

Metodo de Numerov

Rafael Corella, Carlos Felix, Bryan Campa

April 20, 2025

Contents

1	Método de Numerov	2
2	Oscilador Harmonico	3
2.1	Solucion Exacta	5
2.1.1	Analisis asintotico	5
3	Notacion y convenciones numericas	7
3.1	Discretizacion y funciones continuas	7
3.2	Implementacion en Python	7
3.3	Notacion	8
4	Algoritmo de biseccion para el oscilador harmonico	8
4.1	Descripcion general	8
4.2	Malla	9
4.3	Punto de retorno clasico	9
4.3.1	ICL	10
4.3.2	Implementacion	10
4.3.3	Notas	10
4.4	Integracion	11
4.4.1	Condiciones iniciales hacia afuera	11
4.4.2	Nodos y validacion de energia	12
4.4.3	Condiciones iniciales hacia adentro	13
4.4.4	Implementacion	13
4.5	Acoplamiento y normalizacion	14
4.5.1	Implementacion	14
4.6	Criterio de convergencia	14
4.6.1	Determinacion de discontinuidad en la primer derivada	14
4.6.2	Logica de biseccion	15
4.6.3	Implementacion	16
5	Algoritmo de biseccion para el atomo de hidrogeno	16
6	Teoria de perturbaciones	17
7	Central Potentials	19
7.1	Radial equaiton	19
7.2	Malla Logaritmica	19

1 Método de Numerov

Método numérico desarrollado por el astrónomo ruso Boris Vasilyevich Numerov en la década de 1910. Se basa en desarrollar la expansión de Taylor de una función $y : I \subset \mathbb{R} \rightarrow \mathbb{R}$ alrededor de un punto $x_0 \in I$, donde y es solución a la ecuación diferencial de segundo orden

$$y''(x) = -g(x)y(x) + s(x), \quad (1)$$

con $g : I \rightarrow \mathbb{R}$ y $s : I \rightarrow \mathbb{R}$ como funciones conocidas, definidas en el mismo intervalo I que y . Para que la expansión de Taylor sea válida hasta el orden requerido, se exige que y sea al menos C^4 i.e. cuatro veces diferenciable. Además, g y s deben ser funciones suficientemente suaves e.g. continuas, para garantizar la existencia de una solución.

El método de Numerov sigue un esquema de diferencias finitas, por lo que comenzamos con una expansión de Taylor, a quinto orden. Para un paso adelante ($x = x_0 + \Delta x$) y un paso atrás ($x = x_0 - \Delta x$), se tiene

$$y(x_0 + \Delta x) = y(x_0) + y'(x_0)\Delta x + \frac{y''(x_0)}{2!}\Delta x^2 + \frac{y'''(x_0)}{3!}\Delta x^3 + \frac{y^{(4)}(x_0)}{4!}\Delta x^4 + \frac{y^{(5)}(x_0)}{5!}\Delta x^5 + \mathcal{O}(\Delta x^6),$$

$$y(x_0 - \Delta x) = y(x_0) - y'(x_0)\Delta x + \frac{y''(x_0)}{2!}\Delta x^2 - \frac{y'''(x_0)}{3!}\Delta x^3 + \frac{y^{(4)}(x_0)}{4!}\Delta x^4 - \frac{y^{(5)}(x_0)}{5!}\Delta x^5 + \mathcal{O}(\Delta x^6).$$

Se define una malla uniforme $x_n = x_0 + n\Delta x$ y se denota:

- $y(x_n) \equiv y_n$ (valores de la función en puntos de la malla).
- $y(x_n \pm \Delta x) \equiv y_{n\pm 1}$ (puntos adyacentes).

Así, las expansiones se reescriben como:

$$y_{n+1} = y_n + y'_n\Delta x + \frac{1}{2}y''_n(\Delta x)^2 + \frac{1}{6}y'''_n(\Delta x)^3 + \frac{1}{24}y^{(4)}_n(\Delta x)^4 + \frac{1}{120}y^{(5)}_n(\Delta x)^5 + \mathcal{O}(\Delta x^6)$$

$$y_{n-1} = y_n - y'_n\Delta x + \frac{1}{2}y''_n(\Delta x)^2 - \frac{1}{6}y'''_n(\Delta x)^3 + \frac{1}{24}y^{(4)}_n(\Delta x)^4 - \frac{1}{120}y^{(5)}_n(\Delta x)^5 + \mathcal{O}(\Delta x^6)$$

Al sumar ambos desarrollos se obtiene

$$y_{n+1} + y_{n-1} = 2y_n + y''_n(\Delta x)^2 + \frac{1}{12}y^{(4)}_n(\Delta x)^4 + \mathcal{O}[(\Delta x)^6]$$

Luego, se puede escribir

$$z_n \equiv y'' = -g_n y_n + s_n,$$

y aplicamos la expresión obtenida

$$z_{n+1} + z_{n-1} = 2z_n + z''_n(\Delta x)^2 + \mathcal{O}[(\Delta x)^4]$$

Esta es una expresión para la segunda derivada desarrollando en Taylor hasta tercer orden, de modo que, además, se obtiene una expresión para la cuarta derivada $y^{(4)}$

$$z''_n = \frac{z_{n+1} + z_{n-1} - 2z_n}{(\Delta x)^2} + \mathcal{O}[(\Delta x)^2] = y^{(4)}_n.$$

sustituyendo en la expresion

$$y_{n+1} + y_{n-1} = 2y_n + y_n''(\Delta x)^2 + \frac{1}{12}y_n^{(4)}(\Delta x)^4 + \mathcal{O}[(\Delta x)^6]$$

