

Estructura interna de enanas blancas mediante un modelo politrópico

Nicolás Pan

Facultad de Ciencias

UDELAR

Montevideo, Uruguay

nicolaspan.op@gmail.com

Resumen—La evolución post secuencia principal de estrellas de masas menores a $8M_{\odot}$ finaliza en una enana blanca, estas son estrellas compuestas por un núcleo degenerado el cual soporta la estructura por el efecto de la presión de electrones degenerados. Estos medios se modelan con modelos politrópicos para la presión. En este trabajo nos centramos en la resolución numérica de la ecuación de Lane Emden utilizando el método de Runge-Kutta de orden 4 implementado en Python. Obtuvimos perfiles de densidad y presión para distintos índices politrópicos. Además, calculamos las constantes politrópicas de las soluciones obtenidas, útiles en el estudio de los politropos. Por último, discutimos brevemente la estructura del núcleo de una enana blanca utilizando un modelo politrópico de índice $n = 1,5$ para dos composiciones distintas de enanas blancas.

I. INTRODUCCIÓN

Cuando miramos al cielo en una noche despejada podemos observar una cantidad enorme de puntos de luz, los llamamos estrellas. Dos características fácilmente distinguibles a simple vista son el brillo de la estrella y su color, refinaciones técnicas de estas magnitudes observacionales han permitido a los astrónomos clasificar las estrellas en diferentes categorías. Además, el estudio de la astrofísica estelar nos ha permitido entender los procesos físicos que llevan a las diferentes propiedades observadas en las estrellas del cielo. La astrofísica estelar es el estudio de la estructura y la evolución estelar. La forma en que evolucionan las estructuras estelares nos proporciona información sobre los observables que mencionamos anteriormente como el brillo de una estrella. Sin embargo, los tiempos de evolución estelar son lo suficientemente lentos como para que se establezcan equilibrios temporales en los procesos que allí ocurren. [1, p.72]

Asumiremos un modelo simple de estrella como una esfera de gas simétrica, aislada, únicamente descrita por la coordenada radial. Bajo estas condiciones y, de forma estática, la estructura interna de una estrella esta descrita por la resolución del siguiente sistema de ecuaciones diferenciales

$$\frac{dP}{dr} = -\rho \frac{Gm}{r^2} \quad (1)$$

$$\frac{dm}{dr} = 4\pi r^2 \rho \quad (2)$$

$$\frac{dT}{dr} = \frac{-3}{4ac} \frac{\kappa \rho}{T^3} \frac{F}{4\pi r^2} \quad (3)$$

$$\frac{dF}{dr} = 4\pi r^2 \rho q \quad (4)$$

donde 1 representa el equilibrio hidrostático entre las presiones que sostienen a la estrella y el colapso gravitatorio, 2 es la ecuación de continuidad, 3 es la ecuación de transferencia radiativa y 4 es la ecuación de equilibrio térmico entre la energía que emite y que produce la estrella.

Además de estas ecuaciones, debemos modelar algunos de los parámetros presentes en las ecuaciones que dependen de las variables termodinámicas del problema. Así, asumimos las siguientes relaciones

$$P = \frac{R}{\mu} \rho T + P_e + \frac{1}{3} a T^4 \quad (5)$$

$$\kappa = \kappa_0 \rho^a T^b \quad (6)$$

$$q = q_0 \rho^m T^n \quad (7)$$

donde las constantes en las ecuaciones 6 y 7 dependen de la composición del material y las reacciones nucleares que allí se estén produciendo. Notemos que para la presión estamos asumiendo contribuciones a la presión dadas por un aporte de un gas ideal, de un gas de electrones y la presión de radiación. En cada caso particular del estudio de una estrella deberemos utilizar el termino que sea importante según la situación del material. En el caso que nos importa en este trabajo, la presión en el caso de una enana blanca es la presión de electrones degenerados, se puede probar que esta cumple la siguiente relación

$$P_{e,deg} = K \left(\frac{\rho}{\mu_e} \right)^{\frac{5}{3}} \quad (8)$$

donde K es una constante¹ y

$$\mu_e \equiv \sum_i X_i \frac{Z_i}{A_i} \quad (9)$$

donde X_i es la fracción de masa de cada especie química, Z_i es el numero de protones de cada una y A_i es el numero bariónico.[1, p.40]

$$^1 K = 1 \times 10^7 \frac{Nm^{-2}}{(kgm^{-3})^{\frac{5}{3}}}$$

Notemos que las ecuaciones 1 y 2 están relacionadas con 3 y 4 mediante la dependencia de la presión con la temperatura, es decir, una función del tipo $P = P(T)$. Así, si la presión es solamente una función de la densidad y la composición química de la estrella, los pares de ecuaciones quedan desacopladas y pueden ser resueltos de forma independiente. Esto quiere decir que la configuración hidrostática del sistema es independiente del flujo de calor a través de él. Ahora, multiplicando la ecuación 1 por $\frac{r^2}{\rho}$ y derivando respecto a r obtenemos

$$\frac{d}{dr} \left(\frac{r^2}{\rho} \frac{dP}{dr} \right) = -G \frac{dm}{dr} \quad (10)$$

Sustituyendo la ecuación 2 del lado derecho obtenemos

$$\frac{1}{r^2} \frac{d}{dr} \left(\frac{r^2}{\rho} \frac{dP}{dr} \right) = -4\pi G \rho \quad (11)$$

