Tarea 13

CIMAT

Nombre:	**Jairo Saul Diaz Soto**
Maestría:	Ciencias Computacionales
Modulo:	Métodos Numéricos
Instructor:	Dr. Luis Daniel Blanco Cocom
Fecha de entrega:	2023 - 11 - 19

1. Métodos básicos para resolver ecuaciones diferenciales

Otro tipo de problemática a la que se enfrenta mucho en la practica dentro de muchas áreas de trabajo es a las ecuaciones diferenciales, o bien en su defecto, a los sistemas de ecuaciones diferenciales, en muchas ovaciones, donde no se tiene si quiera una respuesta sencilla o exacata desde métodos analíticos, es por eso que tambien se han desarrollado diferentes tecnicas para este tipo de problemas desde el enfoque numérico.

1.1. Método de Euler

Uno de los métodos más básicos que existen dentro de esta rama, es el método de Euler, el cuál, se puede decir, es de los más sencillos para resolver problemas de este tiupo con el enfoque numérico, sin emabrgo, la simplicidad de este propio, genera que no sea uno de los más precisos, pero es bueno para una aproximación inicial.

A continuación se presenta un algoritmo el cual implementa el método de Euler (Este es la generalización para sistemas de ecuacines diferenciales):

Algorithm 1: Método de Euler para ecuaciones diferenciales en pseudocódigo.

Data: Función f, vector x, paso h, tamaño del vector sz
Result: Nuevo vector solución

`euler`(f, x, h, sz) $fy \leftarrow f(x, sz)$;
for $i \leftarrow 0$ **to** $sz - 1$ **do**
  $sol[i] \leftarrow x[i+1] + (h \times fy[i])$;
end
Liberar memoria de fy;
return sol;

1.2. Método de Heun

Este método adopta el método de Euler, pero añade una mejora para generar tanto estabilidad como una mejor convergencia a la solución, siendo de esta forma un método más preciso para la tarea que represantan los problemas de valor inicial de ecuaciones diferenciales

Algorithm 2: Método de Heun para ecuaciones diferenciales en pseudocódigo.

Data: Función f, vector x, paso h, tamaño del vector sz
Result: Nuevo vector solución

`heun(`f, x, h, sz`)` $xc \leftarrow$ nuevo vector de tamaño $sz + 1$ inicializado con ceros;
for $i \leftarrow 0$ **to** sz **do**
  $xc[i] \leftarrow x[i]$;
end
$yp \leftarrow$ nuevo vector de tamaño sz inicializado con ceros;
$fy \leftarrow f(xc, sz)$;
for $i \leftarrow 0$ **to** $sz - 1$ **do**
  $yp[i] \leftarrow x[i + 1] + (h \times fy[i])$;
end
$xc[0] \leftarrow xc[0] + h$;
for $i \leftarrow 0$ **to** $sz - 1$ **do**
  $xc[i + 1] \leftarrow yp[i]$;
end
$fn \leftarrow f(xc, sz)$;
for $i \leftarrow 0$ **to** $sz - 1$ **do**
  $yp[i] \leftarrow x[i + 1] + \frac{h}{2{,}0} \times (fy[i] + fn[i])$;
end
Liberar memoria de xc, fy, fn;
return yp;

1.3. Método de Taylor

Una aproximación que también resulta bastante útil a la hora de trabajar con este tipo de problemas es mediante los polinomios de Taylor, los cuales también mejoran la precisión a la hora de resolver este tipo de problemas, aunque no resulten tan estables. A continuación, se presenta un algoritmo el cual implementa un algoritmo de este método.

Algorithm 3: Método de Taylor de segundo orden para ecuaciones diferenciales en pseudocódigo.

Data: Función f, vector x, paso h, tamaño del vector sz
Result: Nuevo vector solución

`taylor_2(`f, x, h, sz`)` $g \leftarrow$ matriz de derivadas parciales de f en x con respecto a cada variable;
$yp \leftarrow$ nuevo vector de tamaño sz inicializado con ceros;
$fy \leftarrow f(x, sz)$;
for $i \leftarrow 0$ **to** $sz - 1$ **do**
    $yp[i] \leftarrow x[i+1] + (h \times fy[i])$;
end
$sol \leftarrow$ nuevo vector de tamaño sz inicializado con ceros;
$temp \leftarrow$ nuevo vector de tamaño sz inicializado con ceros;
for $i \leftarrow 0$ **to** $sz - 1$ **do**
    $temp[i] \leftarrow g[i][0]$;
    for $j \leftarrow 0$ **to** $sz - 1$ **do**
        $temp[i] \leftarrow temp[i] + g[i][j+1] \times fy[j]$;
    end
    $sol[i] \leftarrow yp[i] + \frac{h^2}{2{,}0} \times temp[i]$;
end
Liberar memoria de $temp, g[0], g, yp, fy$;
return sol;