se obtiene la formula de Numerov

$$\begin{aligned} y_{n+1} &= 2y_n - y_{n-1} + (z_n)(\Delta x)^2 + \frac{1}{12}(z_{n+1} + z_{n-1} - 2z_n)(\Delta x)^2 + \mathcal{O}((\Delta x)^6) \\ y_{n+1} &= 2y_n - y_{n-1} + (-g_n y_n + s_n)(\Delta x)^2 + \frac{1}{12}[(-g_{n+1}y_{n+1} + s_{n+1}) \\ &\quad + (-g_{n-1}y_{n-1} + s_{n-1}) - 2(-g_n y_n + s_n)](\Delta x)^2 + \mathcal{O}((\Delta x)^6) \\ y_{n+1} \left[1 + \frac{1}{12}g_{n+1}(\Delta x)^2 \right] &= 2y_n \left[1 + \left(\frac{-g_n}{2} + \frac{g_n}{12} \right)(\Delta x)^2 \right] \\ &\quad - y_{n-1} \left[1 + \frac{1}{12}g_{n-1}(\Delta x)^2 \right] + \frac{1}{12}(s_{n+1} + 10s_n + s_{n-1})(\Delta x)^2 + \mathcal{O}((\Delta x)^6) \\ y_{n+1} \left[1 + \frac{1}{12}g_{n+1}(\Delta x)^2 \right] &= 2y_n \left[1 - \frac{5}{12}g_n(\Delta x)^2 \right] \\ &\quad - y_{n-1} \left[1 + \frac{1}{12}g_{n-1}(\Delta x)^2 \right] + \frac{1}{12}(s_{n+1} + 10s_n + s_{n-1})(\Delta x)^2 + \mathcal{O}((\Delta x)^6). \end{aligned}$$

Esta expresion nos permite propagar la solucion y_n conociendo los primeros dos valores y_0 y y_1 . Las ecuaciones que se van a resolver por medio de esta formula tienen $s(x) = 0$. Luego, podemos simplificar la formula por medio de la cantidad

$$f_n \equiv 1 + \frac{1}{12}g_n(\Delta x)^2,$$

haciendo

$$\begin{aligned} f_n - 1 &= \frac{1}{12}g_n(\Delta x)^2 \\ 1 - 5(f_n - 1) &= 1 - \frac{5}{12}g_n(\Delta x)^2 \\ 6 - 5f_n &= 1 - \frac{5}{12}g_n(\Delta x)^2 \\ 12 - 10f_n &= 2 \left(1 - \frac{5}{12}g_n(\Delta x)^2 \right) \end{aligned}$$

se reescribe la formula de Numerov

$$\boxed{y_{n+1} = \frac{(12 - 10f_n)y_n - f_{n-1}y_{n-1}}{f_{n+1}}} \quad (2)$$

Este metodo se aplica a los sistemas fisicos que se pueden modelar con una ecuacion diferencial del tipo 1. En particular vamos a resolver la ecuacion de Schrodinger unidimensional independiente del tiempo para el sistema de oscilador armonico y el atomo de hidrogeno, notamos que hay mas ecuaciones interesantes que se pueden resolver como la ecuacion de Poisson.

2 Oscilador Harmonico

Al aplicar el metodo de separacion de variables la ecuacion de Schrodinger se obtiene la ecuacion de Schrodinger unidimensional para una funcion $\psi = \psi(x)$

$$\frac{\partial^2 \psi}{\partial x^2} = -\frac{2m}{\hbar^2}(E - V(x))\psi \quad (3)$$

Un sistema de oscilador armonico es cuando el potencial en la ecuacion (3) es

$$V(x) = -\frac{1}{2}Kx^2.$$

donde K es una constante.

El desarrollo se simplifica en gran medida al hacer el cambio a unidades adimensionales:

- Variable adimensional ξ
- Esta variable se relaciona con x por medio de la longitud λ de modo que $x = \lambda\xi$

$$\begin{aligned} \frac{\partial^2 \psi}{\partial (\lambda\xi)^2} &= \left(-\frac{2mE}{\hbar^2} + \frac{mK\lambda^2}{\hbar^2}\xi^2 \right) \psi \\ \frac{\partial^2 \psi}{\partial \xi^2} &= \left(-\frac{2mE\lambda^2}{\hbar^2} + \frac{mK\lambda^4}{\hbar^2}\xi^2 \right) \psi \end{aligned}$$

- Hacemos $mK\lambda^4/\hbar^2 = 1$, de donde

$$\lambda = (\hbar^2/mK)^{1/4}$$

- Relacionamos la frecuencia angular del oscilador con la constante de fuerza

$$\omega = \sqrt{\frac{K}{m}} \implies K = m\omega^2$$

- La variable adimensional queda

$$\lambda\xi = x \implies \xi = \left(\frac{mK}{\hbar^2} \right)^{1/4} x = \left(\frac{m\omega}{\hbar} \right)^{1/2} x$$

- Introducimos la energia adimensional ϵ

$$\epsilon = \frac{2E}{\hbar\omega}$$

- Sustituyendo estas expresiones en la ecuacion de Schrödinger

$$\begin{aligned} \frac{\partial^2 \psi}{\partial \xi^2} &= \left(-\frac{2mE\lambda^2}{\hbar^2} + \frac{mK\lambda^4}{\hbar^2}\xi^2 \right) \psi \\ \frac{\partial^2 \psi}{\partial \xi^2} &= \left(-\frac{2m(\epsilon\hbar\omega/2)(\hbar^2/m^2\omega^2)^{1/2}}{\hbar^2} + \xi^2 \right) \psi \end{aligned}$$

- Finalmente la ecuacion de Schrödinger adimensional es:

$$\boxed{\frac{\partial^2 \psi}{\partial \xi^2} = -2\left(\epsilon - \frac{\xi^2}{2}\right)\psi} \quad (4)$$

con $V(\xi) = \frac{1}{2}\xi^2$.

2.1 Solucion Exacta

2.1.1 Analisis asintotico

Para grande ξ , las soluciones de (4), donde ϵ se puede despreciar, son de la forma

$$\psi(\xi) \sim \xi^n e^{\pm \xi^2/2},$$

donde n cualquier valor finito. El exponente con signo positivo da lugar a funciones de onda no normalizables por lo que corresponde a soluciones no fisicas. Entonces asumimos que su comportamiento asintotico hace que la funcion de onda sea

$$\psi(\xi) = H(\xi) e^{-\xi^2/2} \quad (5)$$

donde $H(\xi)$ es alguna funcion bien comportada para ξ grande (de modo que el comportamiento asintotico este determinado por el factor $e^{-\xi^2/2}$). En particular $H(\xi)$ no debe crecer como e^{ξ^2} para asi obtener soluciones fisicas. Bajo asumir que la funcion de onda es (5), la ecuacion (4) se convierte en una ecuacion para $H(\xi)$:

$$\begin{aligned} \frac{d^2}{d\xi^2} (H(\xi) e^{-\xi^2/2}) &= -2 \left(\epsilon - \frac{\xi^2}{2} \right) H(\xi) e^{-\xi^2/2} \\ \frac{d^2 H(\xi)}{d\xi^2} e^{-\xi^2/2} - \xi \frac{dH(\xi)}{d\xi} e^{-\xi^2/2} - \xi \frac{dH(\xi)}{d\xi} e^{-\xi^2/2} + \xi^2 H(\xi) e^{-\xi^2/2} - H(\xi) e^{-\xi^2/2} &= -2 \left(\epsilon - \frac{\xi^2}{2} \right) H(\xi) e^{-\xi^2/2} \\ \frac{d^2 H(\xi)}{d\xi^2} - 2\xi \frac{dH(\xi)}{d\xi} + (2\epsilon - 1)H(\xi) &= 0 \end{aligned}$$

