



DEPARTAMENTO DE INGENIERÍA MATEMÁTICA  
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS  
UNIVERSIDAD DE CHILE  
MA2601-5 ECUACIONES DIFERENCIALES ORDINARIAS

## TAREA NUMÉRICA

---

# MODELO SIMPLE DE FORMACIÓN DE ESTRELLAS.

---

Autor: Juan- Bastian Espinoza C.  
Profesor: Christopher Maulén M.

Fecha de entrega: 7 de junio de 2021  
Santiago, Chile

# Índice de Contenidos

<b>1. Motivación</b>	<b>1</b>
1.1. Euler progresivo . . . . .	1
1.2. Runge-Kutta de orden 4 . . . . .	1
1.3. Plano de fases . . . . .	2
1.4. Odeint de la librería SciPy . . . . .	2
<b>2. Resultados</b>	<b>3</b>
2.1. Solución del sistema con método de Euler progresivo . . . . .	4
2.2. Plano de fases del sistema . . . . .	6
2.3. Periodo límite en función de $\alpha$ . . . . .	7
2.3.1. Periodo límite con Euler progresivo. . . . .	7
2.3.2. Periodo límite con Runge-Kutta de orden 4. . . . .	8
2.3.3. Periodo límite con la librería SciPy . . . . .	8
<b>3. Conclusiones</b>	<b>10</b>
<b>4. Bibliografía</b>	<b>11</b>

## Lista de Figuras

1. Ciclos límite <sup>1</sup> . . . . .	2
2. Curvas de $a$ , $m$ y $s$ superpuestas para cada caso. . . . .	4
3. Plano de fases con respecto a $\alpha$ . . . . .	6
4. Periodo límite con Euler progresivo . . . . .	7
5. Periodo límite con Runge-Kutta de orden 4 . . . . .	8
6. Periodo límite con los tres métodos . . . . .	9

## Lista de Tablas

1. Combinaciones de parámetros para el sistema. . . . .	3
---	---

# 1. Motivación

El presente informe analizará un modelo de formación y evolución de estrellas a través de la resolución de las ecuaciones que rigen dicho modelo. Estas resoluciones se desarrollarán con dos métodos numéricos, el método de Euler y el método de Runge Kutta, en ambos se utilizará el lenguaje Python para trabajar sus respectivas soluciones. Se estudiará como varía la evolución de las estrellas al usar distintas constantes para el modelo y que tanto difieren las soluciones de los métodos comparados con la solución de una rutina tradicional de la librería SciPy de Python. Los métodos numéricos son algoritmos mediante los cuales se obtienen soluciones aproximadas, estos son de suma utilidad para encontrar aproximaciones a problemas complejos con un ordenador. Los computadores son poco eficientes al realizar cálculos en variables continuas, ya que este trabaja con información discreta. Los métodos se encargan de hacer esta adaptación para casos complejos y que son impracticables para resolver a mano. Para esta instancia se ocuparán los métodos brevemente explicados más adelante. En los datos de este informe se cuenta con un sistema emparejado de ecuaciones diferenciales ordinarias por lo que será de suma importancia facilitar la programación del modelo. Se tiene que con  $a$  la fracción de masa de gas atómico,  $m$  la fracción de masa de gas molecular y  $s$  la fracción de masa de estrellas activas:

$$a + m + s = 1$$

Estas se rigen por el siguiente sistema de ecuaciones diferenciales:

$$\frac{da}{dt} = s - k_1 m^2 a$$

$$\frac{dm}{dt} = k_1 m^2 a - k_2 m^{\alpha} s$$

Para mejorar la lectura de este informe se aclararán algunos conceptos a continuación.

## 1.1. Euler progresivo

Para este método se define una condición inicial  $y(x_0) = y_0$ . Se aproxima la integral como rectángulos con base  $x_{n+1} - x_n = \Delta t = h$  y con altura dada por el valor del integrando de  $x_n$ . Luego:

$$y(x_{n+1}) \approx y(x_n) + (x_{n+1} - x_n)f(x_n, y(x_n))$$

Así:

$$y_{n+1} = y_n + hf(x_n, y_n).$$

## 1.2. Runge-Kutta de orden 4

Este método nace de realizar una cuadratura del tipo Simpson para la integral:

