

Università degli Studi di Pisa

FACOLTÀ DI SCIENZE MATEMATICHE, FISICHE E NATURALI

Corso di Laurea in Fisica

TESI DI LAUREA

**Risoluzione Numerica dell'Equazione di Schrödinger
Unidimensionale**

Candidato:

Guglielmo Gemignani

Relatore:

Chiarissimo Prof. Giovanni Moruzzi

Anno Accademico 2009-2010

A mia mamma

Indice

1	Il Metodo di Runge-Kutta	5
1.1	Cenni alle equazioni differenziali ordinarie	5
1.2	Il Metodo di Eulero	6
1.3	Il Metodo di Runge-Kutta	7
2	L'Oscillatore Armonico	9
2.1	L'Equazione di Schrödinger per l'Oscillatore Armonico	9
2.2	Livelli Energetici dell'Oscillatore Armonico	10
2.3	Funzioni d'Onda dell'Oscillatore Armonico	11
3	L'Ammoniaca	15
3.1	La Molecola NH_3	15
3.2	Il Potenziale dell'Ammoniaca	16
3.3	L'Equazione di Schrödinger per l' NH_3	17
3.3.1	Il Potenziale Polinomiale	17
3.3.2	Il Potenziale Iperbolico	18
4	Implementazione dei programmi	19
4.1	Implementazione del Programma per l'Oscillatore Armonico	19
4.2	Implementazione del Programma per l'Ammoniaca	21
4.2.1	Il Modello polinomiale	21
4.2.2	Il Modello Iperbolico	23
A	Il Codice dell'Oscillatore Armonico	25
B	Il Codice dell'Ammoniaca	27
B.1	Il Codice del Modello Polinomiale	27
B.2	Il Codice del Modello Iperbolico	29

Capitolo 1

Il Metodo di Runge-Kutta

Quello che verrà analizzato in questa sezione è un algoritmo chiamato *metodo di Runge-Kutta*, utilizzato per ricavare una soluzione numerica approssimata di equazioni differenziali ordinarie ad una variabile.

1.1 Cenni alle equazioni differenziali ordinarie

Ogni equazione differenziale di n-esimo grado può sempre essere ridotta ad un sistema di equazioni differenziali del primo ordine in modo da semplificare il problema. Prendiamo in considerazione un semplice esempio:

$$\frac{d^2y}{dx^2} + f(x)\frac{dy}{dx} = g(x) \quad (1.1)$$

È possibile ridurre questa equazione differenziale ad un semplice sistema:

$$\begin{cases} \frac{dy}{dx} = t \\ \frac{dt}{dx} = g(x) - f(x)t(x, y) \end{cases} \quad (1.2)$$

Quello di cui avremo bisogno dunque, nella risoluzione numerica di un nostro problema differenziale di grado n-esimo, sarà un sistema composto da n equazioni:

$$\frac{dy_i}{dx} = f_i(x, y_1, \dots, y_n) \quad (1.3)$$

con f_i nota a priori $\forall i$. Inoltre, trattandosi di un problema di cui vogliamo conoscere la soluzione numerica, avremo anche bisogno di condizioni iniziali per tutte le nostre variabili y_i . Ottenuto questo avremo dunque un classico problema di Cauchy che andremo a risolvere.

1.2 Il Metodo di Eulero

L'idea di base per la risoluzione numerica di un problema di Cauchy è l'approssimazione della funzione dal punto x_0 al punto $x_0 + \Delta x_0$ attraverso la derivata della funzione calcolata nel punto x_0 . Per fare ciò dovremo passare da una trattazione nel continuo ad una trattazione discreta, approssimando i dy e dx con piccoli incrementi finiti Δy e Δx . Una volta fatto ciò moltiplicando la nostra equazione per un fattore Δx faremo avanzare la funzione dal punto x_0 di partenza al punto $x_0 + \Delta x$. Tutto quello di cui bisognerà curarsi sarà dunque cercare di incrementare l'equazione di un Δx più piccolo possibile, in modo da ottenere una buona approssimazione dell'equazione ad ogni passo eseguito.

L'algoritmo di Runge-Kutta, come è stato sopra detto, è basato principalmente sul *metodo di Eulero*, il quale ci dice che l'equazione nel punto $x_0 + \Delta x$ può essere approssimata come la funzione calcolata nel punto x_0 , incrementata della sua derivata calcolata nello stesso punto e moltiplicata per un fattore Δx (fig. 1.1).

In formule:

$$y(x_0 + \Delta x) = y(x_0) + \Delta x \frac{dy(x_0, y(x_0))}{dx} \quad (1.4)$$

L'errore che si commette in questa operazione è di un ordine di grandezza minore dell'incremento Δx (ovvero $O(\Delta x^2)$).

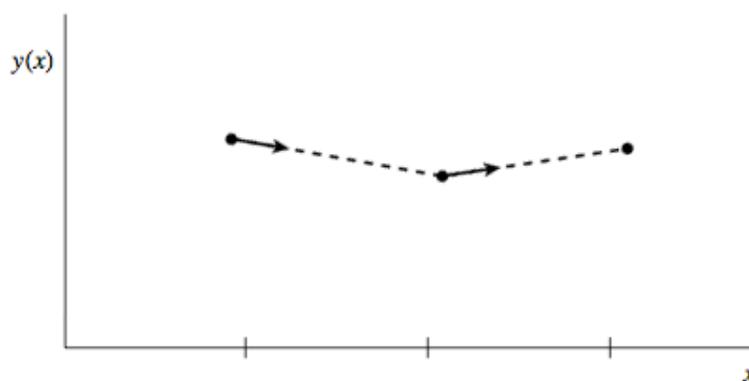


Figura 1.1: Metodo di Eulero. In questo basilare algoritmo per il calcolo numerico di equazioni differenziali ordinarie, viene calcolata la derivata ad un certo punto x_0 per ricavare il valore successivo della funzione.

1.3 Il Metodo di Runge-Kutta

Il metodo di Eulero descritto qui sopra, può essere migliorato andando a calcolare la derivata della funzione in due punti distinti del nostro incremento per aumentare la precisione del procedimento (fig.1.2). Questo algoritmo è chiamato *metodo di Runge-Kutta del secondo ordine* e si può riassumere tramite le seguenti equazioni (chiameremo $\Delta x = h$ e $y(x_0) = y_0$):

$$\left\{ \begin{array}{l} k_1 = h \frac{dy(x_0, y_0)}{dx} \\ k_2 = h \frac{dy(x_0 + \frac{h}{2}, y_0 + \frac{k_1}{2})}{dx} \\ y(x_0 + h) = y_0 + k_2 + O(h^3) \end{array} \right. \quad (1.5)$$

Come si può vedere dall'equazione (4.2) questo passaggio intermedio diminuisce l'errore di un ordine di grandezza (cioè $O(h^3)$) rendendolo appunto un algoritmo del secondo ordine.

