Title: Quantum key expansion based on entanglement-assisted quantum LDPC codes

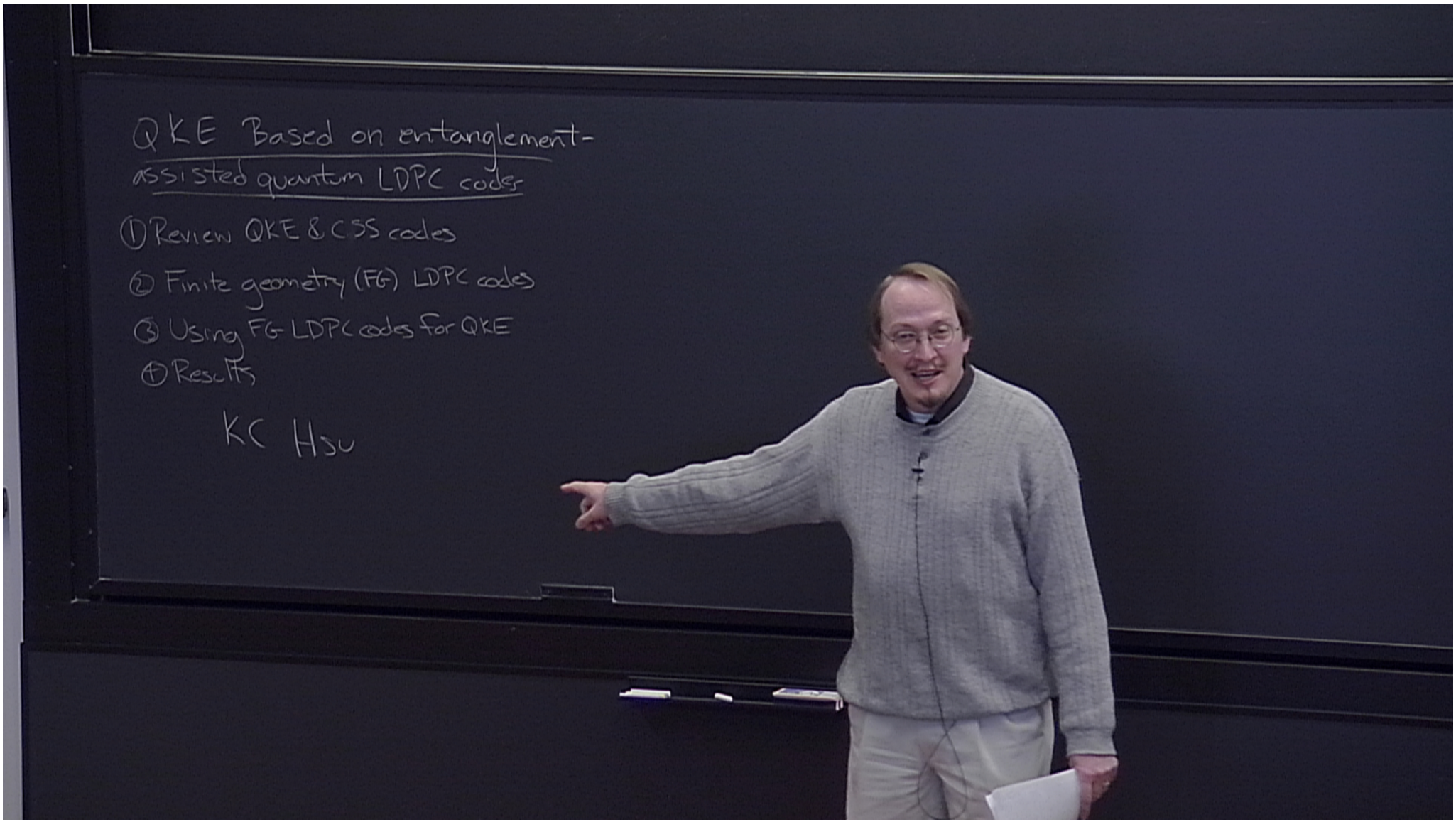
Date: Apr 03, 2013 04:00 PM

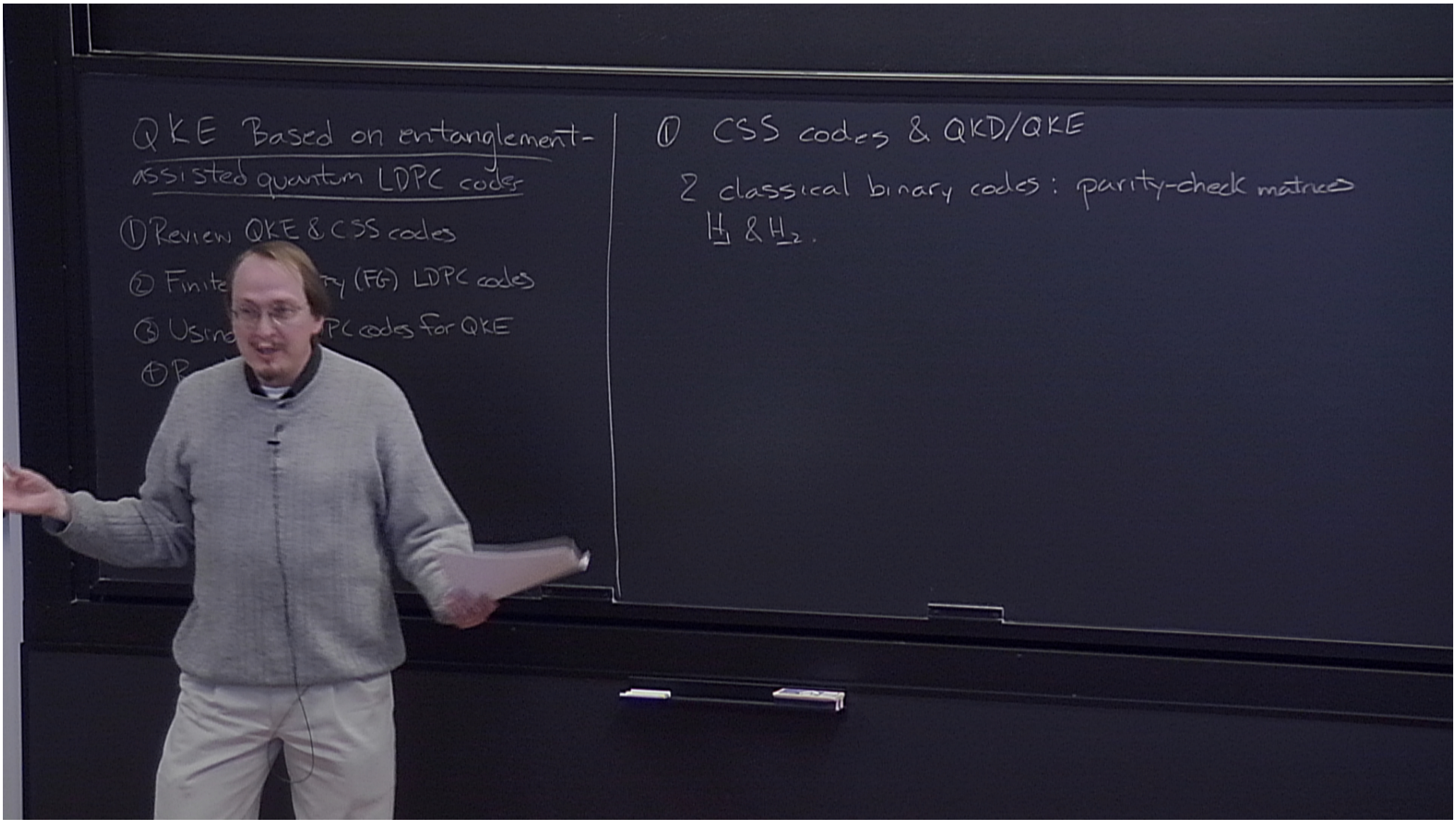
URL: http://www.pirsa.org/13040108

Abstract: Quantum key distribution protocols can be based on
quantum error correcting codes, where the structure of the code determines the
post processing protocol applied to a raw key produced by BB84 or a similar
scheme. Luo and Devetak showed that
basing a similar protocol on entanglement-assisted quantum error-correcting
codes (EAQECCs) leads to quantum key expansion (QKE) protocols, where some
amount of previously shared secret key is used as a seed in the post-processing
stage to produce a larger secret key. One of the promising aspects of EAQECCs
is that they can be constructed from classical linear codes that don't satisfy
the dual-containing property, which among other things allows the use of low
density parity-check (LDPC) codes with girth greater than 4, for which the
iterative decoding algorithm has better performance. We looked into QKE based on a family of
EAQECCs generated by classical finite geometry (FG) LDPC codes. Very efficient iterative decoders exist for
these codes, and they were shown by Hsieh, Yen and Hau to produce quantum LDPC
codes that require very little entanglement.
We modify the original QKE protocol to detect bad code blocks without
the consumption of secret key when the protocol fails. This allows us to greatly reduce the bit
error rate of the key, at the cost of a minor reduction in the net key
production rate, but without increasing the consumption rate of pre-shared
key. Numerical simulations for the
family of FG LDPC codes show that this improved QKE protocol has a good net key
production rate even at relatively high error rates, for appropriate code
choices.

QKE Based on entanglement-assisted quantum LDPC codes

1. Review QKE & CSS codes
2. Finite geometry (FG) LDPC codes
3. Using FG LDPC codes for QKE
4. Results


QKE Based on entanglement-
assisted quantum LDPC codes
① Review QKE & CSS codes
② Finite geometry (FG) LDPC codes
③ Using FG LDPC codes for QKE
④ Results
KC Hsu


QKE Based on entanglement-assisted quantum LDPC codes
① Review QKE & CSS codes
② Finite (FG) LDPC codes
③ Using PC codes for QKE
① CSS codes & QKD/QKE
2 classical binary codes: parity-check matrices
H_1 & H_2.

