Universidad Autónoma de Nuevo León

Facultad de Ciencias Físico Matemáticas

Actividad 20

Autores:
Diego Alejandro Téllez Martínez 1941559
Alexia Sofía Ibarra García 1841890
Cristian Joel Gallegos Yáñez 1990064
Gustavo Hernández Ángeles 1989978

Materia:
Física Computacional

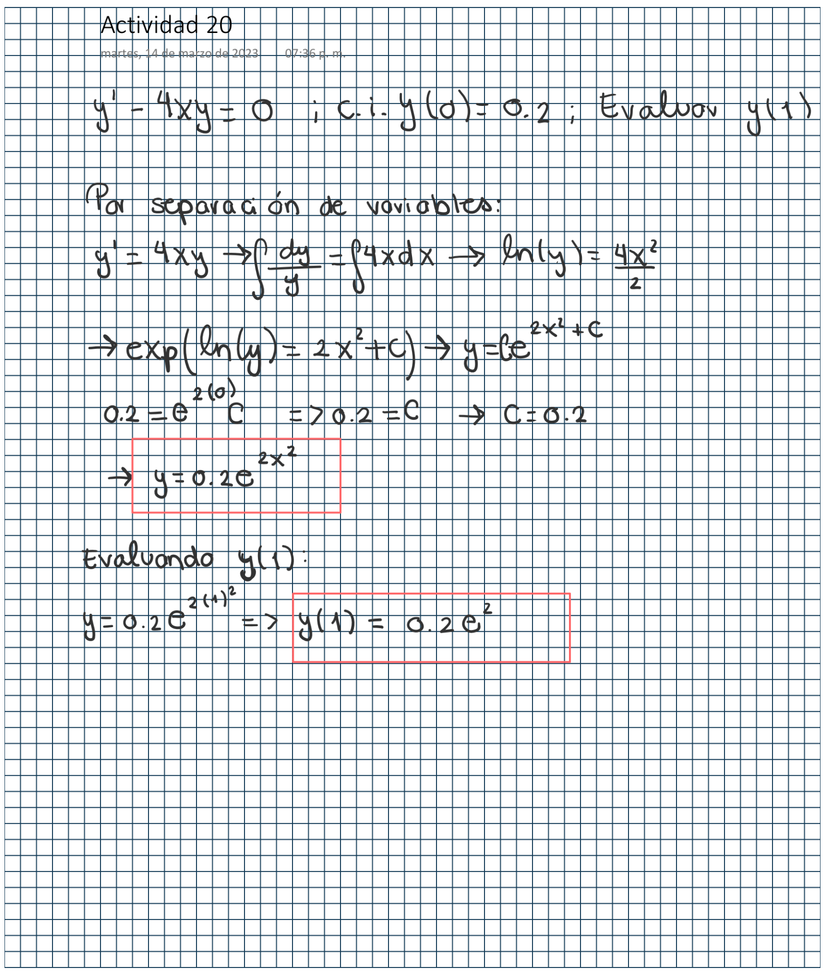
Profesor:
Alfredo Tlahuice Flores *16 de marzo de 2023*

Actividad 20

1. Resolver analítica y numéricamente, usando el método de Runge-Kutta 4to orden, las siguientes ecuaciones diferenciales

$$y' - 4xy = 0\, ; y\,(0) = 2.0$$

Evaluar $y(1)$
Solución Analítica


Actividad 20
martes, 14 de marzo de 2023 07:36 p. m.

$y' - 4xy = 0$; c.i. $y(0) = 0.2$; Evaluar $y(1)$

Por separación de variables:

$y' = 4xy \rightarrow \int \frac{dy}{y} = \int 4x\,dx \rightarrow \ln(y) = \frac{4x^2}{2}$

$\rightarrow \exp\left(\ln(y) = 2x^2 + C\right) \rightarrow y = Ce^{2x^2+C}$

$0.2 = e^{2(0)}C \Rightarrow 0.2 = C \rightarrow C = 0.2$

$\rightarrow \boxed{y = 0.2e^{2x^2}}$

Evaluando $y(1)$:

$y = 0.2e^{2(1)^2} \Rightarrow \boxed{y(1) = 0.2e^2}$

Método de Runge-Kutta 4to orden

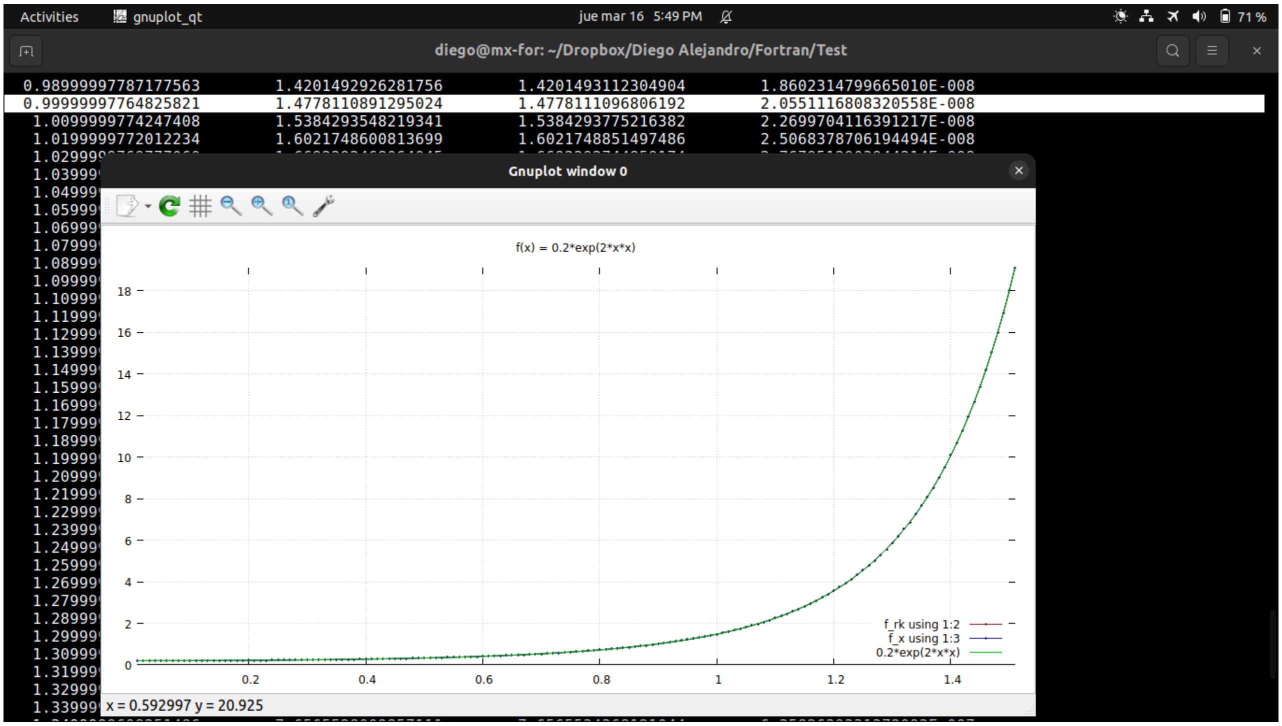


Figura 1: y(1) con un error de 2.055×10^{-8}

$$y' = \frac{4x(y + \sqrt(y))}{1 + x^2} \; ; y\,(0) = 1.0$$

Evaluar $y(1)$
Solución Analítica

$$y' = \frac{4x(y+\sqrt{y})}{1+x^2} \quad ; \quad y(0) = 2.0$$

Solución

$$\frac{dy}{dx} = \frac{4x(y+\sqrt{y})}{1+x^2} \rightarrow \frac{1}{y+\sqrt{y}}\,dy = \frac{4x}{1+x^2}\,dx$$

$$\int \frac{dy}{y+\sqrt{y}} = 4\int \frac{x}{1+x^2}\,dx \qquad (1)$$

Donde

$$4\int \frac{x}{1+x^2}dx = 2\int \frac{dv}{v} = 2\ln|v| = 2\ln|1+x^2| + c \qquad (2)$$

sea $v = 1+x^2$; $dv = 2x\,dx$

$$\int \frac{dy}{y+\sqrt{y}} = \int \frac{2u}{u^2+u}du = 2\int \frac{1}{1+u}du = 2\int \frac{du'}{u'} = 2\ln|u'|$$