Decimos que una ecuación de estado es politrópica si es de la forma

$$P = K \rho^\gamma \quad (12)$$

donde $\gamma = 1 + \frac{1}{n}$, aquí n es el llamado índice politrópico. Notemos que para el caso de 8, $n = 1.5$. Sustituyendo 12 y γ en 11 obtenemos

$$\frac{(n+1)K}{4\pi G n} \frac{1}{r^2} \frac{d}{dr} \left(\frac{r^2}{\rho^{\frac{n-1}{n}}} \frac{d\rho}{dr} \right) = -\rho \quad (13)$$

Como estamos modelando una estructura estelar deben cumplirse condiciones de borde. En particular que $\rho = 0$ en $r = R$ ya que P en la superficie es cero. Además, como el equilibrio hidrostático implica que $\frac{dP}{dr} = 0$ en $r = 0$ entonces debe cumplirse que $\frac{d\rho}{dr} = 0$ en $r = 0$. [1, p.75]. Es conveniente que definamos la cantidad adimensional θ que cumple $0 \leq \theta \leq 1$. Así,

$$\rho = \rho_c \theta^n \quad (14)$$

Juntando toda esta información llegamos a que

$$\left[\frac{(n+1)K}{4\pi G \rho_c^{\frac{n-1}{n}}} \right] \frac{1}{r^2} \frac{d}{dr} \left(r^2 \frac{d\theta}{dr} \right) = -\theta^n \quad (15)$$

Notemos que el término entre paréntesis rectos es una constante con dimensiones de distancia cuadrada, así definimos

$$\alpha^2 = \frac{(n+1)K}{4\pi G \rho_c^{\frac{n-1}{n}}} \quad (16)$$

y usamos esta cantidad α para reemplazar r por una cantidad adimensionalizada ξ ,

$$r = \alpha \xi \quad (17)$$

Sustituyendo esta cantidad en la ecuación obtenemos la ecuación de Lane-Emden

$$\frac{1}{\xi^2} \frac{d}{d\xi} \left(\xi^2 \frac{d\theta}{d\xi} \right) = -\theta^n \quad (18)$$

con las condiciones de borde $\theta = 1$ y $\frac{d\theta}{d\xi} = 0$ en $\xi = 0$.

Curiosamente, esta ecuación diferencial tiene soluciones analíticas para tres valores del índice politrópico. Para $n = 0$ vale

$$\theta(\xi) = 1 - \frac{\xi^2}{6} \quad (19)$$

y para $n = 1$ vale

$$\theta(\xi) = \frac{\sin(\xi)}{\xi} \quad (20)$$

la última solución analítica la podemos obtener para $n = 5$ pero es una solución sin significado físico por lo que no escribimos la expresión. Naturalmente, la solución de esta ecuación tiene una raíz que llamaremos ξ_1 que se alcanza cuando $r = R$ por lo que

$$R = \alpha \xi_1 \quad (21)$$

Se pueden probar diversas relaciones con este modelo politrópico. La masa de un politrópico de índice n esta dada por

$$M = -4\pi \alpha^3 \rho_c \xi_1^2 \left(\frac{d\theta}{d\xi} \right)_{\xi_1} \quad (22)$$

además, eliminando α con la ecuación 21 se obtiene una relación lineal entre la densidad central de una estrella y la densidad promedio de la forma

$$\rho_c = D_n \bar{\rho} = D_n \frac{M}{\frac{4}{3}\pi R^3} \quad (23)$$

donde la constante politrópica D_n es

$$D_n = - \left[\left(\frac{3}{\xi_1} \right) \left(\frac{d\theta}{d\xi} \right)_{\xi_1} \right]^{-1}. \quad (24)$$

Definiendo las constantes politrópicas $M_n = -\xi_1^2 \left(\frac{d\theta}{d\xi} \right)_{\xi_1}$ y $R_n = \xi_1$ se puede probar la siguiente relación

$$\left(\frac{GM}{M_n} \right)^{n-1} \left(\frac{R}{R_n} \right)^{3-n} = \frac{[(n+1)K]^n}{4\pi G} \quad (25)$$

la que nos muestra que el radio R y la masa M de una estrella politrópica cumple la relación de proporcionalidad

$$R^{3-n} \propto \frac{1}{M^{n-1}} \quad (26)$$

es decir que el radio decrece a medida que aumentamos la masa. Por último, se puede probar que se cumple para la presión central la siguiente relación

$$P_c = (4\pi)^{\frac{1}{3}} B_n G M^{\frac{2}{3}} \rho_c^{\frac{4}{3}} \quad (27)$$

donde la constante politrópica B_n se define como

$$B_n = \frac{1}{\sqrt[3]{4\pi G}} \frac{\sqrt[3]{4\pi G}}{n+1} \left(\frac{G}{M_n} \right)^{\frac{n-1}{n}} \frac{1}{R_n^{\frac{3-n}{n}}} \left(\frac{3D_n}{4\pi} \right)^{\frac{3-n}{3n}} \quad (28)$$

De esta forma, podemos conocer bastantes características de una estrella cuya ecuación de estado pueda ser escrita de forma politrópica. El objetivo de este trabajo será entonces resolver la ecuación de Lane-Emden para distintos valores del índice politrópico calculando las distintas constantes politrópicas. Luego aplicaremos algunas de las ecuaciones mencionadas para describir de forma breve la estructura del núcleo de una enana blanca.

