

Comentarios sobre la Tarea #1

Ana Gabriela Carretas Talamante

Actividad 1

Link para mostrar el código: T1-Act1

La verdad no recordaba nada de fortran (uso la versión fortran 90), entonces hacer esta primera actividad me ayudó para poder recordar comandos: cómo declarar funciones y variables, pensarle un poquito con los loops, aplicar expresiones lógicas, etc. También a desempolvar el cómo usar un graficador (en mi caso he estado utilizando gnuplot) y a manejar el compilador (estoy usando minGW).

Las funciones que utilicé fueron dos:

$$g(x) = e^{-\frac{x^2}{2}} \quad (1)$$

$$h(x) = \frac{1}{1+x^2} \quad (2)$$

mingw-w64 - Acceso directo

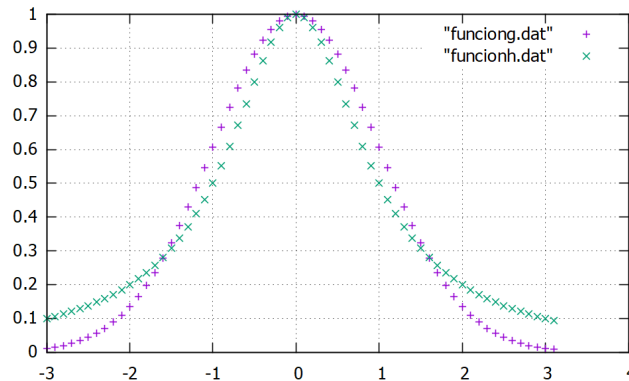
```
C:\Users\Gabriela\Dropbox\DesarrolloExp2\Portafolio1\Tarea1>gfortran T1act1.f90 -o t1-1.exe
C:\Users\Gabriela\Dropbox\DesarrolloExp2\Portafolio1\Tarea1>t1-1.exe
Evaluar en
1
Extremos del intervalo
-3 3
Finura de muestreo
0.1
La funcion h evaluada en 1.00000000 es 0.500000000
Evaluando la funcion h en sus extremos
-3.00000000 0.100000001
-2.90000010 0.106269918
-2.80000019 0.113122158
-2.70000029 0.120627239
-2.60000038 0.128865942
-2.50000048 0.137930989
-2.40000057 0.147928938
-2.30000067 0.158982441
-2.20000076 0.171232775
-2.10000086 0.184842765
-2.00000095 0.199999854
```

(a) Para la función $h(x)$.

mingw-w64 - Acceso directo

```
3.09999871 9.42507759E-02
La funcion g evaluada en 1.00000000 es 0.606530666
Evaluando la funcion g en sus extremos
-3.00000000 1.11089963E-02
-2.90000010 1.49207804E-02
-2.80000019 1.98410843E-02
-2.70000029 2.61213910E-02
-2.60000038 3.40474173E-02
-2.50000048 4.39368822E-02
-2.40000057 5.61346896E-02
-2.30000067 7.10052401E-02
-2.20000076 8.89214650E-02
-2.10000086 0.110250324
-2.00000095 0.135335028
-1.90000093 0.164474174
-1.80000091 0.197898373
-1.70000088 0.235745728
-1.60000086 0.278036922
```

(b) Para la función $g(x)$.



(c) Gráfica de las funciones evaluadas en un intervalo dado.

Figura 1: Vista en la terminal al correr el programa de la actividad 1.

