Práctica Computacional: La Ecuación de Schrödinger Estados ligados por diagonalización [§]

A. Parte computacional

1. Utilizar el *notebook* DIAG.NB para obtener los estados ligados del oscilador armónico.
2. Comparar los resultados obtenidos numéricamente con los teóricos. Variar los parámetros y analizar la convergencia.
3. Comparar las funciones de onda con las teóricas.

Diagonalizando con Mathematica

Rango y Numero de Puntos

```
xmin = 0 ;
xmax = 5 ;
npts = 500 ;
Δ = (xmax - xmin) / npts ;
```

Creacion del Hamiltoniano

```
m = Table[ If [i == j - 1 , -0.5 / (Δ^2),
            If [i == j, 1.0 / (Δ^2) + 0.5 (i * Δ) ^2,
            If [i == j + 1, -0.5 / (Δ^2), 0. ] ] ] ,
     {i, npts}, {j, npts}] ;
```

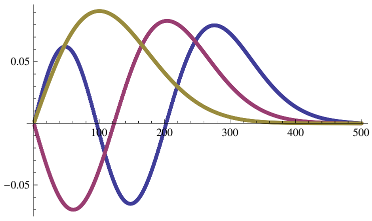
Autovalores (Energias)

```
In[103]:= Eigenvalues[m, -5]

Out[103]= {9.53424, 7.5024, 5.4999, 3.49992, 1.49998}
```

Primeros Autovectores

```
In[104]:= Vvarios = Eigenvectors[m, -3];

In[105]:= ListPlot[Vvarios]
```

Out[105]=




```
In[106]:= Vg = Eigenvectors[m, -3][[-1]];
```

§http://www.df.uba.ar/users/dmitnik/computation/diagonal/diagonal.html

B. Implementación en Fortran

El método de diagonalización directa será explicado brevemente en clase, y el alumno recibira un programa básico (WELL.FOR) con el cual podrá realizar los ejercicios. El programa no está optimizado, su objetivo es pedagógico. Para hacer el programa mas simple, la subrutina de diagonalización (TQLI.FOR) se compila en forma separada.

Se puede obtener una introducción sobre cómo compilar y cómo correr este programa, en

`http://www.df.uba.ar/users/dmitnik/computation/oscillator/intro.html`

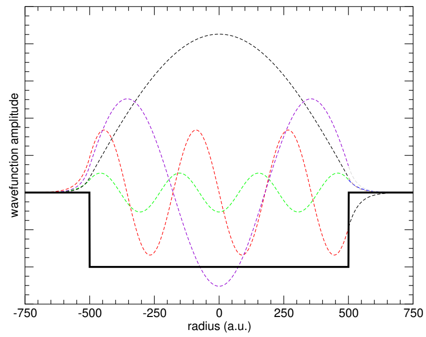


FIG. 1. Algunas de las funciones halladas utilizando el programa WELL.FOR. Los parámetros iniciales se dan como datos dentro del programa.

1. Aumente progresivamente la profundidad del pozo de potencial
   (a) ¿Qué ocurre con el número de funciones halladas?
   (b) ¿Qué ocurre con el número de nodos de ellas?
   (c) ¿Qué ocurre con las energías?
2. Explore cómo cambian las energías variando el radio a
3. ¿Se puede determinar la profundidad del pozo conociendo la energía del estado básico?
4. Explore la paridad de las funciones
5. Modifique el programa y trate de obtener soluciones para otros potenciales. En particular, verifique sus respuestas a las preguntas II-B-7 y 8.