QKE Based on entanglement-assisted quantum LDPC codes

① Review QKE & CSS codes

② Finite geometry (FG) LDPC codes

③ Using FG LDPC codes for QKE

④ Results

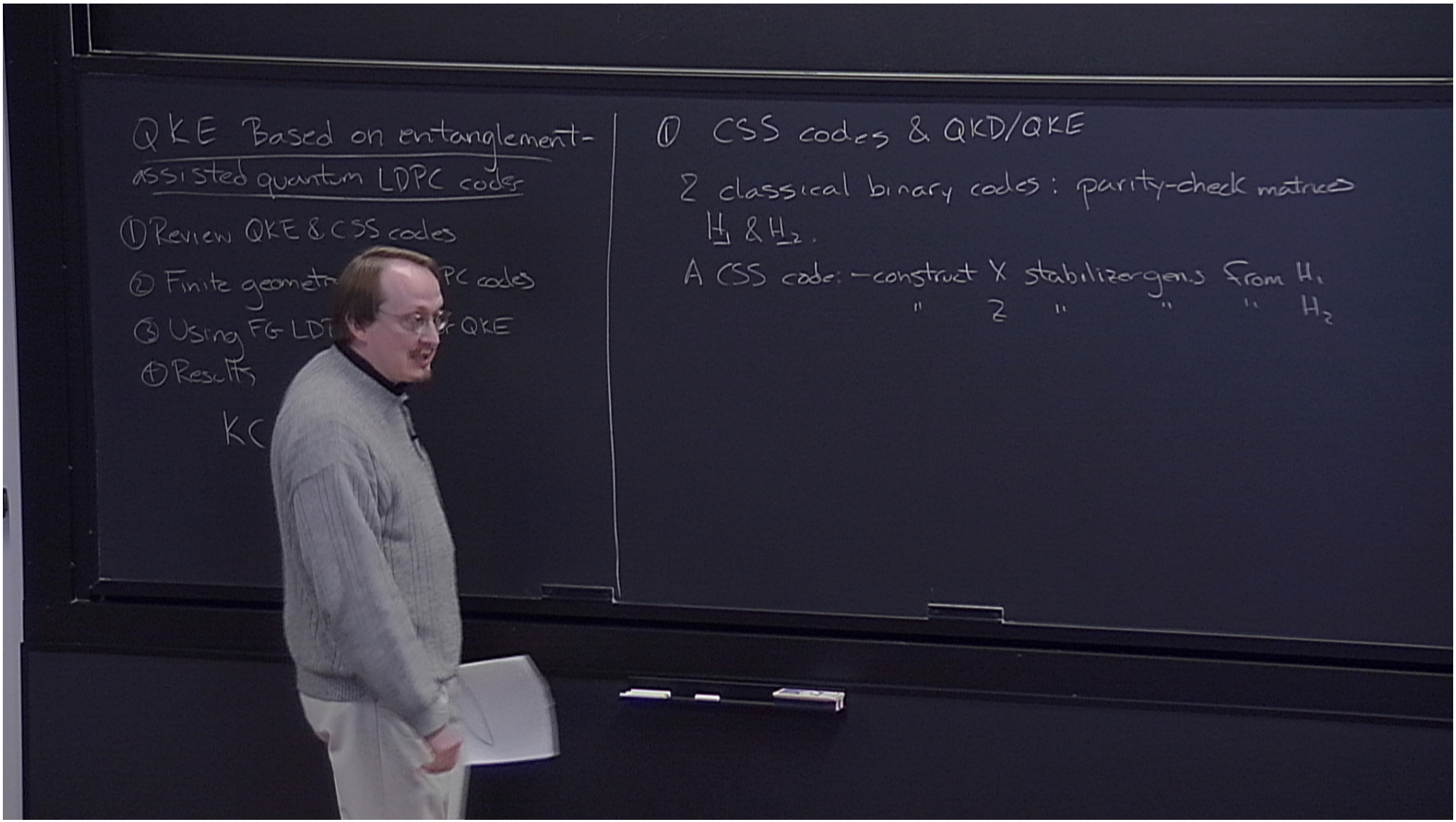
KC Hsu

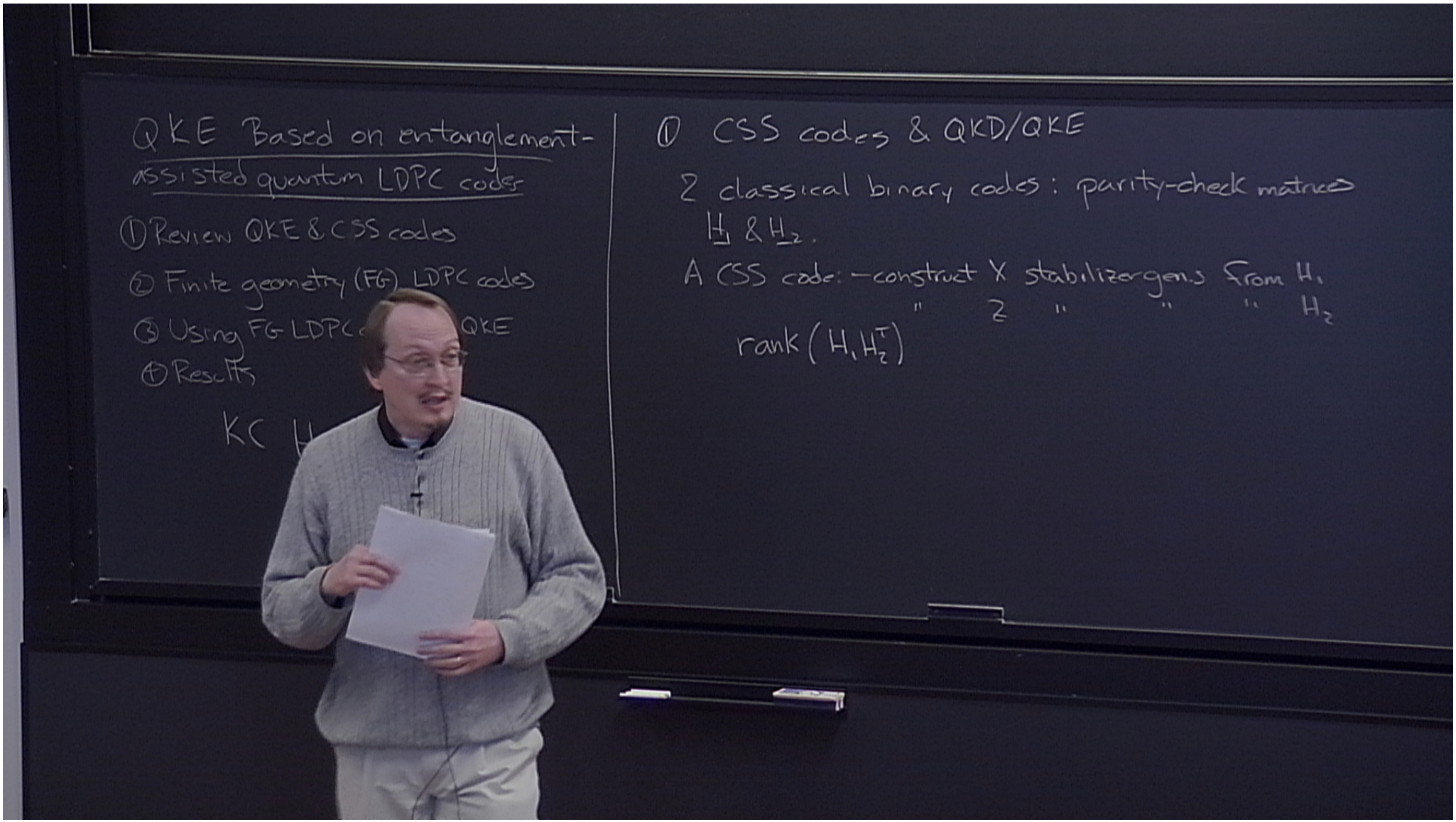
① CSS codes & QKD/QKE

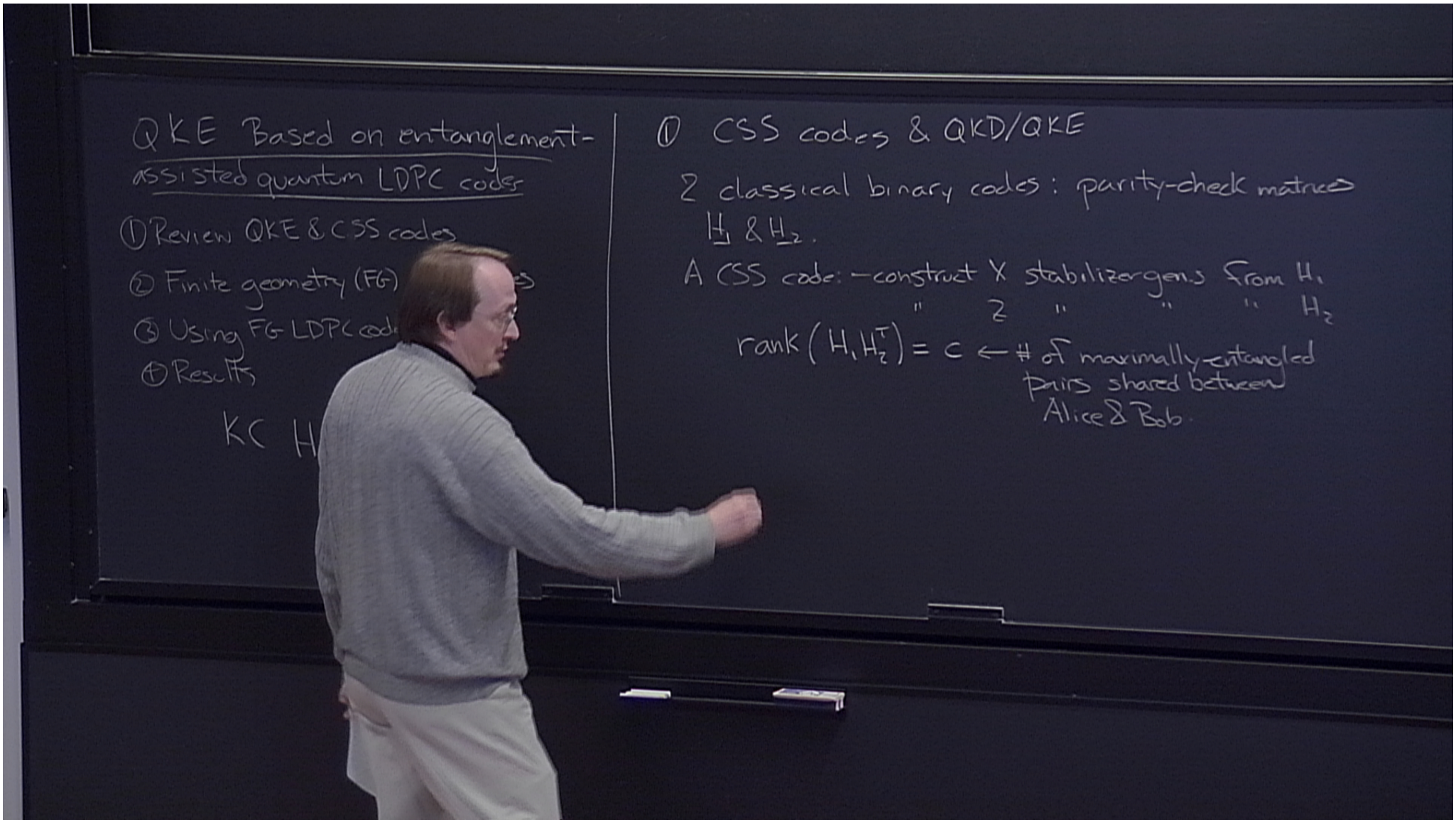
2 classical binary codes: parity-check matrices H_1 & H_2.

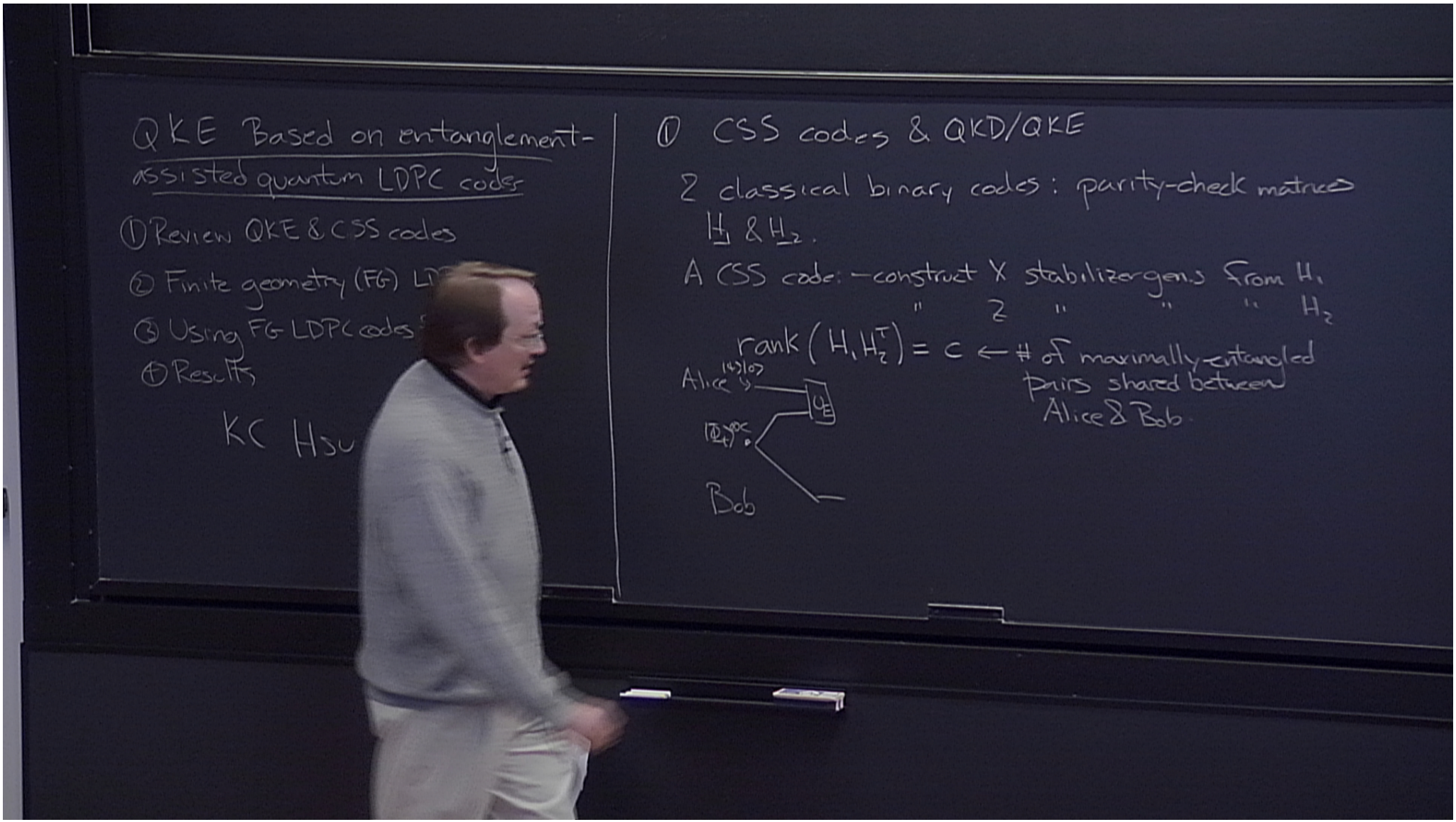
A CSS code: – construct X stabilizer gens from H_1

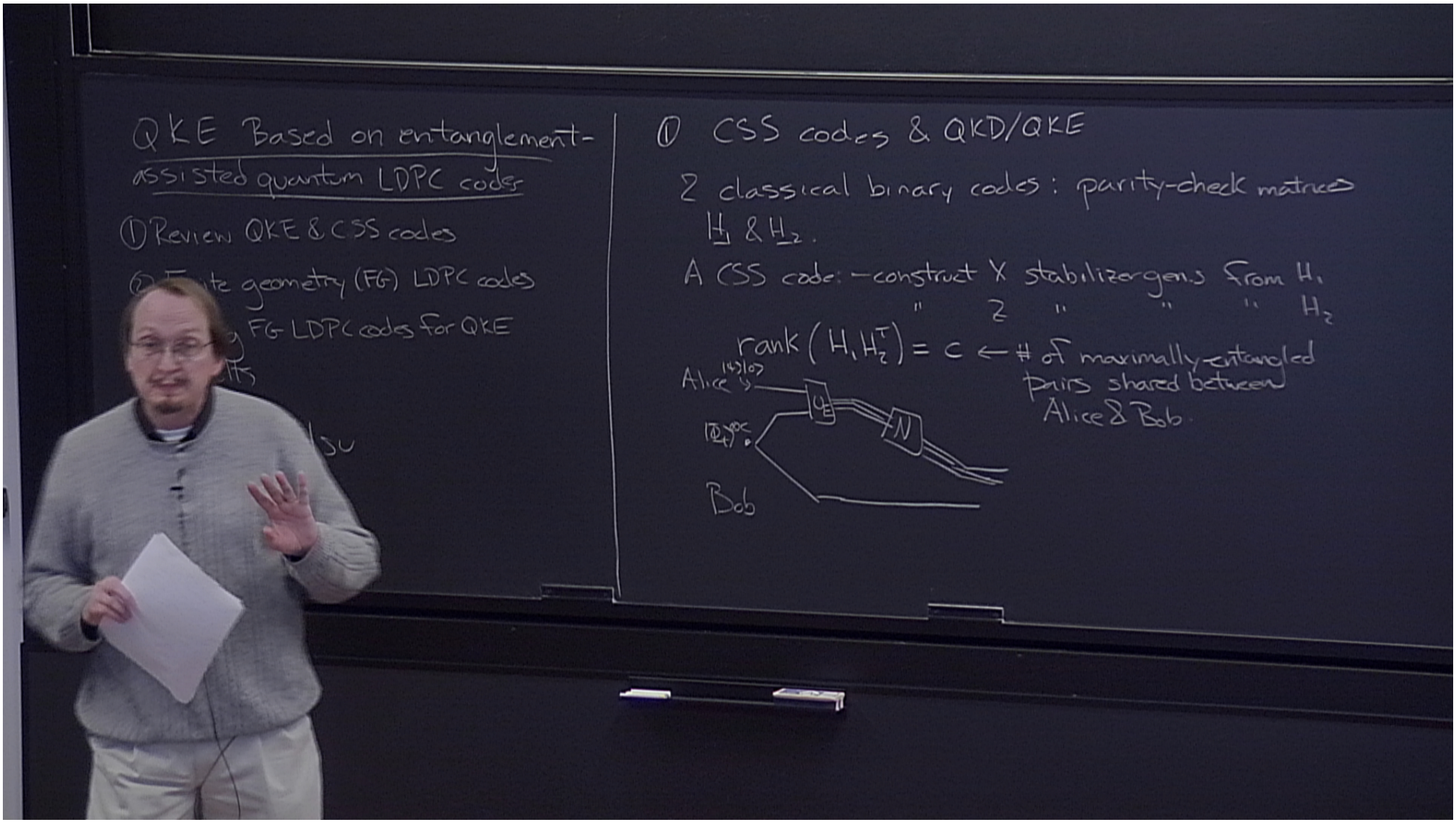
" Z " " " H_2

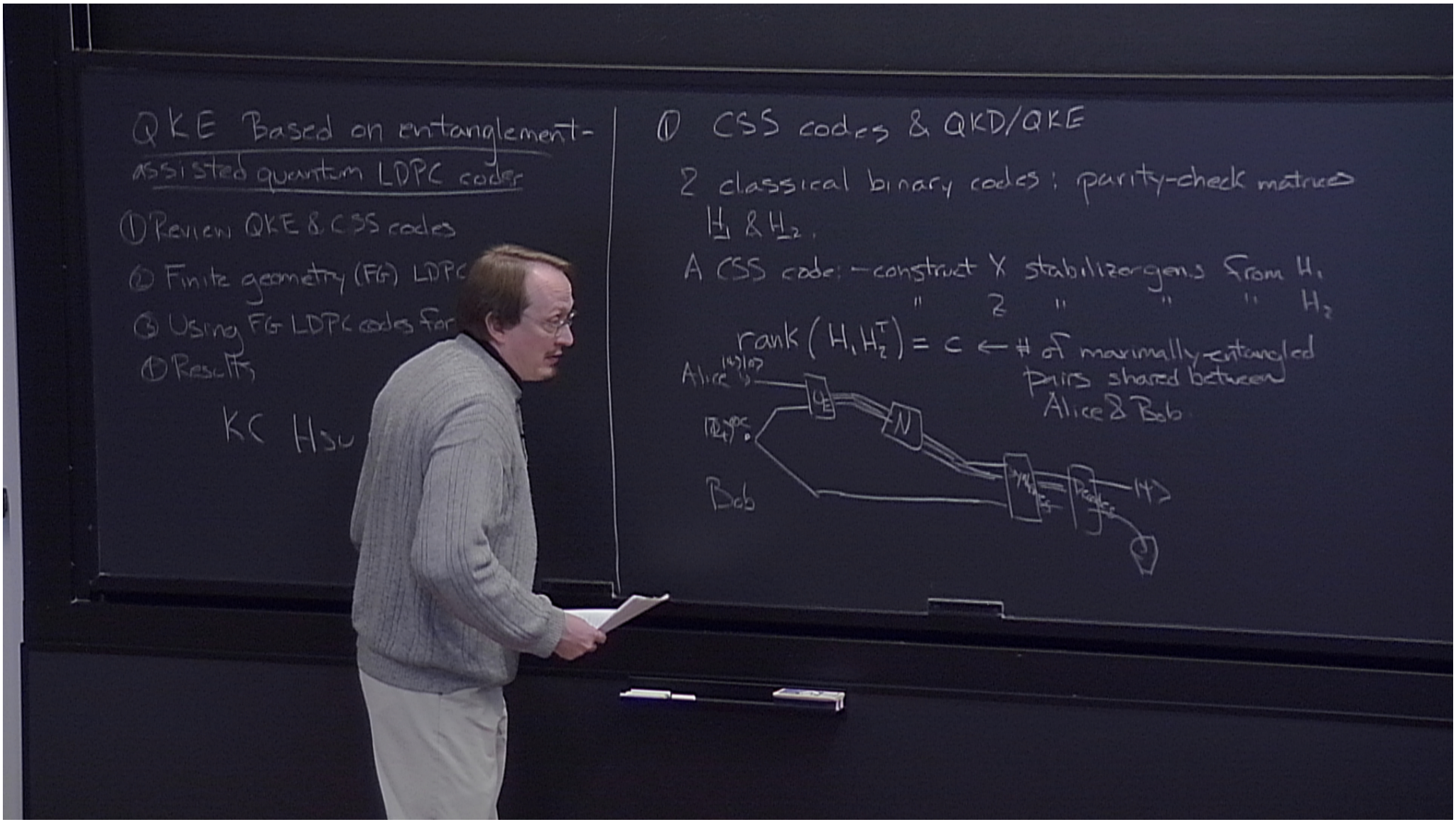

QKE Based on entanglement-assisted quantum LDPC codes
① Review QKE & CSS codes
② Finite geometr PC codes
③ Using FG LDP QKE
④ Results
KC
① CSS codes & QKD/QKE
2 classical binary codes: parity-check matrices
H1 & H2.
A CSS code: – construct X stabilizer gens from H1
" Z " " " H2

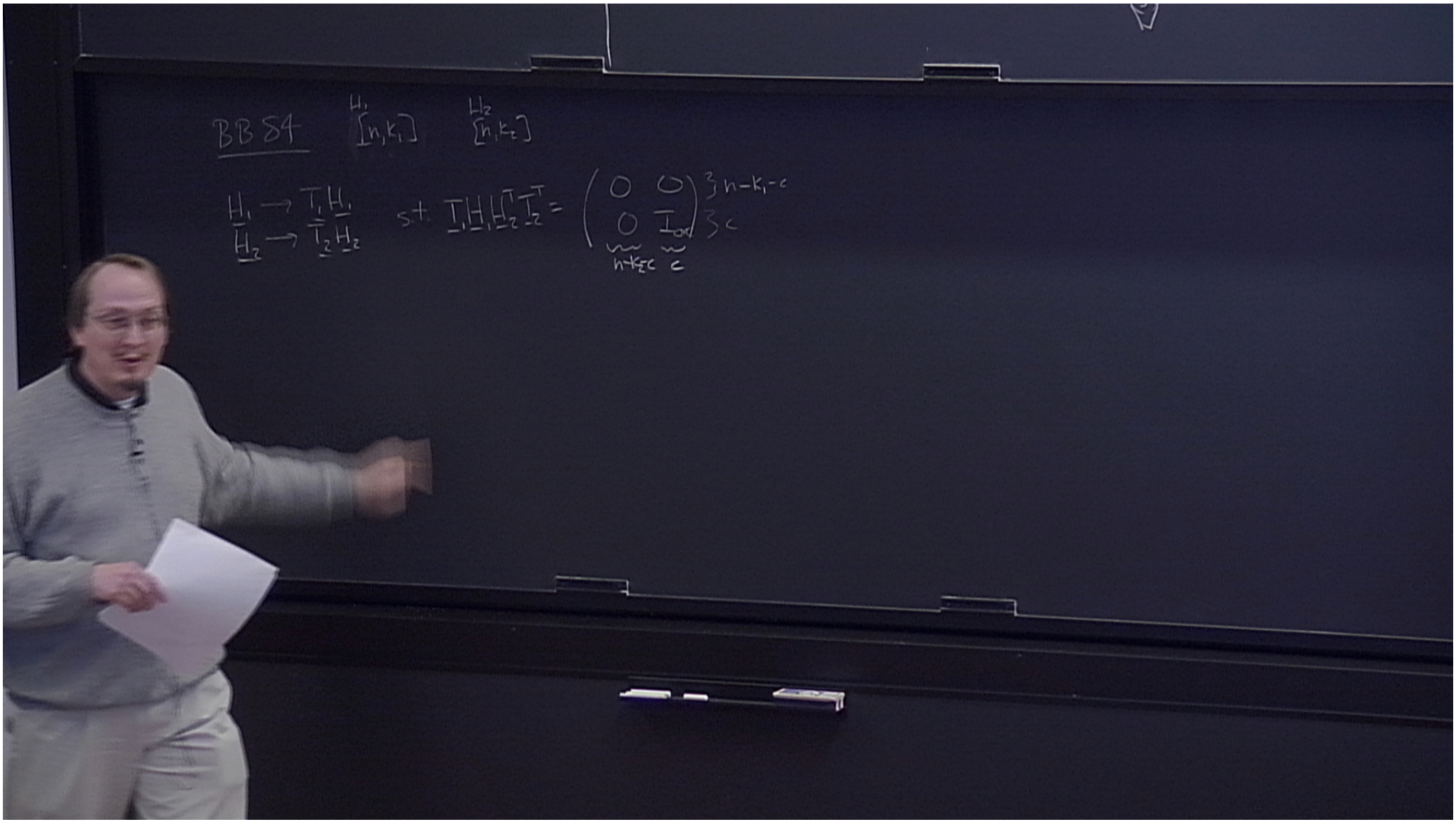

QKE Based on entanglement-assisted quantum LDPC codes
① Review QKE & CSS codes
② Finite geometry (FG) LDPC codes
③ Using FG LDPC
QKE
④ Results
① CSS codes & QKD/QKE
2 classical binary codes: parity-check matrices
H1 & H2.
A CSS code: – construct X stabilizer gens from H1
" Z " " " H2
rank (H1 H2^T)

