Informe de la Tarea Investigativa II, presentado por los estudiantes del equipo No. 30

Francisco Vicente Suárez Bellón C-212
Max Bengochea C-211

December 2022

Título: PERIODIC SOLUTIONS FOR SMALL AND LARGE DELAYS IN A TUMOR-IMMUNE SYSTEM MODEL

Autores del articulo:
RADOUANE YAFIA

Revista:
Electronic Journal of Differential Equations

Año:
2006

Impacto: 1.282

1 Valoración del artículo

Breve explicación

En el artículo a analizado el autor investiga mediante un modelo matemático el comportamiento de un tumor y la respuesta del sistema inmune del individuo. La explicación científica-medica no es un argumento a analizar en dicho paper dado que no es su objetivo de estudio aunque si deja referenciado en que otros estudios y publicaciones se basa dicho modelo

Problemática que se propone resolver

El artículo explica que el modelo matemático estudiado de otra publicación y su modificación tiene una coincidencia lógica con los comportamientos biológicos del individuo, así como la no existencia de soluciones negativas o los puntos de estabilidad del mismo

Breve introducción

El modelo matemático que refleja la situación esta constituido por un sistema de 2 ecuaciones diferenciales (2.1) , donde "w" describe la respuesta inmune ante la aparición del tumor y "t" refleja el tiempo que demora el sistema inmune en proporcionar una respuesta a

2 Introducción al problema a analizar

En este articulo se hace referencia a un modelo de tumor inmune que expresa su comportamiento mediante ecuaciones diferenciales en este se centran principalmente en su análisis de estabilidad y su estudio de soluciones periódicas. Para ello se hace uso de la teoría de la estabilidad de Lyapunov. El sistema el cual

analizan en profundidad es un sistema de ecuaciones diferenciales no ordinarias las cuales se salen del objeto de este trabajo las cuales son:

Ecuaciones no ordinarias

$$\frac{dx}{dt} = \sigma + \omega x(t-\tau)y(t-\tau) - \delta x$$
$$\frac{dy}{dt} = \alpha y(1-\beta y) - xy$$

Sin embargo también nos proporcionan otro sistema de ecuaciones diferenciales ordinarias las cuales son el objeto de estudio en este trabajo

Sistema a trabajar

$$\frac{dx}{dt} = \sigma + \omega xy - \delta x$$
$$\frac{dy}{dt} = \alpha y(1-\beta y) - xy \tag{2.1}$$

3 Metodología

Para poder analizar este sistema no lineal de ecuaciones diferenciales ordinarias usaremos los recursos numéricos como Runge-Kutta 4 y el método de Euler Modificado además de utilizar recursos computacionales para ilustrar los resultados obtenidos mediante gráficas.

Comenzamos buscando los posibles puntos de equilibrio de este sistema de ecuaciones diferenciales ordinarias para ello buscamos para que valores ambas ecuaciones se hacen cero simultáneamente luego para obtener su análisis de estabilidad usaremos el teorema de Hartman–Grobman

Para aplicar las hipótesis del teorema antes descrito es necesario "linealizar" el sistema dado que este no es lineal por lo que se hace uso de la siguiente matriz de jacobiano del sistema respecto a x y y respectivamente

Matriz de jacobiano

$$\begin{pmatrix} y\omega - \delta & x\omega \\ -y & -x - y\alpha\beta + \alpha(-y\beta + 1) \end{pmatrix}$$

Nos centraremos en el análisis en el punto de equilibrio $(0, 0)$ dado que en dicho punto la matriz se convierte en una matriz diagonal lo cual facilita mucho el análisis de estabilidad de este punto de equilibrio.

Matriz jacobina en (0,0)

$$\begin{pmatrix} -\delta & 0 \\ 0 & \alpha \end{pmatrix}$$

Análisis de estabilidad:
TEOREMA 2 Estabilidad de sistemas casi lineales

Sean $\lambda 1$ y $\lambda 2$ los eigenvalues de la matriz de coeficientes del sistema lineal dado asociado con el sistema casi lineal. Entonces:

1. Si $\lambda 1 = \lambda 2$ son eigenvalues reales e iguales, por consiguiente el punto crítico (0, 0) es un nodo o un punto espiral, y es asintóticamente estable si $\lambda 1 = \lambda 2 < 0$, e inestable si $\lambda 1 = \lambda 2 > 0$.
2. Si $\lambda 1$ y $\lambda 2$ son imaginarios puros, entonces (0, 0) es un centro o un punto espiral, y puede ser asintóticamente estable, estable o inestable.
3. En caso contrario —esto es, que $\lambda 1$ y $\lambda 2$ no sean reales e iguales o imaginarios puros—, el punto crítico (0, 0) del sistema casi lineal es del mismo tipo y con la misma estabilidad que el punto crítico (0, 0) del sistema lineal asociado

