

Informática

Clase 7: Funciones y Subrutinas II Representación gráfica de resultados

Mario Merino Martínez
mario.merino@upm.es

Escuela de Ingeniería Aeronáutica y del Espacio
Universidad Politécnica de Madrid

13 de diciembre de 2011

Funciones como argumento

- Hasta ahora hemos visto **cómo pasar variables** (números, cadenas de texto...) **a un subprograma**, para que los **utilice como datos (inputs)** o **escriba sobre ellos (outputs)**.
- En ocasiones, **queremos pasar una función como argumento**.
E.g., tenemos una subrutina que **resuelve ecuaciones**: Queremos que sea capaz de resolver **cualquier ecuación del tipo $f(x) = 0$** , y para ello *le pasamos funciones de una variable como argumento*.

Funciones como argumento II

El modo de hacerlo es muy natural. Simplemente,

- En el **subprograma**, el argumento-función ha de ser **declarado como external**:

```
subroutine solver(f) ! Sólo f, NO f(x)
implicit none
real*8, external :: f ! Sólo f, NO f(x)
! ...
end subroutine solver
```

- El **programa principal** **llama al subprograma** con la **función** que queramos pasar:

```
program main
implicit none
real*8, external :: fun ! External también en el principal
! ...
call solver(fun) ! llama a solver con fun como argumento
! ...
end program main
```

Arrays como argumento

- Los **arrays** se pasan **como argumento** igual que cualquier otra variable.
 - Hay que **declararlos** con **mismo tipo y dimensiones** en el **programa principal** y en el **subprograma**
- Podemos pasar **subarrays** (“**rodajas**” y **secciones**) usando **notación de slicing en la llamada**.

E.g.: `call sub_prueba(A(1:3,2:5))`

- En tal caso, el **subprograma** tiene que declarar el **argumento con el tipo y dimensiones de la rodaja** que le vamos a pasar.
- **Dentro del subprograma**, la rodaja enviada funciona **como un array habitual** (i.e., el **subprograma desconoce** si lo que le hemos pasado es **un trozo de una matriz**, o **una matriz en sí misma**).

Arrays como argumento II

- Realmente, cuando “pasamos argumentos”, estamos indicando al subprograma las ***direcciones de memoria*** donde puede leer y escribir. Los **arrays** son **direcciones de memoria consecutivas**.
- **Si no coinciden las dimensiones**, el **subprograma** tomará las direcciones de memoria **que le hagan falta a partir del primer elemento** del array o subarray. El comportamiento es **complicado** y los **errores** de programación **muy difíciles de detectar**. Así que os aconsejo **no utilizar esto**.

“declarad siempre los arrays (o subarrays) con mismo tipo y dimensiones en ambos lados, y usad “:” siempre que haga falta”

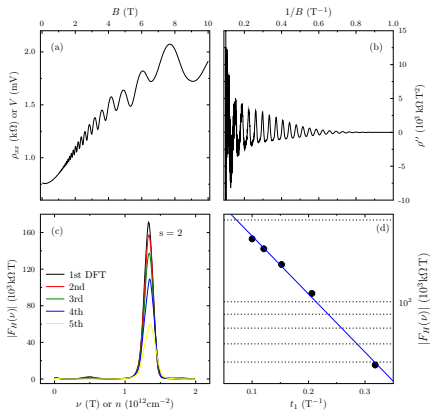
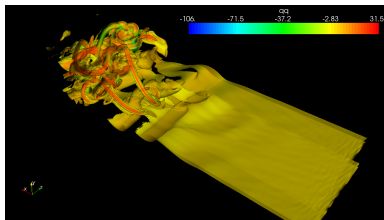
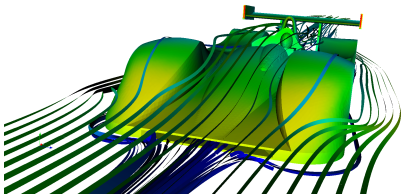
*Imprescindibles

- Es posible pasar **funciones** como **argumento**; no hay más que declararlas como **external** a **ambos** lados.
- Recordad que **un trozo de array** (subarray) **es también un array**. Podemos **pasar rodajas de arrays del programa principal** a los subprogramas con la **notación “:”**.
- **Aseguraros** bien de que los **argumentos** que pasáis **coinciden en tipo y dimensiones**. **Inclusive** cuando pasáis **trozos de arrays**: el trozo ha de ser del tamaño exacto que lo que espera recibir el subprograma.
- **Si no**, el **subprograma** utiliza las **direcciones de memoria subsiguientes a la del primer elemento (desaconsejable)**.

Representación gráfica de resultados

- La **presentación científica de resultados** es una **parte importante** de nuestro trabajo.
- Normalmente, se utilizan **programas especializados** como **gnuplot** (gratuito), **paraview** (gratuito), **tecplot** (comercial), etc.
- Estos programas producen **gráficas de alta calidad**, con **múltiples opciones** de presentación. *Os recomiendo que no os contentéis con menos.*
- Hay **herramientas dedicadas para cada tipo de gráfica/datos**.
- En este curso, utilizaremos **Excel** (2007 ó 2010) para hacer gráficas muy elementales.

Ejemplos



Gráficas en Excel. Pasos a seguir

- 1 **Vuestro programa** ha de escribir los resultados a un fichero “.txt”:
 - en **columnas**
 - con **formato $Fn.d$ ó $En.d$**
- 2 **Cargar datos en Excel:** **Importar datos|desde texto.**
Tipo: **Ancho fijo**
 - **Cuidado: punto/coma como separador decimal.**
- 3 **Crear el gráfico:** **Insertar|Dispersión|Dispersión con líneas RECTAS**
- 4 **Añadir línea:** **Seleccionar datos|Añadir serie.** Aquí se seleccionan los datos para **abscisas**, **ordenadas**, y **nombre** de la línea.
- 5 Dar un **formato adecuado** al gráfico

Formato de gráficos en Excel

La **presentación es importante**, *y será evaluada*:

- 1 Dentro de “Diseños de gráfico”, el **diseño 1** es adecuado.
- 2 Con “**Dar formato al eje**” (**botón derecho sobre cada eje**), ponemos los números con **dos decimales sólo**, y situamos los **ejes fuera de la curva** en lugar de encima de ella.
- 3 **Quitamos marcas** (rejilla) **horizontales**, **seleccionándolas y pulsando suprimir**
- 4 Opcionalmente, **ponemos títulos** (y unidades, si procede) a los ejes y a la gráfica.
- 5 Opcionalmente, con “**Dar formato a la serie**” (**botón derecho sobre la línea**), se puede cambiar color, grosor, tipo de línea...