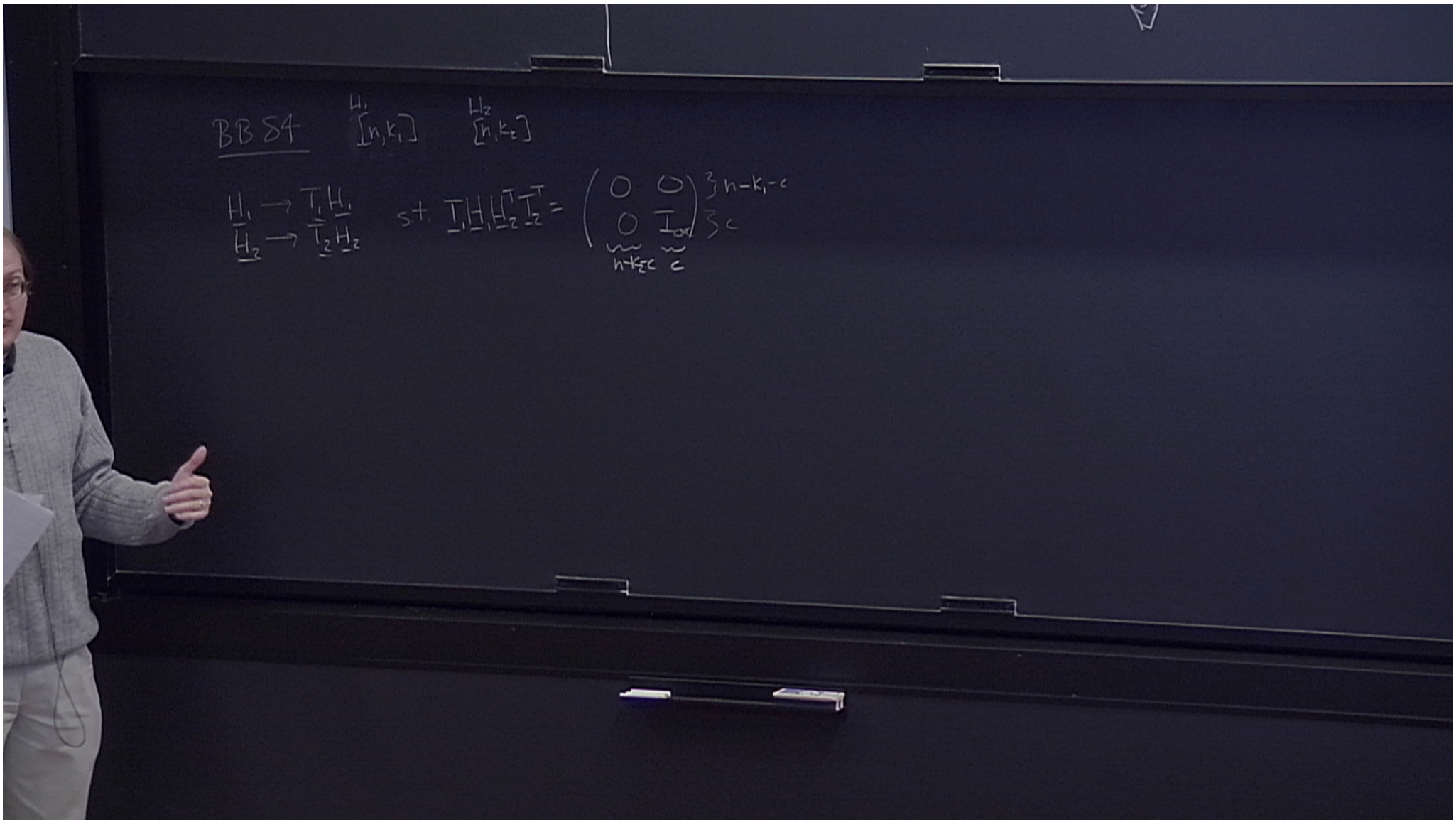

QKE Based on entanglement-assisted quantum LDPC codes
① Review QKE & CSS codes
② Finite geometry (FG)
③ Using FG LDPC
④ Results
① CSS codes & QKD/QKE
2 classical binary codes: parity-check matrices H_1 & H_2.
A CSS code: – construct X stabilizer gens from H_1
" Z " " " H_2
$\mathrm{rank}(H_1 H_2^T) = c$ ← # of maximally-entangled pairs shared between Alice & Bob.

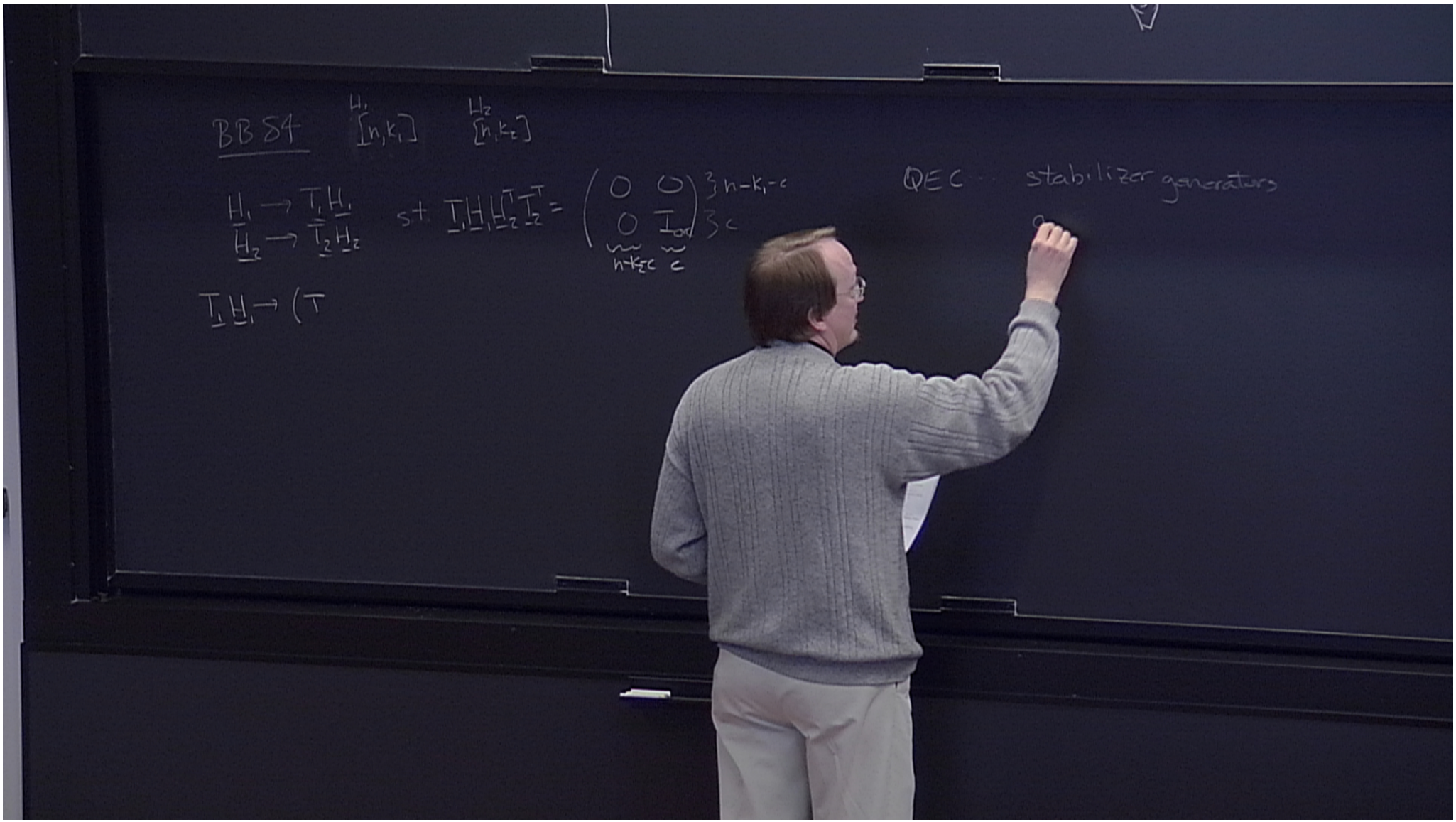

QKE Based on entanglement-assisted quantum LDPC codes
① Review QKE & CSS codes
② Finite geometry (FG) LD
③ Using FG LDPC codes
④ Results
KC Hsu
① CSS codes & QKD/QKE
2 classical binary codes: parity-check matrices H_1 & H_2.
A CSS code: – construct X stabilizer gens from H_1
" Z " " " H_2
$\mathrm{rank}(H_1 H_2^T) = c$ ← # of maximally-entangled pairs shared between Alice & Bob.
Alice
Bob

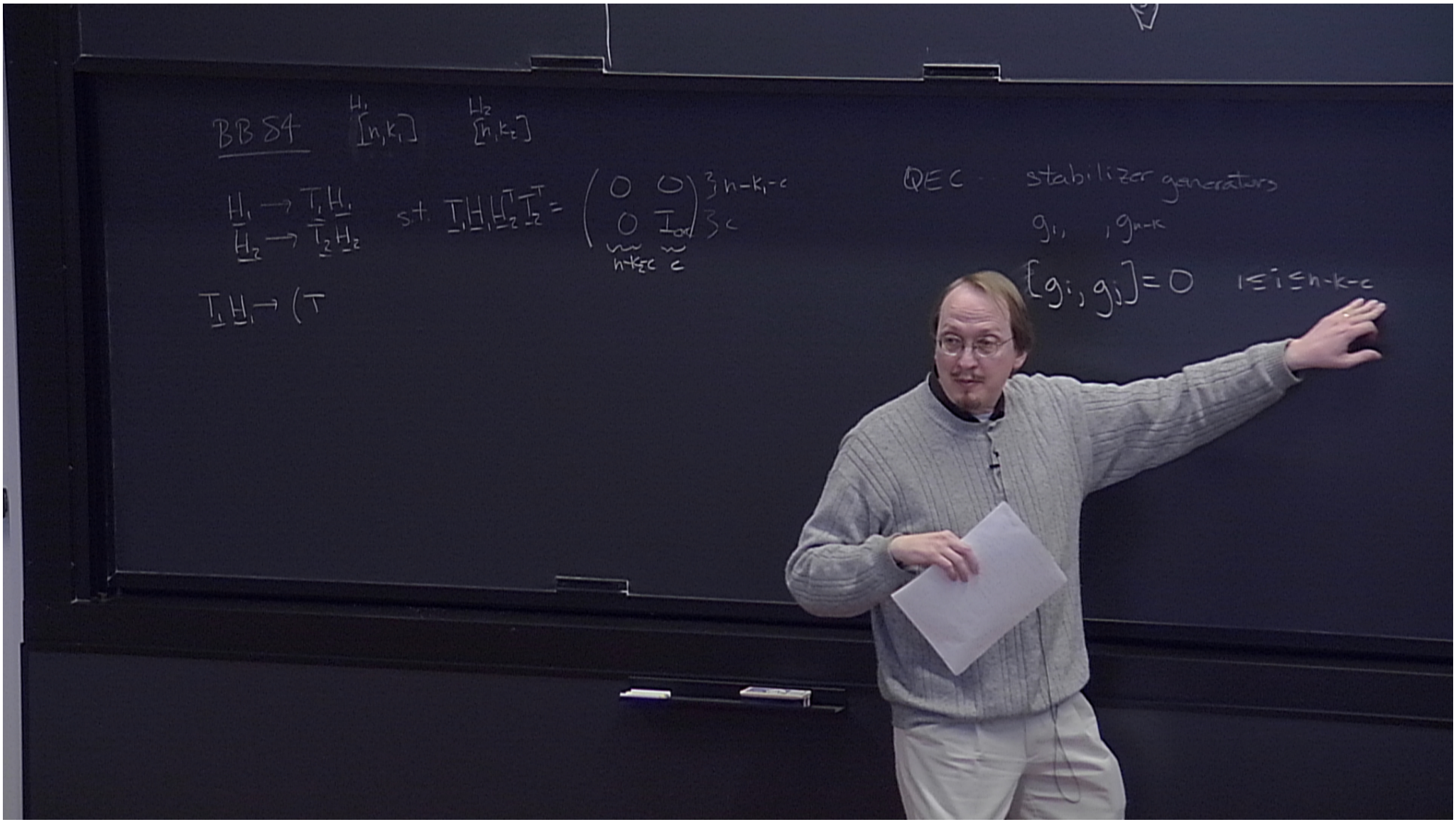

QKE Based on entanglement-assisted quantum LDPC codes
① Review QKE & CSS codes
② Finite geometry (FG) LDPC codes
FG LDPC codes for QKE
① CSS codes & QKD/QKE
2 classical binary codes: parity-check matrices H_1 & H_2.
A CSS code: – construct X stabilizer gens from H_1
" Z " " " H_2
$\mathrm{rank}(H_1 H_2^T) = c$ ← # of maximally-entangled pairs shared between Alice & Bob.
Alice
Bob
N

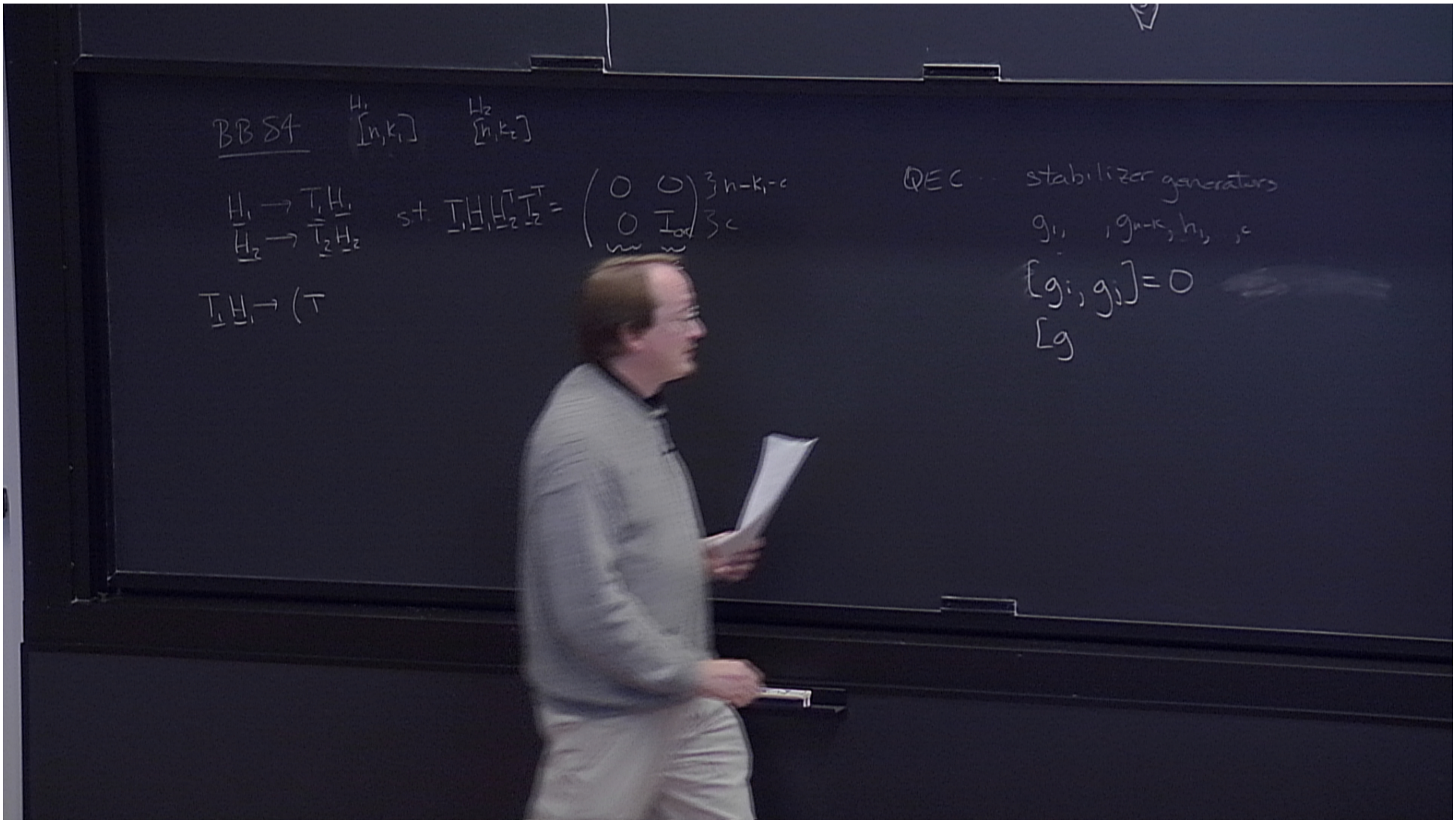

QKE Based on entanglement-assisted quantum LDPC codes
① Review QKE & CSS codes
② Finite geometry (FG) LDPC
③ Using FG LDPC codes for
④ Results
KC Hsu
① CSS codes & QKD/QKE
2 classical binary codes: parity-check matrices
H1 & H2.
A CSS code: – construct X stabilizer gens from H1
" Z " " " H2
rank (H1 H2^T) = c ← # of maximally-entangled pairs shared between Alice & Bob
Alice
Bob
N
|ψ⟩

