2D test: Open box (=hard-wall), $x,y=[-1,1]$, $V=0$

Energy as a function of Lanczos basis size $M$

<table>
<thead>
<tr>
<th>$N=200\times200$</th>
<th>$20$</th>
<th>$146.53700$</th>
<th>$731.057995$</th>
<th>$1807.662851$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$40$</td>
<td>$36.89144$</td>
<td>$197.708305$</td>
<td>$466.352106$</td>
</tr>
<tr>
<td></td>
<td>$60$</td>
<td>$19.78221$</td>
<td>$88.403669$</td>
<td>$216.571047$</td>
</tr>
<tr>
<td></td>
<td>$80$</td>
<td>$14.33864$</td>
<td>$52.011927$</td>
<td>$120.846453$</td>
</tr>
<tr>
<td></td>
<td>$100$</td>
<td>$11.36276$</td>
<td>$33.130912$</td>
<td>$78.125621$</td>
</tr>
<tr>
<td>Convergence</td>
<td>$120$</td>
<td>$9.836334$</td>
<td>$25.714048$</td>
<td>$59.319176$</td>
</tr>
<tr>
<td></td>
<td>$140$</td>
<td>$9.093991$</td>
<td>$21.690312$</td>
<td>$43.007222$</td>
</tr>
<tr>
<td></td>
<td>$160$</td>
<td>$8.460393$</td>
<td>$17.444110$</td>
<td>$31.878198$</td>
</tr>
<tr>
<td></td>
<td>$180$</td>
<td>$7.719381$</td>
<td>$13.667132$</td>
<td>$26.425252$</td>
</tr>
<tr>
<td></td>
<td>$200$</td>
<td>$6.494491$</td>
<td>$10.987755$</td>
<td>$22.538540$</td>
</tr>
<tr>
<td></td>
<td>$240$</td>
<td>$5.310526$</td>
<td>$9.837896$</td>
<td>$18.969252$</td>
</tr>
<tr>
<td></td>
<td>$280$</td>
<td>$3.925524$</td>
<td>$7.766020$</td>
<td>$11.142274$</td>
</tr>
<tr>
<td></td>
<td>$320$</td>
<td>$2.815453$</td>
<td>$6.526244$</td>
<td>$10.305297$</td>
</tr>
<tr>
<td></td>
<td>$360$</td>
<td>$2.482839$</td>
<td>$6.177734$</td>
<td>$9.963101$</td>
</tr>
<tr>
<td></td>
<td>$400$</td>
<td>$2.447032$</td>
<td>$6.119496$</td>
<td>$9.841073$</td>
</tr>
<tr>
<td></td>
<td>$440$</td>
<td>$2.443210$</td>
<td>$6.108594$</td>
<td>$9.789598$</td>
</tr>
<tr>
<td></td>
<td>$480$</td>
<td>$2.442875$</td>
<td>$6.106960$</td>
<td>$9.772831$</td>
</tr>
<tr>
<td>Exact</td>
<td></td>
<td>$2.467401$</td>
<td>$6.168503$</td>
<td>$9.869604$</td>
</tr>
</tbody>
</table>

The method works better in 2D.
The components (i.e., individual basis states) of the initial state must be propagated by the Hamiltonian through the whole system in order for an extended wave function to be representable in the Lanczos basis.

Regions covered after successive operations with $H$ ("generations") on a single basis state in 1D and 2D.

Covered fraction scales as the number of generations $M$ in any $D$

- The Lanczos scheme is advantageous in 2D and 3D.
Ground state of a 100*100 box vs number of iterations (N)

Graphing:

\[ |\Psi_0(x, y)|^2 \]

Starting from a state localized in the center of the box
Ground state of a 100*100 box vs number of iterations (N)

Graphing:

$|\Psi_0(x, y)|^2$

Starting from a random state
First excited state of a 100*100 box vs number of iterations (N)

Graphing:

$$|\Psi_0(x, y)|^2$$

Starting from a random state
Second excited state of a 100*100 box vs number of iterations (N)

Graphing:

$|\Psi_0(x, y)|^2$

Starting from a random state
The first excited state should be doubly-degenerate

- Lanczos only gives one state out of a degenerate multiplet

Go back to the Krylov space

\[ H^m |\Psi\rangle = \sum_k C_k E^m_k |\Psi_k\rangle \]

If states \( k, j \) are degenerate, we have a term

\[ E^m_j (C_j |\Psi_j\rangle + C_k |\Psi_k\rangle) \]

For any \( m \), this vector points in the same direction in the subspace spanned by \( |\Psi_j\rangle, |\Psi_k\rangle \)

Acting with \( H \) cannot “separate” degenerate states

Since the Lanczos basis spans the same Krylov space, we only get one state out of a degenerate multiplet of states

- the particular linear combination depends on the initial state

Numerical round-off errors can lead to apparent degeneracies (multiple copies of the same state). This indicates that the scheme breaks down as the basis becomes non-orthogonal.
Potential problem:
The normalization constants $N_m$ can become very large (think of $E_0^m$)