4 Resultados

El tipo de estabilidad del punto (0,0) dependerá de los valores de: δ y α

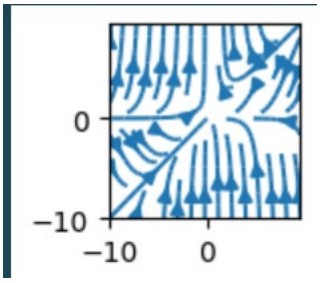
4.1 δ=1 y α=-1

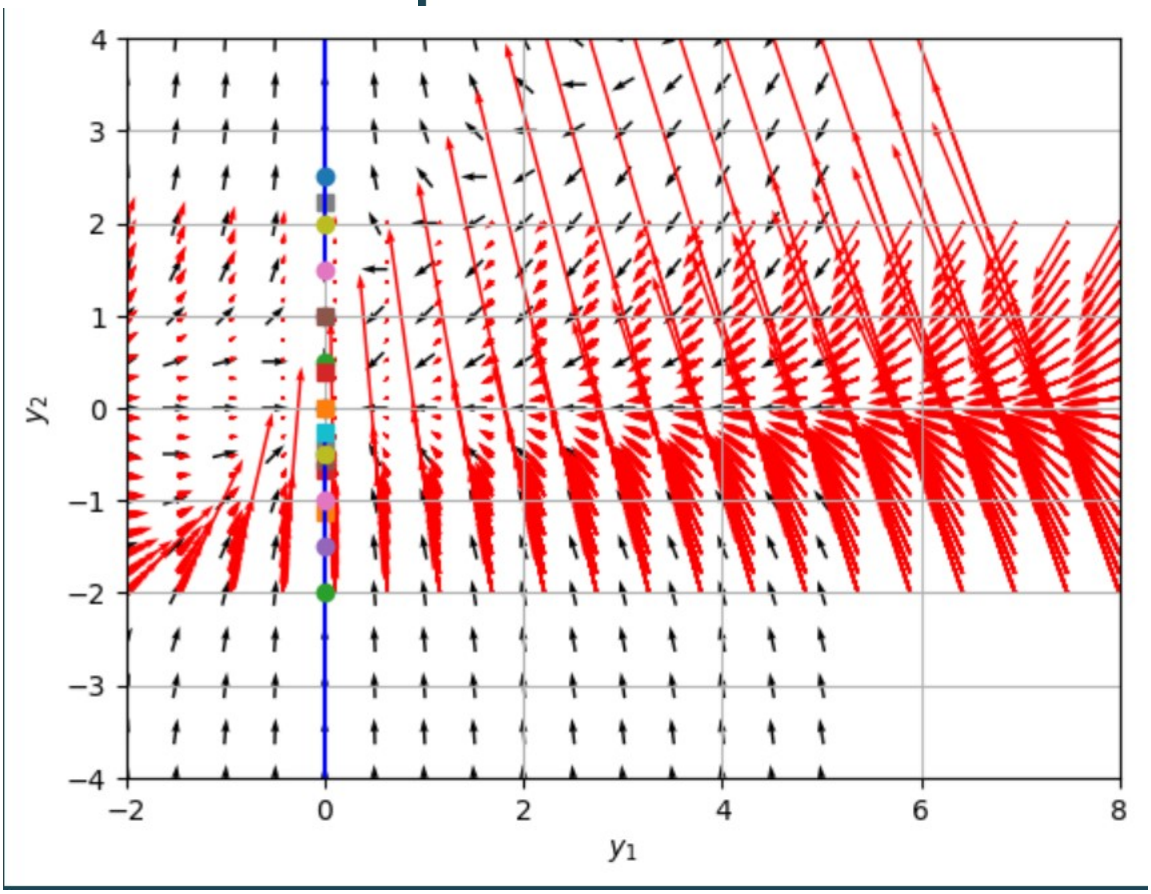
La matriz del jacobiano :

$$\begin{pmatrix} -1 & 0 \\ 0 & -1 \end{pmatrix}$$

Con lo que es un nodo o punto espiral y asintóticamente estable

Planos Fase del punto





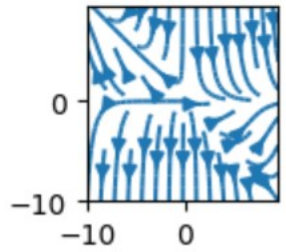
δ=-1 y α=1

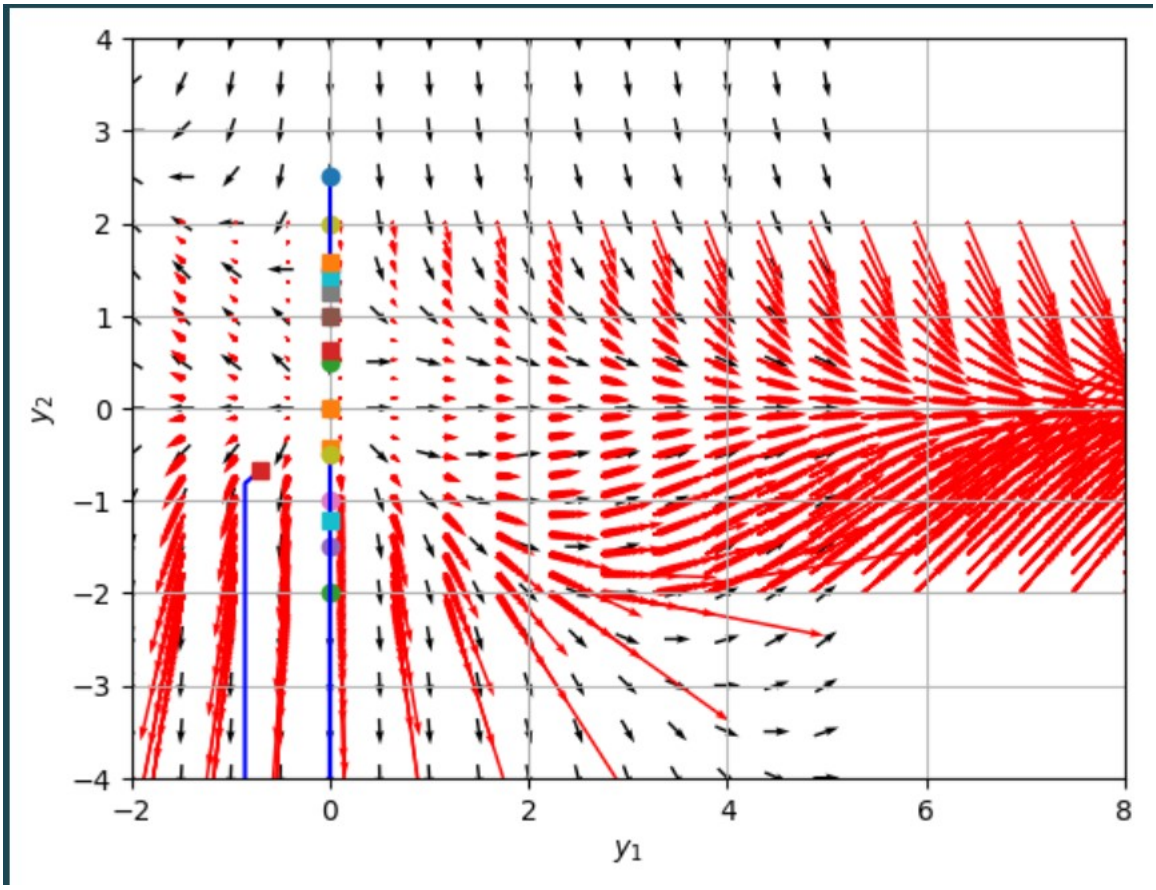
El tipo de estabilidad del punto (0,0) dependerá de los valores de: δ y α La matriz del jacobiano quedaría

