计算物理第三次作业

梁旭民*

Cuiying Hornors College, Lanzhou University

liangxm15@lzu.edu.cn

Abstract

本次计算物理作业主要通过打靶法求解边界为无穷高的线性势阱的一维*schrodinger*方程，并综合了前三章所有内容:*Simpson*积分、搜索法及弦切法求根、*Numernov*算法、打靶法。

I. 问题描述

计算半导体硅中能谷电子在三角形势阱中的所有能量本征值和波函数，其中低能谷电子有效能量$0.916m_e$，势场宽度35nm，势阱深度1eV。(本征值无限接近连续时候根据自己程序精度舍弃)

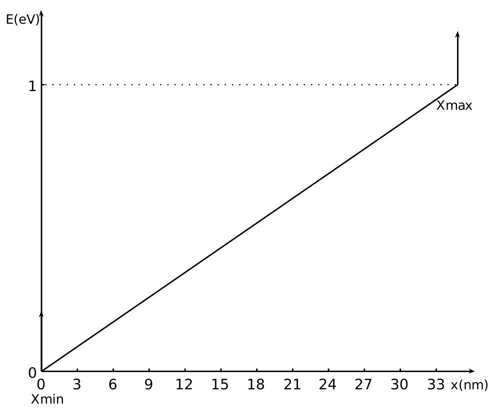


Figure 1: 一维位势*V(x)*图像(国际单位制下)

II. 问题分析

该问题本质上是求解定态schrodinger方程

$$\begin{cases} \left[-\dfrac{d^2}{dx^2}+\dfrac{2m}{\hbar^2}(V(x)-E)\right]\psi(x)=0 \\ \psi(x_{min})=\psi(x_{max})=0 \end{cases}$$

其中，势阱V(x)为

$$\begin{cases} \dfrac{1}{35}x & x<35nm \\ \infty & x\geqslant 35nm \end{cases}$$

*指导老师：齐新老师

该本质问题相当于定解问题

$$\begin{cases} \left(\dfrac{d^2}{dx^2}+k^2\right)\psi(x)=0 \\ \psi(x_{min})=\psi(x_{max})=0 \end{cases}$$

本征值问题多了一个待定参数k，先猜测一个本征值k，设一个非零参数δ，将该问题转化为初值问题

$$\begin{cases} \left(\dfrac{d^2}{dx^2}+k^2\right)\psi(x)=0 \\ \psi(x_{min})=\psi(x_{max})=0 \\ \psi'(x_{min})=\psi'(x_{max})=\delta \end{cases}$$

III. Numerov算法简介

Numerov方法属于四阶线性多步法，用于求解不出现一阶微分项的二阶常微分方程。Numerov方法属于隐式方法，但如果微分方程线性，则可转化为显式方法。该方法由俄国天文学家Boris Vasil'evich Numerov提出。

A. 基本算法

可由Numerov方法求解的微分方程形式为

$$\left(\frac{d^2}{dx^2}+f(x)\right)y(x)=0$$

求出函数$y(x)$在区间$[a,b]$上等距格点上的值，从连续的两个格点上的函数值x_{n-1}和x_n开始，其他的函数值可由

$$y_{n+1}=\frac{\left(2-\frac{5h^2}{6}f_n\right)y_n-\left(1+\frac{h^2}{12}f_{n-1}\right)y_{n-1}}{1+\frac{h^2}{12}f_{n+1}}$$

算得。

其中，$f_n = f(x_n)$和$y_n = y(x_n)$为在格点x_n上的函数值，$h = x_n - x_{n-1}$为格点间距。

对于非线性方程，

$$\frac{d^2}{dt^2}y = f(t, y)$$

则非线性方程的Numerov方法为

$$y_{n+1} = 2y_n - y_{n-1} + \tfrac{1}{12}h^2(f_{n+1} + 10f_n + f_{n-1})$$

该式为隐式的线性多步方法。当f是y的线性函数时，该式变为显式方法，精度为4阶（Hairer，Nørsett & Wanner 1993，§III.10）

B. 算法推导

从$y(x_n)$的Taylor展开开始，我们可求x_n的相接邻点上的函数值

$$y_{n+1} = y(x_n + h) = y(x_n) + hy'(x_n) + \frac{h^2}{2!}y''(x_n) + \frac{h^3}{3!}y'''(x_n) + \frac{h^4}{4!}y''''(x_n) + \frac{h^5}{5!}y'''''(x_n) + \mathcal{O}(h^6)$$

$$y_{n-1} = y(x_n - h) = y(x_n) - hy'(x_n) + \frac{h^2}{2!}y''(x_n) - \frac{h^3}{3!}y'''(x_n) + \frac{h^4}{4!}y''''(x_n) - \frac{h^5}{5!}y'''''(x_n) + \mathcal{O}(h^6)$$

上两式之和为

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

用所求微分方程的定义式$y_n'' = -f_n y_n$替换掉y_n''

$$h^2 f_n y_n = 2y_n - y_{n-1} - y_{n+1} + \frac{h^4}{12}y_n'''' + \mathcal{O}(h^6)$$

