Title: Quantum Spin Simulations (PHYS 7380) - Lecture 9

Date: Apr 15, 2010 11:00 AM

URL: http://pirsa.org/10040051

Abstract:

The hamiltonian in the Lanczos basis

Rewrite the state generation formula

$$H|f_m\rangle = |f_{m+1}\rangle + a_m|f_m\rangle + b_{m-1}|f_{m-1}\rangle$$

Because of the orthogonality, the only non-0 matrix elements are

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

But the f-states or not normalized. The normalized states are:

$$|\phi_m\rangle = \frac{1}{\sqrt{N_m}}|f_m\rangle$$

In this basis the H-matrix is

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

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

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

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

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_{m+1}

Potential problem:

The normalization constants N_m can become very large (think of E_0^m)

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

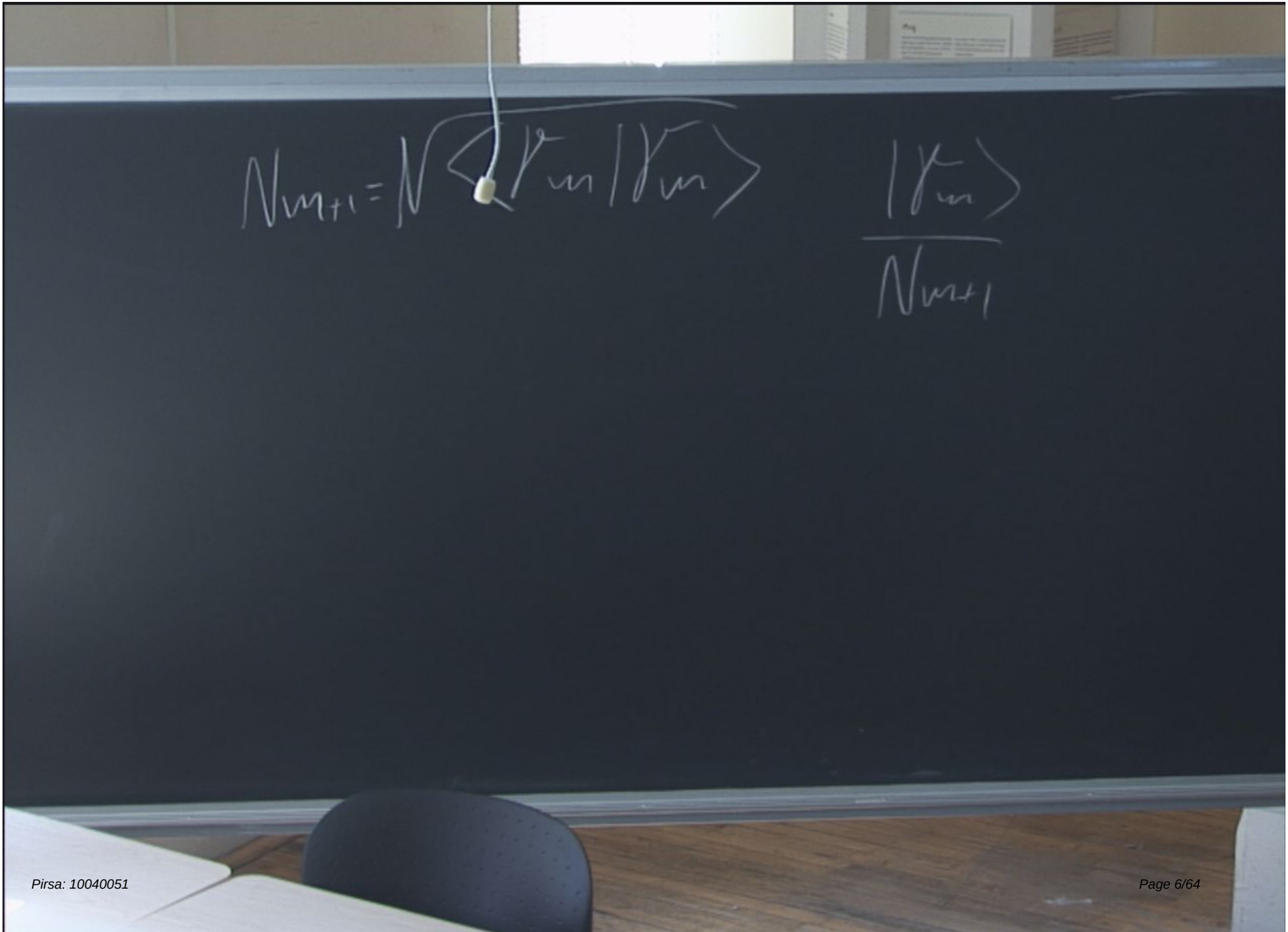
$$|\phi_1\rangle = \frac{1}{N_1}\big(H|\phi_0\rangle - a_0|\phi_0\rangle\big).$$

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

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_{m+1}


$N_{m+1} = N\sqrt{\langle \psi_m | \psi_m \rangle}$
$\frac{|\psi_m\rangle}{N_{m+1}}$

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

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

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

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_{m+1}

The H-matrix is

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

Lanczos basis generation in practice

Here: generate the orthogonal basis $\{\phi_m\}$ directly

$$|\phi_m\rangle = \sum_{a=1}^{M} \phi_m(a)|a\rangle, \quad m = 0, \ldots, \Lambda$$

in a given symmetry block of size M

The coefficients $\phi_m(a)$ are stored as $\Lambda+1$ vectors of size M

- may store only the vectors ϕ_{m-1} and ϕ_m to generate ϕ_{m+1}
  - but basis has to be re-generated when computing expectation values
- stabilization by "re-orthogonalization" (later) requires storage of all ϕ_m

Constructing the Lanczos basis

First: construct **orthogonal but not normalized basis {f_m}.** Define

$$N_m = \langle f_m | f_m \rangle, \quad H_{mm} = \langle f_m | H | f_m \rangle$$

Constructing the Lanczos basis

First: construct **orthogonal but not normalized basis {f_m}.** Define

$$N_m = \langle f_m | f_m \rangle, \quad H_{mm} = \langle f_m | H | f_m \rangle$$

The first state $|f_0\rangle$ is arbitrary, e.g., random. The next one is

$$|f_1\rangle = H|f_0\rangle - a_0|f_0\rangle$$

Demand orthogonality

$$\langle f_1 | f_0 \rangle = \langle f_0 | H | f_0 \rangle - a_0 \langle f_0 | f_0 \rangle = H_{00} - a_0 N_0 \quad \rightarrow \quad a_0 = H_{00}/N_0$$

The next state and its overlaps with the previous states

$$|f_2\rangle = H|f_1\rangle - a_1|f_1\rangle - b_0|f_0\rangle$$

$$\langle f_2 | f_1 \rangle = H_{11} - a_1 N_1, \quad \langle f_2 | f_0 \rangle = N_1 - b_0 N_0$$

For orthogonal states

$$a_1 = H_{11}/N_1, \quad b_0 = N_1/N_0$$

The hamiltonian in the Lanczos basis

Rewrite the state generation formula

$$H|f_m\rangle = |f_{m+1}\rangle + a_m|f_m\rangle + b_{m-1}|f_{m-1}\rangle$$

The hamiltonian in the Lanczos basis

Rewrite the state generation formula

$$H|f_m\rangle = |f_{m+1}\rangle + a_m|f_m\rangle + b_{m-1}|f_{m-1}\rangle$$

Because of the orthogonality, the only non-0 matrix elements are

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

But the f-states or not normalized. The normalized states are:

$$|\phi_m\rangle = \frac{1}{\sqrt{N_m}}|f_m\rangle$$

In this basis the H-matrix is

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

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

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

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

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_{m+1}

Lanczos basis generation in practice

Here: generate the orthogonal basis $\{\phi_m\}$ directly

$$|\phi_m\rangle = \sum_{a=1}^{M} \phi_m(a)|a\rangle, \quad m = 0, \ldots, \Lambda$$

in a given symmetry block of size M

The coefficients $\phi_m(a)$ are stored as $\Lambda+1$ vectors of size M

- may store only the vectors ϕ_{m-1} and ϕ_m to generate ϕ_{m+1}
  - but basis has to be re-generated when computing expectation values
- stabilization by "re-orthogonalization" (later) requires storage of all ϕ_m

Lanczos basis generation in practice

Here: generate the orthogonal basis $\{\phi_m\}$ directly

$$|\phi_m\rangle = \sum_{a=1}^{M} \phi_m(a)|a\rangle, \quad m = 0, \ldots, \Lambda$$

in a given symmetry block of size M

The coefficients $\phi_m(a)$ are stored as $\Lambda+1$ vectors of size M

- may store only the vectors ϕ_{m-1} and ϕ_m to generate ϕ_{m+1}
  - but basis has to be re-generated when computing expectation values
- stabilization by "re-orthogonalization" (later) requires storage of all ϕ_m

The main computational effort is in acting with the hamiltonian; $H|\phi_m\rangle$

- implement as a subroutine **hoperation**(ϕ,γ), where $|\gamma\rangle = H|\phi\rangle$
- state normalization implemented as **normalize**(ϕ,n)
  - ϕ = vector to normalize, n = $\langle\phi|\phi\rangle$ before normalization

Pseudocode; Lanczos basis generation

Initial random state

```
do i = 1, M
    φ_0(i) = random[0 − 1]
enddo
call normalize(φ_0, n_0)
```

Pseudocode; Lanczos basis generation

Initial random state

```
do i = 1, M
    φ0(i) = random[0 − 1]
enddo
call normalize(φ0, n0)
```

second state

```
call hoperation(φ0, φ1)
a0 = ⟨φ0|φ1⟩; φ1 = φ0 − a0|φ1⟩
call normalize(φ1, n1)
```

Pseudocode; Lanczos basis generation

Initial random state

```
do i = 1, M
    φ_0(i) = random[0 − 1]
enddo
call normalize(φ_0, n_0)
```

second state

```
call hoperation(φ_0, φ_1)
a_0 = ⟨φ_0|φ_1⟩; φ_1 = φ_0 − a_0|φ_1⟩
call normalize(φ_1, n_1)
```

Generate the rest of the states

```
do m = 1, Λ − 1
    call hoperation(φ_m, φ_{m+1})
    a_m = ⟨φ_m|φ_{m+1}⟩
    φ_{m+1} = φ_{m+1} − a_m φ_m − n_m φ_{m−1}
    call normalize(φ_{m+1}, n_{m+1})
enddo
```

Pseudocode; Lanczos basis generation

Initial random state

```
do i = 1, M
    φ_0(i) = random[0 − 1]
enddo
call normalize(φ_0, n_0)
```

second state

```
call hoperation(φ_0, φ_1)
a_0 = ⟨φ_0|φ_1⟩; φ_1 = φ_0 − a_0|φ_1⟩
call normalize(φ_1, n_1)
```

Generate the rest of the states

```
do m = 1, Λ − 1
    call hoperation(φ_m, φ_{m+1})
    a_m = ⟨φ_m|φ_{m+1}⟩
    φ_{m+1} = φ_{m+1} − a_m φ_m − n_m φ_{m−1}
    call normalize(φ_{m+1}, n_{m+1})
enddo
```

Note: the H-matrix can be constructed and diagonalized after each step

- follow evolution of energy versus Λ
- stop based on some convergence criterion on E_0 (or higher energy)

The subroutine **hoperation**(ϕ,γ) implements

$$H|\phi\rangle = |\gamma\rangle = \sum_{a=1}^{M}\sum_{b=1}^{M}\phi(a)\langle b|H|a\rangle|b\rangle \qquad |\phi\rangle = \sum_{a=1}^{M}\phi(a)|a\rangle$$

in a given symmetry block (M = block size)

We do not want to store *H* as an *M*×*M* matrix (too big). Two options:

- carry out the operations on the fly; only the vectors are stored
- store *H* in a compact form; only non-0 elements (sparse matrix)

Storing *H* speeds up the Lanczos iterations

- but may require a lot of memory

The subroutine **hoperation**(ϕ,γ) implements

$$H|\phi\rangle = |\gamma\rangle = \sum_{a=1}^{M}\sum_{b=1}^{M}\phi(a)\langle b|H|a\rangle|b\rangle \qquad |\phi\rangle = \sum_{a=1}^{M}\phi(a)|a\rangle$$

in a given symmetry block (M = block size)