sea $u = \sqrt{x}$ $\quad du = \frac{1}{2}\frac{1}{\sqrt{x}}dx$

sea $u' = 1+u$; $du' = du$

$$\int \frac{dy}{y+\sqrt{y}} = 2\ln|1+u| = 2\ln|1+\sqrt{y}| + c \qquad (3)$$

De (2) y (3) en (1)

$$2\ln|1+\sqrt{y}| = 2\ln|1+x^2| + c \Rightarrow \ln|1+\sqrt{y}| = \ln|1+x^2| + c/2$$

$$1+\sqrt{y} = \frac{c_0}{2}(1+x^2) \Rightarrow \sqrt{y} = \frac{c_0}{2}(1+x^2) - 1$$

$\boxed{\therefore\ y(x) = \left[\frac{1}{2}c_0(1+x^2) - 1\right]^2}$ sol. general

Dado que $y(0) = 2$

$$y(0) = \left[\tfrac{1}{2}c_0(1+0) - 1\right]^2 = 2 \rightarrow c_0 = 2$$

$$\Rightarrow y(x) = \left[2(1+x^2) - 1\right]^2 = \left[2 + 2x^2 - 1\right]^2 = \left[2x^2+1\right]^2$$

$\boxed{\therefore\ y(x) = \left[2x^2+1\right]^2}$ sol. particular

Método de Runge-Kutta 4to orden

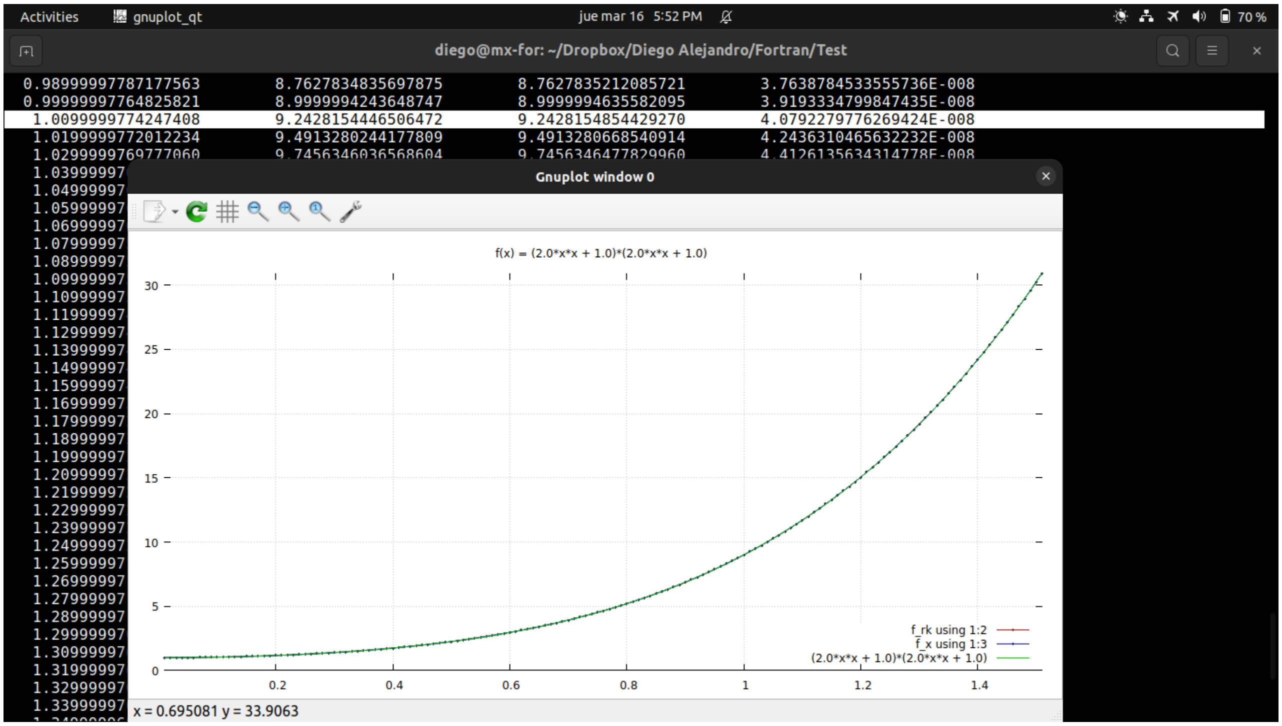


Figura 2: y(1) con un error de 4.079×10^{-8}

$$y' = 3x^2y + 6x^2 \; ; y\,(0) = 2.0$$

Evaluar $y(4)$
Solución Analítica

$$y' = 3x^2 y + 6x^2$$
$$y' = 3x^2(y+2)$$
$$\frac{y'}{y+2} = 3x^2 \rightarrow \int \frac{dy}{y+2} = \int 3x^2 dx$$
$$\ln(y+2) = x^3 \rightarrow \exp(\ln(y+2)) = \exp(x^3 + C)$$
$$\rightarrow y+2 = e^{x^3+C} \rightarrow \boxed{y = e^{x^3+C} - 2}$$
$$\underline{y(0) = 2}$$
$$\rightarrow 2 = Ce^0 - 2$$
$$C = 2 + 2 = 4$$
$$\rightarrow \boxed{y = 4e^{x^3} - 2}$$
$$\boxed{y(4) = 4e^{4^3} - 2 = 2.48 \times 10^{28}}$$

