PROBLEM SET 3 Due Friday May 3

Physics 140B– SPRING 2013

Analytic:

[1.] Sidebottom 13-7.

[2.] Sidebottom 13-8.

[3.] Consider the Hamiltonian

$$H = -\sum_{l=1}^{N} \left(c_{l+1}^{\dagger} c_l + c_l^{\dagger} c_{l+1} \right) + \Delta \, c_{N/2}^{\dagger} c_{N/2}$$

This is our friend the one-dimensional hopping model, except there is a single special site. (I have placed it at the chain "center" $l = N/2$, but with periodic boundary conditions that designation is really meaningless.) Using our occupation number basis, write down the matrix for H for states with a single electron. When $\Delta = 0$ what are the eigenvectors and eigenvalues? What are the participation ratios?

Numeric:

I will be available in the computer lab, room 106, Wednesday noon - 2:00 pm to help anyone who needs assistance with these problems.

[4.] **(Optional)** For those of you who did not take Physics 140A, it might be useful to practice your coding and also your understanding of the finite square well problem by doing this 140A homework problem:

We often encounter transcendental equations in physics. For example, in the quantum problem of the energy levels of a particle in a finite square well, you need to solve,

$$\tan z = \sqrt{(z_0/z)^2 - 1}$$

Write a bisection program (do *not* use some canned root-finding software) and find the solution to this equation for $z_0 = 2.5$. Looking at the figure 2.18 in Griffiths will help you pick a good set of initial values to bracket the solution. (For this problem you can just treat z_0 as number. However, for completeness, let me remind you of the connection of z_0 to the physics: $z_0 = (a/\hbar)\sqrt{2mV_0}$ where $2a$ is the well width, V_0 is the well depth, and m is the particle mass. For $z_0 = 2.5$ it turns out there is just a single bound state.)

[5.] Diagonalize the Hamiltonian from problem 3 numerically. Set $t = 1$, $\Delta = -0.1$, and $N = 64$. Look at the eigenvalues and participation ratios. Does anything special happen? How about when $\Delta = -0.2$? What if $\Delta = -0.5$? Make plots of the square of the components of the "funny" wavefunction (if you find one) for $\Delta = -0.3, -0.4, -0.5, -0.6$. Notice that this problem is very similar to the part of Problem 8 of Assignment 1 where you did the one electron sector of a $N = 6$ site chain. The only differences are (i) that $N = 64$ is probably large enough that you would want to have the computer set up the matrix for you, instead of coding up each of the nonzero matrix elements. (there are 128 of them); and (ii) one of the diagonal entries, $H(32, 32)$ is now non-zero.

1//

Physics 140B PS3 Solns

(1) (Sidebottom 13-7)

We make a number of simplifications in the limit

$b \to \emptyset$ at fixed $V_0 b$ (so that V_0 becomes large)

i) $d = a + b \approx a$

ii) $\cosh \beta b \approx 1$

iii) $\sinh \beta b \approx \beta b$

iv) $\beta^2 = \frac{2m}{\hbar^2}(V_0 - E) \gg \alpha^2 = \frac{2mE}{\hbar^2}$ so $\beta^2 - \alpha^2 \approx \beta^2$

Then 13.17 becomes

$$\frac{\beta^2 - \alpha^2}{2\alpha\beta} \sinh(\beta b)\sin(\alpha a) + \cosh(\beta b)\cos(\alpha a) = \cos k(a+b)$$

βb $\quad$ 1 $\quad$ αd $\quad$ d

β^2

$$\underbrace{\frac{\beta^2 b}{2\alpha}} \sin\alpha a + \cos(\alpha d) = \cos(kd)$$

$$\underbrace{\frac{\beta^2 b a}{2}} \frac{1}{\alpha d} \qquad \text{(using } a \approx d\text{)}$$

$m V_0 b a / \hbar^2$ completing the derivation of 13.18

② (Sidebottom 13-8)

a) $$P\frac{\sin \alpha d}{\alpha d} + \cos \alpha d = \cos kd$$

$\cos kd = 1$ for $k = 0$

Define $\alpha d = x$

$$P\frac{\sin x}{x} + \cos x = 1$$

We are told P is small also. Expanding about $x = 0$

$$P\left[\frac{x - x^3/6 + \cdots}{x}\right] + \left[1 - \frac{x^2}{2} + \cdots\right] = 1$$

$$P\left[1 - \frac{x^2}{6} + \cdots\right] - \frac{x^2}{2} = \phi$$

$Px^2 \ll x^2$, if P small so $x^2 = 2P$

$$E = \frac{\hbar^2}{2m}\alpha^2 = \frac{\hbar^2}{2m}\left(\frac{x}{d}\right)^2 = \frac{\hbar^2}{2md^2}2P = \frac{\hbar^2 P}{md^2}$$

b) Instead of $k = 0$ we now look at $kd = \pi$.

