Building femm-qt on Linux

D. Meeker

May 19, 2026

1 Overview

femm-qt is a cross-platform port of FEMM (Finite Element Method Magnetics) using Qt 6 for the GUI and CMake/Ninja for the build system. This document describes how to build it from source on Linux. The instructions target Ubuntu 24.04 LTS but should work with minor changes on other distributions (Debian, Fedora, Arch, etc.).

The build produces six executables:

- `femm_gui` — the main GUI application
- `femm_headless` — headless IPC server (no GUI)
- `fkn` — magnetics solver
- `belasolv` — electrostatics solver
- `hsolv` — heat flow solver
- `csolv` — current flow solver

2 Prerequisites

2.1 Build Tools

Install the compiler toolchain, CMake, Ninja, and git:

```
sudo apt update
sudo apt install build-essential cmake ninja-build git
```

This provides `g++`, `cmake`, `ninja`, and `git`.

2.2 Qt 6

Install the Qt 6 development libraries:

```
sudo apt install qt6-base-dev libqt6widgets6
```

On Ubuntu 24.04, this installs Qt 6.4 or later from the standard repositories. No separate Qt installer or online account is needed.

To verify the installation:

```
qmake6 --version
```

2.3 OpenMP (Optional)

GCC ships with OpenMP support by default on Linux. The solvers will automatically use multi-threaded solving with no extra packages needed. If for some reason OpenMP is not found, install the runtime:

```
sudo apt install libgomp1
```

2.4 Other Distributions

On Fedora:

```
sudo dnf install gcc-c++ cmake ninja-build git qt6-qtbase-devel
```

On Arch Linux:

```
sudo pacman -S base-devel cmake ninja git qt6-base
```

3 Getting the Source

If you have a source zip, unzip it:

```
unzip femm-qt-src.zip
cd femm-qt
```

Or, if the source is hosted on GitHub, clone the repository:

```
git clone <repository-url> femm-qt
cd femm-qt
```

4 Building

From the `femm-qt` directory, run the build script:

```
./linux_build.sh
```

This checks that the build tools are present, configures the project with CMake, and builds it with Ninja. All executables are placed in `build/bin/`.

To wipe the build directory and rebuild from scratch, pass `clean`:

```
./linux_build.sh clean
```

If the build fails because CMake cannot find Qt 6, see *Manual configuration* under Troubleshooting.

5 Running

Launch the GUI:

```
./bin/femm_gui
```

Or the headless server:

```
./bin/femm_headless
```

5.1 Setting FEMM_DIR

The language bindings (OctaveFEMM, MathFEMM, pyFEMM, JuliaFEMM) locate the FEMM executables via the `FEMM_DIR` environment variable. Set it to the `bin` directory:

```
export FEMM_DIR=/path/to/femm-qt/build/bin
```

Add this to your `~/.bashrc` to make it persistent.

5.2 Desktop Integration (Optional)

To add femm-qt to your application menu, create a `.desktop` file:

```
cat > ~/.local/share/applications/femm-qt.desktop << 'EOF'
[Desktop Entry]
Name=FEMM
Comment=Finite Element Method Magnetics
Exec=/path/to/femm-qt/build/bin/femm_gui
Type=Application
Categories=Science;Engineering;
EOF
```

6 Using with MATLAB or Octave

6.1 Octave

Install Octave and its Java support:

```
sudo apt install octave default-jdk
```

Note that in Ubuntu, the version of Octave installed by App Center does *not* have Java support, so the installation by `apt`, which does have Java support, is needed. Find the path to Java by typing:

```
readlink -f $(which java)
```

at the Bash command line. This usually returns something like:

```
/usr/lib/jvm/java-21-openjdk-amd64/bin/java
```

Edit your `~/.bashrc` file and add as the last lines:

```
export JAVA_HOME=/usr/lib/jvm/java-21-openjdk-amd64
export FEMM_DIR=/path/to/femm-qt/build/bin
```

Run Octave and confirm that Java is running by typing

```
version -java
```

at the Octave prompt. It should return something like:

```
ans = Java 21.0.10+7-Ubuntu-124.04 with Ubuntu OpenJDK 64-Bit Server VM mixed mode, sharing
```

which indicates that Java is properly installed. Then add the path to the the OctaveFEMM mfiles to Octave by typing the following at the Octave command line:

```
addpath("/path/to/femm-qt/octavefemm")
savepath
```

Now, the `openfemm` function in the `octavefemm` directory should find and launch `femm_gui` or `femm_headless` automatically.

6.2 MATLAB

Set `FEMM_DIR` before launching MATLAB, or set it within MATLAB:

```
setenv('FEMM_DIR', '/path/to/femm-qt/build/bin')
```

MATLAB on Linux uses its bundled JRE for socket communication with femm-qt, which should work without additional configuration. The path to the OctaveFEMM mfiles must also be set using the `pathtool` command.

6.3 Python

The Python interface is/will be available as pyfemm-qt. Install it via:

```
pip install pyfemm-qt
```

To install the version that comes with the source code distribution instead, change to the `pyfemm` directory in the source distribution and type:

```
pip install -e .
```

On Ubuntu 24.04 and later, pip may refuse to install into the system Python with an "externally-managed-environment" error. Since pyfemm is a single file with no dependencies, it is safe to override this:

```
pip install -e . --break-system-packages
```

7 Troubleshooting

7.1 Manual configuration

The build script just wraps the standard CMake and Ninja steps. To run them by hand:

```
mkdir build && cd build
cmake .. -G Ninja -DCMAKE_BUILD_TYPE=Release
ninja
```

CMake finds Qt 6 from the system packages automatically. If it does not, pass the Qt prefix explicitly:

```
cmake .. -G Ninja \
    -DCMAKE_BUILD_TYPE=Release \
    -DCMAKE_PREFIX_PATH=/usr/lib/x86_64-linux-gnu/cmake/Qt6
```

7.2 Qt not found

If CMake reports that it cannot find Qt6, check that the development package is installed:

```
dpkg -l | grep qt6-base-dev
```

If installed but not found, pass `-DCMAKE_PREFIX_PATH` pointing to the Qt 6 CMake config directory, typically `/usr/lib/x86_64-linux-gnu/cmake/Qt6`.

7.3 Missing OpenGL headers

On minimal server installations or VMs without a desktop environment, Qt may require OpenGL development headers:

```
sudo apt install libgl1-mesa-dev libglu1-mesa-dev
```

7.4 VirtualBox Guest Additions

If running in a VirtualBox VM, install Guest Additions for proper display scaling and shared folders:

```
sudo apt install virtualbox-guest-utils \
    virtualbox-guest-x11
```

Reboot after installation. For shared folders, add your user to the `vboxsf` group:

```
sudo usermod -aG vboxsf $USER
```

7.5 MESA / OpenGL errors in VirtualBox

VirtualBox's virtual GPU may not support the OpenGL version Qt expects, producing errors like "ZINK: failed to choose pdev" or "failed to create dri2 screen." Two fixes:

1. Enable 3D Acceleration in VirtualBox (Settings → Display → check "Enable 3D Acceleration" and move the slider to at least 128MB video memory out in Expert mode), or
2. Force Qt to use software rendering by adding

   ```
   export LIBGL_ALWAYS_SOFTWARE=1
   ```

   to `~/.bashrc`