

Métodos Numéricos

Guía 1

Marzo de 2022

Nota: La información, en la computadora que use, es guardada de forma permanente, en lo que se suele llamar disco rígido. Es su responsabilidad mantener un sistema ordenado de trabajo, donde la información está guardada en un árbol natural de organización. Dependiendo del sistema operativo, cuando se inicia una ventana de trabajo o terminal, se puede estar en la ubicación inicial, que genéricamente llamaremos *home*, que por ejemplo podría ser:

`/home/jaimito`

si se está en linux (ubuntu). En el lugar donde en el ejemplo aparece `jaimito` debería aparecer su nombre de usuario. De aquí en adelante asumimos que trabaja con un sistema operativo linux, como por ejemplo *ubuntu*.

Ejercicios de linux

Problema 1: En una terminal, estando en su directorio inicial de trabajo (*home*), cree un directorio nuevo llamado `metodos numericos`; a propósito sin acento. Cambie de directorio desde el inicial al directorio `metodos numericos`. En ese directorio cree otro directorio llamado `guias`; a propósito sin acento. Descargue desde el navegador el archivo de esta guía y mueva el mismo al nuevo directorio. Cambie el nombre del archivo a *guia1.pdf* y copie el archivo a otro con nombre *Copia.pdf*. Luego, liste el contenido del directorio y verifique los nombres y el tamaño y fecha de los mismos. Averigüe qué espacio ocupa el directorio usando el comando `du -h`. Finalmente, borre el archivo *Copia.pdf*. Para ver en qué lugar del árbol de directorio está, ejecute el comando `pwd`.

Problema 2:

- Asegúrese de que se encuentra en el directorio `metodos numericos` (`pwd`).
- Cree 2 carpetas: `ejercicios1`, `ejercicios2` (utilizando `mkdir`).
- Acceda a `ejercicios1` (utilizando `cd`).
- Allí, cree un archivo de texto llamado `file1.dat` con la frase `"Hola mundo"` (utilizando `kate`). Asegúrese de guardarlo.
- Sin moverse del directorio `ejercicios1`, haga una copia del archivo `file1.dat` dentro de `ejercicios2`, y llámela `file1_copia.dat` (utilizando `cp`).
- Acceda al directorio `ejercicios2` (con `cd`) y verifique la existencia del archivo `file1_copia.dat` en ese directorio (con `ls`).
- Muestre el contenido del archivo usando el comando `cat`.
- Cambie el nombre del archivo `file1_copia.dat` a `file1_movido.dat`, utilizando `mv`.
- Ubíquese en el directorio `metodos numericos`, haciendo `cd ..`. Haga `ls` (para ver el contenido) y `pwd`.
- Borre el directorio `ejercicios2` con todo su contenido, haciendo `rm -Rf ejercicios2`.

Ejercicios de gnuplot

Problema 3: Utilice el programa *gnuplot* para graficar la función $f(x) = \sin(2\pi x)$, para $x \in [0 : 5]$. Ajuste la cantidad de puntos que usa gnuplot para graficar (*samples*) a 500 para mejorar la calidad del gráfico. Dé nombre a los ejes, título al gráfico y defina la leyenda para que aparezca $f(x)$.

Problema 4: Abra una terminal y realice las siguientes acciones:

a) Vaya al directorio:

```
/home/jaimito/metodos_numericos/
```

Donde, en lugar de “jaimito”, debe ser el nombre de su usuario. Allí cree el directorio

```
guia1
```

b) En ese nuevo directorio ejecute el editor *kate* liberando la línea de comandos (utilice el símbolo `&`).

c) En el editor de texto ingrese los siguientes datos en tres columnas, separados por dos espacios:

0.0	1.0	1.2
1.0	1.5	1.6
2.0	2.0	2.0
3.0	2.5	2.4

Grabe esta tabla en el archivo *datos.dat*. No cierre el editor.

d) Ahora utilice *gnuplot* para graficar la 2da y la 3era columna versus la primera. Grafique usando sólo puntos. En el mismo gráfico incluya las funciones $y = 0.5x + 1$ y $y = 0.4x + 1.2$. Nombre al eje horizontal x y al vertical y . Habilite la grilla. Titule el gráfico *Funciones lineales*. Cree archivos de salida en formato png, pdf y postscript color y blanco y negro (nómbrelos *p4.png*, *p4.pdf*, *p4_color.ps* y *p4_byn.ps* respectivamente). Abra otra terminal y con la aplicación *okular* visualice los archivos de salida, a medida que los va generando). Luego, experimente graficar cambiando el tamaño de los puntos y el grosor de las líneas, etc.

e) Liste los archivos del directorio mostrando su tamaño y fecha. Averigüe qué espacio ocupan los archivos. Finalmente, cierre la ventana del editor. Borre los archivos postscript y cierre la terminal.