Método de Runge-Kutta 4to orden

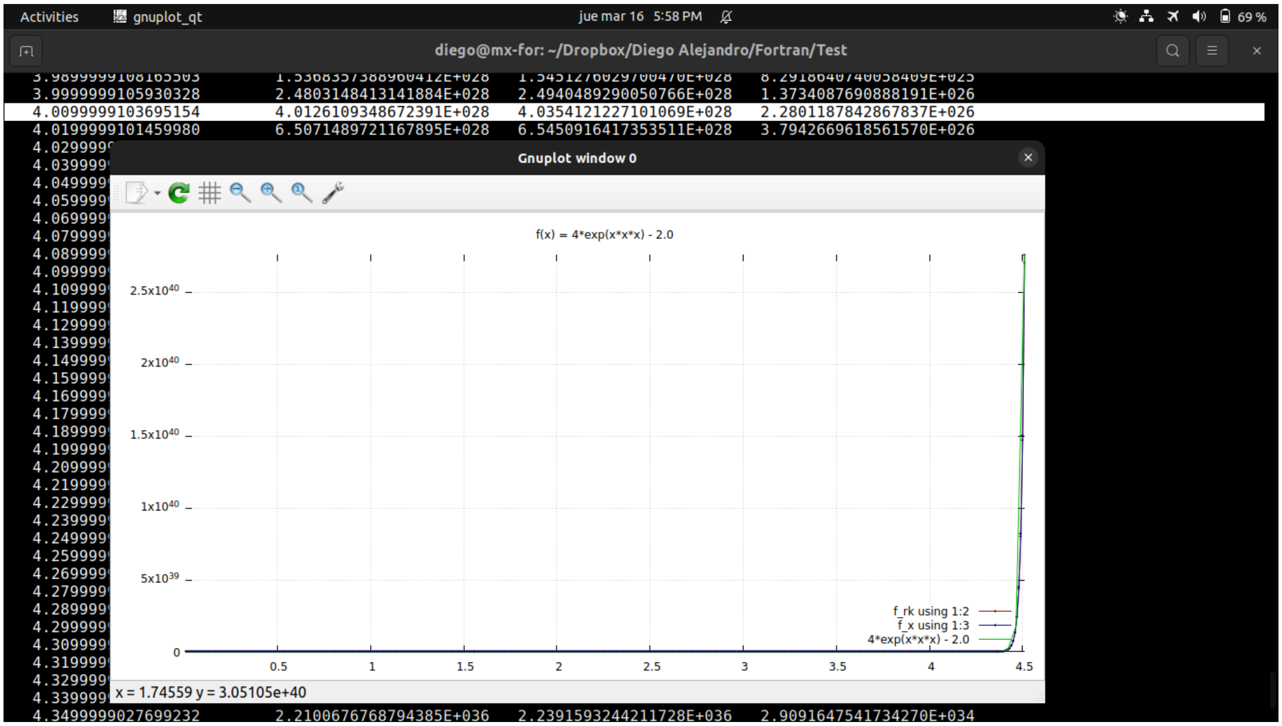
