IMPLEMENTACIÓN DEL MÉTODO DE PRÜFER Y DIFERENCIAS FINITAS PARA RESOLVER LA ECUACIÓN DE SCHRÖDINGER: TEORÍA Y APLICACIÓN

Pontificia Universidad Católica de Chile
Estudiante: Nicolás Ulloa Gatica
Profesor: Rafael Benguria Donoso

I. Resumen

El presente trabajo tiene como objetivo proporcionar lineamientos básicos para obtener aproximaciones numéricas a valores propios de las funciones de onda correspondientes a la ecuación de Schrödinger (independiente en el tiempo). Para ello, se emplea el método de Prüfer, el cual transforma la ecuación diferencial parcial en dos ecuaciones diferenciales ordinarias, permitiendo posteriormente obtener un mapeo T cuya preimagen corresponde a los valores propios E_k (sección II.A). A continuación, mediante el método de diferencias finitas (sección II.B y II.C), se desarrolla un algoritmo que resuelve el problema de forma numérica (sección III). Finalmente, se muestra una aplicación práctica en el lenguaje de programación *Python* (sección IV).

II. Método

A. Transformacón de Prüfer

Se tiene la siguiente ecuación: diferencial[1]:

$$-u'' + Vu = Eu \tag{1}$$

La transformación de Prüfer consiste en escribir la función independiente u como la multiplicación de otras dos funciones independientes, como se muestra a continuación,

$$u(x) = \rho(x) \cdot \sin(\theta(x))$$

con $\rho(x) > 0$. Luego, la primera y segunda derivada de u respecto a x serán:

$$u' = \rho' \sin\theta + \rho\theta' \cos\theta$$

$$u'' = (\rho'' - \theta'^2)\sin\theta + (\rho\theta'' + 2\rho'\theta')\cos\theta$$

Al evaluar u'' en la ecuación (1), se obtiene:

$$-(\rho''-\theta'^2)\sin\theta-(\rho\theta''+2\rho'\theta')\cos\theta+V\rho\sin\theta = E\rho\sin\theta$$

Esto último se puede reescribir como una ecuación para los términos que acompañan a $\sin\theta$ y otra ecuación para los términos que acompañan a $\cos\theta$,

$$\begin{aligned} \sin\theta: &\qquad -(\rho'' - \rho\theta'^2) + V\rho = E\rho \\ \cos\theta: &\qquad (\rho\theta'' + 2\rho'\theta') = 0 \end{aligned}$$

[1]Notación: $f' = \frac{df}{dx}$

La ecuación asociada a $\cos\theta$ se puede reescribir como:

$$(\rho^2\theta')' = 0$$

Por tanto,

$$\rho^2\theta' = l$$

Con l una constante arbitraria. Este resultado se puede evaluar en la ecuación asociada a $\sin\theta$ y se obtiene:

$$-\rho'' + \frac{l^2}{\rho^3} + V\rho = E\rho$$

Así, el problema se traduce en resolver las siguientes dos ecuaciones,

$$\rho'' = \frac{l^2}{\rho^3} + V\rho - E\rho \tag{2}$$

$$\theta' = \frac{l}{\rho^2} \tag{3}$$

Al encontrar θ y ρ se puede obtener u. Luego, si se imponen las condiciones de borde $u(a) = u(b) = 0$ $a, b \in \mathbb{R}$, se sigue que

$$u(b) = 0 \;\Rightarrow\; \rho(b)\cdot\sin\theta(b) = 0$$

$$\Longleftrightarrow$$

$$\theta(b) = k\cdot\pi, \quad n \in \mathbb{N} \tag{4}$$

Este resultado se puede evaluar en la ecuación (3) y reescribir de tal forma que[2],

$$\frac{\theta(b)}{\pi} = \frac{1}{\pi}\int_a^b \frac{l}{\rho^2(E)}dx = k \tag{5}$$

Así, se obtiene un un mapeo en el que la imagen corresponde a $k \in \mathbb{N}$ y la preimagen corresponde a los k-ésimos valores propios:

$$T(E_k) = \frac{\theta(b)}{\pi} = \frac{1}{\pi}\int_a^b \frac{l}{\rho^2(E_k)}dx \tag{6}$$

Este mapeo puede ser construido a partir de $\theta(b)$, que a su vez depende de la función ρ. A continuación, se revisa el método de diferencias finitas que permite resolver numéricamente las ecuaciones (2) y (3) para, posteriormente, obtener el mapeo $T(E_k) : \mathbb{R} \mapsto \mathbb{N}$.

[2]Se escribió $\rho^2(E_k)$ para explicitar la dependencia de ρ a la variable independiente E.