$$\int_{x_n}^{x_{n+1}} f(x, y(x)) ds \approx \frac{h}{2} \left( f(x_n, y(x_n)) + 2f(x_{n+\frac{1}{2}}, y(x_{n+\frac{1}{2}})) + f(x_{n+1}, y(x_{n+1})) \right),$$

donde se considera  $x_{n+\frac{1}{2}} = x_n + \frac{h}{2}$ . Este algoritmo define las siguientes variables:

$$\begin{aligned} g_1 &= f(x_n, y_n), \\ g_2 &= f(x_{n+\frac{1}{2}}, y_n + \frac{h}{2}g_1), \\ g_3 &= f(x_{n+\frac{1}{2}}, y_n + \frac{h}{2}g_2), \\ g_4 &= f(x_{n+1}, y_n + hg_3), \end{aligned}$$

La regla de paso final es:

$$y_{n+1} = y_n + \frac{h}{6}(g_1 + 2g_2 + 2g_3 + g_4).$$

Ambos métodos mencionados ser utilizarán para trabajar y es de suma utilidad conocer sus fundamentos con el objetivo de programar debidamente sus algoritmos.

### 1.3. Plano de fases

Un plano de fase es una gráfica que deja observar ciertas características de ecuaciones diferenciales. Consiste en un plano de coordenadas cuyos ejes son los valores de las dos variables, es decir, se graficaría  $(x, y)$ . Esta representación visual entrega una importante información si trabajamos con funciones periódicas (o que tiendan a periódicas), ya que estas generan curvas que pueden ser ‘acorraladas’ por un *conjunto límite*, el cual ayuda a definir distintos *ciclos límite* de un sistema:

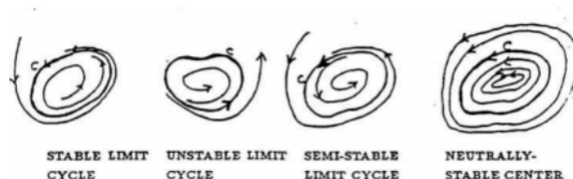


Figura 1: Ciclos límite<sup>1</sup>

- Estable: converge a su conjunto límite.
- Inestable: diverge de su conjunto límite.
- Semi-estable: diverge y converge de su conjunto límite.
- Neutral: Es un ciclo cerrado.

También será de utilidad conocer los *puntos estacionarios*, estos son puntos donde la función se detiene.

### 1.4. Odeint de la librería SciPy

Odeint es una función para resolver ecuaciones diferenciales en el lenguaje Python, se encuentra en la librería SciPy. El módulo ocupa el algoritmo LSODA que resuelve los sistemas con un jacobiano denso o en bandas, este selecciona automáticamente entre los métodos no rígidos (Adams) y rígidos (BDF).<sup>2</sup>

## 2. Resultados

Para la resolución del sistema se explorarán los siguientes casos:

Tabla 1: Combinaciones de parámetros para el sistema.

$k_1$	$k_2$	$\alpha$	$a(0)$	$m(0)$
10	10	1.0	0.15	0.15
8	15	1.2	0.4	0.3
8	15	1.5	0.4	0.3
8	15	1.9	0.4	0.3
8	15	2.0	0.4	0.3
8	15	2.1	0.4	0.3

Además se considerará un intervalo de  $[0, T]$  con  $T = 100$  (millones de años) y un paso de tiempo  $\Delta t = 0,1$  (este valor de  $\Delta t$  será modificado a lo largo del trabajo debido a las vicisitudes de los métodos propuestos).

