SiF2022: OpenSEES Workshop

Session 1: Getting started and nonlinear analysis in OpenSEES

Mhd Anwar Orabi

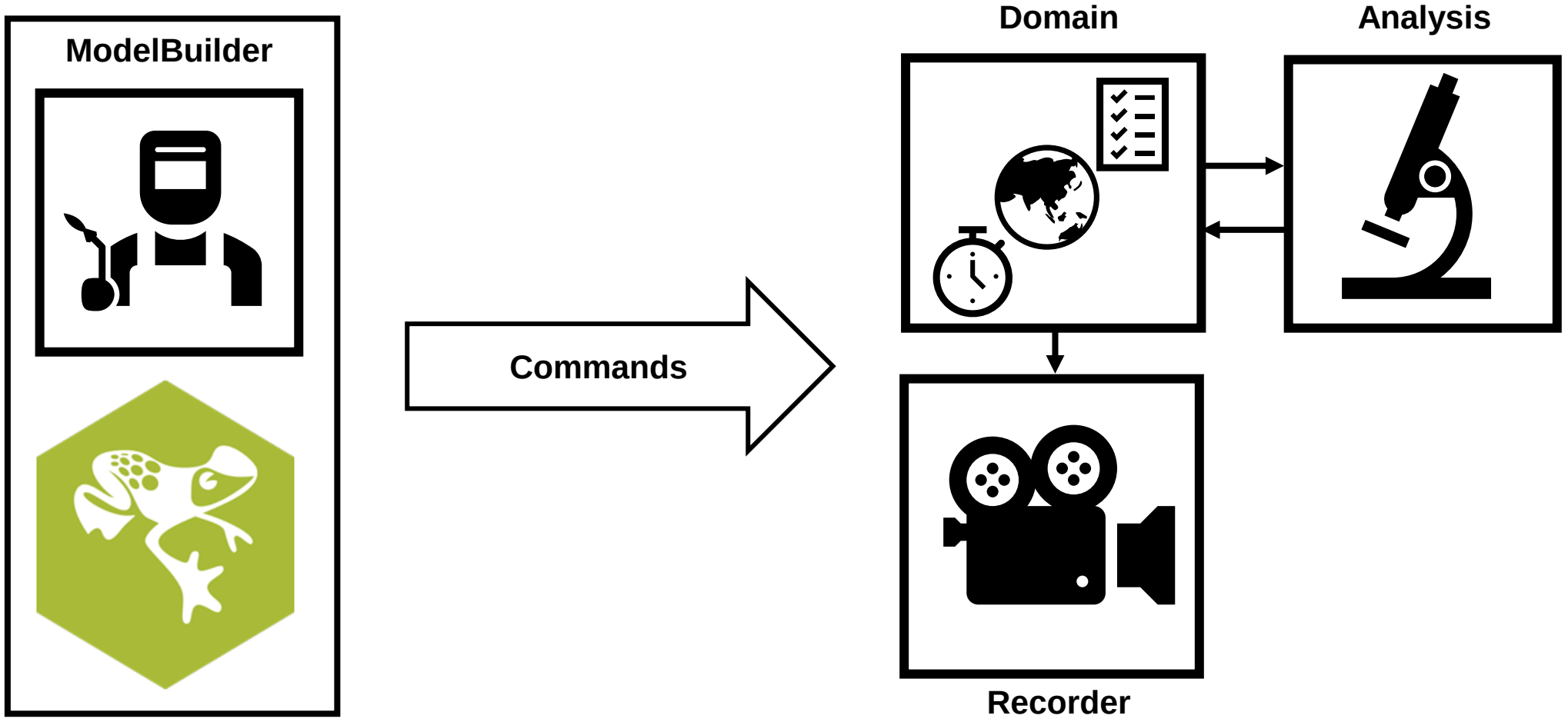
A little about me

- Call me Anwar
- BSc in Civil Engineering, MSc (DIC) in Structural Engineering
- Postdoctoral fellow at PolyU
- Using OpenSees for my research
- Interested in finite element analysis, global behaviour of and deep learning for structures in fire
- You can follow me on GitHub
- Or email me at anwar.orabi@connect.polyu.hk

In this session

- How do the different OpenSees components interact?
- What do you need to run OpenSees?
- How to write code that the OpenSees interpreter understands?
- Practical: Writing a simple OpenSees program

How does OpenSees work?



