PHY 115L
Wag-the-Dog

Michael Mulhearn

January 27, 2024

Chapter 1

Time-Independent Schrödinger Equation

We will numerically integrate the Time-Independent Schrödinger Equation (TISE):

$$-\frac{\hbar^2}{2m}\,\frac{d^2\psi}{dx^2} + V(x)\,\psi(x) = E\,\psi(x) \tag{1.1}$$

For numerical work, it is a good habit to introduce a system of units that keeps quantities near "1". We'll introduce a characteristic length a to be determined by the particular problem we are solving. Multiplying both sides by a^2 and rearranging we have:

$$a^2\,\frac{d^2\psi}{dx^2} = -\frac{2ma^2}{\hbar^2}\,\left(E - V(x)\right)\,\psi(x) \tag{1.2}$$

Defining:

$$E_0 \equiv \frac{\hbar^2}{2ma^2} \tag{1.3}$$

We have:

$$a^2\,\frac{d^2\psi}{dx^2} = -\frac{E - V(x)}{E_0}\,\psi(x) \tag{1.4}$$

We will integrate this equation using the Runge-Kutta technique for a first order differential equation, so we define a new variable ϕ by:

$$a\,\frac{d\psi}{dx} \equiv \phi \tag{1.5}$$

and so:

$$a\,\frac{d\phi}{dx} = -\frac{E - V(x)}{E_0}\,\psi(x) \tag{1.6}$$

In our computer programs, we'll measure x in units of a and E in units of E_0. In these units $a = 1$ and $E_0 = 1$, so our equations will read:

$$\frac{d\psi}{dx} \equiv \phi \tag{1.7}$$

and so:

$$\frac{d\phi}{dx} = \left(V(x) - E\right)\,\psi(x) \tag{1.8}$$

1.1 Euler's Method

We can write our system of first order differential equations by defining:

$$Y \equiv \begin{pmatrix}\psi\\ \phi\end{pmatrix} \tag{1.9}$$

and

$$\frac{dY}{dx} = \begin{pmatrix} \frac{d\psi}{dx} \\ \frac{d\phi}{dx} \end{pmatrix} = F(Y, x, E) \tag{1.10}$$

where:

$$F(Y, x, E) \equiv \begin{pmatrix} \phi \\ (V(x) - E)\ \psi \end{pmatrix} \tag{1.11}$$

for Euler's method, we approximate the change to Y during a step in x of size h as:

$$K_1 = h\frac{dY}{dx} = hF(Y, x, E)$$

and at each step we have:

$$Y \to Y + K_1$$

which have global error of h.

1.2 Fourth-Order Runge-Kutta Method

The Euler method is actually a 1st-order Runge-Kutta Method. The fourth order method samples the derivative column matrix F in more places:

$$\begin{aligned} K_1 &= h\ F(Y, x, E) \\ K_2 &= h\ F\left(Y + \frac{K_1}{2}, x + \frac{h}{2}, E\right) \\ K_3 &= h\ F\left(Y + \frac{K_2}{2}, x + \frac{h}{2}, E\right) \\ K_4 &= h\ F(Y + K_3, x + h, E) \end{aligned}$$

where:

$$Y \equiv \begin{pmatrix} \psi \\ \phi \end{pmatrix} \tag{1.12}$$

and

$$F(Y, x, E) \equiv \begin{pmatrix} \phi \\ (V(x) - E)\ \psi \end{pmatrix} \tag{1.13}$$

At each step we have:

$$Y \to Y + \frac{K_1 + 2\,K_2 + 2\,K_3 + K_4}{6}$$

1.3 The Infinite Square Well

An example python code for numerically integrating the TISE for the infinite square well using Euler's method is provide:

```
# System of Units:
# Position:  a = 1
# Energy:   E0 = hbar^2 / (2 m a^2)

# Potential V(x) in units of E0
def V(x):
    return 0

# TISE as two first order diff eqs:
# Y = (psi, phi)
# F = dY/dx = (dpsi/dx, dphi/dx)
# dpsi/dx = phi
# dphi/dx = (V-E) psi
def F(Y,x,E):
    psi = Y[0]
    phi = Y[1]
    dpsi_dx = phi
    dphi_dx = (V(x)-E)*psi
    F = np.array([dpsi_dx, dphi_dx], float)
    return F

# Numerical integration (using Runge-Kutta Order 1)
def tise_rk1(E,psi0,phi0,a,b,h):
    Y = np.array([psi0, phi0], float)
    X   = np.arange(a,b,h, float)
    PSI = np.array([psi0], float)
    for x in X:
        # 1st order Runge-Kutta:
        K1 = h*F(Y,x,E)
        Y += K1
        PSI = append(PSI,Y[0])
    X = np.append(X,b)
    return X,PSI

X,PSI = tise_rk1(E=20,psi0=1,phi0=0,a=0,b=0.5,h=0.01)
print("psi(b) = ", PSI[-1])

plt.plot(X,PSI,"b")
plt.axhline(c="k")
plt.axvline(x=0.5, c="k")
plt.ylim(-1.5,1.5)
plt.xlabel("x")
plt.ylabel("psi(x)")
```