I. APÉNDICE: PROGRAMA WELL.FOR

Printed by Dario Mitnik

May 30, 05 15:14 **well.for** Page 3/4

```
      program potential_well                         !! 30 May 2005
c----------sorting the eigenvectors/values into ascending energies
c-------Generates complete set of orbitals by diagonalization
      call sort2(nmesh,d,v,mxpts,wksp,iwksp)
c.......by Dario Mitnik
c------ store all bound state wavefunctions
      open(unit=10,file='orbitals.dat',status='unknown')
      open(unit=11,file='potentialwell.dat',status='unknown')
      do 10 nq=1,nrmesh
      sign=one
            ener(nq) = d(nq)
c...... data input
      call readdata
            if (ener(nq).lt.0) print45, nq,ener(nq)
45          format(5x,'n =:',i3,10x,'energy =:',f15.4,' a.u.')
c.......construction of Hamiltonian
      call constrH
            sum=zero
            do 100 irow=1,nrmesh
c...... diagonalization
100         sum=sum+v(irow,nq)**2
      call diagH
            ptot=hr*sum
            if(v(1,nq).lt.zero) sign=-one
c....... output
            do 200 irow=1,nrmesh
200   call printH
            chi(nq,irow)=sign*v(irow,nq)/sqrt(ptot)
500   continue
      stop
      return
      end
      end
c
c*************************************************************************
c
      subroutine readdata
      subroutine printH
c-------Reads input data
c-------Prints the energies and selected bound wavefunctions.
c.......parameters
      parameter(mxpts=500)
c     common/bkmesh/rmin,rmax,hr,nrmesh
      parameter(mxpts=500)
      common/bkeigen/ener(mxpts),chi(mxpts,mxpts)
      character*30 outputfile
      common/bkmesh/rmin,rmax,hr,nrmesh
      common/bkpot/apot,V0
      write(10,5)
5     format(/5x,'Results:')
      write(10,6)
5     format(/5x,'Potential Well: Energies and Wavefunctions:')
6     format(1x,f5.1,f8.4)
      do 20 nq=1,nrmesh
      if (ener(nq).lt.0) write(10,45) nq,ener(nq)
20    continue
45    format(5x,'n =:',i5,10x,'energy =:',f15.4,' a.u.')
c-------wavefunction parameters
      inptfile = 11
c-------output file parameters
      outputfile(1:5)='well'
50    format(a25)
c.......integration interval
100   print*,'give the n quantum number (0 to exit)'
      print*,'give the name of the output file (extension only)'
c.......mesh points
      read*,nq
      if(nq.le.0) return
c.......rmin=0, rmax=nrmesh*hr
      print*,'give the name of the output file (extension only)'
      read(*,50) outputfile(6:30)
c.......default parameters
      open(unit=inptfile,file=outputfile,status='unknown')
      open(1
      nrmesh=100
      do 75 irow=1,nrmesh
      print*,'give the space integration interval hr (a.u.):'
      print*,'give the space integration interval hr (a.u.):'
75    write(inptfile,*) r,' ',chi(nq,irow)+ener(nq)
      read*,hr
      print*,'give the number of total steps (nrmesh) (max=',mxpts,')'
      close(inptfile,status='keep')
      read*,nrmesh
      goto 100
      write(10,220), hr,nrmesh
220   format(5x,'hr=:',f10.4,2x,'nrmesh=:',i5)
      if(nrmesh.gt.mxpts) pause 'increase mxpts parameter'
      return
      end
      rmin=0.0
c     rmax=nrmesh*hr
c*************************************************************************
      write(10,330) rmin,rmax
330   format(5x,'rmin=:',f10.4,5x,'rmax=:',f10.4)
      function Vpot(x)
```

May 30, 05 15:14 **well.for** Page 4/4

```
c-------potential V parameters
c-------Generates the Potential.
c.......V0--deep of the potential
c.......a--range
      parameter(mxpts=500)
      common/bkmesh/rmin,rmax,hr,nrmesh
      common/bkpot/apot,V0
      print*,'give the deep of the potential V0 (a.u.):'
      read*,V0
      common/bkpot/apot,V0
400   print*,'give the potential range a (a.u.):'
      read*,apot
      data half,one,two,five/0.5,1.0,2.0,5.0/
      write(10,440), V0,apot
c-------building potential on the grid
      if (apot.gt.rmax) then
         x0 = (rmax + rmin)/two
         print*,' this potential is too broad for your mesh'
         go to 400
      endif
         rlow  = x0-apot/two
440   format(5x,'V0=:',f10.4,2x,'a=:',f10.4)
         rhigh = x0+apot/two
         Vpot = 0.0
      return
         if (x.gt.rlow.and.x.le.rhigh)  Vpot=-apot
      end
c        write(40,*) x,' ',Vpot
c*************************************************************************
c        return
      subroutine constrH
      end
c
c*******Generates the Hamiltonian.****************************************
c
      parameter (mxpts=500)
      common/bkmesh/rmin,rmax,hr,nrmesh
      common/bkhamilt/d(mxpts),e(mxpts)

      data half,one,two/0.5,1.0,2.0/

      write(10,50)
50    format(//5x,'Construction of complete set of orbitals')
      write(10,*) ' '

c----------building hamiltonian on the grid
         x0 = (rmax + rmin)/two
         do 40 i=1,nrmesh
            r = i*hr
c..............diagonal terms
            d(i)= one/hr**2 + Vpot(r)
c..............non-diagonal terms
            e(i)=-half/hr**2
40       continue

      return
      end
c
c*************************************************************************
c
      subroutine diagH

c------- Matrix diagonalization.
c------- Also stores the Eigenvalues and Eigenvectors.

      parameter (mxpts=500)
      common/bkmesh/rmin,rmax,hr,nrmesh
      common/bkhamilt/d(mxpts),e(mxpts)
c     d(mxpts)                        ! diagonal part of the Hamiltonian
c     e(mxpts)                        ! non-diagonal part of H
      dimension  v(mxpts,mxpts)       ! eigenvectors
      dimension  wksp(mxpts),iwksp(mxpts) ! working arrays for sorting
      common/bkeigen/ener(mxpts),chi(mxpts,mxpts)
c     chi(n,mxpts)            ! normalized eigenvectors

      data zero,one/0.0,1.0/

c----------matrix diagonalization
         call tqli(d,e,nrmesh,mxpts,v)
```

Tuesday May 31, 2005   well.for   2/2

FIG. 2. Programa WELL.FOR para la solución de la ecuación de Schrödinger por diagonalización.