对所求微分方程的定义式$y_n'' = -f_n y_n$取二次微分

$$y''''(x) = -\frac{d^2}{dx^2}\left[f(x)y(x)\right]$$

将其代入到四阶微分项中，并把二阶导$\frac{d^2}{dx^2}\left[f(x)y(x)\right]$替换为$f_n y_n$的二阶差分公式$\frac{f_{n-1}y_{n-1} - 2f_n y_n + f_{n+1}y_{n+1}}{h^2}$

$$h^2 f_n y_n = 2y_n - y_{n-1} - y_{n+1} - \frac{h^4}{12}\frac{f_{n-1}y_{n-1} - 2f_n y_n + f_{n+1}y_{n+1}}{h^2} + \mathcal{O}(h^6)$$

求解y_{n+1}可得

$$y_{n+1} = \frac{\left(2 - \frac{5h^2}{6}f_n\right)y_n - \left(1 + \frac{h^2}{12}f_{n-1}\right)y_{n-1}}{1 + \frac{h^2}{12}f_{n+1}} + \mathcal{O}(h^6)$$

忽略掉$\mathcal{O}(h^6)$就可以得到Numerov方法，最终收敛阶数为4(假定稳定)。

C. 算法应用

在物理中用于数值求解任意势场中径向schrodinger方程:

$$\left[-\frac{\hbar^2}{2\mu}\left(\frac{1}{r}\frac{\partial^2}{\partial r^2}r - \frac{l(l+1)}{r^2}\right) + V(r)\right]R(r) = ER(r)$$

此式可重写为

$$\left[\frac{\partial^2}{\partial r^2} - \frac{l(l+1)}{r^2} + \frac{2\mu}{\hbar^2}\left(E - V(r)\right)\right]u(r) = 0$$

其中$u(r) = rR(r)$与Numerov方法求解的方程形式做比较，则有

$$f(x) = \frac{2\mu}{\hbar^2}\left(E - V(x)\right) - \frac{l(l+1)}{x^2}$$

这样，我们可以数值求解schrodinger方程。

IV. 算法设计

选择打靶法求解该本征值问题。

从x_{min}出发积分，产生一个数值解$\psi_<$，它在经典禁戒的区指数增长，经过x_{min}进入经典允许的区域，并在此区域内振荡，但如果再经过x_{max}积分下去，会因为某个指数增长部分而使得数值不稳定。因此便需要将x分为两部分，并分别使用numerov算法积分求出波函数，对于中间的转折点则应该有条件

$$\begin{cases} \psi_< = \psi_> \\ \psi'_< = \psi'_> \end{cases}$$

因此，我们先通过给左右两部分的某一支数乘一个系数C使得

$$\psi_< = \psi_>$$

并不断通过猜测的能量本征值，调节波函数光滑连续，将光滑作为判定条件即还需满足

$$\psi'_< = \psi'_>$$

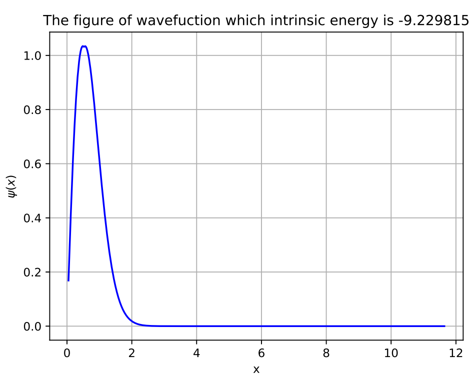
V. 计算结果

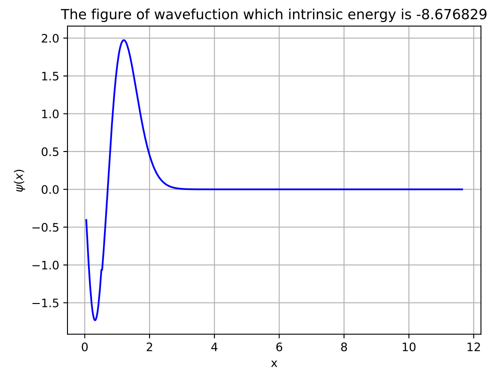
对于该问题，存在并求解出了36个能量本征值，从小到大依次为

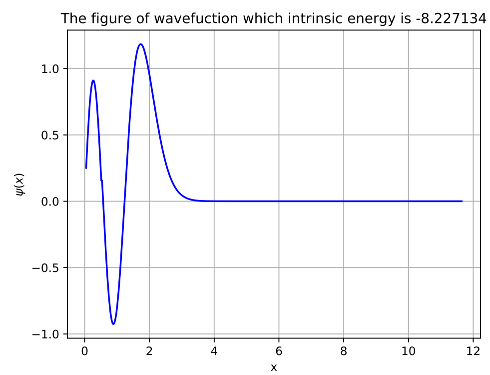
-9.229815	-8.676829	-8.227134
-7.819172	-7.439463	-7.083014
-6.746237	-6.426222	-6.120651
-5.827706	-5.545966	-5.011822
-4.757705	-4.511182	-4.271408
-4.037440	-3.808270	-3.582975
-3.360825	-3.141352	-2.709584
-2.497157	-2.287020	-2.079168
-1.873584	-1.670241	-1.469100
-1.270116	-1.073227	-0.878328
-0.685096	-0.492458	-0.297843
-0.097407	$+0.112194$	