Let’s get started!

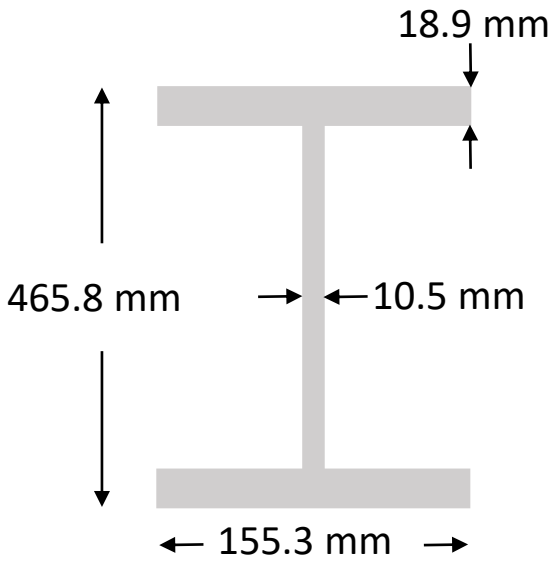
- Download Tcl/Tk
- Download the official OpenSees for Fire
- Or Anwar’s OpenSees for Fire
- Download NotePad++ (optional)

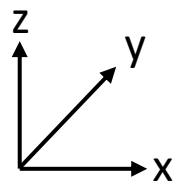
Tcl basics

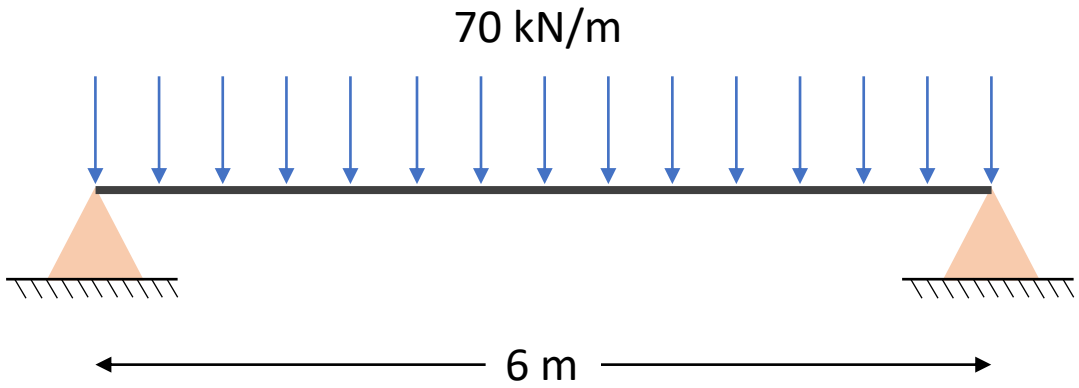
- General syntax
- Setting a variable
- Getting a variable’s value
- Mathematical expressions
- Conditionals
- Loops
- Take a look here when you’re lost

Let's try an exercise

- Axially restrained beam subjected to uniform load and then to uniform heating up to 900 °C
- UKB 457×191×98 with f_y = 275 MPa





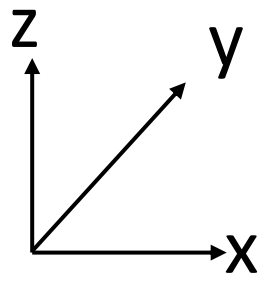


What's the process like?