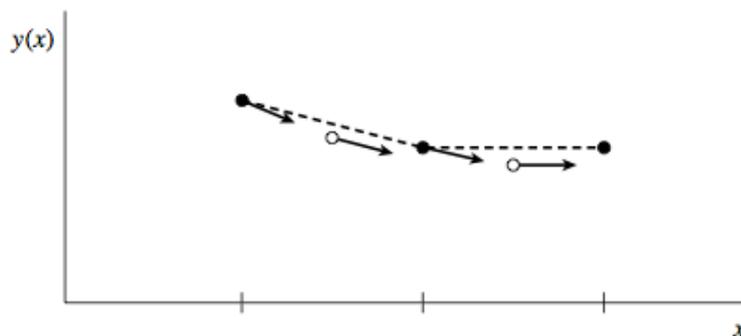


Figura 1.2: Metodo di Runge-Kutta del secondo ordine. In questo algoritmo è ottenuta una precisione del secondo ordine calcolando la derivata in due punti interni all'intervallo. Nella figura i cerchi pieni rappresentano i valori definitivi della funzione, mentre i cerchi vuoti rappresentano i punti a metà dell'intervallo.

L'algoritmo di gran lunga più usato per il suo buon compromesso tra numero di passaggi matematici da effettuare e grandezza dell'errore ottenuto, è il *metodo di Runge-Kutta di quarto ordine* che viene ricavato come il precedente di secondo ordine, andando a calcolare la derivata della funzione in tre punti distinti dell'incremento (fig.1.3). Le equazioni che definiscono

questo algoritmo sono date da:

$$\left\{ \begin{array}{l} k_1 = h \frac{dy(x_0, y_0)}{dx} \\ k_2 = h \frac{dy(x_0 + \frac{h}{2}, y_0 + \frac{k_1}{2})}{dx} \\ k_3 = h \frac{dy(x_0 + \frac{h}{2}, y_0 + \frac{k_2}{2})}{dx} \\ k_4 = h \frac{dy(x_0 + h, y_0 + k_3)}{dx} \\ y(x_0 + h) = y_0 + \frac{k_1}{6} + \frac{k_2}{3} + \frac{k_3}{3} + \frac{k_4}{6} + O(h^5) \end{array} \right. \quad (1.6)$$

Questo algoritmo è quello che verrà usato nel programma che ci permetterà di valutare la risoluzione dell'equazione di Schrödinger di diversi problemi fisici che andremo ad analizzare.

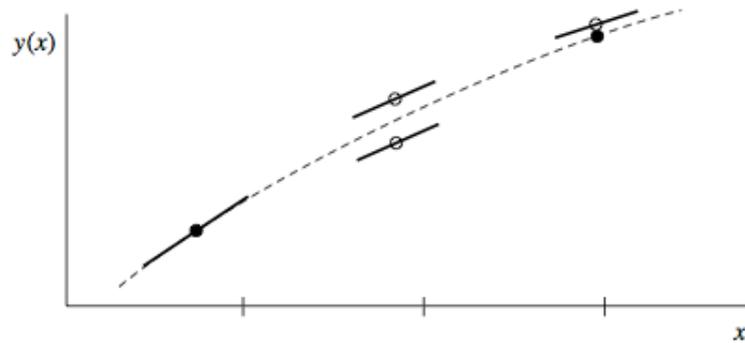


Figura 1.3: Metodo di Runge-Kutta del quarto ordine. In questo algoritmo è ottenuta una precisione del secondo ordine calcolando la derivata in quattro punti interni all'intervallo: uno al punto iniziale, due a metà dell'intervallo e un ultimo alla fine di esso. Nella figura i cerchi pieni rappresentano i valori definitivi della funzione, mentre i cerchi vuoti rappresentano i punti a metà dell'intervallo.

Capitolo 2

L'Oscillatore Armonico

L'oscillatore armonico è uno dei sistemi più importanti nella fisica quantistica per la sua possibile e completa risoluzione analitica e, dunque, per il suo utilizzo come approssimazione nei punti di equilibrio stabili di altri sistemi più complicati.

In questa sezione verrà brevemente ricavata ed analizzata la soluzione analitica del problema dell'oscillatore armonico unidimensionale, usata in seguito per la verifica dei risultati ottenuti dal programma scritto per questo elaborato.

2.1 L'Equazione di Schrödinger per l'Oscillatore Armonico

L'oscillatore armonico quantistico è l'analogo quantistico dell'oscillatore armonico classico, che è descritto dal potenziale:

$$V(x) = \frac{m\omega^2 x^2}{2} \quad (2.1)$$

avendo quindi un'hamiltoniana classica:

$$H = \frac{p^2}{2m} + \frac{m\omega^2 x^2}{2} \quad (2.2)$$

con m massa dell'oscillatore, ω impulso al quale esso oscilla e x spostamento dell'oscillatore.

L'equazione di Schrödinger, derivata dalla teoria quantistica, che dobbiamo risolvere per trovare i livelli energetici e le funzioni d'onda di questo problema è dunque:

$$-\frac{\hbar^2}{2m} \frac{d^2 \psi_n(x)}{dx^2} + \frac{m\omega^2 x^2}{2} \psi_n(x) = E_n \psi_n(x) \quad (2.3)$$

Per semplificare l'equazione e far lavorare il calcolatore con numeri puri, vogliamo ridurre questa equazione ad un'equazione adimensionale. Per fare questo introduciamo la variabile:

$$\xi = kx \quad (2.4)$$

con k variabile con dimensioni di una lunghezza che andremo a definire in seguito. Sostituendo la 2.4 nell'equazione 2.3 otteniamo:

$$-\frac{\hbar^2 k^2}{2m} \frac{d^2 \psi_n(\xi)}{d\xi^2} + \frac{m\omega^2}{2k^2} \left(\xi^2 - \frac{2k^2 E_n}{m\omega^2} \right) \psi_n(\xi) = 0 \quad (2.5)$$

e dividendo per il fattore moltiplicativo del secondo addendo otteniamo:

$$-\frac{\hbar^2 k^4}{\omega^2 m^2} \frac{d^2 \psi_n(\xi)}{d\xi^2} + \left(\xi^2 - \frac{2k^2 E_n}{m\omega^2} \right) \psi_n(\xi) = 0. \quad (2.6)$$

Ponendo infine $k = \sqrt{\frac{\omega m}{\hbar}}$, in modo da semplificare il fattore davanti alla derivata seconda, otteniamo l'espressione finale che andremo a risolvere:

$$\frac{d^2 \psi_n(\xi)}{d\xi^2} = \left(\xi^2 - \frac{2E_n}{\hbar\omega} \right) \psi_n(\xi) \quad (2.7)$$

2.2 Livelli Energetici dell'Oscillatore Armonico

Possiamo risolvere il problema dell'oscillatore armonico unidimensionale con il metodo di seconda quantizzazione, utilizzando gli operatori di creazione e di distruzione [1], definiti rispettivamente da:

$$a^\dagger = \sqrt{\frac{m\omega}{2\hbar}} x - i\sqrt{\frac{1}{2m\omega\hbar}} p \quad (2.8)$$

$$a = \sqrt{\frac{m\omega}{2\hbar}} x + i\sqrt{\frac{1}{2m\omega\hbar}} p. \quad (2.9)$$