1.4 The Harmonic Oscillator

For the harmonic oscillator with

$$\omega = \sqrt{\frac{k}{m}}$$

we take our characteristic length scale as:

$$a \equiv \sqrt{\frac{\hbar}{m\omega}}$$

and:

$$E_0 = \frac{\hbar^2}{2ma^2} = \frac{\hbar\omega}{2}$$

so that in our system of units:

$$\frac{V(x)}{E_0} = \frac{1}{2}m\omega^2\, x^2 \cdot \frac{2}{\hbar\omega} = \frac{x^2}{a^2}$$

The allowed energies in our system of units[1] are:

$$\frac{E_n}{E_0} = 2n + 1 \qquad\qquad n = 0, 1, 2, \ldots$$

1.5 Homework Problems

Problem 1: For an infinite potential well of width a, the allowed energies are

$$E = \frac{\pi^2\hbar^2 n^2}{2\,m\,a^2}$$

Determine the allowed energies in units of

$$E_0 = \frac{\hbar^2}{2\,m\,a^2}$$

(A) Use your own code or modify the example code to numerically integrate the TISE for $n = 1$ which should meet boundary condition $\psi(1/2) = 0$. Make a plot that shows $n = 1$ meets the boundary conditions but $n = 0.7$ and $n = 1.3$ do not.

(B) Create plots of for $n = 1, 3, 5$ with $h = 0.01$ You should notice that Euler's method is not stable: the amplitude of the wave function is changing!

(C) Make a copy of the function `tise_rk1` called `tise_rk4` and modify it to implement the 4th order Runge-Kutta method. You should not need to change much! Just the calculation of the K_i and the weighted average! Make a plot comparing the output of RK-1 and RK-4 for the same step size that illustrates the instability of Euler's method.

(D) As written, the software will only integrate an even solution. Modify your code so that it can handle odd solutions as well. In the same plot, show the *properly normalized wave functions* for $n = 1, 2, 3, 4, 5, 6$ from $x = -1/2$ to $x = 1/2$. Be clever about how you deduce the wave function in $[-1/2, 0]$.

Problem 2: In this problem, we'll apply RK-4 numerical integration to the harmonic oscillator potential. You should be able to reuse the same RK-4 function that you developed for Problem 1 without any changes. You will only need to change the definition of the potential to $V(x) = x^2$ in our system of units with $a = 1$.

(A) Numerically integrate the harmonic oscillator potential from $x = 0$ to $x = 5$ using $h = 0.01$ for the four lowest energy solutions. Take care to properly set the boundary conditions for even and odd solutions. Plot the normalized wave functions for these four solutions from $x = -5$ to $x = 5$.

[1] Notice that our choice of a has ensured that $E_n/E_0 = 1$ for $n = 0$

(B) Extend your numerical integration further, say, out to $x = 10$, plot the results, and show that your wave functions begin to diverge from the x-axis.

Just what is going on here? Notice that if we have for some x_0:

$$\psi(x_0) = \left.\frac{d\psi}{dx}\right|_{x_0} = 0$$

then the TISE shows that:

$$\psi(x) = 0$$

for all x. The wave functions at the allowed energies approach this behavior asymptotically, so that both $\psi(x)$ and $d\psi/dx$ go to zero. This is trivial to consider analytically (just the line $y = 0$) but it is a bit of a nightmare computationally, because round-off errors will always dominate as quantities approach zero. You should see that all of your solutions approach the asymptotic behavior, run along the x-axis for awhile, and only then begin to diverge (due to round off errors). Fortunately, we don't need to rely on our numerical integration to predict what happens after the wave function and its derivative become very small, because we already know what happens at that point all the way out to infinity. **Therefore, you are justified to simply cut off your numerical simulation once the wave function and derivative are small!**

Problem 3: So far, we have already known the allowed energies ahead of time, but how can we determine them using computational techniques in cases where we do not? Before considering that case, let's practice first on this potential. We'll use the "wag-the-dog" method. For each energy E we integrate the TISE out to $x = b$ for some large value of b. We note the sign of $\Psi(b)$, and find the energies E at which the sign of $\Psi(b)$ changes... like the tail of a dog wagging up and down. Numerically, we are finding the roots of the function:

$$f(E) \equiv \Psi_E(b)$$

To implement the wag-the-dog method, you can use the `scipy.optimize.bisect` function. You'll need to define a helper function (e.g. `f(E)`) that takes a single parameter E, then runs your numerical integration out to $x = b$, and returns $\psi(b)$. Note that the value $\psi(b)$ in the example function `tise_rk1` is available as `PSI[-1]` (i.e. the last entry in the array of ψ values returned by the numerical simulation). I find it convenient to define two helper functions `feven(E)` and `fodd(E)` which set up the numerical integration for even and odd solutions respectively. You'll also want to set the parameter `xtol=1E-4` to save time: we don't need `bisect` finding the roots out to 16 significant digits!
(A) Find the first six allowed energies of the harmonic oscillator potential using the wag-the-dog method.

Problem 4: In this problem we'll consider a potential for which we have not yet found the allowed energies, the saw tooth potential:

$$V(x) = |x|$$

in our units where $a = 1$.

(A) Find the first six allowed energies of the saw tooth potential using the wag-the-dog method.

(B) Plot the properly normalized wave functions for the first six allowed energies of the saw tooth potential.

These solutions are the Airy functions, which exhibit both exponential and sinusoidal behavior. They are used in the WKB approximation near the classical turning point, when the walls of the potential are not infinitely steep.