IRI EDP

C-based modeling and simulation program: creation of a C-style wrapper to the FORTRAN IRI code (2016 version).

To use, run “make” to execute the Makefile.
To execute in Windows:> .\assess1.exe

NOTES

Any editions to the C or FORTRAN code will be picked up by ”make”.

The main principle behind the wrapper is to minimize the editing of the main code (IRI FORTRAN). However, the FORTRAN code path logic only reads the local absolute path. Thus, the C-wrapper as well as the executable is all within the same directory as the IRI codes.

Therefore, the path in the Makefile is absolute to the local IRI directory. The INDICES data are all in the same local directory. The data files are also not in the Makefile as these have not been vetted to ensure they are all accounted in the compilation.

Initially created a static library file (archive: libiri.a) to test out the fortran codes in Windows environment before creating a shared object library.

CODES

Originally, the plan was to have the FORTRAN codes stay inside their sub directory and build a PYTHON code through “f2py” located one level above. It is through “irisub.for” that f2py will wrap into.

However, 2 issues were found.

The “f2py” utility module has problems running in a WINDOWS environment. Moreover, it is only compatible with certain PYTHON versions.

The utility cannot implement the library option in WINDOWS. Various options, links, paths have been tried but the library linker will not recognize the shared file. The shared file is the static archive file and was created using the standard “make”, “ar” command. No information on integrating this format with f2py was found. Another issue encountered relates to the path be absolute. The f2py utility can only execute in an earlier PYTHON version 3.9. This version was discovered by accident. Not even v3.9.15, the last 3.9 version, would make the utility work. Although the developers used PYTHON v3.10 as the example in their website, the utility could not execute the structure used for this project.

Another issue encountered was the compiler from MSYS2, gfortran. Because the IRI code followed the strict fixed standard, research was done to determine the proper options to compile under the new compiler. In sum, a pre F95 code used with a post F95 compiler, i.e. using -lgfortran. In addition, the FORTRAN codes uses local absolute path for referencing external files. So, all the codes and data must reside in the same path.

SOLUTION

Thus, “f2py” was abandoned and a C-wrapper was used.

A C code was developed to call the external FORTRAN code, pass the required data, and use GNU plot for visuals. The first attempt was to connect with “irisub.for” but could not make the subroutine transfer the data. Due to the time already spent debugging, another attempt used the “iritest.for” module. There was a noticeable issue with “irisub.for” declaration with OARR(100). It is a 1-dimensional array but for both “iritest.for” & “iri_web” it is a 2-dimensional array, OARR(100, 1000). This will be investigated when more time can be allocated to the project.

“iritest.for”

Part of the development undertaken was to learn and understand how the subroutine worked. This also required some understanding of IRI and how it calculates the algorithm and the nomenclature. Initially, the web has a working equivalent of the algorithm and this was used to gain an understanding of the domain. (https://ccmc.gsfc.nasa.gov/modelweb/models/iri2016_vitmo.php)

One big aspect of the algorithm is the initial conditions (IC). These are key to determining the type of algorithm used throughout the system. Because of the limited knowledge of the IRI system, the web site is used to cross check the options ran by “iritest.for” (the JF switch options).

In order to utilize “iritest.for”, the C-code connects to the subroutine, manually enter the options, and continue to produce the plot. Within the C-code, several designs were made. Constants were few so preprocessor was used. Identifying the arrays was also extracted. “iritest.for” had to be edited to make the connection work. The header was added with a predefined arrays and the assignment from the internal arrays. A local .dat file (assessiri.dat) generated from iritest.for to verify the plot as well as the default output data extracted from the module (fort.7).

Lastly, the density was extracted and converted to plasma frequency. This data was used for plotting.

Options used to execute the module (entered manually) for the first plot & second plot:

> geography with latitude/longitude: geo/37.8/75.4
> time (2021, Mar 03, Universal time, 1100) : 2021/0303/UT/11
> height: 600
> x-variable selected: 1 – for height
> > min: 60, max: 600, steps: 10
>
> all other options are 0

> geography with latitude/longitude: geo/37.8/75.4
> time (2021, Mar 04, Universal time, 2300) : 2021/0304/UT/23
> height: 600
> x-variable selected: 1 – for height
> > min: 60, max: 600, steps: 10
>
> all other options are 0

LIMITATIONS

Due to the lack of general knowledge of the domain, several data were hard coded and assumed. The JF options were taken from “iritest.for” basic defaults for standard output (option 0). Other options for computations were ignored. Peak frequencies and bottomside attributes were ignored. Most importantly, iri_sub subroutine seems to have different interface compared to “irisub.for” module. To resolve all these issues, more time would be required to gain an understanding of the main aspects of the code.

ENVIRONMENT

MS VSCode

MSYS2