## 2.1. Solución del sistema con método de Euler progresivo

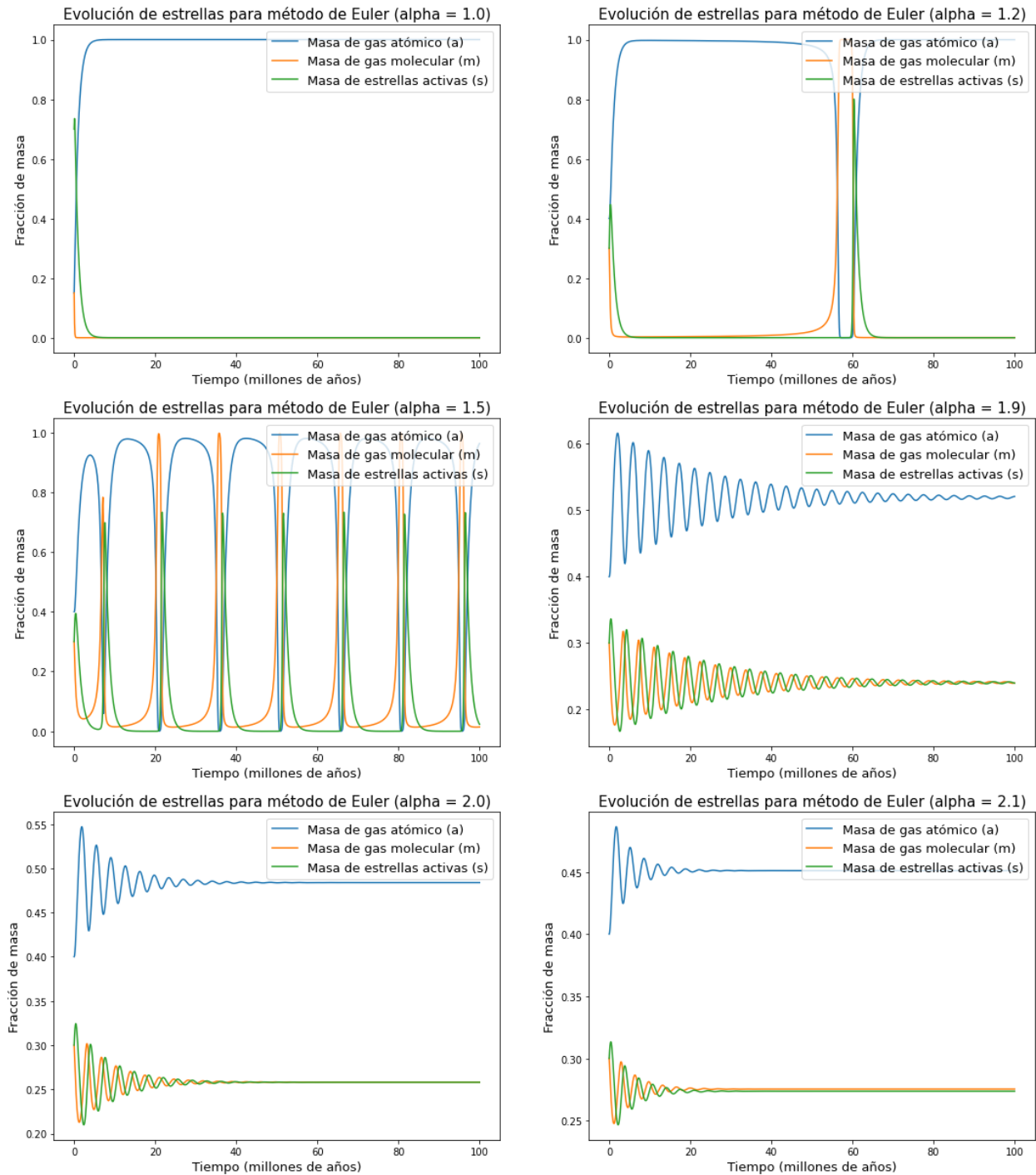


Figura 2: Curvas de  $a$ ,  $m$  y  $s$  superpuestas para cada caso.

Se ocupa el método de Euler progresivo para las soluciones de  $a$  y  $m$ , sin embargo, para  $s$  se utiliza  $s = 1 - (a + m)$ . Al momento de graficar para  $\alpha = 1,2$  los valores divergían del rango  $[0,1]$ ,

para lograr una convergencia se cambió a  $\Delta t = 0,09$  para este caso en específico, con esta variación se observa un gráfico legible. Para el desarrollo de esta parte se necesitó de las librerías de Python: *NumPy*, para arreglos matemáticos, y *Matplotlib* para las gráficas. Al graficar para el caso con  $\alpha = 1,0$ , se observa como el sistema se torna lineal, en cambio para el caso con  $\alpha = 1,5$  se aprecia una periodicidad fuerte y en los siguientes esta periodicidad decae con el transcurso del tiempo.