$$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

Con lo que es un nodo o punto espiral y asintóticamente inestable

Planos Fase del punto





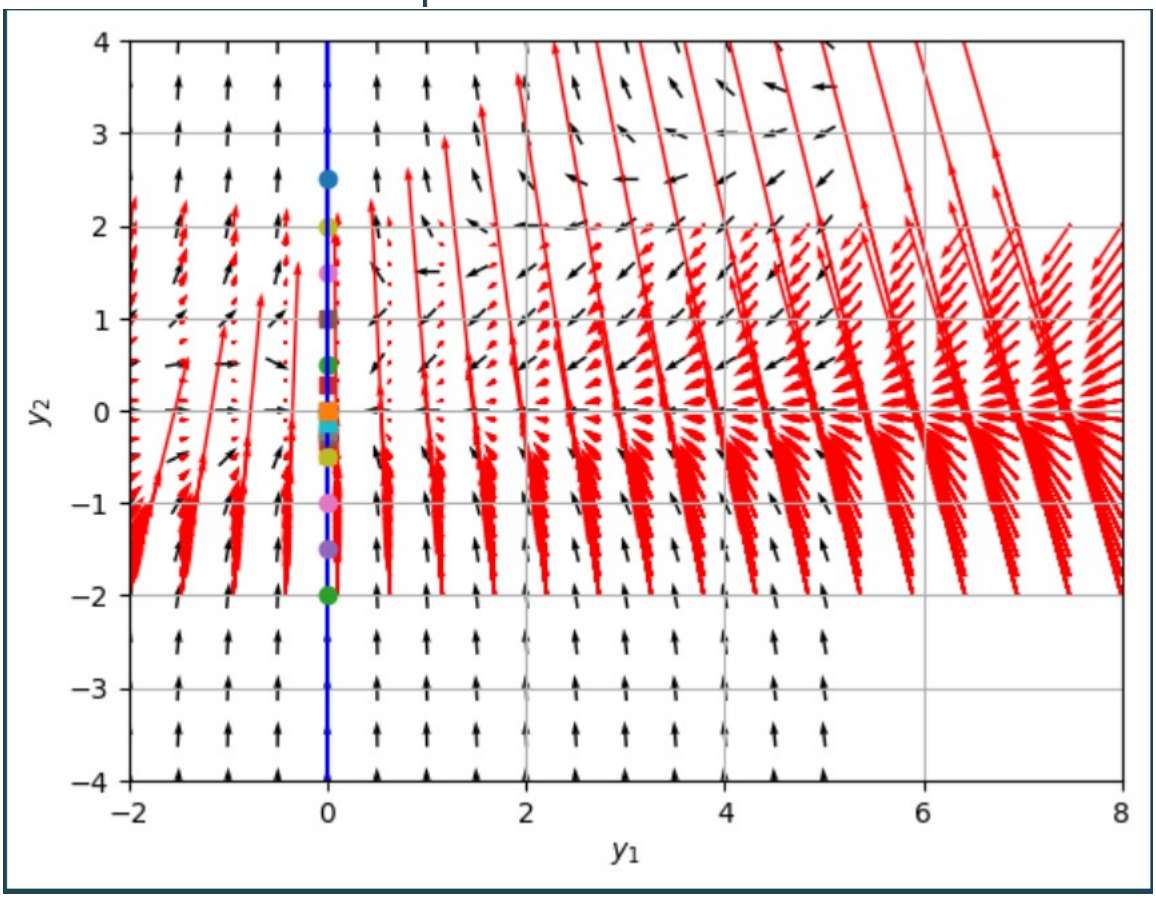
δ=1 y α=-2

La matriz del jacobiano quedaría

$$\begin{pmatrix} -1 & 0 \\ 0 & -2 \end{pmatrix}$$

Con lo que es un nodo impropio estable

Planos Fase del punto



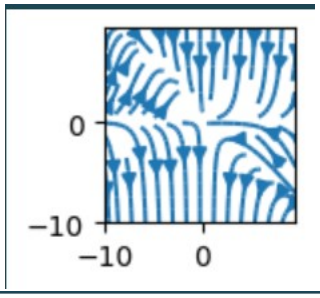
δ=1 y α=1

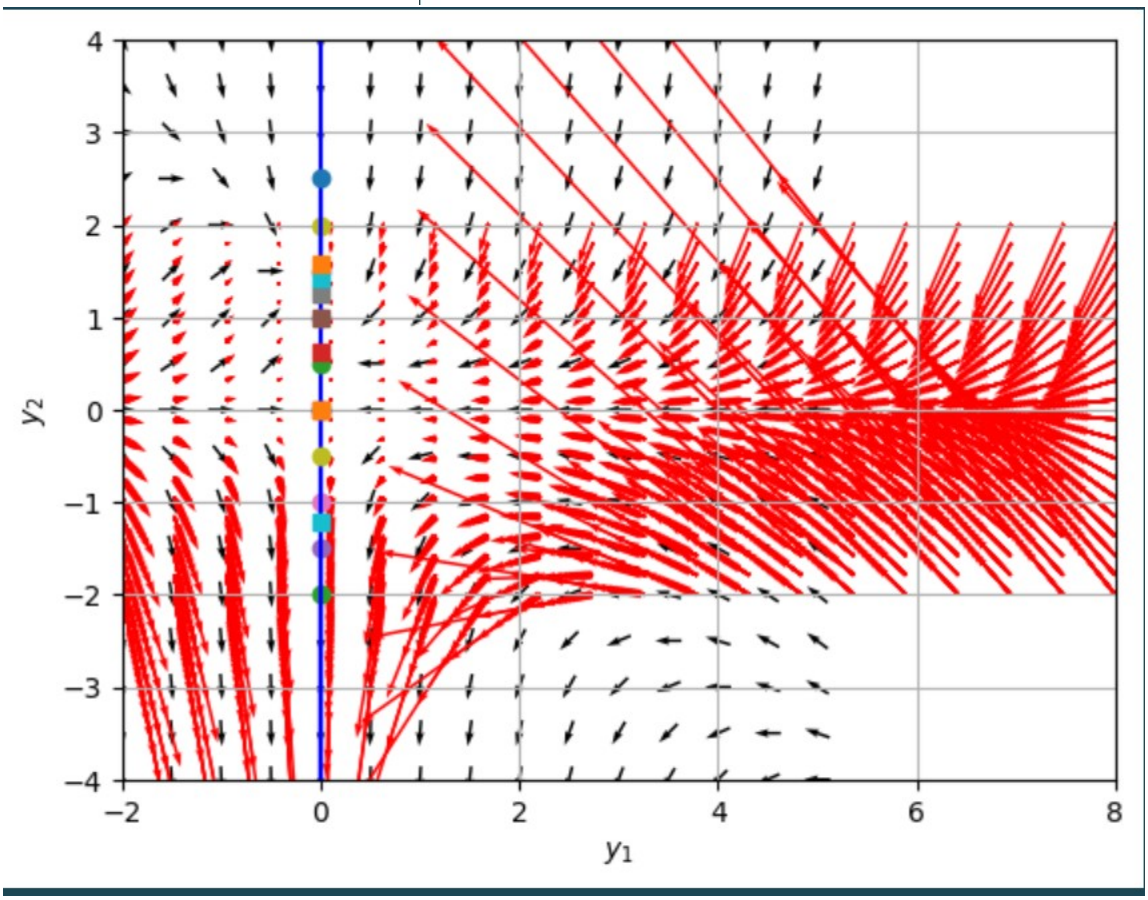
La matriz del jacobiano quedaría

$$\begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix}$$