We do not want to store *H* as an *M×M* matrix (too big). Two options:

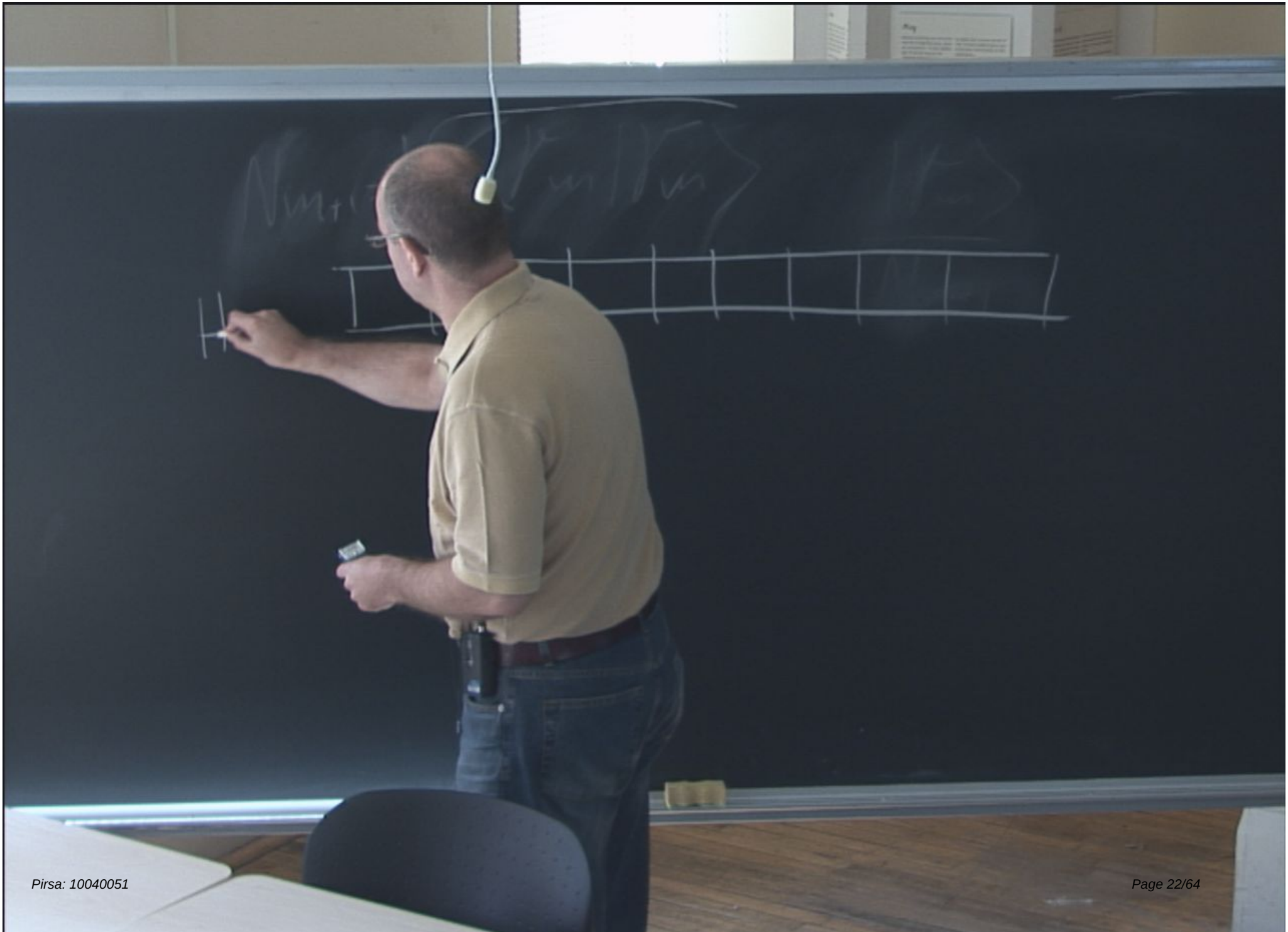
- carry out the operations on the fly; only the vectors are stored
- store *H* in a compact form; only non-0 elements (sparse matrix)

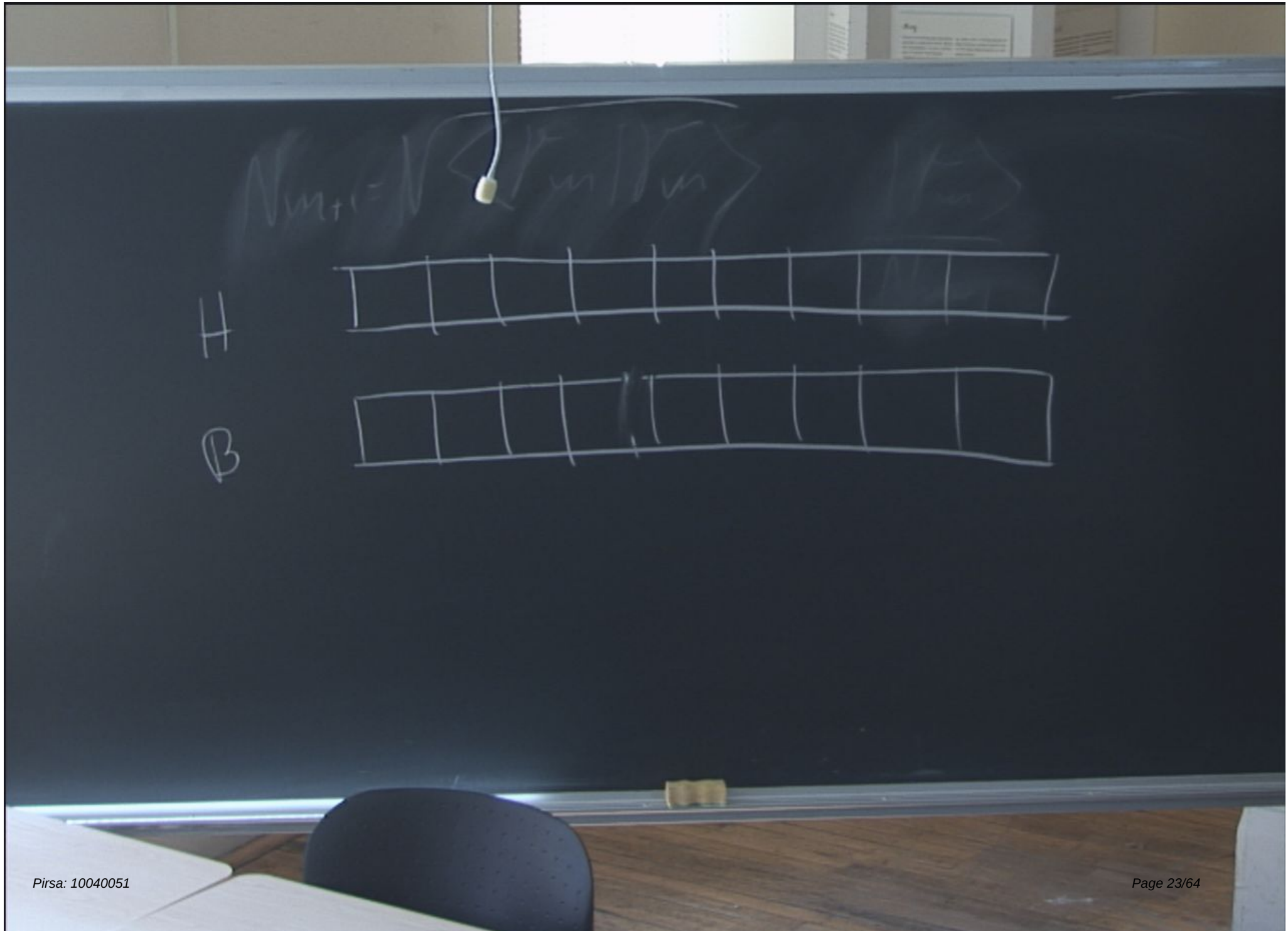
Storing *H* speeds up the Lanczos iterations