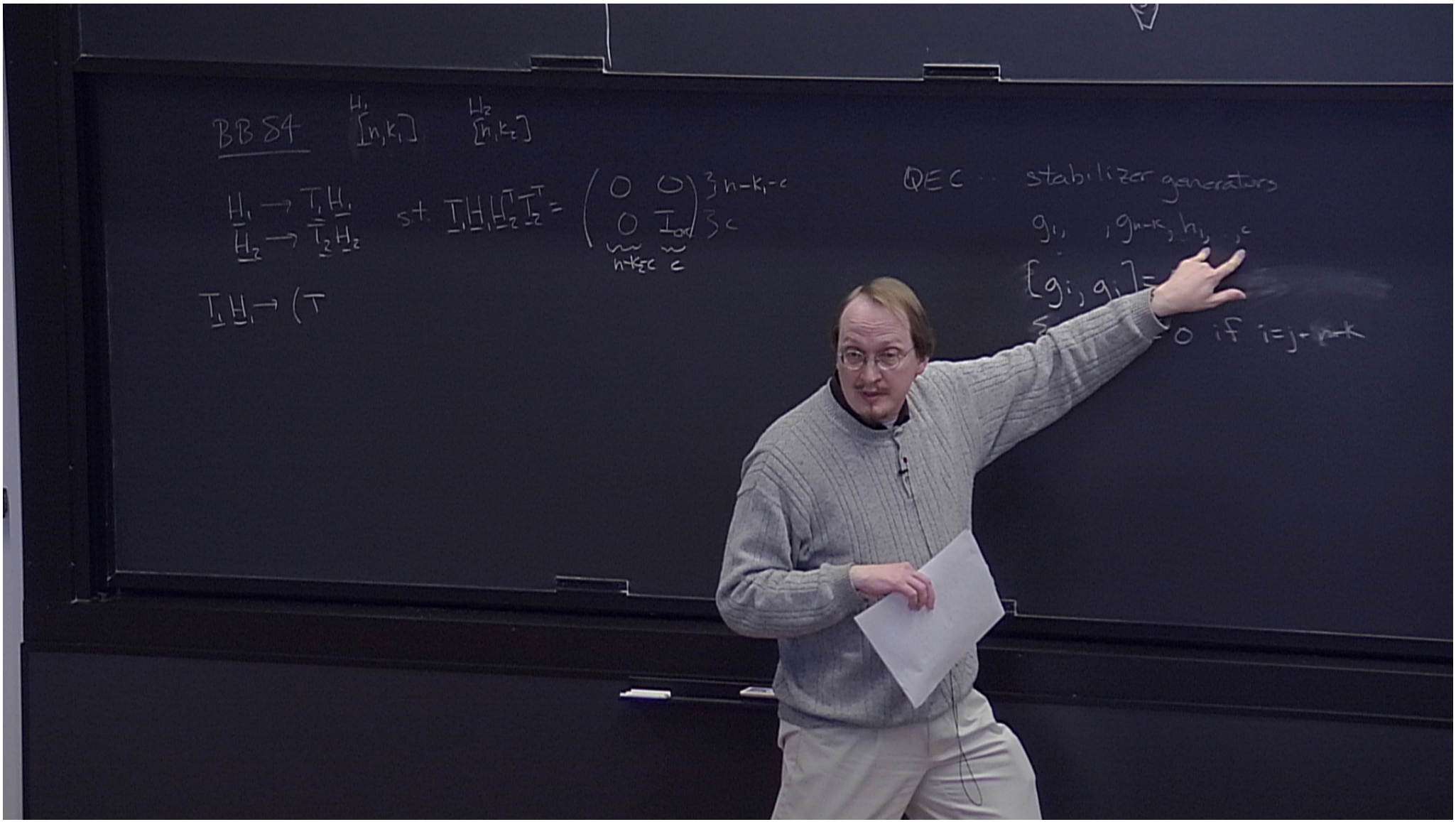

BB84
H1 [n,k1]
H2 [n,k2]
H1 → T1 H1
H2 → T2 H2
s.t. T1 H1 H2^T T2^T =
n-k1-c
c
n-k2-c
c

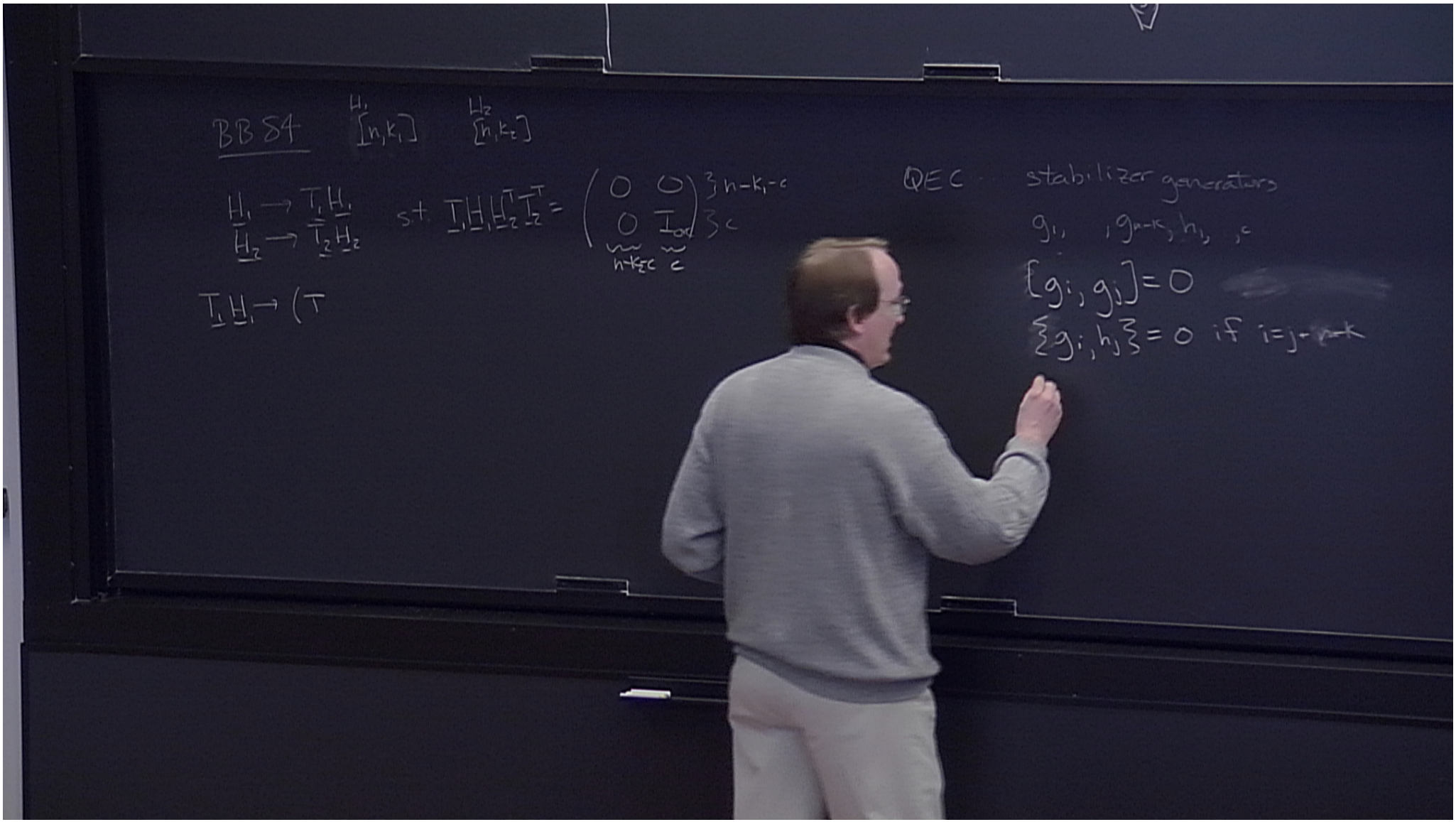

BB84
H_1 $[n,k_1]$
H_2 $[n,k_2]$
$H_1 \to T_1 H_1$
$H_2 \to T_2 H_2$
s.t. $T_1 H_1 H_2^T T_2^T = \begin{pmatrix} 0 & 0 \\ 0 & I_{c\times c} \end{pmatrix}$
$n-k_1-c$
c
$n-k_2-c$
c

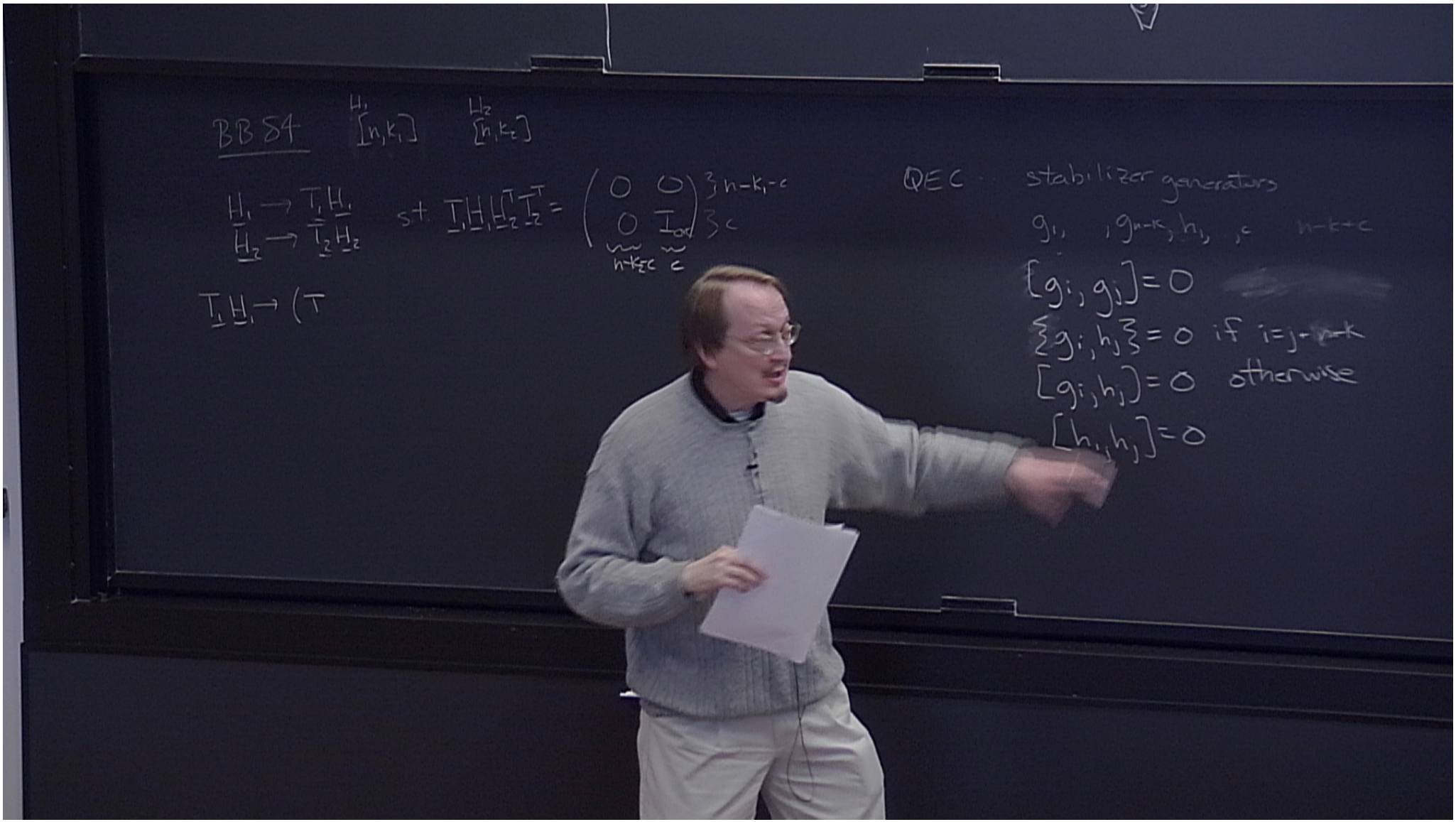

BB84
H1 [n,k1]
H2 [n,k2]
H1 → T1 H1
H2 → T2 H2
s.t. T1 H1 H2^T T2^T =
n-k1-c
c
n-k2-c
c
T1 H1 → (T
QEC ... stabilizer generators

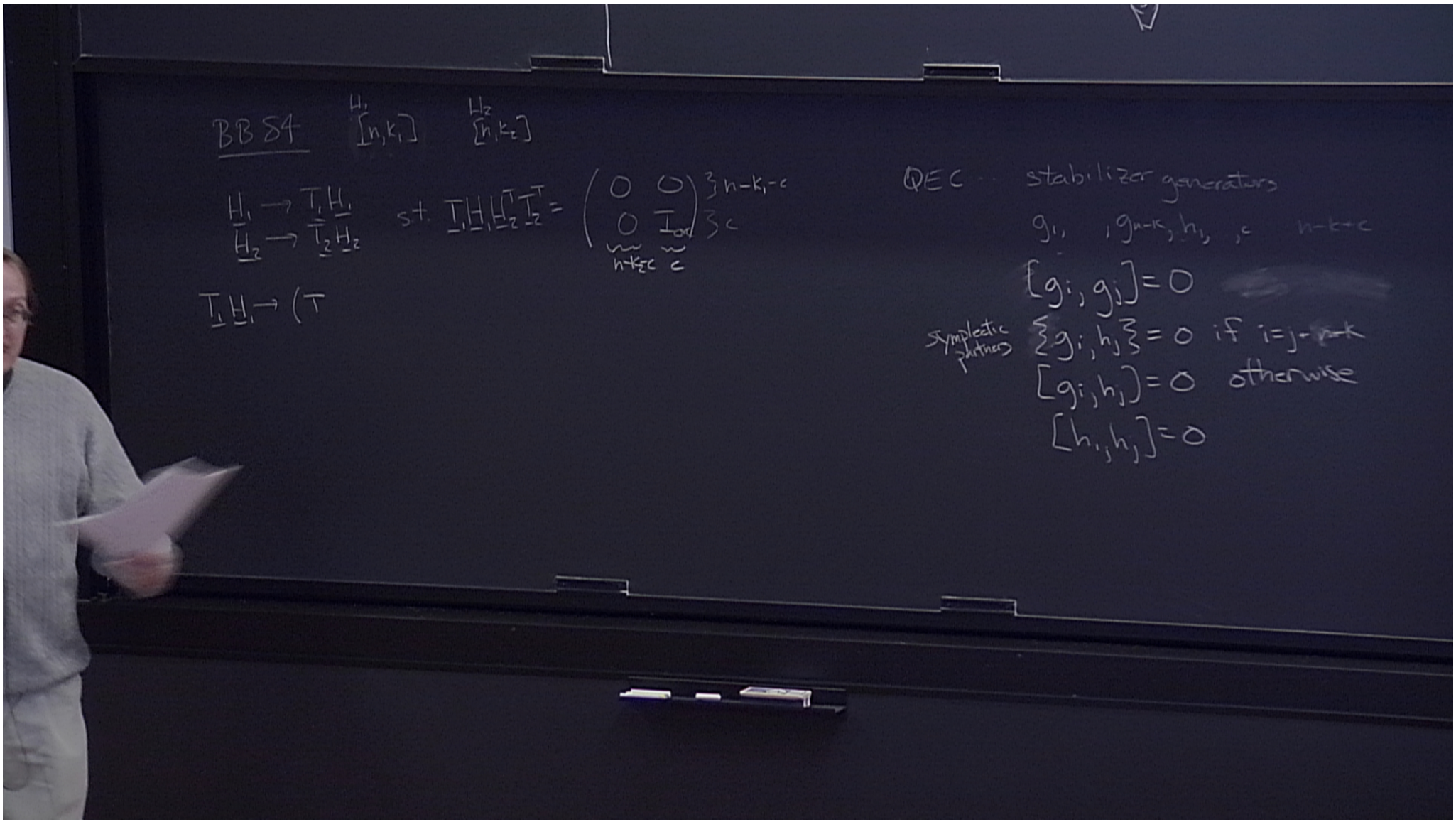

BB84
H1 [n,k1]
H2 [n,k2]
H1 → T1 H1
H2 → T2 H2
s.t. T1 H1 H2^T T2^T =
n-k1-c
c
n-k2-c
c
T1 H1 → (T
QEC stabilizer generators
g1, , gn-k
[gi, gj] = 0 1 ≤ i ≤ n-k-c

