Technical Report on the Software used at Tibbar Plasma Tech.

October 31, 2017

1 DEBS MHD code

1.1 Obtaining and building DEBS

- Obtain a zipped file containing all of the DEBS source code from the Tibbar URL (www.tibbartech.com/technology) and unzip it in a convenient location on your computer.

- The contents of the zip file are the original DEBS source code, the modified Fortran files for modeling the Tibbar device, a sample debs.in, a Makefile, and .pdf documentation on the code, and how to run it.

- Replace the original files by the ones with matching names that are located on the Tibbar website for the serial version (note `fft_module.f` is a new file that we separated from the main Fortran code at Tibbar)

- Type 'make' inside the /trunk directory under /plasmaPhys-DEBS/DEBS_src.

- Create a run directory in a convenient location and put debs.in in there as well as a copy of the DEBS executable or a link to the DEBS executable

1.2 Description of the new DEBS input variables to model the Tibbar transformer

- `mlock` = m mode number to use for the helical field

- `klock` = n mode number to use for the helical field

- `mtotal` = total number of modes needed to model the localized electrodes

- `ermode` = amplitude of the helical potential

- `ertime` = duration of the linear ramp time in resistive time units

- `BCtype` = switches between specifying a tangential electric with 'E' and a normal current (J_r) with 'J'

- `local_electrodes` = logical flag for implementing concentrated electrodes.

- `el_width` = electrode width expressed in degrees

- `b_r` = magnitude of the normal magnetic to inject helicity. If `local_electrodes`=1, then the B_r also becomes concentrated.

- `rkappa` = coefficient. for implementing Robin velocity BC. rkappa = 0 $\Rightarrow$ Neum rkappa $\approx$ Inf (set to 1.e15 for example) $\Rightarrow$ Dirich

2 NIMROD MHD Code

2.1 Obtaining and building NIMROD

The file labeled NIMROD_materials on www.tibbartech.com/technology contains a gunzipped tar ball of the entire source for NIMROD. A copy of NIMROD can also be obtained by contacting Professor Carl Sovinec of UW Madison at csovinec@wisc.edu. There is also a separate developer version kept at the Tech-X company, but, the Tibbar simulations were performed with the UW version.

NIMROD can be built on any platform as a serial code to be used on a single processor or as a parallel code to be used in production runs making use of hundreds and thousands of computer cores. It is usually helpful to have a local, serial version for development work and debugging. A Linux machine is the easiest platform to build NIMROD on. I recommend Ubuntu for OS. The build or compilation requires a mother makefile, make.inc, whose contents will contain platform and hardware-dependent information as well as locations of the libraries that NIMROD needs (e.g., SuperLU and METIS). More often then not, it takes some persistence and careful perusal of the error messages to get make.inc appropriately modified to build NIMROD. Luckily, NIMROD's /`make_includes` directory contains a bunch of different make.inc's that have been successfully built on various platforms. Always, use one of these as your template and modify it according to your needs. Within each separate NIMROD subdirectories,

there are additional makefiles called Makefile which are recipes for actually making the F90 object (.o) files from F90 .f files. They also include subroutine dependencies. If you add a new subroutine to NIMROD, you have to include the name of your new subroutine in the Makefile corresponding to the correct directory in which you make the modification. You will also have to specify the dependencies for your new file in the Makefile.190 Both the serial and parallel versions require SuperLU and METIS libraries for direct solves and pre-conditioning of the matrices for iterations. These are both open source and can easily be obtained. For the purposes of Tibbar simulations, we built these libraries ourselves on our 88-core local cluster. But, they are readily available to users on other platforms like the NERSC machines. Assuming you have all the libraries built correctly and have the right make.inc, all you have to do is go inside the directory where all your NIMROD directories sit and type 'make' at the top level then hit return. You will get a series of messages as the code compiles. Pay attention to these although at first they will appear overwhelming. Luckily, the build is only halted by an error, which is the last thing to get displayed.

2.2 Summary of NIMROD modifications

- In /`nimset`:
  - A series of new variables have been added to `input.f` for the Tibbar device.
  - `nimset_init.f` was modified to change the plasma resistivity profile and split it from the viscous dissipation profile.
  - `dump.f` was modified to include data for the resistive walls.
- In /`nimrod`:
  - `nimrod.f` was modified to incorporate finite-electrode shape and injection of normal magnetic field.
  - `surface_ints.f` was modified to implement tangential electric boundary conditions that impose the helical primary voltage of the Tibbar device.
  - `field_comps.f` was modified to inject a helical magnetic field at the radial boundary.
- In /`nimcore`:

- `rblock_type_mod.f` now contains code for interfacing the external Green's function solver that implements the resistive walls.

- The file DCB_BND.tar.gz contains the external Green's function solver that implements the resistive walls.