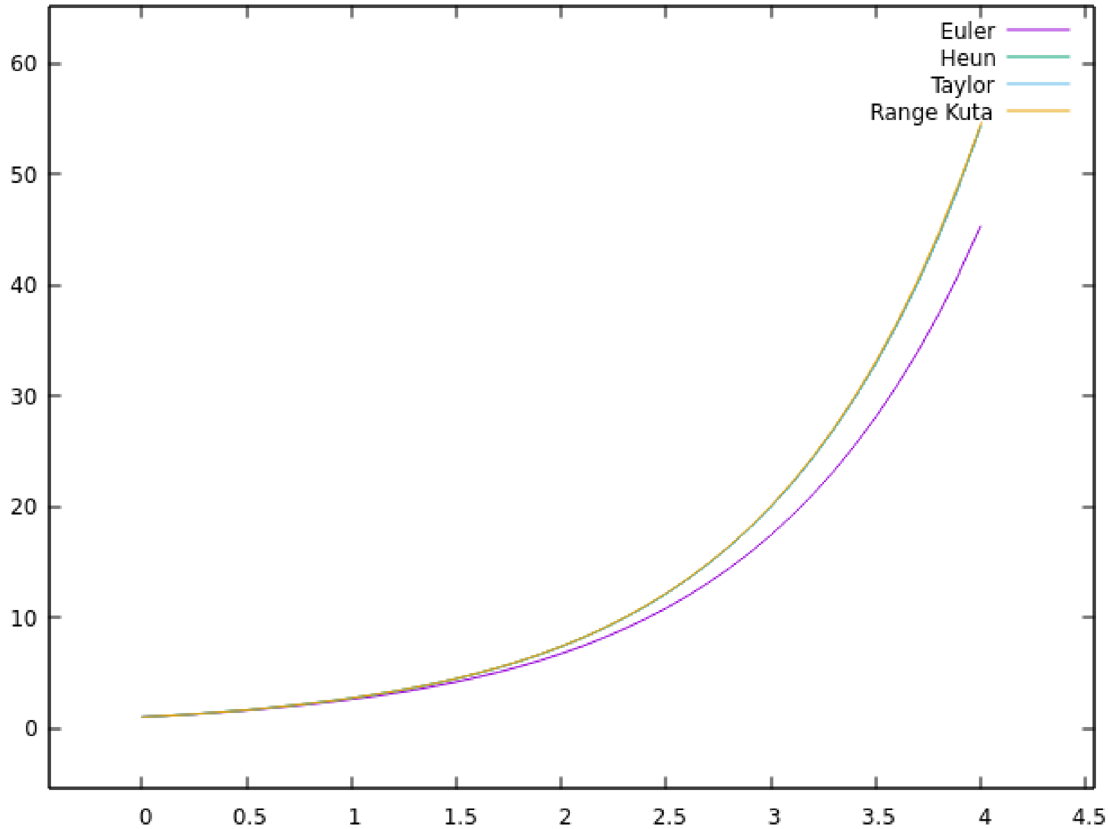
1.4. Método de Range Kuta

Uno de los métodos que mejora tanto la precisión así mismo como la estabilidad se basa en el método de Range Kuta, a continuación se presenta uno de los métodos de Range Kuta

Algorithm 4: Método de Runge-Kutta de cuarto orden (RK4) para ecuaciones diferenciales en pseudocódigo.

Data: Función f, vector x, paso h, tamaño del vector sz
Result: Nuevo vector solución

`rk_4(`f, x, h, sz`)` $xc \leftarrow$ nuevo vector de tamaño $sz + 1$ inicializado con ceros;
$k1 \leftarrow f(xc, sz)$;
$xc[0] \leftarrow xc[0] + \frac{h}{2{,}0}$;
for $i \leftarrow 0$ **to** $sz - 1$ **do**
  $xc[i+1] \leftarrow xc[i+1] + \frac{h}{2{,}0} \times k1[i]$;
end
$k2 \leftarrow f(xc, sz)$;
for $i \leftarrow 0$ **to** sz **do**
  $xc[i] \leftarrow x[i]$;
end
$xc[0] \leftarrow xc[0] + \frac{h}{2{,}0}$;
for $i \leftarrow 0$ **to** $sz - 1$ **do**
  $xc[i+1] \leftarrow xc[i+1] + \frac{h}{2{,}0} \times k2[i]$;
end
$k3 \leftarrow f(xc, sz)$;
for $i \leftarrow 0$ **to** sz **do**
  $xc[i] \leftarrow x[i]$;
end
$xc[0] \leftarrow xc[0] + h$;
for $i \leftarrow 0$ **to** $sz - 1$ **do**
  $xc[i+1] \leftarrow xc[i+1] + h \times k3[i]$;
end
$k4 \leftarrow f(xc, sz)$;
$sol \leftarrow$ nuevo vector de tamaño sz inicializado con ceros;
for $i \leftarrow 0$ **to** $sz - 1$ **do**
  $sol[i] \leftarrow x[i+1] + \frac{h}{6{,}0} \times (k1[i] + 2{,}0 \times k2[i] + 2{,}0 \times k3[i] + k4[i])$;
end
Liberar memoria de $xc, k1, k2, k3, k4$;
return sol;



!h

Figura 1: Prueba de los métodos para un PVI.

