Tarea 4

Gustavo Hernández Angeles[1]

[1]Facultad de Ciencias Físico Matemáticas – Universidad Autónoma de Nuevo León.

gustavoha40@gmail.com

1 Volumen de un sólido de revolución

El volumen de un sólido de revolución se puede calcular mediante el método de arandelas. Considere un sólido formado cuando $f(x) = 1/x$ comprendida en el intervalo $[1, 2]$ gira sobre el eje x. Haga el programa en FORTRAN que resuelva el problema. Agregue la gráfica obtenida en GNUPLOT y el script empleado.

Solución

El método de las arandelas es una pequeña variación del método de discos, contrastando únicamente en que los anillos tienen un radio inferior, es decir, el volumen es hueco. Para el programa utilizaremos el método de sumatoria (es decir, la sumatoria de Riemann), iniciamos definiendo unas cuantas variables: pi, iniciamos la integral $I = 0$ al tratarse de una sumatoria, los extremos del intervalo x_i, x_f, un auxiliar $x2$, la anchura de cada arandela o anillo dx y el número de anillos dados dx, n.

```
program arandelas
    implicit none
    real*8 :: f, pi = 4.d0*atan(1.d0), I=0, xi, xf, dx,x2
    integer :: n, j
    write(*,'(a)',advance='no')'Limite inferior y superior del intervalo: '
    read*,xi,xf
    write(*,'(a)',advance='no')'Anchura de las arandelas: '
    read*,dx
    n = int(abs(xf-xi)/dx)
```

Ahora hacemos un ciclo con la variable j que vaya generando la posición j-ésima del intervalo para así evaluar la integral. En la expresión de la sumatoria realicé un promedio de las "alturas"dadas por $f(x)$ respecto a la posición actual con una anterior con el objetivo de tener una mayor precisión. Al finalizar el ciclo solamente nos queda multiplicar nuestro valor de la integral por pi para obtener el volumen de revolución.

```
    do j = 1, n
        x2 = xi + j*dx
        I = I + f(x2)*dx
    enddo
    I = I*pi
    write(*,'(a,f20.16)')'El volumen del solido de revolucion es: ',I
end
```

Pero la mayor particularidad de este problema es cómo definimos la función a integrar, que analíticamente tendría que ser $[f(x)]^2 - [g(x)]^2$ con $f(x) > g(x)$. Con este fin, recreé esta expresión de tal manera que solo necesitemos una función para definir esta diferencia, las variables f y g serán dos funciones que definimos nosotros, y el valor que vaya a tomar la función $f(x)$ está determinado por las condicionales: Si $f(x) \geq g(x)$ entonces la expresión a integrar es $[f(x)]^2 - [g(x)]^2$, caso contrario, la expresión a integrar será $[g(x)]^2 - [f(x)]^2$.

```
real*8 function f(x)
    real*8 :: x,g
    f = 1/x   ! Función 1
    g = 0     ! Función 2
    if (f.ge.g) f = f**2 - g**2
    if (f.lt.g) f = g**2 - f**2
    return
end
```

En nuestro caso, tendremos que determinar el volumen de la función $f(x) = 1/x$ cuando está girando alrededor del eje x con un radio menor, que asumiremos por la falta en la redacción, igual a 0. La gráfica en Gnuplot quedaría de la siguiente manera:

