FeenoX

A cloud-first free no-fee no-X uniX-like finite-element(ish) computational engineering tool

Contents

1 About FeenoX

FeenoX is a cloud-first free and open source tool to solve engineering-related problems using a computer (or many computers in parallel) with a particular design basis.

Choose your background for further details about the what, how and whys:

- Industry Engineer
- Unix Hacker
- Academic Professor

1.1 Extents

1.1.1 Examples

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

1.1.2 Tutorials

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

1.1.3 Tests

The `tests` directory in the repository has hundreds of

- `grep`-able examples
- unit and regression tests,
- (crude) mathematical & code verification tests (as in "are we solving right the equations?"),
- subdirectories with further case studies
  - mms
    - thermal
      - 2d
      - 3d
  - nafems
    - le10

1.2 Capabilities

- Both *free* as in "free speech" and in "free beer"

- The problem to solve is defined through a syntactically-sugared self-descriptive English-like plain-text input file that should resemble the original human-friendly problem formulation as much as possible:
  - no need to recompile if the problem changes (FeenoX is a *program*, not a library)
  - nouns are definitions and verbs are instructions
  - simple problems need simple inputs
  - everything is an expression
  - 100%-defined user output (no `PRINT` nor `WRITE_RESULTS` instructions, no output)
- Cloud-first design (cloud friendliness is not enough)
- MPI parallelization
- Leverages high-quality well-established free and open source libraries to solve…
  - general mathematical problems using GNU GSL
  - sets of ODEs/DAEs using SUNDIALS
  - PDEs formulated with the finite element method
    * reads mesh in Gmsh format
    * uses PETSc/SLEPc to solve
      · linear systems (KSP)
      · non-linear systems (SNES)
      · time-dependent systems (TS)
      · generalized eigen-value problems (EPS)
    * writes results in either Gmsh or VTK (Paraview) format
- Focuses on flexibility, especially when defining non-uniform multi-solid material properties from ASME tables
- Follows the Unix programming philosophy
  - rule of separation
  - rule of silence
  - rule of economy
- Each PDE (i.e. from Laplace downward in the list of examples) is implemented in a subdirectory within `src/pde` of the source tree
  - any subdirectory can be removed if a particular PDE is not needed
  - any subdirectory can be used as a template to add a new PDE to the capabilities
- Space, time and/or solution-dependent material properties and boundary conditions
  - algebraic expressions, and/or
  - point-wise interpolated data
- Command-line argument expansion for
  - increased flexibility,
  - parametric sweeps, and/or
  - optimization loops
- Steady-state, [quasi-static] and/or transient problems
- Linear and non-linear problems
- Possibility to verify the code using the Method of Manufactured Solutions
- Separate repository to profile and study code performance using Google's benchmark library

1.3   Usefulness

- CAEplex: a web-based thermo-mechanical solver running on the cloud
- Non-conformal mesh mapping
- ASME stress linearization for pressurized pipes and vessels

FeenoX

- Assessment of material properties from tabulated sources
- Environmentally-assisted fatigue analysis in dissimilar interfaces of nuclear pipes
- Neutron transport in the cloud
- Solving mazes without AI
- Parametric NAFEMS LE10 benchmark: comparison of resource consumption for different FEA programs
- Some Youtube videos

2 Documentation

Browse through the documentation index and/or the `doc` subdirectory of the Github repository for

- FAQs, including what FeenoX means
- Manual
- Description
- Software Design Requirements
- Software Design Specifications
- Unix man page (accessible through `man feenox` after installation)
- History
- Compilation guide
- Programming guide

3 Quickstart

3.1 Download

FeenoX is distributed under the terms of the GNU General Public License version 3 or (at your option) any later version.

Debian/Ubuntu packages (unofficial)	https://www.seamplex.com/feenox/dist/deb
GNU/Linux static binaries	https://www.seamplex.com/feenox/dist/linux
Windows binaries	https://www.seamplex.com/feenox/dist/windows
Source tarballs	https://www.seamplex.com/feenox/dist/src
Github repository	https://github.com/seamplex/feenox/

- FeenoX is cloud-first. It was designed to run on servers.

- Be aware that FeenoX **does not have a GUI**. Read the documentation, especially the description and the FAQs. Ask for help on the GitHub discussions page if you do now understand what this bullet means.

- Debian/Ubuntu packages are unofficial, i.e. they are not available in `apt` repositories. They contain dynamically-linked binaries and their dependencies are hard-coded for each Debian/Ubuntu release. Make sure you get the right `.deb` for your release (i.e. `bookworm/bullseye` for Debian, `kinetic/focal` for Ubuntu).

- Generic GNU/Linux binaries are provided as statically-linked executables for convenience. They do not support MUMPS nor MPI and have only basic optimization flags. Please compile from source for high-end applications. See detailed compilation instructions.

- Try to avoid Windows as much as you can. The binaries are provided as transitional packages for people that for some reason still use such an outdated, anachronous, awful and invasive operating system. They are compiled with Cygwin and have no support whatsoever. Really, really, **get rid of Windows ASAP**.

  > "It is really worth any amount of time and effort to get away from Windows if you are doing computational science."
  >
  > https://lists.mcs.anl.gov/pipermail/petsc-users/2015-July/026388.html

3.2   Git repository

To compile the Git repository, proceed as follows. This procedure does need `git` and `autoconf` but new versions can be pulled and recompiled easily. If something goes wrong and you get an error, do not hesitate to ask in FeenoX's discussion page.

1. Install mandatory dependencies

```
sudo apt-get update
sudo apt-get install gcc make git automake autoconf libgsl-dev
```

   If you cannot install `libgsl-dev` but still have `git` and the build toolchain, you can have the `configure` script to download and compile it for you. See point 4 below.