Se expande la solucion $H(\xi)$ en una serie de potencias

$$H(\xi) = \sum_{n=0}^{\infty} A_n \xi^n$$

la primer derivada es simplemente

$$\frac{dH}{d\xi} = \sum_{n=0}^{\infty} n A_n \xi^{n-1}$$

para la segunda derivada, diferenciamos cada termino

$$\frac{d^2 H}{d\xi^2} = \frac{d}{d\xi} (A_1 + 2A_2\xi + 3A_3\xi^2 + \dots) = 2A_2 + 2*3A_3\xi + 3*4A_4\xi^2 + \dots = \sum_{n=0}^{\infty} (n+1)(n+2) A_{n+2} \xi^n$$

sustituyendo en la ecuacion para $H(\xi)$ se tiene

$$\begin{aligned} \frac{d^2 H(\xi)}{d\xi^2} - 2\xi \frac{dH(\xi)}{d\xi} + (2\epsilon - 1)H(\xi) &= 0 \\ \sum_{n=0}^{\infty} \{ (n+1)(n+2) A_{n+2} \xi^n - 2\xi (n A_n \xi^{n-1}) + (2\epsilon - 1) A_n \xi^n \} &= 0 \\ \sum_{n=0}^{\infty} \{ (n+1)(n+1) A_{n+2} + (2\epsilon - 2n - 1) A_n \} \xi^n &= 0 \end{aligned}$$

esta expresion se debe satisfacer para todo ξ por el teorema de existencia y unicidad, entonces los coeficientes de todo orden deben ser cero:

$$(n+2)(n+1)A_{n+2} + (2\epsilon - 2n - 1)A_n = 0$$

asi, dados A_0 y A_1 , se puede determinar por recursion $H(\xi)$ como una serie de potencias

$$A_{n+2} = \frac{(2\epsilon - 2n - 1)A_n}{n^2 + 3n + 2} \quad (6)$$

Para n muy grande, se tiene:

$$A_{n+2} \sim \frac{2A_n}{n}$$

Se resuelve esta recursion para el caso par e impar:

- Para una potencia par $n = 2k$:

$$A_{2k+2} \sim \frac{1}{k} A_{2k}$$

- Iterando:

$$A_{2k} \sim \frac{1}{k-1} A_{2k-2} \sim \frac{1}{k-1} \cdot \frac{1}{k-2} A_{2k-4} \sim \frac{A_0}{(k-1)!}$$

- Usando $(k-1)! = k!/k$, para k muy grande, la solucion a la recursion es

$$A_{2k} \sim \frac{A_0}{k!}$$

- Similarmente, para una potencia impar $n = 2k + 1$:

$$A_{2k+3} \sim \frac{2}{2k+1} A_{2k+1} \sim \frac{1}{k} A_{2k+1}$$

$$A_{2k+1} \sim \frac{A_1}{k!}$$

- Por lo tanto, para n muy grande, la recursion se comporta como:

$$A_n \sim \frac{1}{(n/2)!}$$

Esto implica:

$$H(\xi) \sim \sum_k \left[\frac{A_0}{k!} \xi^{2k} + \frac{A_1}{k!} \xi^{2k+1} \right] = A_0 e^{\xi^2} + A_1 \xi e^{\xi^2}$$

Esta expresion se interpreta como que la recurrencia (6) produce una funcion $H(\xi)$ que crece como e^{ξ^2} y da soluciones divergentes, i.e. no fisicas. Para prevenir este comportamiento, debemos truncar la serie despues de algun n y asi reducir la solucion a un polinomio de grado finito. Entonces, en la recursion (6), para que la serie termine,

$$A_{n+2} = \frac{(2\epsilon - 2n - 1)A_n}{(n+2)(n+1)}$$

$$2\epsilon - 2n - 1 = 0$$

$$\epsilon = n + \frac{1}{2}$$

donde n es un entero positivo. Esta condicion nos da la cuantizacion de la energia del oscilador harmonico:

$$E_n = (n + \frac{1}{2})\hbar\omega, \quad n \in \mathbb{Z}^+ \quad (7)$$

Los polinomios correspondientes $H_n(\xi)$ son los polinomios de Hermite, donde $H_n(\xi)$:

- Es de grado n en ξ
- Tiene n nodos
- Es par para n par e impar para n impar

Finalmente, la funcion de onda correspondiente a la energia E_n es

$$\psi_n(\xi) = H_n(\xi)e^{-\xi^2/2}$$

3 Notacion y convenciones numericas

3.1 Discretizacion y funciones continuas

Para representar funciones continuas $\psi(x) : x \in I \subset \mathbb{R} \rightarrow \mathbb{R}$ de forma numerica:

- Se discretiza el dominio en una malla equiespaciada $x_i = i * \Delta x$, donde

$$i \in \{0, 1, \dots, mesh\}$$

- Los valores de $\psi(x_i)$ se almacenan en un arreglo ψ_i indexado, donde

$$\psi_i \equiv \psi(x_i) \rightarrow \psi[i] \text{ en Python}$$

3.2 Implementacion en Python

Para una malla con $mesh$ intervalos ($mesh + 1$ puntos), el ultimo elemento del arreglo es

```
mesh = 100
x = np.linspace(mesh+1)
psi = f(x) # Alguna funcion de x
psi[0] # Primer elemento de psi
psi[mesh] # Ultimo elemento de psi
```

- Como Python es cero indexado, hacer el tamano de la malla $mesh + 1$ asegura que $x_{max} = mesh * \Delta x$ y que $\psi_{x_{max}} = psi[mesh]$