Ejercicios de Fortran

Problema 5: Evaluar con un programa FORTRAN 90 las siguientes operaciones matemáticas. Realice todas las operaciones en forma secuencial en un mismo programa. Imprima por pantalla cada operación y su resultado.

a) $A = 5 / 2 + 20 / 6$

b) $B = 4 * 6 / 2 - 15 / 2$

c) $C = 5 * 15 / 2 / (4 - 2)$

d) $D = 1 + 1/4$

e) $E = 1. + 1/4$

f) $F = 1 + 1./4$

g) $G = 1. + 1./4.$

Problema 6: Escriba un programa FORTRAN 90 que pida dos números reales e imprima en la pantalla el mayor de ellos. El programa debe indicar si los números son iguales.

Problema 7: Escriba un programa FORTRAN 90 que pida un número entero y determine si es múltiplo de 2 y de 5.

Problema 8: Escriba una programa que ingrese los coeficientes A , B y C de un polinomio real de segundo grado ($Ax^2 + Bx + C$), calcule e imprima en pantalla las dos raíces del polinomio en formato complejo $a + ib$, sin utilizar algebra compleja.

Problema 9: Implementar un programa Fortran para evaluar la suma (en precisión simple)

$$\sum_{n=1}^{10\,000\,000} \frac{1}{n}$$

primero, en el orden usual, y luego, en el orden opuesto. Explique las diferencias obtenidas e indique cuál es más preciso y su justificación.

Problema 10: Escriba un programa para calcular la posición y la velocidad en función del tiempo, para un problema de tiro oblicuo. Debe preguntar el ángulo (en grados) y la velocidad inicial (en m/seg.), asumiendo que el proyectil parte del origen. Elija el incremento temporal (Δt) de manera que la gráfica tenga 600 puntos y abarque el intervalo entre el disparo y el instante en que el proyectil vuelve a tener altura 0. Utilice *funciones* para calcular la posición y la velocidad. Escriba la salida del programa en un archivo de texto, con 5 columnas (t , $x(t)$, $y(t)$, $v_x(t)$, $v_y(t)$). La primera línea debe comenzar con #, e incluir la descripción de los datos de cada columna. Los datos en la tabla deben tener 6 cifras significativas y estar escritos en notación exponencial. Grafique $x(t)$, $y(t)$ y $v_y(t)$ en función de t , y la trayectoria del proyectil, utilizando *gnuplot*.

Problema 11: Escriba un programa, que utilizando una subrutina, multiplique un vector de N elementos, por una matriz de $N \times N$. El programa debe preguntar el valor de N y luego definir los arreglos, y darle valores iniciales tal que, la matriz sea triangular superior, con todos sus elementos igual a 1, excepto los de la diagonal que toman valor 3, el vector tendrá todos sus elementos pares igual a 2, y los impares igual a 3. No utilice `D0` para las inicializar el vector, ni `D0` anidados para inicializar la matriz.

Problema 12: Escriba un programa que permita convertir números naturales con base 10 a la base $b \leq 16$. El programa debe pedir como entrada b y el número natural a convertir (que debe estar en base 10). Utilice `SELECT CASE` para asignar los dígitos (del resultado en base b). Utilice `CASE DEFAULT` (para asignar los dígitos del 0 al 9).

Ejercicios Complementarios

Problema 13: Efectúe con un programa en Fortran en simple precisión los siguientes cálculos, matemáticamente equivalentes,

a) $1\,000\,000 \times 0.1$

b) $\sum_{n=1}^{1\,000\,000} 0.1$

c) $\sum_{m=1}^{1\,000} \left(\sum_{n=1}^{1\,000} 0.1 \right)$

Explique las diferencias obtenidas entre resultados finales de a), b) y c) y muestre que el error relativo en b) es del orden del 1%, pero es mucho menor en c). Resalte la conclusión de este ejercicio.

- d) En los puntos b) y c), vaya guardando con un write (cada 1000 iteraciones en el caso b, y en todas las iteraciones de la suma externa en c), los errores parciales en un archivo de datos. La primera columna con los valores de los índices (multiplicados por 1000 para el caso c) de las sumas y segunda columna con el error parcial, i.e., la diferencia entre el valor de la respectiva suma parcial y el valor exacto ($i * 0.1$). Grafique usando *gnuplot*, los errores parciales de b) y c) superpuestas, en función del índice correspondiente, en un archivo “prob13.pdf” en color, con título de gráfica “Problema 18, Guía 1”, con nombres adecuados en los ejes y con leyendas adecuadas. Los datos del b), graficarla con puntos de tamaño 0.75 (el default es 1), y los de c) con línea de grosor 1.6 (el default es también 1). Luego de ver cómo le queda la gráfica en escala lineal, cambie el eje x, a escala logarítmica. Según su criterio, en qué escala se aprecia mejor el resultado del problema para su análisis? Puede al eje y darle escala logarítmica? Pruebe qué pasa si lo hace. Analice la conveniencia en general del uso de escalas logarítmicas.