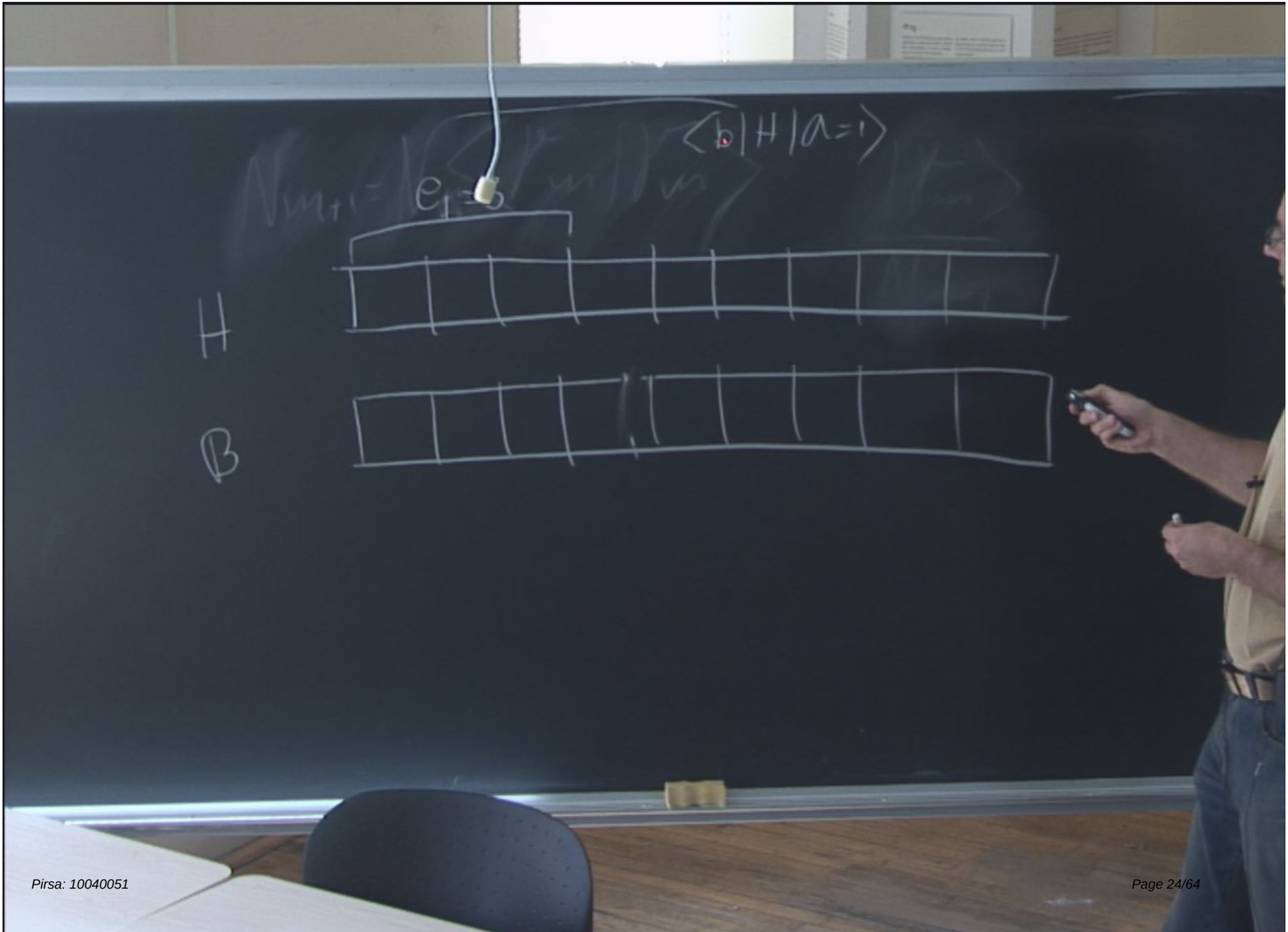
- but may require a lot of memory

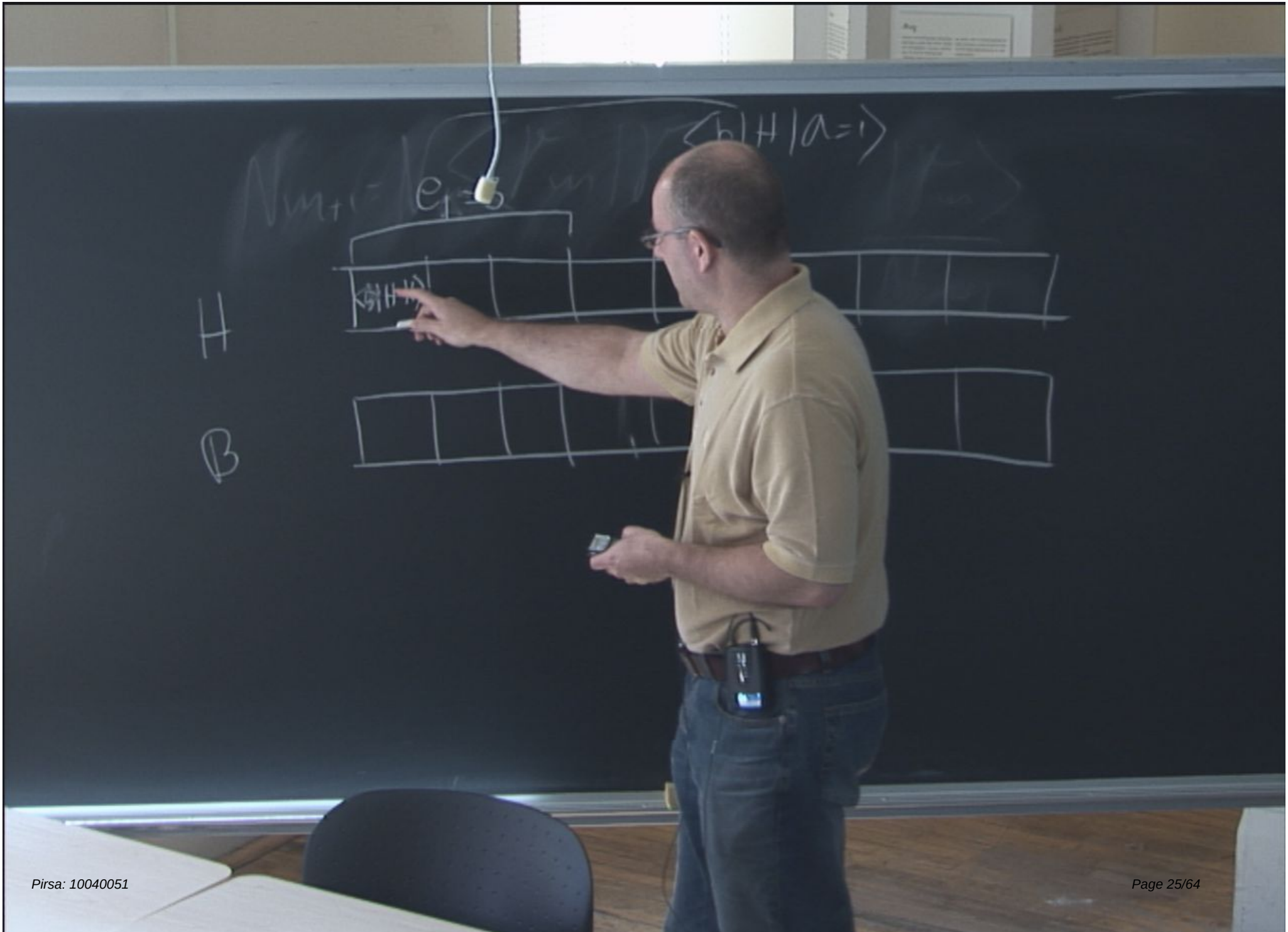
Compact storage of H: For each ***a=1,M***

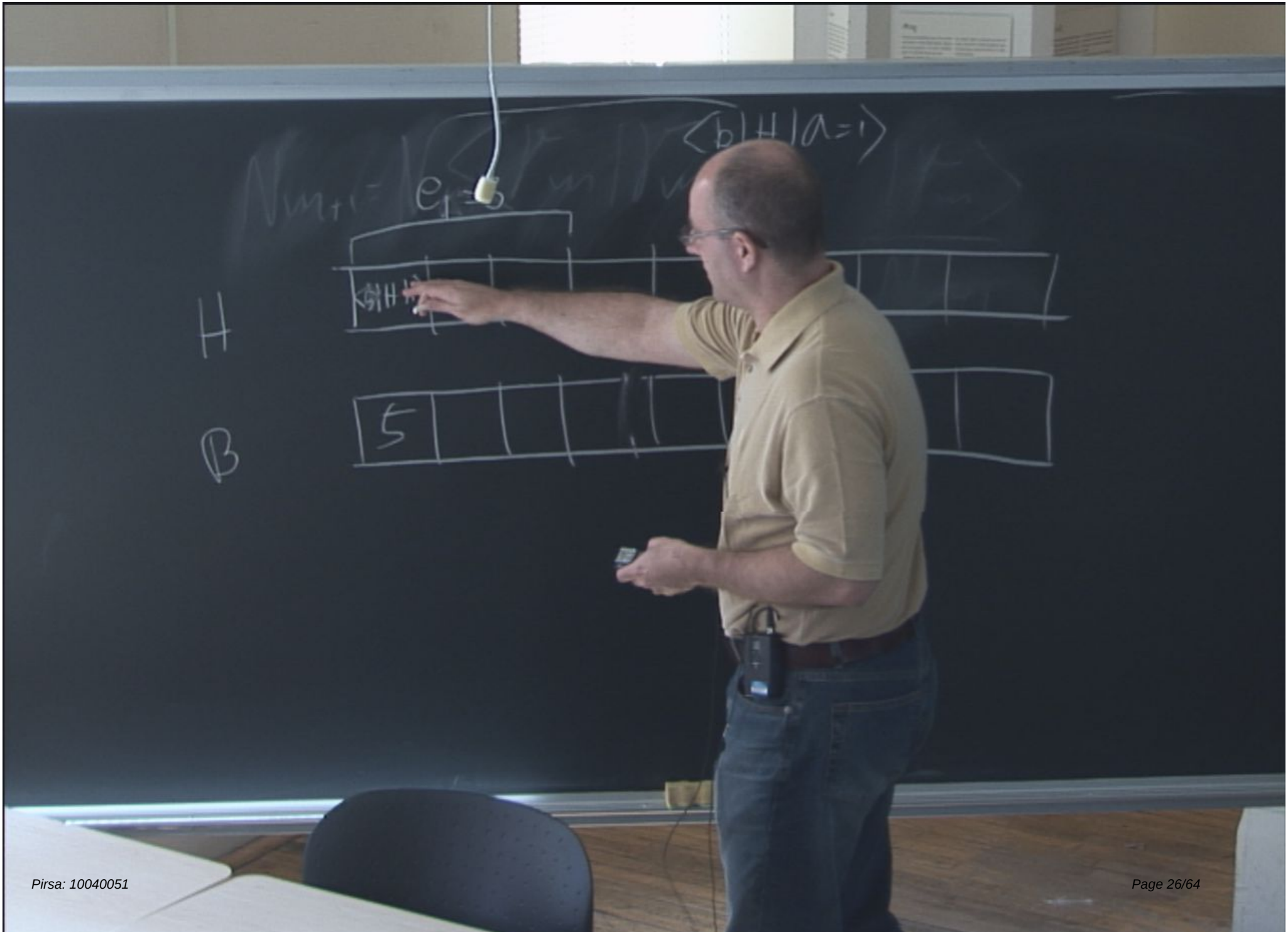
- e_a is the number of non-0 elements $\langle b|H|a\rangle$
- labels $i=s_a+1, s_a+e_a$ will refer to these matrix elements; $s_a = \sum_{c=1}^{a-1} e_a$
- ***H***(i) contains the values of the matrix elements $\langle b|H|a\rangle$
- ***B***(i) contains the corresponding "target" state index ***b***
- The Hamiltonian is symmetric

