



UNIVERSITÀ DI PISA

# Metodi di soluzione numerica per l'equazione di Schrödinger

Candidata: Matilde Battisti

Tutor: Claudio Bonati

Ottobre 2024

# Problema generale e metodo delle differenze finite

- Perché usare una soluzione numerica?

Alcuni sistemi non hanno soluzione analitica per gli autovalori (e.g. buca di potenziale unidimensionale).

- Sistema dell'oscillatore armonico quantistico 1D:

$$H = \frac{\hat{p}^2}{2m} + \frac{1}{2}m\omega^2\hat{x}^2$$

$$\left[ -\frac{\hbar^2 \nabla^2}{2m} + \frac{1}{2}m\omega^2 x^2 \right] \psi = E\psi$$

$$E_n = \hbar\omega \left( n + \frac{1}{2} \right)$$

Si introduce il metodo delle **differenze finite**.

- Si risolve l'equazione su una maglia equispaziata:

$$a = x_0 < \dots < x_N = b$$

$$x_i = a + ih, h = \frac{b-a}{N}$$

- Si riscrive la derivata seconda in termini di differenze finite.  
Differenza finita in avanti:

$$\frac{d\psi(x)}{dx} \approx \frac{\psi(x+h) - \psi(x)}{h} \quad (1)$$

# Metodo delle differenze finite

Differenza finita indietro:

$$\frac{d\psi(x)}{dx} \approx \frac{\psi(x) - \psi(x-h)}{h} \quad (2)$$

- Derivata seconda come derivata della derivata prima usando  $\psi(x_i) = \psi_i$ ,  $\psi(x_i + h) = \psi_{i+1}$ ,  $\psi(x_i - h) = \psi_{i-1}$ :

$$\frac{d^2\psi(x)}{dx^2} \approx \frac{\psi_{i+1} + \psi_{i-1} - 2\psi_i}{h^2} \quad (3)$$

- Si riscrive Schrödinger usando l'approssimazione:

$$\frac{2\psi_i}{h^2} - \frac{\psi_{i+1} + \psi_{i-1}}{h^2} + q_i\psi_i = E'\psi_i \quad (4)$$

con  $q_i = \left(\frac{m\omega}{\hbar}x_i\right)^2$  ed  $E' = \frac{2mE}{\hbar^2}$ .

- Si deduce la forma dell'Hamiltoniana:

$$H = \begin{cases} \frac{2}{h^2} + q_i & j = i \\ -\frac{1}{h^2} & j = i+1, i-1 \\ 0 & \text{altrove} \end{cases}$$

# Applicazione del metodo delle differenze finite

Con la seguente porzione di codice si implementa la matrice Hamiltoniana

```
# Tridiagonal matrix definition
def EigenvaluesEigenvectors():
    N = MeshPoints()
    h = Spacing()
    q = PotentialEnergy()
    hbar = Hbar()
    m = Mass()
    H = np.zeros((N, N))

    for i in range(1, N-1):
        H[i, i-1] = -hbar**2 / (2 * m * h**2)
        H[i, i] = hbar**2 / (m * h**2) + q[i]
        H[i, i+1] = H[i, i-1]
    H[0, 0] = H[N-1, N-1] = 1e10
    return eigh(H)
```

I risultati ottenuti per  $N = 1000$  punti nella maglia sono i seguenti:

n	$E_{n,att}$	$E_n$	$\Delta E$
0	0.5	0.499997	3e-6
1	1.5	1.49998	2e-5

Table: Energie dei primi due stati

Ripetendo il calcolo a  $N$  diversi si osserva che l'errore diminuisce all'aumentare di  $N$ . Per  $N = 1000$  si ottengono le seguenti funzioni d'onda:

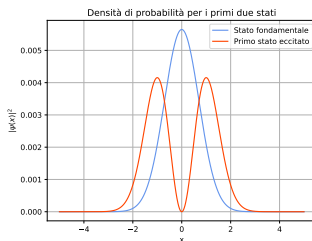


Figure: Densità di probabilità delle prime due funzioni d'onda

# Metodo di shooting

- Il metodo di **shooting** consiste nel trasformare una condizione al bordo in una condizione iniziale. Per la normalizzazione delle funzioni d'onda, si ha come condizione al bordo:

$$\psi(-\infty) = \psi(+\infty) = 0$$

- Si riscrive l'equazione di Schrödinger adimensionale:

$$\psi''(\tilde{x}) = [\tilde{x}^2 - 2\tilde{E}] \psi(\tilde{x})$$

$$\text{dove } \tilde{x} = x\sqrt{\frac{m\omega}{\hbar}}, \quad \tilde{E} = \frac{E}{\hbar\omega}.$$

- Computazionalmente, questa si risolve con la funzione **solve\_ivp** di Scipy

```
# Shooting function
def Shooting(E):
    X = LengthVariable()
    psi_0 = Psi0()
    psi_prime_0 = PsiPrime0()

    # Initial condition's array
    psi_in = [psi_0, psi_prime_0]

    # ODE's numerical solution with integration method for ivp
    sol = solve_ivp(Schrodinger, [X[0], X[-1]], psi_in, t_eval=X, args=(E,))

    return sol.y[0], sol.y[1] # Returning psi and psi'
```

- Si seleziona la funzione d'onda con la giusta energia tramite il metodo della **bisezione**

```
# Bisection method for energy guessing
def Energy():
    E_guess = EGuess()
    E_min, E_max = E_guess[0], E_guess[1]
    tol = 1e-6

    while E_max - E_min > tol:
        E_mid = (E_min + E_max) / 2
        psi, _ = Shooting(E_mid)

        # Shooting condition: we want psi close to 0 for x = x_max
        if psi[-1] * Shooting(E_min)[0][-1] < 0:
            E_max = E_mid
        else:
            E_min = E_mid

    return (E_min + E_max) / 2
```

# Metodo di shooting

- Si può verificare che vadano imposte le condizioni iniziali  $\psi_0 = 1$  e  $\psi'_0 = 0$  per  $n$  pari e  $\psi_0 = 0$  e  $\psi'_0 = 1$  per  $n$  dispari.
- Altrettanto importante è fornire un giusto guess per l'energia.

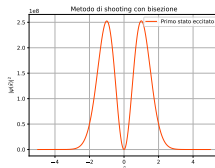
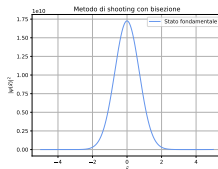
$$E_{guess,n=0} = [0., 2.]$$

$$E_{guess,n=1} = [1., 3.]$$

$n$	$E_{n,att}$	$E_n$	$\Delta E$
0	0.50	0.4999	1e-4
1	1.50	1.4996	4e-4

**Table:** Energie dello stato fondamentale e del primo eccitato

Densità di probabilità per lo stato fondamentale e per il primo eccitato:



# Bibliografia e ringraziamenti

- Pryce J.D., *Numerical solution of Sturm-Liouville Problems*, 52-56, Oxford Science Publications, **1993**
- Flannery, Press, Teukolsky, Vetterling, *Numerical Recipes in C, The Art of Scientific Computing*, 757-762, Cambridge University Press, **1997**
- Ballentine L. E., *Quantum Mechanics, A Modern Development*, 151-158, Simon Fraser University, **1998**

Grazie per l'attenzione

Ottobre 2024