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

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

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

The definition of N_m is different, and no b_m:

$$\begin{aligned} a_m &= \langle\phi_m|H|\phi_m\rangle \\ N_m &= \langle\gamma_m|\gamma_m\rangle^{+1/2} \end{aligned}$$

Generate $|\gamma_m\rangle$ first, normalize to get N_n

The H-matrix is

$$\begin{aligned} \langle\phi_{m-1}|H|\phi_m\rangle &= N_m \\ \langle\phi_m|H|\phi_m\rangle &= a_m \\ \langle\phi_{m+1}|H|\phi_m\rangle &= N_{m+1} \end{aligned}$$

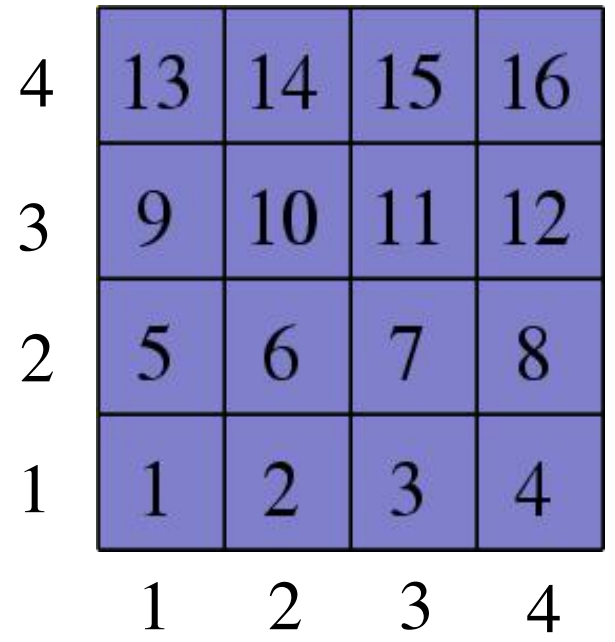
Example in two dimensions: box with open boundaries

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

State n stored in `f1[1:nx*ny]`

State $H|f_n\rangle$ in `f2[1:nx*ny]`

labeling for 4*4 elements



t = hopping (kinetic) matrix element

- consider hopping from all boxes j

```
function hoperation(f1,f2)
f2.=vpot.*f1
for j=1:nx*ny
    x=1+mod(j-1,nx)
    y=1+div(j-1,nx)
    if x!=1  f2[j-1]=f2[j-1]-t*f1[j]
    if x!=nx f2[j+1]=f2[j+1]-t*f1[j]
    if y!=1  f2[j-nx]=f2[j-nx]-t*f1[j]
    if y!=ny f2[j+nx]=f2[j+nx]-t*f1[j]
end
```

One step in the iteration of the a and N coefficients

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

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

```
if m==1
   hoperation(f0,f1)
   aa[1]=dot(f0,f1)
   f1.=f1.-aa[1].*f0
   nn[2]=dot(f1,f1)^0.5
   f1.=f1./nn[2]
else
   hoperation(f1,f2)
   aa[m]=dot(f1,f2)
   f2.=f2.-aa[m].*f1-nn[m].*f0
   nn[m+1]=dot(f2,f2)^0.5
   f2.=f2./nn[m+1]
   f0.=f1
   f1.=f2
end
```

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

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

The method of constructing the normalized states directly

Note:
aa[m] contains a_{m-1}
nn[m] contains N_{m-1}

The full basis and Hamiltonian construction

Random initial state

```
for i=1:n
   psi[i]=rand()-0.5
end
norm=1./dot(psi,psi)^0.5
psi.=psi.*norm
```

Perform `niter` Lanczos steps and diagonalize

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

Diagonalize the matrix of size niter*niter made using the diagonal and subdiagonal elements from aa and nn

Example in progran random.ipynb online

Calculation of the states

In order to calculate states (wave functions) we have to perform another Lanczos procedure, unless we have saved all the states $|f_n\rangle$ (which we can if there is enough memory)

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 `states`; `vec=states(:,m)`

```
f0.=psi
psi.=psi.*vec[1]
hoperation(f0,f1)
f1.=f1.-aa[1].*f0
psi.=psi.+vec[2].*f1./nn[2]
for i:2,niter-1
   hoperation(f1,f2)
   f2.=f2.-aa[i].*f1.-nn[i].*f0
   psi.=psi.+vec[i+1].*f2/nn[i+1]
   f0.=f1
   f1.=f2
end
```

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

If the ϕ vectors are not too large, they can also all be stored and this procedure is then simpler