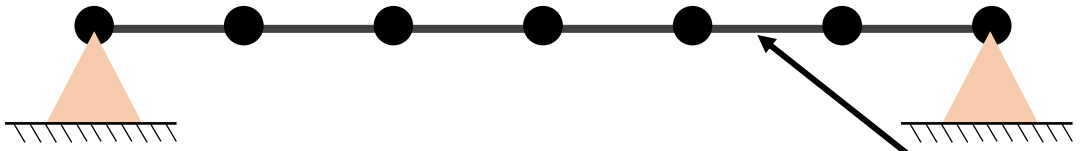
uniaxialMaterial Steel01Thermal $matTag $Fy $E0 $b

section fiberSecThermal secTag {
fiber $yLoc $zLoc $A $matTag
}

model BasicBuilder -ndm 3 -ndf 6



node nodeTag x y z



fix nodeTag u1 u2 u3 u11 u22 u33

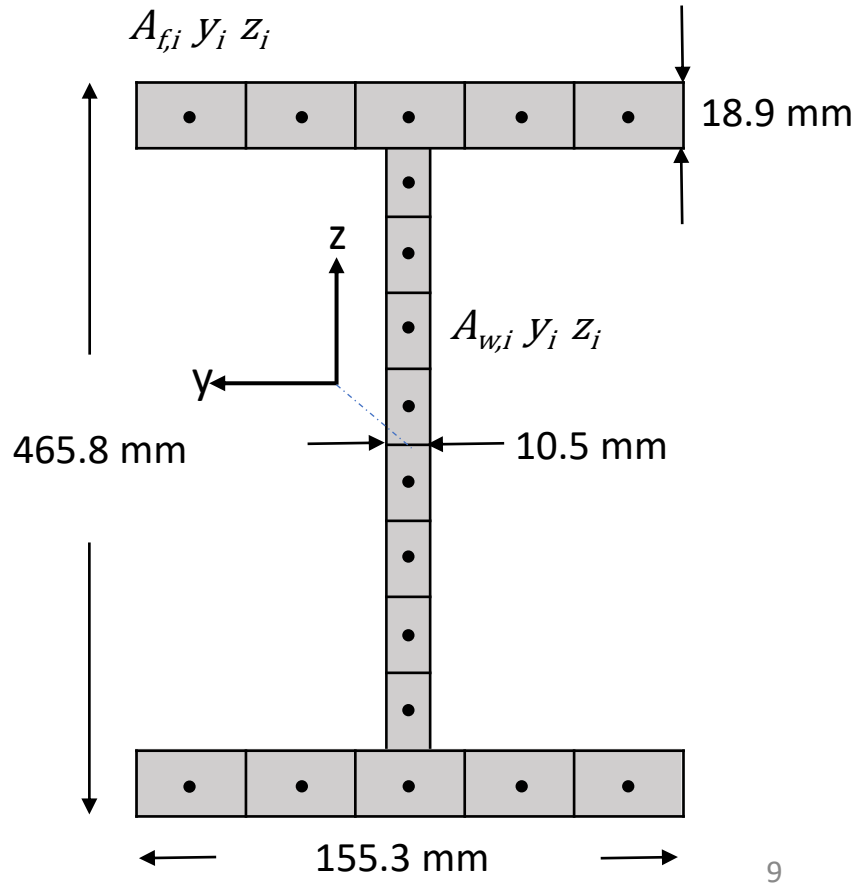
element dispBeamColumnThermal $eleID $NodeTag0 $NodeTag1 $NumInts $secTag $TransfTag

Section discretisation

- To consider material nonlinearity, we will discretise our section into fibres

$$A_{f,i} = \frac{b_f \times t_f}{5} = \frac{155.3 \times 18.9}{5} = 587\ mm^2 = 0.000587\ m^2$$

$$A_{w,i} = \frac{h_w \times t_w}{8} = \frac{(465.8 - 18.9 \times 2) \times 10.5}{8} = 561.75\ mm^2 = 0.00056175\ m^2$$

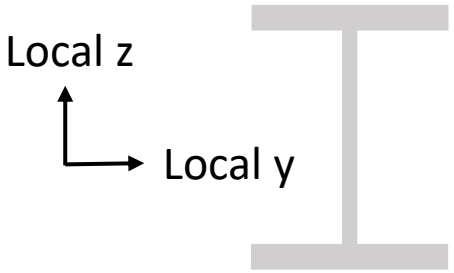


Which side is up?