È possibile esprimere gli operatori x e p in funzione di a e a^\dagger come:

$$x = \sqrt{\frac{\hbar}{2m\omega}} (a + a^\dagger) \quad (2.10)$$

$$p = -i\sqrt{\frac{\hbar}{2m\omega}} (a - a^\dagger) \quad (2.11)$$

per poi sostituirli nell'hamiltoniana e riscrivere anch'essa in funzione di a e a^\dagger , ottenendo:

$$H = \omega\hbar \left(a^\dagger a + \frac{1}{2} \right). \quad (2.12)$$

Usando gli elementi di matrice dei due operatori [1]:

$$\langle n-1|a|n\rangle = \sqrt{n}, \quad \langle n+1|a^\dagger|n\rangle = \sqrt{n+1} \quad (2.13)$$

se ne ricava che:

$$a^\dagger a|n\rangle = n|n\rangle \quad (2.14)$$

e dunque, definendo l'operatore *numero di occupazione* $N = a^\dagger a$, l'hamiltoniana che otteniamo è:

$$H = \omega\hbar \left(N + \frac{1}{2} \right) \quad (2.15)$$

e infine i livelli energetici che cercavamo:

$$E_n = \omega\hbar \left(n + \frac{1}{2} \right) |n\rangle \quad (2.16)$$

2.3 Funzioni d'Onda dell'Oscillatore Armonico

Attraverso gli operatori di creazione e distruzione, è possibile ricavare anche le funzioni d'onda dell'oscillatore armonico. Questo è possibile sfruttando il fatto che, applicando l'operatore di distruzione:

$$\langle x|a|0\rangle = 0 \quad (2.17)$$

essendo $|0\rangle$ il più basso stato possibile. Andiamo quindi a sostituire l'equazione 2.9, nell'equazione 2.17:

$$\left(\sqrt{\frac{m\omega}{2\hbar}} x + \sqrt{\frac{\hbar}{2m\omega}} \frac{d}{dx} \right) \psi_0(x) = 0 \quad (2.18)$$

Definiamo ora $x_0 = \sqrt{\frac{2\hbar}{m\omega}}$ e, andandola a sostituire dentro la 2.18 e semplificando, otteniamo:

$$x\psi_0(x) + x_0^2 \frac{d\psi_0(x)}{dx} = 0 \quad (2.19)$$

risolvendo ora l'equazione differenziale, otteniamo la funzione d'onda dello stato fondamentale:

$$\psi_0(x) = A e^{-\frac{x^2}{x_0^2}} \quad (2.20)$$

con $A = \sqrt[4]{\frac{m\omega}{\pi\hbar}}$ ottenuta per normalizzazione.

Le successive funzione d'onda possono essere calcolate utilizzando l'operatore di creazione sullo stato fondamentale e il risultato 2.20 ottenuto sopra. Il risultato generale per $\psi_n(x)$ ottenuto iterando il procedimento è dunque (posto $x_0 = \sqrt{\frac{\hbar}{m\omega}}$):

$$\psi_n = \frac{H_n\left(\frac{x}{x_0}\right) e^{-\frac{x^2}{2x_0^2}}}{\pi^{\frac{1}{4}} \sqrt{2^n n! x_0}} \quad (2.21)$$

con H_n chiamato *polinomio di Hermite*, definito come:

$$H_n(x) = (-1)^n e^{x^2} \frac{d^n e^{-x^2}}{dx^n}. \quad (2.22)$$

I grafici delle prime funzioni d'onda ottenute qui sopra, che poi andremo a confrontare con quelle ottenute attraverso il metodo di Runge-Kutta, sono mostrate in figura 2.1 e 2.2.

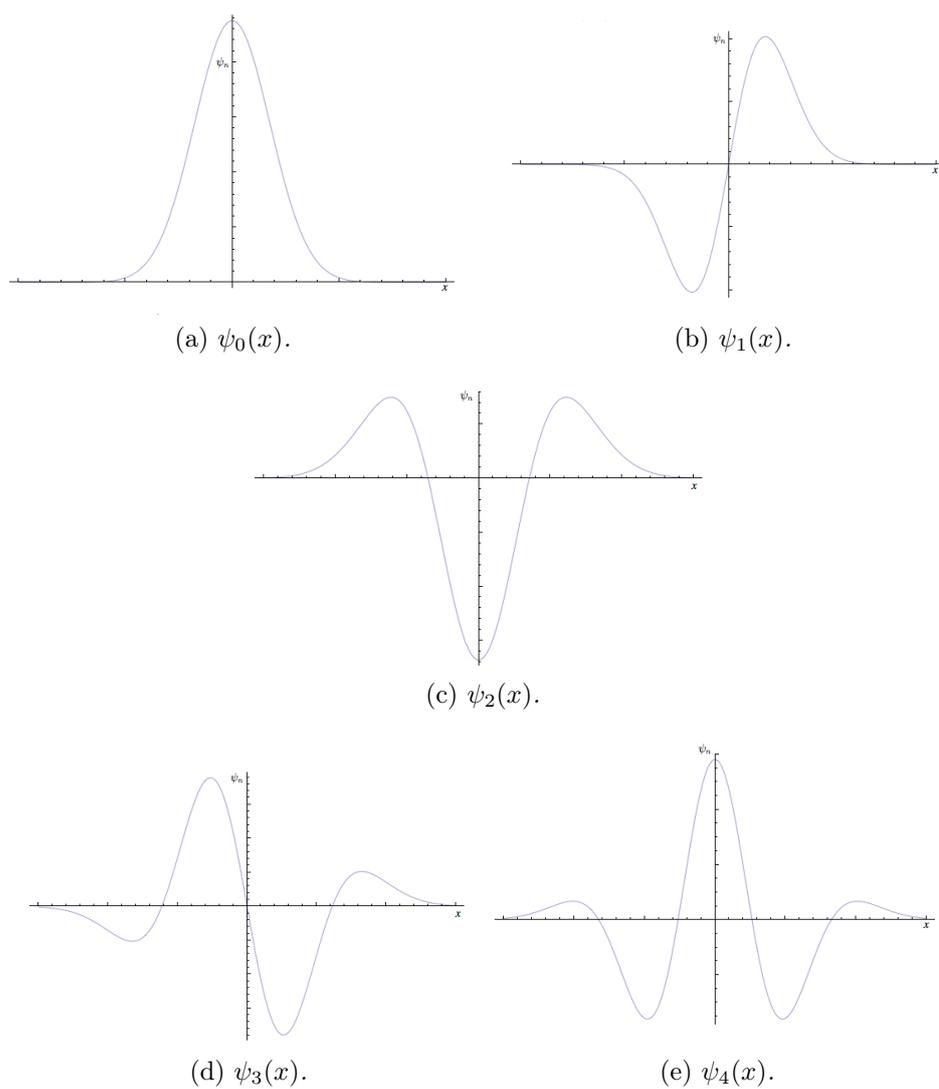


Figura 2.1: Le prime funzioni d'onda dell'oscillatore armonico unidimensionale.