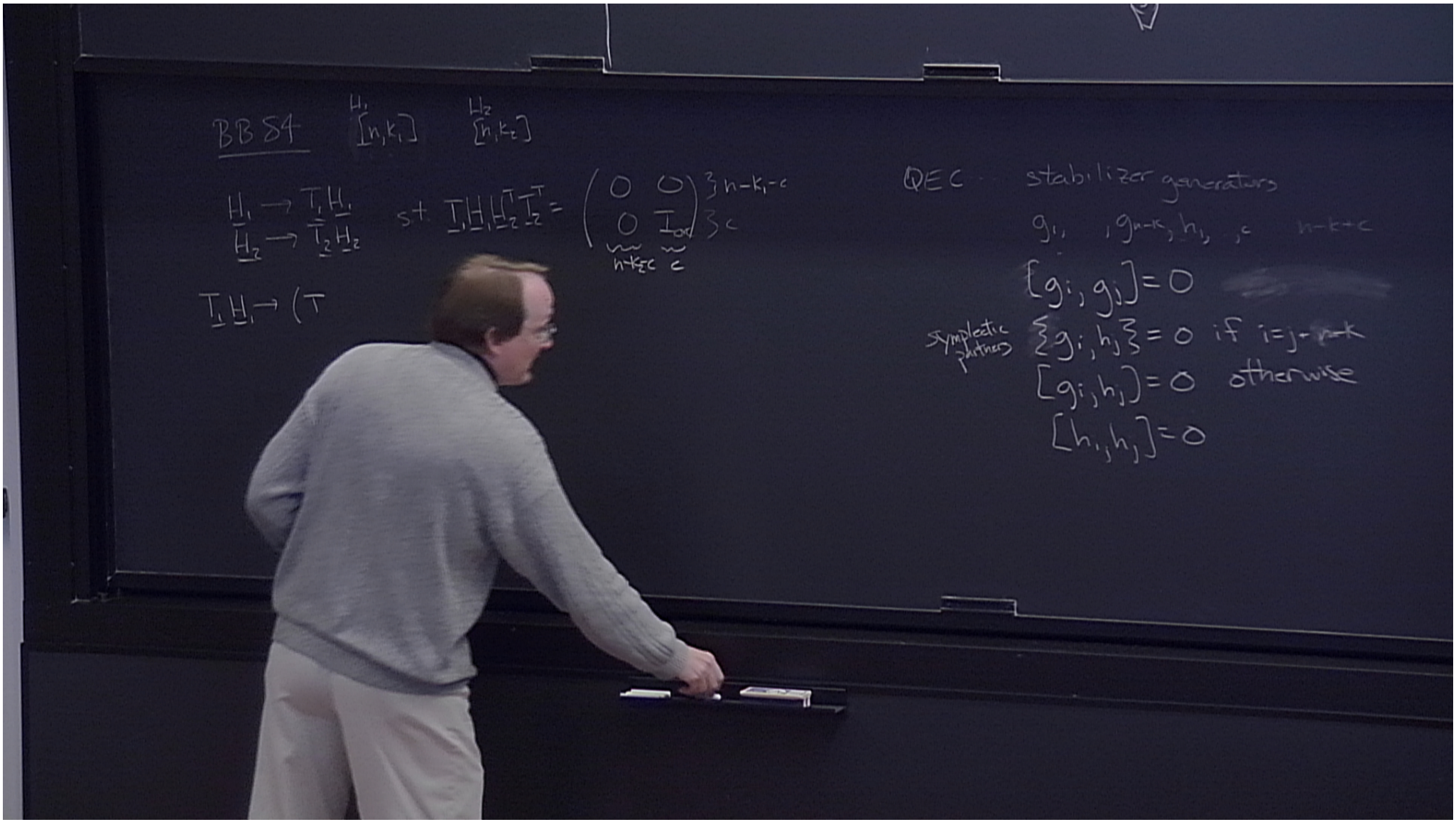

BB84
H1 [n,k1]
H2 [n,k2]
H1 → T1 H1
H2 → T2 H2
s.t. T1 H1 H2^T T2^T =
n-k1-c
c
T1 H1 → (T
QEC
stabilizer generators
[gi, gj] = 0
[g

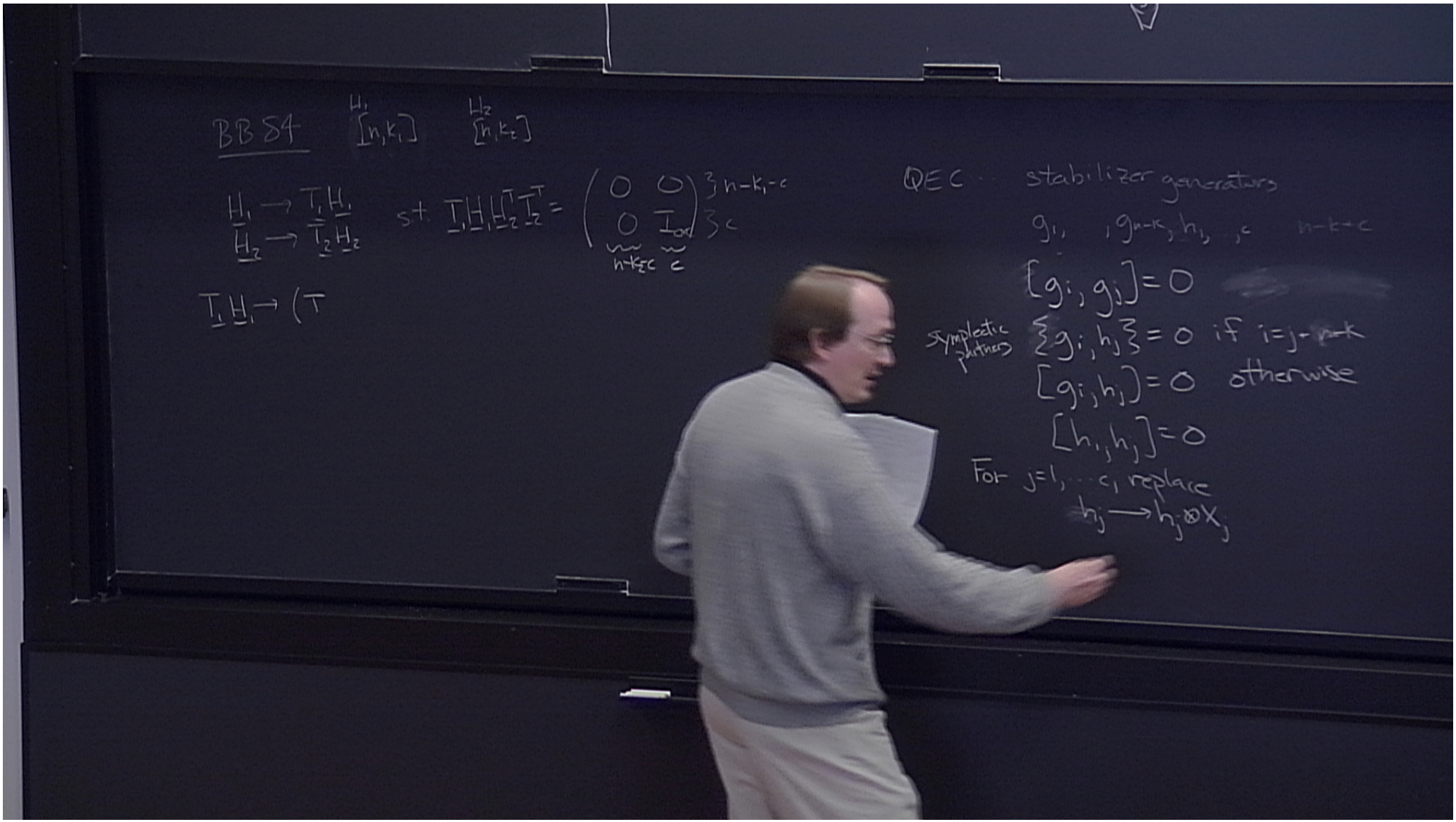

BB84
H1 [n,k1]
H2 [n,k2]
H1 → T1 H1
H2 → T2 H2
s.t. T1 H1 H2^T T2^T =
n-k1-c
c
n-k2-c
c
T1 H1 → (T
QEC
stabilizer generators
g1, , gn-k, h1, , c
[gi, gj] =
= 0 if i=j-n-k

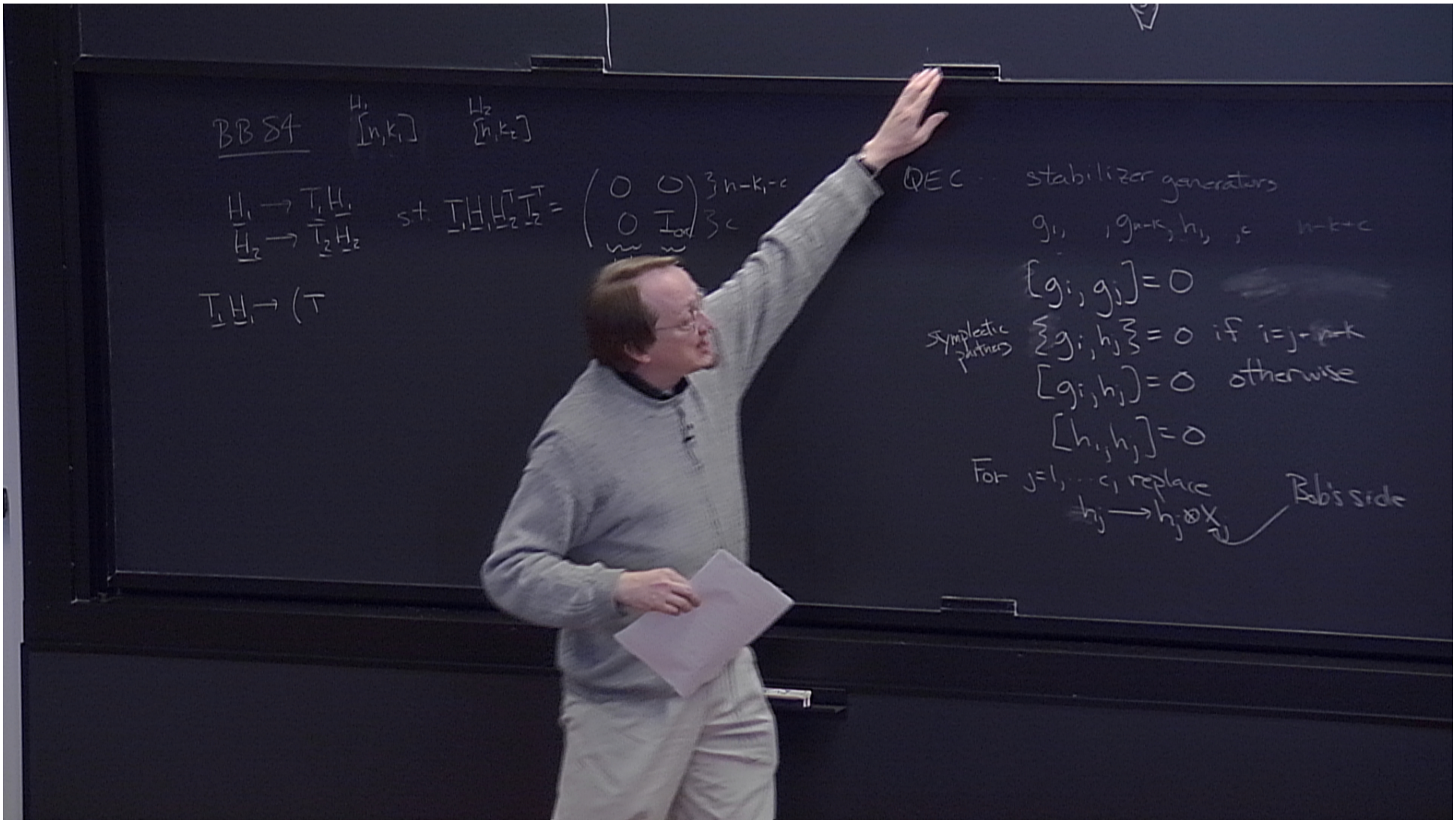

BB84
H1 [n,k1]
H2 [n,k2]
H1 → T1 H1
H2 → T2 H2
s.t. T1 H1 H2^T T2^T =
n-k1-c
c
n-k2-c
QEC
stabilizer generators
g1, , gn-k, h1, , c
[gi, gj] = 0
{gi, hj} = 0 if i=j

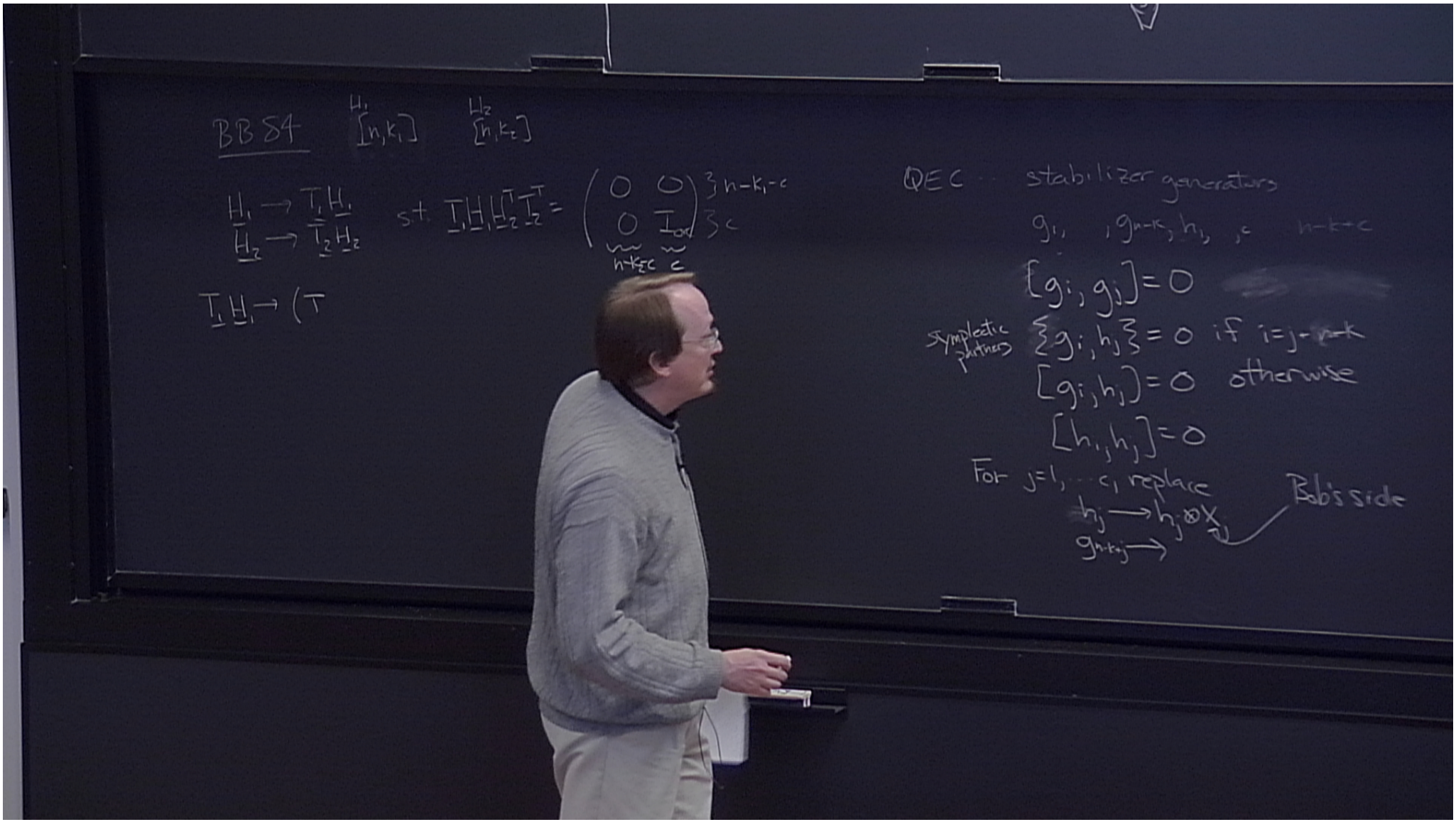

BB84
H1 [n,k1]
H2 [n,k2]
H1 → T1 H1
H2 → T2 H2
s.t. T1 H1 H2^T T2^T =
n-k1-c
c
n-k2-c
c
T1 H1 → (T
QEC
stabilizer generators
g1, , gn-k, h1, , c
n-k-c
[gi, gj] = 0
{gi, hj} = 0 if i=j
[gi, hj] = 0 otherwise
[hi, hj] = 0