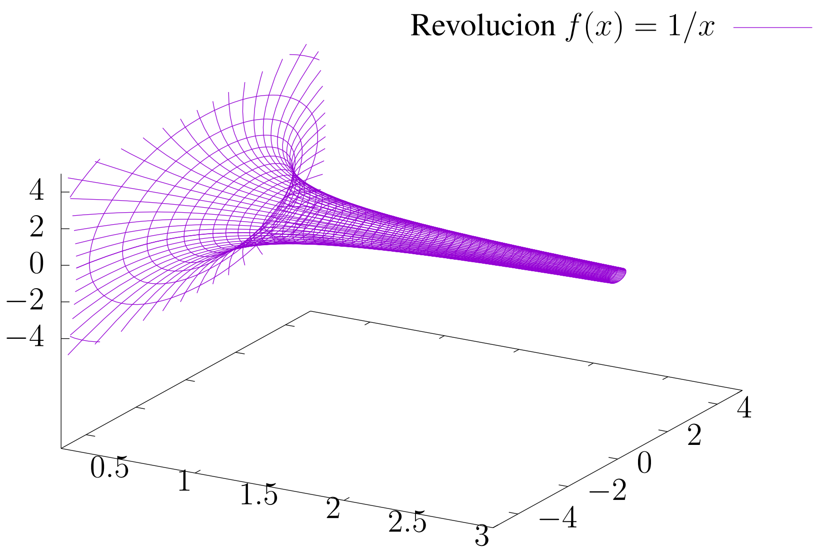


Figura 1. Sólido de revolución $f(x)$ girando alrededor del eje x.

Ejecutando el programa con $f(x) = 1/x$, $g(x) = 0$ en el intervalo $[1, 2]$ podremos obtener el siguiente resultado.

```
Limite inferior y superior del intervalo: 1,2
Anchura de las arandelas: 0.001
El volumen del solido de revolucion es:   1.5700109937865236
```

Lo que está muy cerca del valor calculado en WolframAlpha:

Integral definida

$$\int_1^2 \frac{\pi}{x^2}\,dx = \frac{\pi}{2} \approx 1{,}5708$$

El script utilizado para poder graficar el sólido de revolución en GNUPLOT es el siguiente:

```
set terminal epslatex color
set output 'rev.tex'
set parametric
set isosamples 50
splot [0.1:3] [-pi:pi] [0.1:3] [-5:5] [-5:5] u, 1/u*cos(v), 1/u*sin(v)
    t 'Revolucion $f(x) = 1/x$'
```

2 Área entre curvas

Encuentre el área comprendida entre las curvas $f(x) = x^2 - 3$ y $g(x) = 1$. Haga el programa en `FORTRAN` que resuelva el problema.

Solución

En este problema se puede utilizar el algoritmo que realicé en la tarea pasada, que de manera general halla el área comprendida entre dos funciones con la restricción de que la región a integrar debe ser del tipo simple. Y para graficar podemos utilizar GNUPLOT con el comando filledcurves para rellenar el espacio entre curvas.

Analíticamente podemos deducir que las intersecciones de ambas gráficas son con $x_1 = -2$ y $x_2 = 2$, sin embargo utilicé un algoritmo que se programó en una tarea anterior, la cuál encuentra las múltiples raíces de una función y los intervalos (tan pequeños como deseemos) de donde se encuentran las raíces. En este caso, utilizamos la función $h(x) = |f(x) - g(x)|$ cuyas raíces son las intersecciones de las gráficas $f(x)$ y $g(x)$. El código se explicó en una tarea anterior.

```
 Inserte el intervalo donde se encuentren las intersecciones[a,b]:
-5,5
El intervalo acotado es aproximadamente [ -2.000000,  1.999000].
 El area de la integral es:   10.6666727
```

Figura 3. Solución numérica mediante Simpson con $n = 1000$.

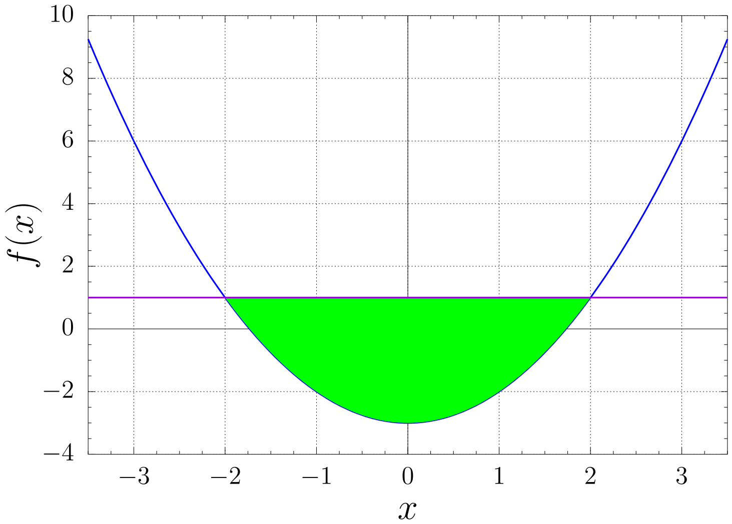


Figura 2. Área entre las curvas

Podemos hacer la comparación con el resultado analítico, el cual es el siguiente resultado:

