Introduction to Fortran

FORTRAN is a general-purpose programming language originally developed in the 1950s. It has experienced multiple stages of evolution since its first release that have resulted in multiple standards of the language, with the most-recent “modern standard” being *Fortran 2018*.

Despite not maintaining the same popularity of the past, it is still used heavily in specialized fields in applied sciences and engineering for its scientific numerical computing ability. Fortran’s ability to perform calculations with speed make it helpful when analyzing topics including fluid dynamics, weather forecasts, computational chemistry, and other problem that demand high computational performance.

In addition to being a powerful tool, over 65 years of development and collaboration amongst users has resulted in an extensive network of libraries and resources. Many of the programs written in older versions of Fortran are still compatible with modern versions of the language, making it possible to integrate legacy versions of the language or programs in current workflows and analyses.

Initial Setup

There are several different compliers to choose from when it comes to getting started with Fortran. Each options comes with its own costs, benefits, or level of compliance to different Fortran Standards (e.g. open-source software versus commercial, Fortran 95 versus Fortran 2018 standard, etc.).

For casual use and getting started with the basics of the language, we will use the GNU Fortran Complier (GFortran).

In addition to needing a complier for Fortran, you also will need a text-editor/IDE. There are many options to choose from depending on personal preferences, I will be using Visual Studio Code (VS Code) as my preferred environment.

If using a version of Linux OS, GFortran might already be installed. To check, you can run the shell command

```
which gfortran
```

If not already installed, you can run the command below to install the program

```
sudo apt install gfortran
```

Otherwise, running the following command will show what version(s) are currently installed.

```
gfortran --version
```

The following series of commands will install the latest versions of GFortran on older versions of Linux

After completing these steps for the initial setup, next we configure our preferred text

```
sudo add-apt-repository ppa:ubuntu-toolchain-r/test
sudo apt update
sudo apt install gfortran-10
```

editor to support Fortran. Since we are using VS Code, this can be done by searching for 'Modern Fortran' from the extensions menu option (ctrl + shift + x). Depending on the local device's OS and settings configurations, the default terminal may need to be changed before proceeding.

Creating a Program

Consider the classic "Hello, World!" program.

First, we need to create a folder to contain our working project. For this example, I created a folder titled "fortran_testing".

Next, we can begin composing the program. The Fortran code for this example will be as follows:

```
program hello
    ! This is a comment line; it is ignored by the compiler
    print *, 'Hello, World!'
  end program hello
```

Once the program is written, we can save it in the folder we just created (make sure to make note of the file type -- *.f90 is the standard extension for Modern Fortran).

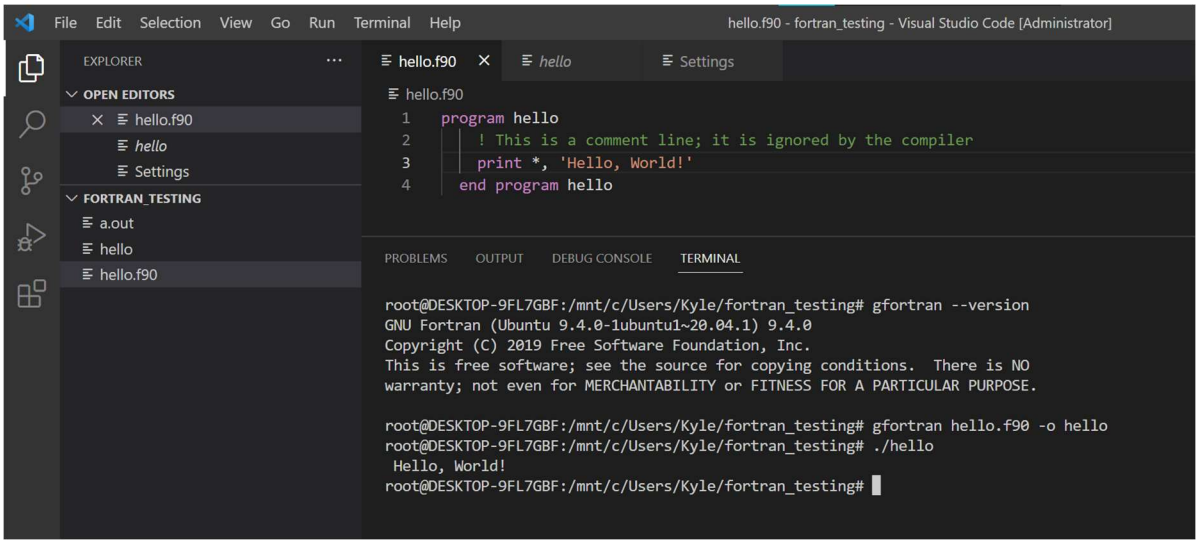
Upon saving, open the shell and run the command below to compile.

```
gfortran hello.f90 -o hello
```

Then you can run the final line of code shown below (line 1) to run the compiled program and see the results (line 2) that are returned

```
$> ./hello
Hello, World!
```

The screenshot below shows an overview of what this process looks like from VS Code.



Creating Another Program

Now that we've covered the basics, we can continue to explore the language and the myriad of purposes that Fortran can serve.

One way to harness Fortran's power is as a graphing calculator. One of ways we can create this plot is with `gnuplot`, a command-line graphing interface that can be downloaded to Linux with commands shown below and follow directions if prompted to download additional required libraries or source-codes.