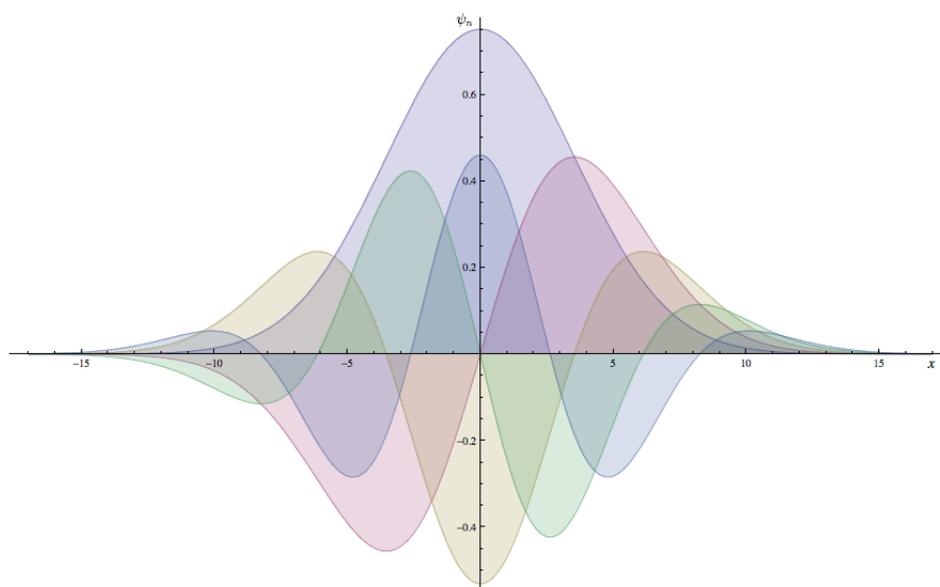


Figura 2.2: Andamento generale delle funzioni d'onda al crescere del parametro n .

Capitolo 3

L'Ammoniaca

In questo capitolo andremo ad analizzare vari aspetti dell'ammoniaca. Gli argomenti trattati saranno il comportamento della molecola, il suo potenziale e l'equazione di Schrödinger dipendente da esso.

3.1 La Molecola NH_3

La molecola dell'ammoniaca ha la caratteristica di essere una delle più semplici molecole con un potenziale *frenante* [3] e, per questo, è stata da sempre usata per capire meglio il movimento oscillatorio in esso compiuto. Le oscillazioni che avvengono sono dovute all'effetto tunnel quantistico che permette, grazie alla natura ondulatoria della particella, di penetrare in zone classicamente proibite dal punto di vista energetico. La sua struttura è a forma di piramide a base triangolare con ai vertici della base i tre atomi di idrogeno e sul vertice restante l'atomo di azoto, come mostrato in figura 3.1.

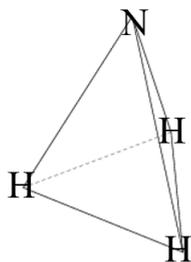


Figura 3.1: Struttura della molecola dell'ammoniaca.

Le frequenze date di solito dalle vibrazioni delle molecole, risiedono nella regione dell'infrarosso mentre il movimento dell' NH_3 è talmente rallentato dal suo potenziale che la frequenza di oscillazione cade nella regione delle microonde. Quando questo rallentamento avviene, lo spettro di energia è difficile da interpretare a causa dell'elevato numero di livelli energetici presenti in questa zona e, per questo, non è possibile trovare un'espressione analitica per gli autovalori dell'hamiltoniana della molecola.

Il potenziale in funzione della distanza dell'azoto dal piano, è descritto approssimativamente dal grafico in figura 3.2. Il

moto compiuto dall' NH_3 è costituito dallo spostamento dell'azoto perpendicolarmente al piano dove risiedono gli atomi di idrogeno, portando quindi, una volta sorpassato questo piano, all'inversione della molecola.

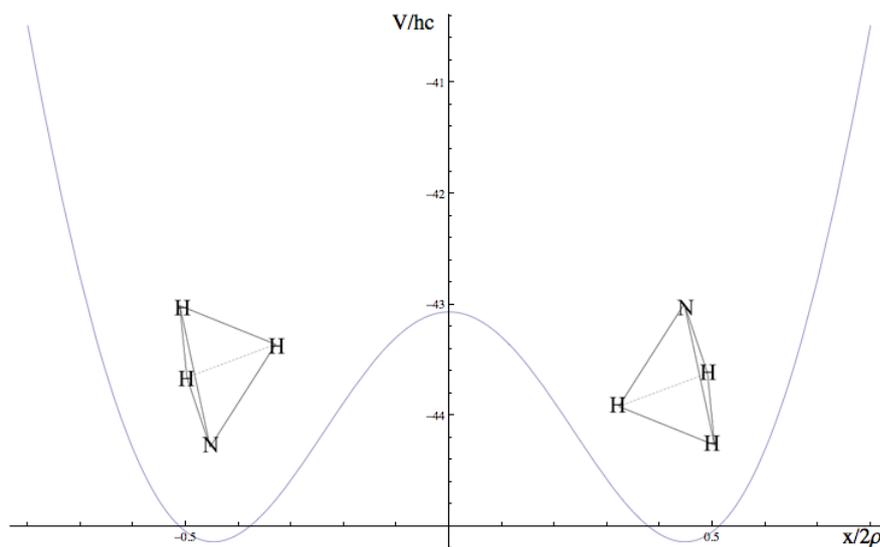


Figura 3.2: Potenziale dell' NH_3 . Viene rappresentato il potenziale dell'ammoniaca misurato in cm^{-1} in funzione della distanza tra l'azoto e il piano dove risiede l'idrogeno.

3.2 Il Potenziale dell'Ammoniaca

Diverse forme analitiche sono state proposte come modello per il potenziale dell' NH_3 per riprodurre al meglio i dati sperimentali. In questo lavoro è stata proposta un'approssimazione polinomiale:

$$\frac{V}{hc} = Ax^4 - Bx^2 - C \quad (3.1)$$

con A, B e C costanti numeriche determinate in seguito, confrontata poi con l'approssimazione di Manning fornita da Townes e Shawlow [3]:

$$\frac{V}{hc} = 66\,551 \operatorname{sech}^4\left(\frac{x}{2\rho}\right) - 109\,619 \operatorname{sech}^2\left(\frac{x}{2\rho}\right) \quad (3.2)$$

con $\frac{V}{hc}$ potenziale espresso in cm^{-1} , x la distanza tra l'atomo d'azoto e il piano dove giace l'idrogeno, $\rho = \frac{6.98 \cdot 10^{-8}}{\sqrt{\mu}}$ e $\mu = \frac{3mM}{3m + M}$ dove $m = 1.00794$ e

$M = 14,0067$ sono la massa rispettivamente dell'idrogeno e dell'azoto espresse in unità di massa atomica.

Questi due potenziali verranno utilizzati nella sezione successiva per ricavare le equazioni di Schrödinger da risolvere attraverso il metodo trattato nella sezione 1.3. La risoluzione di questo problema ci farà ottenere i livelli energetici della molecola e le sue funzioni d'onda, che verranno poi confrontate con i valori sperimentali.

3.3 L'Equazione di Schrödinger per l'NH₃

Avendo a disposizione il potenziale approssimato dalle funzioni proposte nella sezione 3.2, andiamo ora a ricavare le equazioni differenziali rispettive ai potenziali 3.1 e 3.2.