II. MÉTODOS

Tal como mencionamos en la sección I el objetivo de este trabajo es encontrar soluciones numéricas a la ecuación de Lane Emden para diversos valores de n , así como calcular los correspondientes valores de las constantes politrópicas que brindan información clave en el estudio de estos objetos. Para ello, notemos que la ecuación 18 contiene derivadas segundas en ξ por lo que usaremos el método estándar de pasar a un sistema de dos ecuaciones diferenciales ordinarias de primer orden. El siguiente sistema es equivalente a la ecuación de Lane-Emden.

$$\frac{d\theta}{d\xi} = \frac{-\phi}{\xi^2} \quad (29)$$

$$\frac{d\phi}{d\xi} = \xi^2 \theta^n \quad (30)$$

donde ϕ es una variable auxiliar que usamos simplemente para la resolución de la ecuación. Para resolver este sistema utilizaremos un algoritmo de Runge-Kutta de 4to orden debido a su fácil implementación.

II-A. Método de Runge-Kutta

De forma general, una ecuación diferencial de primer orden en una variable toma la forma

$$\frac{dx}{dt} = f(x, t) \quad (31)$$

donde $f(x, t)$ es una función que debemos especificar. Nuestro objetivo al resolver una ecuación diferencial de este tipo es, partiendo de una condición inicial $x(t) = x_0$ obtener la solución en un tiempo posterior $x(t + h)$ de forma iterativa. Una idea natural que se nos puede ocurrir es hacer un desarrollo de Taylor, despreciando los términos de orden h^2 obtenemos

$$x(t + h) = x(t) + hf(x, t). \quad (32)$$

Este es el conocido como *método de Euler*. Si bien no obtenemos la solución para todo punto, tomando valores de h cada vez mas pequeños obtenemos una aproximación cada vez mejor. Ahora, podríamos pensar que una forma de mejorar este método es tomar mas términos en la expansión de Taylor pero esto conlleva al inconveniente de que tenemos que conocer la derivada de $f(x, t)$ que no siempre es posible obtener. Sin embargo, podemos hacer una expansión de Taylor al rededor de $t + \frac{1}{2}h$, esto se conoce como el *método de Runge-Kutta de orden 2*. Se puede probar que los términos de orden h^2 desaparecen por lo que el error de este método es menor que para el método de Euler. Se puede llevar esta idea mas lejos, haciendo un desarrollo al rededor de varios puntos y luego tomando combinaciones lineales se pueden lograr eliminar términos de orden h^3 y h^4 , esto se conoce como el *método de Runge-Kutta de orden 4*. El resultado del álgebra es el siguiente [2, p.336].

$$k1 = hf(x, t) \quad (33)$$

$$k2 = hf(x + \frac{1}{2}k1, t + \frac{1}{2}h) \quad (34)$$

$$k3 = hf(x + \frac{1}{2}k2, t + \frac{1}{2}h) \quad (35)$$

$$k4 = hf(x + k3, t + h) \quad (36)$$

$$x(t + h) = x(t) + \frac{1}{6}(k1 + 2k2 + 2k3 + k4). \quad (37)$$

Este es el método mas popular para la resolución de ecuaciones diferenciales ordinarias debido a su gran precisión² de calculo y a su fácil implementación.

Es muy sencillo implementar este método si tenemos un sistema con dos ecuaciones diferenciales de primer orden. De forma general este problema es de la forma

$$\frac{dx}{dt} = f_x(x, y, t) \quad \frac{dy}{dt} = f_y(x, y, t). \quad (38)$$

Utilizamos el mismo algoritmo descrito anteriormente para resolver las dos ecuaciones de forma simultánea. Definiendo $\mathbf{r} = (x, y)$ y $\mathbf{f}(\mathbf{r}, t) = (f_x(\mathbf{r}, t), f_y(\mathbf{r}, t))$ el sistema queda en una sola ecuación de la forma

$$\frac{d\mathbf{r}}{dt} = \mathbf{f}(\mathbf{r}, t) \quad (39)$$

sobre el cual aplicamos las ecuaciones 33 - 37 de forma vectorial. Esto nos permite resolver las ecuaciones 29 y 30 de forma sencilla.

II-B. Derivadas

Notemos que algunos de los resultados que llegamos dependen de las derivadas de la función $\theta(\xi)$, en particular los coeficientes D_n y M_n . Recordemos que la definición usual de derivada es

$$\frac{df}{dx} = \lim_{h \rightarrow 0} \frac{f(x + h) - f(x)}{h} \quad (40)$$

la cual podemos aproximar en nuestra desretización a

$$\frac{df}{dx} \approx \frac{f(x + h) - f(x)}{h} \quad (41)$$

Así podemos calcular las derivadas de las soluciones halladas para $\theta(\xi)$ para calcular los coeficientes politrópicos que lo requieran.

III. RESULTADOS

Luego de implementar los algoritmos descritos en la sección II en *Python* resolvimos una familia de ecuaciones de Lane-Emden para diversos valores del índice politrópico. Como mencionamos anteriormente, para 3 valores de n tenemos soluciones analíticas. Tal como vemos en la figura 1, nuestra solución numérica se ajusta de forma correcta a las soluciones analíticas conocidas.

²Error de orden h^4

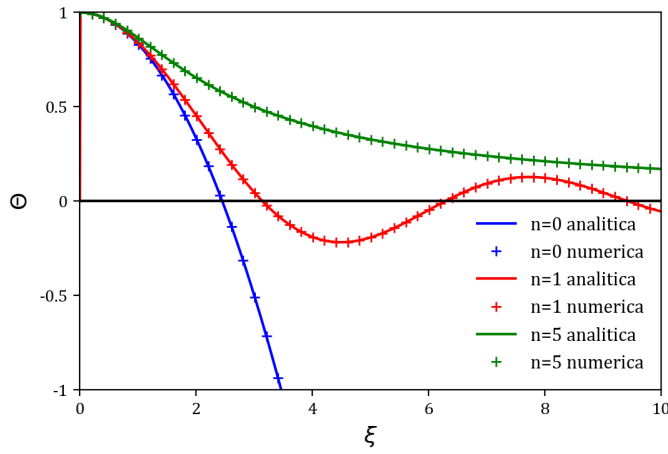


Figura 1: Soluciones analíticas a la ecuación de Lane-Emden superpuestas con las halladas numéricamente para $n = 0$, $n = 1$ y $n = 5$. Vemos que se ajustan de forma correcta.