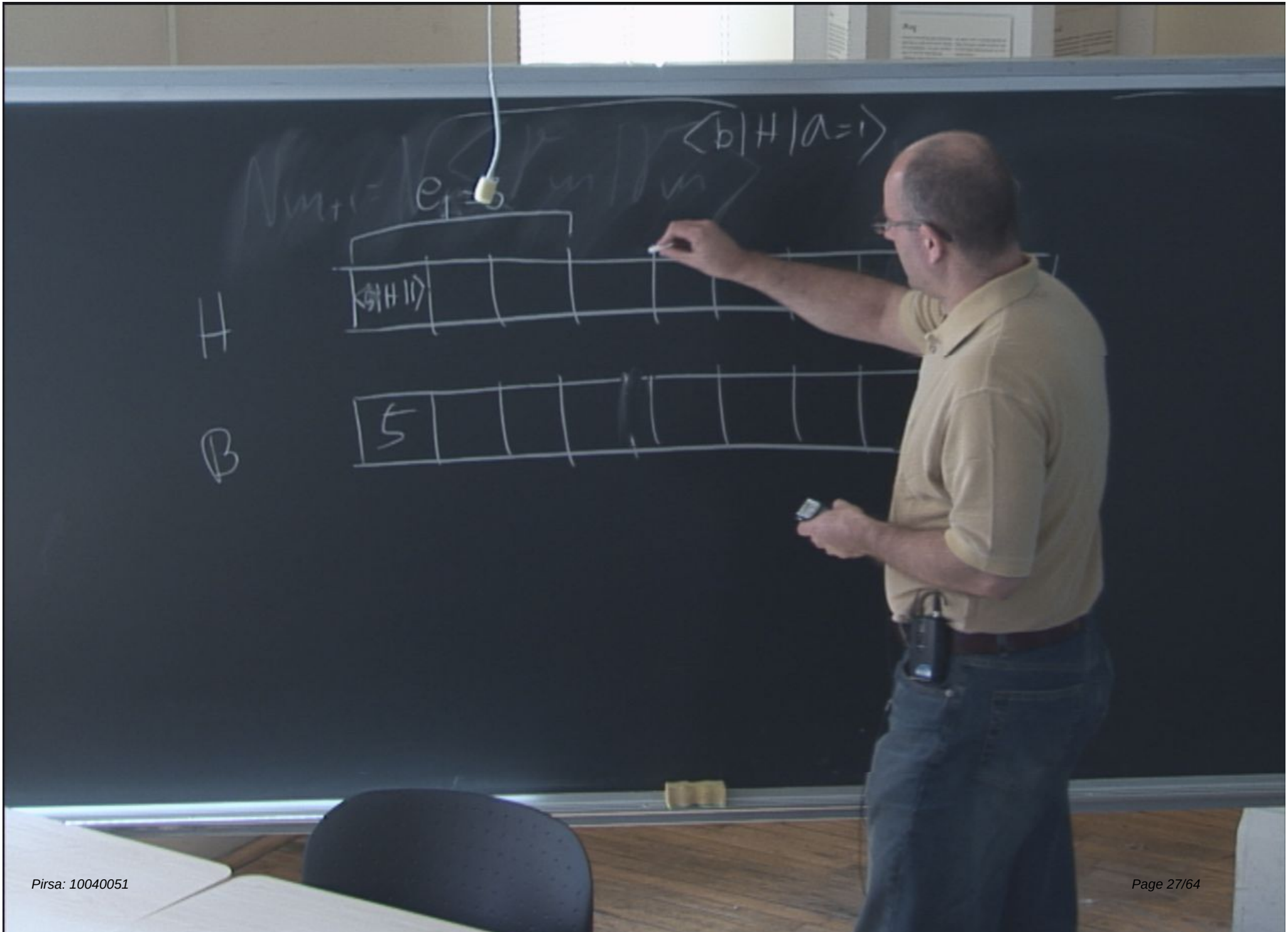


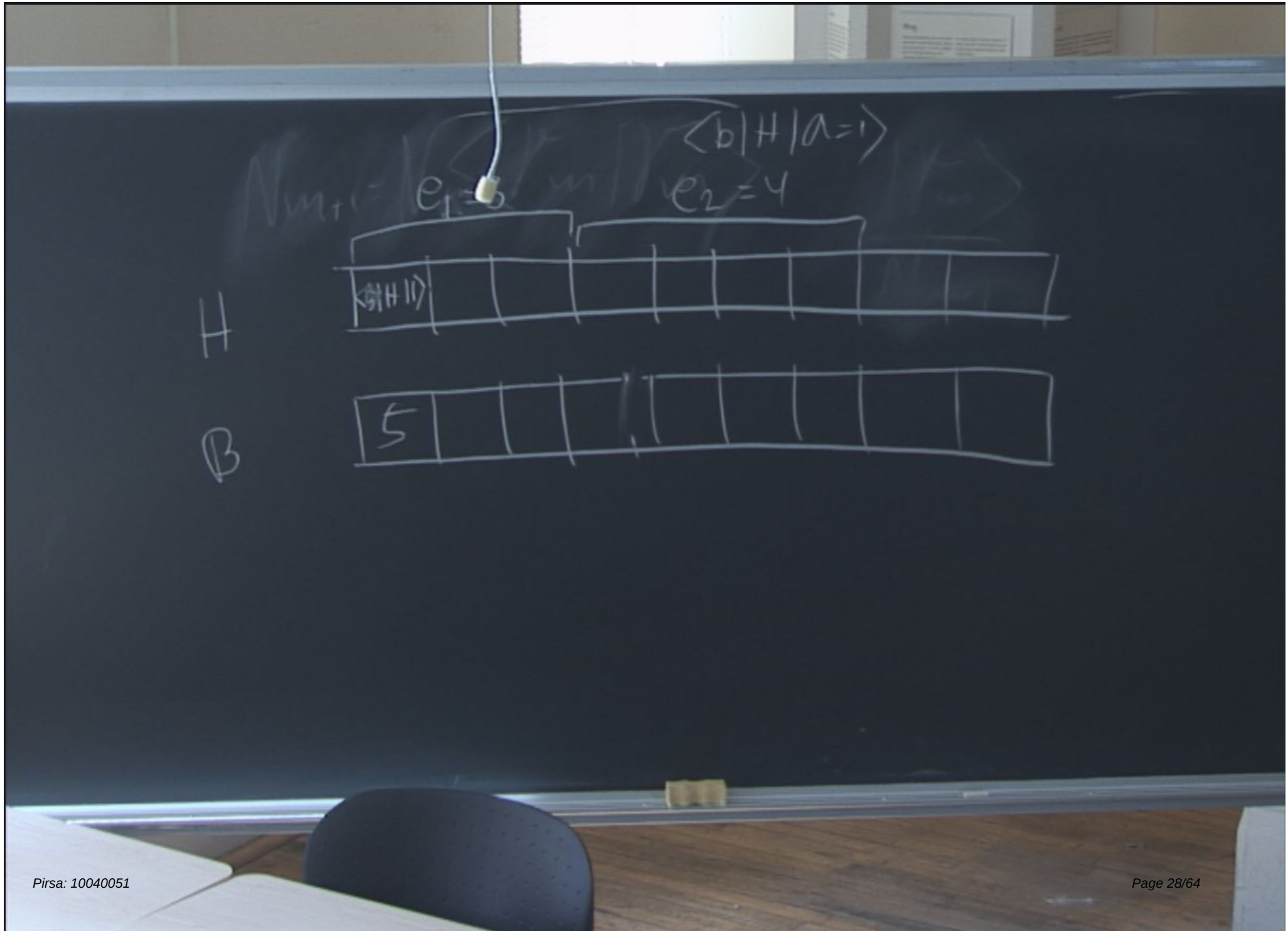

H
B

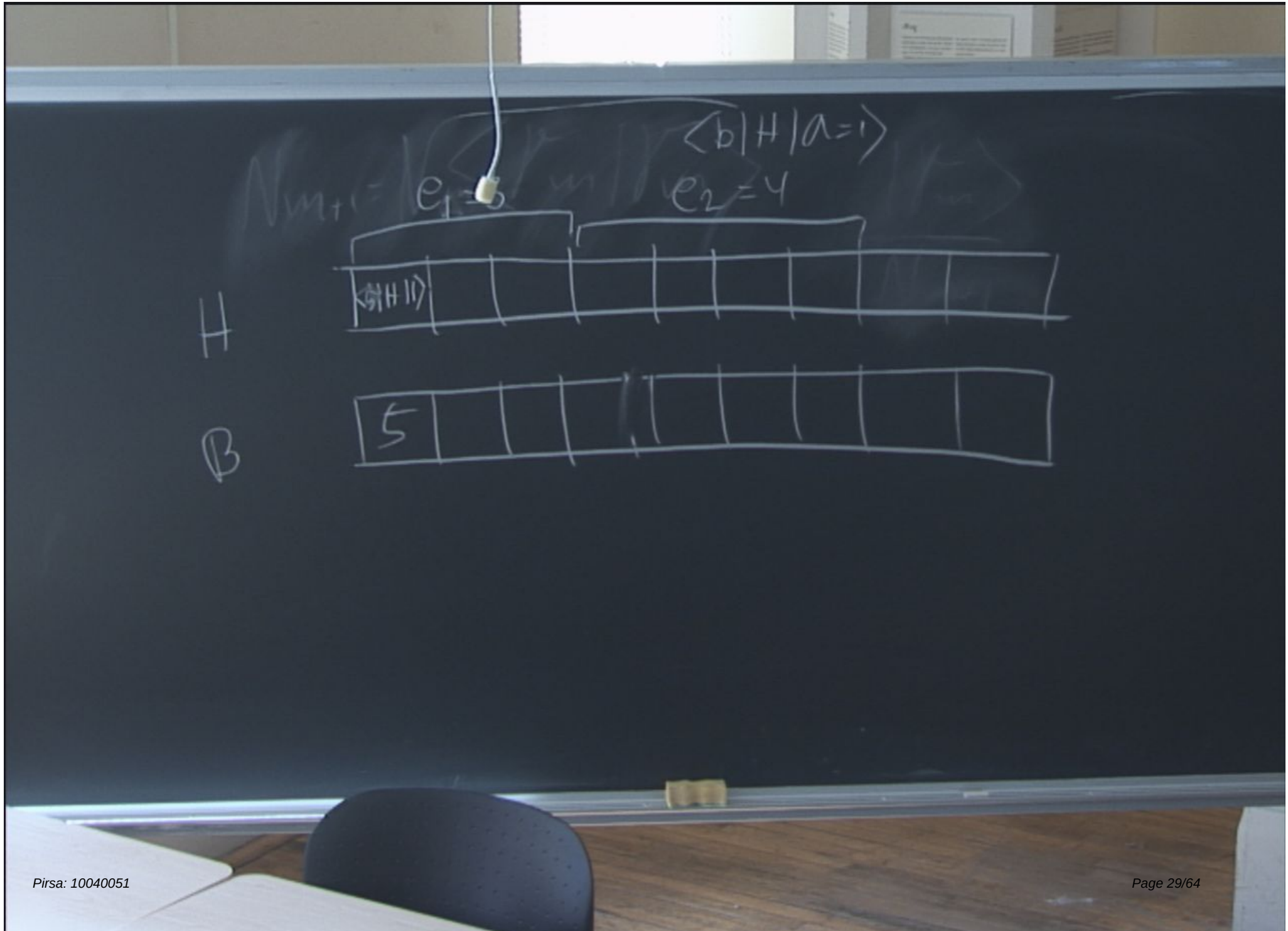

H
B


H
B


H
B


H
B


$\langle b|H|A=1\rangle$
$e_1=5$
$e_2=4$
H
B
5


$\langle b|H|A=1\rangle$
$e_1=5$
$e_2=4$
H
B
5

Construct all the matrix elements

```
do a = 1, M
    do i = 0, N − 1
        j = mod(i + 1, N)
        if (s_a[i] = s_a[j]) then
            H(a, a) = H(a, a) + 1/4
        else
            H(a, a) = H(a, a) − 1/4
            s = flip(s_a, i, j)
            call representative(s, r, l)
            call findstate(r, b)
            if (b ≥ 0) then
                H(a, b) = H(a, b) + (1/2) √(R_a/R_b) e^{i2πkl/N}
            endif
        endif
    enddo
enddo
```

Normalization of semi-momentum states

$$N_a = \left(\frac{N}{R_a}\right)^2 \sum_{r=1}^{R_a} [C_k^\sigma(r)]^2 = \frac{N^2}{2R_a}$$

Hamiltonian: ac with H

$$H|a^\pm(k)\rangle = \sum_{j=0}^{N} h_a^j \sqrt{\frac{R_a}{R_{b_j}}} \left(C_k^+(l_j)|b_j^\pm(k)\rangle \mp C_k^-(l_j)|b_j^\mp(k)\rangle \right),$$

The matrix elements are

$$\langle b^\tau(k)|H_j|a^\sigma(k)\rangle = \tau^{(\sigma-\tau)/2} h_a^j \sqrt{\frac{N_{b_j}}{N_a}} C_k^{\sigma\tau}(l_j)$$

σ is mot a conserved quantum number

- H and T mix σ=+1 and σ=–1 states
- the H matrix is twice as large as for momentum states

Hamiltonian : Act with an operator H_j on a representative state:

$$H_j|a\rangle = h_a^j P^{q_j} T^{-l_j} |b_j\rangle$$

We can write H acting on a basis state as

$$H|a^\sigma(k,p)\rangle = \sum_{j=0}^{N} \frac{h_a^j (\sigma p)^{q_j}}{\sqrt{N_a^\sigma}} \sum_{r=0}^{N-1} C_k^\sigma (r + l_j)(1 + pP)T^r |b_j\rangle$$

Using the properties (trigonometry) of the C-functions:

$$H|a^\sigma(k,p)\rangle = \sum_{j=0}^{N} h_a^j (\sigma p)^{q_j} \sqrt{\frac{N_{b_j}^\sigma}{N_a^\sigma}} \times$$

$$\left(\cos(kl_j)|b_j^\sigma(k,p)\rangle - \sigma \sqrt{\frac{N_{b_j}^{-\sigma}}{N_{b_j}^\sigma}} \sin(kl_j)|b^{-\sigma}(k,p)\rangle\right)$$

If, for some m, $T^m P|b_j\rangle = |b_j\rangle$ then

$$\sqrt{\frac{N_{b_j}^{-\sigma}}{N_{b_j}^\sigma}} = \sqrt{\frac{1 - \sigma p \cos(km)}{1 + \sigma p \cos(km)}} = \frac{|\sin(km)|}{1 + \sigma p \cos(km)}$$

$$\langle b^\mp(k,p)|b^\pm(k,p)\rangle = -p$$

Using spin-inversion symmetry

Spin inversion operator: $Z|S_1^z, S_2^z, \ldots, S_N^z\rangle = |-S_1^z, -S_2^z, \ldots, -S_N^z\rangle$

In the magnetization block $m^z=0$ we can use eigenstates of Z

$$|a^\sigma(k,p,z)\rangle = \frac{1}{\sqrt{N_a^\sigma}} \sum_{r=0}^{N-1} C_k^\sigma(r) T^r (1+pP)(1+zZ)|a\rangle$$

$$Z|a^\sigma(k,p,z)\rangle = z|a^\sigma(k,p,z)\rangle, \quad z = \pm 1$$

Normalization: must check how a representative transforms under Z,P,T

$$\begin{array}{lllll}
1) & T^m P|a\rangle \neq |a\rangle, & T^m Z|a\rangle \neq |a\rangle & & T^m PZ|a\rangle \neq |a\rangle \\
2) & T^m P|a\rangle = |a\rangle, & T^m Z|a\rangle \neq |a\rangle & & T^m PZ|a\rangle \neq |a\rangle \\
3) & T^m P|a\rangle \neq |a\rangle, & T^m Z|a\rangle = |a\rangle & & T^m PZ|a\rangle \neq |a\rangle \\
4) & T^m P|a\rangle \neq |a\rangle, & T^m Z|a\rangle \neq |a\rangle & & T^m PZ|a\rangle = |a\rangle \\
5) & T^m P|a\rangle = |a\rangle, & T^n Z|a\rangle = |a\rangle & \Rightarrow & T^{m+n} PZ|a\rangle = |a\rangle
\end{array}$$