2. Install optional dependencies (of course these are *optional* but recommended)

```
sudo apt-get install libsundials-dev petsc-dev slepc-dev
```

3. Clone Github repository

```
git clone https://github.com/seamplex/feenox
```

4. Boostrap, configure, compile & make

```
cd feenox
./autogen.sh
./configure
make -j4
```

   If you cannot (or do not want to) use `libgsl-dev` from a package repository, call `configure` with `-- ↩
enable-download-gsl`:

```
./configure --enable-download-gsl
```

   If you do not have Internet access, get the tarball manually, copy it to the same directory as `configure` and run again. See the detailed compilation instructions for an explanation.

5. Run test suite (optional)

```
make check
```

6. Install the binary system wide (optional)

FeenoX

```
sudo make install
```

To stay up to date, pull and then autogen, configure and make (and optionally install):

```
git pull
./autogen.sh; ./configure; make -j4
sudo make install
```

See the download page and the compilation guide for detailed information.

4   Licensing

FeenoX is distributed under the terms of the GNU General Public License version 3 or (at your option) any later version. The following text was borrowed from the Gmsh documentation. Replacing "Gmsh" with "FeenoX" (using Unix's sed) gives:

> FeenoX is "free software"; this means that everyone is free to use it and to redistribute it on a free basis. FeenoX is not in the public domain; it is copyrighted and there are restrictions on its distribution, but these restrictions are designed to permit everything that a good cooperating citizen would want to do. What is not allowed is to try to prevent others from further sharing any version of FeenoX that they might get from you.
>
> Specifically, we want to make sure that you have the right to give away copies of FeenoX, that you receive source code or else can get it if you want it, that you can change FeenoX or use pieces of FeenoX in new free programs, and that you know you can do these things.
>
> To make sure that everyone has such rights, we have to forbid you to deprive anyone else of these rights. For example, if you distribute copies of FeenoX, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must tell them their rights.
>
> Also, for our own protection, we must make certain that everyone finds out that there is no warranty for FeenoX. If FeenoX is modified by someone else and passed on, we want their recipients to know that what they have is not what we distributed, so that any problems introduced by others will not reflect on our reputation.
>
> The precise conditions of the license for FeenoX are found in the General Public License that accompanies the source code. Further information about this license is available from the GNU Project webpage http://www.gnu.org/copyleft/gpl-faq.html.

FeenoX is licensed under the terms of the GNU General Public License version 3 or, at the user convenience, any later version. This means that users get the four essential freedoms:[1]

0. The freedom to *run* the program as they wish, for *any* purpose.

---

[1] There are some examples of pieces of computational software which are described as "open source" in which even the first of the four freedoms is denied. The most iconic case is that of Android, whose sources are readily available online but there is no straightforward way of updating one's mobile phone firmware with a customized version, not to mention vendor and hardware lock ins and the possibility of bricking devices if something unexpected happens. In the nuclear industry, it is the case of a Monte Carlo particle-transport program that requests users to sign an agreement about the objective of its usage before allowing its execution. The software itself might be open source because the source code is provided after signing the agreement, but it is not free (as in freedom) at all.

FeenoX

1. The freedom to *study* how the program works, and *change* it so it does their computing as they wish.
2. The freedom to *redistribute* copies so they can help others.
3. The freedom to *distribute* copies of their *modified* versions to others.

So a free program has to be open source, but it also has to explicitly provide the four freedoms above both through the written license and through appropriate mechanisms to get, modify, compile, run and document these modifications using well-established and/or reasonable straightforward procedures. That is why licensing FeenoX as GPLv3+ also implies that the source code and all the scripts and makefiles needed to compile and run it are available for anyone that requires it (i.e. it is compiled with `./configure ↩ && make`). Anyone wanting to modify the program either to fix bugs, improve it or add new features is free to do so. And if they do not know how to program, the have the freedom to hire a programmer to do it without needing to ask permission to the original authors. Even more, the documentation is released under the terms of the GNU Free Documentation License so these new (or modified) features can be properly documented as well.

Nevertheless, since these original authors are the copyright holders, they still can use it to either enforce or prevent further actions from the users that receive FeenoX under the GPLv3+. In particular, the license allows re-distribution of modified versions only if

a. they are clearly marked as different from the original, and
b. they are distributed under the same terms of the GPLv3+.

There are also some other subtle technicalities that need not be discussed here such as

- what constitutes a modified version (which cannot be redistributed under a different license)
- what is an aggregate (in which each part be distributed under different licenses)
- usage over a network and the possibility of using AGPL instead of GPL to further enforce freedom

These issues are already taken into account in the FeenoX licensing scheme.

It should be noted that not only is FeenoX free and open source, but also all of the libraries it depends on (and their dependencies) also are. It can also be compiled using free and open source build tool chains running over free and open source operating systems.

4.1 Contributing

Contributions from hackers and/or academics are welcome, especially new types of PDEs and new formulations of existing PDEs. For elliptic operators feel free to use the Laplace equation at `src/pdes/laplace` as a template.

1. Read the Programming Guide.
2. Browse Github discussions and open a new thread explaining what you want to do and/or asking for help.
3. Fork the Git repository under your Github account
4. Create a pull request, including
   - code,
   - documentation, and
   - tests.
5. Follow up the review procedure.

Note that

- It is mandatory to observe the Code of Conduct.

FeenoX

- The contributed code has to be compatible with the GPLv3+ license.
- Each author keeps the copyright of the contribution.
- You can ask!

5   Further information

Home page: https://www.seamplex.com/feenox
Repository: https://github.com/seamplex/feenox
Bug reporting: https://github.com/seamplex/feenox/issues
Discussions: https://github.com/seamplex/feenox/discussions
Follow us: YouTube LinkedIn Github