3.3 Notacion

Usamos tres diferentes maneras de representar el mismo numero:

$$\underbrace{\psi_R(x)}_{\text{Funcion que depende de } x} \rightarrow \underbrace{\psi_i^R}_{\text{iesimo valor de } \psi^R} \rightarrow \underbrace{psi_R[i]}_{\text{Elemento } i \text{ del arreglo } psi_R}$$

4 Algoritmo de biseccion para el oscilador harmonico

4.1 Descripcion general

- **Inicializacion de la malla**

- Se discretiza el dominio espacial en $mesh + 1$ puntos uniformemente espaciados $x \in [0, xmax]$, donde $xmax$ es un numero suficientemente grande para que la solucion ψ cumpla con las condiciones de frontera.
- Se puede construir la funcion de onda para x negativo usando simetria, dado que $\psi_n(-x) = (-1)^n \psi_n(x)$, lo cual es facilitado por la simetria del potencial de oscilador harmonico; de otro modo la integracion se tendria que dar sobre todo el intervalo $[-xmax, xmax]$
- El potencial de oscilador harmonico es

$$V(x) = \frac{1}{2}x^2 \rightarrow V_i = 0.5 * x_i^2$$

- Implementacion en Python

- **Busqueda de eigenvalores por biseccion**

- Cotas iniciales: $e_lower = \min(V(x))$, $e_upper = \max(V(x))$. Con el objetivo de encontrar E tal que la solucion dada por la formula de Numerov $\psi(x)$ sea fisica, e.g. suave, normalizable y cumple condiciones de frontera.

- **Calculo del punto de retorno clasico**

- Se determina el primer indice donde $V(x) > E$ usando la funcion auxiliar

$$f^{aux} = 2(V - E) \frac{\Delta x^2}{12}$$

- Detalles fisicos: Punto de retorno clasico

- **Integracion numerica de $\psi(x)$**

Se hacen dos integraciones

- Hacia afuera ($0 \rightarrow icl$): Se inicia ψ_0 y ψ_1 segun paridad y se propaga hasta ψ_{icl} con Numerov, asi propagando la solucion izquiera ψ^L
- Hacia adentro ($xmax \rightarrow icl$): Se impone la condicion de frontera $\psi(xmax) = 0$, luego se puede calcular ψ_{mesh-1} con la formula de Numerov y considerando que $\psi_{mesh+1} = 0$; asi propagando la solucion derecha ψ^R
- En general estas dos funciones tienen valores diferentes en $x_c = icl * \Delta x$
- Ver integracion

- **Acoplamiento y normalizacion**

Queremos que la solución dada por Numerov sea físicamente válida, e.g. continua y normalizada:

- En $x_c = icl * \Delta x$
 - * Las soluciones ψ^L y ψ^R generalmente no coinciden en amplitud
 - * Se escala ψ^R (asumiendo que ψ^L es la solución correcta) para garantizar continuidad en este punto de la malla

$$\psi^R \leftarrow \psi^R \cdot \frac{\psi_{icl}^L}{\psi_{icl}^R}$$

- Se calcula la norma, \mathcal{N} , de ψ numericamente, i.e. regla del trapecio, tomando en cuenta la simetría

$$\mathcal{N} = \int 2|\psi|^2 dx$$

y se normaliza

$$\psi \rightarrow \frac{\psi}{\int 2|\psi|^2 dx}$$

- Detalles de la implementación 4.5

- **Criterio de convergencia**

- Se calcula la discontinuidad de la derivada $\Delta\psi'$ en icl :
- Actualización de energía:
 - * Si $\Delta\psi' * \psi_{icl} > 0 \implies E$ es demasiado alto, entonces se actualiza la cota superior $e_upper = E$
 - * Si $\Delta\psi' * \psi_{icl} < 0 \implies E$ demasiado bajo, entonces se actualiza la cota inferior $e_lower = E$
- Detalles de convergencia

4.2 Malla

- Utilizamos la librería numpy para tener acceso a operaciones vectorizadas sobre los arreglos

```
import numpy as np
```

```
x, dx = np.linspace(0, xmax, mesh+1, retstep=True)
vpot = 0.5 * x**2 # Potencial de oscilador armonico
```

4.3 Punto de retorno clasico

Es el punto, x_{rc} , que marca el límite entre las regiones clásicamente permitida y prohibida:

- En $x < x_{rc}$ ($V(x) < E$), $\psi(x)$ oscila; los nodos se encuentran en esta región
- En $x > x_{rc}$ ($V(x) > E$), $\psi(x)$ decae exponencialmente

- Analizamos el comportamiento de f^{aux}

$$f^{aux} = \frac{2(V - E) * \Delta x^2}{12}$$

- $f^{aux} < 0$ en la region clasicamente permitida $V(x) < E$
- $f^{aux} > 0$ en la region prohibida $V(x) > E$
- El cruce $f^{aux} = 0$ coincide con $V(x_{rc}) = E$ (punto de retorno clasico exacto), pero como f^{aux} vive en el espacio discretizado, no esta garantizado que $f^{aux} = 0$ se cumpla, en otras palabras, no se garantiza que exista el indice exacto irc de modo que un punto en la malla $x_{rc} = irc * \Delta x$ haga que $f_{irc}^{aux} = 0$

4.3.1 ICL

- El indice icl es una aproximacion discreta al punto de retorno clasico x_{rc}
 - Corresponde al primer punto de la malla $x_c = icl * \Delta x$ donde $V > E$
 - El punto de retorno clasico exacto x_{rc} esta entre $x_c - \Delta x$ y x_c :

$$x_{rc} \in [x_c - \Delta x, x_c]$$

4.3.2 Implementacion

Funcion auxiliar

`f_aux = 2*(V-E) * (dx**2/12)`

Aseguramos que haya cambio de signo evitando el cero

`f_aux = np.where(f == 0.0, 1e-20, f_aux)`

Deteccion de cambios de signo en f_aux

`sign_changes = np.where(np.diff(np.sign(f_aux)))[0]` *# Devuelve los cambios de signo*

`icl = sign_changes[-1] + 1` *# Primer punto en la region prohibida*

4.3.3 Notas

- La funcion auxiliar f^{aux} nos proporciona una relacion simple para determinar icl . **La formula de Numerov usa:**

$$f = 1 - f^{aux}$$

- En el raro caso en el que f^{aux} sea exactamente igual a cero, no se detecta correctamente el cambio de signo. Utilizamos la funcion `np.where(condition, x, y)` para asegurar que se puedan detectar los cambios de signo correctamente:
 - Es una funcion vectorizada que actua como un *if-else* sobre arreglos de NumPy
 - Para cada elemento en f^{aux}

$$\begin{cases} \text{Si } f_i^{aux} = 0.0, & f_i^{aux} \rightarrow 1e-20 \\ \text{Si } f_i^{aux} \neq 0.0, & f_i^{aux} \rightarrow f_i^{aux} \end{cases}$$