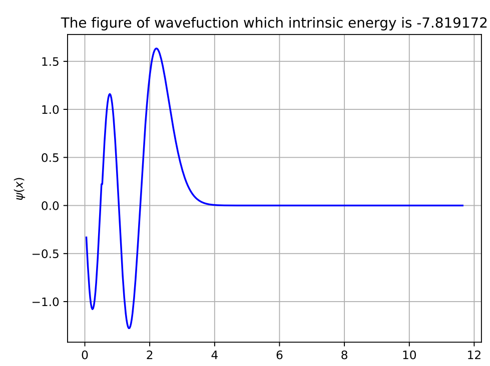
VI. 绘制波函数图像

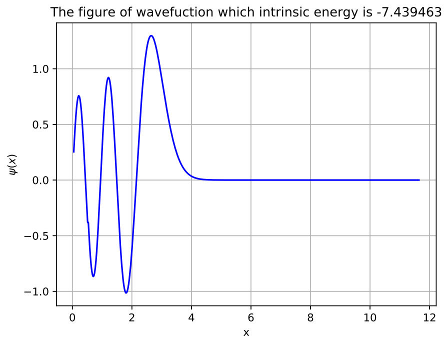
运用Python的openpyxl和pyplot库，通过循环读取工作簿不同工作表中的数据，分别绘制出对应与不同的能量本征值波函数图像

