femaxx reference card

Applicability: Commands in this reference card apply to the femaxx eigenvalue code only. The electromagnetic frontend heronion is documented elsewhere. **Iron Rule:** parameters that are used for the solution of a specific electromagnetic problem are communicated to the code at one single location only, i.e. as command line parameters when the solver is run. **Units:** the femaxx code uses the MKS system, aka. meter, kilogram, second, for all its computations. **Computing Platforms:** the femaxx code has been compiled and tested on these platforms: Linux, Cray, IBM/bluegene. Reasonably, the code will compile and execute on all Major Unix platforms.

Usage Paradigm

`femaxx` is an advanced computational electrodynamics package consisting of three different, subsequently used programs: the electromagnetic frontend heronion, the eigenvalue solver femaxx and the postprocessor femaxx_post3d. The frontend program heronion is currently under development and will be released in due course. The eigenvalue solver and the postprocessor are explained in this reference card. This document describes code usage. For detailed information on implementational questions studying the code is essential and the documentation generated by Doxygen is helpful, cf. `amas.web.psi.ch` for in-depth study.

Physics

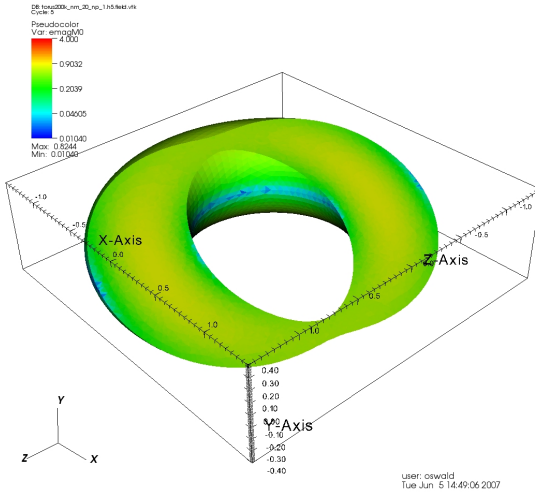
The femaxx code computes the eigenmodal fields of electromagnetic resonators by solving the electric field vector wave (*curl-curl*) equation in 3-dimensional space, in frequency domain and on a tetrahedral mesh for scalar, non-dispersive dielectric and magnetic materials.

$$\nabla \times \frac{1}{\mu_r} \nabla \times \mathbf{E} + k_0^2 \epsilon_r \mathbf{E} = 0 \tag{1}$$

At present materials do not model electromagnetic losses. The computation of a resonator's quality figure uses an *a posteriori* perturbation approach. The eigenvalue $\ell = k_0^2$ is defined in terms of electromagnetic constants and the resonance frequency f_{res}. The resonance frequency f_{res} is computed from the eigenvalue ℓ

$$f = \left(\sqrt{\frac{\ell}{\epsilon_0 \mu_0}} \right) \frac{1}{2\pi} \tag{2}$$

The eigenvalue shift σ can also be understood as a frequency shift, using Eq. (2), cf. also the `--sigma` parameter. The theory and the implementation of femaxx is explained in [1, 2].



Command Line Parameters

General

The executable of the femaxx code is `femaxx_driver`. Running the code depends on the operating system.

```
femaxx_driver
```

Nota bene: On a Linux cluster and major high performance computing (HPC) systems, it is common practice to use a batch queuing system. For specific details refer to your local system administrator. Within a batch script, command sequences using `mpirun` command are often used, e.g.

```
mpirun -np 2 <path_to_femaxx_executable>/femaxx_driver ...
```

However, this may vary wildly. On a Cray system for example a program called `yod` is used to submit jobs.

Help

To request a description of all built-in parameters of femaxx use

```
--help
```

Example: `femaxx_driver --help`

Tetrahedral mesh file

To specify the path to the file that contains the input tetrahedral volume and triangular surface mesh for the computation:

```
--mesh=<string>
```

Example: `--mesh=cavity.h5`

Eigenvalue shift

To specify the eigenvalue shift σ used in the iterative solution of the eigenvalue computation

```
--sigma=<double>
```

Example: `--sigma=2.0`

Number of computed eigenmodes

To specify the desired number of eigenmodal solution (eigenmodes) to be computed, specify:

```
--kmax=<integer>
```

Example: `--kmax=10`

Nota bene: the femaxx code uses an iterative scheme to compute eigenmodes, i.e. a variable number of iterations is required to obtain an eigenmode with given accuracy (cf. parameter `--tol` parameter). By default 5 eigenmodes are computed for which the default number of 200iterations (cf. parameter `--jitmax`) is sufficient. If more eigensolutions are required, then `--jitmax` should be increased. The increase depends on mesh quality, complexity of the problem and the number of desired modes.