- Cuando se usa sin los argumentos (x, y) , la funcion $np.where(condition)$ devuelve una tupla de arreglos con los indices donde $condition$ es $True$. Para arreglos de $1D$, el primer elemento de la tupla es el arreglo de estos indices, por ejemplo

`zeros_index = np.where(f == 0.0)[0] # Arreglo que contiene los indices donde`

- El objetivo es encontrar el primer punto donde f^{aux} cambia de negativo a positivo, i.e. la transicion de la region permitida a la prohibida. Se implementa en tres pasos:
 - Calculo de signos

`signs = np.sign(f_aux)`

Devuelve un arreglo con los signos de f^{aux}

$$\begin{cases} signs[i] = -1 & \text{si } f_i^{aux} < 0 \\ signs[i] = +1 & \text{si } f_i^{aux} > 0 \\ signs[i] = 0 & \text{si } f_i^{aux} = 0 \end{cases}$$

- Luego:

`diffs = np.diff(signs)`

Es un arreglo con las diferencias entre elementos adyacentes de $signs$. Es cero cuando no hay cambio de signo entre f_i^{aux} y f_{i+1}^{aux} , cuando cambia de negativo a positivo, $diffs[i] = +2$

- Finalmente, usamos el arreglo $diffs$ como condiciones

`sign_changes = np.where(diffs)[0]`

cuando $diffs[i]$ es positivo, es como pasar $True$ y guarda el indice i en el primer arreglo de la tupla que devuelve $np.where()$. Accedemos al arreglo con el primer elemento de la tupla.

- Asi, el primer elemento de f^{aux} en la region prohibida es

`icl = sign_changes[-1] + 1`
`f_aux[icl]`

4.4 Integracion

4.4.1 Condiciones iniciales hacia afuera

La paridad de la funcion de onda determina los puntos iniciales para la recursion hacia adentro, se tiene que:

$$\psi_n(-x) = (-1)^n \psi_n(x)$$

- Para n impar, los primeros dos puntos se inician como:
 - $\psi_0 = 0$

- $\psi_1 = dx$ es un numero apropiadamente pequeno para las dimensiones del problema
- Para n par:
 - $\psi_0 = 1$ es un numero arbitrario positivo apropiado para las dimensiones del problema. La magnitud es arbitraria ya que procedemos a normalizar la solucion
 - ψ_1 se determina por la formula de Numerov 2

$$\psi_1 = \frac{(12 - 10f_0\psi_0) - f_{-1}\psi_{-1}}{f_1}$$

donde f_{-1} y ψ_{-1} son el valor de f y ψ en $x_{-1} = -\Delta x$, pero por simetria, se tiene que $(f_1, \psi_1) = (f_{-1}, \psi_{-1})$ para obtener

$$\begin{aligned}\psi_1 &= \frac{(12 - 10f_0\psi_0) - f_1\psi_1}{f_1} \\ f_1\psi_1 + f_1\psi_1 &= (12 - 10f_0\psi_0) \\ \psi_1 &= \frac{(12 - 10f_0\psi_0)}{2f_1}\end{aligned}$$

4.4.2 Nodos y validacion de energia

- La solucion ψ^L contiene todos los nodos de $\psi(x)$, ya que ψ^R decae exponencialmente sin oscilar.
 - Antes de acoplar ψ^L y ψ^R , se verifica si ψ^L tiene el numero correcto de nodos para el n -esimo eigenvalor. Si no coincide, se ajustan las cotas de energia (E_{min}, E_{max}) y se reinicia el ciclo de biseccion.
- La funcion *outward_outward* integration devuelve el conteo de cruces por cero (*ncross*) para la validacion
- Para este paso consideramos que ψ es la solucion en $[0, xmax]$, entonces el numero correcto de nodos se obtiene por simetria y considerando la paridad:

```
# Ajuste de nodos por simetria basado en paridad del estado energetico
if nodes % 2 == 0:
    # Par
    ncross *= 2
else:
    # Impar
    ncross = 2 * ncross + 1

# Si los nodos no son correctos
# Actualizar cotas de energia
if ncross != nodes:
    if ncross > nodes:
        e_upper = e
    else:
        e_lower = e
    continue # Reiniciar ciclo de biseccion
```

4.4.3 Condiciones iniciales hacia adentro

Invertimos el orden de integracion, empezamos en la frontera hasta *icl*. La formula de Numerov funciona igual que la integracion hacia afuera, entonces necesitamos los ultimos dos puntos ψ_{malla} y $\psi_{malla-1}$:

- Para ψ_{malla} , si aplicamos directamente la condicion de frontera $\psi(x_{max}) = 0$, la recursion da un arreglo de puros ceros, entonces iniciamos con un valor arbitrariamente pequeno, apropiado para las dimensiones del problema, que en este caso es el paso de la malla

$$\psi_{malla} = dx$$

- Para $\psi_{malla-1}$ usamos directamente la formula de Numerov, con $\psi_{malla+1} = 0$

$$\psi_{malla-1} = \frac{(12 - 10f_{malla})\psi_{malla}}{f_{malla-1}}$$

Usando estos valores podemos propagar el resto de ψ^R hasta el punto de retorno clasico.