La gran utilidad que le podemos dar a los métodos numéricos es resolver aquellas ecuaciones que no tienen solución analítica. En la figura 2 vemos las soluciones de la ecuación de Lane-Emden para una variedad mayor de índices politrópicos.

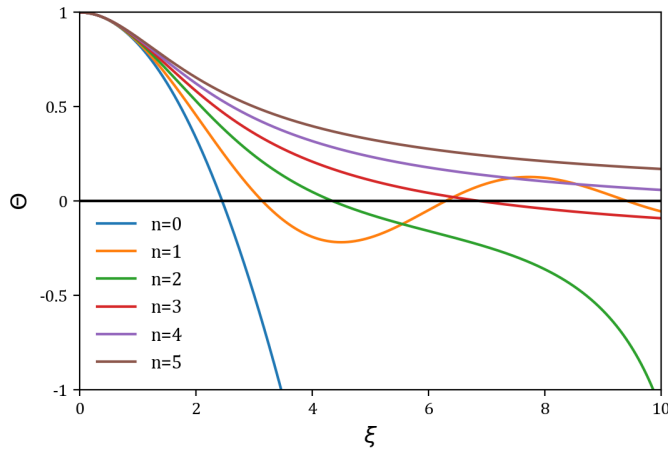


Figura 2: Soluciones numéricas de la ecuación de Lane-Emden para diversos valores de n enteros, observamos como a partir de $n = 4$ no hay raíces para valores de $\xi < 10$.

Debido a los radicales en las ecuaciones de Lane-Emden con índice politrópico no entero, mostramos las soluciones obtenidas hasta la primera raíz para dichas ecuaciones en la figura 3.

De estas soluciones se obtienen fácilmente los coeficientes R_n . Además, utilizando la ecuación 41 podemos calcular la derivada en cada punto iterado y así obtener los coeficientes M_n y D_n . Por último, con los coeficientes calculados y utilizando la ecuación 28 obtenemos los coeficientes B_n . Los resultados de los coeficientes para algunos índices politrópicos se muestran en la tabla I.

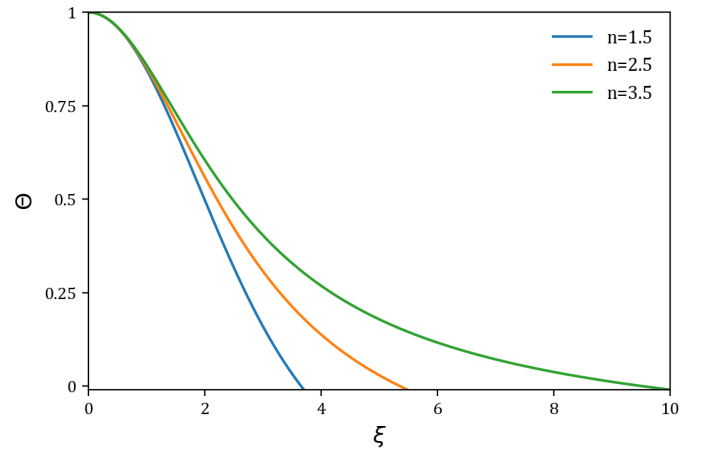


Figura 3: Soluciones numéricas de la ecuación de Lane-Emden para diversos valores de n no enteros.

n	R_n	M_n	D_n	B_n
1.0	3.1429	3.1438	3.2916	0.2330
1.5	3.6544	2.7156	5.9901	0.2055
2.0	4.3537	2.4122	11.403	0.1853
2.5	5.3567	2.1880	23.415	0.1695
3.0	6.8971	2.0188	54.173	0.1565
3.5	9.5365	1.8909	152.89	0.1453

Tabla I: Coeficientes politrópicos para diversos valores de n tanto enteros como no enteros. Los valores que hallamos están en concordancia con los tabulados.

Estos valores se ajustan a los coeficientes tabulados, por una referencia se puede consultar [1, p.77].

Ahora, recordando la ecuación 14 podemos calcular los perfiles de densidad. Mostramos en la figura 4 los resultados para algunos valores del índice politrópico.

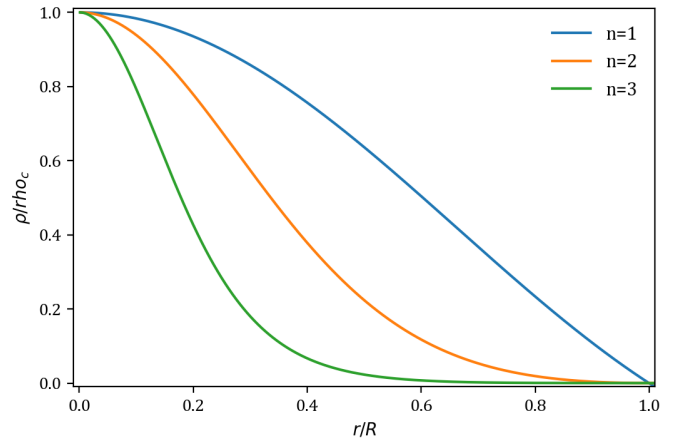


Figura 4: Perfiles de densidad para $n = 1, 2, 3$. Vemos que a medida que el índice politrópico se hace mas grande, la caída de la densidad se acelera a medida que nos acercamos a la superficie.