2. Resultados

A continuación se presenta resultados obtenidos al momento de aplicar estos métodos en diferentes tipos de problemas

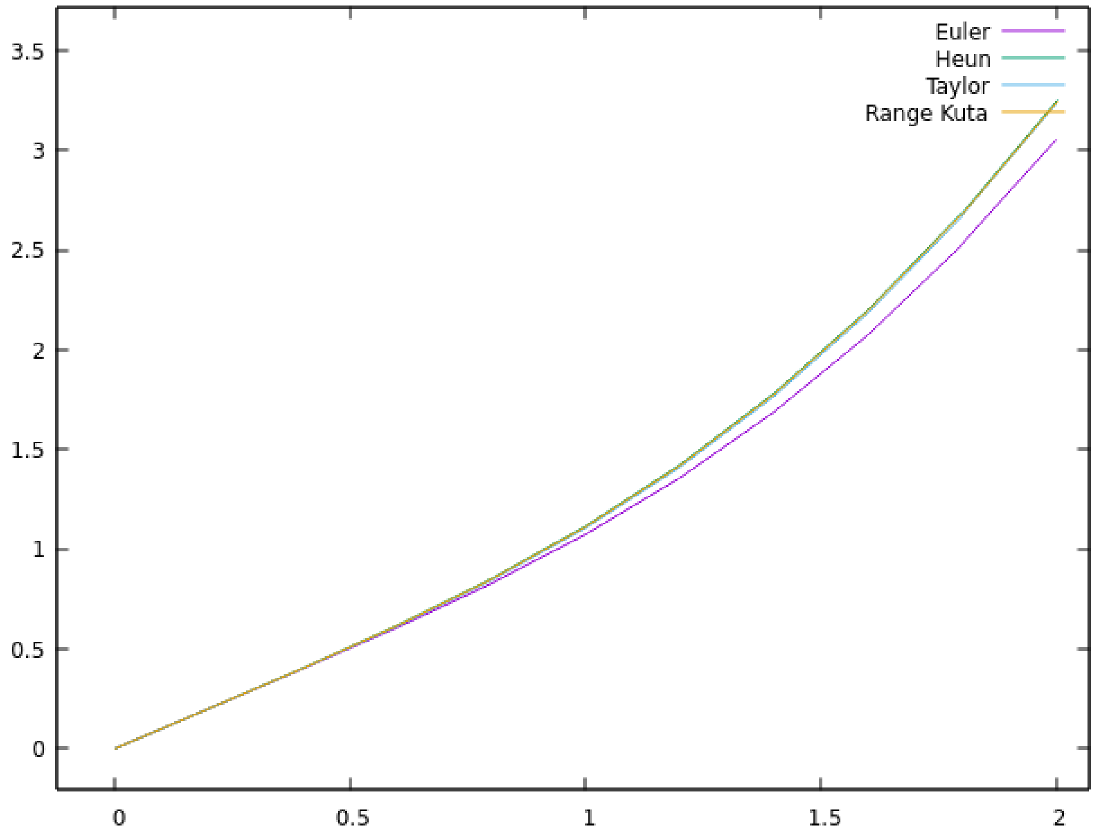


Figura 2: Aplicación de los métodos para resolver una ecuación derivada de una integral elíptica..

El valor de la arpoximacion de y(2)=3.241106

Tabla de aproximaciones por metodo

xi	yi Euler	yi Heun	yi T2	yi RK4
0.000000	0.000000	0.000000	0.000000	0.000000
0.200000	0.200000	0.200399	0.200000	0.200200
0.400000	0.400798	0.403949	0.401994	0.403171
0.600000	0.607099	0.617372	0.612948	0.615733
0.800000	0.827644	0.850607	0.843287	0.847996
1.000000	1.073571	1.114992	1.104828	1.111446
1.200000	1.356413	1.421580	1.408884	1.417210
1.400000	1.686747	1.780241	1.765372	1.775166
1.600000	2.073735	2.199478	2.182749	2.193798
1.800000	2.525221	2.686602	2.668257	2.680395
2.000000	3.047983	3.247983	3.228206	3.241307

Figura 3: Tabla comparativa de los métodos en la aplicación..