3.3.1 Il Potenziale Polinomiale

L'equazione di Schrödinger per un determinato sistema è data da:

$$-\frac{\hbar^2}{2\mu} \frac{d^2\psi_n(x)}{dx^2} + V\psi_n(x) = E_n\psi_n(x). \quad (3.3)$$

Dividendo per $\hbar c$ e sostituendo il potenziale 3.1 ricaviamo:

$$-\frac{h}{8\mu\pi^2c} \frac{d^2\psi_n(x)}{dx^2} + (Ax^4 - Bx^2 - C)\psi_n(x) = E_n\psi_n(x) \quad (3.4)$$

con E_n espressa in cm^{-1} , da cui otteniamo l'equazione:

$$\frac{d^2\psi_n(x)}{dx^2} = k(Ax^4 - Bx^2 - C - E_n)\psi_n(x) \quad (3.5)$$

con $k = \frac{8\mu\pi^2c}{h}$. Infine, per rendere l'equazione adimensionale e per riportarsi al caso iperbolico, per poi andare a confrontare i due risultati ottenuti, si è raccolto da tutte le costanti al secondo membro dell'equazione il termine $4\rho^2$, con $\rho = \frac{6.98 \cdot 10^{-8}}{\sqrt{\mu}}$ e μ massa ridotta dell'ammoniaca, ottenendo:

$$\frac{d^2\psi_n(x)}{dx^2} = 4k\rho^2(A'x^4 - B'x^2 - C' - E_n)\psi_n(x) \quad (3.6)$$

con $A' = 72\,422$, $B' = 23\,483$ e $C' = 43\,068$. Questo è quello che andremo a risolvere nella sezione 4.2.1

3.3.2 Il Potenziale Iperbolico

Come negli altri casi, usando il potenziale 3.2, ricaviamo l'equazione di Schrödinger:

$$-\frac{\hbar^2}{2\mu} \frac{d^2\psi_n(x)}{dx^2} + V\psi_n(x) = E_n\psi_n(x). \quad (3.7)$$

Da qui, dividendo tutto per $\hbar c$ e sostituendo $\frac{V}{\hbar c}$ otteniamo:

$$-\frac{\hbar}{8\mu\pi^2c} \frac{d^2\psi_n(x)}{dx^2} + \left[k_1 \operatorname{sech}^4\left(\frac{x}{2\rho}\right) - k_2 \operatorname{sech}^2\left(\frac{x}{2\rho}\right) \right] \psi_n(x) = E_n\psi_n(x) \quad (3.8)$$

con $k_1 = 66\,551$ e $k_2 = 109\,619$. Al solito vogliamo semplificare questa equazione rendendola adimensionale. Introduciamo quindi:

$$\xi = \frac{x}{2\rho} \quad (3.9)$$

andando quindi a sostituire dentro la 3.8 e, chiamando $k = \frac{8\mu\pi^2c}{\hbar}$, otteniamo:

$$\frac{d^2\psi_n(\xi)}{d\xi^2} = 4k\rho^2(k_1 \operatorname{sech}^4(\xi) - k_2 \operatorname{sech}^2(\xi) - E_n)\psi_n(\xi) \quad (3.10)$$

che è l'equazione differenziale che andremo a risolvere nella sezione 4.2.2.

Capitolo 4

Implementazione dei programmi

Questa sezione tratterà il modo in cui sono stati creati gli algoritmi per la risoluzione del problema dell'oscillatore armonico e della molecola dell'ammoniaca utilizzando, in questo caso, due diversi modelli per il potenziale.

4.1 Implementazione del Programma per l'Oscillatore Armonico

Il corpo principale è dato dal file `oscillatore.cc` che è diviso principalmente in tre sezioni:

- La prima parte è composta dal richiamo di tutte le funzioni ausiliarie per l'impostazione dell'ambiente grafico X11 e la definizione di tutti i parametri che serviranno nei vari calcoli durante l'esecuzione dell'algoritmo.
- Nella seconda parte viene disegnato il potenziale sulla finestra bianca creata nel primo punto, inserendo in successione i vari punti della funzione fornitagli e collegandoli con un segmento attraverso la funzione `Drawline`.
- Nell'ultima parte viene calcolato l'autovalore attraverso il metodo dello *shooting* [1], vengono stimati i vari valori della funzione d'onda al variare della posizione e, tale funzione, viene disegnata rinormalizzata attraverso il calcolo della sua norma tramite la *regola di simpson* [2]:

$$\int_{x_1}^{x_3} f(x)dx = h \left[\frac{1}{3}f(x_1) + \frac{4}{3}f(x_2) + \frac{1}{3}f(x_3) \right] \quad (4.1)$$

I valori forniti al programma per il calcolo delle derivate sono:

$$\begin{cases} \frac{dy(x_1)}{dx} = \frac{dy(x_0)}{dx} \\ \frac{d^2y(x_1)}{dx^2} = (x_0^2 - 2E_n)y(x_0) \end{cases} \quad (4.2)$$

con x_0 punto di partenza e $x_1 = x_0 + \Delta x$.

Il codice di questo programma è mostrato nell'appendice A.

L'output del programma è:

Gli autovalori dell'oscillatore armonico, in unità di $\hbar\omega$, sono dati da:

```
0  5.00000e-01
1  1.50000e+00
2  2.50000e+00
3  3.50000e+00
4  4.50000e+00
```

che rispecchiano perfettamente i risultati teorici trovati nell'equazione 2.16.

Le funzioni d'onda disegnate sono invece quelle riportate nella figura 4.1 in accordo con i grafici 2.1.

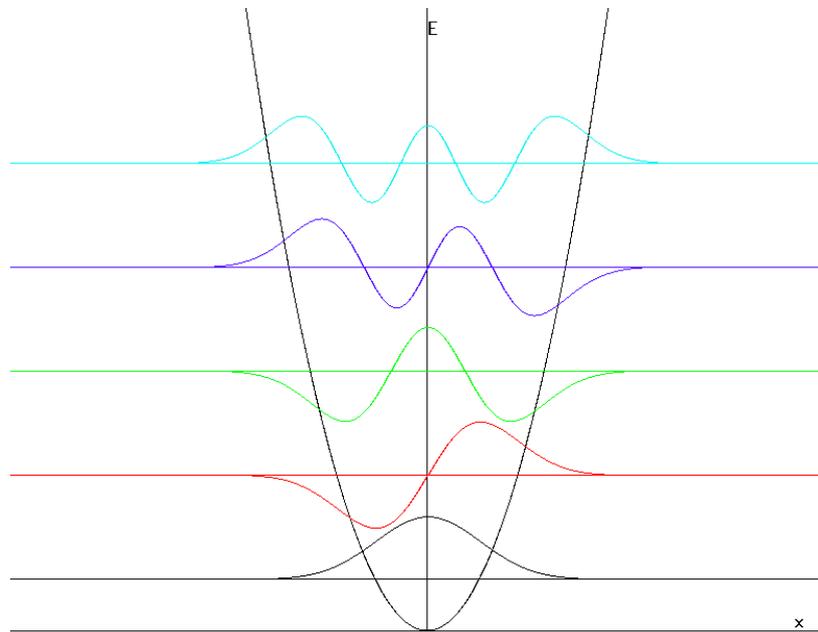


Figura 4.1: Funzioni d'onda dell'oscillatore armonico disegnate dal programma Oscillatore.cc

4.2 Implementazione del Programma per l'Ammoniacca

In questa parte vengono analizzati i programmi, con i relativi dati di output, scritti per la risoluzione del problema del potenziale dell' NH_3 approssimato nel modo mostrato nelle equazioni 3.1 e 3.2. La struttura dei due programmi è la stessa trattata nella sezione 4.1; quello che è stato modificato per ognuno di questi programmi sono i potenziali disegnati, i vari parametri iniziali e l'equazione risolta.