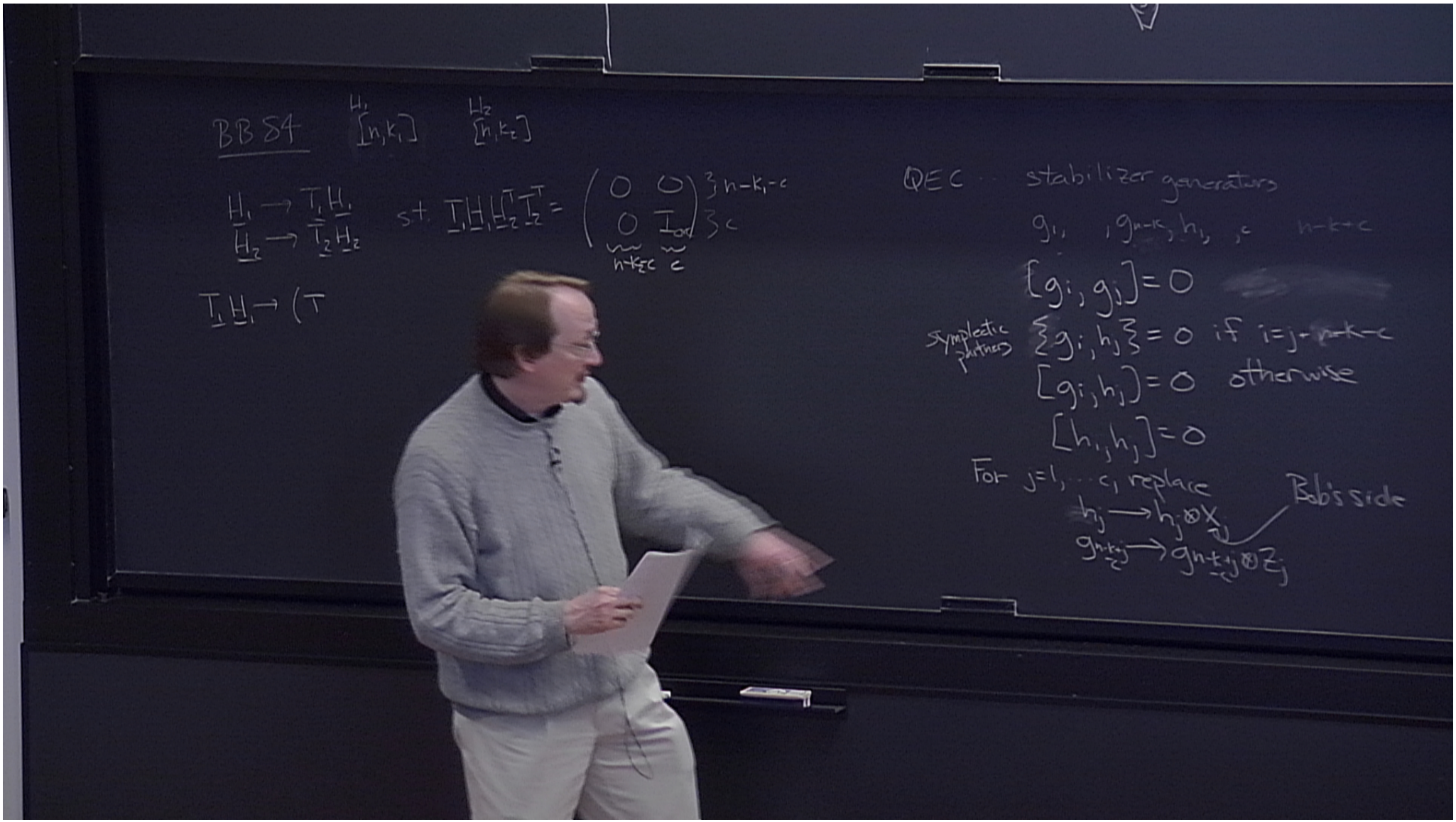

BB84
H1 [n,k1]
H2 [n,k2]
H1 → T1 H1
H2 → T2 H2
s.t. T1 H1 H2^T T2^T = (0 0; 0 I_ck) } n-k1-c, } c
n-k2-c c
T1 H1 → (T
QEC stabilizer generators
g1, , gn-k, h1, , c n-k+c
[gi, gj] = 0
symplectic partners {gi, hj} = 0 if i=j-n-k
[gi, hj] = 0 otherwise
[hi, hj] = 0


BB84
H1 [n,k1]
H2 [n,k2]
H1 → T1 H1
H2 → T2 H2
s.t. T1 H1 H2^T T2^T =
n-k1-c
c
T1 H1 → (T
QEC stabilizer generators
g1, , gn-k, h1, , c n-k+c
[gi, gj] = 0
symplectic partners
{gi, hj} = 0 if i=j-n-k
[gi, hj] = 0 otherwise
[hi, hj] = 0


BB84
H1 [n,k1]
H2 [n,k2]
H1 → T1 H1
H2 → T2 H2
s.t. T1 H1 H2^T T2^T =
n-k1-c
c
T1 H1 → (T
QEC ... stabilizer generators
g1, ..., gn-k, h1, ..., c
n-k+c
[gi, gj] = 0
symplectic partners
{gi, hj} = 0 if i=j-n-k
[gi, hj] = 0 otherwise
[hi, hj] = 0
For j=1, ... c, replace
hj → hj ⊗ Xj


BB84
H1 [n,k1]
H2 [n,k2]
H1 → T1 H1
H2 → T2 H2
s.t. T1 H1 H2^T T2^T =
n-k1-c
c
T1 H1 → (T
QEC
stabilizer generators
g1, , gn-k, h1, , c
n-k-c
[gi, gj] = 0
symplectic partners
{gi, hj} = 0 if i=j
[gi, hj] = 0 otherwise
[hi, hj] = 0
For j=1, ... c, replace
hj → hj ⊗ Xj
Bob's side


BB84
H1 [n,k1]
H2 [n,k2]
H1 → T1 H1
H2 → T2 H2
s.t. T1 H1 H2^T T2^T =
n-k1-c
c
T1 H1 → (T
QEC ... stabilizer generators
[gi, gj] = 0
symplectic partners
{gi, hj} = 0 if i=j-n-k
[gi, hj] = 0 otherwise
[hi, hj] = 0
For j=1, ... c, replace
hj → hj ⊗ Xj
gn-k+j →
Bob's side


BB84
H1 [n,k1]
H2 [n,k2]
H1 → T1 H1
H2 → T2 H2
s.t. T1 H1 H2^T T2^T =
n-k1-c
c
n-k2-c
T1 H1 → (T
QEC stabilizer generators
g1, , gn-k, h1, , c
n-k-c
[gi, gj] = 0
symplectic partners
{gi, hj} = 0 if i=j
[gi, hj] = 0 otherwise
[hi, hj] = 0
For j=1, ... c, replace
hj → hj ⊗ Xj
gn-k+j → gn-k+j ⊗ Zj
Bob's side

QKE Based on entanglement-assisted quantum LDPC codes

① Review QKE & CSS codes

② Finite geometry (FG) LDPC codes

③ Using FG LDPC codes for QKE

④ Results

KC Hsu

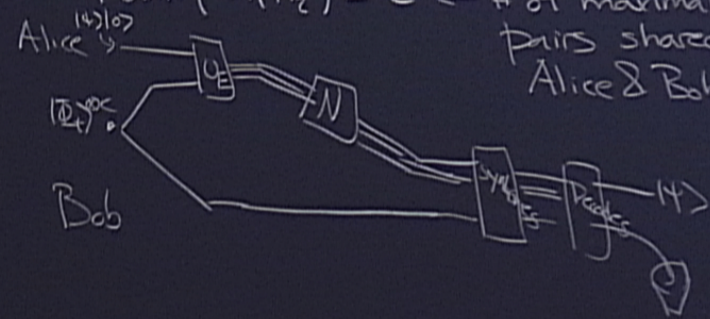
① CSS codes & QKD/QKE

2 classical binary codes: parity-check matrices H_1 & H_2.

A CSS code: – construct X stabilizer gens from H_1

" Z " " " H_2

$\text{rank}(H_1 H_2^T) = c \leftarrow$ # of maximally-entangled pairs shared between Alice & Bob.



$g_{n+c+j} \to g_{n+c+j} \otimes Z_j$

BB84 $\quad \overset{H_1}{[n,k_1]} \quad \overset{H_2}{[n,k_2]}$

$\underline{H}_1 \to \underline{T}_1 \underline{H}_1$

$\underline{H}_2 \to \underline{T}_2 \underline{H}_2$

s.t. $\underline{T}_1 \underline{H}_1 \underline{H}_2^T \underline{T}_2^T = \begin{array}{c} \begin{pmatrix} 0 & 0 \\ 0 & I_{cx} \end{pmatrix} \begin{array}{l} \}\, n-k_1-c \\ \}\, c \end{array} \\ \begin{array}{cc} \underbrace{\quad}_{n-k_2-c} & \underbrace{\quad}_{c} \end{array} \end{array}$

$\underline{T}_1 \underline{H}_1 \to (T$

QEC ... stabilizer generators

$g_1, \quad , g_{n-k}, h_1, \quad , c \qquad n-k+c$

$[g_i, g_j] = 0$

symplectic partners $\{g_i, h_j\} = 0$ if $i = j - n-k-c$

$[g_i, h_j] = 0$ otherwise

$[h_i, h_j] = 0$

For $j = 1, \dots c$, replace

$h_j \to h_j \otimes X_j$

$g_{n-k-c+j} \to g_{n-k-c+j} \otimes Z_j$

Bob's side

BB84 $\quad \overset{H_1}{[n,k_1]} \quad \overset{H_2}{[n,k_2]}$

$$\underline{H}_1 \to \underline{T}_1 \underline{H}_1, \quad \underline{H}_2 \to \underline{T}_2 \underline{H}_2 \quad \text{s.t.} \quad \underline{T}_1 \underline{H}_1 \underline{H}_2^T \underline{T}_2^T = \begin{array}{c} \begin{pmatrix} 0 & 0 \\ 0 & I_{c\times c} \end{pmatrix} \begin{array}{l} \}\, n-k_1-c \\ \}\, c \end{array} \\ \quad n+k_2-c \quad\; c \end{array}$$

$$\underline{T}_1 \underline{H}_1 \to \underbrace{(\underline{T}_1 \underline{H}_1 \;\; \underline{I}_{c\times c})}_{n+c} \}\, n-k_1$$

$$\underline{T}_2 \underline{H}_2 \to \underbrace{(\underline{T}_2 \underline{H}_2 \;\; \underline{I}_{c\times c})}_{n+c} \}\, n-k_2$$

QEC: stabilizer generators

$g_1, \ldots, g_{n-k}, h_1, \ldots, h_c \qquad n-k+c$

$$[g_i, g_j] = 0$$

symplectic partners $\{g_i, h_j\} = 0$ if $i = j + n-k-c$

$$[g_i, h_j] = 0 \quad \text{otherwise}$$

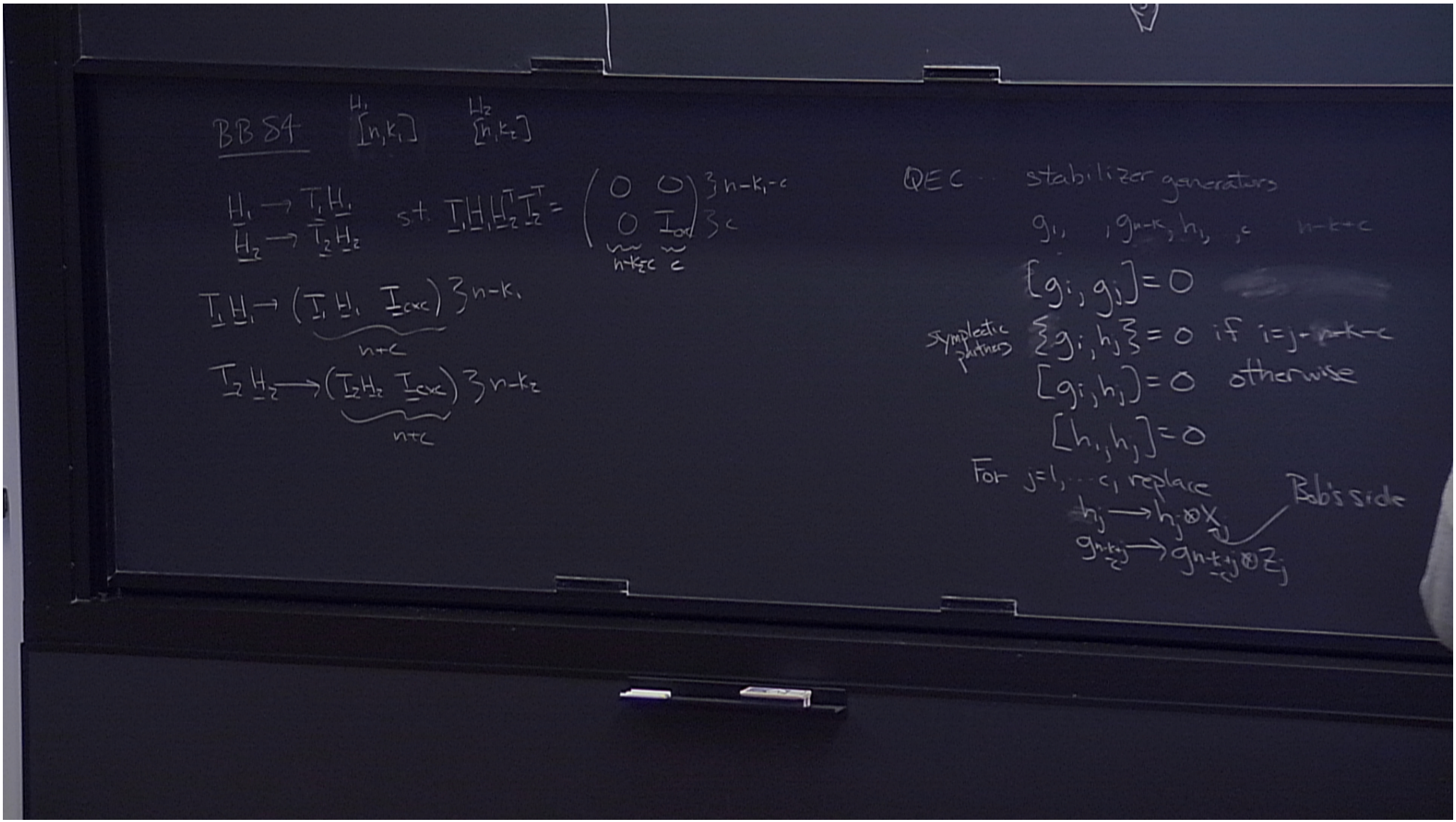
$$[h_i, h_j] = 0$$

For $j = 1, \ldots, c$, replace

$$h_j \to h_j \otimes X_j$$

$$g_{n-k+j} \to g_{n-k+j} \otimes Z_j$$

Bob's side


BB84
H1 [n,k1]
H2 [n,k2]
H1 → T1 H1
H2 → T2 H2
s.t. T1 H1 H2^T T2^T = (0 0; 0 Icxc) } n-k1-c, } c
n-k2-c c
T1 H1 → (T1 H1 Icxc) } n-k1
n+c
T2 H2 → (T2 H2 Icxc) } n-k2
n+c
QEC ... stabilizer generators
g1, , gn-k, h1, , c n-k+c
[gi, gj] = 0
symplectic partners {gi, hj} = 0 if i=j-n-k-c
[gi, hj] = 0 otherwise
[hi, hj] = 0
For j=1, ... c, replace
hj → hj ⊗ Xj
gn-k-c+j → gn-k-c+j ⊗ Zj
Bob's side

BB84 $\quad \overset{H_1}{[n,k_1]} \quad \overset{H_2}{[n,k_2]}$

$$\underline{H}_1 \to \underline{T}_1\underline{H}_1, \quad \underline{H}_2 \to \underline{T}_2\underline{H}_2 \quad \text{s.t.} \quad \underline{T}_1\underline{H}_1\underline{H}_2^T\underline{T}_2^T = \begin{array}{c} \begin{pmatrix} 0 & 0 \\ 0 & I_{c\times c} \end{pmatrix} \begin{array}{l} \}\, n-k_1-c \\ \}\, c \end{array} \\ \quad\; n+k_2-c \quad\; c \end{array}$$

$$\underline{T}_1\underline{H}_1 \to \underbrace{(\underline{T}_1\underline{H}_1 \;\; \underline{I}_{c\times c})}_{n+c} \}\, n-k_1 = \underline{H}_1'$$

$$\underline{T}_2\underline{H}_2 \to \underbrace{(\underline{T}_2\underline{H}_2 \;\; \underline{I}_{c\times c})}_{n+c} \}\, n-k_2 = \underline{H}_2'$$