4.4.4 Implementacion

- Integracion hacia afuera:
 - Durante la integracion hacia adentro, el valor de ψ_{icl} (calculado previamente en la integracion hacia afuera) se sobrescribe. En general estos valores son diferentes y crean una discontinuidad en la solucion. Guardamos ψ_{icl}^L en una variable auxiliar *psi_icl* antes de iniciar la integracion hacia adentro

```
psi = np.zeros_like(x)
# Iniciacion de psi basado en paridad
if nodes % 2:
    # Impar
    psi[0] = 0.0
    psi[1] = dx
else:
    # Par
    psi[0] = 1.0
    psi[1] = (6.0 - 5.0 * f[0]) * psi[0] / f[1]

psi_icl, ncross = outward_integration(psi, f, icl)

def outward_integration(psi, f, icl):
    ncross = 0
    for i in range(1, icl):
        psi[i+1] = ((12.0 - 10.0)*f[i] * psi[i] - f[i-1] * psi[i-1]) / f[i]
        ncross += (psi[i] * psi[i+1] < 0.0) # Boolean to int
    return psi[icl], ncross
```

- Integracion hacia adentro:
 - La solución general cerca de *xmax* es una superposicion de exponenciales crecientes/decrecientes. El rescale suprime artificialmente la parte divergente, i.e. no física, preservando el decaimiento exponencial válido en la region classicamente prohibida

```

# Inward integration on the tail: initialize boundary conditions.
psi[-1] = dx
psi[-2] = f_10[-1] * psi[-1] / f[-2]

inward_integration(psi, f, icl, mesh)

def inward_integration(psi, f, icl, mesh):
    # Inward integration in [xmax, icl]
    for i in range(mesh-1, icl, -1):
        psi[i-1] = ((12.0 - 10.0)*f[i] * psi[i] - f[i+1] * psi[i+1]) / f[i]
        if abs(psi[i-1]) > 1e10:
            psi[i-1:-2] /= psi[i-1] # Rescale para suprimir comportamientos

```

4.5 Acoplamiento y normalizacion

4.5.1 Implementacion

- Es una simple funcion que empalma ψ^R con ψ^L en $x_c = icl * \Delta x$ escalando ψ^R por el factor

$$\frac{\psi_{icl}^L}{\psi_{icl}^R}$$

- Al tener una solucion continua ψ procedemos a normalizar:
 - Como estamos aprovechando la simetria para solo calcular una mitad de ψ , al normalizar tenemos que considerar que

$$\int |\psi|^2 dx = \frac{1}{2} \int |\psi_{full}|^2 dx$$

```

# Normalizar funcion de onda
scale_normalize_ho(psi, psi_icl, icl, x)

def scale_normalize_ho(psi, psi_icl, icl, x):
    # Match wavefunction at icl and normalize
    scaling_factor = psi_icl / psi[icl]
    psi[icl:-2] *= scaling_factor

    norm = np.sqrt(np.trapezoid(2*psi**2, x)) # Symmetric normalization
    psi /= norm

```

4.6 Criterio de convergencia

4.6.1 Determinacion de discontinuidad en la primer derivada

Al haber normalizado la funcion de onda, nuestra solucion, en general tendra una discontinuidad en su primera derivada, que podemos expresar como

$$\psi_{icl}^R - \psi_{icl}^L$$

Esta diferencia debe ser cero para una solucion apropiada, lo que solo ocurre cuando E esta muy cerca de ser un eigenvalor E_n . El signo de la diferencia nos ayuda a entender si la energia

de prueba E es muy alta o muy baja, para así hacer una actualización apropiada en el método de bisección.

Con $i = icl$, calculamos la discontinuidad en la primera derivada usando las expansiones de Taylor:

$$\begin{aligned}\psi_{i-1}^L &= \psi_i^L - \psi_i'^L \Delta x + \frac{1}{2} \psi_i''^L (\Delta x)^2 + \mathcal{O}((\Delta x)^3) \\ \psi_{i+1}^L &= \psi_i^L + \psi_i'^L \Delta x + \frac{1}{2} \psi_i''^L (\Delta x)^2 + \mathcal{O}((\Delta x)^3)\end{aligned}$$

Sumando estas dos expresiones, tomando en cuenta que al haber acoplado las recursiones, se tiene $\psi_i^L = \psi_i^R = \psi_i$, y que $\psi_i''^L = \psi_i''^R = -g_i \psi_i$, por el método de Numerov:

$$\psi_{i-1}^L + \psi_{i+1}^R = 2\psi_i + (\psi_i'^R - \psi_i'^L) \Delta x - g_i \psi_i (\Delta x)^2 + \mathcal{O}((\Delta x)^3)$$

esto es

$$\psi_i'^R - \psi_i'^L = \frac{\psi_{i-1}^L + \psi_{i+1}^R - [2 - g_i (\Delta x)^2] \psi_i}{\Delta x} + \mathcal{O}((\Delta x)^2)$$

en términos de f , obtenemos la expresión para la discontinuidad en la primera derivada:

$$\psi_i'^R - \psi_i'^L = \frac{\psi_{i-1}^L + \psi_{i+1}^R - [14 - 12f_i] \psi_i}{\Delta x} + \mathcal{O}((\Delta x)^2)$$

4.6.2 Lógica de bisección

La discontinuidad en la derivada

$$\psi_i'^R - \psi_i'^L \equiv d\delta$$

indica cómo la solución ψ se desvía del decaimiento exponencial físico en $x > x_{rc}$. Su signo determina si E es demasiado alta o baja.

- Para que ψ sea una solución física:
 - En la región clásicamente prohibida $x > x_{rc}$, la solución correcta debe decaer exponencialmente
 - Para que ψ decaiga exponencialmente en la región clásicamente prohibida, la magnitud de ψ'^R debe ser mayor que la de ψ'^L , lo que implica que la pendiente de la recta tangente en esta región es más pronunciada.
- El signo de ψ_i está determinado por la paridad del estado energético, con paridad par, $\psi_i > 0$ y con paridad impar $\psi_i < 0$. Luego el signo de $d\delta$ nos dice:

Paridad ψ_i	$d\delta > 0$	$d\delta < 0$
Par	Decae muy lento (E muy alta)	Decae muy rápido (E muy baja)
Impar	Decae muy rápido (E muy baja)	Decae muy lento (E muy alta)

- El producto $d\delta * \psi_i$ codifica estos cuatro casos correctamente:
 - Si $d\delta * \psi_i > 0$, ajustamos las cotas a la mitad inferior del intervalo
 - Si $d\delta * \psi_i < 0$ ajustamos las cotas a la mitad superior del intervalo

- Se declara convergencia cuando:

$$e_upper - e_lower < tol$$

donde tol es una tolerancia arbitraria. La energia final es

$$E_n \approx 0.5 * (e_upper + e_lower)$$

4.6.3 Implementacion

```
# Compute the derivative discontinuity at the matching point
ddelta = (psi[icl+1] + psi[icl-1] - (14.0 - 12.0 * f[icl]) * psi[icl]) / dx

# Check convergence: update energy bounds based on the sign of the discontinuity
if (e_upper - e_lower) < tol:
    break

if ddelta * psi[icl] > 0.0:
    e_upper = e
else:
    e_lower = e
```