$$I = \int_{-2}^{2} (1 - x^2 + 3)dx = 2\int_{0}^{2} (4 - x^2)dx = -\frac{16}{3} + 8 \cdot 2 = 32/3 = 10.6666... \quad (1)$$

Por lo que la solución numérica posee un error relativo del orden de 10^{-5}. El script empleado para la gráfica en GNUPLOT es el siguiente:

```
set terminal epslatex color
set output "area.tex"
set zeroaxis lt -1 lc -1
unset key
set xlabel '$x$'
set ylabel '$f(x)$'
set grid xtics
set grid ytics
set mytics 4
set mxtics 4


f(x) = x**2-3
plot [-3.5:3.5] f(x) w l lw 3 lc "blue",[-2:2] f(x) w filledcurves y1 = 1 fc "green"
    ,[-3.5:3.5] 1 w l lw 3 lc 1
```

3 Método de Simpson compuesto

Haga el programa en `FORTRAN` del método de Simpson compuesto que se describe en el .pdf proporcionado por TEAMS.

Solución

El método de Simpson compuesto no es más que el mismo método de Simpson simple con el único añadido que podremos escoger el número de decimales que deseamos, aúnado a eso, se realiza iterativamente el método de Simpson simple con un número de subíntervalos cada vez más grandes creciendo de forma exponencial. Utilizaremos que $n = 2^i$, donde i será la iteración del método.

```
subroutine simple(xi,xf,n,i)
    implicit none
    real*8 :: h,a1,xf,xi,f,x2,sum,i
    integer :: j,n
    sum=0.0
    h = 1.0*abs(xf-xi)/n
    a1 = (h/3.d0)*(f(xi)+f(xf))

    do j=1,n-1
        x2 = xi + h*j
        if (mod(j,2).eq.1) then
            sum = sum + 4.0*f(x2)
        else
            sum = sum + 2.0 * f(x2)
        end if
    end do

    i = h/3.0 * sum + a1
    return
end subroutine
```

Antes que nada, como estaré realizando una integral por cada vez que se quiera conseguir más precisión, decidí introducir el algoritmo de Simpson simple en una subrutina cuyos valores de referencia serán el intervalo, el número de particiones y la integral I.

```
program SimpsonCompuesto
    integer :: n=2, m
    real*8 :: xi,xf,dl=10,i,auxi=3.0e8,tolerancia
    write(*,'(a)',advance='no')'Extremos del intervalo inferior y superior: ' ;read*,xi,xf
    write(*,'(a)',advance='no')'Digitos decimales correctos: ';read*,m
    tolerancia = 10.0**(-m)
```

Ahora sí, en esta primera parte del código definimos nuestras variables de entrada e iniciales; el intervalo de integración $[x_i, x_f]$, el número de dígitos correctos (precisión) con m, los cuales serán nuestra tolerancia como 10^{-m}, comenzamos con 2 particiones, y las variables dl y auxi son únicamente auxiliares para definir nuestra precisión.

```
    do while (dl.gt.tolerancia)
        call simple(xi,xf,n,i)
        print*,n,i
        n=n*2
        dl = abs(i-auxi)
        auxi=i
    end do
    write(*,*)'El valor de la integral es:',i
    stop
end
```

En cada iteración del código se obtiene un valor de la integral con la subrutina, se imprime el valor de la integral junto a el número de particiones que se utilizó para la misma, se duplican las particiones y se define el error como la diferencia entre la integral que se acaba de calcular y el valor de la integral anterior, que se guardó en la variable auxiliar "auxi". Al final se imprime el valor de la integral con la precisión deseada.

```
Extremos del intervalo inferior y superior: 0,4
Digitos decimales correctos: 5
           2   56.769582952577892
           4   53.863845745864133
           8   53.616220796005820
          16   53.599304589454093
          32   53.598222595284000
          64   53.598154574603676
         128   53.598150317084446
 El valor de la integral es:   53.598150317084446
```

En el dado caso en que querramos realizar la integral que se describe en el pdf con 5 decimales de precisión, este es el resultado. La integral en concreto es:

$$\int_0^4 e^x dx$$

4 Cálculo de π

Programe un algoritmo para aproximar el valore de π a 7 cifras significativas.