Numerical precision

To specify the accuracy for the computation of the eigenmodes, write

```
--tol=<double>
```
$(1.0 \cdot 10^{-8})$

Example: `--tol=1.0e-6` which is often good enough or a good start to explore the eigenmodal space of a resonator structure.

Finite element approximation order

At present 1st and 2nd order base functions are available in femaxx. To select either of them, write

```
--order=1
```

or

```
--order=2
```

Caveat: While 1st order base functions are robust and reliable, computations using 2nd order base functions are currently being tested.

Preconditioner selection

femaxx offers a choice of different preconditioners, used in the iterative solution of the eigenmodal system. Possible selections are, where `|` denotes the logical `OR`

```
--aprec=ml | neumann | lu | diag | 2level | if
```
(ml)

Available preconditioner types:
`--aprec=neumann` : requests a Neumann type preconditioner
`--aprec=ml` : requests a multilevel type preconditioner
`--aprec=lu` : requests a LU type preconditioner
`--aprec=2level` : requests a 2level type preconditioner
`--aprec=diag` : requests a diagonal type preconditioner
`--aprec=if` : requests a if type preconditioner

Switch off log file

By default, a femaxx run produces a log file with information on all aspects of the eigenmodal computation. To switch log file generation off, write

```
--no-logfile
```

Symmetry plane configuration

Often, the geometry of electromagnetic resonators exhibits symmetries which can be exploited to reduce the computational effort considerably. A computational mesh in femaxx compliand HDF5 file also contains a list of surface triangles, each of which is attributed a surface tag whose values are either $1, 2$ or 3. The surface triangle tags, 1, 2 or 3, denote the lowest three bits in a binary number; depending on the bits being either set (1) or not set (0), the resulting binary number can express decimal values in the range of 0 through 7. If the corresponding bit of a surface triangle tag is set, then triangles with this tag are assigned the perfect electric conductor (PEC) boundary condition. If the bit is not set, then triangles with this tag are assigned the perfect magnetic conductor (PMC) boundary condition, in short:

Bit $= 0$: $\mathbf{E} \times \mathbf{n} = 0$ (PEC)
Bit $= 1$: $\mathbf{E} \cdot \mathbf{n} = 0$ (PMC)

```
--sym-plane-config=<integer>
```

Example: `--sym-plane-config=5` means that on surface triangles with tag 1 we have a PMC boundary condition, on surface triangles with tag 2 we have a PEC boundary condition, and on surface triangles with surface tag 3 we have, once again, a PMC boundary condition.

Debug mode

To switch on debug mode during code execution, use

```
--debug
```

Result file

To specify the name of the HDF5 file into which the eigenmodal calculations' results are written

```
--eigendatafile=<string>
```

Example: `--eigendatafile=box_output.h5`

Cartesian sampling

To specify an orthogonal cartesian region within which the eigenmodal solution is sampled, use

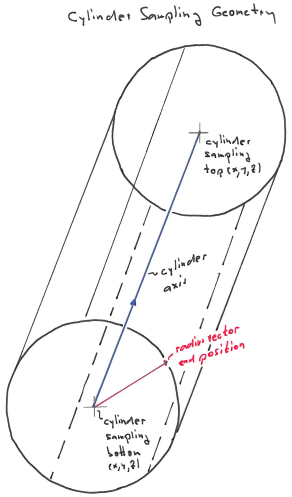
```
--cartesian-sampling-x-start = <double>
--cartesian-sampling-x-stop  = <double>
--number-samples-x-axis      = <integer>
--cartesian-sampling-y-start = <double>
--cartesian-sampling-y-stop  = <double>
--number-samples-y-axis      = <integer>
--cartesian-sampling-z-start = <double>
--cartesian-sampling-z-stop  = <double>
--number-samples-z-axis      = <integer>
```

Example:

```
--cartesian-sampling-x-start = 0.0
--cartesian-sampling-x-stop  = 1.0
--number-samples-x-axis      = 11
--cartesian-sampling-y-start = 0.5
--cartesian-sampling-y-stop  = 1.5
--number-samples-y-axis      = 21
--cartesian-sampling-z-start = -1.5
--cartesian-sampling-z-stop  = -0.5
--number-samples-z-axis      = 41
```

Cylinder sampling

Specifying an orthogonal cylinder region within which the eigenmodal solution is sampled, is similar to the Cartesian region selection. The cylindric region is defined through the axis of the sampling cylinder volume and the end position of its radius vector, w.r.t cylinder axis' start location.