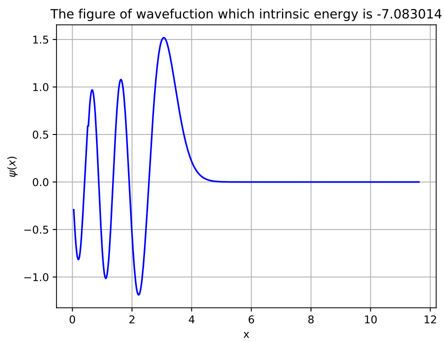


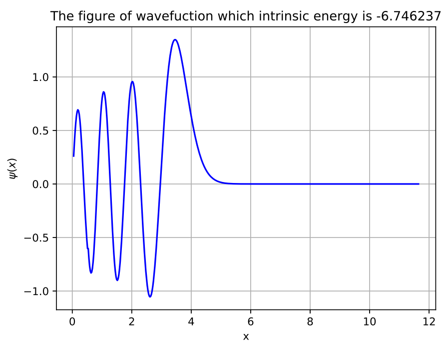


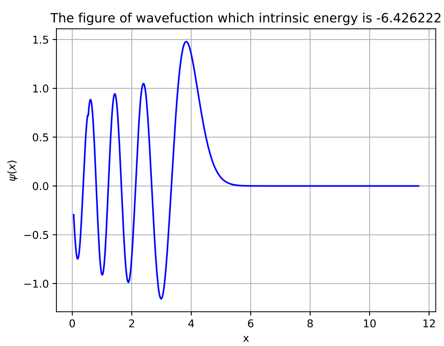


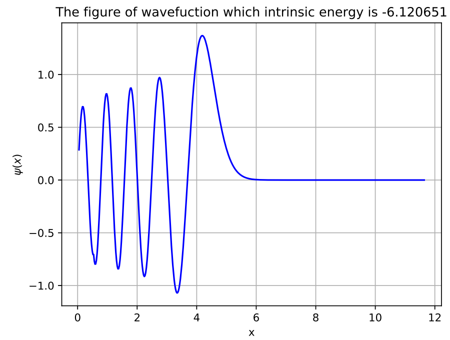


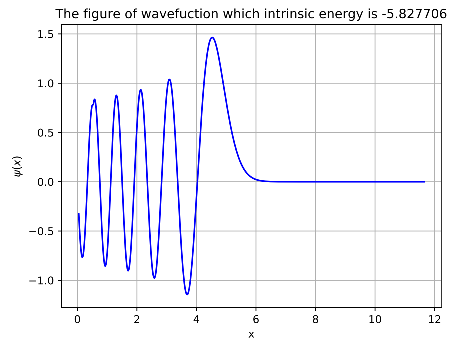

The figure of wavefuction which intrinsic energy is -7.439463
ψ(x)
x

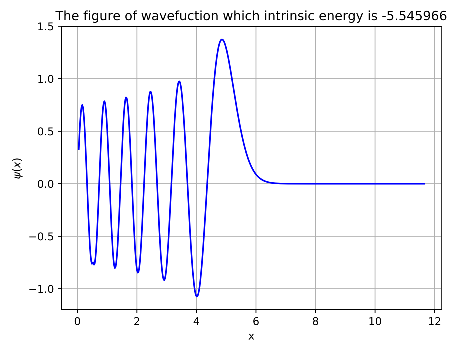

The figure of wavefuction which intrinsic energy is -7.083014
ψ(x)
x

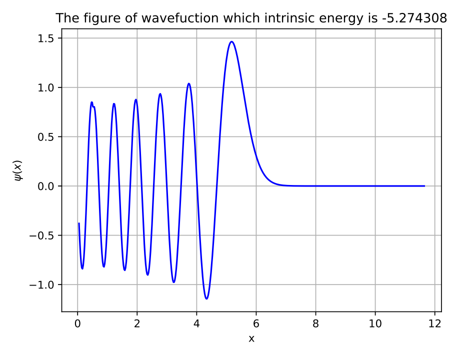

The figure of wavefuction which intrinsic energy is -6.746237
ψ(x)
x

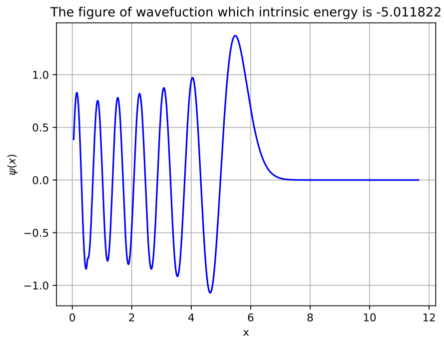

The figure of wavefuction which intrinsic energy is -6.426222
ψ(x)
x

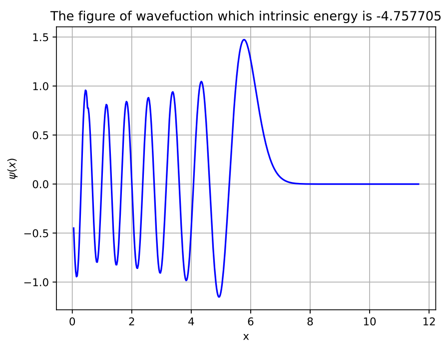

The figure of wavefuction which intrinsic energy is -6.120651
ψ(x)
x

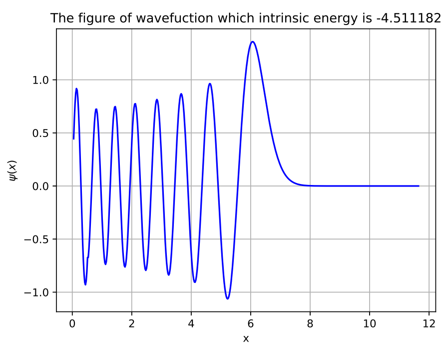

The figure of wavefuction which intrinsic energy is -5.827706
ψ(x)
x

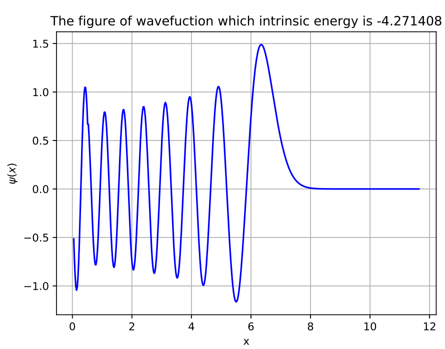

The figure of wavefuction which intrinsic energy is -5.545966
ψ(x)
x

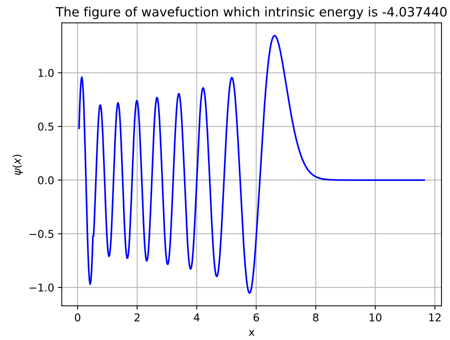

The figure of wavefuction which intrinsic energy is -5.274308
ψ(x)
x

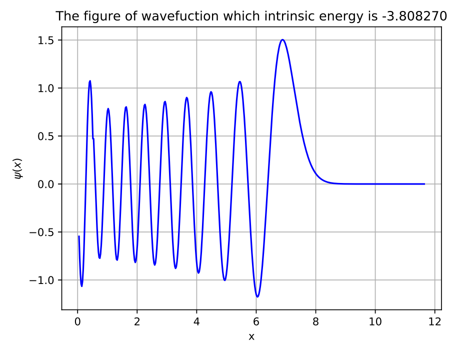

The figure of wavefuction which intrinsic energy is -5.011822
ψ(x)
x

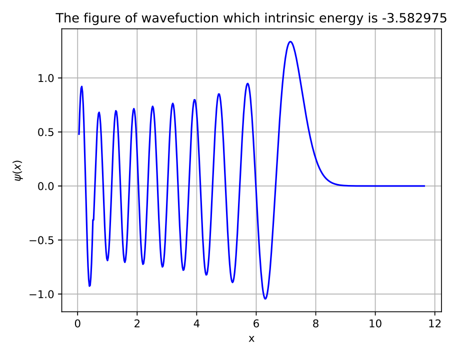

The figure of wavefuction which intrinsic energy is -4.757705
ψ(x)
x

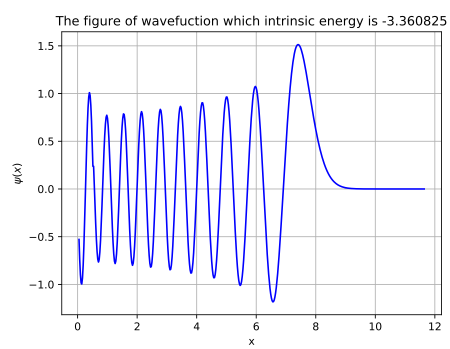

The figure of wavefuction which intrinsic energy is -4.511182
ψ(x)
x

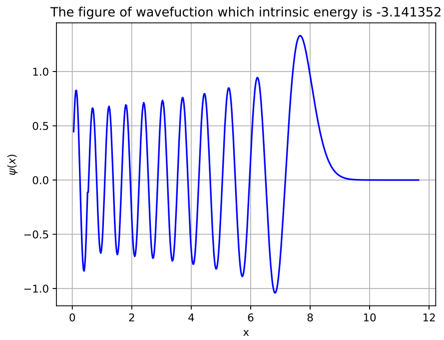

The figure of wavefuction which intrinsic energy is -4.271408
ψ(x)
x

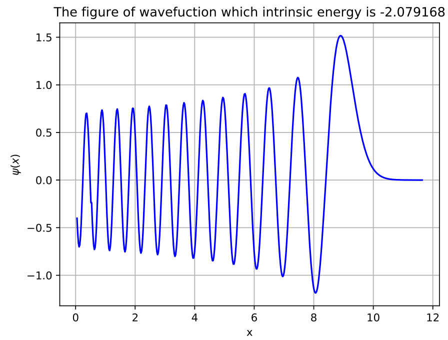

The figure of wavefuction which intrinsic energy is -4.037440
ψ(x)
x

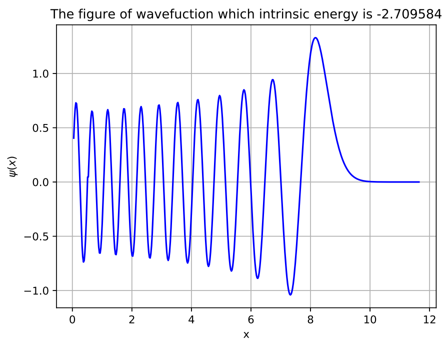

The figure of wavefuction which intrinsic energy is -3.808270
ψ(x)
x

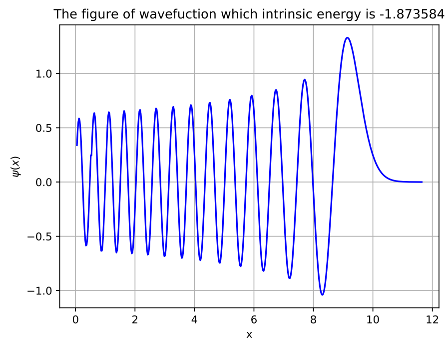

The figure of wavefuction which intrinsic energy is -3.582975
ψ(x)
x

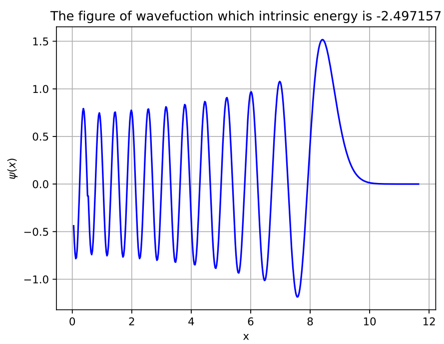

The figure of wavefuction which intrinsic energy is -3.360825
ψ(x)
x

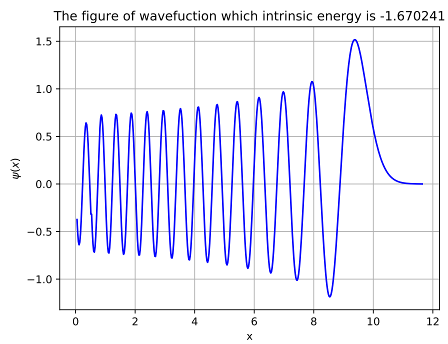