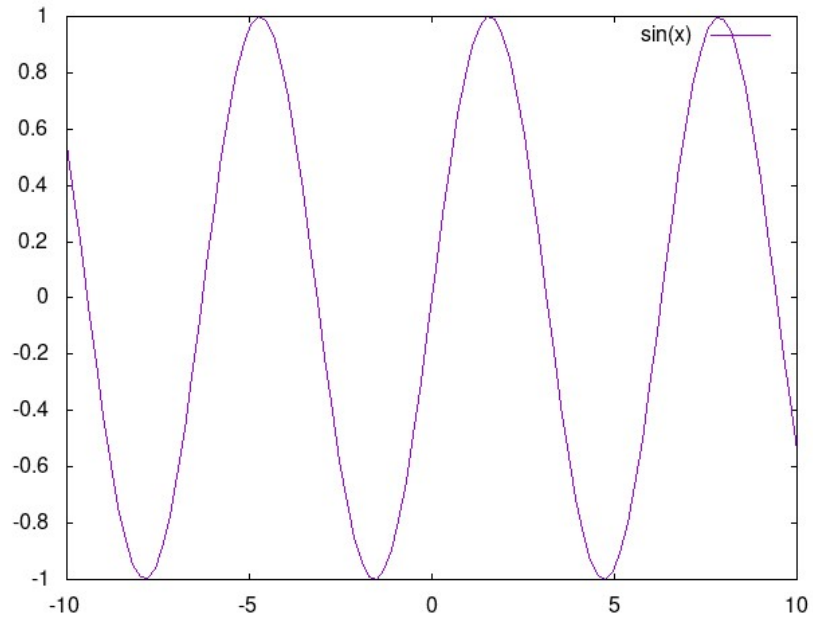
```
sudo apt-get install gnuplot-x11
```

Gnuplot can be used independently through the Linux command line interface and it can also interact with Fortran programs. Before a graph can be created, certain configurations are needed to `gnuplot` from the terminal. In the example below, we check to ensure the directory is set the same for our Fortran project folder, specify the terminal and path for output.

```
gnuplot> pwd
/mnt/c/Users/Kyle/fortran_testing2
gnuplot> set terminal png
Terminal type is now 'png'
Options are 'nocrop enhanced size 640,480 font "arial,12.0" '
gnuplot> set output "./output.png"
```

If we wanted to create a graph to represent a trend that can be described by the function f(x) = sin(x), we would run the command below with our specified function and a .png file will be created in our specified output path.

```
gnuplot> plot sin(x)
```



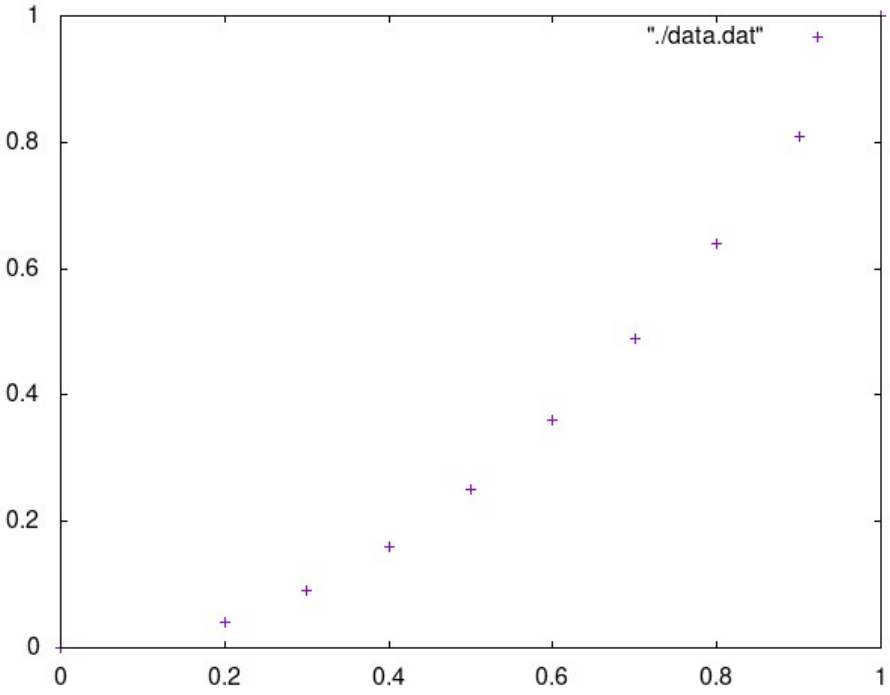
Suppose we create the following program in Fortran expressed by the code on the right that represents a given set of data (or in this case a simulation).

In addition to defining the parameters of the equation, the program also specifies the data output file path and defines graphing parameters in a file set that is called `data_plot.plt`.

Once satisfied with the program, we compile and then executre the program to provide us with a file in our project foldered titled `data.dat`.

```
PROGRAM testing2
    INTEGER :: i,n=10
    REAL :: x(10),y(10)

    x(1)=0.0
    y(1)=0.0

    DO i=2,n
     x(i)=0.1*i
     y(i)=x(i)*x(i)
    ENDDO

    OPEN(UNIT=48,FILE='data.dat')
    DO i=1,n
     WRITE(48,*) x(i),y(i)
    ENDDO
    CLOSE(48)

    CALL SYSTEM('gnuplot -p data_plot.plt')

    END PROGRAM testing2
```

Once we have this file, we can return to the terminal to create a plot of our simulated data as shown by the graph below (make sure to redefine the output if we do not wish to override the previous graph).



Sources

http://www.gnuplot.info/

https://cyber.dabamos.de/programming/modernfortran/gnuplot.html#process

https://fortran-lang.org/en/

https://fortranwiki.org/fortran/show/HomePage