4.2.1 Il Modello polinomiale

Il corpo principale del programma è costituito dal documento ammoniacca.cc (vedi appendice B.1). Il potenziale che è stato disegnato è del tipo:

$$s[k_1(xt)^4 - k_2(xt)^2 - tf] \quad (4.3)$$

con $k_1 = 72422$, $k_2 = 23483$ e $s = 0.005$, $f = 8.0 \cdot 10^{-3}$ e $t = 8 \cdot 10^6$ costanti usate per scalare il grafico a misura della finestra creata. Le funzioni fornite per il calcolo delle derivate sono:

$$\left\{ \begin{array}{l} \frac{dy(x_1)}{dx} = \frac{dy(x_0)}{dx} \\ \frac{d^2y(x_1)}{dx^2} = 4k\rho^2(k_1x^4 - k_2x^2 - C - E)y(x_0) \end{array} \right. \quad (4.4)$$

con $k\rho^2 = 2.885 \cdot 10^{-2}$, $C = 43068$, k_1 e k_2 dati sopra. L'output ottenuto attraverso l'esecuzione del programma è:

Gli autovalori dell'Hamiltoniana con potenziale di Manning, espressi in cm^{-1} , sono dati da:

```
0 -4.43652e+04
1 -4.43622e+04
2 -4.33664e+04
3 -4.32483e+04
4 -4.26316e+04
```

I valori ottenuti possono essere confrontati con i risultati sperimentali e teorici forniti da Manning [7] e riassunti nella tabella di seguito.

	calc. M.	calc. R.K.	oss.
ΔE_{10}	0.83	3	0.67
ΔE_{20}	935	998.8	932
ΔE_{30}	961	1116.9	964
ΔE_{40}	1610	1733.6	1600

le differenze di energia tra i livelli ij (ΔE_{ij}) sono date in cm^{-1} , i valori nella colonna calc. M. sono i dati calcolati da Manning, i valori sotto calc. R.K. sono quelli calcolati attraverso l'algoritmo di Runge-Kutta mentre i valori sotto oss. sono i dati ottenuti sperimentalmente; come si può vedere i risultati sono dello stesso ordine di grandezza dei dati sperimentali. Le funzioni d'onda disegnate dal programma sono illustrate in figura 4.2.

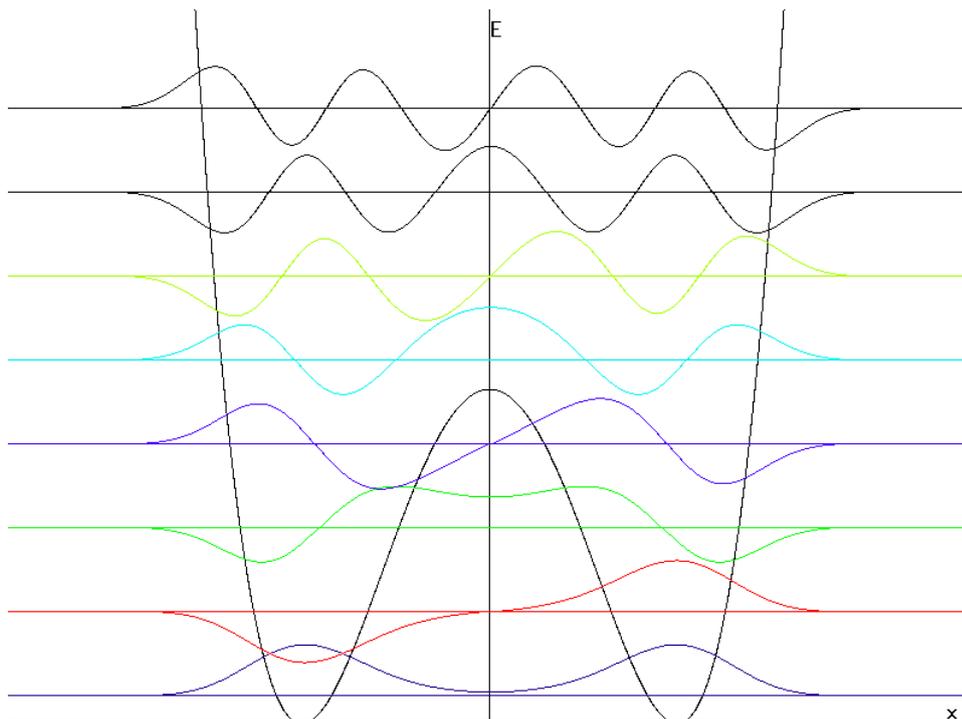


Figura 4.2: Funzioni d'onda dell'ammoniaca, con potenziale polinomiale, disegnate dal programma ammoniaca.cc

4.2. IMPLEMENTAZIONE DEL PROGRAMMA PER L'AMMONIACA23

4.2.2 Il Modello Iperbolico

Il file principale del programma per la risoluzione del potenziale dell'ammoniaca, approssimato tramite funzioni iperboliche, è ammoniacamanning.cc. Il potenziale che e' stato disegnato è del tipo:

$$s[k_1 \operatorname{sech}^4(xf) - k_2 \operatorname{sech}^2(xf)] \quad (4.5)$$

con $f = 8 \cdot 10^{-3}$ ed $s = 0.013$ fattori usati per adattare il grafico alla finestra creata e con $k_1 = 66\,551$ e $k_2 = 109\,619$ dati da Manning nel potenziale 3.2. I valori forniti al programma per il calcolo delle derivate sono:

$$\left\{ \begin{array}{l} \frac{dy(x_1)}{dx} = \frac{dy(x_0)}{dx} \\ \frac{d^2y(x_1)}{dx^2} = 4k\rho^2[k_1 \operatorname{sech}^4(x_0) - k_2 \operatorname{sech}^2(x_0) - E]y(x_0) \end{array} \right. \quad (4.6)$$

con $k\rho^2 = 2.885 \cdot 10^{-2}$ e k_1 e k_2 dati sopra. Il codice del programma è consultabile nell'appendice B.2 mentre l'output ricavato dal programma è dato da:

Gli autovalori dell'Hamiltoniana con potenziale di Manning, espressi in cm^{-1} , sono dati da:

```
0 -4.46340e+04
1 -4.46333e+04
2 -4.37070e+04
3 -4.36770e+04
4 -4.30281e+04
```

se compariamo i dati ottenuti con quelli forniti da Manning[7], si ottiene:

	calc. M.	calc. R.K.	oss.
ΔE_{10}	0.83	0.7	0.67
ΔE_{20}	935	927	932
ΔE_{30}	961	957	964
ΔE_{40}	1610	1606	1600

le differenze di energia tra i livelli ij (ΔE_{ij}) sono date in cm^{-1} , i valori nella colonna calc. M. sono i dati calcolati da Manning, i valori sotto calc. R.K. sono quelli calcolati attraverso l'algoritmo di Runge-Kutta mentre i valori sotto oss. sono i dati ottenuti sperimentalmente; come si può vedere i risultati ottenuti sono molto prossimi ai dati sperimentali. Di seguito viene mostrato il grafico delle funzioni d'onda ottenute. Si noti che i livelli degli autovalori sono stati spostati per permettere una migliore visibilita' delle funzioni d'onda.

