Some Exercises

Settings

For those of you who can access the fusion computers (fusc01..fusc10), use the ifort compiler.
Otherwise download the executable of gfortran http://gcc.gnu.org/wiki/GFortranBinaries .
A window installer is also provided by http://www.equation.com/servlet/equation.cmd?fa=fortran .
The gfortran version to download is version 4.8.x .
Write the Makefile and the program as in the slide named “Un esempio”.

The most famous Fortran bug

Just to convince yourself that it is really a bug, Compile, link and execute the most famous Fortran bug.

Some exercises on the kinds

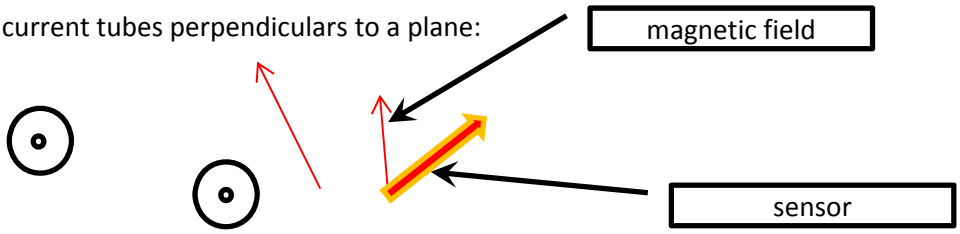
1. Print the **integer** kinds which offer a range R for R ranging from 1 to 20 decimal digits.
2. Print the **real** kinds which offer a precision P for P ranging from 1 to 20 digits (use **selected_real_kind**, the range is optional in this fubction).
3. Declare a **real** variable with a precision of 22 digits. Which is the full precision of this variable, and the range?
4. which is the value of tiny, huge and epsilon of that kind.

The Fibonacci number

Write a recursive function which calculates the Fibonacci numbers.

The current tubes

There are some current tubes perpendiculars to a plane:



The tubes are characterized by a current flowing through them and by their position on the plane.
There are also some sensors that measure (like the classical MHD tokamak coils) the magnetic field in the plane along a given direction.
Define a “user defined type” for a single tube. That type will have as subtype, the position in the plane, the current and the width of the tube (so the magnetic field wouldn’t be singular). And a type for the sensor, with the position on the plane and the direction along which it measures the magnetic field.
Declare them to be of double precision in a portable way using: kind(1.0d0).
Put these definitions in a module. The module can be placed in the same file as the main program, before declaring the main program, this is useful if you have small modules used only in that program.
Put also a function inside the module which calculate the magnetic measured by a sensor. The function should take as input (declared intent in) a tube and a sensor and should calculate the output of the sensor.
Declare also inside the module a double precision constant constant with the value of greek-pi.
Declare also a couple of subroutines that can set some characteristic of the current tubes and the sensors. Write it using optional argument, and call it in the main program using keyword arguments.