For cases 2,4,5 only σ=+1 or σ=−1 included

$$N_a^\sigma = \frac{2N^2}{R_a} \times \begin{cases} 1, & 1) \\ 1+\sigma p \cos(km), & 2) \\ 1+z\cos(km), & 3) \\ 1+\sigma p z \cos(km), & 4) \end{cases}$$

The Lanczos method

If we need only the ground state and a small number of excitations

- can use "Krylov space" methods, which work for much larger matrices
- basis states with 10^7 states or more can be easily handled (30-40 spins)

Constructing the Lanczos basis

First: construct **orthogonal but not normalized basis $\{f_m\}$.** Define

$$N_m = \langle f_m | f_m \rangle, \quad H_{mm} = \langle f_m | H | f_m \rangle$$

The first state $|f_0\rangle$ is arbitrary, e.g., random. The next one is

$$|f_1\rangle = H|f_0\rangle - a_0|f_0\rangle$$

Demand orthogonality

$$\langle f_1 | f_0 \rangle = \langle f_0 | H | f_0 \rangle - a_0 \langle f_0 | f_0 \rangle = H_{00} - a_0 N_0 \quad \rightarrow \quad a_0 = H_{00}/N_0$$

The next state and its overlaps with the previous states

$$|f_2\rangle = H|f_1\rangle - a_1|f_1\rangle - b_0|f_0\rangle$$

$$\langle f_2 | f_1 \rangle = H_{11} - a_1 N_1, \quad \langle f_2 | f_0 \rangle = N_1 - b_0 N_0$$

For orthogonal states

$$a_1 = H_{11}/N_1, \quad b_0 = N_1/N_0$$

All subsequent states are constructed according to

$$|f_{m+1}\rangle = H|f_m\rangle - a_m|f_m\rangle - b_{m-1}|f_{m-1}\rangle$$

Lanczos basis generation in practice

Here: generate the orthogonal basis $\{\phi_m\}$ directly

$$|\phi_m\rangle = \sum_{a=1}^{M} \phi_m(a)|a\rangle, \quad m = 0, \ldots, \Lambda$$

in a given symmetry block of size M

The coefficients $\phi_m(a)$ are stored as $\Lambda+1$ vectors of size M

- may store only the vectors ϕ_{m-1} and ϕ_m to generate ϕ_{m+1}
  - but basis has to be re-generated when computing expectation values
- stabilization by "re-orthogonalization" (later) requires storage of all ϕ_m

The main computational effort is in acting with the hamiltonian; $H|\phi_m\rangle$

- implement as a subroutine **hoperation**(ϕ,γ), where $|\gamma\rangle = H|\phi\rangle$
- state normalization implemented as **normalize**(ϕ,n)
  - ϕ = vector to normalize, n = $\langle\phi|\phi\rangle$ before normalization

The subroutine **hoperation**(ϕ,γ) implements

$$H|\phi\rangle = |\gamma\rangle = \sum_{a=1}^{M}\sum_{b=1}^{M}\phi(a)\langle b|H|a\rangle|b\rangle \qquad |\phi\rangle = \sum_{a=1}^{M}\phi(a)|a\rangle$$

in a given symmetry block (M = block size)

We do not want to store *H* as an *M*×*M* matrix (too big). Two options:

- carry out the operations on the fly; only the vectors are stored
- store *H* in a compact form; only non-0 elements (sparse matrix)

Storing *H* speeds up the Lanczos iterations

- but may require a lot of memory

The subroutine **hoperation**(ϕ,γ) implements

$$H|\phi\rangle = |\gamma\rangle = \sum_{a=1}^{M}\sum_{b=1}^{M}\phi(a)\langle b|H|a\rangle|b\rangle \qquad |\phi\rangle = \sum_{a=1}^{M}\phi(a)|a\rangle$$

in a given symmetry block (M = block size)

We do not want to store *H* as an *M×M* matrix (too big). Two options:

- carry out the operations on the fly; only the vectors are stored
- store *H* in a compact form; only non-0 elements (sparse matrix)

Storing *H* speeds up the Lanczos iterations

- but may require a lot of memory

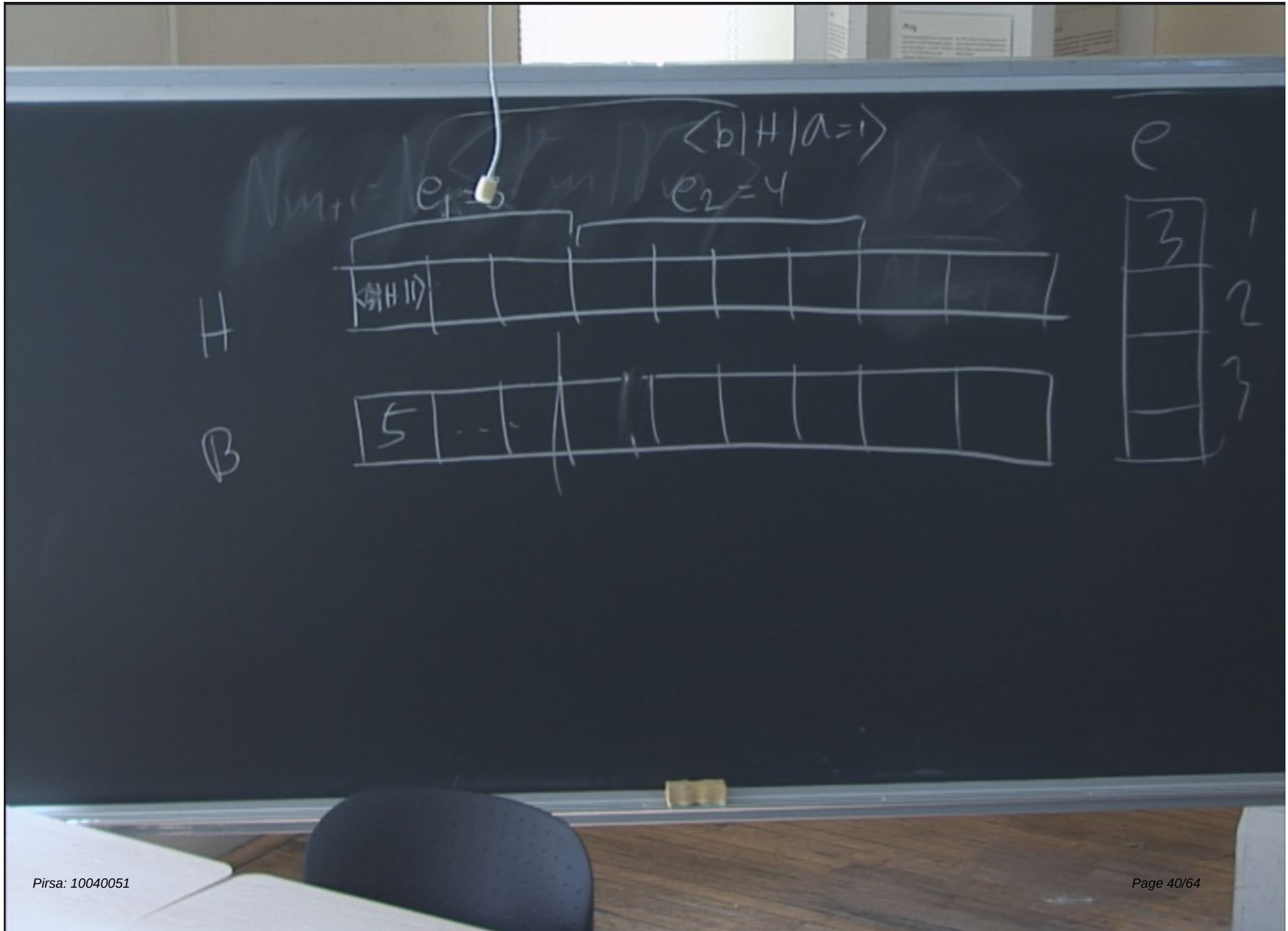
Compact storage of H: For each ***a=1,M***

- e_a is the number of non-0 elements $\langle b|H|a\rangle$
- labels $i=s_a+1, s_a+e_a$ will refer to these matrix elements; $s_a = \sum_{c=1}^{a-1} e_a$
- ***H***(i) contains the values of the matrix elements $\langle b|H|a\rangle$
- ***B***(i) contains the corresponding "target" state index ***b***
- The Hamiltonian is symmetric

Pseudocode; hamiltonian operation with compact storage

subroutine hoperation(ϕ, γ)
$\gamma = 0;\ i = 0$
do $a = 1, M$
  do $j = 1, e_a$
    $i = i + 1$
    $\gamma(B(i)) = \gamma(B(i)) + H(i)\phi(a)$
    $\gamma(a) = \gamma(a) + H(i)\phi(B(i))$
  enddo
enddo

$$H|\phi\rangle = |\gamma\rangle = \sum_{a=1}^{M}\sum_{b=1}^{M} \phi(a)\langle b|H|a\rangle|b\rangle$$


$\langle b|H|a=1\rangle$
$e_1=6$
$e_2=4$
H
B
5
e
3
1
2
3

Pseudocode; hamiltonian operation with compact storage

```
subroutine hoperation(φ, γ)
γ = 0; i = 0
do a = 1, M
    do j = 1, e_a
        i = i + 1
        γ(B(i)) = γ(B(i)) + H(i)φ(a)
        γ(a) = γ(a) + H(i)φ(B(i))
    enddo
enddo
```

$$H|\phi\rangle = |\gamma\rangle = \sum_{a=1}^{M}\sum_{b=1}^{M} \phi(a)\langle b|H|a\rangle|b\rangle$$

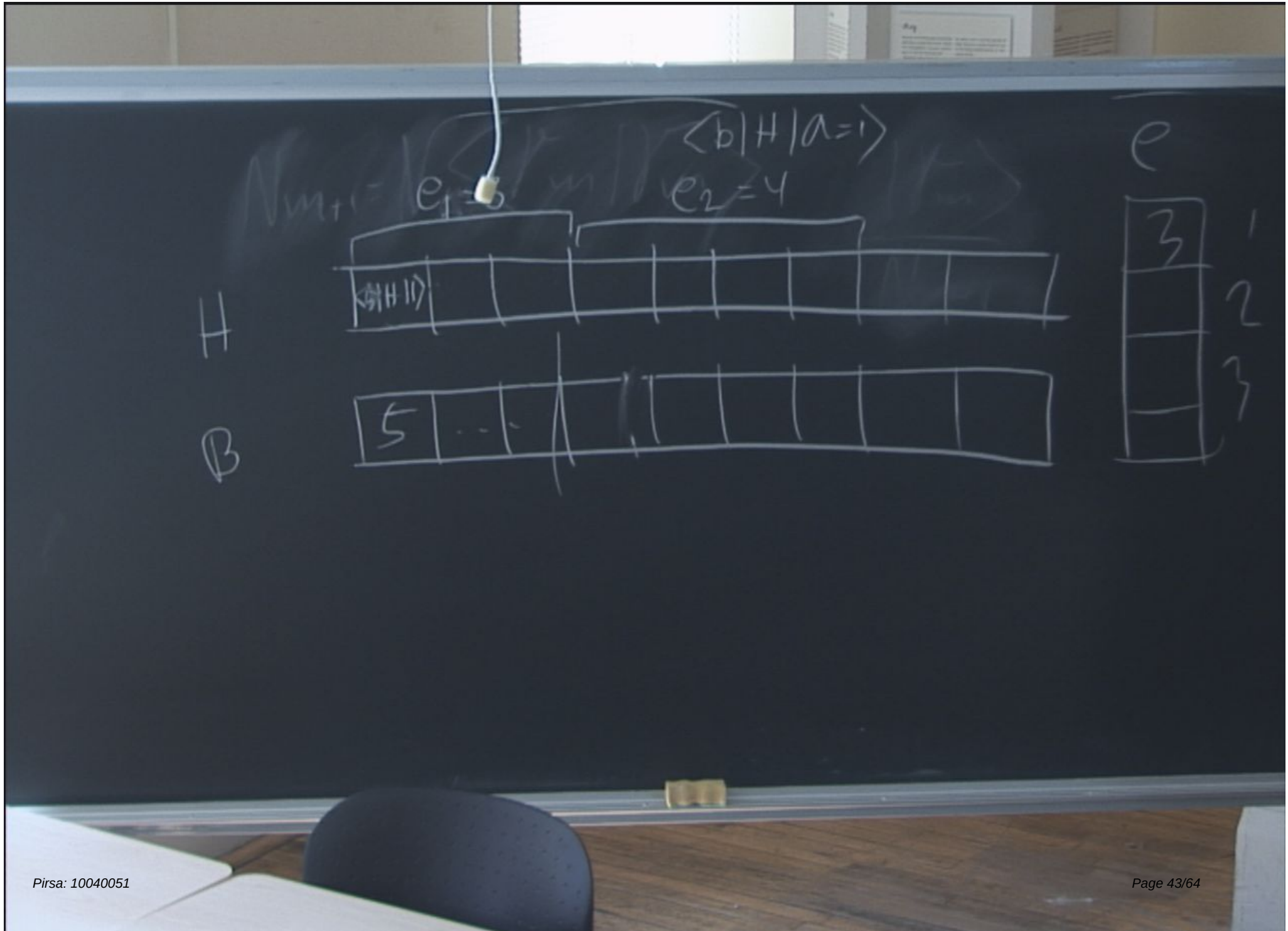
Pseudocode; hamiltonian operation with compact storage

```
subroutine hoperation(φ, γ)
γ = 0; i = 0
do a = 1, M
    do j = 1, e_a
        i = i + 1
        γ(B(i)) = γ(B(i)) + H(i)φ(a)
        γ(a) = γ(a) + H(i)φ(B(i))
    enddo
enddo
```

