Coloquio Programación en FORTRAN

Día 3 (clases 5 y 6).

- Entrada y salida de datos.
- Trabajo con bibliotecas.
- Tarea 1.

Entrada y salida básica

read (núm_unidad, núm_formato) lista_de_variables

write(núm_unidad, núm_formato) lista_de_variables

El número de unidad se puede referir a la salida estándar, entrada estándar o a un archivo.

Entrada estándar (teclado): * ó 5

Salida estándar (pantalla): * ó 6

El número de formato se refiere a una etiqueta para la sentencia FORMAT .

Ejemplos:

```
  write(6,*) 'Fermi level (eV):'
  read(5,*) ef
  open(10,file=input, status='old')
  open(11,file=output, status='unknown')
.......
     read(10,*) (k(j), j=1,3)
     write(11,9020) (k(j), j=1,3)
9020 format (14x, 3f7.4)  !
  close(10)
  close(11)
```

Sintaxis de la sentencia OPEN

OPEN ([UNIT=]io-unit [, FILE=name] [, ERR=label] [, IOSTAT=i-var], slist)

Modificadores (slist)

- ACCESS = 'SEQUENTIAL', 'DIRECT' .
- ACTION = 'READWRITE', 'READ', 'WRITE' .
- FORM = 'FORMATTED', 'UNFORMATTED' .
- POSITION = 'ASIS', 'REWIND', 'APPEND' .
- STATUS = 'UNKNOWN' , 'OLD', 'NEW', 'SCRATCH', 'REPLACE'

Especificadores adicionales

read ([UNIT=]u, [FMT=]fmt, IOSTAT=ios, ERR=err, END=s)

write([UNIT=]u, [FMT=]fmt, IOSTAT=ios, ERR=err, END=s)

Ejemplo:

```
   open(UNIT=10,FILE='datos.dat')
   do
     read(10,END=110) c(i)
     i=i+1
   end do
110 continue
   close(10)
```

FORMAT

La secuencia

```
write(6,9020) (k(j), j=1,3)
9020 format (14x, 3f7.4)
```

hace el mismo efecto que

```
write(6,'(14x, 3f7.4)') (k(j), j=1,3)
```

La primera forma es útil cuando hay múltiples write que usan el formato 9020.

Escribe 14 espacios en blanco, seguido de 3 reales que ocupan cada uno 7 caracteres, incluyendo 4 cifras decimales y el punto decimal.

Descriptores de FORMATo

Descriptor	Use
rIw	Integer data
rFw.d	Real data in decimal notation. Ej: 1F16.6
rEw.d	Real data in scientific notation. Ej: 1E16.6
Aw	Character data
'x...x'	Character strings
nX	Horizontal spacing (skip spaces)
/	Vertical spacing (skip lines)
Tc	Tab

Where:
w = positive integer specifying FIELD WIDTH
r = positive integer REPEAT COUNT
d = non-negative integer specifying number of digits to display to right of decimal.
x = any character
n = positive integer specifying number of columns
c = positive integer representing column number

Ejercicio

directorio EntradaSalida

1) Escribir un programa corto que haga lo siguiente

- Pregunte por teclado el nombre de un archivo y la dimensión de una matriz cuadrada.
- Cree una matriz de números reales con una regla cualquiera y escríbala en el archivo. Puede ser util la función TRIM.

2) Haga otro programa que lea la matriz.

Explore diferencias entre FORM='FORMATTED' y 'UNFORMATTED'.

Trabajo con bibliotecas

Dados n subprogramas (subroutinas y/o funciones): file1.f, …, filen.f, se pueden combinar en una biblioteca (library)

Use the f77 command to compile each routine to create its object files.

gfortran -c file1.f file2.f …. filen.f
ar crv libsubsmy.a file1.o … filen.o

Luego, los subprogramas presentes en libsubsmy.a se pueden enlazar en un programa X

gfortran -o programX -Ldirectorio_biblioteca -lsubsmy.a main.o

Bibliotecas estándares

- Existen muchas bibliotecas disponibles. Se instalan en directorios estandares como /lib, /usr/lib, /usr/local/lib, /lib32, /lib64 . Ver archivo `/etc/ld.so.conf`
- Populares: libc, libmpi, libblas, libatlas, liblapack,libpython.
- Dos tipos: estáticas (lib*.a) y dinámicas (lib*.so). En Windows se llaman DLL.

BLAS (Basic Linear Algebra Subprograms)

- Nivel 1. Operaciones con vectores
- Nivel 2: Operaciones matriz-vector.
- Nivel 3: Operaciones matriz-matriz.