- And also, how do we consider geometric nonlinearity?

1. Local x direction is from node i to node f
2. Define a vector **vecxz** that is **not** parallel to local x
3. Find local y by taking the **cross product of vecxz and local x**
4. Find local z by taking **cross product of local x and local y**

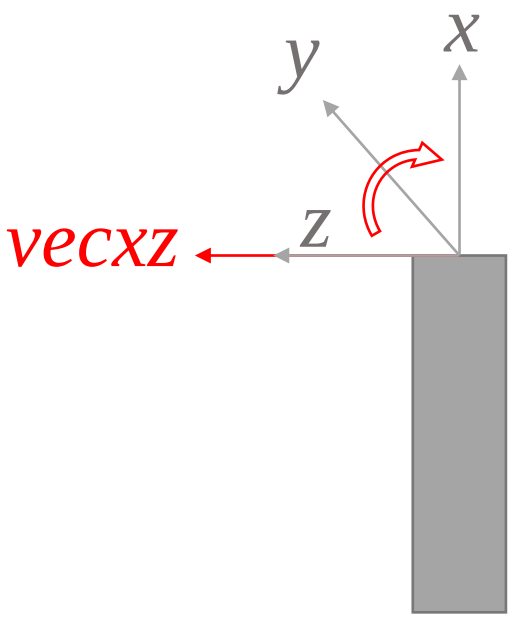
Confusing!



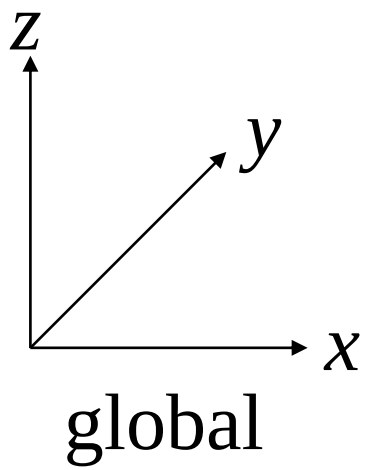
Just follow this convention

Perfectly vertical?

vecxz along negative global x!

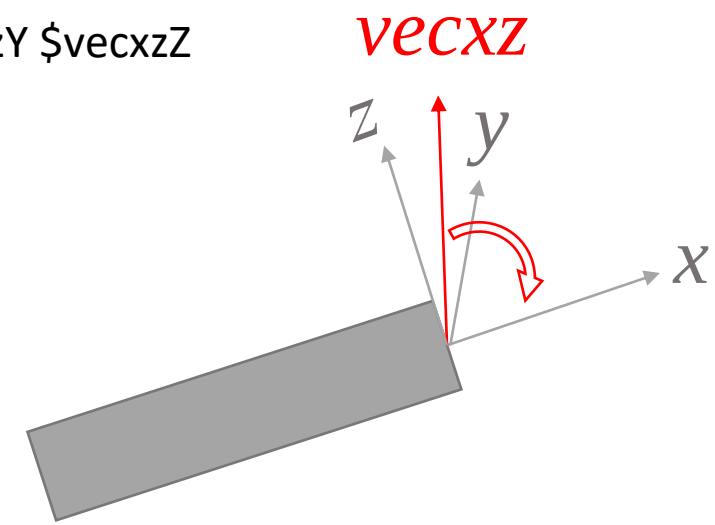


geomTransf Corotational $transfTag $vecxzX $vecxzY $vecxzZ



Otherwise:

vecxz along positive global z!



What is the workflow?

I. Building the model

1. Define the domain's dimensionality and import commands
2. Define the nodes (and nodal masses)
3. Establish nodal boundary conditions
4. Define the elements and establish their type, sections, and the relationship between local and global coordinates
5. Define the load pattern and apply to the desired nodes and/or elements
6. Define the recorder object and what it will record

What is the workflow?