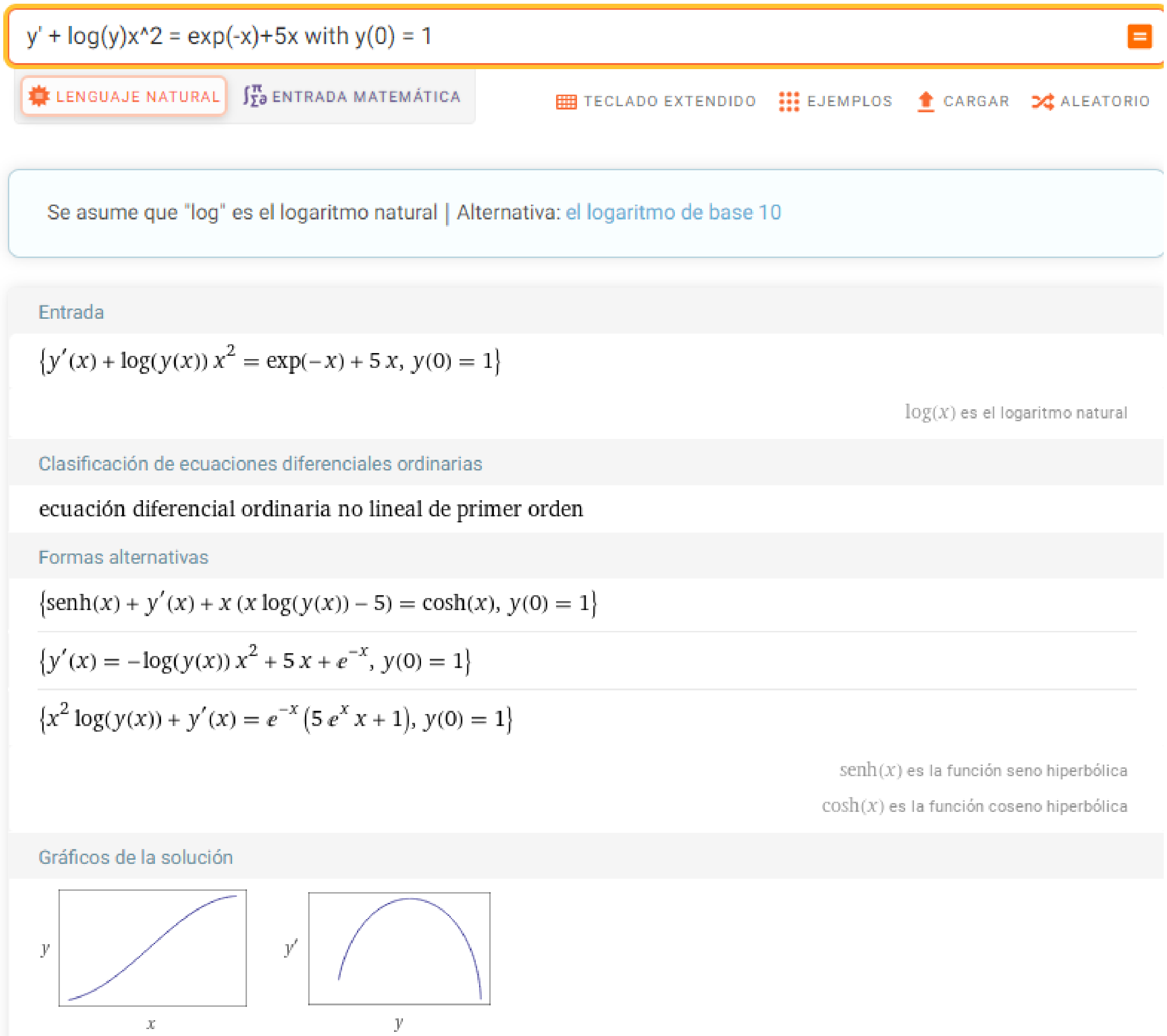