The figure of wavefuction which intrinsic energy is -3.141352
ψ(x)
x

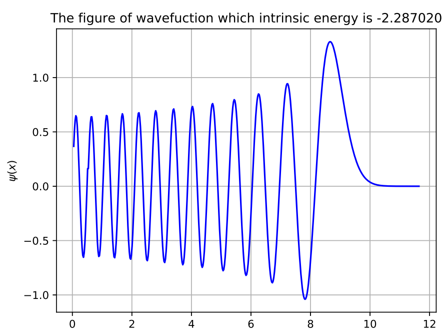

The figure of wavefuction which intrinsic energy is -2.079168
ψ(x)
x

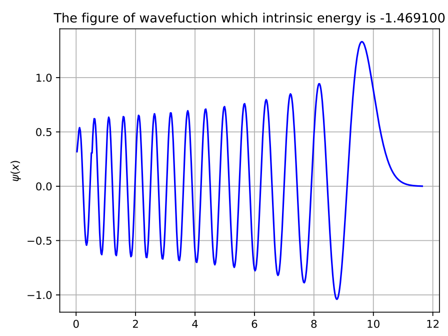

The figure of wavefuction which intrinsic energy is -2.709584
ψ(x)
x

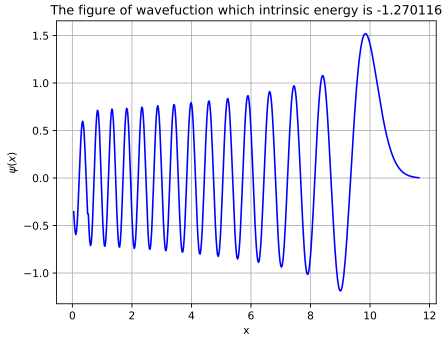

The figure of wavefuction which intrinsic energy is -1.873584
ψ(x)
x

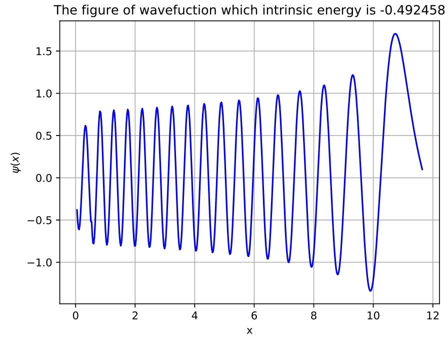

The figure of wavefuction which intrinsic energy is -2.497157
ψ(x)
x

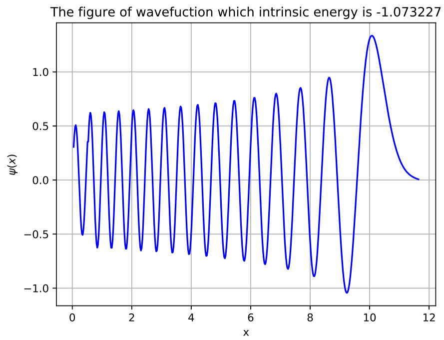

The figure of wavefuction which intrinsic energy is -1.670241
ψ(x)
x

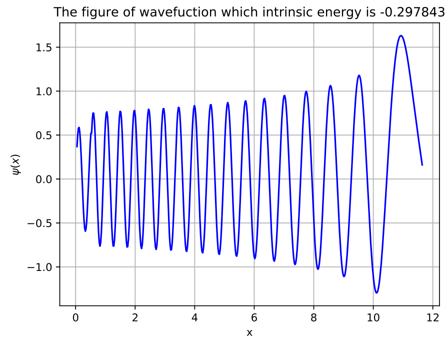

The figure of wavefuction which intrinsic energy is -2.287020
ψ(x)
x

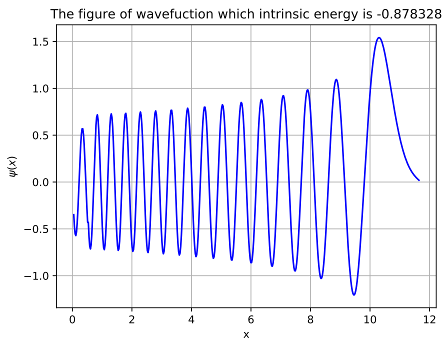

The figure of wavefuction which intrinsic energy is -1.469100
ψ(x)
x

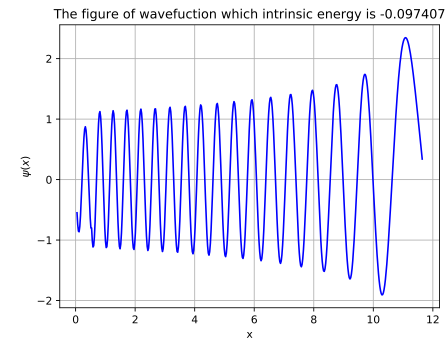

The figure of wavefuction which intrinsic energy is -1.270116
ψ(x)
x

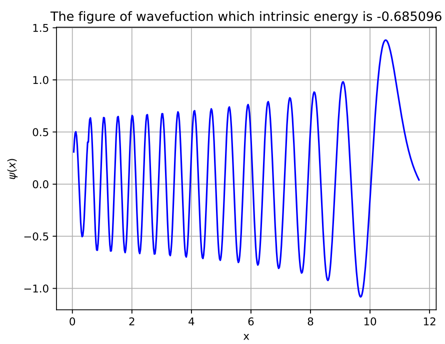

The figure of wavefuction which intrinsic energy is -0.492458
ψ(x)
x

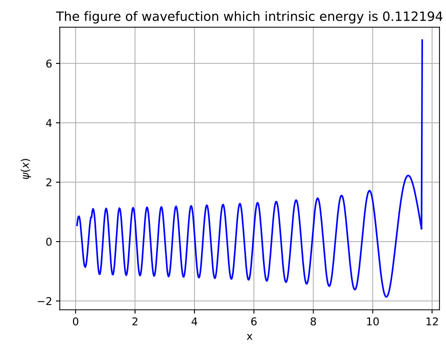

The figure of wavefuction which intrinsic energy is -1.073227
ψ(x)
x


The figure of wavefuction which intrinsic energy is -0.297843
ψ(x)
x


The figure of wavefuction which intrinsic energy is -0.878328
ψ(x)
x


The figure of wavefuction which intrinsic energy is -0.097407
ψ(x)
x


The figure of wavefuction which intrinsic energy is -0.685096
ψ(x)
x


The figure of wavefuction which intrinsic energy is 0.112194
ψ(x)
x

Appendices

A. 计算求解部分

这是运用打靶法求解schrodinger方程的C语言程序。

Input C source:

```
#include<stdio.h>
#include<math.h>
double V1(double x);
double F(int n,double x0,double y0,double xn,double yn,double xm,double K[n+1],double k);
double Fa(int n,double x0,double y0,double xn,double yn,double xm,double K[n+1],double k);
double VTarget_2(int n,double x0,double y0,double xn,double yn,double xm,double K[n+1],
double k);
int main()
{
    int n=500,i;