Si estan instaladas, basta agregar -lblas a la línea de compilación.

Hay versiones optimizadas para cada tipo de CPU (Intel MKL, AMD ACML, IBM ESSL, NVIDIA cuBLAS) y una version automágicamente ajustada (ATLAS).También hay versiones paralelas.

Acerca de las BLAS en Ubuntu

- Ver ejemplo de matrices.f90 compilado en mi PC, o en Mac. Corre en paralelo con 2 hilos. ¿Por qué?
- Las libblas.so.3gf no es realmente BLAS

```
emenendez@habanados:~$ update-alternatives --config libblas.so.3gf
There are 2 choices for the alternative libblas.so.3gf
(providing /usr/lib/libblas.so.3gf).

  Selection    Path                                       Priority
 Status
------------------------------------------------------------
* 0            /usr/lib/atlas-base/atlas/libblas.so.3gf   35
 auto mode
  1            /usr/lib/atlas-base/atlas/libblas.so.3gf   35
 manual mode
  2            /usr/lib/libblas/libblas.so.3gf            10
 manual mode
```

Si se instala openblas, aparecería como 3ra opcion.
http://www.stat.cmu.edu/~nmv/2013/07/09/for-faster-r-use-openblas-instead-better-than-atlas-trivial-to-switch-to-on-ubuntu/

LAPACK (Linear Algebra Package)

Hacen operaciones más complejas de matrices, tales como inversión, diagonalización.

Es libre, bajo licencia BSD-new.

Existen versiones optimizadas de los fabricantes de CPU: MKL, ACML, ESSL, etc.

Generalmente conviene usarlas para las operaciones con matrices o vectores. Están programadas de forma eficiente.

GNU Scientific Library (GSL): para C and C++

http://www.gnu.org/software/gsl/

Complex Numbers Roots of Polynomials Special Functions Vectors and Matrices Permutations Sorting BLAS Support Linear Algebra Eigensystems Fast Fourier Transforms Quadrature Random Numbers Quasi-Random Sequences Random Distributions Statistics Histograms N-Tuples Monte Carlo Integration Simulated Annealing Differential Equations Interpolation Numerical Differentiation Chebyshev Approximation Series Acceleration Discrete Hankel Transforms Root-Finding Minimization Least-Squares Fitting

FGSL: A Fortran interface to the GNU Scientific Library
http://www.lrz.de/services/software/mathematik/gsl/fortran/

Ejercicio

directorio Bibliotecas

- Hacer un programa que cree dos matrices de números reales de doble precisión, de dimensión NxN, según cierta regla y haga su multiplicación. Estudiar la dependencia del tiempo de ejecución del programa con la dimension de la matriz.
- Hágalo de tres maneras: 1) programando la multiplicación, 2) con el comando MATMUL, 3) utilizando DGEMM de BLAS.
- Si están instaladas, use también ATLAS y MKL.

Acerca de las BLAS

- Ver ejemplo de matrices.f90 compilado en mi PC, o en Mac. Corre en paralelo con 2 hilos. ¿Por qué?
- Las libblas.so.3gf no es realmente BLAS

```
emenendez@habanados:~$ update-alternatives --config libblas.so.3gf
There are 2 choices for the alternative libblas.so.3gf
(providing /usr/lib/libblas.so.3gf).

  Selection    Path                                       Priority
 Status
------------------------------------------------------------
* 0            /usr/lib/atlas-base/atlas/libblas.so.3gf   35
 auto mode
  1            /usr/lib/atlas-base/atlas/libblas.so.3gf   35
 manual mode
  2            /usr/lib/libblas/libblas.so.3gf            10
 manual mode
```

Si se instala openblas, aparecería como 3ra opcion.
http://www.stat.cmu.edu/~nmv/2013/07/09/for-faster-r-use-openblas-instead-better-than-atlas-trivial-to-switch-to-on-ubuntu/

Tarea 1

Considere el desarrollo en serie

$$\sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \ldots = \sum_{n=0}^{\infty} (-1)^n \frac{x^{2n+1}}{(2n+1)!}$$

Haga un programa que calcule eficientemente sin(x), obviamente sin utilizar la funcion intrínseca. Investigue hasta qué valores positivos de x se puede usar la serie para calcular la función seno de forma práctica. Utilice las propiedades de periodicidad y paridad para optimizar el programa calcular el seno para todo *x* real. Optimice el cálculo de los factoriales en la sumatoria. Compare los tiempos de cálculo de su programa con la función intrínseca SIN(X).