We will find two solutions and want the gap between them

↳ the top of the first band and the bottom of the second band

3//

$kd = \pi$ ← $\cos kd = -1$

$$P\frac{\sin x}{x} + \cos x = -1$$

One soln has $x = \pi = \alpha d$

and hence $E = \frac{\hbar^2}{2m}\alpha^2 = \frac{\hbar^2}{2m}\frac{\pi^2}{d^2}$

To get second soln, expand around $x = \pi$ ie $x = \pi + \epsilon$

$$P\frac{\sin(\pi+\epsilon)}{\pi+\epsilon} + \cos(\pi+\epsilon) = -1$$

$-\epsilon$ ↑ $-1 + \epsilon^2/2$

$$-P\frac{\epsilon}{\pi} + -1 + \epsilon^2/2 = -1$$

$$\epsilon\left(\frac{-P}{\pi} + \frac{1}{2}\epsilon\right) = 0 \qquad \epsilon = 0 \text{ (old soln)}$$

$$\epsilon = \frac{2P}{\pi}$$

~~$x = \alpha d$~~ ~~$\epsilon = \frac{2P}{\pi}$~~

$$x = \alpha d = \pi + 2P/\pi$$

$$E = \frac{\hbar^2}{2m}\left(\frac{\pi}{d} + \frac{2P}{\pi}\right)^2 = \frac{\hbar^2}{2m}\left(\frac{\pi^2}{d^2} + \frac{4P}{d} + \ldots\right)$$

$$E = \underbrace{\frac{\hbar^2}{2m}\frac{\pi^2}{d^2}}_{\text{first soln}} + \underbrace{\frac{2\hbar^2 P}{md}}_{\text{gap to second soln}}$$

From (a) this is twice the ground state energy at $k = 0$

$E = \hbar^2 P/md^2$

PBC ↓

(3) $H|1000\ldots\rangle = -t|0100\ldots\rangle - t|00\ldots1\rangle$

$H|0100\ldots\rangle = -t|1000\ldots\rangle - t|0010\ldots\rangle$

⋮

$H|00\ldots010\ldots\rangle$ has extra term

↑ $N/2$ slot

$\Delta|00\ldots010\ldots\rangle$

↑ $N/2$ slot

Matrix for H is

column $N/2$

$$\begin{pmatrix} 0 & -t & 0 & 0 & \cdots \\ -t & 0 & -t & 0 & \\ 0 & -t & 0 & -t & \\ & & & & \\ \text{row } \frac{N}{2} & & & -t\ \Delta\ -t & \end{pmatrix}$$

Extra "Δ" on diagonal at $i = j = N/2$

When $\Delta = 0$ $\quad \epsilon_k = -2t\cos k$

ℓth component of kth eigenvector $\rightarrow$ $V_\ell^k = \frac{1}{\sqrt{N}} e^{ik\ell}$ $\qquad \frac{1}{I_k} = \frac{1}{\sum_\ell |V_\ell^k|^4} = N$

```
/* defect_atom.c */

#include <stdio.h>
#include "nrutil.h"

void jacobi(float **a, int n, float d[], float **v, int *nrot);

int main() {

  int N  /* dimension of the matrix */
    , num_rot  /* number of Jacobi applied */
    , i, j;  /* temp. variables for counter */
  float ** A, ** eigenvectors, * eigenvalues;
  char format[] = "\n %7.3f";  /* format of output */
  float t,delta,P,compsq;

  printf("dimension of the matrix: ");  scanf("%i", &N);
  // N=64;
  t=1.0;
  printf("impurity potential: ");  scanf("%f", &delta);
  // delta=-1.0;

  /* the following statement declare FORTRAN-style matrix and vector for NR */
  A = matrix( 1, N, 1, N);  /* must be a symmetric real matrix */
  eigenvectors = matrix( 1, N, 1, N);  /* eigenvectors as columns */
  eigenvalues = vector( 1, N);

  /* get input of A */
//  printf("enter a %i x %i matrix (separated by space):\n", N, N);
// for( i = 1; i <= N; i++)
//  for( j = 1; j <= N; j++)
//   scanf( "%f", &A[i][j]);

 for( i = 1; i <= N; i++)
  for( j = 1; j <= N; j++)
   A[i][j]=0.0;

  for( i = 2; i <= N-1; i++){
   A[i][i-1]=-t;
   A[i][i+1]=-t;
   }
   A[1][N]=-t;
   A[1][2]=-t;
   A[N][N-1]=-t;
   A[N][1]=-t;

   A[N/2][N/2]=delta;

//  puts("\neigen problem for matrix A:");
//  for( i = 1; i <= N; i++) {
//    for( j = 1; j <= N; j++)
//      printf(format, A[i][j]);
//    printf("\n");  /* next line */
//  }

  /* solve the eigen problem with NR jacobi routin (A will be changed) */
  jacobi( A, N, eigenvalues, eigenvectors, &num_rot);

  /* output the result here */
  printf( "\nnumber of Jacobi applied: %i\neigenvalues:\n", num_rot);
  for( i = 1; i <= N; i++)
//    printf(format, eigenvalues[i]);
    printf("\n %6i   %10.5f",i,eigenvalues[i]);/* next line */
```