Figura 3: y(4) con un error de 4.079×10^{-8}

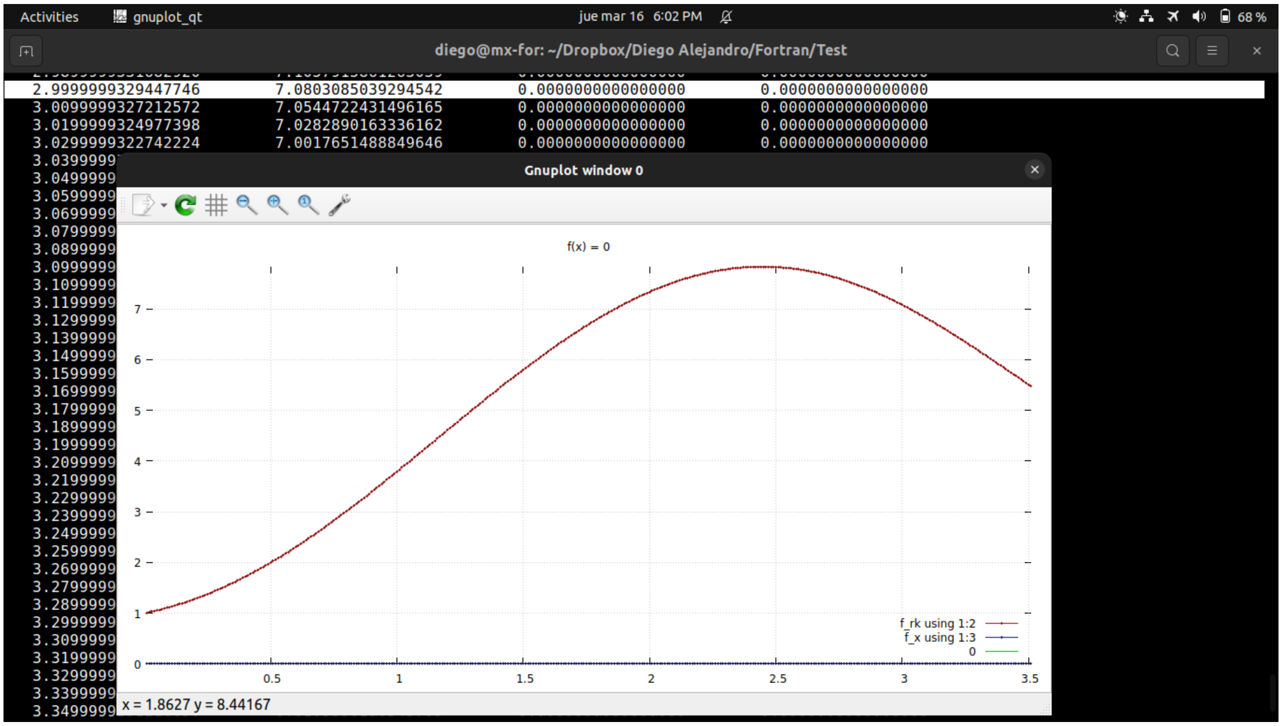
$$y' + x^2 \ln(y) = 5x + e^{-x} \,; y(0) = 1.0$$

Evaluar $y(3)$
Solución Analítica
Dado que no pudimos encontrar una solución analítica, procedimos a verificarla en WolframAlpha obteniendo el siguiente resultado:



Donde observamos que tampoco encuentra alguna solución analítica, sin embargo esboza la solución dada la condición inicial y(0) = 1 que se muestra en la gráfica xy.