    double m0=9.108e-31,ml=0.916*m0,V0=1.602e-20,a=3e-9,h1=1.055e-34/pow(m0*V0,1.0/2)/a,
l=35e-9;
    double x0=0,xn=l/a,y0=0,yn=0,k=-10,x,h,tx0,xm,d,t;
    double v1[n+1];
    t=21.614;
    x=x0;
    h=(xn-x0)/n;
    for(i=0;i<=n;i++)
    {
        v1[i]=-V1(x)*t;
        x=x+h;
    }
    d=0.01;
    tx0=x0;
    xm=tx0+50*d;
    VTarget_2(n,x0,y0,xn,yn,xm,v1,k*t);
    return 0;
}
double V1(double x)
{
    double v;
    v=(3.0/35*x-1)*10;
    return v;
}
double VTarget_2(int n,double x0,double y0,double xn,double yn,double xm,double K[n+1],
double k)
{
    int i,l;
    double ty0,ty1,tyn,tyn1,d,x,h,f1,f2,tf,tk,ys,yb,a,Sum,t;
    double f[n-3];
    FILE *fp;
    d=0.1;
    h=(xn-x0)/n;
for(l=0;k<=0;l++)
{
    tk=k+d;
    f1=1;
    for(i=1;fabs(f1)>=0.000001;i++)
    {
        f1=F(n,x0,y0,xn,yn,xm,K,k);
        f2=F(n,x0,y0,xn,yn,xm,K,tk);
        if(f2*f1>=0)
        {
            f1=f2;
            k=tk;
```

```
            tk=tk+d;
            f2=F(n,x0,y0,xn,yn,xm,K,tk);
        }
        else
        {
            tf=F(n,x0,y0,xn,yn,xm,K,(tk+k)/2);
            if(tf*f1>=0)
            {
                k=(tk+k)/2;
                t=k;
                f1=tf;
            }
            else
            {
                tk=(tk+k)/2;
                t=tk;
                f2=tf;
            }
        }
        if(fabs(f1-f2)>=1000000)
        {
            k=k+d;
            tk=tk+d;
        }
    }
        a=Fa(n,x0,y0,xn,yn,xm,K,t);
        x=x0+h;
        ty0=y0;
        ty1=y0+0.001;
        tyn=yn;
        tyn1=yn+0.001;
        Sum=1.0/3*ty0*h+4.0/3*ty1*h;
        for(i=1;x<=xm;i++)
        {
            ys=(-(1+pow(h,2)*(t+K[i-1])/12)*ty0+2*(1-5*pow(h,2)*(t+K[i])/12)*ty1)/(1+pow
(h,2)*(t+K[i+1])/12);
            ty0=ty1;
            ty1=ys;
            f[i-1]=ys;
            if(i%2==0)
            {
                Sum=Sum+2.0/3*ys*h;
            }
            else
            {
                Sum=Sum+4.0/3*ys*h;
            }
            x=x+h;
        }
        x=xn-h;