$$H|\phi\rangle = |\gamma\rangle = \sum_{a=1}^{M}\sum_{b=1}^{M} \phi(a)\langle b|H|a\rangle|b\rangle$$

Further storage compactification possible

- small number of different elements
- use mapping $\langle \mathbf{b}|\mathbf{H}|\mathbf{a}\rangle \rightarrow \text{integer}$
- many operations on |a> give same |b>


$\langle b|H|a=1\rangle$
$e_1=6$
$e_2=4$
H
B
5
e
3
1
2
3

Pseudocode; hamiltonian operation with compact storage

```
subroutine hoperation(φ, γ)
γ = 0; i = 0
do a = 1, M
    do j = 1, e_a
        i = i + 1
        γ(B(i)) = γ(B(i)) + H(i)φ(a)
        γ(a) = γ(a) + H(i)φ(B(i))
    enddo
enddo
```

$$H|\phi\rangle = |\gamma\rangle = \sum_{a=1}^{M}\sum_{b=1}^{M} \phi(a)\langle b|H|a\rangle|b\rangle$$

Further storage compactification possible

- small number of different elements
- use mapping $\langle \mathbf{b}|\mathbf{H}|\mathbf{a}\rangle \rightarrow \text{integer}$
- many operations on |a> give same |b>

Operator expectation values

Diagonalizing the tri-diagonal matrix → eigenstates in the Lanczos basis

- eigenvectors v_n, energies E_n
- only some number of low-energy states ($<< \Lambda$) are correct eigenstates of H

Operator expectation values

Diagonalizing the tri-diagonal matrix → eigenstates in the Lanczos basis

- eigenvectors v_n, energies E_n
- only some number of low-energy states ($<< \Lambda$) are correct eigenstates of H

To compute expectation values we normally go back to the original basis

$$\psi_n(a) = \sum_{m=0}^{\Lambda} v_n(m)\phi_m(a), \qquad a = 1, \ldots, M$$

Operator expectation values

Diagonalizing the tri-diagonal matrix → eigenstates in the Lanczos basis

- eigenvectors v_n, energies E_n
- only some number of low-energy states ($<< \Lambda$) are correct eigenstates of H

To compute expectation values we normally go back to the original basis

$$\psi_n(a) = \sum_{m=0}^{\Lambda} v_n(m)\phi_m(a), \qquad a = 1, \ldots, M$$

To compute $\langle \psi_\mathbf{n} | \mathbf{O} | \psi_\mathbf{n} \rangle$ first construct

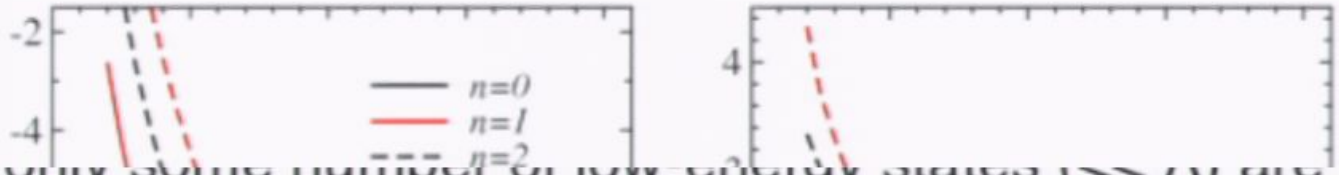
$$O|\psi_n\rangle = |\psi_n^O\rangle = \sum_{a=1}^{M} \psi_n(a) O|a\rangle$$

$$= \sum_{a=1}^{M}\sum_{b=1}^{M} \psi_n(a)|b\rangle\langle b|O|a\rangle$$

<bIOIa> done exactly as when constructing of the H matrix

$$= \sum_{b=1}^{M} \psi_n^O(b)|b\rangle \qquad \psi_n^O(b) = \sum_{a=1}^{M} \psi_n(a)\langle b|O|a\rangle$$

Convergence properties of the Lanczos method



Example; 24-site chain

$m_z = 0$, $k = 0$, $p = 1$, $z = 1$

block size M=28416

only some number of low energy states ($\ll N$) are correct eigenstates of H

Operator expectation values

Diagonalizing the tri-diagonal matrix → eigenstates in the Lanczos basis

- eigenvectors v_n, energies E_n
- only some number of low-energy states ($<< \Lambda$) are correct eigenstates of H

To compute expectation values we normally go back to the original basis

$$\psi_n(a) = \sum_{m=0}^{\Lambda} v_n(m)\phi_m(a), \qquad a = 1, \ldots, M$$

To compute $\langle \psi_\mathbf{n} | \mathbf{O} | \psi_\mathbf{n} \rangle$ first construct

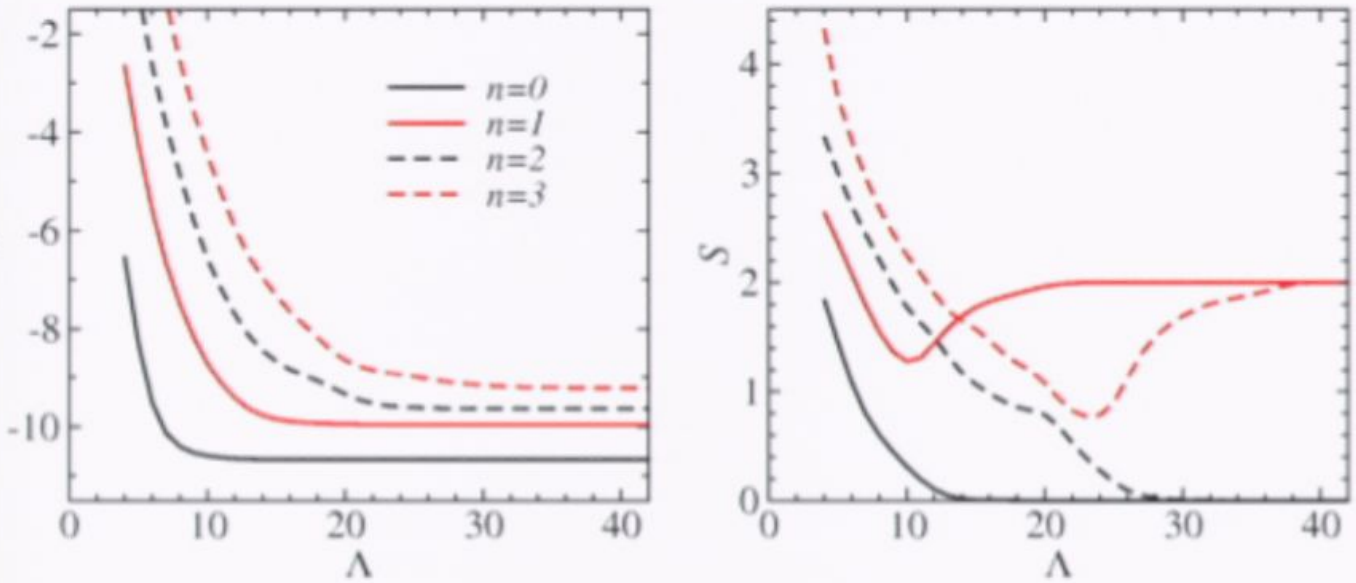
$$O|\psi_n\rangle = |\psi_n^O\rangle = \sum_{a=1}^{M} \psi_n(a) O|a\rangle$$

$$= \sum_{a=1}^{M}\sum_{b=1}^{M} \psi_n(a)|b\rangle\langle b|O|a\rangle$$

<bIOIa> done exactly as when constructing of the H matrix

$$= \sum_{b=1}^{M} \psi_n^O(b)|b\rangle \qquad \psi_n^O(b) = \sum_{a=1}^{M} \psi_n(a)\langle b|O|a\rangle$$

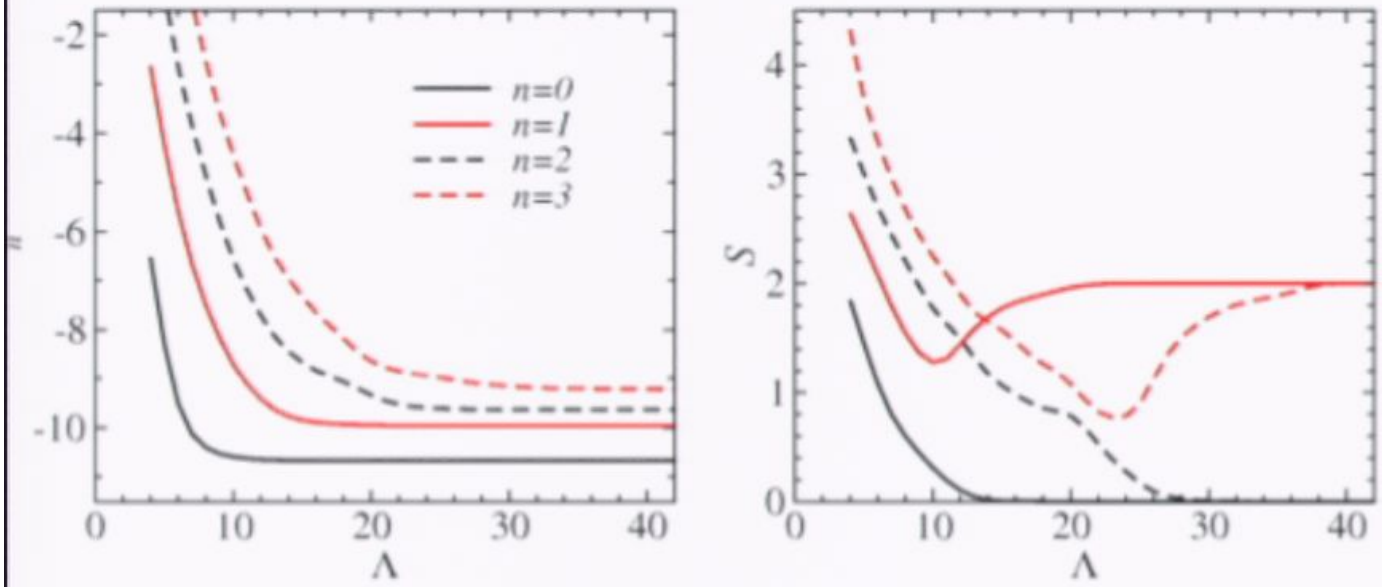
Convergence properties of the Lanczos method