## 2.2. Plano de fases del sistema

Al graficar  $(a, m)$  con los valores obtenidos con Euler progresivo se obtienen los siguientes planos de fase:

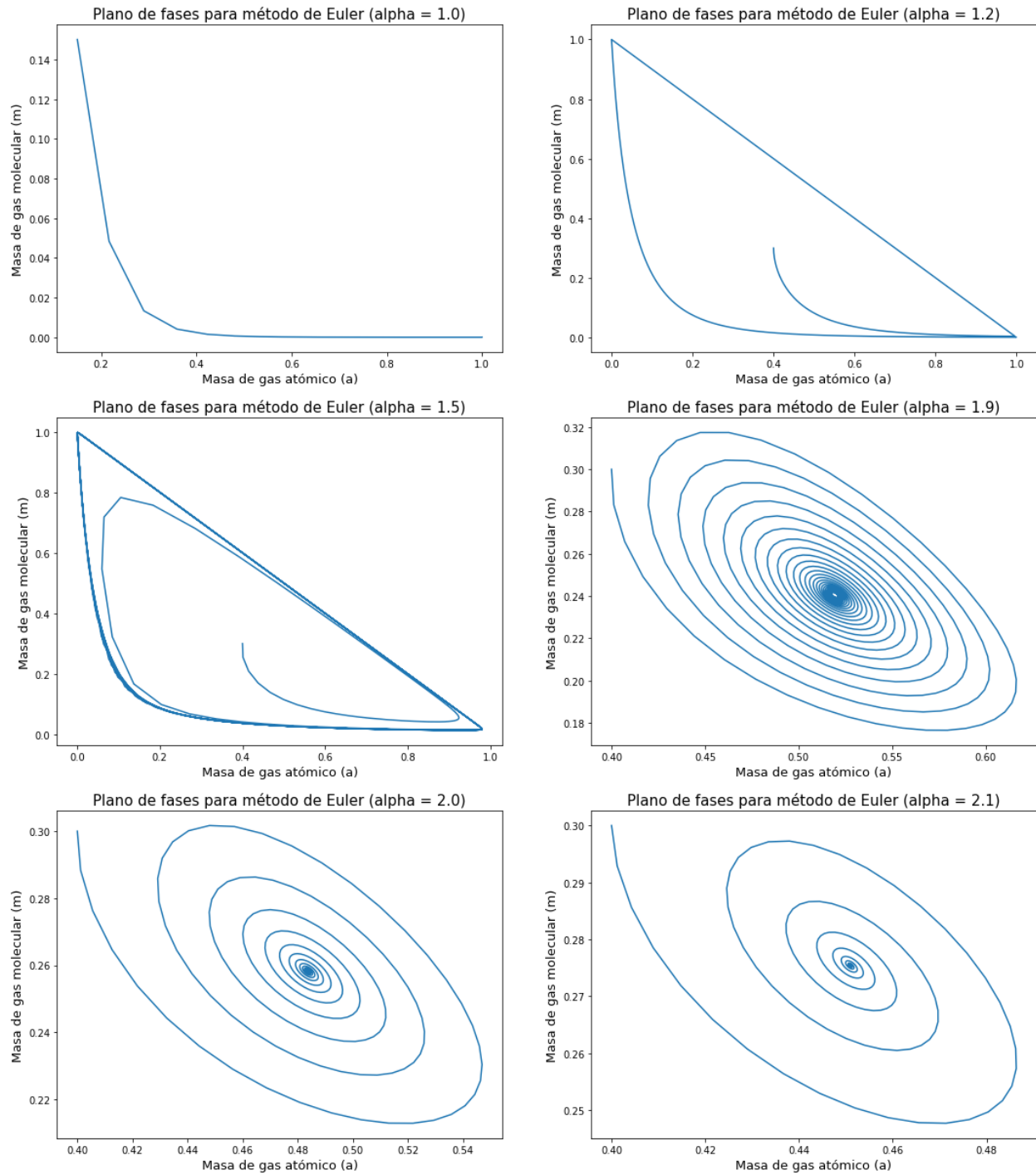


Figura 3: Plano de fases con respecto a  $\alpha$



Se grafica  $(a, m)$ , ya que son las dos variables del sistema. Se logra ver que para el caso con  $\alpha = 1$  no existe ciclo, ya que como vimos en la Figura 2, en ese caso el sistema no es periódico. Para  $\alpha$  con valores 1.2 y 1.5 se observa como son estables con un conjunto límite muy marcado y parecido. El punto estacionario para estos casos debería estar en algún lugar del borde del ciclo límite de ambos casos. Para  $\alpha$  1.9, 2.0 y 2.1 se observa como los ciclos límites son inestables, ya que divergen del conjunto límite, se aprecia como tienden a un punto estacionario cuando  $t \rightarrow \infty$ , punto comprendido cerca de  $[0.45, 0.55]$  para  $a$  y  $[0.24, 0.28]$  para  $m$ .