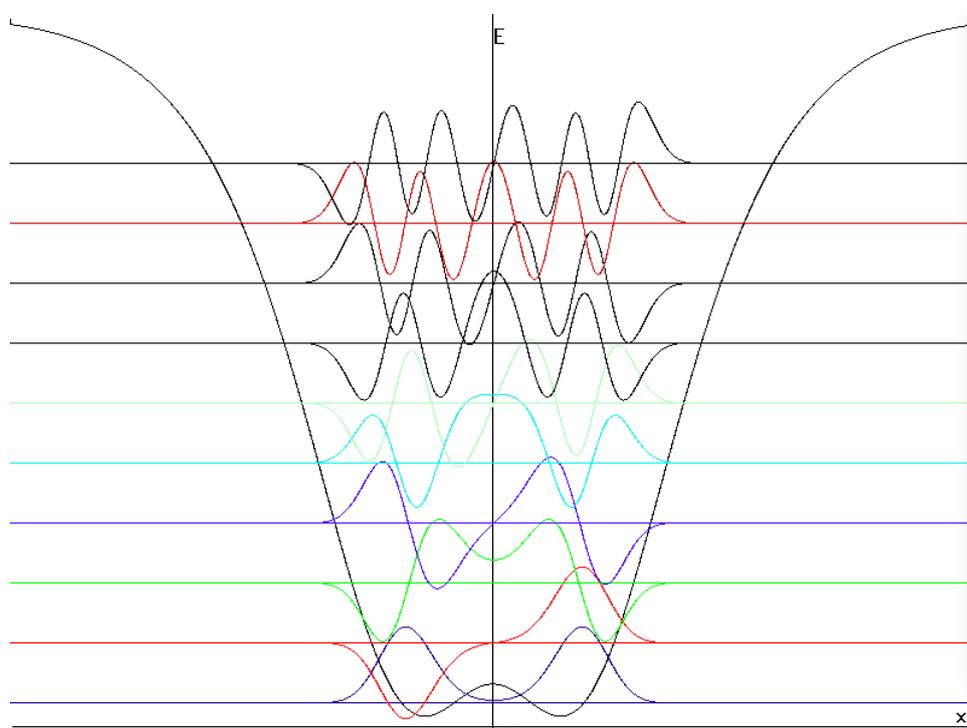


Figura 4.3: Funzioni d'onda dell'ammoniaca, con potenziale di Manning, disegnate dal programma ammoniacamanning.cc

Appendice A

Il Codice dell'Oscillatore Armonico

In questa parte viene riportato il programma principale utilizzato per ricavare la soluzione al problema dell'oscillatore armonico. Il codice del programma è:

```
#include <curses.h>
#include <math.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <time.h>
#include <header.h>

extern XVideoData xvd;
void func(double *x, double *y, double *dydx);

main(){
    int DeltaSave, i, j, nEigen, nPoints, iy1, iy2, yEigv;
    unsigned long color [10];
    double DeltaX, DeltaX2, EigvStart, EigvStep, tolerance, xx, xMax, yy;
    double par [4];
    double *eigv, *x;
    double **y;
    // ..... modo grafico
    StartXWindow(800,600,0,0,"Oscillatore_Armonico",1);
    XSelectInput(xvd.display, xvd.win[0], ButtonPressMask | ButtonReleaseMask |
        PointerMotionMask | KeyPressMask | KeyReleaseMask | ExposureMask |
        StructureNotifyMask);
    // ..... definizione colori
    color[0]=xvd.black;
    color[1]=xvd.red[0];
    color[2]=xvd.green[0];
    color[3]=xvd.blue[0];
    color[4]=xvd.cyan[1];
    // ..... definizione parametri
    DeltaSave=4; //numero di punti per passo
    nEigen=5;
    eigv=new double [nEigen];
    nPoints=500;
    x=new double [nPoints+8];
    y=new double * [2];
```

```

for (i=0;i<2;i++) y[i]=new double [nPoints+8];
EigvStep=0.005;
tolerance=1.0e-12;
par[2]=1.0;
par[3]=2.0;
xMax=10.0;
// ..... disegno potenziale
DeltaX=double(xMax)/double(nPoints);
DeltaX2=DeltaX*DeltaX;
XSetForeground(xvd.display,xvd.gc,xvd.black);
iy1=0;
for (i=0;i<800;i++)
{
    yy=DeltaX*double((i-399)*(i-399));
    iy2=int(599.0-yy);
    XDrawLine(xvd.display,xvd.backpix[0],xvd.gc,i-1,iy1,i,iy2);
    iy1=iy2;
}
XDrawLine(xvd.display,xvd.backpix[0],xvd.gc,399,0,399,599);
XRedraw();
// ..... disegno funzioni d'onda
EigvStart=0.0;
printf("Gli autovalori dell'oscillatore armonico sono dati da:\n");
for (i=0;i<nEigen;i++)
{
    eigv[i]=SymmWell(par,x,y,nPoints,DeltaSave,xMax,i,EigvStart,
                    EigvStep,tolerance,func);
    printf("%4d_%15.5le\n",i,eigv[i]);
    EigvStart=eigv[i]+EigvStep;
    yEigv=599-int(2.0*eigv[i]/DeltaX);
    PlotPsi(y[0],nPoints,(i+1)%2,400.0,yEigv,color[i]);
    iy1=599-int(2.0*eigv[i]/DeltaX);
    XDrawLine(xvd.display,xvd.backpix[0],xvd.gc,0,iy1,799,iy1);
    XRedraw();
}
// ..... chiusura programma
XWait(0);
CloseXWindow();
for (i=1;i>=0;i--) delete [] y[i];
delete [] y;
delete [] x;
delete [] eigv;
}
// ..... funzione potenziale
void func(double *x,double *y,double *dydx){
double A,B,E;
E=x[1];
A=x[2];
B=x[3];
dydx[0]=y[1];
dydx[1]=(A*x[0]*x[0]-B*E)*y[0];
}

```

Il codice dell'header e delle funzioni utilizzate nel programma è riportato nell'appendice C.

Appendice B

Il Codice dell'Ammoniaca

In questa parte viene riportato il programma principale utilizzato per ricavare la soluzione al problema dell'ammoniaca.