Solución

Intentaremos aproximar la integral mediante un algoritmo de Montecarlo, en concreto la variante que trata con el cociente del área de un cuadrado con el de un circulo. La idea es arrojar puntos aleatorios en un cuadrado unitario y verificar cuántos puntos caen dentro de un circulo inscrito y centrado en el cuadrado. Gracias a la subrutina intrínseca RANDOM_NUMBER() en el compilador GNU Fortran podemos generar números aleatorios al instante. Para comenzar con el procedimiento se definen las variables que utilizaremos:

```
program calculopicuadrado

    implicit none
    integer :: i, totales, dentro=0
    real*8 :: x1, y1, xc = 0.5d0 , yc = 0.5d0, propuesta, distance
    write(*,'(a)',advance = 'no')'Numero de iteraciones: ';read*,totales
```

Utilizaremos las variables x1, y1 como las coordenadas del punto que estemos generando, y xc, yc serán las coordenadas del centro del círculo inscrito. "propuesta"será la variable de la estimación de π, la variable "distancia"únicamente define la distancia entre el punto generado y el centro del círculo.

```
do i = 1,totales
    call random_number(x1)
    call random_number(y1)
    distance = ((xc-x1)**2.d0+(yc-y1)**2.d0)**(1.d0/2.d0)
    if (distance .lt. 0.5) dentro = dentro + 1
enddo
```

Una vez dentro del ciclo, cada iteración será cada punto que generemos y determinemos si está dentro del círculo o no. Para generar los puntos utilizamos la subrutina random_number() cuyo argumento es la variable donde guardaremos el numero aleatorio generado, la subrutina únicamente genera números aleatorios en el rango $(0, 1)$, por lo que convenientemente situamos nuestro cuadrado en la región $[0, 1] \times [0, 1]$ del plano real, y es por esto que el centro del círculo se encuentra en $(x_c, y_c) = (0.5, 0.5)$. Se define la distancia entre ambos puntos y si está es menor a 0.5 (el radio) decimos que está dentro del círculo.

```
    propuesta = 4.d0*(dble(dentro)/dble(totales))
    write(*,'(a,f18.15)')'El valor calculado para pi es:',propuesta
end
```

Y por último, nuestro valor de π aproximado por la relación $\pi \sim 4\frac{P_{circulo}}{P_{total}}$ y se imprime en la terminal.

```
Numero de iteraciones: 300000000
El valor calculado para pi es: 3.141595186666667
```

Ejecutando el programa, el resultado muestra que el código converge muy lentamente al valor de pi, teniendo que realizar hasta 300 millones de puntos para obtener 5 decimales correctas. Para este número de puntos el código tarda alrededor de 2 minutos.

5 Olas del mar

La velocidad de las olas en el mar depende de la longitud de onda λ, que es la distancia entre cresta y cresta, y para aguas poco profundas también depende de la profundidad del agua h por donde viajan las olas. La función que relaciona estas variables es:

$$v^2 = \frac{g\lambda}{2\pi} \tanh \frac{2\pi h}{\lambda} \tag{2}$$

donde v es la velocidad, y g es la gravedad.

- a Encuentre las expresiones analíticas para los casos límites, es decir, para valores de h/λ grandes y pequeños.
- b ¿Cuál es la velocidad de la ola, cuando $h = 2000\text{m}$ y $\lambda = 300\text{m}$?
- c ¿Qué h se necesita para tener esa misma velocidad de la ola en aguas poco profundas?
- d ¿Cuál es la velocidad de la ola cuando $h = 5\text{m}$ y $\lambda = 4\text{m}$.
- e Suponiendo que se mantiene h constante, esto es, que el oleaje se propaga por un océano de profundidad constante ($h = 50\text{m}$), determinar si la velocidad aumenta o disminuye cuando $\lambda = 5\text{m}$.
- f ¿Por qué se dice que cuando la relación h/λ es mayor o igual a $1/7$ la ola rompe? Justifique su respuesta.

