The first excited state should be doubly-degenerate

- Lanczos only gives one state out of a degenerate multiplet

Go back to the Krylov space

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

If states k, j are degenerate, we have a term

\[ E_j^m (C_j |\Psi_j\rangle + C_k |\Psi_k\rangle) \]

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.

- Avoid by storing all states and orthogonalizing at each step
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 $b_m$:

$$a_m = \langle \phi_m | H | \phi_m \rangle$$
$$N_m = \sqrt{\langle \gamma_m | \gamma_m \rangle}$$

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 $H|f_n\rangle$ (open corresponds to hard walls)

State $n$ stored in $f1[1:nx*ny]$
State $H|f_n\rangle$ constructed in $f2[1:nx*ny]$

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

function hoperation(lx,ly,vv,f1,f2)
  f2.=vv.*f1
  for j=1:lx*ly
    x=1+mod(j-1,lx)
    y=1+div(j-1,lx)
    x != 1 ? f2(j-1)=f2(j-1)-t*f1(j) : nothing
    x != lx ? f2(j+1)=f2(j+1)-t*f1(j) : nothing
    y != 1 ? f2(j-nx)=f2(j-nx)-t*f1(j) : nothing
    if y != ly ? f2(j+nx)=f2(j+nx)-t*f1(j) : nothing
  end
end

labeling for 4*4 elements

<table>
<thead>
<tr>
<th>13</th>
<th>14</th>
<th>15</th>
<th>16</th>
</tr>
</thead>
<tbody>
<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>
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 \( m = 1 \)

\[ \text{hoperation}(lx, ly, vv, f0, f1) \]
\[ aa[1] = \text{dot}(f0, f1) \]
\[ @. f1 = f1 - aa[1] * f0 \]
\[ nn[2] = \text{dot}(f1, f1) \]

else

\[ \text{hoperation}(lx, ly, vv, f1, f2) \]
\[ aa[m] = \text{dot}(f1, f2) / nn[m] \]
\[ bb[m-1] = nn[m] / nn[m-1] \]
\[ @. f2 = f2 - aa[m] * f1 - bb[m-1] * f0 \]
\[ nn[m+1] = \text{dot}(f2, f2) \]
\[ f0. = f1 \]
\[ f1. = f2 \]
end

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

here \( m=n+1 \),
- \( f0 \) in the first step is normalized
- \( nn[1]=1 \)
The full basis and Hamiltonian construction

Random initial state

```matlab
for i=1:n
    psi[i]=rand()-0.5d0
end

norm=1/dot(psi,psi)^0.5
psi.=psi.*norm
```

Perform \texttt{niter} Lanczos steps and diagonalize

```matlab
f0.=psi
nn[1]=1
for m=1:niter
    perform code on previous page
end

d.=aa
e.=bb.^0.5
diagonalize matrix with elements d,e
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 $|f_n>$.

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 \texttt{states}; \texttt{vec=states[:,m]}

\[
f_0. = \psi
\]
\[
\psi. = \psi.*vec[1]
\]
\[
\text{hoperation}(lx,ly,vv,f0,f1) @. f1=f1-aa[1]*f0
\]
\[
\psi=\psi+vec[2]*f1/nn[2]^0.5
\]
for \texttt{i=2:niter-1}
\[
\text{hoperation}(lx,ly,vv,f1,f2) @. f2=f2-aa[i]*f1-bb[i-1]*f0
\]
\[
@. \psi=\psi+vec[i+1]*f2/nn[i+1]^0.5
\]
\[
f0.=f1
\]
\[
f1.=f2
\]
end

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