Punto silla inestable

Planos Fase del punto





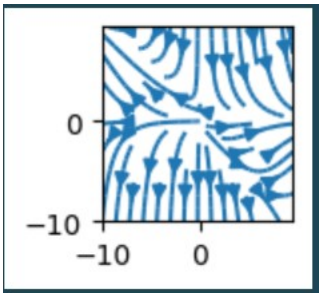
δ=1 y α=1

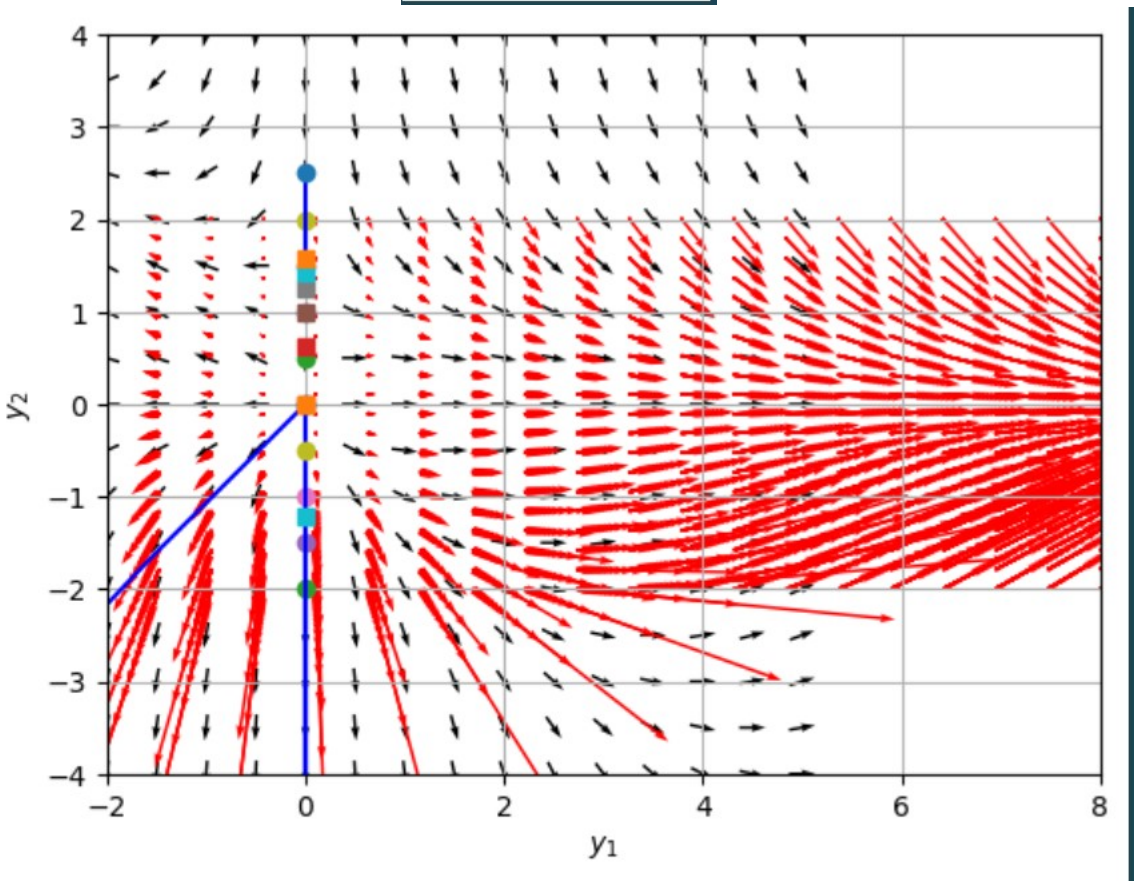
La matriz del jacobiano quedaría

$$\begin{pmatrix} 2 & 0 \\ 0 & 1 \end{pmatrix}$$

Nodo impropio inestable

Planos Fase del punto





Valores dados por métodos numéricos

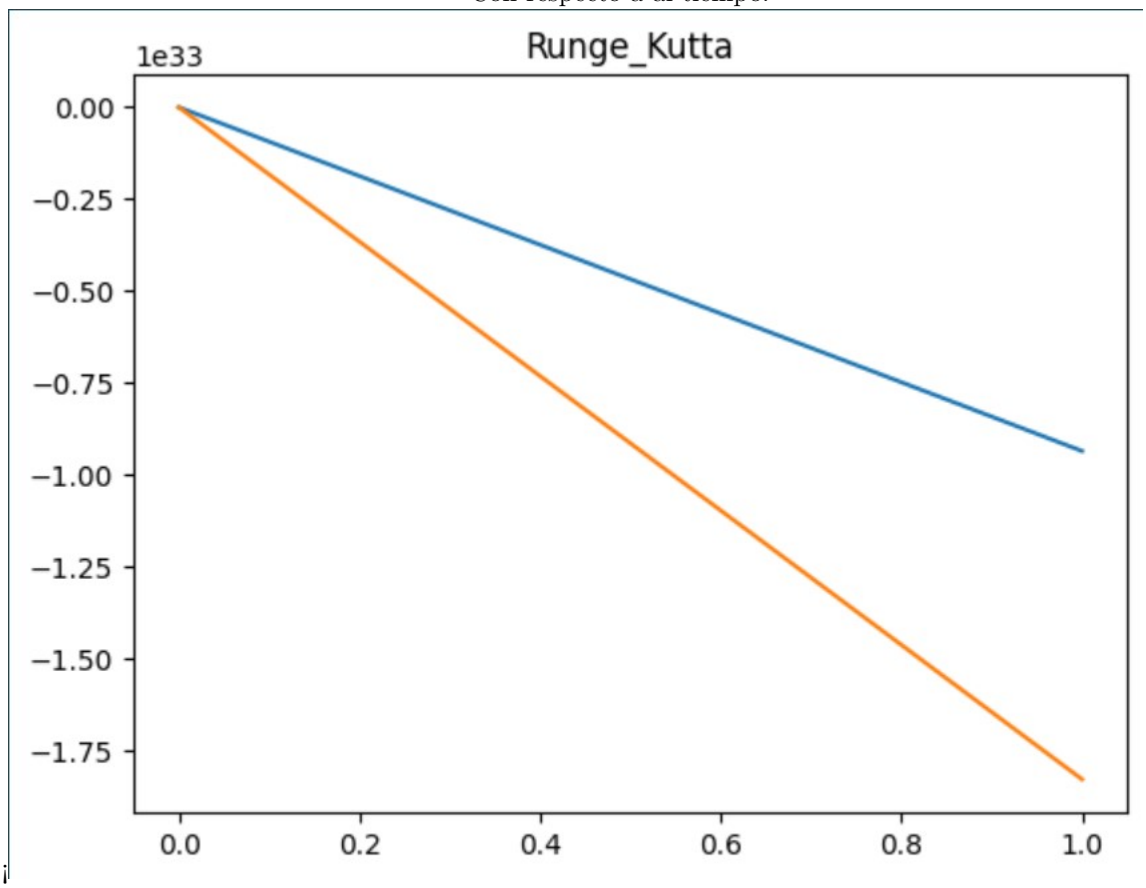
Se mostrarán los resultados numéricos del sistema con valores $y = 2x$ con $x = 100$ un tamaño de paso $h = 1$ iniciado en 0 y finalizado en 10. Se compararan datos tanto de Runge-Kutta como de Euler Mejorado