Vemos que para cualquier de n la densidad de la estrella

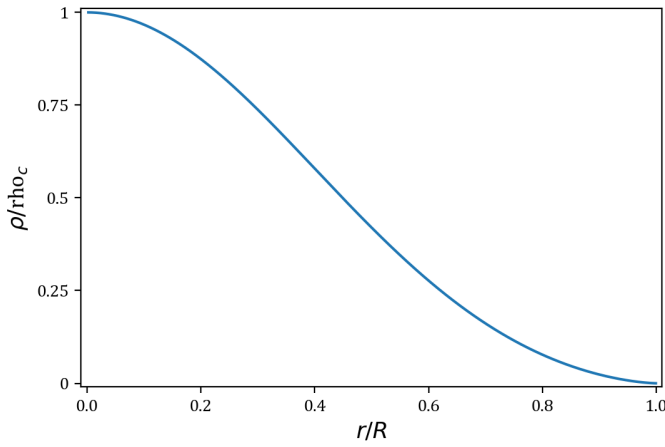


Figura 5: Perfil de densidad en la estructura de una enana blanca, desde el valor central esta cae monótonamente hasta llegar a la superficie.

cae hasta alcanzar un valor nulo en su superficie, tal como esperábamos. Notemos que si bien no aparece en la gráfica, para el caso $n = 0$ la densidad vale la densidad central en toda la estructura. Además, podemos ver que a medida que el índice politrópico aumenta la caída de densidad es cada vez mas rápida en la estructura.

III-A. Enanas blancas

Tal como mencionamos anteriormente, la estructura del núcleo de una enana blanca esta soportada por la presión de electrones degenerados que cumple la relación 8. Usando también que 14 obtenemos

$$P = K \left(\frac{\rho_c}{\mu_e} \right)^{\frac{5}{3}} \theta^{\frac{5n}{3}} \quad (42)$$

Para $n = 1,5$ mostramos el perfil de densidad en la figura 5.

Para ver los perfiles de presión en las enanas blancas debemos conocer las constantes de la relación anterior. Podemos estimar ρ_c recordando 23. Utilizaremos algunos valores típicos para la masa y el radio de una enana blanca. Particularmente en este trabajo tomamos $M = 0,5M_\odot$ y $R = 0,01R_\odot$. Obtenemos entonces un valor para la densidad central de $\rho_c = 4,22 \times 10^9 \frac{kg}{m^3}$ la cual nos muestra que estas estrellas son mas de 4 millones de veces mas densas que el agua.

En cuanto al valor de μ_e , según [1, p.80] para enanas blancas constituidas de C y O $\mu_e = 2$ y para las mismas constituidas de Fe $\mu_e = 2,15$. Con estos valores, se obtienen los perfiles de presión que mostramos en la figura 6.

Observamos cómo los perfiles de presión tienen la misma forma independientemente de la composición de la enana blanca. Notemos que la presión central es menor para el caso de una composición de Fe que para el caso de la composición de C,O. Utilizando 27 obtenemos un valor para la presión central es $P_c = 2,17 \times 10^{22} Pa$, el cual es del mismo orden que los hallados numéricamente.

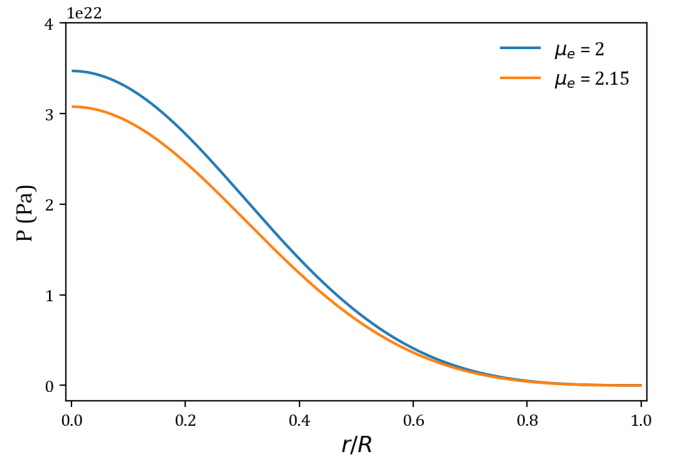


Figura 6: Perfiles de presión para 2 enanas blancas compuestas de C,O ($\mu_e = 2$) y Fe ($\mu_e = 2,15$). Observamos que la forma de la curva es independiente de la composición química de la estrella.

IV. CONCLUSIONES

Logramos a partir del método de Runge-Kutta de orden 4 resolver numéricamente la ecuación de Lane-Emden para un diverso numero de índices politrópicos tanto enteros como no enteros. En particular, mostramos como las soluciones numéricas halladas se ajustan a las soluciones analíticas de los índices que tienen una.

Además, obtuvimos valores para los coeficientes politrópicos que concuerdan con los tabulados, esto permite estudiar los politropos en mas detalle. Por último, obtuvimos los perfiles de densidad y presión para 2 composiciones de enanas blancas, obteniendo curvas esperables a partir de los perfiles de densidad hallados anteriormente.

V. ANEXO

En este anexo aparecen 2 códigos de Python que muestran cómo se hallaron los resultados de este trabajo.