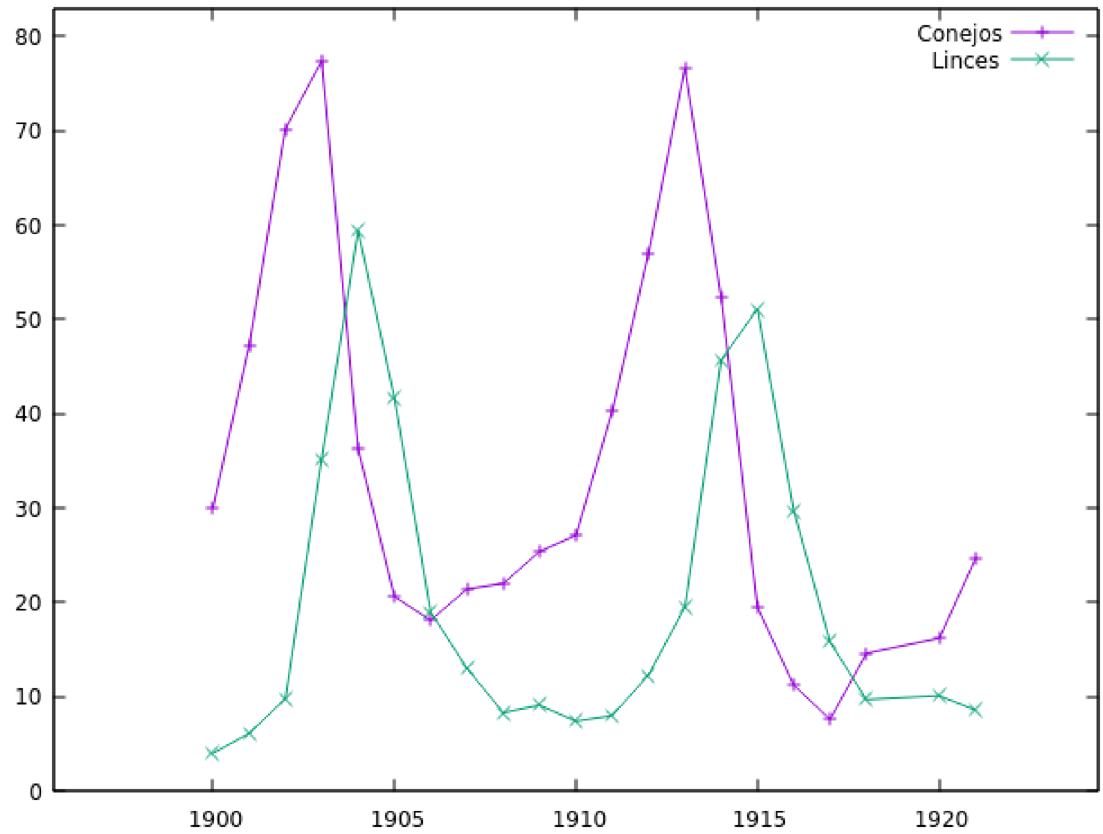


Figura 4: Datos experimentales de un modelo de Lotka - Volterra.

De las figuras relacionadas a la aplicación de los métods, se puede observar que dentreo de este intervalo y con tamaño de paso dado, los métodos son basatante similares, con la excepción del de Euler, el cual es el que siempre queda bastante retirado de los demás.

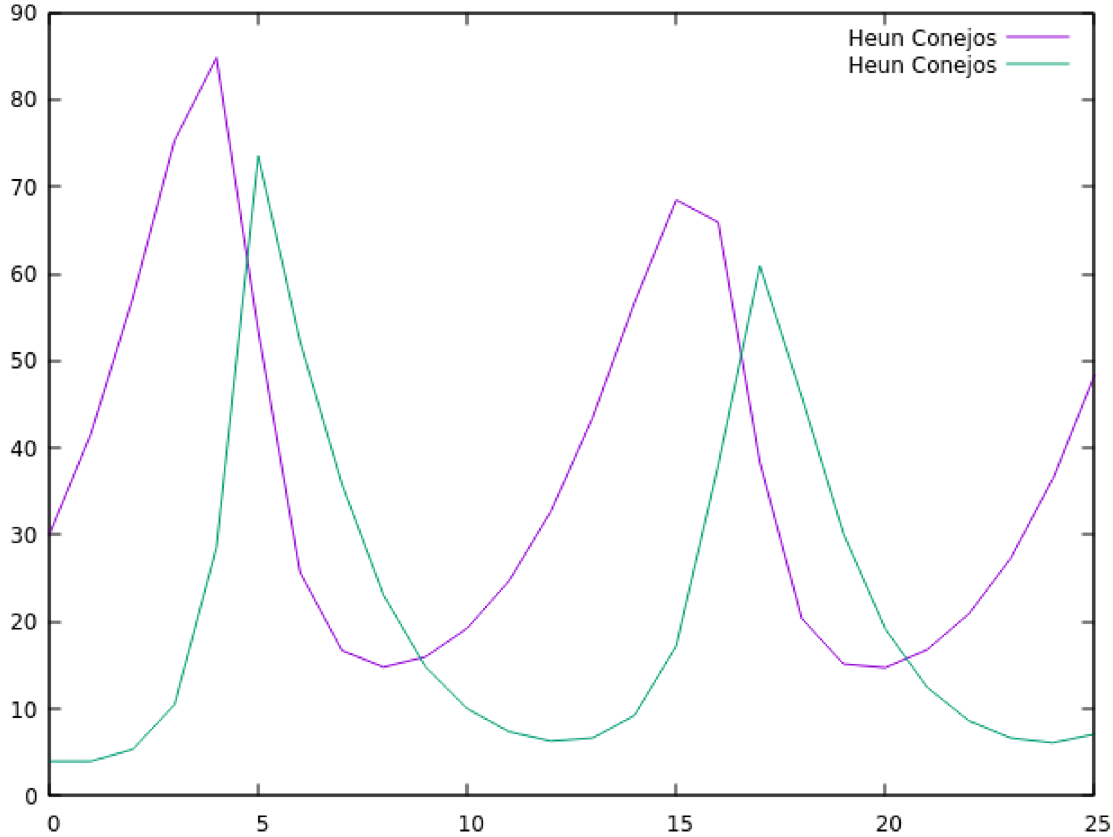


Figura 5: Aproximación del modelo de Lotka - Volterra por el método de Heun (paso 1 año).

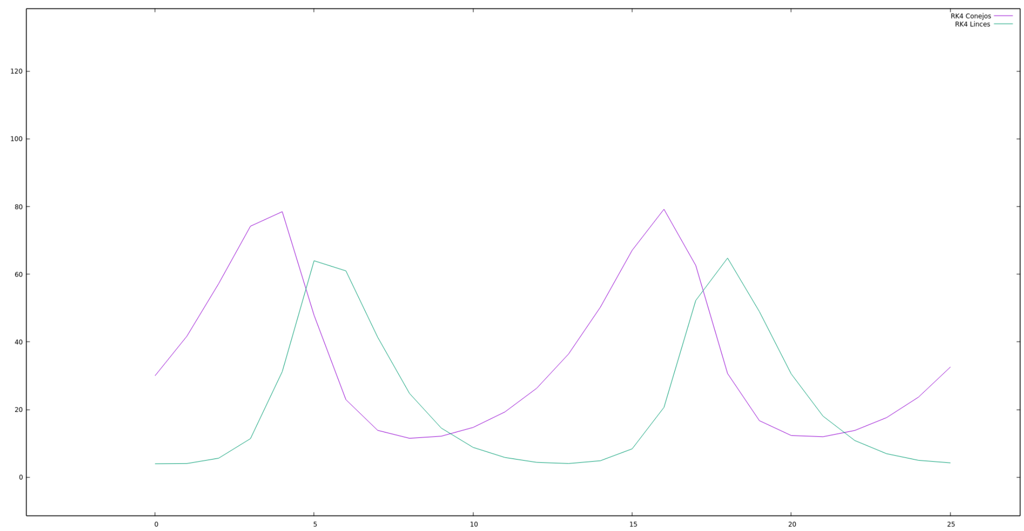


Figura 6: Aproximación del modelo de Lotka - Volterra por el método de Heun (paso 1 año).