Runge-Kutta

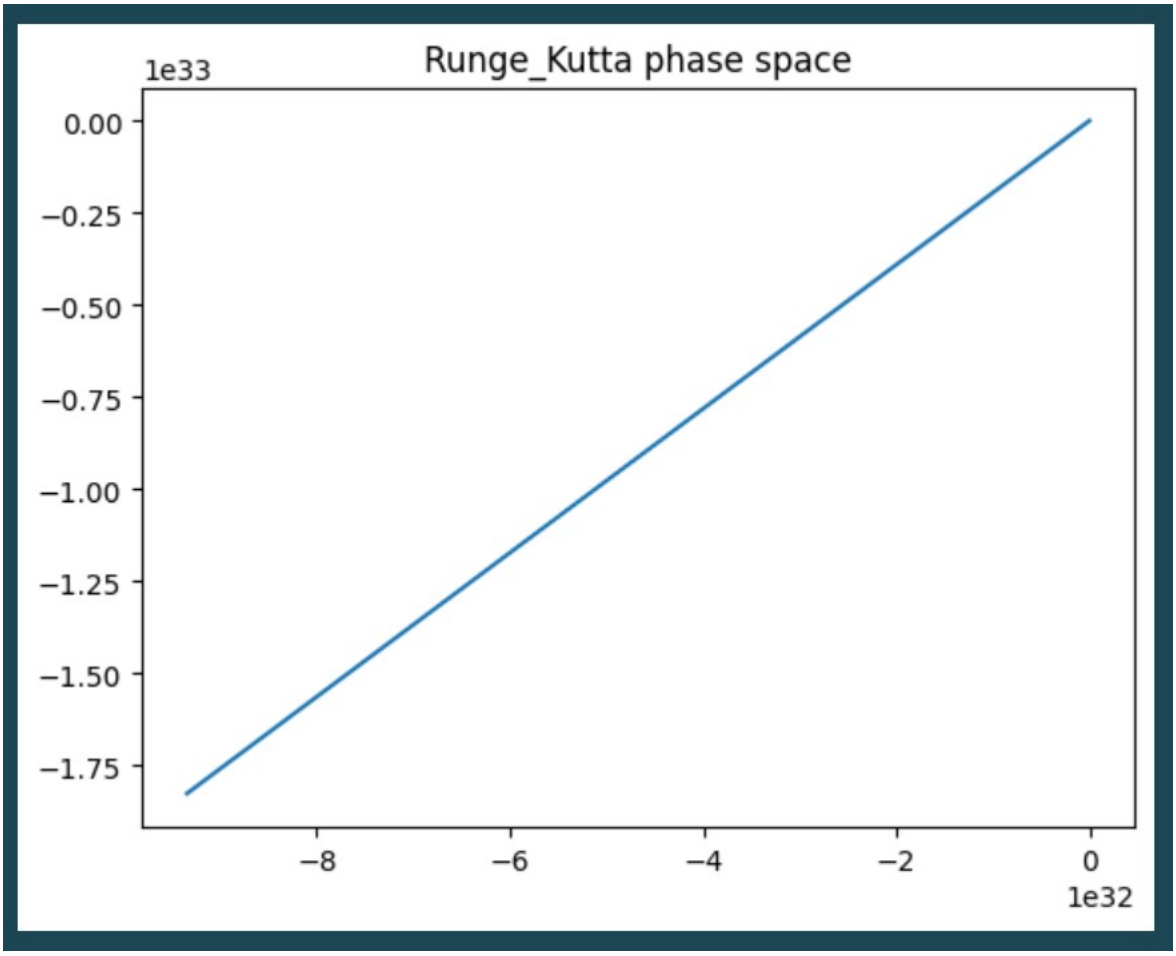
```
[ 1.00000000e+02 -9.34283984e+32            -inf             nan
             nan             nan             nan             nan
             nan             nan             nan]
x values
[ 2.00000000e+02 -1.82662843e+33             nan             nan
             nan             nan             nan             nan
             nan             nan             nan]
y values
```

Gráficas:

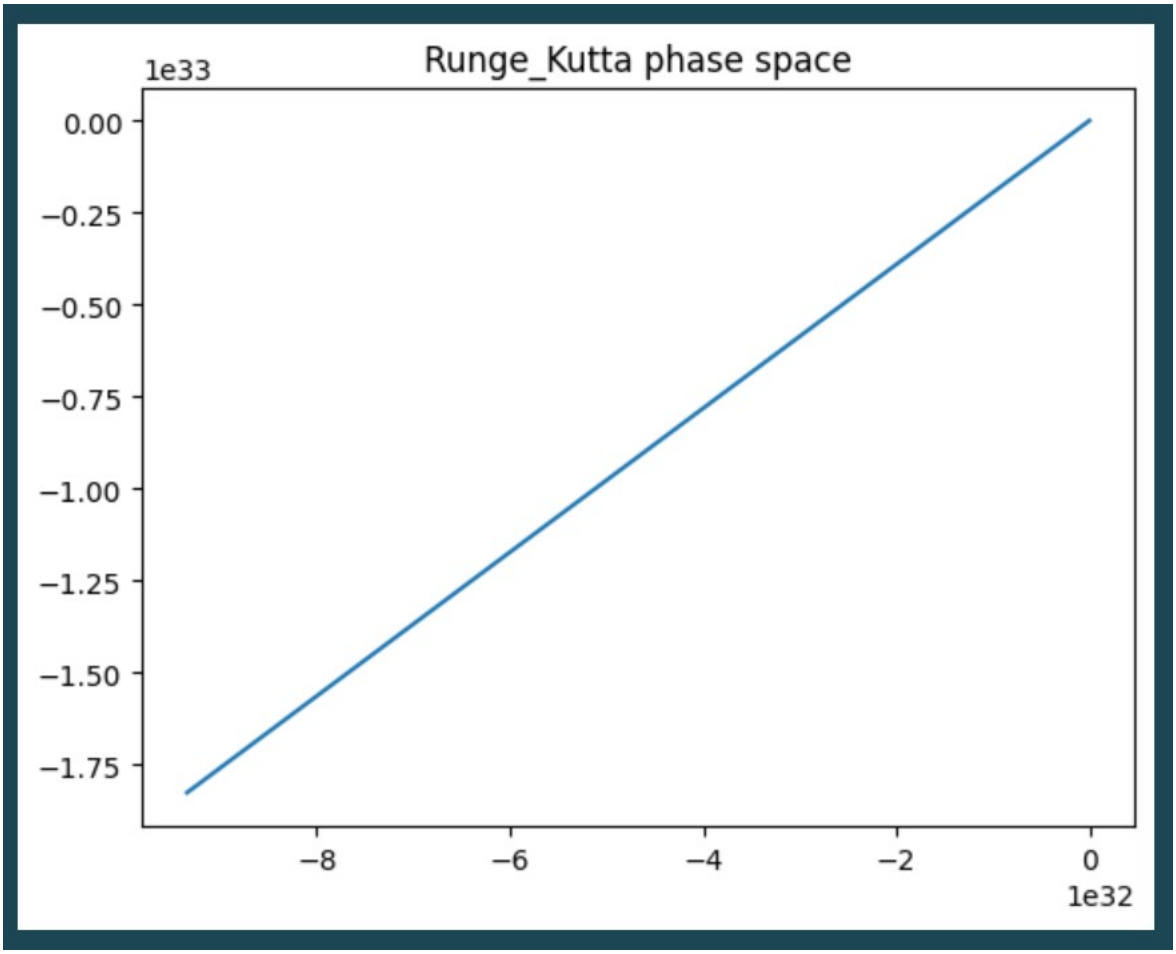
Con respecto a al tiempo:



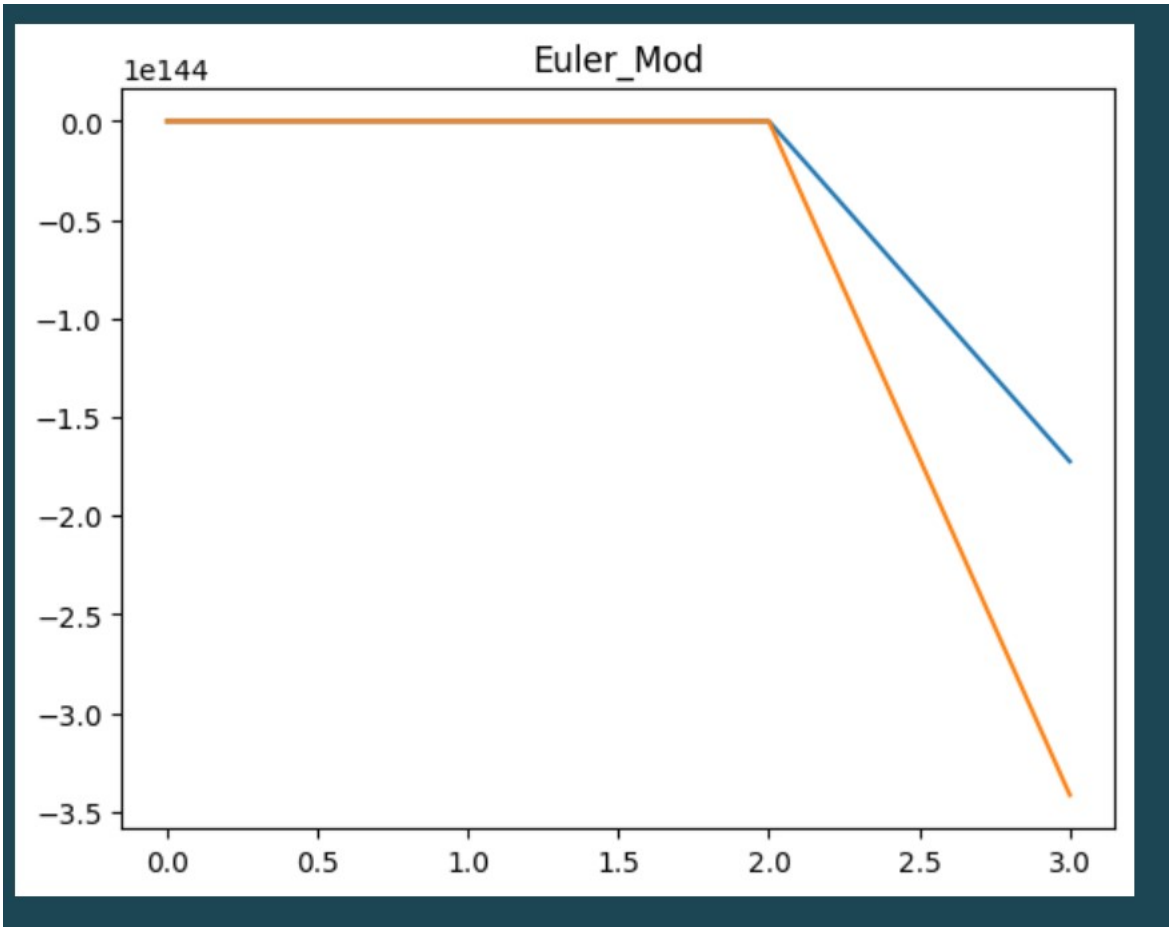
¡ Con las funciones una respecto a la otra:


Runge_Kutta phase space
1e33
0.00
−0.25
−0.50
−0.75
−1.00
−1.25
−1.50
−1.75
−8
−6
−4
−2
0
1e32

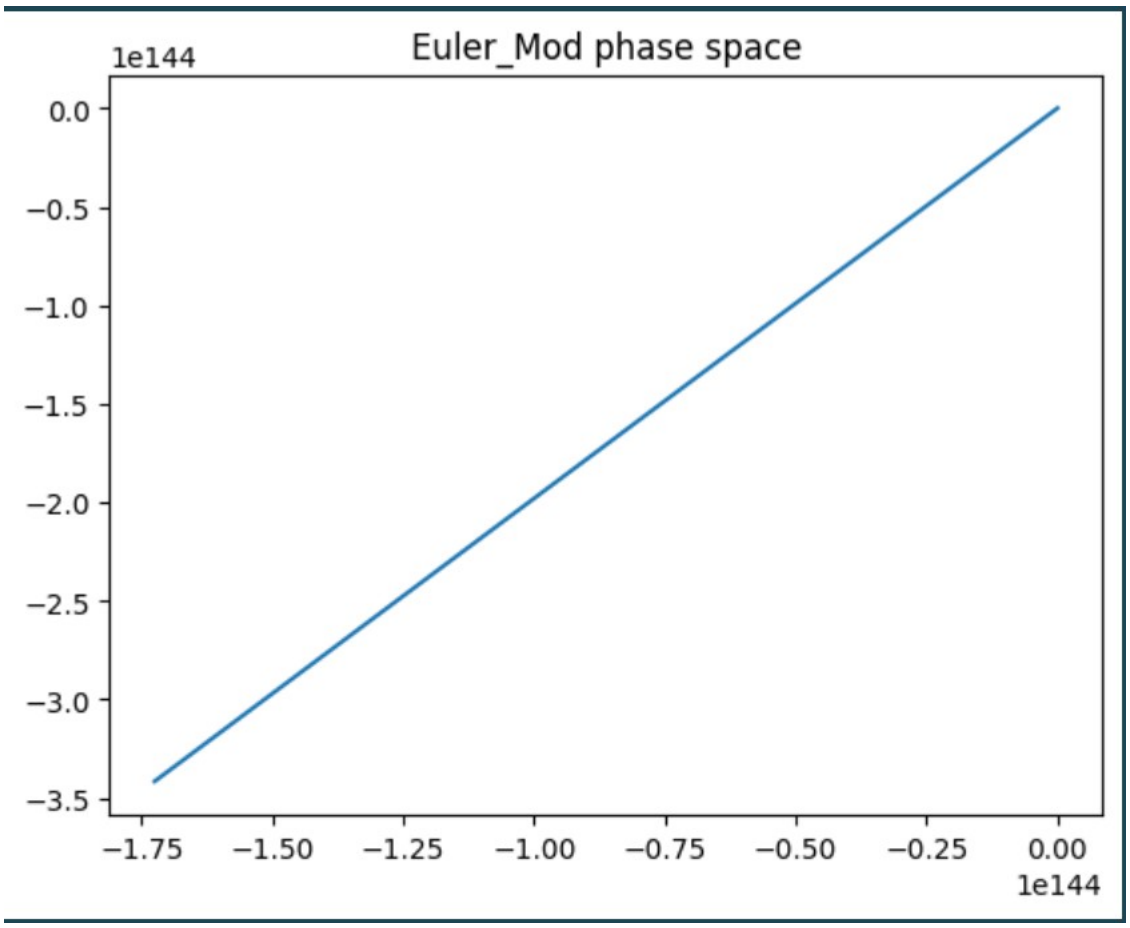
Euler Mejorado



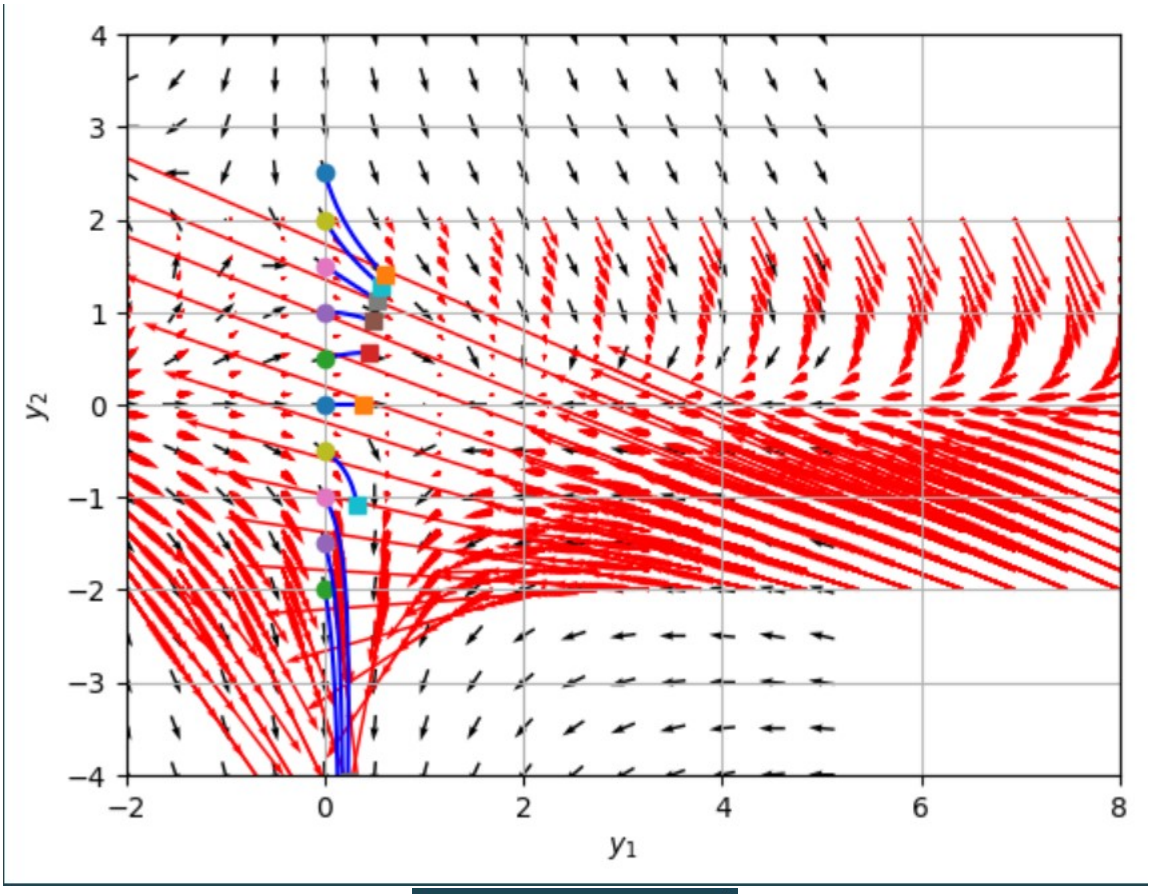
Gráficas:
Con respecto a al tiempo:

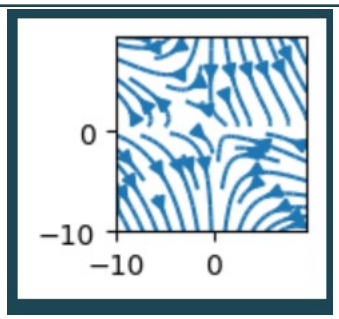


Con las funciones una respecto a la otra:


Euler_Mod phase space
1e144
0.0
−0.5
−1.0
−1.5
−2.0
−2.5
−3.0
−3.5
−1.75
−1.50
−1.25
−1.00
−0.75
−0.50
−0.25
0.00
1e144

Plano Fase





Análisis

En los resultados anteriores se evidencia que para números de esa magnitud las soluciones son más grandes que la aritmética de la computadora y además se ve que dado la naturaleza de los métodos numéricos Euler Mejorado nos permite aunque perdiendo precisión tener otro dato más

Anexos de códigos

Métodos Numéricos

Runge-Kutta 4

```
def runge_kutta_system(f, g, x0, y0, a, b, h):
  t = np.arange(a, b + h, h)
  n = len(t)
  x = np.zeros(n)
  y = np.zeros(n)
  x[0] = x0
  y[0] = y0
  for i in range(n - 1):
      k1 = h * f(x[i], y[i], t[i])
      l1 = h * g(x[i], y[i], t[i])
      k2 = h * f(x[i] + (k1 / 2), y[i] + (l1 / 2), t[i] + (h / 2))
      l2 = h * g(x[i] + (k1 / 2), y[i] + (l1 / 2), t[i] + (h / 2))
      k3 = h * f(x[i] + (k2 / 2), y[i] + (l2 / 2), t[i] + (h / 2))
      l3 = h * g(x[i] + (k2 / 2), y[i] + (l2 / 2), t[i] + (h / 2))
      k4 = h * f(x[i] + k3, y[i] + l3, t[i] + h)
      l4 = h * g(x[i] + k3, y[i] + l3, t[i] + h)
      x[i + 1] = x[i] + (1 / 6) * (k1 + (2 * k2) + (2 * k3) + k4)
      y[i + 1] = y[i] + (1 / 6) * (l1 + (2 * l2) + (2 * l3) + l4)

  return t, x, y
```

Euler Mejorado

```
def Euler_Mod(f, g, x0, y0, a, b, h):
  t = np.arange(a, b+h, h)
  n = len(t)
  x = np.zeros(n)
  y = np.zeros(n)
  x[0] = x0
  y[0] = y0
  for i in range(n - 1):
      k1 = h * f(x[i], y[i], t[i])
      l1 = h * g(x[i], y[i], t[i])
      k2 = h * f(x[i]+k1, y[i]+l1, t[i]+(h/2))
      l2 = h * g(x[i]+k1, y[i]+l1, t[i]+(h/2))
      x[i+1] = x[i]+((1/2)*(k1+k2))
      y[i+1] = y[i]+((1/2)*(l1+l2))

  return t, x, y
```

Gráficas

```
import matplotlib.pyplot as plt
from scipy.integrate import odeint
import numpy as np
```

Gráficas de métodos numéricos

```
def plot(t, x, y, title):
plt.plot(t, x, t, y)

plt.title(title)

plt.show()
plt.close()

plt.plot(x, y)
plt.title(str(title))
```

```
plt.show()
plt.plot(t,x,label="x")
plt.plot(t,y,label="y")
plt.legend(loc="best")
```

Diagrama de fases

```
def plotdf(f, xran=[-5, 5], yran=[-5, 5], grid=[21, 21], color='k'):
"""
Plot the direction field for an ODE written in the form
    x' = F(x,y)
    y' = G(x,y)

The functions F,G are defined in the list of strings f.

Input
-----
f:    list of strings ["F(X,Y)", "G(X,Y)"
      F,G are functions of X and Y (capitals).
xran: list [xmin, xmax] (optional)
yran: list [ymin, ymax] (optional)
grid: list [npoints_x, npoints_y] (optional)
      Defines the number of points in the x-y grid.
color: string (optional)
      Color for the vector field (as color defined in matplotlib)
"""
x = np.linspace(xran[0], xran[1], grid[0])
y = np.linspace(yran[0], yran[1], grid[1])
def dX_dt(X, Y, t=0): return map(eval, f)

X, Y = np.meshgrid(x, y)  # create a grid
DX, DY = dX_dt(X, Y)        # compute growth rate on the grid
M = (np.hypot(DX, DY))      # Norm of the growth rate
M[M == 0] = 1.             # Avoid zero division errors
DX = DX/M                   # Normalize each arrows
DY = DY/M

plt.quiver(X, Y, DX, DY, pivot='mid', color=color)
plt.xlim(xran), plt.ylim(yran)
plt.grid('on')
```

Isoclinas

```
def Isoclinas(f, points=[-2-5, -2, -1.5, -1, -0.5,  0, 0.5, 1, 1.5, 2, 2.5]):

  for y20 in points:
      tspan = np.linspace(0, 0.5, 20)
      y0 = [0.0, y20]
      ys = odeint(f, y0, tspan)

      plt.plot(ys[:, 0], ys[:, 1], 'b-')  # path
      plt.plot([ys[0, 0]], [ys[0, 1]], 'o')  # start
      plt.plot([ys[-1, 0]], [ys[-1, 1]], 's')  # end

  plt.xlim([-2, 8])

  plt.show()
```