V-A. Código que resuelve la ecuación de Lane-Emden y calcula los coeficientes politrópicos, además de graficar los resultados.

```
1 """
2 Este código fue escrito por Nicolás Pan Rivero el
3 22/10/2021
4 Esta es mi proyecto final de Física Computacional
5 Última modificación: 5/12/2021
6 """
7 #Importamos paquetes y funciones que usaremos
8 import matplotlib.pyplot as plt
9 from numpy import array, arange, linspace, sin, sqrt, pi
10
11 #Calidad de la grafica
12 plt.figure(dpi=200)
13
14 #Iteramos para varios índices de la ecuación
15 for n in [0,1,2,3,4,5]:
16
17     #Definimos el sistema de ecuaciones
18     #diferenciales a resolver:
19     def f(r,xi):
20         theta=r[0]
21         phi=r[1]
22         ftheta = -phi/xi**2
23         fphi = (xi**2)*(theta**n)
24         return array([ftheta,fphi],float)
25
26     #Definimos el rango de xi en el que queremos
27     #integrar
28     a=0.01
29     b=10.0
30     N=5000
31     h=(b-a)/N
32
33     #Definimos el vector de tiempos
34     xipoints=arange(a,b,h)
35
36     #En estas listas iremos guardando la solución
37     thetapoints=[]
38     phipoints=[]
39
40     #Definimos las condiciones iniciales
41     r=array([1.0,0.0],float)
42     indice=0
43     for xi in xipoints:
44         thetapoints.append(r[0])
45         phipoints.append(r[1])
46         k1=h*f(r,xi)
47         k2=h*f(r+0.5*k1,xi+0.5*h)
48         k3=h*f(r+0.5*k2,xi+0.5*h)
49         k4=h*f(r+k3,xi+h)
50         r += (k1+2*k2+2*k3+k4)/6
51
52     #Vamos graficando para cada valor de n la
53     #solución
54     plt.plot(xipoints,thetapoints,label='n=%.0f'%n)
55     #Guardamos las soluciones para cada n
56     if n==0:
57         theta0num=thetapoints
58     if n==1:
59         theta1num=thetapoints
60     if n==5:
61         theta5num=thetapoints
62     if n==2:
63         theta2num=thetapoints
64     if n==3:
```

```
61         theta3num=thetapoints
62     if n==4:
63         theta4num=thetapoints
64
65     raices_n=[]
66     #Calculamos las raíces
67     for i in range(1,len(theta0num)):
68         if (theta0num[i]<0 and theta0num[i-1]>=0):
69             print(0,xipoints[i])
70             raices_n.append(xipoints[i])
71         if (theta0num[i]>0 and theta0num[i-1]<=0):
72             print(0,xipoints[i])
73             raices_n.append(xipoints[i])
74     for i in range(1,len(theta1num)):
75         if (theta1num[i]<0 and theta1num[i-1]>=0):
76             print(1,xipoints[i])
77             raices_n.append(xipoints[i])
78         if (theta1num[i]>0 and theta1num[i-1]<=0):
79             print(1,xipoints[i])
80             raices_n.append(xipoints[i])
81     for i in range(1,len(theta2num)):
82         if (theta2num[i]<0 and theta2num[i-1]>=0):
83             print(2,xipoints[i])
84             raices_n.append(xipoints[i])
85         if (theta2num[i]>0 and theta2num[i-1]<=0):
86             print(2,xipoints[i])
87             raices_n.append(xipoints[i])
88     for i in range(1,len(theta3num)):
89         if (theta3num[i]<0 and theta3num[i-1]>=0):
90             print(3,xipoints[i])
91             raices_n.append(xipoints[i])
92         if (theta3num[i]>0 and theta3num[i-1]<=0):
93             print(3,xipoints[i])
94             raices_n.append(xipoints[i])
95     #Nota: Para n=4 ya no tiene raíces en el rango de xi
96     <10
97
98     #Agregamos detalle a la grafica
99     plt.rc('font',family='Cambria')
100     plt.xlim([0,10])
101     plt.ylim([-1,1])
102     plt.ylabel(r'$\Theta$', fontsize=13)
103     plt.xlabel(r'$\xi$', fontsize=13)
104     plt.axhline(y=0.0,c='k')
105     plt.xticks([0,2,4,6,8,10],[0,2,4,6,8,10])
106     plt.yticks([-1,-0.5,0,0.5,1],[-1,-0.5,0,0.5,1])
107     plt.legend(fontsize=11,frameon=False)
108     plt.show()
109
110     #Vemos las que tienen solución analítica
111     xianalitica = linspace(0,10,1000)
112     theta0=[]
113     theta1=[]
114     theta5=[]
115     for i in range(len(xianalitica)):
116         a=1-((xianalitica[i]**2)/6)
117         theta0.append(a)
118         if i==0:
119             b=0
120         else:
121             b=sin(xianalitica[i])/xianalitica[i]
122         theta1.append(b)
123         c=1/sqrt(1+(xianalitica[i]**2)/3)
124         theta5.append(c)
125
126     #Graficamos las curvas teóricas con las numéricas
127     #superpuestas
128     plt.figure(dpi=200)
129     plt.plot(xianalitica,theta0,'b',label='n=0 analítica')
130     plt.plot(xipoints,theta0num,'b+',label='n=0 numérica')
131     plt.plot(xianalitica,theta1,'r',label='n=1 analítica')
```

```

130 plt.plot(xipoints, theta1num, 'r+', label='n=1 numerica'
131 plt.plot(xianalitica, theta5, 'g', label='n=5 analitica'
132 plt.plot(xipoints, theta5num, 'g+', label='n=5 numerica'
133
134 #Agregamos detalle al grafico
135 plt.rc('font', family='Cambria')
136 plt.xlim([0, 10])
137 plt.ylim([-1, 1])
138 plt.ylabel(r'$\Theta$', fontsize=13)
139 plt.xlabel(r'$\xi$', fontsize=13)
140 plt.axhline(y=0.0, c='k')
141 plt.xticks([0, 2, 4, 6, 8, 10], [0, 2, 4, 6, 8, 10])
142 plt.yticks([-1, -0.5, 0, 0.5, 1], [-1, -0.5, 0, 0.5, 1])
143 plt.legend(fontsize=11, frameon=False)
144 plt.show()
145
146 #Ahora graficaremos rho/rho_central en funcion de r/
147 R para n=1,2,3
148
149 #para r/R tenemos que hacer xi/xi_1
150 xil_n1=raices_n[1]
151 xil_n2=raices_n[4]
152 xil_n3=raices_n[5]
153
154 densidad1 = []
155 densidad2 = []
156 densidad3 = []
157
158 #Calulamos rho/rho_c a partir de theta
159 for i in range(len(thetapoints)):
160     densidad1.append(theta1num[i]**1)
161     densidad2.append(theta2num[i]**2)
162     densidad3.append(theta3num[i]**3)
163
164 xipoints1=[]
165 xipoints2=[]
166 xipoints3=[]
167
168 #Renombramos los xi dividiendo entre la raiz primera
169 for i in range(len(xipoints)):
170     xipoints1.append(xipoints[i]/xil_n1)
171     xipoints2.append(xipoints[i]/xil_n2)
172     xipoints3.append(xipoints[i]/xil_n3)
173
174 #Graficamos y agregamos detalle a la grafica
175 plt.figure(dpi=200)
176 plt.plot(xipoints1, densidad1, label='n=1')
177 plt.plot(xipoints2, densidad2, label='n=2')
178 plt.plot(xipoints3, densidad3, label='n=3')
179 plt.xlim([-0.01, 1.01])
180 plt.ylim([-0.01, 1.01])
181 plt.ylabel(r'$\rho / \rho_{c}$', fontsize=13)
182 plt.xlabel(r'$r/R$', fontsize=13)
183 plt.legend(fontsize=11, frameon=False)
184 plt.show()
185
186 #Calulemos los coeficientes de los politropos
187
188 #Comenzamos por el Dn
189 dtheta1dphi=[]
190 #Calulamos la derivada
191 for i in range(len(theta1num[0:1568])):
192     dtheta1dphi.append(-(theta1num[i]-theta1num[i
193 +1])/h)
194 dtheta1dphi_xi1=dtheta1dphi[-1]
195 D1=-1/((3/raices_n[1])*(dtheta1dphi_xi1))
196 print('D1=', D1)
197
198 dtheta2dphi=[]
199 #Calulamos la derivada
200 for i in range(len(theta2num[0:2174])):
201     dtheta2dphi.append(-(theta2num[i]-theta2num[i
202 +1])/h)
203 dtheta2dphi_xi1=dtheta2dphi[-1]
204 D2=-1/((3/raices_n[4])*(dtheta2dphi_xi1))
205 print('D2=', D2)
206
207 dtheta3dphi=[]
208 #Calulamos la derivada
209 for i in range(len(theta3num[0:3447])):
210     dtheta3dphi.append(-(theta3num[i]-theta3num[i
211 +1])/h)
212 dtheta3dphi_xi1=dtheta3dphi[-1]
213 #print(dtheta3dphi_xi1)
214 D3=-1/((3/raices_n[5])*(dtheta3dphi_xi1))
215 print('D3=', D3)
216
217 #Ahora vamos por las Mn:
218 M1=-1*((raices_n[1])**2)*dtheta1dphi_xi1
219 M2=-1*((raices_n[4])**2)*dtheta2dphi_xi1
220 M3=-1*((raices_n[5])**2)*dtheta3dphi_xi1
221 print('M1=', M1)
222 print('M2=', M2)
223 print('M3=', M3)
224
225 #Ahora vamos por las Rn:
226 R1=raices_n[1]
227 R2=raices_n[4]
228 R3=raices_n[5]
229 print('R1=', R1)
230 print('R2=', R2)
231 print('R3=', R3)
232
233 #Por ultimo las Bn:
234 G=6.67e-11 #constante de gravitacion universal
235 n=1
236 B1=1/(((4*pi)**(1/3))*G)*(((4*pi*G)**(1/n))/(n+1))
237 *((G/M1)**(n-1/n))*((1/(R1**((3-n)/n)))*(((3*D1)
238 /(4*pi))**((3-n)/(3*n))))
239 print('B1=', B1)
240 n=2
241 B2=1/(((4*pi)**(1/3))*G)*(((4*pi*G)**(1/n))/(n+1))
242 *sqrt(G/M2)*((1/(R2**((3-n)/n)))*(((3*D2)/(4*pi))
243 **((3-n)/(3*n))))
244 print('B2=', B2)
245 n=3
246 B3=1/(((4*pi)**(1/3))*G)*(((4*pi*G)**(1/n))/(n+1))
247 *((G/M3)**(2/3))*((1/(R3**((3-n)/n)))*(((3*D3)
248 /(4*pi))**((3-n)/(3*n))))
249 print('B3=', B3)