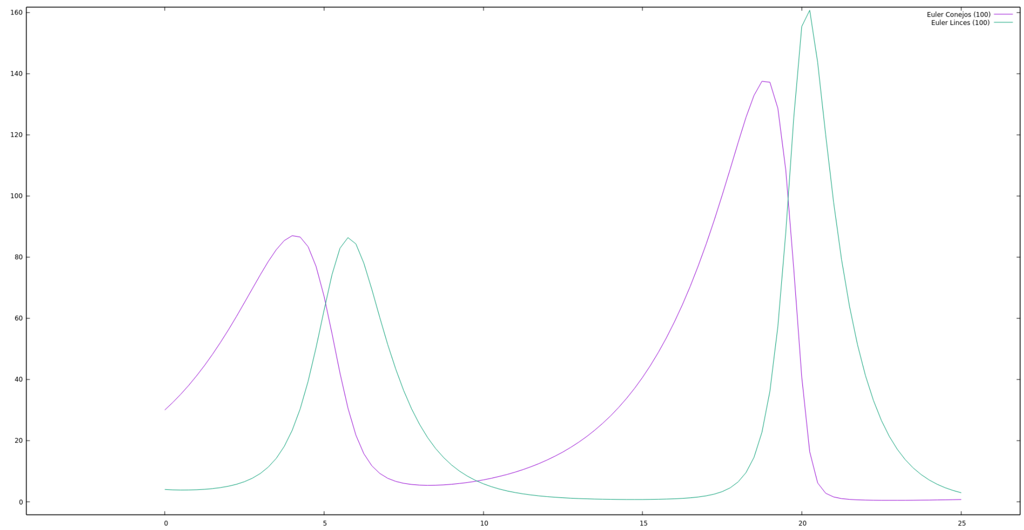


Figura 7: Aproximación del modelo de Lotka - Volterra por el método de Euler (paso 3 meses).

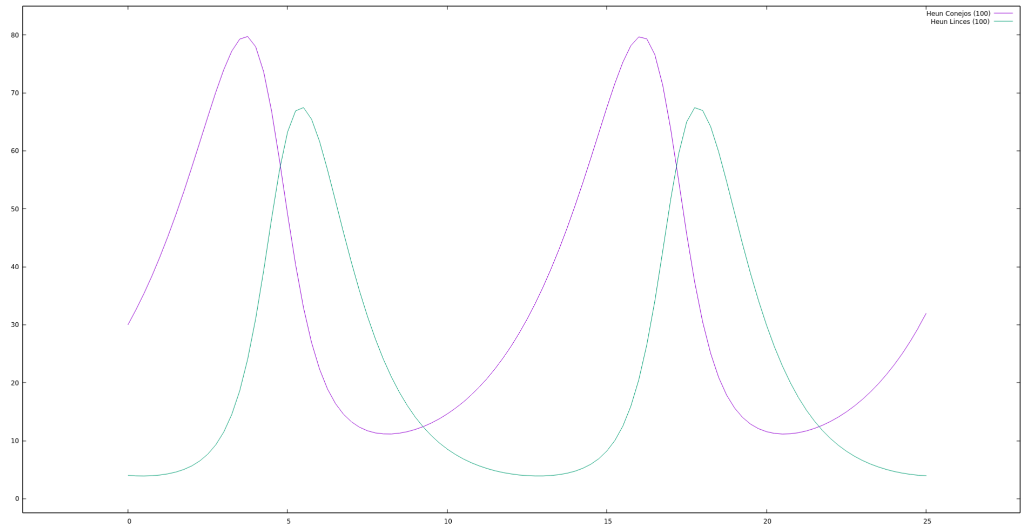


Figura 8: Aproximación del modelo de Lotka - Volterra por el método de Heun (paso 3 meses).