B. Diferencias Finitas

El método de diferencias finitas consiste en aproximar las derivadas de una función utilizando diferencias sobre un dominio discreto. Por definición, la derivada de la función f es:

$$f'(x) = \lim_{h \to 0} \frac{f(x+h) - f(x)}{h}$$

La aproximación se obtiene al tomar un h muy cercano a cero:

$$f'(x) \approx \frac{f(x+h) - f(x)}{h}, \qquad h << 1 \tag{7}$$

Además, se tendrá una función y dominio discreto con dos condiciones iniciales para la función f. Por tanto, $x \in [a,b] \mid a,b \in \mathbb{R}$. De igual forma, $f(x_0) = f_0, \;\; f(x_1) = f_1 \mid f_0, f_1 \in \mathbb{R}$.

Luego, se toman intervalos regulares de medida h para definir los puntos que pertenecen al dominio. Esto se ilustra en la figura 1.

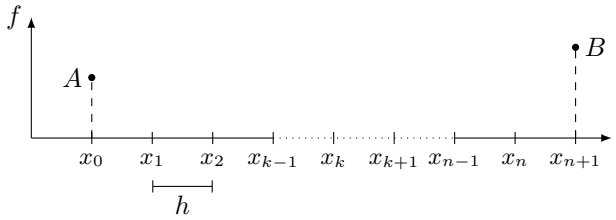

Figura 1: Discretización del intervalo $[a,b]$ en puntos equidistantes.

Así, con $k \in \{0, 1, ..., n+1\}$ donde $n \in \mathbb{N}$, se tiene que $x_k = k \cdot h$. Por tanto, $x_k + h = x_{k+1}$. Luego, la ecuación (7) se puede reescribir como:

$$f'(x_k) \approx \frac{f(x_{k+1}) - f(x_k)}{h} \tag{8}$$

De aquí en adelante, para explicitar que las funciones son discretas y facilitar la lectura del algoritmo, se utilizará la siguiente notación:

$$f[k] \equiv f(x_k)$$

Por tanto, si $f'[k]$ es la aproximación para $f'(x_k)$ dada por la ecuación (8), se tendrá que:

$$f'[k] = \frac{f[k+1] - f[k]}{h} \tag{9}$$

De igual forma, para la segunda derivada se tendrá:

$$\begin{aligned} f''[k] &= \frac{f'[k+1] - f'[k]}{h} \\ &= \frac{1}{h^2}\left(f[k+2] - f[k+1] + f[k] - f[k+1]\right) \\ &= \frac{1}{h^2}\left(f[k+2] - 2f[k+1] + f[k]\right) \end{aligned}$$

Es decir,

$$f''[k] = \frac{1}{h^2}\left(f[k+2] - 2f[k+1] + f[k]\right) \tag{10}$$

C. Aplicación diferencias finitas

A continuación se resolverán las ecuaciones (2) y (3) con el método de diferencias finitas. Utilizando la aproximación, la ecuación (2) se reescribe como:

$$\rho''[k] = \frac{1}{h^2}(\rho[k+2] - 2\rho[k+1] + \rho[k]) = \frac{l^2}{\rho^3[k]} + V\rho[k] - E\rho[k]$$

$$\Rightarrow$$

$$\rho[k+2] = \left(\frac{l^2}{\rho^3[k]} + V\rho[k] - E\rho[k]\right)h^2 + 2\rho[k+1] - \rho[k] \tag{11}$$

Con $\rho[0]$ y $\rho[1]$ valores iniciales conocidos. Para la ecuación (3) se tendrá que:

$$\theta'[k] = \frac{(\theta[k+1] - \theta[k])}{h} = \frac{l}{\rho^2[k]}$$

$$\Rightarrow$$

$$\theta[k+1] = \frac{lh}{\rho^2[k]} + \theta[k] \tag{12}$$

Con $\theta[0]$ un valor conocido. Así, se pueden obtener todos los valores de ρ y θ de forma sucesiva. Finalmente, se puede utilizar la ecuación (6) para encontrar el mapeo buscado,

$$T[E[k]] = \frac{\theta[b]}{\pi}$$

Es decir, se debe encontrar $\theta[b]$ para cada valor propio $E[k]$.

III. Algoritmo

En el algoritmo 1 se expone un algoritmo con el que se puede ejecutar computacionalmente el método estudiado. En la tabla I se encuentra la descripción de los parámetros utilizados y sugerencias para escogerlos. El algoritmo continuará la iteración y creará la secuencia T hasta algún n-ésimo valor propio de interés. El valor propio será el elemento de la secuencia E que esté en la posición correspondiente al largo de la secuencia T.