Finalmente deje el archivo grabado en escala lineal en y , y logarítmica en x , y diga el tamaño del mismo tanto en bytes como Kbytes.

Problema 14: Escribir un programa que pida una contraseña de tres dígitos y permita leer tres intentos. Si el usuario da la contraseña correcta responde responde “Correcto” y queda inactivo, con este mensaje. En caso contrario el programa escribe “Lo siento, contraseña equivocada” y se cierra de inmediato.

Problema 15: Escribir un programa que, dado un año y el nombre de un mes, saque por pantalla el número de días del mes (tenga en cuenta que algunos años son bisiestos).

Problema 16: Escriba un programa para calcular un valor aproximado de π utilizando

- a) la fórmula recurrente de Arquímedes, que acota π entre P_n y p_n , con $p_n < \pi < P_n$. Siendo p_n y P_n los perímetros de los polígonos regulares de n lados inscriptos y circunscriptos, respectivamente, en la circunferencia de radio $1/2$. La fórmula de recurrencia que encontró Arquímedes es la siguiente:

$$P_{2n} = \frac{2p_n P_n}{p_n + P_n}$$

$$p_{2n} = \sqrt{P_{2n} p_n}$$

Usando los valores $P_6 = 2\sqrt{3}$ y $p_6 = 3$, correspondientes al hexágono, escriba un programa que realice 20 iteraciones, con $n = 6 \times 2^k$, y $k = 1, \dots, 20$, y escriba los resultados en pantalla.

- b) la productoria de Wallis

$$\frac{\pi}{2} = \prod_{n=1}^{\infty} \frac{(2n)^2}{(2n)^2 - 1} = \frac{4}{3} \frac{16}{15} \frac{36}{35} \frac{64}{63} \dots$$

Calcule el valor de π truncando la productoria a 10^6 factores.

Problema 17: Escriba el siguiente programa y explique por qué los valores obtenidos no son iguales:

```
program test
implicit none
integer, parameter :: pr = kind(1.0)
real(pr)           :: sum0,sum1
integer             :: i
!
sum0 = 0._pr ; sum1 = 1._pr
do i=1,10000
  sum0 = sum0 + 1.e-8_pr
  sum1 = sum1 + 1.e-8_pr
```

```

end do
sum0 = sum0 + 1._pr
write(*,*) sum0, sum1
end program test

```

Problema 18: Dado el siguiente programa y su correspondiente subrutina:

<pre> program test implicit none real(kind(1.)) :: x, y, s integer :: i, j i = 6; j = 3; x = 4.; y = 8. call sum(i, x, s) write(*, '(F4.1, " + ", F4.1, " = ", F5.1)') real(i), x, s end program test </pre>	<pre> subroutine sum(z, w, ss) implicit none real(kind(1.)), INTENT(IN) :: z, w real(kind(1.)), INTENT(OUT) :: ss ss = z + w end subroutine sum </pre>
--	--

Escriba, compile y ejecute el programa de las siguiente manera:

- con el programa tal como está, incluyendo la subrutina en el mismo archivo o en archivo separado.
- modificando el programa con la cláusula `CONTAINS`, para que *contenga* a la subrutina.
- creando un modulo que contenga la subrutina, y usando este en el programa.

Verifique que en el primer caso compila sin errores y produce *resultados incorrectos*, y en los otros casos no.

Problema 19: Interprete el resultado de los siguientes programas en virtud de la representación de punto flotante de los números reales.

- ```

program test_igualdad
implicit none
integer, parameter :: pr=kind(1.d0) ! pr puede ser simple o doble

if (19.08_pr + 2.01_pr == 21.09_pr) then
 write(*,*) '19.08 + 2.01 = 21.09 '
else
 write(*,*) '19.08 + 2.01 /= 21.09 '
endif
end program test_igualdad

```

- ```

program test_igualdad2
implicit none
integer, parameter :: pr=kind(1.d0) ! pr puede ser simple o doble
real(pr) :: a

a = 2.05_pr
if (a*100._pr == 205._pr) then
  write(*,*) '2.05*100 = 205 '
else
  write(*,*) '2.05*100 /= 205 '
endif
end program test_igualdad2

```