
Chapter 1: Introduction

Welcome to “Everything I wanted to know about Fortran, but was afraid to ask.” This seminar operates as an introductory crash course in Fortran for applied mathematicians. Although Fortran dates back to the 1950s and many newer languages have proved more useful for a broader range of projects, Fortran is still widely used by the scientific computing community. The advantages of Fortran are especially prevalent in vector operations.

The name Fortran comes from FORMula TRANslation. That is, the language was originally developed for easy implementation of mathematical formulas, vector and matrix operations in particular. The flexibility of Fortran to natively handle arrays makes your life much easier when coding basic routines, like matrix vector products, to more advanced routines like linear solvers or conjugate gradient.

This seminar will not include all aspects of Fortran, but will be enough to get you up and running, solving problems, and coding with organization. The seminar is structured to learn Fortran through example, with commentary along the way. All materials for the course are available at <http://www.math.fsu.edu/~jmckenna/fortran>.

Software

For Linux users, all the software that you need likely comes prepackaged for your distribution, and you can install as you would normally with a package manager. For Windows users, “MinGW” and “Cygwin” are systems that provide open source software and functionality similar to Linux distributions on Windows machines. For Mac users, “XCode” in addition to the open source software package manager “Homebrew” provide functionality similar to Linux. Alternatively, on either Windows or Mac, you can emulate a Linux operating system with a virtual machine (<https://www.virtualbox.org/>).

Required Software

In order to create programs, you will need a Fortran *compiler*. The job of the compiler is to translate Fortran *source code*, i.e. the part that you write, to machine code that can be executed on your computer. There are a few choices for Fortran compilers, but for the purposes of this course we will want to use the one freely available from the GNU software movement, `gfortran`. This compiler comes prepackaged for Linux systems and is easily installed. The only other software that you will need to start coding is a text editor. There are many text editors. The one you choose to use is up to you. Several programmers prefer “Emacs” or “Vim” since they are free, extensible and have a lot of features geared toward program development.

Optional Software

Before coding scientific computing programs in low-level procedural languages such as Fortran or C/C++, it is useful to test your proposed algorithms in a higher-level interactive environment. These allow you to define variables and perform computations with them without having to recompile everything from source any time you want to alter your code. There are a few choices for these.

- MATLAB - A proprietary high-level language interactive environment available from MathWorks. The license is expensive, but the program is available in the math grad basement.
- Octave - A free alternative to MATLAB provided by the GNU software movement.

After generating data from your code, you need a convenient way to present it. This will include making plots and tables and collecting these into a typed report. For this, I recommend

- `gnuplot` - A free plotting program from the GNU free software movement. With this program, you can write scripts that generate plots. It is well documented on the web. It comes prepackaged for a lot of Linux distributions and is available for Windows and Mac.
- \LaTeX - The *de facto* standard for typesetting scientific documents. It comes prepackaged for a lot of Linux distributions (as the package `texlive`). It is also available for Windows and Mac. For Windows or Mac users, I suggest installing “MikTeX” or “MacTeX”, respectively, to get started with \LaTeX .

Fortran Standards

Since Fortran was invented in the 1950s, it has gone through a number of revisions (i.e. FORTRAN 66, 77 and Fortran 90, 95, 2003, 2008) that modified the syntax and changed features of the language. With each revision, official standards that precisely specify the syntax and behavior are released by the Fortran standards committee (<http://www.j3-fortran.org/>). Unfortunately, these are not freely available. If you desire, you can purchase the official Fortran standard through the International Organization of Standardization: ISO/IEC 1539-1:2010.

There are a number of other resources, although less official, that you can use for free to find just about everything you’ll need to know about Fortran. These are

- The `gfortran` compiler documentation - The `gfortran` compiler fully supports Fortran 95 and partially supports Fortran 2003 and 2008. The documentation specifies exactly which functions and routines intrinsic to Fortran are supported. This can be found online.
- fortran90.org - An unofficial collection of Fortran tips and information.