

FeenoX for Engineers

2024-06-14

Contents

1	How FeenoX works	2
2	What Feenox can solve	3
3	Why FeenoX works the way it does	4

The NAFEMS LE10 problem statement and the corresponding FeenoX input

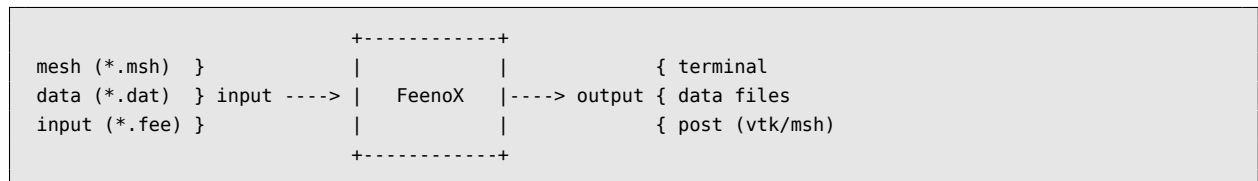
Figure 1: The NAFEMS LE10 problem statement and the corresponding FeenoX input

1 How FeenoX works

Heads up! First things first:

1. FeenoX is not a traditional point-and-click finite-element solver. It **does not include** a graphical interface.
2. FeenoX is designed as a cloud-first back end for generic computational workflows to solve engineering-related problems. One of these many workflows can be point-and-click graphical interface such as CAEplex which allows to solve thermo-mechanical problems directly from the browser.
3. Since it is a cloud-first tool, FeenoX targets at GNU/Linux. If you are using Windows, FeenoX may not be for you (but CAEplex may). See below to find out why FeenoX works the way it does.

In plain engineering terms, FeenoX works like a transfer function between one or more input files and zero or more output files:



When solving problems using the finite element method, FeenoX sits in the middle of pre and post-processing tools. The former should aid the engineer to prepare the mesh and, eventually, the input file. The latter should show the numerical results. See the tensile test tutorial for an in-depth step-by-step explanation.

To fix ideas, let us consider the NAFEMS LE10 “Thick plate pressure” benchmark. Fig. 1 shows that there is a one-to-one correspondence between the human-friendly problem formulation and the input file FeenoX reads. There is no need to give extra settings if the problem does not ask for them. Note that since the problem has only one volume, E means “the” Young modulus. No need to deal with a map between materials and mesh entities (in this case the mapping is not needed but in multi-material problems the mapping is needed indeed). Nothing more, nothing less.

Say we already have a `nafems-le10.geo` file which tells Gmsh how to create a mesh `nafems-le10.msh` (check out the tensile test tutorial for details). Then, we can create an input file for FeenoX (using editors with syntax highlighting for example) as follows:

```

# NAFEMS Benchmark LE-10: thick plate pressure
PROBLEM mechanical DIMENSIONS 3
READ_MESH nafems-le10.msh # mesh in millimeters

# LOADING: uniform normal pressure on the upper surface
BC upper p=1 # 1 Mpa

# BOUNDARY CONDITIONS:
BC DCD'C' v=0 # Face DCD'C' zero y-displacement
BC ABA'B' u=0 # Face ABA'B' zero x-displacement
    
```

```
BC BCB'C' u=0 v=0 # Face BCB'C' x and y displ. fixed
BC midplane w=0 # z displacements fixed along mid-plane

# MATERIAL PROPERTIES: isotropic single-material properties
E = 210e3 # Young modulus in MPa
nu = 0.3 # Poisson's ratio

SOLVE_PROBLEM # solve!

# print the direct stress y at D (and nothing more)
PRINT "σ_y @ D = " sigmay(2000,0,300) "MPa"
```

Heads up! The .msh file from Gmsh can be either

- a. version 2.2, or
- b. version 4.0, or
- c. version 4.1.

and can be partitioned or not.

Once we put these two files, `nafems-1e10.geo` and `nafems-1e10.fee` in the same directory, say in the `examples` ↔ directory of the repository, then we call first Gmsh and then FeenoX from the terminal to solve the benchmark problem:

Check out the section about invocation in the FeenoX manual.

The trick is that this workflow is susceptible of being automated and customized to run in the cloud, possibly in parallel throughout several servers using the MPI standard. Check out this 1-minute video that covers a similar case from the tensile-test tutorial:

There is a sound explanation about why FeenoX works this way and not like other FEA tools you might have encountered in the past. If you are feeling curious, take a look at what FeenoX has to offer to hackers and academics.

2 What Feenox can solve

FeenoX can solve the following types of problems:

- Basic mathematics
- Systems of ODEs/DAEs
- Laplace's equation
- Heat conduction
- Linear elasticity
- Modal analysis
- Neutron diffusion
- Neutron SN

Take the tutorials to learn how to solve those types of problems:

0. Setting up your workspace
1. Overview: the tensile test case
2. Fun & games: solving mazes with PDES instead of AI
3. Heat conduction

Browse through the documentation to dive deeper into the details.

3 Why FeenoX works the way it does

There are two “why” questions we have to answer.

1. Why is FeenoX different from other “similar” tools?

Consider again the NAFEMS LE10 case from fig. 1 above. Take some time to think (or investigate) how other FEA tools handle this case. Note the following features FeenoX provides:

- ready-to-run executable that reads the problem at runtime (no need to compile anything to solve a particular problem)
- self-explanatory plain-text near-English input file
 - one-to-one correspondence between computer input file and human description of the problem
 - Git-traceable (mesh is not part of the input file)
 - everything is an expression
 - * material properties can be given as functions of space and/or time and/or temperature (or other intermediate variables)
 - * these functions can be given as algebraic expressions or come from interpolated point-wise defined data
- 100% user-defined output
- simple problems need simple inputs
- similar problems need similar inputs
- parametric and optimization runs through command-line arguments
- flexibility to handle many workflows, including web-based interfaces

2. Why FeenoX works the way it does?

Because it is cloud first and its objective is to be flexible enough to power web-based interfaces like CAEplex and many other workflows. More information in the details for Unix experts and academic professors.

Depending on the complexity of the case, CAEplex might be enough or not. If the latter, one has to see what’s sitting under the hood. Peek into the documentation and the repository for further details.