Solución

5.1. a) Encuentre las expresiones analíticas para los casos límites, es decir, para valores de h/λ grandes y pequeños.

a) Para $\frac{h}{\lambda}$ grandes.

$$V^2 = \frac{g\lambda}{2\pi} \lim_{h/\lambda \to \infty} \tanh\left(\frac{2\pi h}{\lambda}\right)$$

• Calculamos $\lim_{h/\lambda \to \infty} \tanh\left(2\pi \frac{h}{\lambda}\right)$:

$$\lim_{h/\lambda \to \infty} \tanh\left(2\pi \frac{h}{\lambda}\right) = \lim_{h/\lambda \to \infty} \frac{e^{2\pi \frac{h}{\lambda}} - e^{-2\pi \frac{h}{\lambda}}}{e^{2\pi \frac{h}{\lambda}} + e^{-2\pi \frac{h}{\lambda}}} \cdot \frac{e^{-2\pi \frac{h}{\lambda}}}{e^{-2\pi h/\lambda}} =$$

$$\lim_{h/\lambda \to \infty} \frac{1 - e^{-4\pi \frac{h}{\lambda}}}{1 + e^{-4\pi \frac{h}{\lambda}}} = \frac{1-0}{1-0} = 1$$

$\therefore V^2 = \frac{g\lambda}{2\pi}$ para $\frac{h}{\lambda}$ grandes

Para h/λ pequeños.

$$v^2 = \frac{g\lambda}{2\pi} \lim_{h/\lambda \to 0} \tanh\left(2\pi \frac{h}{\lambda}\right)$$

• Calculamos $\lim_{h/\lambda \to 0} \tanh\left(2\pi \frac{h}{\lambda}\right)$:

$$\lim_{h/\lambda \to 0} \tanh\left(2\pi \frac{h}{\lambda}\right) = \lim_{h/\lambda \to 0} \frac{e^{2\pi \frac{h}{\lambda}} - e^{-2\pi \frac{h}{\lambda}}}{e^{2\pi \frac{h}{\lambda}} + e^{-2\pi \frac{h}{\lambda}}} \cdot \frac{e^{2\pi \frac{h}{\lambda}}}{e^{2\pi \frac{h}{\lambda}}} =$$

$$= \lim_{h/\lambda \to 0} \frac{e^{4\pi \frac{h}{\lambda}} - 1}{e^{4\pi \frac{h}{\lambda}} + 1} = \frac{1-1}{2} = 0$$

$$\therefore\ v^2 = 0 \text{ para } \frac{h}{\lambda} \text{ pequeños.}$$

5.2. b) ¿Cuál es la velocidad de la ola, cuando $h = 2000$m y $\lambda = 300$m?

Ya hemos visto lo que pasa cuando la relación h/λ obtiene un valor relativamente grande, en este caso, ese valor es de alrededor de 6.666 por lo que podemos utilizar la expresión:

$$v = \sqrt{\frac{g\lambda}{2\pi}} \tag{3}$$

Al sustituir el valor $\lambda = 300m$ y el parámetro $g = 9.81m/s^2$ resulta:

$$v = \sqrt{\frac{9.81 \cdot 300}{2\pi}}$$

$$\therefore v = 21.64m/s$$

5.3. c) ¿Qué h se necesita para tener esa misma velocidad de la ola en aguas poco profundas?