5 Algoritmo de biseccion para el atomo de hidrogeno

Para resolver la ecuacion radial modificamos ligeramente el algoritmo

- Al hacer la transformacion a coordenadas logaritmicas, ajustamos la malla

$$-x = \ln(Z * r), \text{ donde } r \text{ es la coordenada radial fisica}$$

- El potencial efectivo es

$$v_eff = -2*Z/r + l*(l + 1)/r**2$$

- Iniciacion de la funcion de onda tomando en cuenta su comportamiento asintotico

```
psi[0] = (r[0] ** (l + 1)) * (1 - (2 * Z * r[0]) / (2 * l + 2)) / np.sqrt(r[0])
psi[1] = (r[1] ** (l + 1)) * (1 - (2 * Z * r[1]) / (2 * l + 2)) / np.sqrt(r[1])

psi_icl, ncross = outward_integration(psi, f, f_10, icl)
```

- Usando ψ_0 y ψ_1 propagamos la solucion con el paso constante de la malla logaritmica dx usando la formula de Numerov
- Antes de hacer la integracion hacia adentro, consideramos que la solucion debe tener $n - l - 1$ nodos


```

# Checar que las cotas contienen el numero correcto de nodos
nodes_expected = n - 1 - 1
if ncross != nodes_expected:
    if ncross > nodes_expected:
        e_upper = e
    else:
        e_lower = e
    e = 0.5 * (e_lower + e_upper)
    continue # Saltarse la integracion hacia adentro si la cantidad de nodos

```

- Para la integracion hacia adentro, iniciamos los ultimos dos valores de la misma manera

```

psi[-1] = dx
psi[-2] = f_10[-1] * psi[-1] / f[-2]
inward_integration(psi, f, icl, mesh, f_10)

```

- La normalizacion se hace en coordenadas esfericas

```

norm = np.sqrt(np.trapezoid(psi**2 * r**2, x))
psi /= np.sqrt(norm)

```

6 Teoria de perturbaciones

Para mejorar la convergencia, podemos utilizar teoria de perturbaciones para obtener correcciones de energia, asi cambiamos el criterio de convergencia para cuando la correccion en la energia se mas pequena que la tolerancia, usamos la funcion:

```

def update_energy(icl, f, psi, dx, ddx12, e, e_lower, e_upper):
    # ddx12 es una constante auxiliar -> dx**2 / 12.0
    # f -> 1 + ddx12 * (lnhfsq + r2 * (e - vpot)) dada por Numerov
    # icl es el primer indice en la region prohibida
    i = icl
    psi_cusp = (psi[i-1] * f[i-1] + psi[i+1] * f[i+1] + 10 * f[i] * psi[i]) / 12
    dfcusp = f[i] * (psi[i] / psi_cusp - 1.0)
    de = dfcusp / ddx12 * (psi_cusp ** 2) * dx

    if de > 0:
        e_lower = e
    elif de < 0:
        e_upper = e
    e += de
    e = max(min(e, e_upper), e_lower)
    return e, e_lower, e_upper, de # Devuelve la energia corregida y nuevas cotas

```

- Usando la formula de Numerov, calculamos ψ_{icl} tomando $icl - 1$ o $icl + 1$ como punto central, luego tenemos la cantidad $psicusp$ que es el valor calculado usando icl como punto central. En general $psicusp \neq \psi_{icl}$

- Usando teoria de perturbaciones, consideramos a la funcion donde $psicusp \neq \psi_{icl}$ como la solucion exacta a un problema diferente; uno en el que se superimpone una funcion delta $v_0\delta(x - x_c)$ en $x_c \equiv x_{icl}$ (porque estamos en la malla logaritmica) al potencial. La presencia de una delta causa una discontinuidad en la primera derivada y el tamano de la discontinuidad esta relacionada al coeficiente de la delta. Al conocer este coeficiente, podemos dar una estimacion, basada en teoria de perturbaciones, de la diferencia entre el eigenvalor actual (para este potencial diferente) y el eigenvalor para el potencial que nos interesa
- Para lidiar con la delta en una integracion numerica, asumimos que solo tiene un valor en el intervalo Δx centrado en ψ_{icl} .
- La formula de Numerov que usamos es

$$\psi_{i+1}f_{i+1} = (12 - 10f_i)\psi_i - f_{i-1}\psi_{i-1}$$

donde normalmente extraemos ψ_{i+1} dividiendo entre f_{i+1} , pero ahora suponemos que f_{icl} tiene un valor diferente que no conocemos f_{cusp} , de modo que nuestra funcion satisface la formula de Numerov tambien en un punto icl . Entonces se debe cumplir lo siguiente:

$$f_{cusp} * psicusp = f_{icl} * \psi_{icl}$$

ya que este producto esta dado por el metodo de Numerov integrando desde $icl - 1$ hasta $icl + 1$ y $psicusp$ es el valor que la funcion ψ debe tener para que satisfaga la formula de Numerov tambien en icl . Como consecuencia, el valor de df_{cusp} es solamente $f_{cusp} - f_{icl}$, o bien δf

- El siguiente paso es calcular la variacion δV del potencial $V(r)$ correspondiente a δf . De la ecuacion de Numerov $\psi''(x) = g(x)\psi(x)$, para la ecuacion radial tenemos

$$g(x) = \frac{2m}{\hbar^2}r^2(x)(E - V(r(x))) - (l + \frac{1}{2})^2$$

diferenciando se obtiene

$$\delta g(x) = -(2m/\hbar^2)r^2\delta V$$

dado que $f(x) = 1 + g(x)(\Delta x^2/12)$, tenemos $\delta g = (12/\Delta x^2)\delta f$, por lo que

$$\delta V = -\frac{\hbar^2}{2m} \frac{1}{r^2} \frac{12}{\Delta x^2} \delta f$$

- Teoria de perturbaciones a primer orden da la variacion del eigenvalor correspondiente

$$\delta e = \langle \phi | \delta V | \phi \rangle = \int |\psi(x)|^2 r(x)^2 \delta V dx$$

donde ϕ es la solucion radial completa y ψ es la funcion auxiliar que estamos trabajando

- Escribimos la integral como una suma finita sobre puntos de la malla, con una sola contribucion diferente de cero proveniente de la region Δx centrada en el punto $x_c = x_{icl}$. Finalmente la correccion de energia:

$$\delta e = |\psi(x_c)|^2 r(x_c)^2 \delta V \Delta x = -\frac{\hbar^2}{2m} \frac{12}{\Delta x^2} |\psi(x_c)|^2 \Delta x \delta f$$

esta es la expresion que utilizamos para calcular diferencia entre el eigenvalor para el potencial con la delta superimpuesta y el potencial correcto. Ya que en el primer paso, esta aproximacion puede tener errores muy grandes, usamos

$$e = \max(\min(e, e_{\text{upper}}), e_{\text{lower}})$$

para que la nueva energia no se salga de las cotas de energia.