Example; 24-site chain
$m_z = 0$, $k = 0$, $p = 1$, $z = 1$
block size M=28416

Ground state converges first, then successively excited states

Convergence properties of the Lanczos method



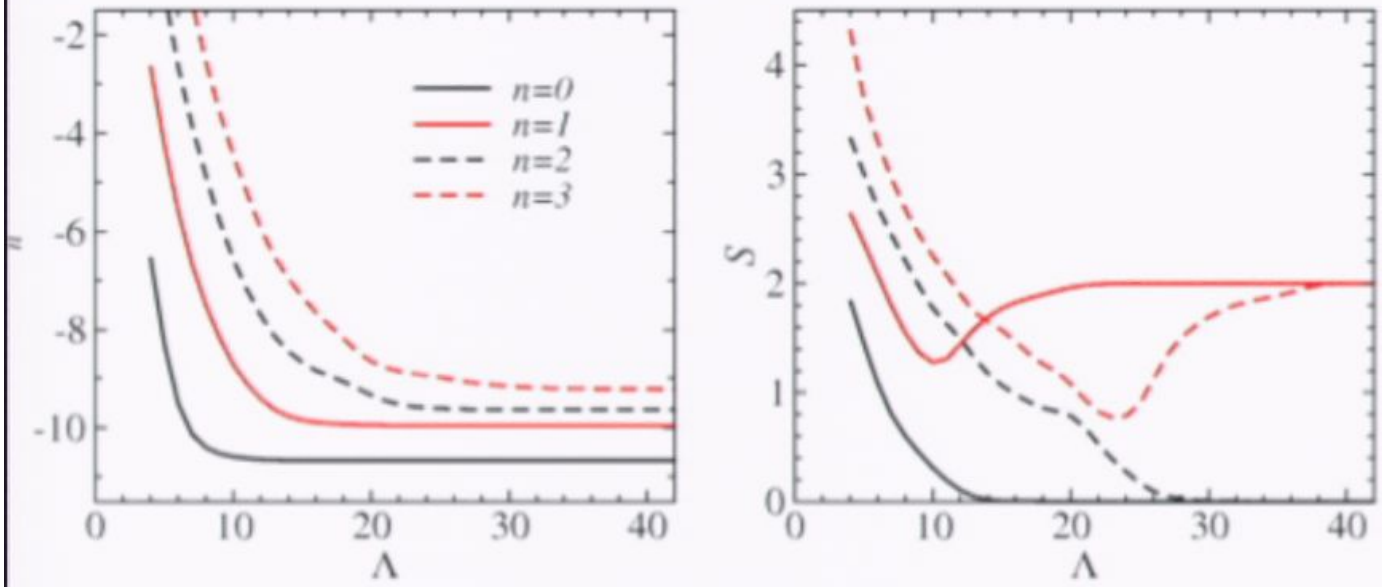
Example; 24-site chain
$m_z = 0$, $k = 0$, $p = 1$, $z = 1$
block size M=28416

Ground state converges first, then successively excited states

Loss of orthogonality: accumulation of numerical error → basis becomes non-orthogonal

- higher states collapse down onto lower ones
- can be cured with re-orthogonalization

Convergence properties of the Lanczos method

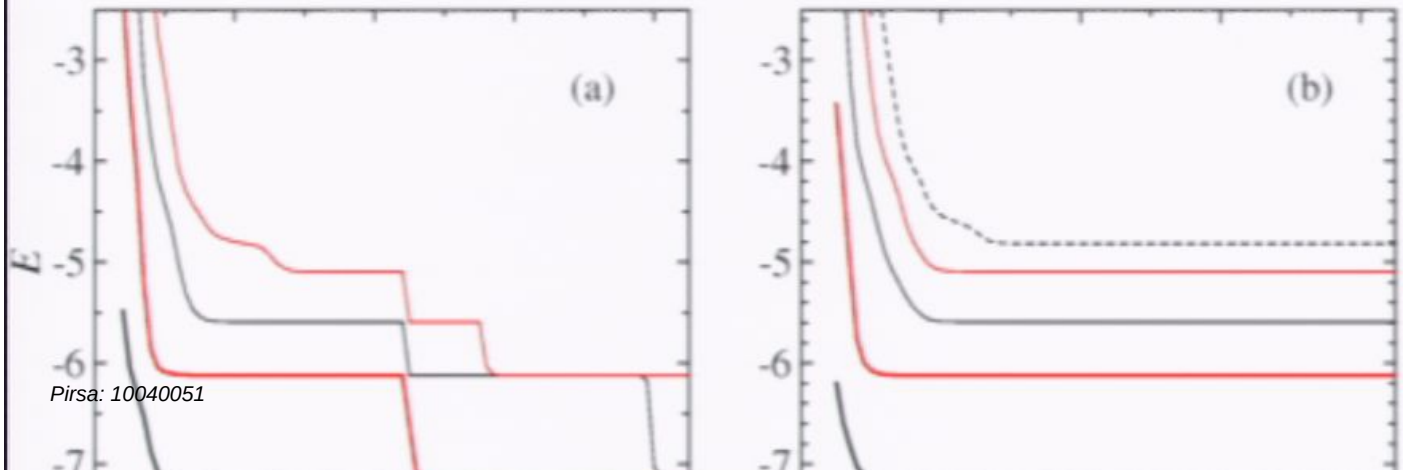


Example; 24-site chain
$m_z = 0$, $k = 0$, $p = 1$, $z = 1$
block size M=28416

Ground state converges first, then successively excited states

Loss of orthogonality: accumulation of numerical error → basis becomes non-orthogonal

- higher states collapse down onto lower ones
- can be cured with re-orthogonalization



Example; 16-site chain
$m_z = 0$, $k = 0$, $p = 1$, $z = 1$
block size M=212

- (a) non-orthogonality
- (b) re-orthogonalized

Re-orthogonalization procedure

For each state generated, remove all components of prior states, *i=1,...,m*

- easy if we work with the normalized basis and all states are stored

$$|\phi_m\rangle \rightarrow \frac{|\phi_m\rangle - q|\phi_i\rangle}{1-q^2}, \qquad q = \langle\phi_i|\phi_m\rangle$$

Re-orthogonalization procedure

For each state generated, remove all components of prior states, *i=1,...,m*

- easy if we work with the normalized basis and all states are stored

$$|\phi_m\rangle \rightarrow \frac{|\phi_m\rangle - q|\phi_i\rangle}{1-q^2}, \qquad q = \langle\phi_i|\phi_m\rangle$$

Pseudocode: modify state generation

```
do m = 1, Λ − 1
    call hoperation(φ_m, φ_{m+1})
    a_m = ⟨φ_m|φ_{m+1}⟩; φ_{m+1} = φ_{m+1} − a_m φ_m − n_m φ_{m−1}
    call normalize(φ_{m+1}, n_{m+1})
    do i = 1, m
        q = ⟨φ_{m+1}|φ_i⟩; φ_{m+1} = (φ_{m+1} − q φ_i)/(1 − q^2)
    enddo
enddo
```

Re-orthogonalization procedure

For each state generated, remove all components of prior states, *i=1,...,m*

- easy if we work with the normalized basis and all states are stored

$$|\phi_m\rangle \rightarrow \frac{|\phi_m\rangle - q|\phi_i\rangle}{1-q^2}, \qquad q = \langle\phi_i|\phi_m\rangle$$

Pseudocode: modify state generation

```
do m = 1, Λ − 1
    call hoperation(φ_m, φ_{m+1})
    a_m = ⟨φ_m|φ_{m+1}⟩; φ_{m+1} = φ_{m+1} − a_m φ_m − n_m φ_{m−1}
    call normalize(φ_{m+1}, n_{m+1})
    do i = 1, m
        q = ⟨φ_{m+1}|φ_i⟩; φ_{m+1} = (φ_{m+1} − q φ_i)/(1 − q^2)
    enddo
enddo
```

Note: the Lanczos method can only generate a single state of a multiplet

- some random linear combination of degenerate states

Example: 2 degenerate states i, j:

$$H^{\Lambda}|\Psi\rangle = \sum c_m E^{\Lambda}_m |\psi_m\rangle + E^m_{i,j}(c_i|\psi_i\rangle + c_j|\psi_j\rangle)$$

Spin correlations in the Heisenberg chain

Let's look at the (staggered) spin correlation function

$$C(r) = \langle \mathbf{S}_i \cdot \mathbf{S}_{i+r} \rangle (-1)^r$$

versus the distance r and at r=N/2 versus system size N

Theory (bosonization, conformal field theory) predicts (for large r, N)

$$C(r) \propto \frac{\ln^{1/2}(r/r_0)}{r}$$

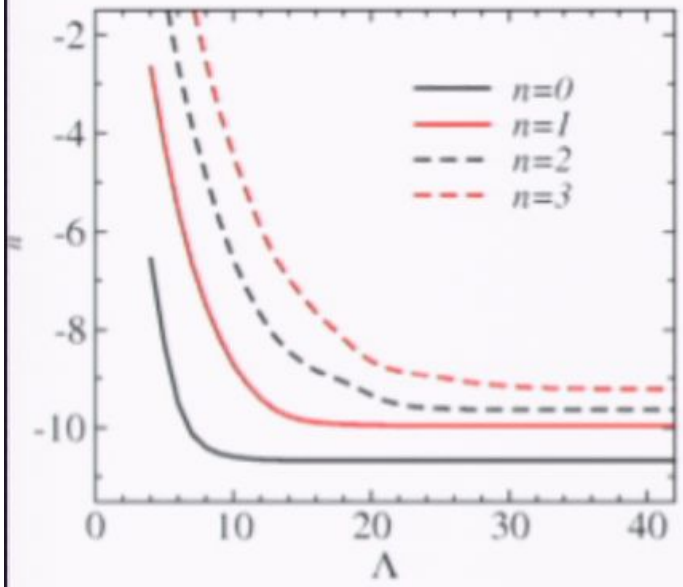
Re-orthogonalization procedure

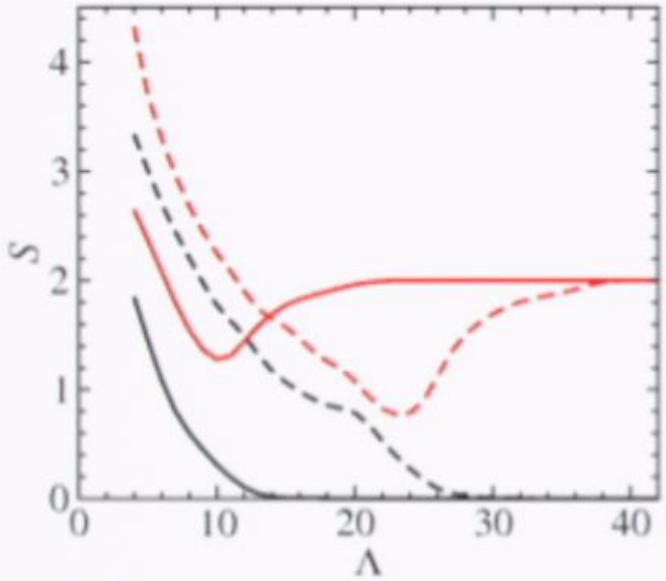
For each state generated, remove all components of prior states, *i=1,...,m*

- easy if we work with the normalized basis and all states are stored

$$|\phi_m\rangle \rightarrow \frac{|\phi_m\rangle - q|\phi_i\rangle}{1-q^2}, \qquad q = \langle\phi_i|\phi_m\rangle$$