QEC: stabilizer generators

$g_1, \ldots, g_{n-k}, h_1, \ldots, h_c \qquad n-k+c$

$$[g_i, g_j] = 0$$

symplectic partners $\{g_i, h_j\} = 0$ if $i = j + n-k-c$

$$[g_i, h_j] = 0 \quad \text{otherwise}$$

$$[h_i, h_j] = 0$$

For $j = 1, \ldots, c$, replace

$$h_j \to h_j \otimes X_j$$

$$g_{n-k+j} \to g_{n-k+j} \otimes Z_j$$

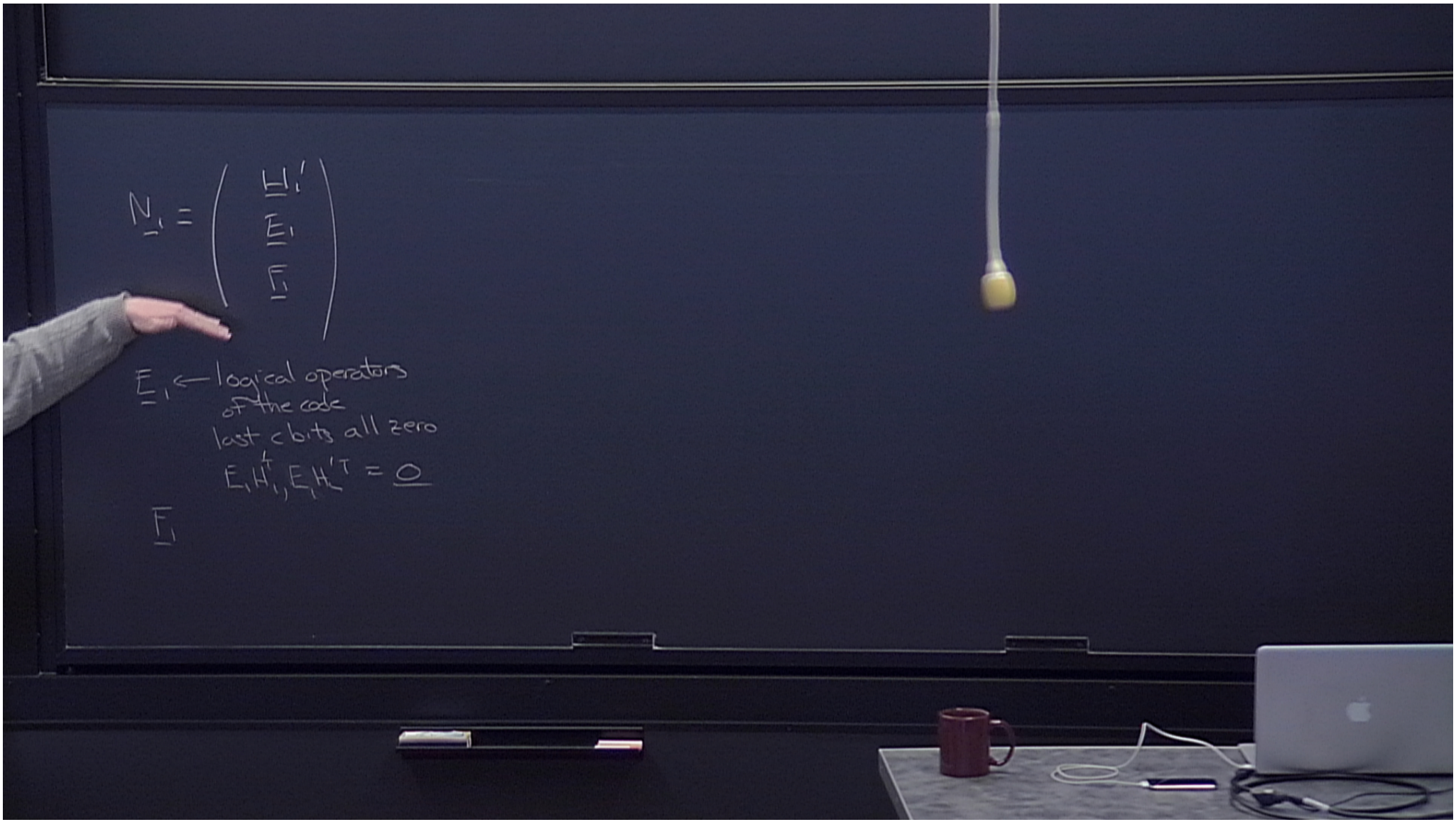
Bob's side

$$\underline{N}_1 = \begin{pmatrix} \underline{H}_1' \\ \underline{E}_1 \\ \underline{F}_1 \end{pmatrix}$$

$\underline{E}_1 \leftarrow$ logical operators of the code

last c bits all zero

E_1


$\underline{N}_1 = \begin{pmatrix} \underline{H}_1' \\ \underline{E}_1 \\ \underline{F}_1 \end{pmatrix}$
$\underline{E}_1 \leftarrow$ logical operators of the code
last c bits all zero
$E_1 H_1'^T, E_1 H_2'^T = \underline{0}$
$\underline{F}_1$

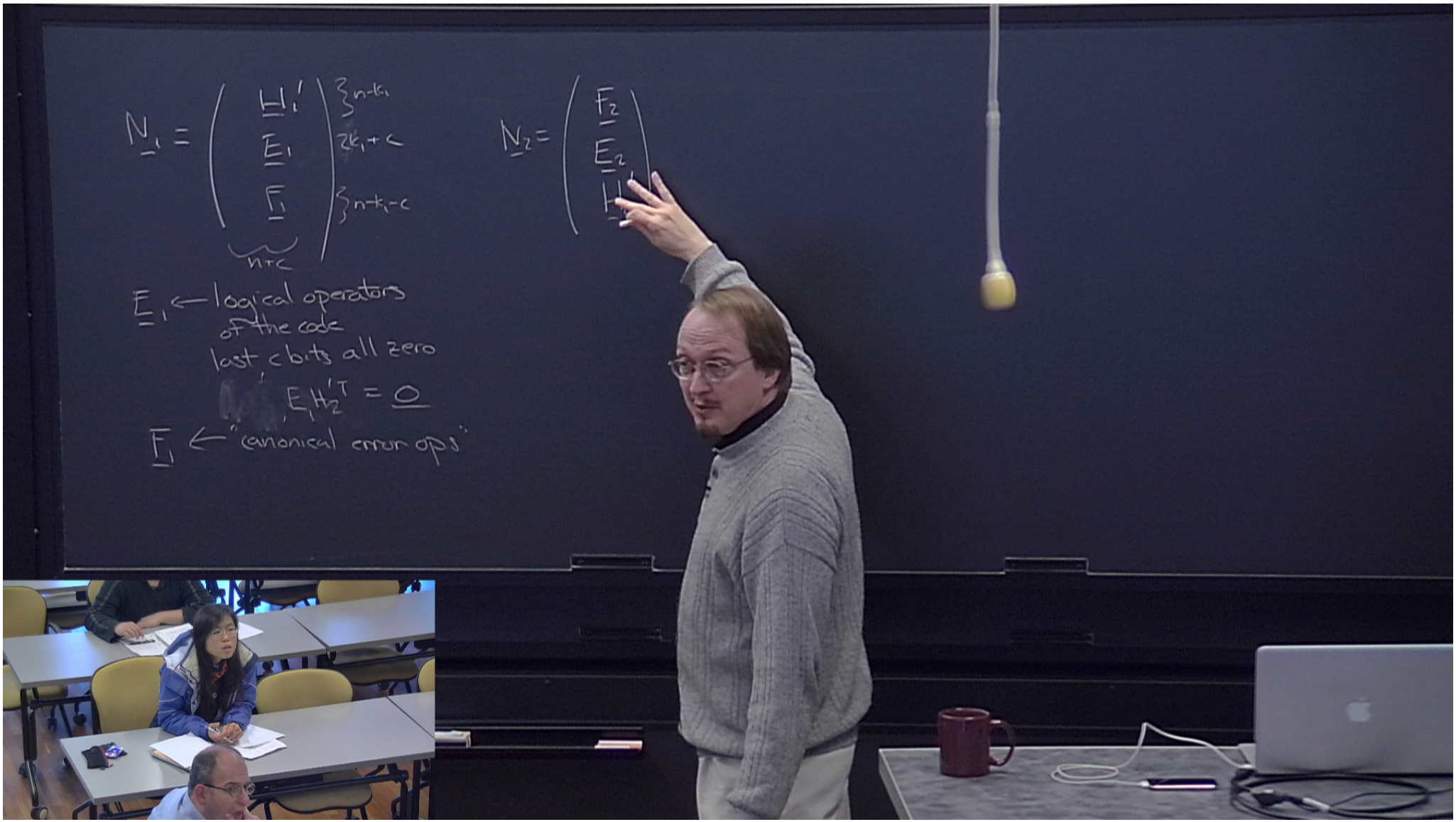
$$\underline{N}_1 = \begin{pmatrix} \underline{H}_1' \\ \underline{E}_1 \\ \underline{F}_1 \end{pmatrix} \begin{matrix} \}\, n-k_1 \\ 2k_1+c \\ \}\, n-k_1-c \end{matrix}$$

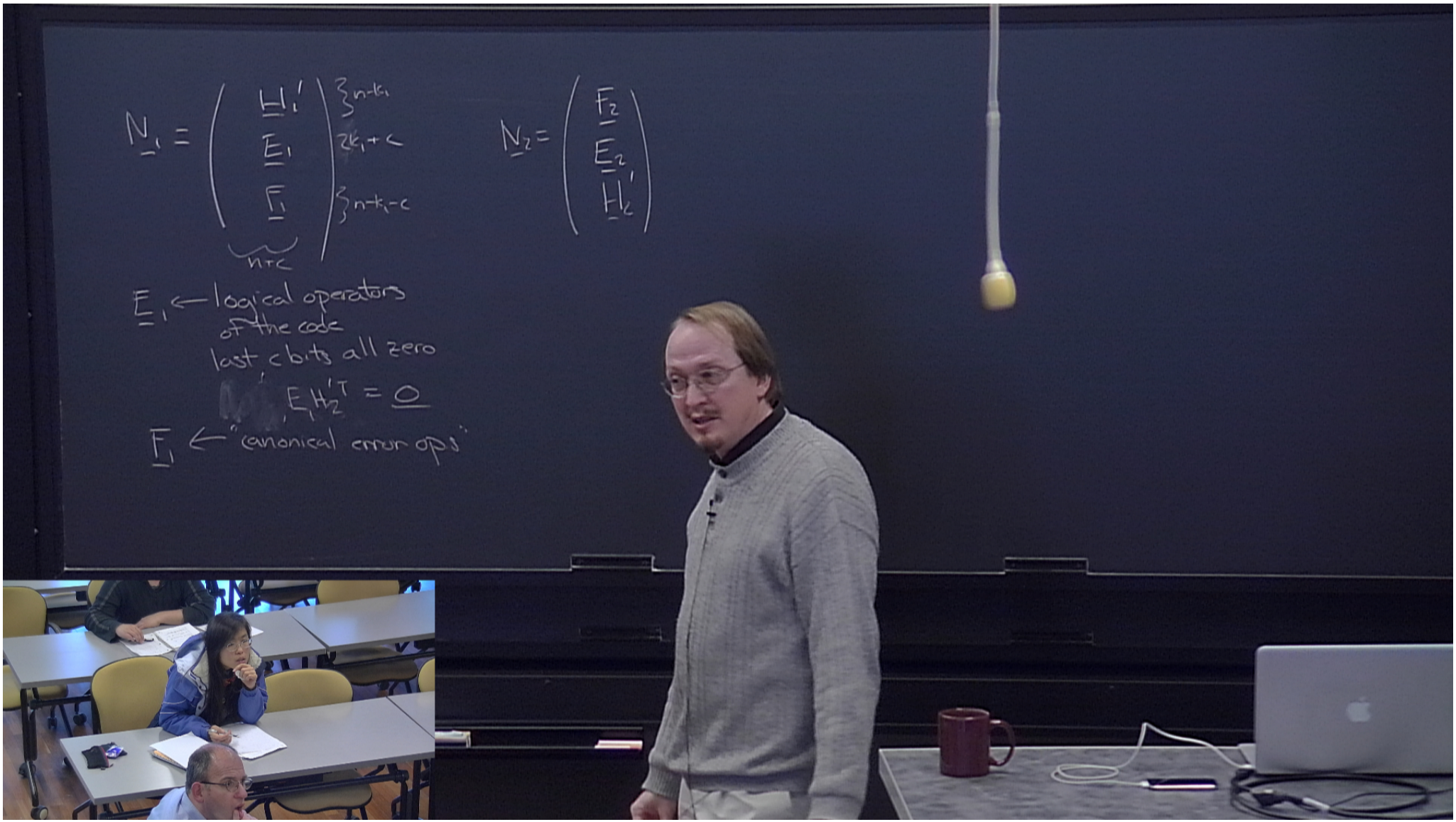
$\underline{E}_1 \leftarrow$ logical operators of the code

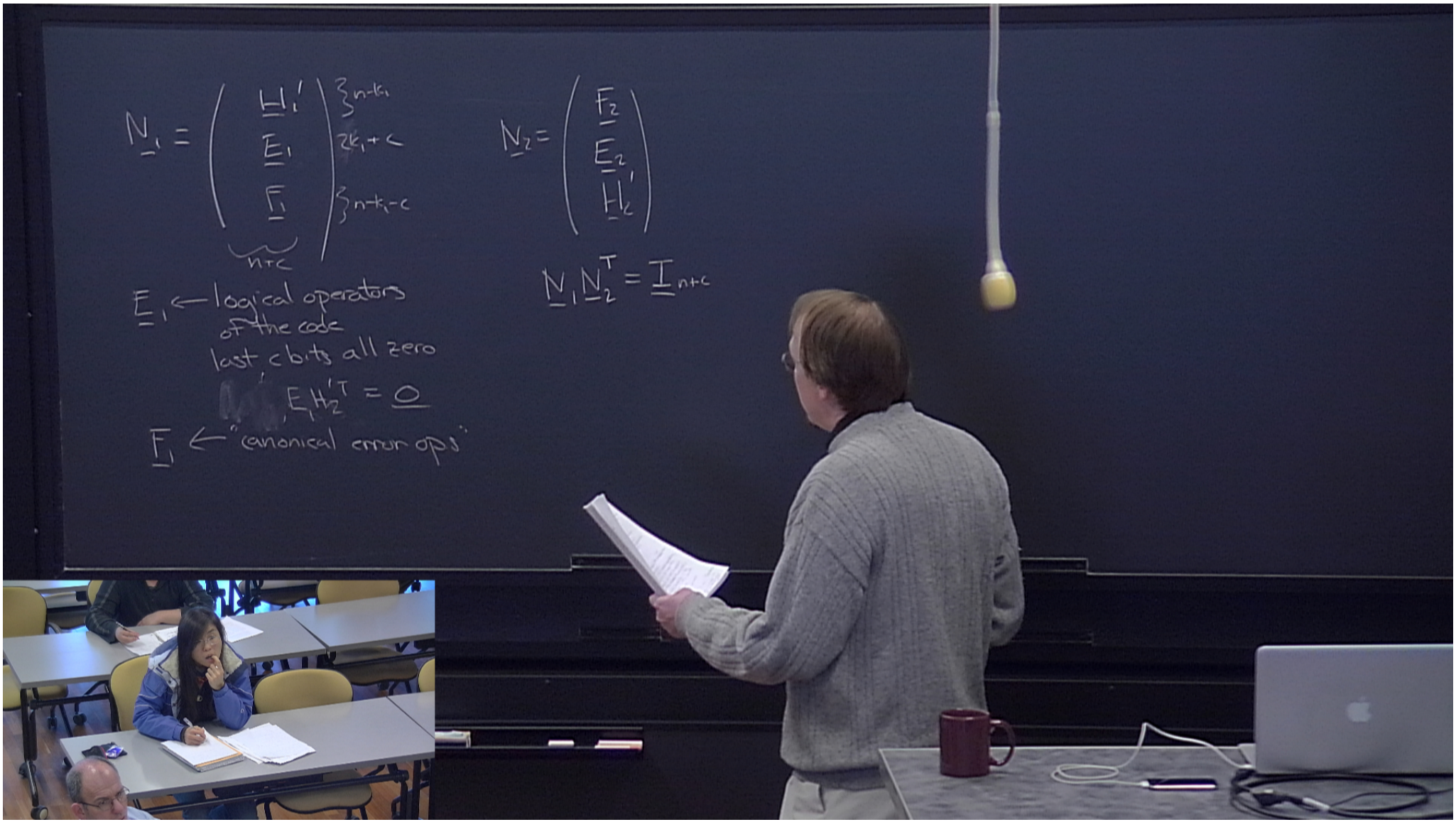
last c bits all zero

$$\underline{E}_1 \underline{H}_1'^T, \underline{E}_1 \underline{H}_2'^T = \underline{0}$$

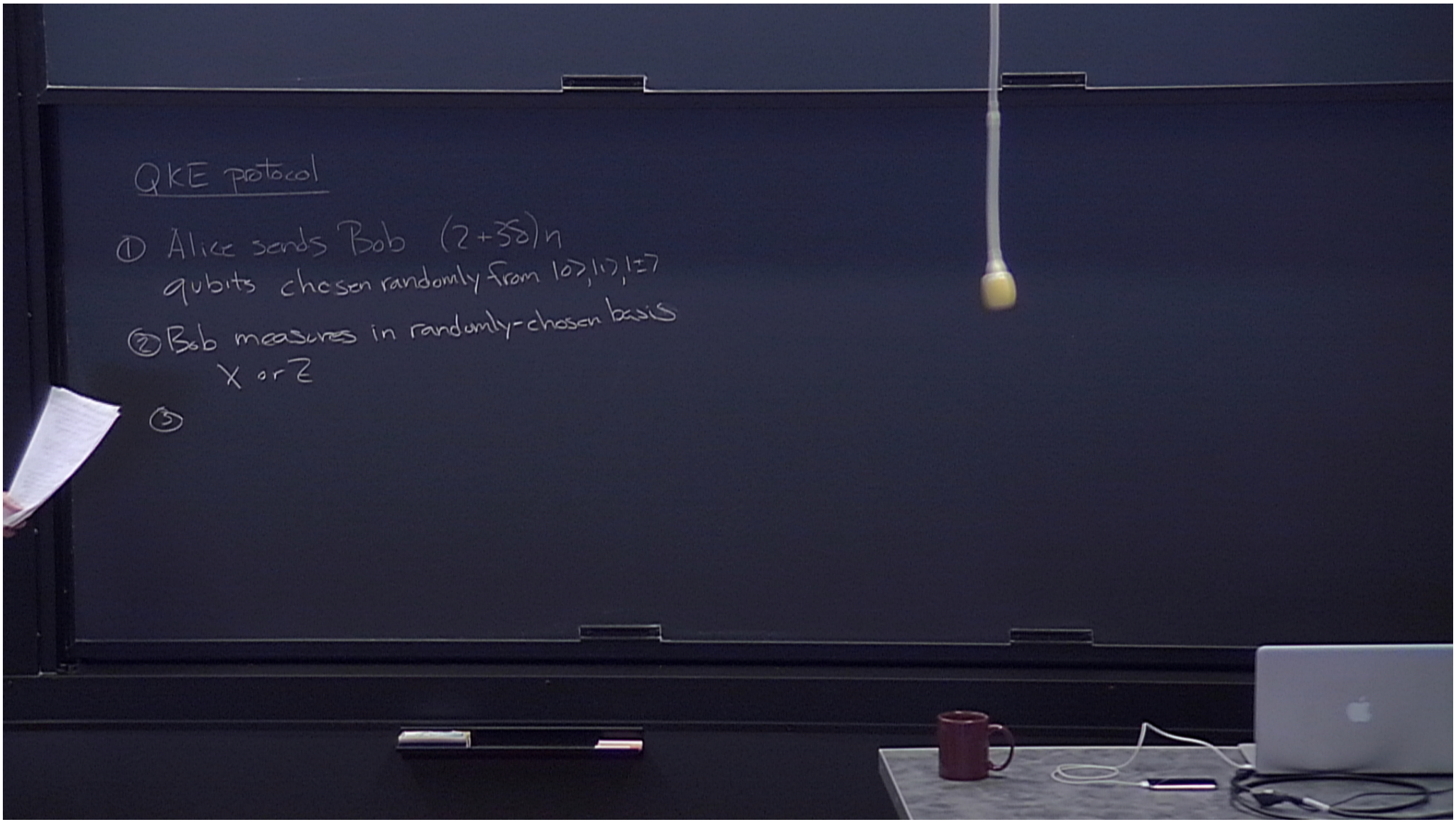
$\underline{F}_1 \leftarrow$ "canonical error ops"