```

V-B. Código que resuelve la estructura interna de una enana blanca.

```

1 """
2 Este código fue escrito por Nicolás Pan Rivero
3 durante el segundo semestre
4 de 2021. Nuestro código resuelve la estructura de una
5 enana blanca mediante un
6 modelo politrópico.
7 """
8 #Importamos paquetes y funciones
9 import matplotlib.pyplot as plt
10 from numpy import array, arange, linspace, sin, sqrt,
11     zeros, pi
12
13 plt.figure(dpi=200) #Calidad de la figura
14
15 #Índice politrópico para electrones degenerados no
16 relativistas
17 n=1.5
18
19 #Definimos el sistema de ecuaciones diferenciales a
20 resolver:
21 def f(r, xi):

```

```

17 theta=r[0]
18 if theta<0:
19     #Nota: este if esta solamente para evitar
    problemas al calcular raices
20     #de numeros negativos. No es relevante ya
    que la soluci n f sica vale
21     #hasta que theta se hace cero (r=R).
    theta=abs(theta)
22 phi=r[1]
23 ftheta = -phi/xi**2
24 fphi = (xi**2)*((theta)**n)
25 return array([ftheta,fphi],float)
26
27
28 #Definimos el rango de xi en el que queremos
    integrar
29 a=0.01
30 b=10.0
31 N=5000
32 h=(b-a)/N
33
34 #Definimos el vector de tiempos
35 xipoints=arange(a,b,h)
36
37 #En estas listas iremos guardando la solucion
38 thetapoints=[]
39 phipoints=[]
40
41 #Definimos las condiciones iniciales
42 r=array([1.0,0.0],float)
43 indice=0
44 for xi in xipoints:
45     thetapoints.append(r[0])
46     phipoints.append(r[1])
47     k1=h*f(r,xi)
48     k2=h*f(r+0.5*k1,xi+0.5*h)
49     k3=h*f(r+0.5*k2,xi+0.5*h)
50     k4=h*f(r+k3,xi+h)
51     r += (k1+2*k2+2*k3+k4)/6
52
53 #Graficamos y le damos detalle a la figura
54 plt.plot(xipoints[1:1823],thetapoints[1:1823],label=
    'n=%.1f'%n)
55 plt.rc('font',family='Cambria')
56 plt.xlim([0,10])
57 plt.ylim([-0.01,1])
58 plt.ylabel(r'\Theta$',fontsize=13)
59 plt.xlabel(r'\xi$',fontsize=13)
60 plt.axhline(y=0.0,c='k')
61 plt.xticks([0,2,4,6,8,10],[0,2,4,6,8,10])
62 plt.legend(fontsize=11)
63 plt.show()
64
65 #Buscamos la raiz de la soluci n y la guardamos en
    raiz15
66 for i in range(1,len(thetapoints)):
67     if (thetapoints[i]<0 and thetapoints[i-1]>=0):
68         print(n,xipoints[i])
69         raiz15=xipoints[i]
70     if (thetapoints[i]>0 and thetapoints[i-1]<=0):
71         print(n,xipoints[i])
72         raiz15=xipoints[i]
73
74 #Calculamos la densidad/densidad central
75 densidad15=[]
76 for i in range(1823):#Iteramos hasta donde se hace
    negativo el theta
77     densidad15.append(thetapoints[i]**(n))
78 xipoints15=[]
79 for i in range(1823):
80     xipoints15.append(xipoints[i]/raiz15)
81
82 #Graficamos el perfil de densidad en la estrella
83 plt.figure(dpi=200)
84 plt.plot(xipoints15,densidad15)
85 plt.xlim([-0.01,1.01])
86 plt.ylim([-0.01,1.01])
87 plt.ylabel(r'\rho / \rho_c$',fontsize=13)
88 plt.xlabel(r'$r/R$',fontsize=13)
89 plt.yticks([0,0.25,0.5,0.75,1],[0,0.25,0.5,0.75,1])
90 #plt.legend(fontsize=11,frameon=False)
91 plt.show()
92
93 #Ahora calcularemos el perfil de presion para dos
    composiciones qu micas
94 pn15_mu1=[]
95 pn15_mu2=[]
96
97 rho_central=4.22e9
98 k=1e7
99
100 cte1=(rho_central/2)**(5/3)
101 cte2=(rho_central/2.15)**(5/3)
102 for i in range(1823):
103     pn15_mu1.append(k*cte1*(densidad15[i]**(5/3)))
104     pn15_mu2.append(k*cte2*(densidad15[i]**(5/3)))
105 plt.figure(dpi=200)
106 plt.plot(xipoints15[0:1823],pn15_mu1,label='$\mu_e$
    = 2')
107 plt.plot(xipoints15[0:1823],pn15_mu2,label='$\mu_e$
    = 2.15')
108
109 #Agregamos detalle a la figura
110 plt.rc('font',family='Cambria')
111 plt.ylabel('P (Pa)',fontsize=13)
112 plt.xlabel(r'$r/R$',fontsize=13)
113 plt.yticks([0,1e22,2e22,3e22,4e22])
114 plt.xlim([-0.01,1.01])
115 plt.legend(fontsize=11,frameon=False)
116 plt.show()
117
118 #Valores para la presion central
119 Pnum = pn15_mu1[0]
120 B15 = 0.2055
121 G = 6.67e-11
122 M=0.5*1.989e30
123 Pteo = ((4*pi)**(1/3))*B15*G*(M**(2/3))*(rho_central
    **(4/3))
124 print('El valor estimado para Pc es:',Pteo)
125 print('El valor hallado numericamente para Pc es:',
    Pnum)

```

REFERENCIAS

- [1] Dina Prialnik. *An introduction to the Theory of Stellar Structure and Evolution*. 2000.
- [2] Mark Newman. *Computational physics*. 2013.