Solution:
generate the normalized basis directly

- start with $|\phi_0\rangle$ arbitrary, normalized, and then

$$
|\phi_1\rangle = \frac{1}{N_1} \left( H|\phi_0\rangle - a_0 |\phi_0\rangle \right).
$$

$$
|\phi_{m+1}\rangle = \frac{1}{N_{m+1}} \left( H|\phi_m\rangle - a_m |\phi_m\rangle - N_m |\phi_{m-1}\rangle \right) = \frac{|\gamma_m\rangle}{N_{m+1}}
$$

The definition of $N_m$ is different, and no $b_m$:

$$
a_m = \langle \phi_m | H | \phi_m \rangle
$$

$$
N_m = \langle \gamma_m | \gamma_m \rangle^{-1/2}
$$

Generate $|\gamma_m\rangle$ first, normalize to get $N_{m+1}$

The $H$-matrix is

$$
\langle \phi_{m-1} | H | \phi_m \rangle = N_m
$$

$$
\langle \phi_m | H | \phi_m \rangle = a_m
$$

$$
\langle \phi_{m+1} | H | \phi_m \rangle = N_{m+1}
$$
Example in two dimensions: box with open boundaries

Constructing $\mathbf{H}|f_n\rangle$ (open corresponds to hard walls)

State $n$ stored in $f_1(nx*ny)$
State $\mathbf{H}|f_n\rangle$ constructed in $f_2(nx*ny)$

$t =$ hopping (kinetic) matrix element
- consider hopping into all boxes $j$

**subroutine hoperation(f1,f2)**

```fortran
  f2(:)=vpot(:)*f1(:)
  do j=1,nx*ny
    x=1+mod(j-1,nx)
    y=1+(j-1)/nx
    if (x.ne.1)  f2(j-1)=f2(j-1)-t*f1(j)
    if (x.ne.nx) f2(j+1)=f2(j+1)-t*f1(j)
    if (y.ne.1)  f2(j-nx)=f2(j-nx)-t*f1(j)
    if (y.ne.ny) f2(j+nx)=f2(j+nx)-t*f1(j)
  enddo
```

Constructing

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>13</td>
<td>14</td>
<td>15</td>
<td>16</td>
</tr>
<tr>
<td>9</td>
<td>10</td>
<td>11</td>
<td>12</td>
</tr>
<tr>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
</tbody>
</table>

labeling for 4*4 elements
One step in the iteration of the a and b coefficients

\[ |f_1\rangle = H |f_0\rangle - a_0 |f_0\rangle \]
\[ |f_{n+1}\rangle = H |f_n\rangle - a_n |f_n\rangle - b_{n-1} |f_{n-1}\rangle \]
\[ a_n = H_{nn}/N_n, \quad b_{n-1} = N_n/N_{n-1} \]

if (m==1) then
    call hoperation(f0,f1)
    aa(0)=dot_product(f0,f1)
    f1=f1-aa(0)*f0
    nn(1)=dot_product(f1,f1)
endif

else
    call hoperation(f1,f2)
    aa(m-1)=dot_product(f1,f2)/nn(m-1)
    bb(m-2)=nn(m-1)/nn(m-2)
    f2=f2-aa(m-1)*f1-bb(m-2)*f0
    nn(m)=dot_product(f2,f2)
    f0=f1
    f1=f2
endif

The method of constructing the normalized states directly is very similar (program on-line)

here m=n+1
The full basis and Hamiltonian construction

Random initial state

do  i=1,n
    psi(i)=rand()-0.5d0
enddo

norm=1.d0/sqrt(dot_product(psi,psi))

psi(:)=psi(:)*norm

Perform \texttt{niter} Lanczos steps and diagonalize

f\theta(:)=psi(:)

\texttt{nn(0)=1.d0}

Do \texttt{m=1,niter}
    perform code on previous page
enddo

d(:)=aa(:)

\texttt{e(:)=sqrt(bb(:))}

call diatri(niter,d,e,eig,states)
Calculation of the states

In order to calculate states (wave functions) we have to perform another Lanczos procedure, since we have not saved all the states \(|f_n\rangle\).

If we want the m-th lowest state, we transform with the m-th eigenvector obtained in the diagonalization. The eigenvectors are in the matrix \text{states}; \texttt{vec=states(:,m)}

\[
\begin{align*}
  f_0 &= \psi_0 \\
  \psi &= \psi_0 \times \text{vec}(0) \\
  \text{call hoperation}(n,f_0,f_1) \\
  f_1 &= f_1 - aa(0) \times f_0 \\
  \psi &= \psi_0 + \text{vec}(1) \times f_1 / \sqrt{nn(1)} \\
  \text{do } i = 2, niter-1 \\
    \text{call hoperation}(n,f_1,f_2) \\
    f_2 &= f_2 - aa(i-1) \times f_1 - bb(i-2) \times f_0 \\
    \psi &= \psi_0 + \text{vec}(i) \times f_2 / \sqrt{nn(i)} \\
    f_0 &= f_1 \\
    f_1 &= f_2 \\
  \text{enddo}
\end{align*}
\]

Normalized states \(|\phi_n\rangle = N_n^{-1/2} |f_n\rangle\)