NOVA and NOVA-K code user manual

BY YOUJUN HU

Institute of Plasma Physics, Chinese Academy of Sciences
yjhu@ipp.ac.cn

1 Definition of Alfven frequency

The normalized angular frequency used in NOVA code is defined by $\varpi = \omega/\omega_A$, where ω_A is defined by

$$\omega_A \equiv \frac{V_A}{q(a)R_0}, \tag{1}$$

where $q(a)$ is the safety factor on the boundary flux surface, R_0 is the major radius of the magnetic axis, V_A is defined by

$$V_A \equiv \frac{B(0)}{\sqrt{\mu_0 \rho(0)}}, \tag{2}$$

which is the the Alfvén speed at the magnetic axis, where $B(0)$ and $\rho(0)$ is the magnetic field strength and mass density at the magnetic axis.

2 Run NOVA

To be clear, I suggest the users of NOVA organize the directory structure of the code as follows:

```
/home/username/project/nova
                        |-src
                        |-bin
                        |-scripts
                        |-work
```

And store the source codes of NOVA in "src" directory and the generated executable files in "bin" directory. The "scripts" directory can be used to contain various shell scripts. The "work" directory can be used as your working directory. In the following, I will assume that your directory structure is in this form and the five executable files of NOVA are named respectively as "goeq_ifort", "gomap_ifort", "gocont_ifort", "goin_ifort", and "gost_ifort", which are all located in the "bin" directory. Then follow the steps below:

```
cd ~/project/nova/work
cp ../src/inequ_east inequ
```

The "inequ" file provides input for the equilibrium solver. The file is formatted as a table. Every line of the file is called a "card" and the first entry of the line (entry 0) is called "card number". In the following, the No. m entry of the No. n card will be denoted by $n \times m$. The value of the entries in "inequ" file will be read in to set various parameters of NOVA code. For example, the value of the 13×4 entry of "inequ" file will be used to set the value of the parameter "itooff" in NOVA code. I will explain the meaning of most of the entries of the "inequ" file in Sec. (). For the present purpose of getting familiar with running NOVA, I only mention the entries that we need to change.

If you are running the equilibrium solver "goeq_ifort" for the first time, the value of 13×4 entry should be set to 11.0 and the value of 08×7 should be set to 0.0. Then run the equilibrium solver by using

```
../bin/goeq_ifort
```

After the program finish, set the value of 13×4 entry to +02.0 and the value of 08×7 to +1.0, and

```
mv eqb1 eqxz
```

Then run the equilibrium solver again to refine the equilibrium:

```
../bin/goeq_ifort
```

Now you can view graphics of the equilibrium by using

```
ctrans -d X11 ploteq
```

Run the mapping code:

```
../bin/gomap
```

Then you will be asked about changing the values of various parameters, type the following lines to use the default:

```
&dfun /
q
&wdat /
```

where "dfun" and "wdat" are two namelists, which contain various parameters. If you want to change, for example, the value of "igrid" parameter in the "wdat" namelist, you can specify it as the following lines

```
&dfun /
q
&wdat igrid=1/
```

The case of igrid=1 is to choose the uniform radial grid in sqrt(psi) (the default case is igrid=0, which choose the uniform grid in psi, here psi the poloidal flux). After "goemap" finish, you can view the results by using

```
ctrans -d X11 plotmap
```

To get the Alfven continuum of the equilibrium, run the continuum solver "gocont_ifort":

```
../bin/gocont_ifort
```

Then you will be asked about changing parameters, type the following line to use the default:

```
&inp /
```

or changed them using the following line:

```
&inp ntor=1, iden=1, alphar=0./
```

where alphar, prho, and arho are parameters for density profile. If iden=1 then rhom=(1.00001-alphar*psi**prho)**arho. After the program return, you can view the results by using

```
ctrans -d X11 plotcont
```

Next, we calculate the eigenmodes. We need to run two codes, "goin" and "gost_ifort". The "goin" program needs, as input, a Fortran namelist file called "modinv". An example "modinv" file can be found at the "src" directory. You can copy this file to your working directory:

```
cp ../src/modinv .
```

Then run "goin" to calculate the matrix elements of the eigenmode equations

```
../bin/goin_ifort
```

Then provide the namelist "inp" by typing:

```
&inp /
```

to use the default values, or set the values in the namelist by typing:

```
&inp ntor=3 gamma=1.66667, iden=1, alphar=0. ksound=1 /
```

When the program finish, the results can be viewed via the following command:

```
ctrans -d X11 plotidin
```

Calculate the frequency and mode structure:

```
../bin/gost_ifort
0
```

where "0" means searching for fixed boundary mode, i.e., there is a wall on the boundary and thus the amplitude of the mode is zero there. If you want to search for free boundary mode, type "1" instead.

```
zero
0.42,0.1,25,1.e-7,1,1
```

where "zero" indicates that you want to find the zero points of the determinant, the first on the second line is a guess of the root, the second is the range of the guess root outside which the iteration will not proceed, the third is the maximum number of iteration you want to perform, the fourth is the accuracy (the maximum residual permitted for the function).

When a mode is found, you can type the following line to tell the code to draw some graphics

```
1,1
```

The code will show some messages about the radial structure of the mode. The most important is the message about whether singularities are found in the mode structure. If no singularity is found, then the mode you find is probably a global Alfven eigenmodes. Note that most of the modes found by the codes are singular continuum modes, instead of global modes. So one needs patience to find a good global mode. To end the program, type the following lines:

```
finish
end
```

After this, you can view the mode structure by using the following command:

```
ctrans -d X11 plotidst
```

In the above, we set all the input parameters of NOVA in command line. It is obviouslly inconvenient to run NOVA this way. It turns out that all the above input in command lines can be provided to NOVA by writting them in a script file.

3 Run NOVA-K

The NOVA code calculates the frequency and eigen-function of the ideal MHD perturbation. The NOVA-K code can be viewed as a post-processor of NOVA, which uses the mode structure given by NOVA as input to calculate the response of energetic particles to the MHD perturbation.

Similar to the case of NOVA, I organize the directory structure of the NOVA-K as follows

```
/home/username/project/novak
                        |-src
                        |-bin
                        |-scripts
                        |-work
```

NOVA-K needs five input files, namely, (1) "eigenf" which contains the MHD perturbation (eigenfunction), (2) "mpout1" which is one of the equilibrium files generated by the mapping code of NOVA, (3) "mapdsk" which is also one of the equilibrium files generated by the mapping code, (4) "transp.dat" file, which is needed when numerical results from TRANSP code is used as the equilibrium, and (5) "NOVA_param" file, which, if absent, is generated when you run "gotae_ifort".

Now enter the working directory and copy the output file of NOVA to the working directory:

```
cd ~/project/novak/work
cp ~/project/nova/work/mapdsk .
cp ~/project/nova/work/mpout1 .
cp ~/project/nova/work/egn01w.4297E+00 eigenf
```

Then run the NOVAK post-processor "gotae_ifort" for the first time:

```
../bin/gotae
```

which will generator a "NOVA_param" file. Modify this file to set parameters, such as ntor, rmajor, and rminor. After this, run "gotae_ifort" again:

```
../bin/gotae_ifort
```

This will give the drive and damping rates contributed by various sources, such as

- energetic particles drive/damping
- thermal electron/ion Landau damping
- continuum damping
- collisional damping of trapped electrons
- radiative damping

If the energetic particles drive exceeds the sum of all the damping listed above, an instability may occur.

——To be finished.