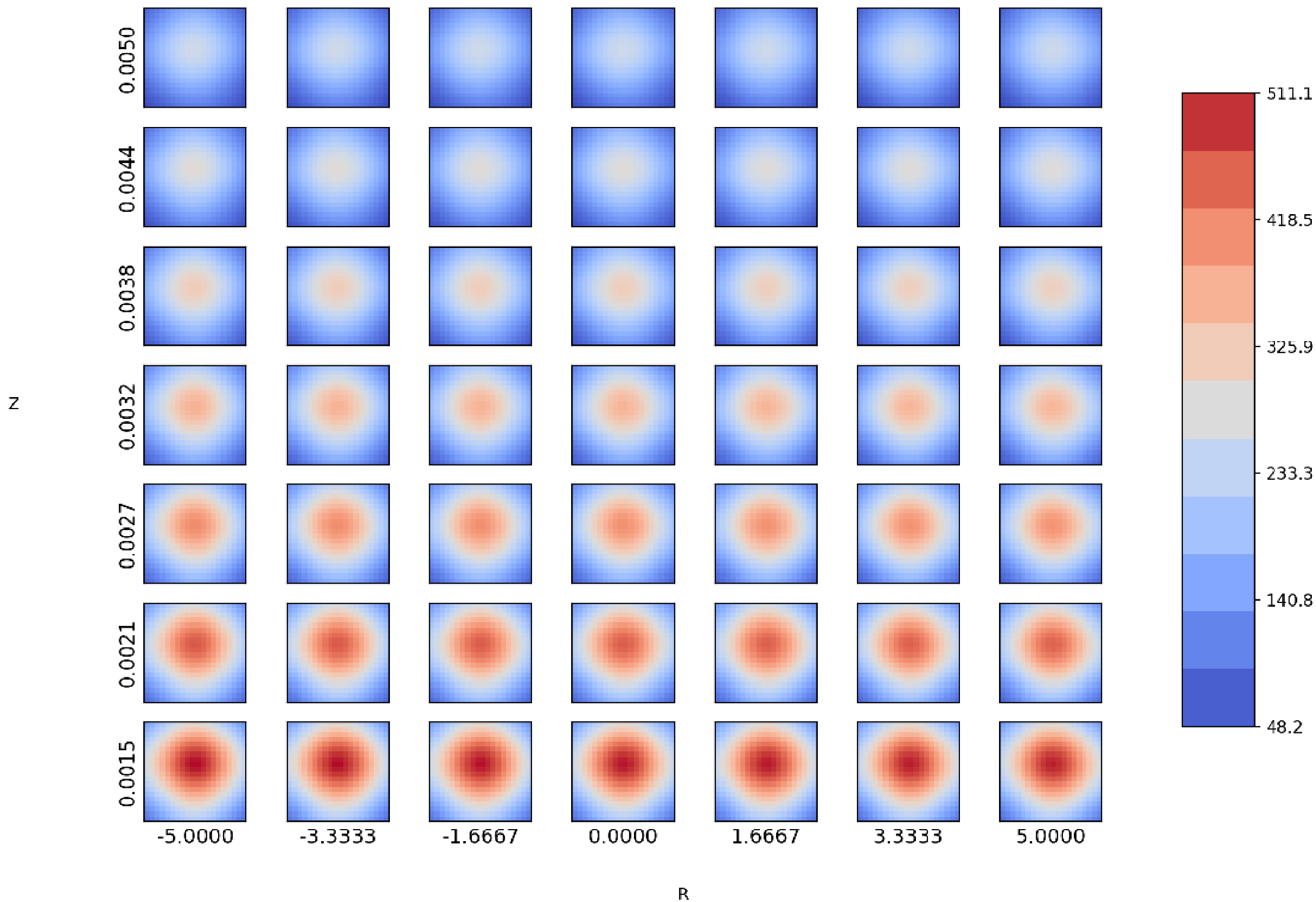
- Go to 'Output/HIVE/ Tutorials/ML/GPR_Halton.
- ML models.
- Data used for training.



- Go to ‘Output/HIVE/ Tutorials/ML/GPR_Halton.
- ML models.
- Data used for training.
- Convergence plot.
- Parameters used.



Tutorial #5: Task 2 – 4D Plot



Tutorial #5: Task 2 - Summary

```
Anticipated power at optimum configuration: 512.13 W
Actual power at optimum configuration: 521.59 W
```

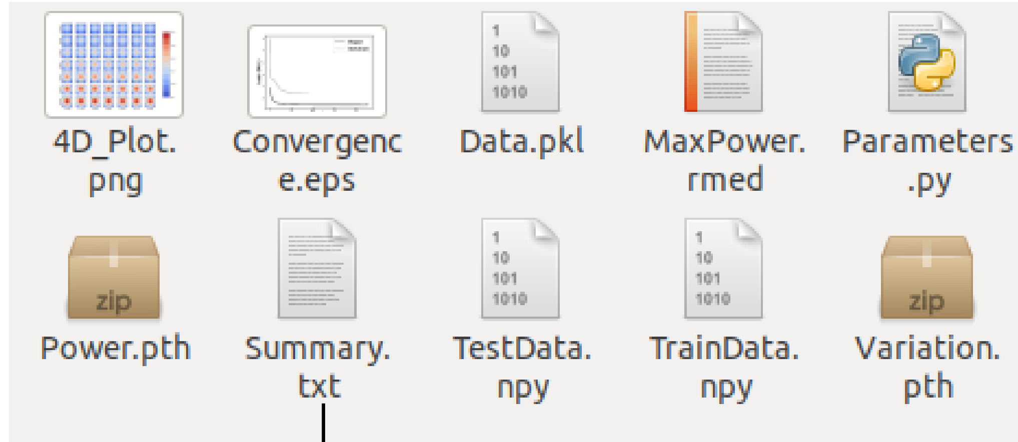
Please see screen
for instructions

Tutorial #5: Task 3 – Hybrid Halton

- Open 'Input/HIVE/Tutorials/GPR_Master.py'
- Change DA.Name to 'ML/GPR_Hybrid'.
- Change DA.TrainData to 'Hybrid_Halton/PU_3'.

Tutorial #5: Task 2 - Output

- Go to 'Output/HIVE/ Tutorials/ML/GPR_Hybrid.
- Improvement in performance at optima.




```
Anticipated power at optimum configuration: 522.90 W
Actual power at optimum configuration: 522.13 W
```

Summary: Tutorial #4 & #5

- Used an image-based mesh to run a simulation.
- Demonstrated how to sample a parameter space.
- Created a ML model using data collected.
- Discovered optimal input configuration.

Thanks for your attention