Statement of need

Open-source finite-element tools are either

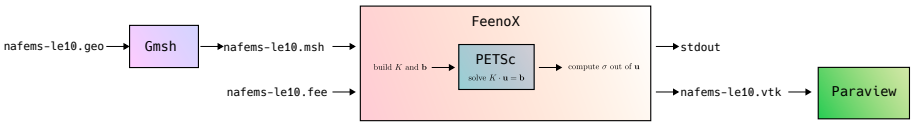
a. libraries which need code to use them such as
   - Sparselizard (Halbach, 2017)
   - MoFEM (Kaczmarczyk et al., 2020)
   - FEniCS (Baratta et al., 2023)
   - MFEM (Anderson et al., 2021)
b. end-user programs which need a GUI such as
   - CalculiX
   - CodeAster

FeenoX sits in the middle. First, it can solve

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

Second, it is the only free and open-source tool that satisfies the Software Requirement Specifications, including that...

- in order to solve a problem one needs to prepare a (relatively) simple input file (not a script nor a deck) which is read at run-time (not code which calls a library). For example, considering the NAFEMS LE10 Benchmark problem from fig. 1, FeenoX works as two "glue layers" (Raymond, 2003)
  1. between the mesher Gmsh (Geuzaine & Remacle, 2009) and the PETSc library (Balay et al., 1997, 2023)
  2. between the PETSc library and a post-processor such as Paraview (Ayachit, 2015)



- these input files can expand generic command-line options using Bash syntax as `$1`, `$2`, etc., which allow parametric or optimization loops driven by higher-level scripts.
- for solving partial differential equations (PDEs), the input file has to refer to at least one Gmsh `.msh` file that defines the domain where the PDE is solved.
- the material properties and boundary conditions are defined using physical groups and not individual nodes nor elements, so the input file is independent of the mesh and thus can be tracked with Git to increase traceability and repeatability.
- it follows the Unix philosophy (Raymond, 2003) which, among others, separates policy from mechanism rendering FeenoX as a natural choice for web-based interfaces like CAEplex (fig. 2).