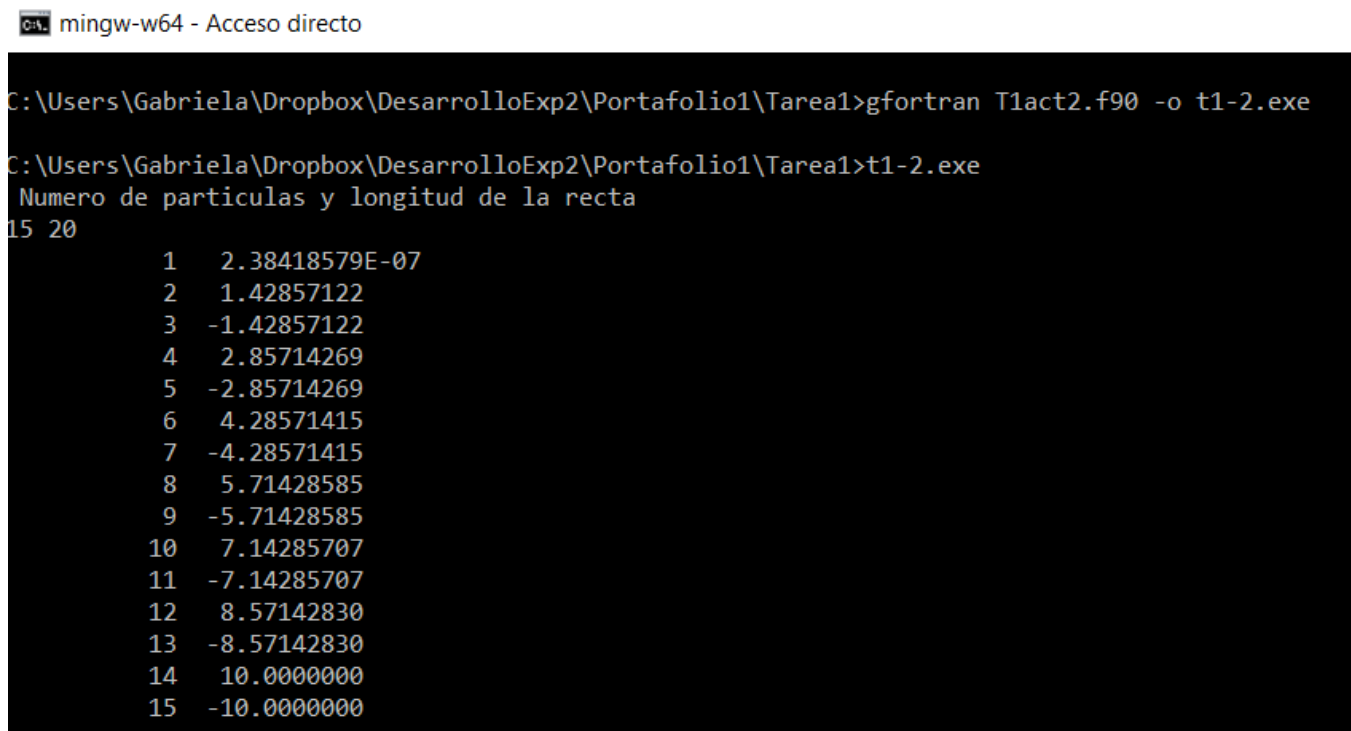
Actividad 2

Link para mostrar el código: T1-Act2

Fue la actividad con la que más batallé, principalmente para poder formular un algoritmo que acomodara las pares y las impares donde se pretendía (en los lugares positivos/negativos).

Primero, me preocupaba que no estuvieran dentro de la recta definida por la longitud que ingresa el usuario, tal caso era cuando se ingresaba un número de puntos impar, pues el programa se saltaba el cero (ya que no es positivo ni negativo). Ahí un número se repetía, y pues ya no era uniforme la distribución de las partículas. Es por eso que añadí el condicional que comienza en la línea 33, para que se modificara el algoritmo e incluyera la posición del cero cuando fuera una N impar.

Sobre cómo acomodaba las partículas, utilicé dos contadores (ip, iimp) que me decían cuántos puntos pares/impares ya había acomodado en la recta. Esto era necesario porque el programa comienza a acomodar a los puntos en los extremos de la recta, y conforme vaya recorriendo los N (recordemos que en cada lado irían N/2 de puntos), nota si es par/impar, y le resta/suma las distancias correspondientes, hasta llegar al origen. Para el caso de las N impares, se recorre el contador de partículas totales por lado para que incluya al 0. Creo que no es la manera más rápida, pero al momento fue lo que se me ocurrió.



```
C:\mingw-w64 - Acceso directo
C:\Users\Gabriela\Dropbox\DesarrolloExp2\Portafolio1\Tarea1>gfortran T1act2.f90 -o t1-2.exe
C:\Users\Gabriela\Dropbox\DesarrolloExp2\Portafolio1\Tarea1>t1-2.exe
Numero de particulas y longitud de la recta
15 20
1 2.38418579E-07
2 1.42857122
3 -1.42857122
4 2.85714269
5 -2.85714269
6 4.28571415
7 -4.28571415
8 5.71428585
9 -5.71428585
10 7.14285707
11 -7.14285707
12 8.57142830
13 -8.57142830
14 10.0000000
15 -10.0000000
```

Figura 2: Vista en la terminal al correr el programa de la actividad 2.