Convergence properties of the Lanczos method



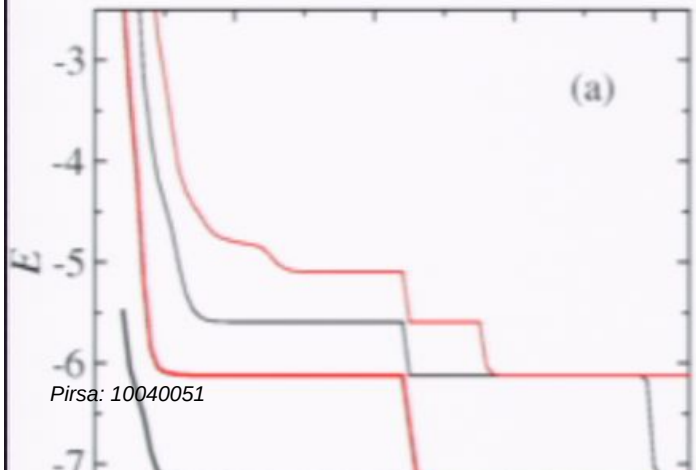


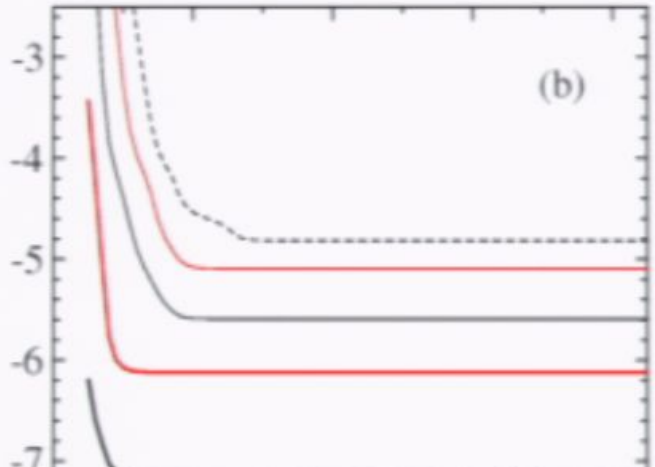
Example; 24-site chain
$m_z = 0$, $k = 0$, $p = 1$, $z = 1$
block size M=28416

Ground state converges first, then successively excited states

Loss of orthogonality: accumulation of numerical error → basis becomes non-orthogonal

- higher states collapse down onto lower ones
- can be cured with re-orthogonalization





Example; 16-site chain
$m_z = 0$, $k = 0$, $p = 1$, $z = 1$
block size M=212

- (a) non-orthogonality
- (b) re-orthogonalized

Re-orthogonalization procedure

For each state generated, remove all components of prior states, *i=1,...,m*

- easy if we work with the normalized basis and all states are stored

$$|\phi_m\rangle \rightarrow \frac{|\phi_m\rangle - q|\phi_i\rangle}{1-q^2}, \qquad q = \langle\phi_i|\phi_m\rangle$$

Re-orthogonalization procedure

For each state generated, remove all components of prior states, *i=1,...,m*

- easy if we work with the normalized basis and all states are stored

$$|\phi_m\rangle \rightarrow \frac{|\phi_m\rangle - q|\phi_i\rangle}{1-q^2}, \qquad q = \langle\phi_i|\phi_m\rangle$$

Pseudocode: modify state generation

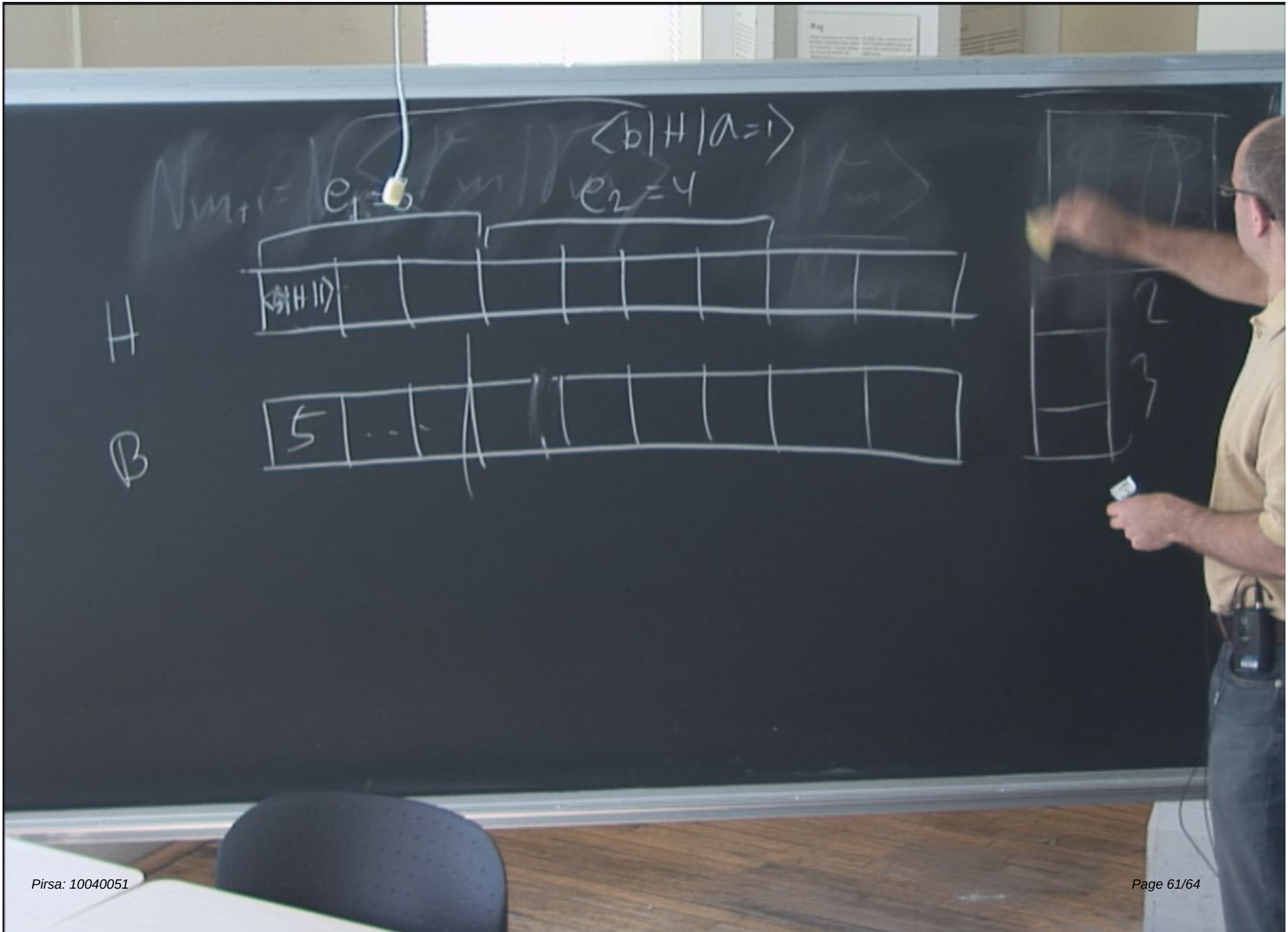
```
do m = 1, Λ − 1
    call hoperation(φ_m, φ_{m+1})
    a_m = ⟨φ_m|φ_{m+1}⟩; φ_{m+1} = φ_{m+1} − a_m φ_m − n_m φ_{m−1}
    call normalize(φ_{m+1}, n_{m+1})
    do i = 1, m
        q = ⟨φ_{m+1}|φ_i⟩; φ_{m+1} = (φ_{m+1} − q φ_i)/(1 − q^2)
    enddo
enddo
```

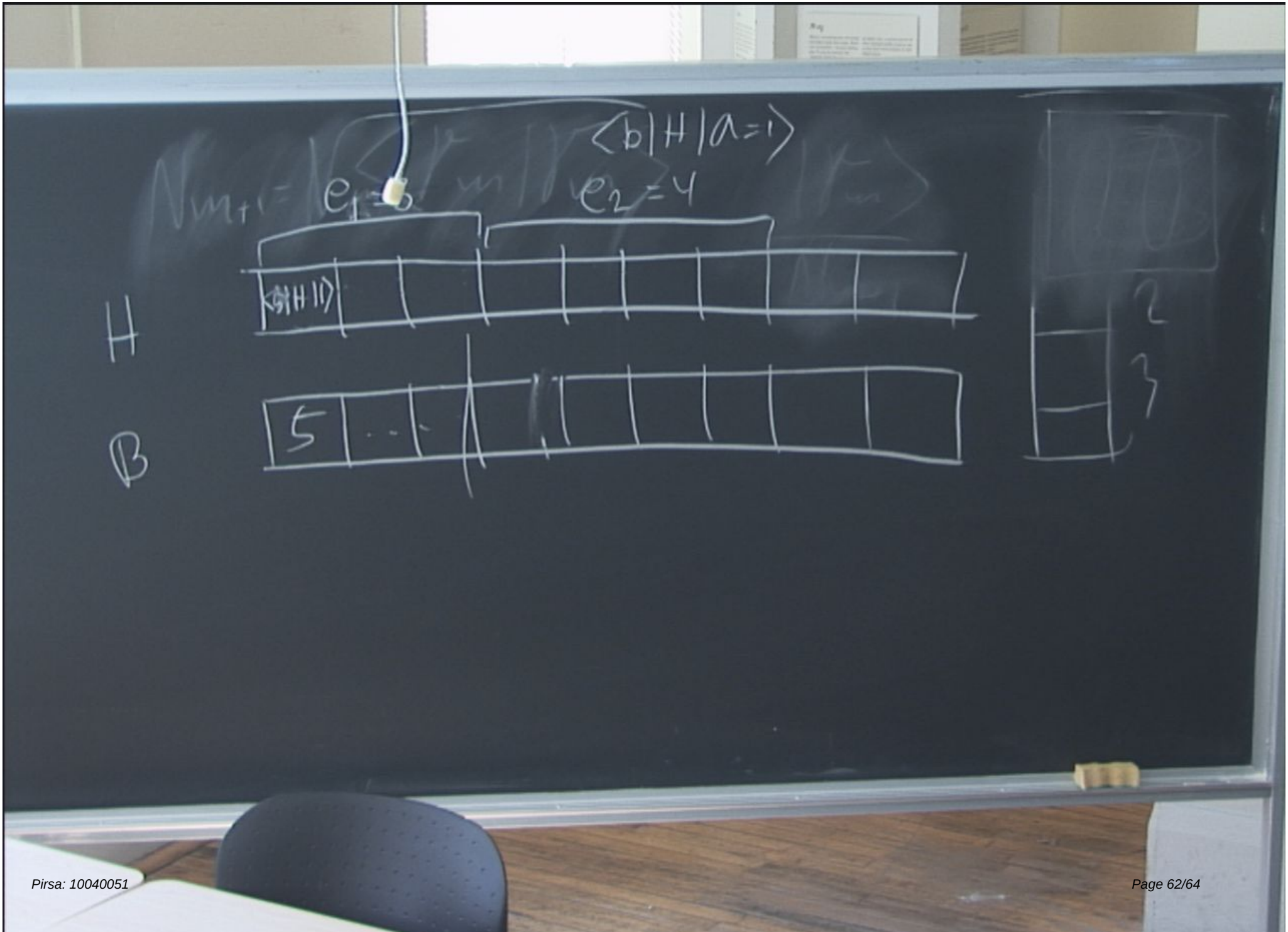
Note: the Lanczos method can only generate a single state of a multiplet

- some random linear combination of degenerate states

Example: 2 degenerate states i, j:

$$H^{\Lambda}|\Psi\rangle = \sum c_m E^{\Lambda}_m |\psi_m\rangle + E^m_{i,j}(c_i|\psi_i\rangle + c_j|\psi_j\rangle)$$


$\langle b|H|\Delta=1\rangle$
$e_1=6$
$e_2=4$
H
B
5


$\langle b|H|0=1\rangle$
$e_1=6$
$e_2=4$
H
B
5
2
3

Spin correlations in the Heisenberg chain

Let's look at the (staggered) spin correlation function

$$C(r) = \langle \mathbf{S}_i \cdot \mathbf{S}_{i+r} \rangle (-1)^r$$

versus the distance r and at r=N/2 versus system size N

Theory (bosonization, conformal field theory) predicts (for large r, N)

$$C(r) \propto \frac{\ln^{1/2}(r/r_0)}{r}$$

Spin correlations in the Heisenberg chain

Let's look at the (staggered) spin correlation function

$$C(r) = \langle \mathbf{S}_i \cdot \mathbf{S}_{i+r} \rangle (-1)^r$$

versus the distance r and at r=N/2 versus system size N

Theory (bosonization, conformal field theory) predicts (for large r, N)

$$C(r) \propto \frac{\ln^{1/2}(r/r_0)}{r}$$