The above picture shows the geometry of the sampling cylinder volume.

```
--cylinder-sampling-x-bottom                = <double>
--cylinder-sampling-x-top                   = <double>
--cylinder-sampling-y-bottom                = <double>
--cylinder-sampling-y-top                   = <double>
--cylinder-sampling-z-bottom                = <double>
--cylinder-sampling-z-top                   = <double>
--number-samples-on-cylinder-axis           = <integer>
--cylinder-radial-vector-end-position-x     = <double>
--cylinder-radial-vector-end-position-y     = <double>
--cylinder-radial-vector-end-position-z     = <double>
--number-samples-on-cylinder-radial-vector  = <integer>
--number-samples-in-cylinder-azimuthal-plane = <integer>
```

Example:

```
--cylinder-sampling-x-bottom                = 0.5
--cylinder-sampling-x-top                   = 0.5
--cylinder-sampling-y-bottom                = 0.7
--cylinder-sampling-y-top                   = 0.7
--cylinder-sampling-z-bottom                = 0.0
--cylinder-sampling-z-top                   = 1.6
--number-samples-on-cylinder-axis           = 171
--cylinder-radial-vector-end-position-x     = 1.0
--cylinder-radial-vector-end-position-y     = 1.3
--cylinder-radial-vector-end-position-z     = 0.0
--number-samples-on-cylinder-radial-vector  = 11
--number-samples-in-cylinder-azimuthal-plane = 4
```

which specifies a cylinder sampling volume where the cylinder's axis starts at $[x = 0.5, y = 0.7, z = 0.0]$ and ends at $[x = 0.5, y = 0.7, z = 1.6]$ and the end position of th radius vector is $[x = 1.0, y = 1.3, z = 0.0]$ with as many as $[171, 11, 4]$ samples on the cylinder axis, the radius vector and on the cylinder azimuthal plane, respectively.

Postprocessing

The femaxx code stores its computational results into one HDF5 compliant file. The femaxx_post3d program reads this file and converts the results into VTK legacy file format compliant files and, in particular, it creates T7 files which can be used by the Opal beam dynamics program for particle tracking.

Help

To request a description of all built-in parameters of femaxx_post3d use

```
--help
```

Example: `femaxx_post3d --help`

Input file

To specify the input file for the post processor (which is actually the the output file produced by femaxx), use

```
--input-file=<string>
```

Example: `--input-file=boxresonator_output.h5`

Output file

To specify the *base name* for the post processor's output file, i.e. the VTK legacy file format compliant files, used for visualization of results, use

```
--output-file=<string>
```

Example: `--output-file=boxresonator_post_output`

Opal T7 base file name

To specify the *base name* for the Opal compatible T7 file, i.e. the file format used for particle tracking, write

```
--opal-t7-base-file-name=<string>
```

Example: `--opal-t7-base-file-name=boxresonator_post_t7_basename`

References

[1] P. Arbenz, R. Geus, and S. Adam. Solving maxwell eigenvalue problems for accelerating cavities. *Physical Review Special Topics - Accelerators and Beams*, 4:022001–1:022001–10, 2001.

[2] R. Geus. *The Jacobi-Davidson Algorithm for solving large sparse symmetric eigenvalue problems with application to the design of accelerator cavities*. PhD thesis, Swiss Federal Institute of Technology Zurich, 2002.

History: 2008 Jun 13, benedikt oswald, corrections and post processor parameters added - 2008 Jun 11, benedikt oswald, added more parameters - 2008 Jun 10, benedikt oswald, creation and structure. Please send suggestions and corrections to `benedikt.oswald@psi.ch`. **DISCLAIMER** All data in this reference card have been collected and presented with great care. However, no responsibility is accepted for any consequences resulting from errors or omissions. Neither do we take any responsibility for using the codes. Because nobody is perfect we encourage you to report bugs, problems and suggestions to us. We will study them in depth. But, again, do not derive any responsibility from our side. You use femaxx at your very own risk. The US legal system has shown an unloved tendency to set law outside US borders. We consider this out of place and neither do we like it. We emphasize that femaxx respects the **Swiss** legal system, and that's it. On the internet type `www.psi.ch` to find us. Document last tex'ed: June 13, 2008.