Suponiendo que la longitud de onda λ permanece intacta, consideramos ahora la contribución de aguas profundas como en (2). Por lo que debemos resolver la siguiente ecuación para h con $v = 21.64m/s$, $\lambda = 300m$.

$$v^2 - \frac{g\lambda}{2\pi} \tanh \frac{2\pi h}{\lambda} = 0 \tag{4}$$

Después de crear la función auxiliar $f(h) = v^2 - \frac{g\lambda}{2\pi} \tanh \frac{2\pi h}{\lambda}$ estuve jugando un rato con su gráfica hasta poder encontrar un intervalo en donde se encuentre la raíz:

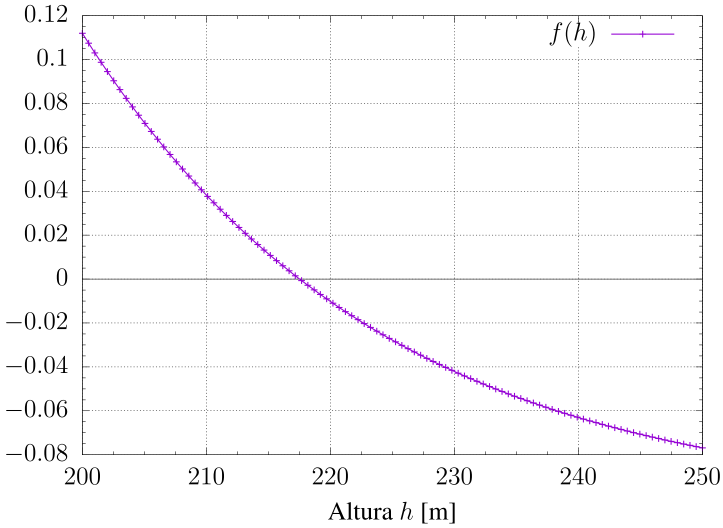


Figura 4. Gráfica de $f(h)$.

Y para no complicar el asunto solo la metí al código para conseguir raíces por el método de la secante (que usted nos pasó) de la siguiente forma:

```
FUNCTION F(h)
  real :: v = 21.64, g = 9.31, la = 300, pi = 4.0*atan(1.0)
  F = v**2 - (g*la/(2*pi)) * tanh(2*pi*h/la)
RETURN

END
```

Ejecutando el código podemos encontrar nuestro resultado:

```
 Dame los extremos del intervalo que contienen la raiz:
210,230
   5    217.51376719      0.00000006
```

Por lo tanto, el valor de h requerido para esa misma velocidad en aguas poco profundas es $h \simeq 217.5137m$. El script de la gráfica es el siguiente:

```
set terminal epslatex color
set output "5c.tex"
set zeroaxis lt -1 lc -1
set xlabel 'Altura $h$ [m]'
set grid xtics
set grid ytics
set mytics 4
set mxtics 4
a = 200
b = 250
la = 300
```

```
g = 9.81
v = 21.64
f(x) = v**2 - (g*la/(2*pi)) * tanh(2*pi*x/la)
plot [a:b] f(x) t '$f(h)$' w lp lw 2
```

5.4. d) ¿Cuál es la velocidad de la ola cuando $h = 5$m y $\lambda = 4$m.

En este caso, la relación h/λ no tiende a ningún caso límite, por lo que debemos considerar la aportación de la tangente hiperbólica y por tanto, utilizar la expresión (2):

$$v = \sqrt{\frac{9.81 \cdot 4}{2\pi}} \tanh^{1/2} \frac{2\pi \cdot 5}{4}$$

$$\therefore v \simeq 2.5m/s$$

5.5. e) Suponiendo que se mantiene h constante, esto es, que el oleaje se propaga por un océano de profundidad constante ($h = 50$m), determinar si la velocidad aumenta o disminuye cuando $\lambda = 5$m.

Para este inciso elaboré un pequeño programa de FORTRAN que evalúa la primera derivada de la función $f(\lambda) = v^2(\lambda, h = 50m)$ en una longitud de onda dada. Las funciones las definimos de esta forma:

```
real*8 function v2(la)
    real*8 :: g = 9.81, pi = 4.d0*atan(1.d0), h = 50, la
    v2 = (g*la)/(2*pi) * tanh(2*pi*h/la)
    return
end function

real*8 function der(la)
    real*8 :: v2, la, k = 1.0e-7
    der = (v2(la+k)-v2(la-k))/(2.0*k)
    return
end function
```

Y el procedimiento para determinar si la velocidad aumenta o decrece fue mediante condicionales, proponiendo que si la derivada es mayor a 0 entonces crece, si es menor a 0 entonces decrece y si se hace 0 entonces es constante con esa longitud de onda.