II. Creating the analysis object by defining:

1. Constraints: how DOF relate to one another (compatibility)
2. Numberer: maps the nodes and their respective equations
3. System: solver and storer of the equation system
4. Test: tests for convergence during a solution algorithm
5. Algorithm: specify which solution algorithm to use for tracing the equilibrium path
6. Integrator: does the numerical work at each iteration in order to move to the next step
7. Analysis: uses all previous objects to create the analysis object

Where to get information about the commands used?

- The OpenSees command manual
- The OpenSees for Fire command manual
- The examples posted on our page

SiF2022: OpenSees Workshop

Session 2: OpenSees for Heat Transfer, and Graphical User Interface

Mhd Anwar Orabi

This session

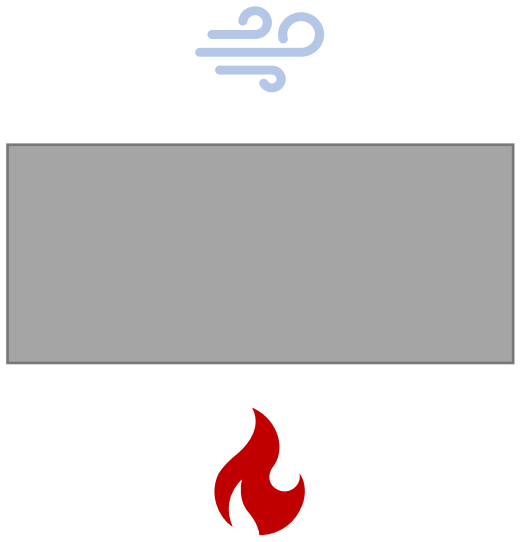
- How does heat transfer work in OpenSees?
- Can we use a graphical user interface for OpenSees for fire?

Where to get information?

- The OpenSees for Fire Heat transfer page

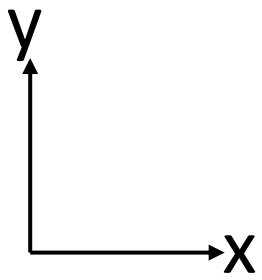
Let's try another exercise!

- A large block of steel subject to "standard" temperature-time exposure on one side, and exposed to ambient air on the other
- Assume the Eurocode 3 thermal properties

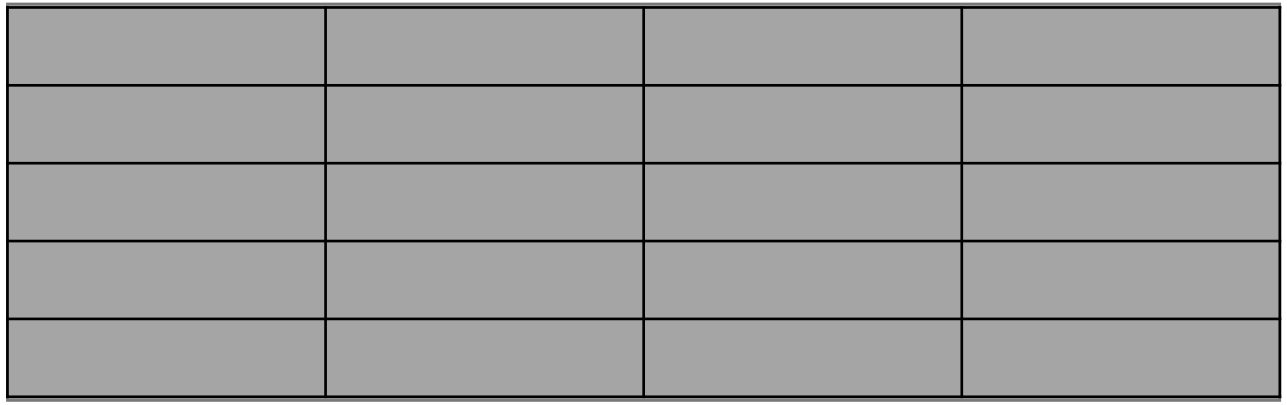


What's the process like?