7 Central Potentials

$$H = \left[-\frac{\hbar^2}{2m} \nabla^2 + V(r) \right]$$

7.1 Radial equation

asdd

La probabilidad $p(r) dr$ de encontrar a una partícula a una distancia entre r y $r + dr$ del centro esta dada por la integracion sobre solamente las variables angulares del cuadrado de la funcion de onda

$$p(r) dr = \int_{\Omega} |\psi_{nlm}(r, \theta, \phi)|^2 r d\theta r \sin \theta d\phi dr = |R_{nl}|^2 r^2 dr = |\chi_{nl}|^2 dr$$

donde introducimos la funcion auxiliar $\chi(r)$, que se conoce como funcion de onda orbital

$$\chi(r) = rR(r)$$

como consecuencia de la normalizacion de los harmonicos esfericos

$$\int_0^{2\pi} d\phi \int_0^\pi d\theta |Y_{lm}(\theta, \phi)|^2 \sin \theta = 1,$$

la condicion de normalizacion para χ es

$$\int_0^\infty |\chi_{nl}(r)|^2 dr = 1.$$

Esto significa que la funcion $|\chi(r)|^2$ se puede interpretar directamente como la densidad de probabilidad radial. Entonces escribimos la ecuacion radial para $\chi(r)$ en vez de para $R(r)$

$$-\frac{\hbar^2}{2m} \frac{d^2 \chi}{dr^2} + \left[V(r) + \frac{\hbar^2 l(l+1)}{2mr^2} - E \right] \chi(r) = 0 \quad (8)$$

esta es la forma de la ecuacion de Schrodinger unidimensional para una partícula bajo un potencial efectivo

$$\hat{V}(r) = V(r) + \frac{\hbar^2 l(l+1)}{2mr^2}$$

7.2 Malla Logaritmica

$$x = x(r)$$

La relacion entre la malla con paso constante Δx y la malla de paso variable esta dada por

$$\Delta x = x'(r) \Delta r$$

La malla logaritmica toma la forma

$$x(r) \equiv \ln\left(\frac{Zr}{a_0}\right)$$

y se obtiene

$$\Delta x = \frac{\Delta r}{r}$$

La razon $\Delta r/r$ se mantiene constante en la malla de r . Al transformar la ecuacion 8 en la nueva variable x

- Expresando las derivadas con respecto a r en terminos de derivadas con respecto a x . Dado que:

$$x = \ln\left(\frac{Zr}{a_0}\right) \implies r = \frac{a_0}{Z}e^x$$

- La primera derivada de χ con respecto a r es

$$\frac{d\chi}{dr} = \frac{d\chi}{dx} \cdot \frac{dx}{dr}$$

luego

$$\begin{aligned} \frac{dx}{dr} &= \frac{d}{dr} \ln\left(\frac{Zr}{a_0}\right) = \frac{1}{r} \\ \frac{d\chi}{dr} &= \frac{1}{r} \frac{d\chi}{dx} \\ \frac{d^2\chi}{dr^2} &= \frac{1}{r^2} \frac{d\chi}{dx} + \frac{1}{r} \frac{d}{dr} \left(\frac{d\chi}{dx} \right) \\ \frac{d}{dr} \left(\frac{d\chi}{dx} \right) &= \frac{d^2\chi}{dx^2} \cdot \frac{dx}{dr} = \frac{1}{r} \frac{d^2\chi}{dx^2} \\ \frac{d^2\chi}{dx^2} &= -\frac{1}{r^2} \frac{d\chi}{dx} + \frac{1}{r^2} \frac{d^2\chi}{dx^2} = \frac{1}{r^2} \left(\frac{d^2\chi}{dx^2} - \frac{d\chi}{dx} \right) \end{aligned}$$

- Sustituyendo en la ecuacion 8

$$\begin{aligned} -\frac{\hbar^2}{2m} \frac{1}{r^2} \left(\frac{d^2\chi}{dx^2} - \frac{d\chi}{dx} \right) + \left[V(r) + \frac{\hbar^2 l(l+1)}{2mr^2} - Er^2 \right] \chi &= 0 \\ -\frac{\hbar^2}{2m} \left(\frac{d^2\chi}{dx^2} - \frac{d\chi}{dx} \right) + \left[V(r)r^2 + \frac{\hbar^2 l(l+1)}{2m} - Er^2 \right] \chi &= 0 \\ -\frac{\hbar^2}{2m} \frac{d^2\chi}{dx^2} + \frac{\hbar^2}{2m} \frac{d\chi}{dx} + \left[V(r)r^2 + \frac{\hbar^2 l(l+1)}{2m} - Er^2 \right] \chi &= 0 \end{aligned}$$

- Expresando r en terminos de x

$$-\frac{\hbar^2}{2m} \frac{d^2\chi}{dx^2} + \frac{\hbar^2}{2m} \frac{d\chi}{dx} + \left[V\left(\frac{a_0}{Z}e^x\right) \left(\frac{a_0}{Z}\right)^2 e^{2x} + \frac{\hbar^2 l(l+1)}{2m} - E\left(\frac{a_0}{Z}\right)^2 e^{2x} \right] \chi = 0$$

- En esta ecuacion aparece un termino de la primera derivada

$$\frac{\hbar^2}{2m} \frac{d\chi}{dx}$$

en consecuencia, no se pueden utilizar los metodos de integracion convencionales para esta ecuacion.

Para obtener una ecuacion que se pueda integrar sobre una malla logaritmica, tenemos que hacer la transformacion

$$y(x) = \frac{1}{\sqrt{r}} \chi(r(x))$$

$$\frac{d^2 y}{dx^2} + \left[\frac{2m_e}{\hbar^2} r^2 (E - V(r)) - \left(l + \frac{1}{2} \right)^2 \right] y(x) = 0$$