        Sum=Sum+1.0/3*tyn*h+4.0/3*tyn1*h;
        for(i=n-1;x>xm;i--)
        {
            yb=(-(1+pow(h,2)*(t+K[i+1])/12)*tyn+2*(1-5*pow(h,2)*(t+K[i])/12)*tyn1)/(1+pow
(h,2)*(t+K[i-1])/12);
            tyn=tyn1;
            tyn1=yb;
            f[i-1]=yb*a;
            if(i%2==0)
            {
                Sum=Sum+2.0/3*yb*a*h;
            }
            else
            {
```

```
                Sum=Sum+4.0/3*yb*a*h;
            }
            x=x-h;
        }
        printf("%lf,%lf\n",a*yb,ys);

        x=x0+h;
        for(i=0;i<=n-2;i++)
        {
            x=x+h;
            fp=fopen("wavefuction.txt","a");
            fprintf(fp,"%lf,%lf,%lf\n",x,f[i]/Sum,t/21.614);
            fclose(fp);
        }
        k=tk;
}
    return 0;
}
double F(int n,double x0,double y0,double xn,double yn,double xm,double K[n+1],double k)
{
    double h=(xn-x0)/n,y1=y0+0.001,yn1=yn+0.001,ys,ss,yb,sb,a,x=x0+h;
    int i;
    for(i=1;x<=xm;i++)
    {
        ys=(-(1+pow(h,2)*(k+K[i-1])/12)*y0+2*(1-5*pow(h,2)*(k+K[i])/12)*y1)/(1+pow(h,2)*
(k+K[i+1])/12);
        ss=1.0/2/h*(-ys+4*y1-3*y0);
        y0=y1;
        y1=ys;
        x=x+h;
    }
    x=xn-h;
    for(i=n-1;x>xm;i--)
    {
        yb=(-(1+pow(h,2)*(k+K[i+1])/12)*yn+2*(1-5*pow(h,2)*(k+K[i])/12)*yn1)/(1+pow(h,2)*
(k+K[i-1])/12);
        sb=1.0/2/h*(-yn+4*yn1-3*yb);
        yn=yn1;
        yn1=yb;
        x=x-h;
    }
    a=ys/yb;
    sb=sb*a;
    return (sb-ss);
}
double Fa(int n,double x0,double y0,double xn,double yn,double xm,double K[n+1],double k)
{
    double h=(xn-x0)/n,y1=y0+0.001,yn1=yn+0.001,ys,yb,a,x=x0+h;
    int i;
    for(i=1;x<=xm;i++)
    {
        ys=(-(1+pow(h,2)*(k+K[i-1])/12)*y0+2*(1-5*pow(h,2)*(k+K[i])/12)*y1)/(1+pow(h,2)*
(k+K[i+1])/12);
        y0=y1;
        y1=ys;
        x=x+h;
    }
    x=xn-h;
    for(i=n-1;x>xm;i--)
    {
        yb=(-(1+pow(h,2)*(k+K[i+1])/12)*yn+2*(1-5*pow(h,2)*(k+K[i])/12)*yn1)/(1+pow(h,2)*
(k+K[i-1])/12);
        yn=yn1;
        yn1=yb;
```

```
        x=x-h;
    }
    a=ys/yb;
    return a;
}
```