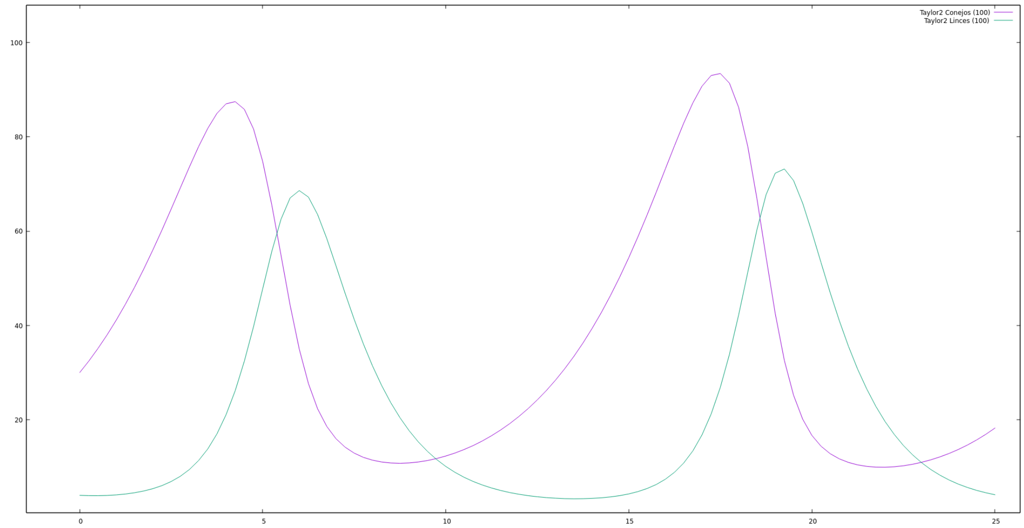


Figura 9: Aproximación del modelo de Lotka - Volterra por el método de Taylor segundo orden (paso 3 meses).

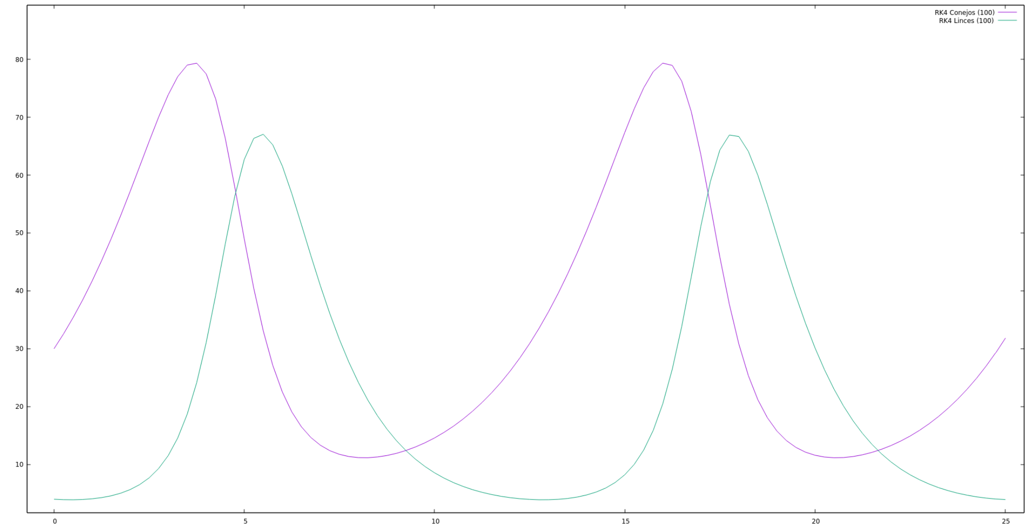


Figura 10: Aproximación del modelo de Lotka - Volterra por el método de Range - Kuta 4 (paso 3 meses).

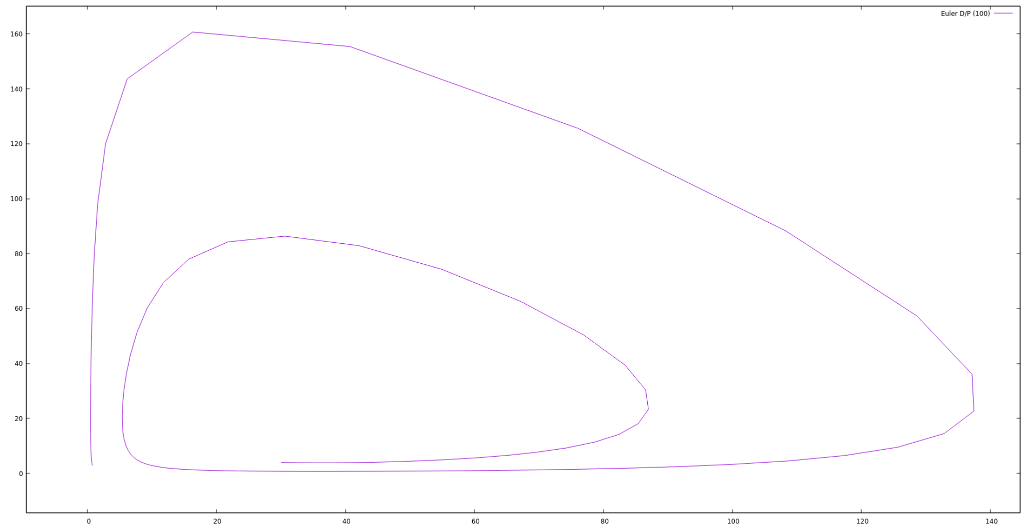


Figura 11: Aproximación del modelo de Lotka - Volterra por el método de Euler (paso 3 meses) Depredador vs Presa.