Actividad 3

Link para mostrar el código: T1-Act3

Después de hacer la actividad 2, me quedé con la idea de acomodar pares e impares en los positivos/negativos del plano, por lo que la primera versión de mi programa estaba cumpliendo esa condición. Era prácticamente el mismo código, solo le anidé un loop para las posiciones en y. Pasaron unos días para darme cuenta que me estaba complicando la vida, y lo cambié para que solo acomodara regularmente las N partículas, quitando las condiciones de N par o impar de todas partes.

El segundo problema lo encontré haciendo la actividad 4 del segundo portafolio (casi la misma pero en 3D), y era que estaba pidiendo un número total de partículas, ¡pero en realidad estaba acomodando $N*N$ en el plano! Entonces como modificación le puse una condicional que pide ingresar un número total de partículas que tenga raíz cuadrada exacta, para que realmente sea un arreglo regular cuadrado.

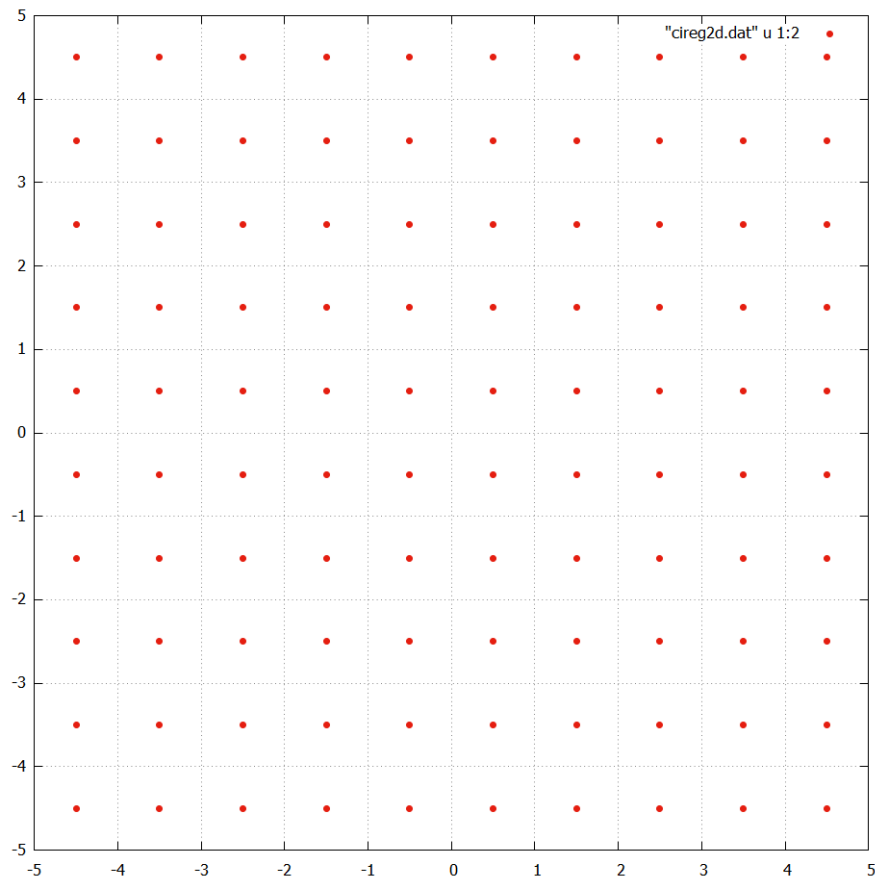


Figura 3: Arreglo regular 2D en una celda cuadrada para $N=100$, $L=5$.