B. 读数绘图部分

这是运用Python直接读取Excel数据绘制波函数图像的程序。

Input Python source:

```
import numpy as np
import openpyxl
from openpyxl import Workbook
import matplotlib
import matplotlib.pyplot as plt
wb=openpyxl.load_workbook('data.xlsx')
for i in range(36):
    sheet=wb.get_sheet_by_name('Sheet'+'%d'%(i+1))
    x=[];psix=[];E=float(sheet['C1'].value)
    for cell in list(sheet.columns)[0]:
        x.append(float(cell.value))
    for cell in list(sheet.columns)[1]:
        psix.append(float(cell.value))
    plt.figure()
    plt.plot(x,psix,'b')
    plt.title('The figure of wavefuction which intrinsic energy is '+'%f'%(E))
    plt.xlabel('x')
    plt.ylabel('$\psi(x)$')
    plt.grid()
    plt.savefig("%d.eps"%(i+1))
```

参考文献

[1] P. A. M. Dirac (1958). The Principles of Quantum Mechanics (4th ed.). Oxford University Press. ISBN 0-198-51208-2.

[2] B.H. Bransden & C.J. Joachain (2000). Quantum Mechanics (2nd ed.). Prentice Hall PTR. ISBN 0-582-35691-1.

[3] David J. Griffiths (2004). Introduction to Quantum Mechanics (2nd ed.). Benjamin Cummings. ISBN 0-13-124405-1.

[4] Hall, Brian C. (2013), Quantum Theory for Mathematicians, Graduate Texts in Mathematics, 267, Springer, ISBN 978-1461471158.

[5] Richard Liboff (2002). Introductory Quantum Mechanics (4th ed.). Addison Wesley. ISBN 0-8053-8714-5.

[6] Serway, Moses, and Moyer (2004). Modern Physics (3rd ed.). Brooks Cole. ISBN 0-534-49340-8.

[7] Schrödinger, Erwin (December 1926). "An Undulatory Theory of the Mechanics of Atoms and Molecules". Phys. Rev. 28 (6): 1049–1070. Bibcode:1926PhRv...28.1049S. doi:10.1103/PhysRev.28.1049.

[8] Teschl, Gerald (2009). Mathematical Methods in Quantum Mechanics; With Applications to Schrödinger Operators. Providence: American Mathematical Society. ISBN 978-0-8218-4660-5.

[9] Numerov, Boris Vasil'evich (1924), "A method of extrapolation of perturbations", Monthly Notices of the Royal Astronomical Society, 84: 592–601, Bibcode:1924MNRAS..84..592N, doi:10.1093/mnras/84.8.592.

[10] Numerov, Boris Vasil'evich (1927), "Note on the numerical integration of d2x/dt2 = f(x,t)", Astronomische Nachrichten, 230: 359–364, Bibcode:1927AN....230..359N, doi:10.1002/asna.19272301903.

[11] J Killingbeck, Shooting methods for the Schrodinger equation, J. Phys. A: Math. Gen. 20 (1987) 1411-1417. Printed in the U.K.

[12] https://zqyin.wordpress.com/2011/08/02/1dshrodinger/