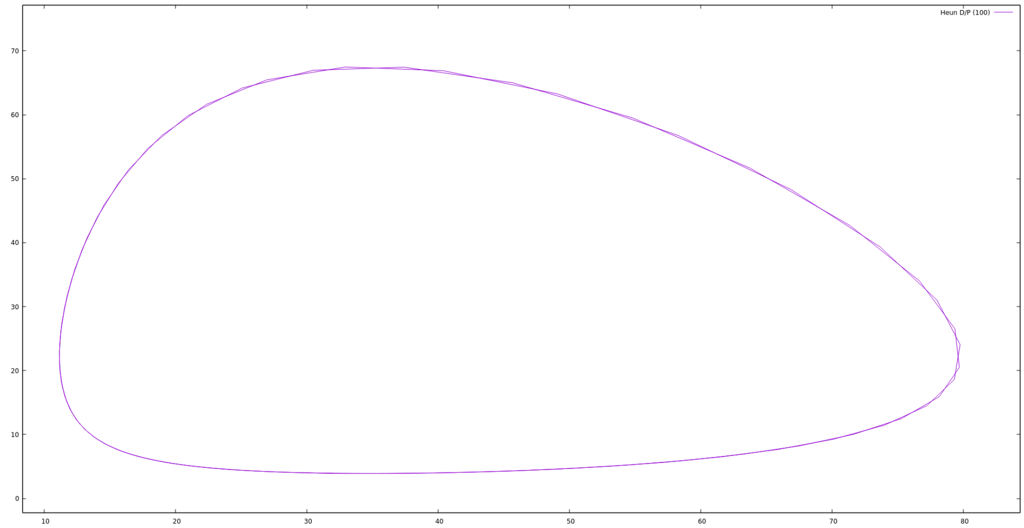


Figura 12: Aproximación del modelo de Lotka - Volterra por el método de Heun (paso 3 meses) Depredador vs Presa.

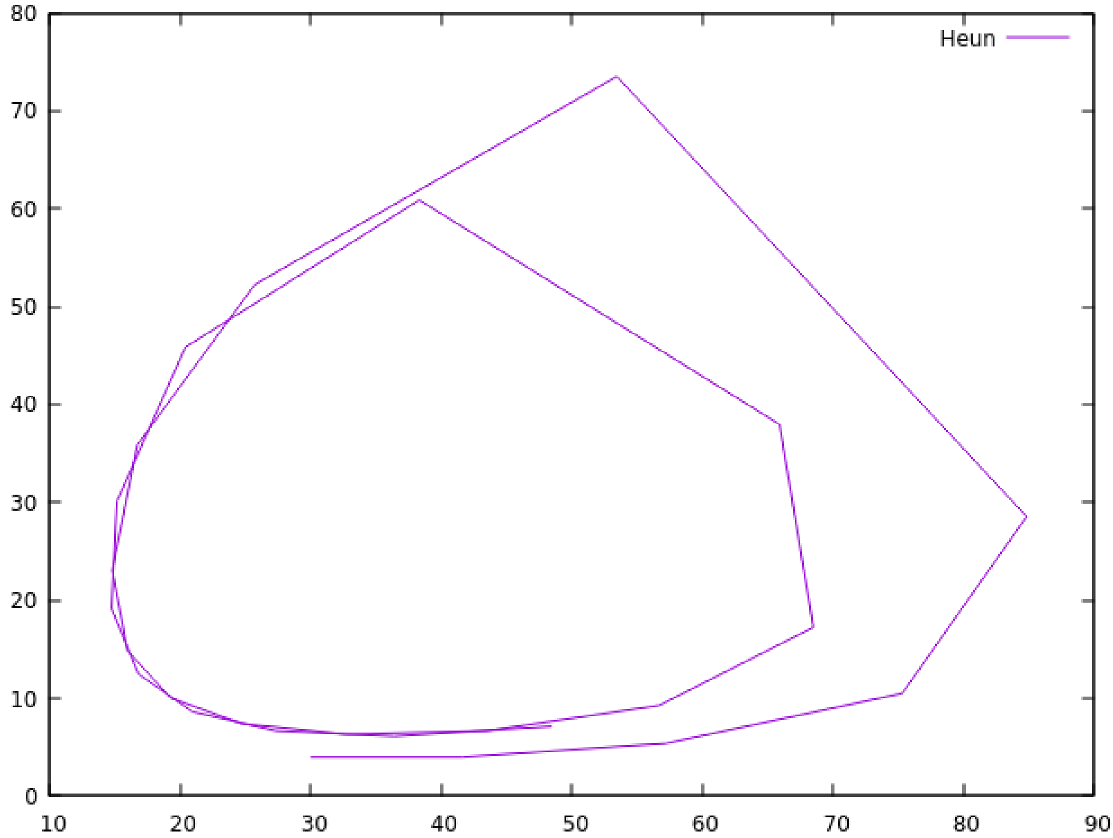


Figura 13: Aproximación del modelo de Lotka - Volterra por el método de Heun (paso 1 año) Depredador vs Presa.