### 2.3. Periodo límite en función de $\alpha$

Para encontrar este periodo límite se ocupará el segundo caso de la tabla y se hará variar  $\alpha$  entre 1.3 y 1.9, con saltos de 0.1. Ahora bien, el razonamiento para calcular el periodo fue medir la diferencia entre máximos de  $s$  (fracción de masa de estrellas activas). Para conocer el periodo se ocupó un algoritmo que media el valor  $s_n$ , comparaba si este era mayor que  $s_{n_1}$  y a su vez mayor que  $s_{n+1}$  y lo guardaba como máximo relativo, luego se calculaba la diferencia entre el último máximo y el penúltimo máximo. Para saber el límite se observó que usando  $T = 150$  para los gráficos de la figura 2, las curvas ya se encontraban estables, por lo que se toma este intervalo y la diferencia de los dos últimos máximos como *periodo límite*.

#### 2.3.1. Periodo límite con Euler progresivo.

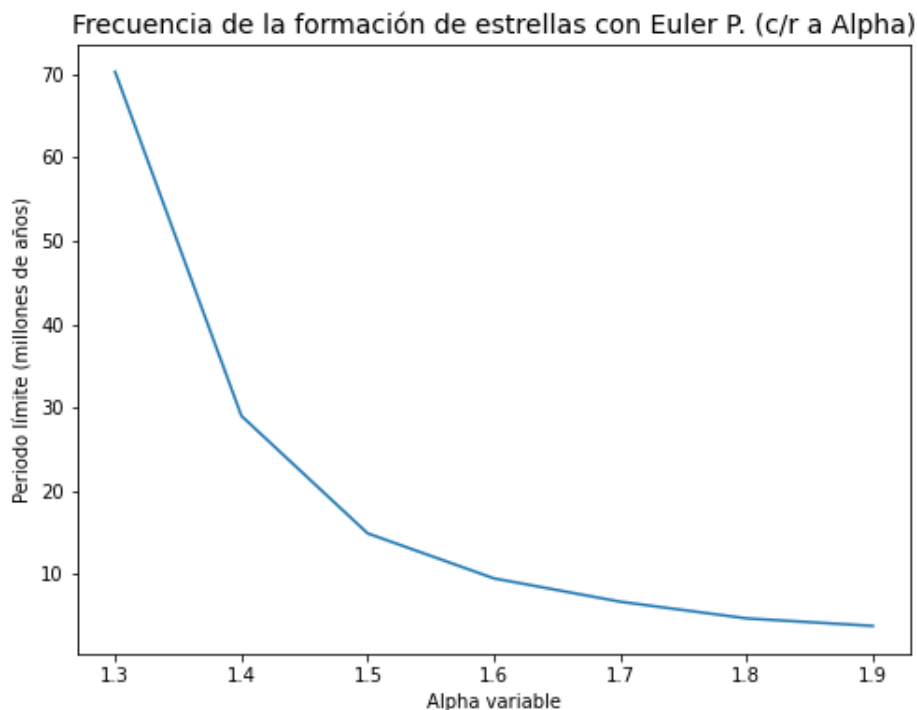


Figura 4: Periodo límite con Euler progresivo

Se ocupó  $\Delta t = 0,1$  para los casos con  $\alpha > 1,3$ , ya que con  $\alpha = 1,3$  no se obtenía convergencia así que se decide trabajar con  $\Delta t = 0,09$  para ese  $\alpha$  en específico. Además ocurrieron otros conflictos en

este caso, ya que se encontraban *máximos falsos*, los cuales provocaban que el programa arrojara periodos extraños, para solucionar esto se agregó al algoritmo que detectara que  $s_n > 0,010.01$  porque era en esos valores cercanos a cero donde se generaban *máximos falsos*. Se observa claramente una función decreciente al aumentar  $\alpha$ , es decir, se reproducirían estrellas con mayor frecuencia.