B.1 Il Codice del Modello Polinomiale

```
#include <curses.h>
#include <math.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <time.h>
#include <header.h>

extern XVideoData xvd;
void func(double *x, double *y, double *dydx);

main()
{
    int DeltaSave, i, j, nEigen, nPoints, iy1, iy2, yEigv, trasl;
    unsigned long color [10];
    double DeltaX, DeltaX2, EigvStart, EigvStep, tolerance, xx, xMax, yy,
           ymax, fattore, scala;

    double par [6];
    double *eigv, *x;
    double **y;
    // ..... modo grafico
    StartXWindow(800,600,0,0,"Ammoniaca_Manning",1);
    XSelectInput(xvd.display,xvd.win[0],ButtonPressMask|ButtonReleaseMask|
                PointerMotionMask|KeyPressMask|KeyReleaseMask|ExposureMask|
                StructureNotifyMask);
    // ..... definizione colori
    color[0]=xvd.blue[3];
    color[1]=xvd.red[0];
    color[2]=xvd.green[0];
    color[3]=xvd.blue[0];
    color[4]=xvd.cyan[1];
    // ..... definizione parametri
    DeltaSave=4;
    nEigen=8;
```

```

eigv=new double [nEigen];
nPoints=1000;
x=new double [nPoints+8];
y=new double * [2];
for (i=0;i<2;i++) y[i]=new double [nPoints+8];
EigvStep=0.5;
fattore=8.0e-3;
scala=0.005;
tolerance=1.0e-10;
trasl=8e6;
par[2]=72422;
par[3]=23483;
par[4]=43068;
par[5]=4*2.885e-2;
xMax=2.5;
// ..... disegno potenziale
DeltaX=double(xMax)/double(nPoints);
DeltaX2=DeltaX*DeltaX;
XSetForeground(xvd.display,xvd.gc,xvd.black);
iy1=1;
ymax = 600;
for (i=0;i<800;i++)
{
    yy=scala*double(par[3]*pow((i-399)*fattore,4)-par[2]*
        pow((i-399)*fattore,2)-trasl*fattore);
    iy2=int(-yy);
    XDrawLine(xvd.display,xvd.backpix[0],xvd.gc,i-1,iy1,i,iy2);
    iy1=iy2;
}
XPrintf(0,400,10,1,0,"E");
XDrawLine(xvd.display,xvd.backpix[0],xvd.gc,399,0,399,599);
XRedraw();
// ..... disegno funzioni d'onda
EigvStart=-45000;
printf("Gli autovalori dell'ammoniaca sono da:\n");
for (i=0;i<nEigen;i++)
{
    eigv[i]=SymmWell(par,x,y,nPoints,DeltaSave,xMax,i,
        EigvStart,EigvStep,tolerance,func);
    printf("%4d_%15.5le\n",i,eigv[i]);
    EigvStart=eigv[i]+EigvStep;
    yEigv=575-70*i;
    PlotPsi(y[0],nPoints,(i+1)%2,400.0,yEigv,color[i]);
    iy1=575-70*i;
    XDrawLine(xvd.display,xvd.backpix[0],xvd.gc,0,iy1,799,iy1);
    XRedraw();
}
// ..... chiusura programma
XWait(0);
CloseXWindow();
for (i=1;i>=0;i--) delete [] y[i];
delete [] y;
delete [] x;
delete [] eigv;
}
// ..... funzione potenziale
void func(double *x,double *y,double *dydx)
{
    double A,B,C,D,E,fattore;
    fattore=1.0e-2;
    E=x[1];
    A=x[2];

```

```

    B=x[3];
    C=x[4];
    D=x[5];
    dydx[0]=y[1];
    dydx[1]=(A*pow(x[0],4) - B*pow(x[0],2) - C - E)*y[0]*D;
}

```

B.2 Il Codice del Modello Iperbolico

```

#include <curses.h>
#include <math.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <time.h>
#include <header.h>

extern XVideoData xvd;
void func(double *x, double *y, double *dydx);
double sech(double x);

main()
{
    int DeltaSave, i, j, nEigen, nPoints, iy1, iy2, yEigv;
    unsigned long color[10];
    double DeltaX, DeltaX2, EigvStart, EigvStep, tolerance, xx, xMax,
           yy, ymax, fattore, scala;

    double par[5];
    double *eigv, *x;
    double **y;
    // ..... modo grafico
    StartXWindow(800,600,0,0,"Ammoniaca_Manning",1);
    XSelectInput(xvd.display,xvd.win[0],ButtonPressMask|ButtonReleaseMask|
                PointerMotionMask|KeyPressMask|KeyReleaseMask|ExposureMask|
                StructureNotifyMask);
    // ..... definizione colori
    color[0]=xvd.blue[3];
    color[1]=xvd.red[0];
    color[2]=xvd.green[0];
    color[3]=xvd.blue[0];
    color[4]=xvd.cyan[1];
    // ..... definizione parametri
    DeltaSave=4;
    nEigen=10;
    eigv=new double [nEigen];
    nPoints=1000;
    x=new double [nPoints+8];
    y=new double * [2];
    for (i=0;i<2;i++) y[i]=new double [nPoints+8];
    EigvStep=0.05;
    fattore=8.0e-3;
    scala=0.013;
    tolerance=1.0e-10;
    par[2]=66551.12652;
    par[3]=109618.7175;
    par[4]=4*2.885e-2;
    xMax=6.0;
    // ..... disegno potenziale
    DeltaX=double(xMax)/double(nPoints);

```

```

DeltaX2=DeltaX*DeltaX;
XSetForeground(xvd.display,xvd.gc,xvd.black);
iy1=1;
ymax = 600;
for (i=0;i<800;i++)
{
    yy=scala*double(par[2]*pow(sech((i-399)*fattore),4) -
                    par[3]*pow(sech((i-399)*fattore),2));
    iy2=int(-yy);
    XDrawLine(xvd.display,xvd.backpix[0],xvd.gc,i-1,iy1,i,iy2);
    iy1=iy2;
}
XPrintf(0,400,10,1,0,"E");
XDrawLine(xvd.display,xvd.backpix[0],xvd.gc,399,0,399,599);
XRedraw();
// ..... disegno funzioni d'onda
EigvStart=-45140;
printf("Gli autovalori dell'ammoniaca_sono_dati_da:\n");
for (i=0;i<nEigen;i++)
{
    eigv[i]=SymmWell(par,x,y,nPoints,DeltaSave,xMax,i,EigvStart,
                    EigvStep,tolerance,func);
    printf("%4d_%.155le\n",i,eigv[i]);
    EigvStart=eigv[i]+EigvStep;
    yEigv=575-50*i;
    PlotPsi(y[0],nPoints,(i+1)%2,400.0,yEigv,color[i]);
    iy1=575-50*i;
    XDrawLine(xvd.display,xvd.backpix[0],xvd.gc,0,iy1,799,iy1);
    XRedraw();
}
// ..... chiusura programma
XWait(0);
CloseXWindow();
for (i=1;i>=0;i--) delete [] y[i];
delete [] y;
delete [] x;
delete [] eigv;
}
// ..... funzione potenziale
void func(double *x,double *y,double *dydx)
{
    double A,B,C,E,fattore;
    fattore=1.0e-2;
    E=x[1];
    A=x[2];
    B=x[3];
    C=x[4];
    dydx[0]=y[1];
    dydx[1]=(A*pow(sech(x[0]),4) - B*pow(sech(x[0]),2) - E)*y[0]*C;
}
// ..... funzione secante iperbolica
double sech(double x)
{
    double e=2.718281828, secante;
    secante= 2/(pow(e,x)+pow(e,-x));
    return secante;
}

```

Il codice dell'header e delle funzioni utilizzate nel programma è riportato nell'appendice C.

Bibliografia

- [1] K. Konishi e G. Paffuti, *Meccanica quantistica: nuova introduzione*, Edizioni Plus, Pisa, 2007, Cap 3.
- [2] B. P. Flannery, W. H. Press, S. A. Teukolsky e W. T. Vetterling, *Numerical Recipes in C*, Cambridge University Press, Cambridge, 1992, Cap 4,16.
- [3] A. L. Shawlow e C. H. Townes, *Microwave spectroscopy*, McGraw-Hill Book Company, New York, Toronto, London, 1955, Cap 12.
- [4] A. Nye, *Xlib Programming Manual*, O'Reilly & Associates Inc., United States of America, 1992.
- [5] C. Cohen Tannoudji, B. Diu, F. Laloë, *Quantum Mechanics*, Wiley-Vhc Verlag GmbH & Co. KGaA, Paris, 2005
- [6] G. Moruzzi, *Corso di Algoritmi di Spettroscopia Molecolare*, Università degli Studi di Pisa, Pisa, 2010.
- [7] Manning M.F.J., *Chemical Physics*, 3, Pag. 136.