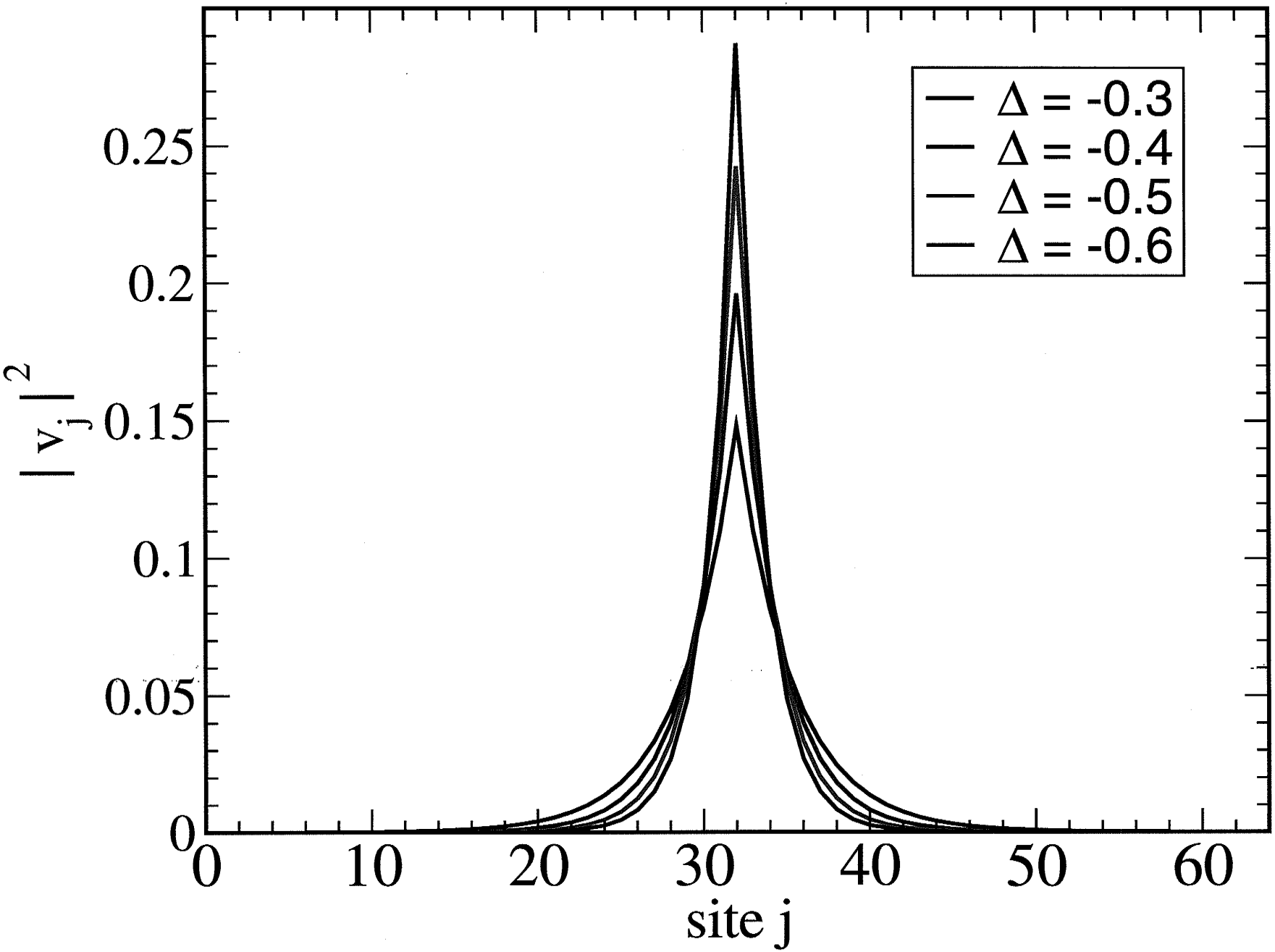
```
  printf("\n");  /* next line */

  puts( "\neigenvectors:");
  for( j = 1; j <= N; j++) {
    printf("\n %i ",j);
    for( i = 1; i <= N; i++)
    {
    compsq=eigenvectors[i][j]*eigenvectors[i][j];
    printf("\n %6i   %10.5f   %10.5f",i,eigenvectors[i][j],compsq);
    }
    printf("\n");  /* next line */
  } printf("\n");  /* next line */


    printf("\n PARTICIPATION RATIOS \n");
  for( i = 1; i <= N; i++) {
    P=0.;
    for( j = 1; j <= N; j++)
    P=P+eigenvectors[j][i]*eigenvectors[j][i]*eigenvectors[j][i]*eigenvectors[j][
i];
    printf("\n");
    P=1./P;
    printf("%i %lf",i,P);/* next line */
  } printf("\n");  /* next line */

  /* free memory here (strange NR style :-( )*/
  free_matrix( A, 1, N, 1, N);  free_matrix( eigenvectors, 1, N, 1, N);
  free_vector( eigenvalues, 1, N);

  return 0;
}
```


Δ = -0.3
Δ = -0.4
Δ = -0.5
Δ = -0.6
0.25
0.2
0.15
0.1
0.05
0
$|v_j|^2$
0
10
20
30
40
50
60
site j