Algoritmo 1 Cálculo de T vs E

$V(x) \leftarrow \mathbf{v}(x)$
$h \leftarrow (x_{n+1} - x_0)/n$
$E \leftarrow [E_0, E_1, ..., E_m]$
$T \leftarrow [\ ]$
$i \leftarrow 0$
for all $\lambda \in E$ **do**
  $\rho \leftarrow [\rho_0, \rho_1]$
  $\theta \leftarrow [\theta_0]$
  $i \leftarrow i + 1$
  for all $k \in \{0, \ldots, n-3\}$ **do**

$$\rho[k+2] \leftarrow \left(\frac{l^2}{\rho^3[k]} + V(k \cdot h)\rho[k] - E\rho[k]\right) h^2 + 2\rho[k+1] - \rho[k]$$

  end for
  for all $k \in \{0, \ldots, n-2\}$ **do**

$$\theta[k+1] \leftarrow \frac{l \cdot h}{\rho^2[k]} + \theta[k]$$

  end for

$T[i] \leftarrow \frac{\theta[n-2]}{\pi}$

  if $T[i] \geq$ n_eigenvalue **then**
    break
  end if
end for

Parámetros	Descripción	Sugerencia
$\mathbf{v}(x)$	Función potencial	Comenzar con potencial nulo (partícula libre)
ρ_0, ρ_1	Primeros dos valores de la secuencia ρ	$\rho_0 = \rho_1 = 1$
θ_0	Primer valor de la secuencia θ	$\theta = 0$
x_0, x_{n+1}	Primer y último valor de la preimagen, respectivamente	Comenzar con $x_0 = 0$ y $x_{n+1} > 0$
$E_0, ..., E_m$	m valores para la secuencia E	Mientras más grande sea la secuencia, más suave será la curva $T(E)$ y mientras más pequeña sea la distancia entre los valores, más exacta será la aproximación de ρ y θ.
n	Cantidad de puntos con los que se realiza la aproximación	$n \geq 1000$
l	Constante de integración	$l = 1$
n_eigenvalue	Valor del enésimo valor propio buscado	Comenzar con 1 y luego modificar según necesidad

Tabla I: Parámetros para algoritmo 1

IV. Código

A continuación se muestra un módulo de Python en el que se ejecuta el algoritmo presentado, con leves adecuaciones al lenguaje y el output correspondiente:

```
import matplotlib.pyplot as plt
import numpy as np

n = 2000
xa = 0
xb = 1
hx = (xb-xa)/n
he = 0.01
n_valor_propio = 3

# Secuencia E con espaciado he
def secuenciaE(he):
    i = 0
    while True:
        yield i * he
        i += 1

# Aquí se define el potencial
def potencial(x):
    return 0

# Lista que contendrá los puntos de E
E = []

# Lista que contendrá los puntos de T
T = []

for i in secuenciaE(he):
    p = [1,1]
    thetha = [0]

    for k in range(0, n-2):
        pk2 = ((1/p[k]**3) + potencial(
            k*hx)*p[k] - i*p[k])*(hx**2)
             + 2*p[k+1] - p[k]
        p.append(pk2)

    for k in range(0, n-1):
        thetha1 = hx/(p[k]**2) + thetha
            [k]
        thetha.append(thetha1)

    E.append(i)
    T.append(thetha[-1]/np.pi)

    # Se detiene cuando se llega al
       valor propio buscado
    if T[-1] >= n_valor_propio:
        break

# Se imprimen los valores
print(f"Valor de energía con n = {round
    (T[-1])} es {E[-1]}")

# Crear la gráfica
plt.plot(E, T)
plt.grid(True)

# Añadir títulos y etiquetas
plt.title('Gráfica de T(E)')
plt.xlabel('E')
plt.ylabel('T')

# Mostrar la gráfica
plt.show()
```

A. *Output*

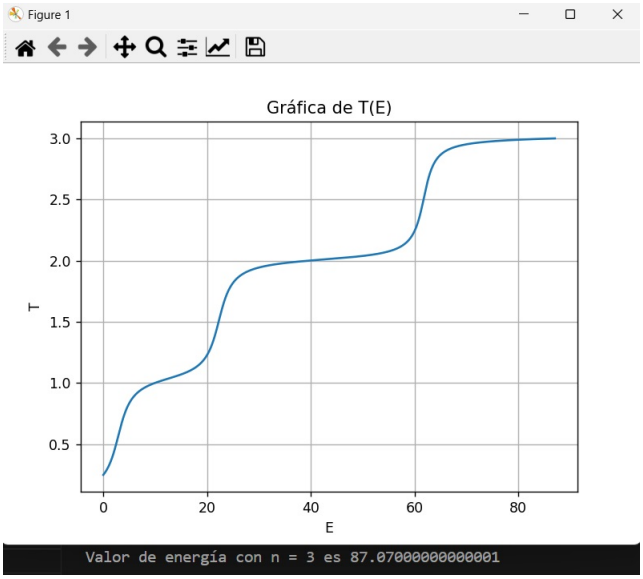

Figura 2: Resultado obtenido al ejecutar el código