HeatTransfer 2D



Create mesh controls and mesh



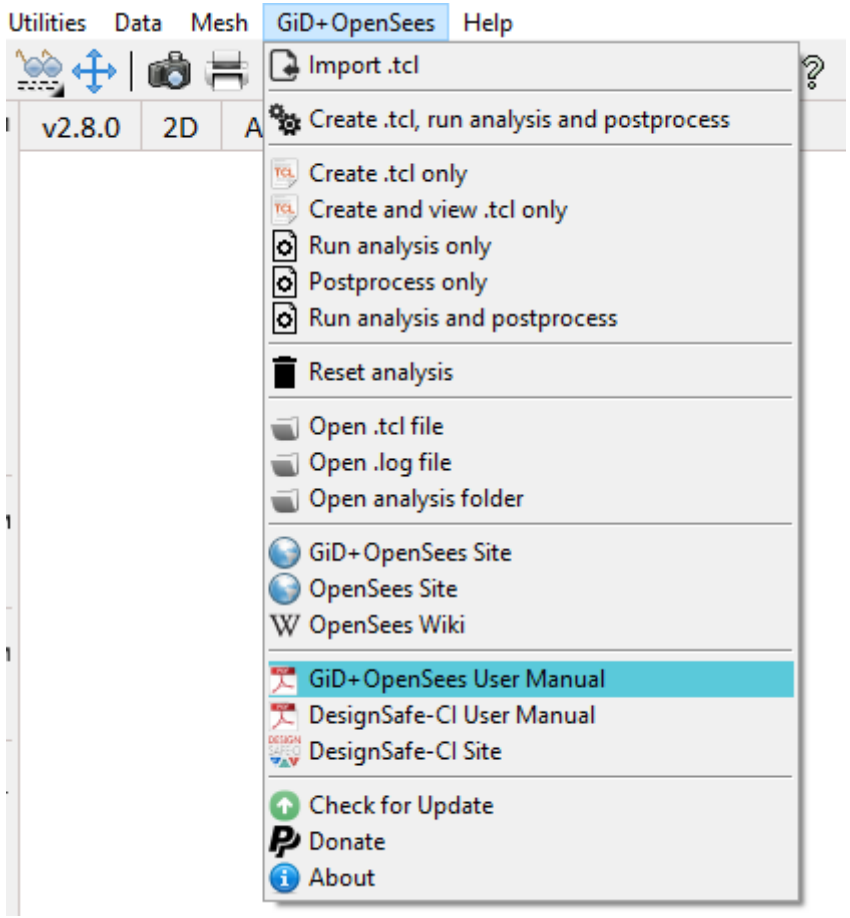
Create an HT "entity"

What is GiD?

- “GiD is a universal, adaptive and user-friendly pre and post processor for numerical simulations in science and engineering.”
- Free for up to 5000 nodes (and some additional limitations).
- Basically, a graphical interface we can use for OpenSees.
- Let’s start setting it up:
- GiD
- GiD+OpenSees Interface 2.8.0
- OpenSees for Fire Interface

Is there any documentation?

- GiD has an extensive set of documentation found here.
- You got the extensive set of documentation that Aristotle University of Thessaloniki wrote for the original interface with your download.
- As you will see in this workshop, this is almost all you need.



Let’s jump right in!

- Two-bay steel frame
- Linear gradient with 800 °C bottom and 100 °C top

Follow this checklist:

- Define units
- Define geometry
- Check and refresh model dimensions to 3D
- Define materials
- Define sections
- Define elements
- Assign elements
- Assign boundary conditions
- Set analysis interval
- Assign loads
- Assign mesh controls
- Mesh
- Check connectivity
- Check outputs
- Create Tcl file
- Check Tcl file for compliance with your model
- Run analysis