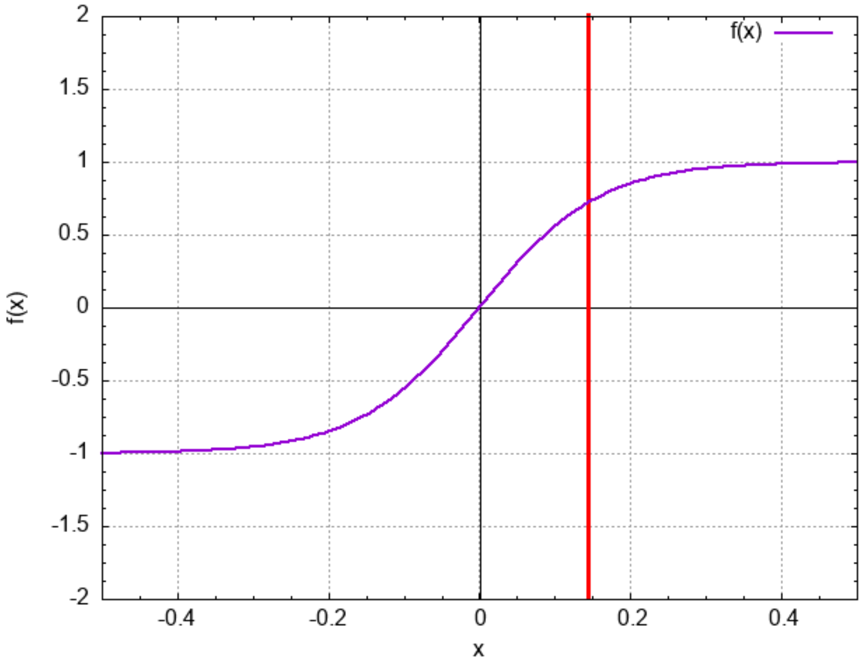
```
program problema5
implicit none
    real*8 :: lambda, der
    write(*,'(a)',advance='no')'Longitud de onda a evaluar: ';read*,lambda
    if (der(lambda).gt.0.) write(*,*)'La velocidad aumenta'
    if (der(lambda).lt.0.) write(*,*)'La velocidad disminuye'
    if (der(lambda).eq.0.) write(*,*)'La velocidad se mantiene constante'
end program
```

Dandonos como resultado lo siguiente:

```
Longitud de onda a evaluar: 5
 La velocidad aumenta
```

5.6. f) ¿Por qué se dice que cuando la relación h/λ es mayor o igual a $1/7$ la ola rompe? Justifique su respuesta.

Esto es debido a que la velocidad de las partículas del agua en la cresta de la ola comienza a aumentar a su valor máximo, rebasándo la celeridad total de la ola y por lo tanto, causando el rompimiento de la misma [Miche, 1944]. Este resultado se puede apreciar en la gráfica de la función $f(x) = \tanh(2\pi x)$:



6 Conclusión

En el primer problema pude ver el detalle de que no es necesario crear muchas funciones si todas las funciones tienen un único propósito; operar entre ellas, además de realizar graficas paramétricas en Gnuplot. En el segundo problema aprendí a graficar el área de una curva en Gnuplot. En el tercer problema ocurrió algo interesante, el programa fallaba al llamar iterativamente a la subrutina simple() y esto es debido a que en Fortran las variables locales que es inicializada (se le da un valor) al declararla, tiene implícitamente un atributo de guardado, es decir, cada vez que se llamaba la subrutina el valor de la variable SUM no se reiniciaba. Esto se solucionó simplemente inicializando la variable después de declararla. En https://www.cs.rpi.edu/ szymansk/OOF90/bugs.html se dan una revisada a este y otros errores peculiares que vienen con Fortran 90. En el cálculo de pi conocí las múltiples subrutinas intrínsecas que vienen con el compilador GNU Fortran.