Método de Runge-Kutta 4to orden



A. Código implementado

```fortran
program runge_4ktt
      implicit none
      real(8):: a, b, y_0, t, yreal, h, error
     ! Step dx
      h = 0.01
      ! Intervalo [a , b]
      a = 0.0
      b = 3.5
      ! Condicion y_0
      y_0 = 1.0
      open(unit = 1, file = 'runge_kutta.dat')
      write(*,'(10x,a,20x,a,20x,a,20x,a,20x)')'| t |','| y |','| yreal |', '| error |'
      call rungekutta(y_0,a,b,h)
      close(1)
      contains
subroutine rungekutta(y_0,a,b,h)
      implicit none
      real(8), intent(inout)::a, h, b, y_0
      real(8):: y,t, f_1, f_2, f_3, f_4
      t = a
      y = y_0
      do while(t <= b)
      f_1 = h*f(y,t)
      f_2 = h*f(y + (1d0/2d0)*f_1, t + h*(1d0/2d0))
      f_3 = h*f(y + (1d0/2d0)*f_2, t + h*(1d0/2d0))
```

```
    f_4 = h*f(y + f_3,t + h)
    y = y + (f_1 + 2d0*f_2 + 2d0*f_3 + f_4)/6d0
    t = t + h ! se incrementa t
    !yreal = 4*exp(t*t*t) - 2.0
    !error = abs(y - yreal)
    write(*,*)t, y, yreal, error
    write(1,*)t, y, yreal
    end do
    call gnu_plot('0')
end subroutine
subroutine gnu_plot(funcion)
  character(len = *), parameter :: datos_arch = 'runge_kutta.dat'  ! datos
  character(len = *), parameter :: gnplt_arch= 'ode_plot.plt' ! archivo gnuplot
  character(len = *), intent(in):: funcion
  integer :: plt
  open (action = 'write', file = gnplt_arch, newunit = plt, status = 'replace')
  write (plt, '("unset border")')
  write (plt, '("set terminal qt size 1000,500")')
  write (plt, '("set zeroaxis lt -1")')
  write (plt, '("set autoscale yfix ")')
  write (plt, '("set autoscale xfix ")')
  write (plt, '("set key  bottom ")')
  write (plt, '("set grid ")')
  write (plt, '("set title ''",2a, "''" )') 'f(x) = ', funcion
  write (plt, '("f_rk = ''",a, "''")') datos_arch
  write (plt, '("f_x  = ''",a, "''")') datos_arch
  write (plt, '("plot f_rk using 1:2 w lp lw 0.9 ps 0.3  pt 15 lt rgb ''",a," '' ")') 'dark-red'
  write (plt, '("replot f_x using 1:3 w lp lw 0.9 ps 0.3  pt 15 lt rgb ''",a," '' ")') 'dark-blue'
  write (plt, '("replot ",a," w lp ps 0.01 lw 1.0 pt 15 lt rgb ''",a," '' ")') funcion, 'web-green'
  write (plt, '("replot 0 notitle lt rgb ''",a," '' ")') 'black'
  close(plt)
  call execute_command_line('gnuplot -p ' // gnplt_arch) ! linux
end subroutine gnu_plot
function f(y,t) ! y' =
    implicit none
    real(8)::f,y,t,pi

    f = 5*t + exp(-t) - t*t*log(y)
end function
end program runge_4ktt
```

Conclusiones

Alexia Sofía Ibarra García: Es interesante notar como los métodos numéricos cambian los resultados incluso aunque sean para un mismo método. Fue muy notorio con estos métodos para resolver ecuaciones diferenciales, pues con Euler y Euler modificado se necesitaban muchos términos de h para poder alcanzar un resultado al menos parecido al analítico, mientras que para Runge-Kutta para pocos términos se tienen resultados certeros.

Gustavo Hernández Ángeles: La verdad no entiendo cómo funciona el algoritmo de Runge-Kutta pero funciona sorprendentemente bien incluso para valores de h relativamente grandes. En esta práctica y a lo largo de estos días hemos podido observar cómo le da 3 vueltas en cuestión de optimización y resultados a los métodos de Euler y Euler modificado. Definitivamente es el método que estaré utilizando para resolver numéricamente las ecuaciones diferenciales. **Diego Alejandro Téllez Martínez:**
La resolución de ecuaciones diferenciales de forma numérica es una herramienta útil para encontrar la solución de un sistema dinámico representado de manera matemática.
La familia de técnicas del método Runge Kutta nos permite elegir la relación de tiempo de cómputo y error de aproximación que se requiere. Mientras se quiera un error pequeño, es necesario utilizar una técnica de orden mayor, aumentando el número de ecuaciones y, a su vez, aumentando el tiempo de computadora para resolver el algoritmo, pero dicho método llega a tener una muy buena aproximación para pocas iteraciones, lo cual es una ventaja sobre otros métodos.

Cristian Joel Gallegos Yañez: Como vimos al resolver estas ecuaciones diferenciales, este método es más adecuado ya que con menos términos obtenemos muy buenas precisiones en relación a los cálculos desarrollados a diferencia de Euler e inclusive EulerModificado lo cual nos brinda mayor poder en caso de necesitarlo.