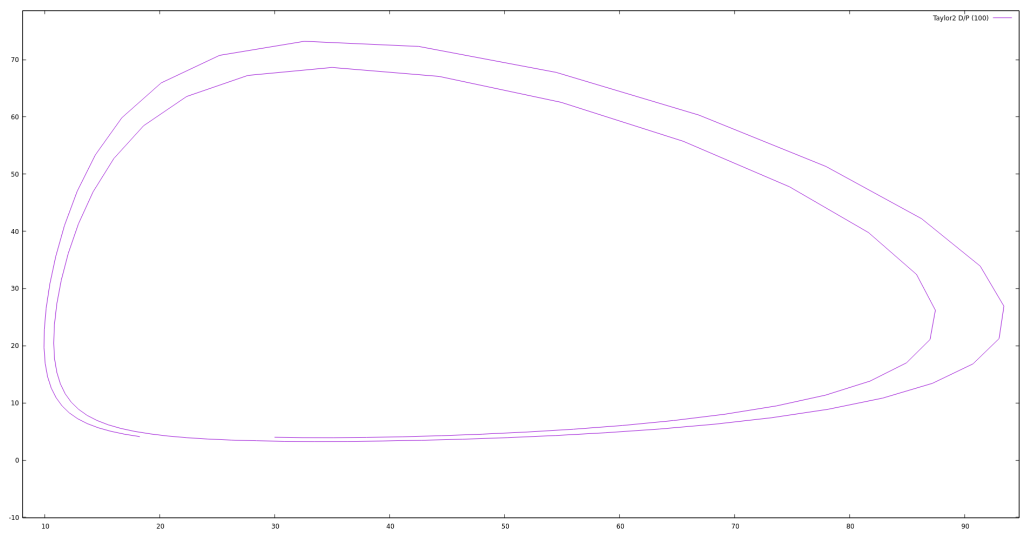


Figura 14: Aproximación del modelo de Lotka - Volterra por el método de Taylor segundo orden (paso 3 meses) Depredador vs Presa.

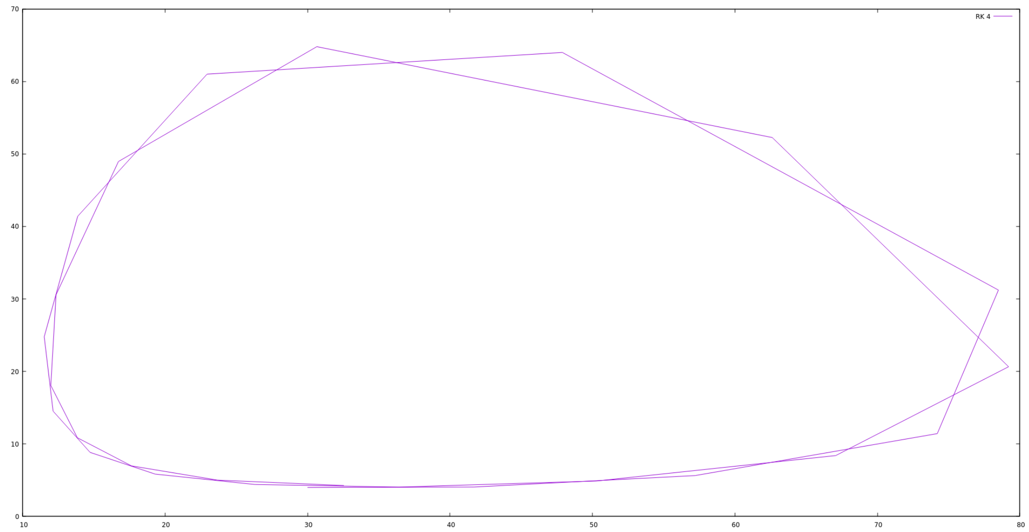


Figura 15: Aproximación del modelo de Lotka - Volterra por el método de Rabge Kuta 4 (paso 1 año) Depredador vs Presa.

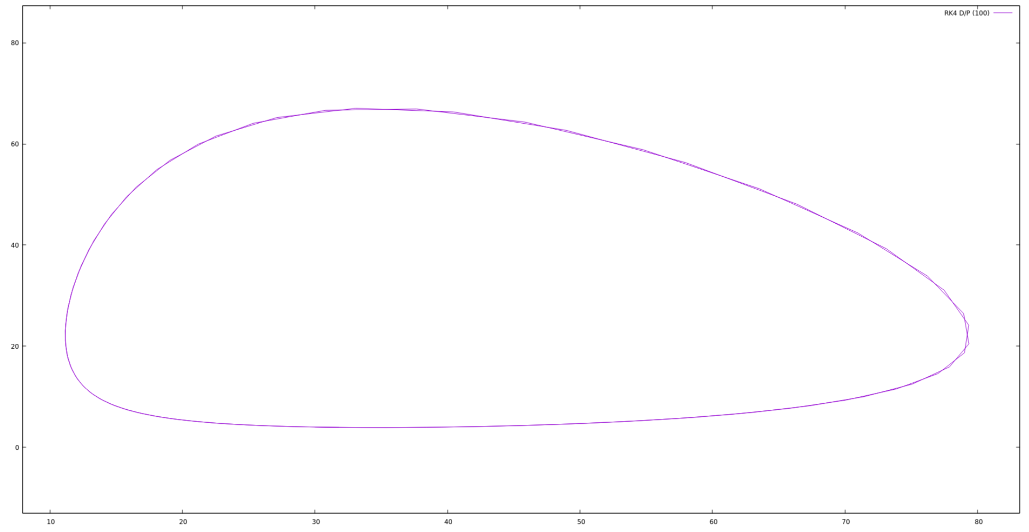


Figura 16: Aproximación del modelo de Lotka - Volterra por el método de Rabge Kuta 4 (paso 3 meses) Depredador vs Presa.

3. Conclusión

De todo lo anterior, podemos hablar de que, si se tiene el suficiente cuidado en como emplear estos métodos, se pueden obtener muy buenas aproximaciones para resolver este tipo de problemas, ya que, por ejemplo, en mi caso, intente de primer mano resolver el problema de depredador presa para un paso de 1 año y tanto el método de Taylor como el método de Euler divergían, sin embargo, al cambiar el tamaño de paso, se logro que estos se estabilizaran lo suficiente para brindar una aproximación.

4. Referencias

[1] Burden, R.L., Faires, J.D. and Burden, A.M. 2015. *Numerical analysis.* Cengage learning.
[2] Quarteroni, A., Sacco, R. and Saleri, F. 1999. *Numerical Mathematics.* Springer.