### 2.3.2. Periodo límite con Runge-Kutta de orden 4.

Al programar este método no se encontraron problemas con algún  $\alpha$  problemático por lo que se ocupa  $\Delta t = 0,1$  en todos los casos. Se obtiene los siguientes valores:

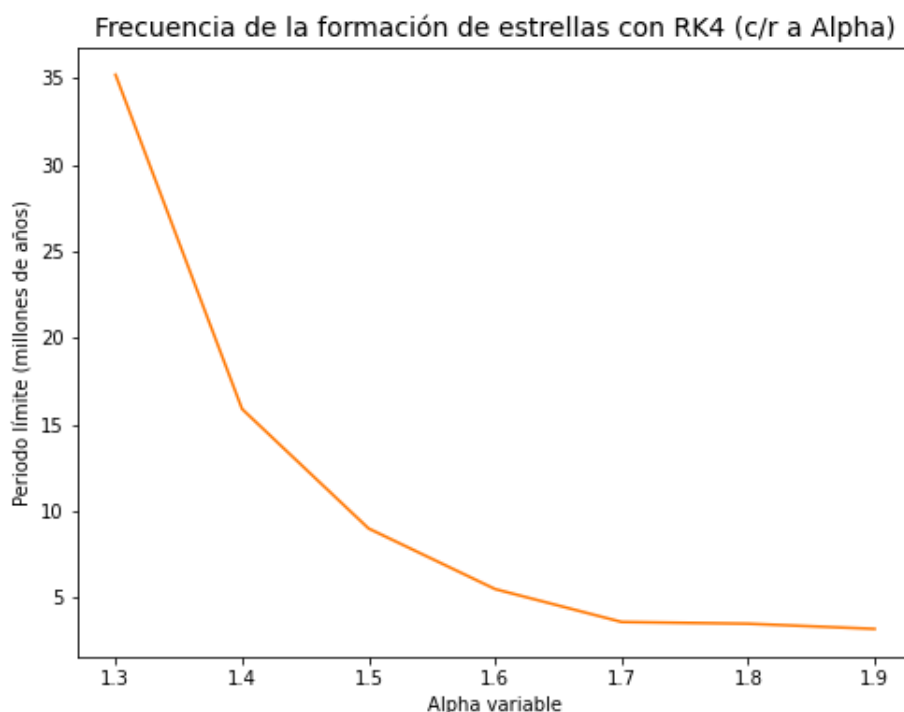


Figura 5: Periodo límite con Runge-Kutta de orden 4

Se observa otra vez una curva decreciente, pero esta vez con valores algo distintos que los conseguidos con el método de Euler. Para  $\alpha = 1,3$  en Euler se tiene un periodo límite de 70 millones de años, mientras que con Runge-Kutta se tiene que este límite es 35 millones de años, lo cual supone la mitad del valor anterior. A partir de  $\alpha = 1,7$  los datos se comienzan a parecer más. A rasgos generales se podría decir que Runge-Kutta sigue la misma tendencia en cuanto a límite del gráfico en comparación a Euler progresivo, sin embargo los valores que toma para los primeros  $\alpha$  difieren una cantidad considerable. En el siguiente punto se hablará de precisión debido a que hasta el momento no sabemos la solución *tradicional*.

### 2.3.3. Periodo límite con la librería SciPy

Resolviendo el sistema con el módulo *Odeint* de la librería *SciPy* y graficado con los métodos anteriores se obtiene:

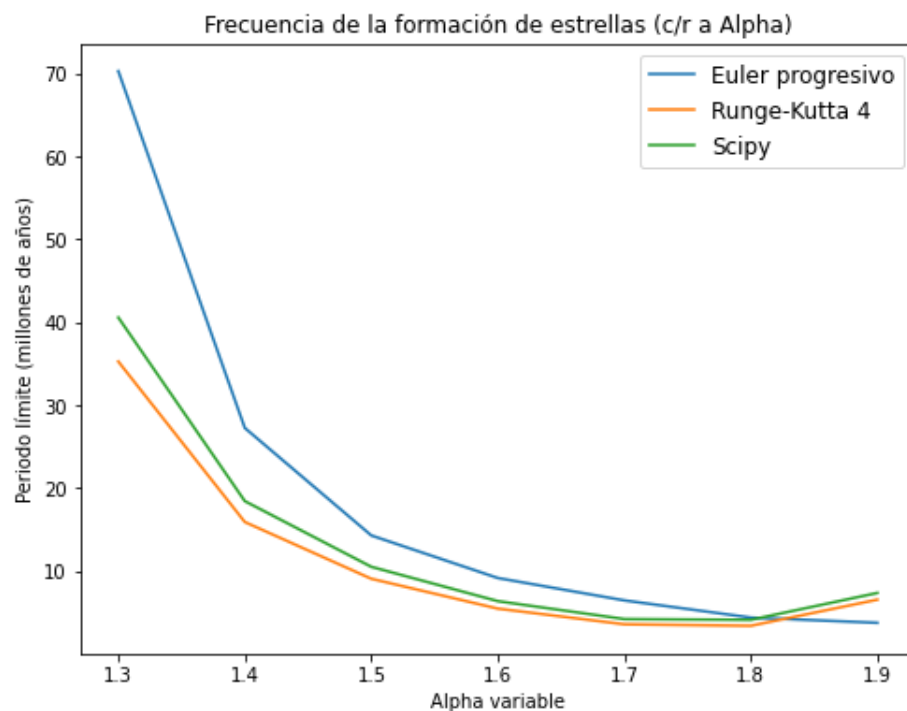


Figura 6: Periodo límite con los tres métodos

Para los programas se decidió ocupar  $\Delta t = 0,09$  para cada método, ya que hacía converger todos los valores. Se observa como el método Runge-Kutta es el que más se acerca a la resolución *tradicional* de las EDOS, mientras que el método de Euler dista considerablemente en un inicio de las demás curvas. Para  $\alpha \geq 1,8$  las soluciones son muy parecidas y arrojan valores que no difieren demasiado.

### 3. Conclusiones

Mediante el transcurso del presente informe se logra modelar el sistema propuesto y analizar sus propiedades a través de dos métodos numéricos programados para graficarlas. Se presentaron problemas al momento de probar el código con el Método de Euler, algunos valores no convergían por lo que se hicieron ajustes para esos casos específicos. Con Runge-Kutta y con Odeint no se presentaron estos problemas por lo que se podrían considerar métodos más estables en comparación a Euler al no encontrar algún  $\alpha$  conflictivo. Al graficar se concluye que Runge-Kutta es más preciso que Euler (al menos con el  $\Delta t$  utilizado), ya que en comparación con la resolución tradicional con Odeint, el primero se acerca mucho más al ser graficado en la búsqueda del periodo límite. Los conceptos y fundamentos de Runge-Kutta se observan mucho más completos que el método de Euler progresivo, así que esa podría ser una causa de que la primera sea una aproximación más precisa. De todas formas se logró apreciar que ambos métodos programados se aproximaron a una solución compleja de manera satisfactoria y sin llegar a valores muy distantes. Se intuye que usando algún  $\Delta t$  menor ambos métodos se habrían acercado más a la solución tradicional del sistema con la librería SciPy. En los planos de fases se concluyó que aquellos inestables tendían a un equilibrio periódico más tranquilo, es decir cuando el tiempo tendía a infinito las oscilaciones no eran excesivas, en cambio para los casos estables, la periodicidad era mucho más marcada (por ejemplo  $\alpha = 1,5$ ). En lo referente al sistema se consigue apreciar como los cambios de las constantes afectan a las soluciones y las propiedades de cada caso, llegando a ser muy distinto uno de otro, como lo fue  $\alpha = 1$  que entregaba un sistema lineal, mientras que los demás uno periódico. Los métodos numéricos fueron de gran utilidad para comprender el comportamiento de las estrellas y es llamativo ver como algoritmos, dentro de lo que cabe, simples logran resolver modelos tan complejos.

## 4. Bibliografía

1. Mattuk, A., 2011. LC. Limit Cycles. [online] Available at: <<https://math.mit.edu/~jorloff/suppnotes/suppnotes03/lc.pdf>> [Accessed 7 June 2021].
2. Hindmarsh, A. and Petzold, L., 2021. LSODA, Ordinary Differential Equation Solver for Stiff or Non-Stiff System. [online] Inis.iaea.org. Available at: <[https://inis.iaea.org/search/search.aspx?orig\\_q=RN:41086668](https://inis.iaea.org/search/search.aspx?orig_q=RN:41086668)> [Accessed 7 June 2021].