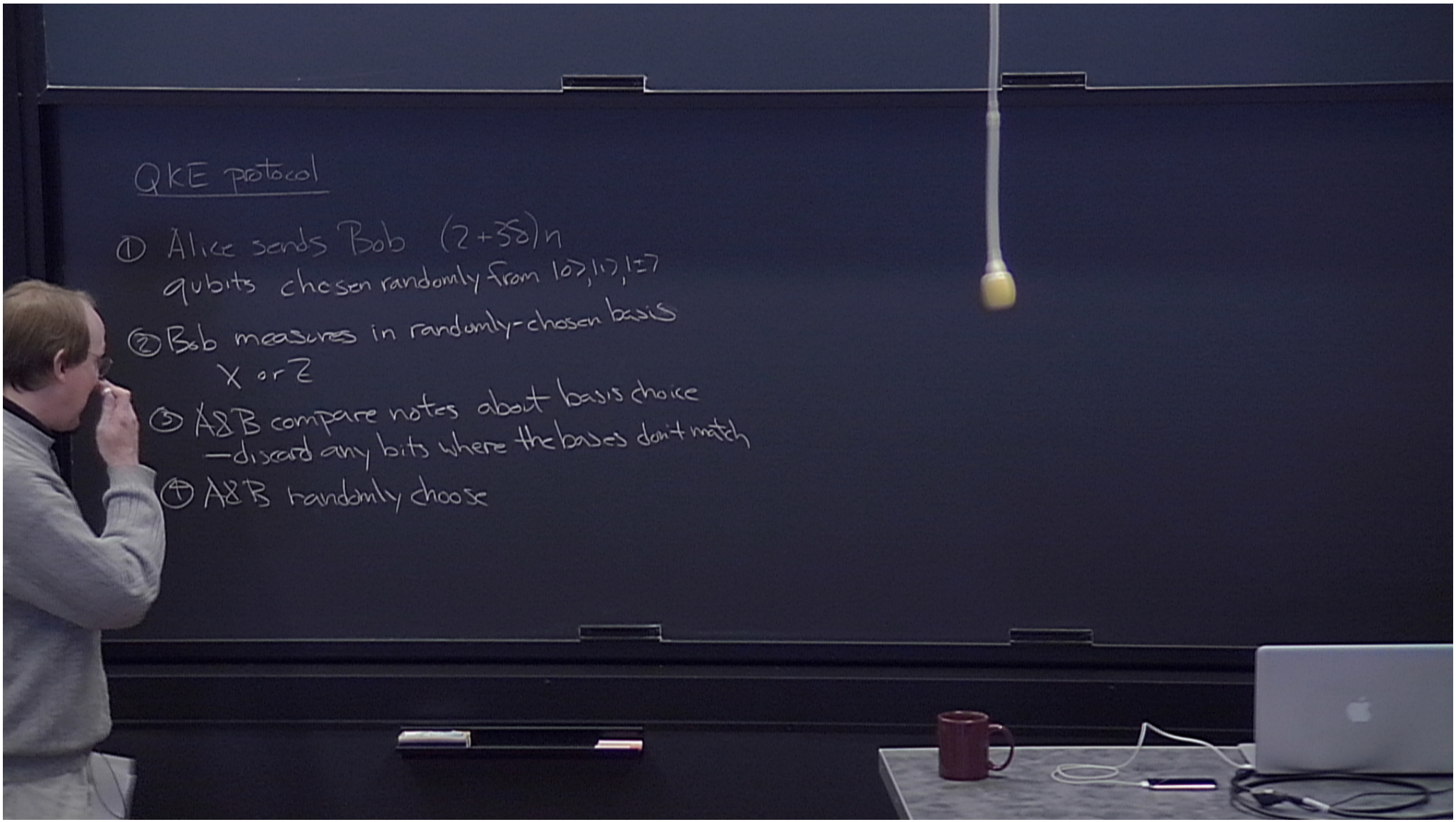

$N_1 = \begin{pmatrix} H_1' \\ E_1 \\ F_1 \end{pmatrix}$
$n-k_1$
$2k_1+c$
$n-k_1-c$
$n+c$
$N_2 = \begin{pmatrix} F_2 \\ E_2 \\ H_2' \end{pmatrix}$
$E_1 \leftarrow$ logical operators of the code
last c bits all zero
$E_1 H_2'^T = 0$
$F_1 \leftarrow$ "canonical error ops"

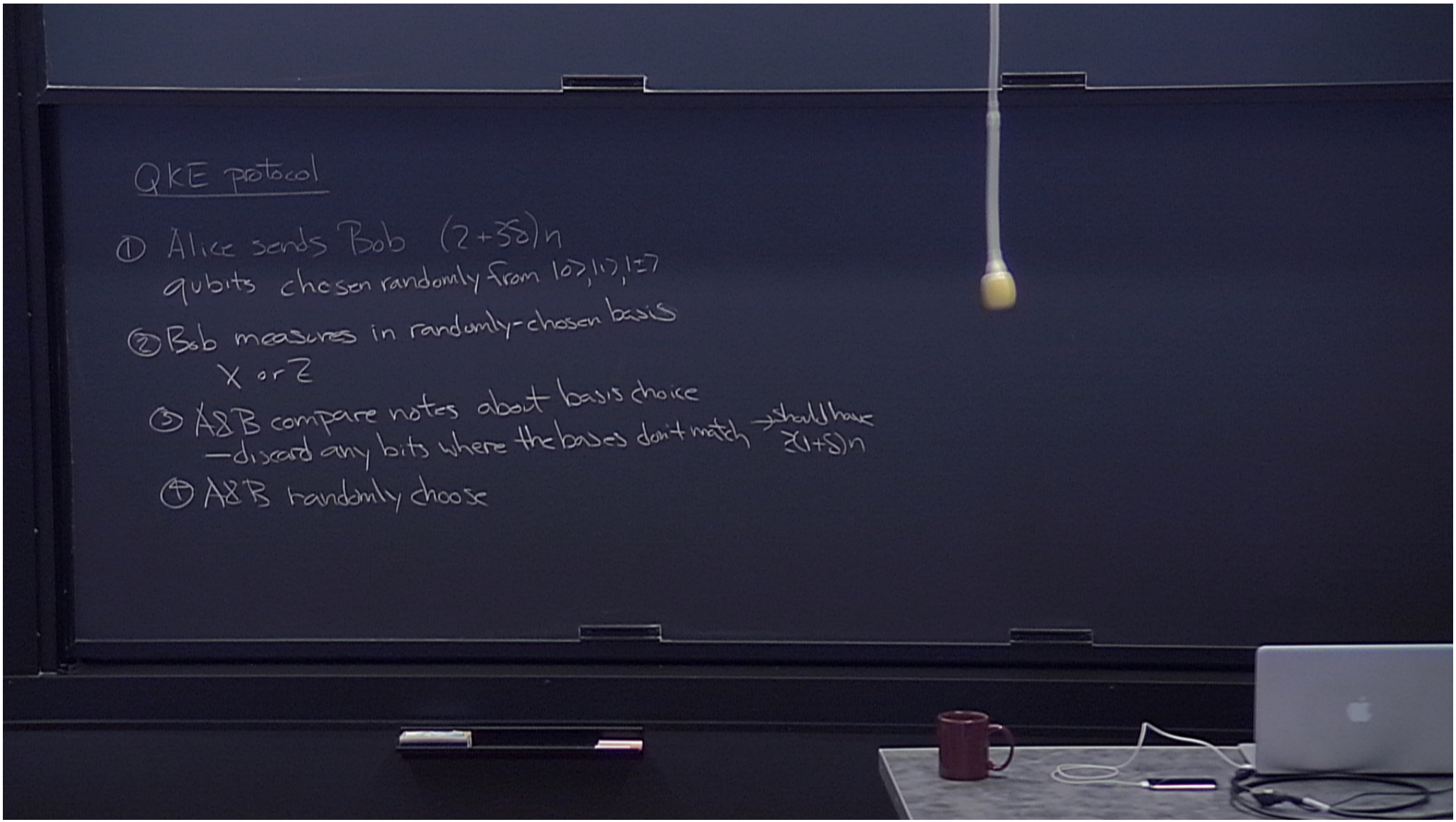

$N_1 = \begin{pmatrix} H_1' \\ E_1 \\ F_1 \end{pmatrix}$
$n-k_1$
$2k_1+c$
$n-k_1-c$
$n+c$
$N_2 = \begin{pmatrix} F_2 \\ E_2 \\ H_2' \end{pmatrix}$
$E_1 \leftarrow$ logical operators of the code
last c bits all zero
$E_1 H_2'^T = 0$
$F_1 \leftarrow$ "canonical error ops"

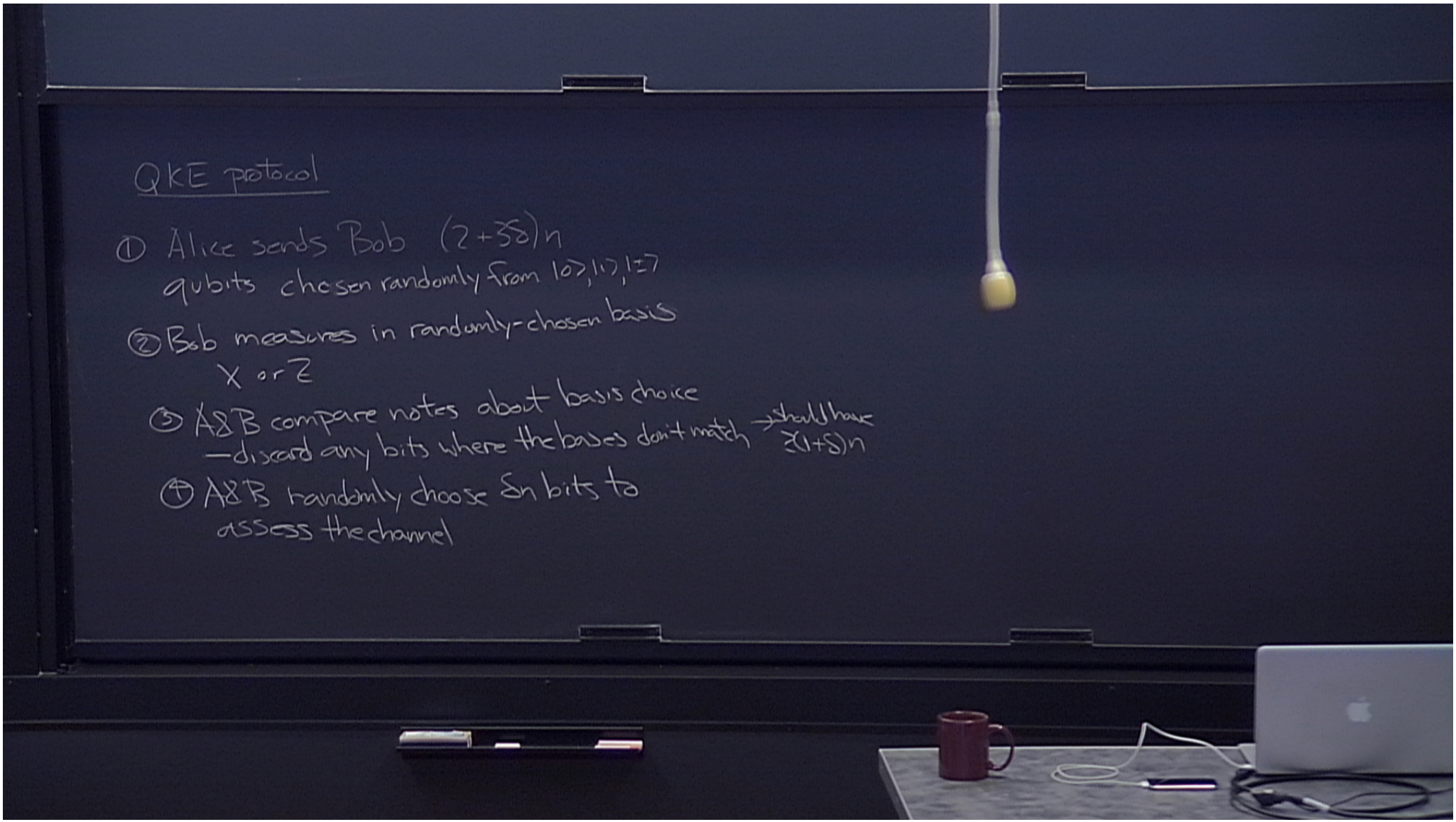

$\underline{N}_1 = \begin{pmatrix} \underline{H}_1' \\ \underline{E}_1 \\ \underline{F}_1 \end{pmatrix}$
$n-k_1$
$2k_1+c$
$n-k_1-c$
$n+c$
$\underline{N}_2 = \begin{pmatrix} \underline{F}_2 \\ \underline{E}_2 \\ \underline{H}_2' \end{pmatrix}$
$\underline{N}_1 \underline{N}_2^T = \underline{I}_{n+c}$
$\underline{E}_1 \leftarrow$ logical operators of the code
last c bits all zero
$\underline{E}_1 \underline{H}_2'^T = \underline{0}$
$\underline{F}_1 \leftarrow$ "canonical error ops"

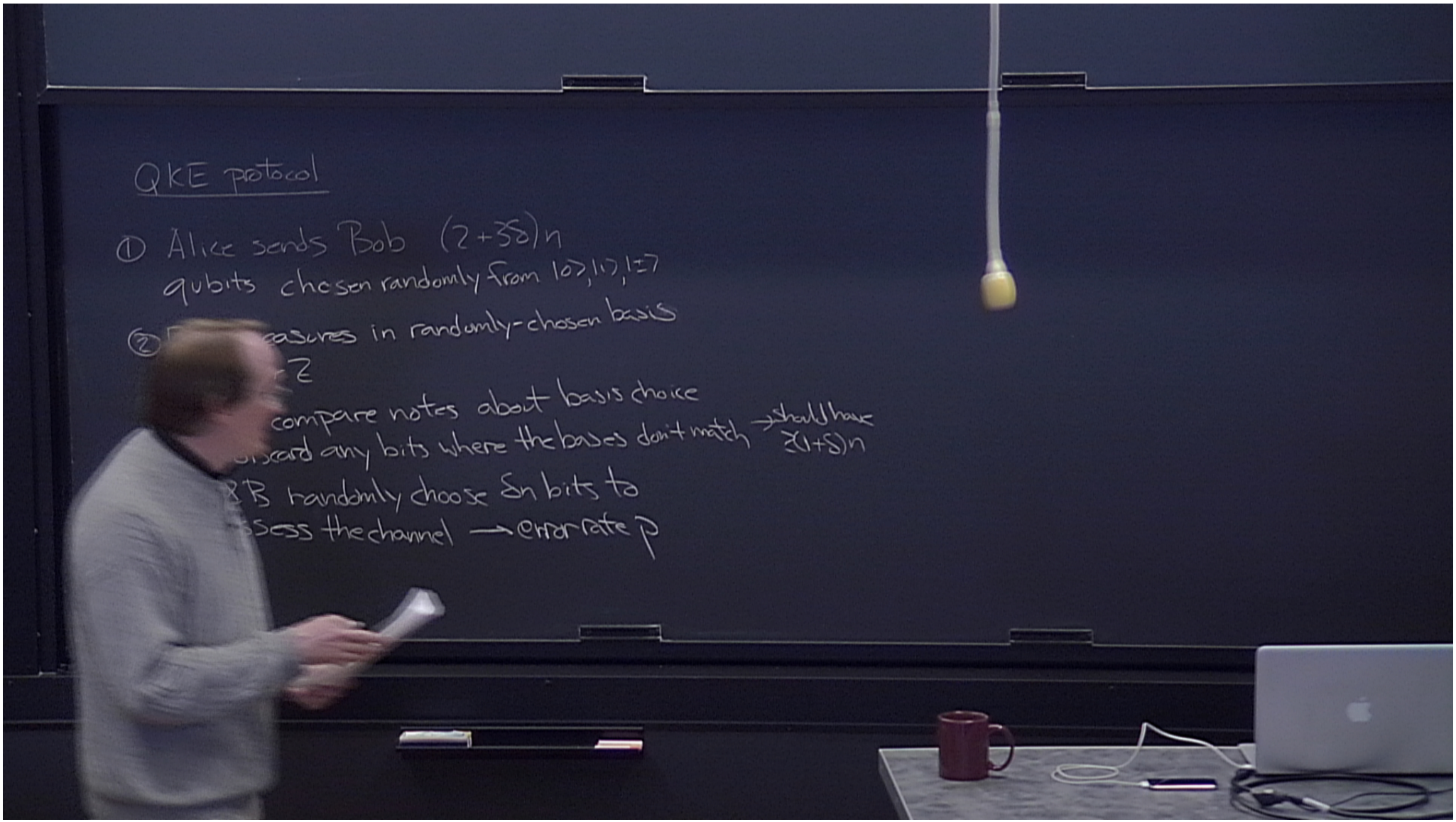
QKE protocol
1

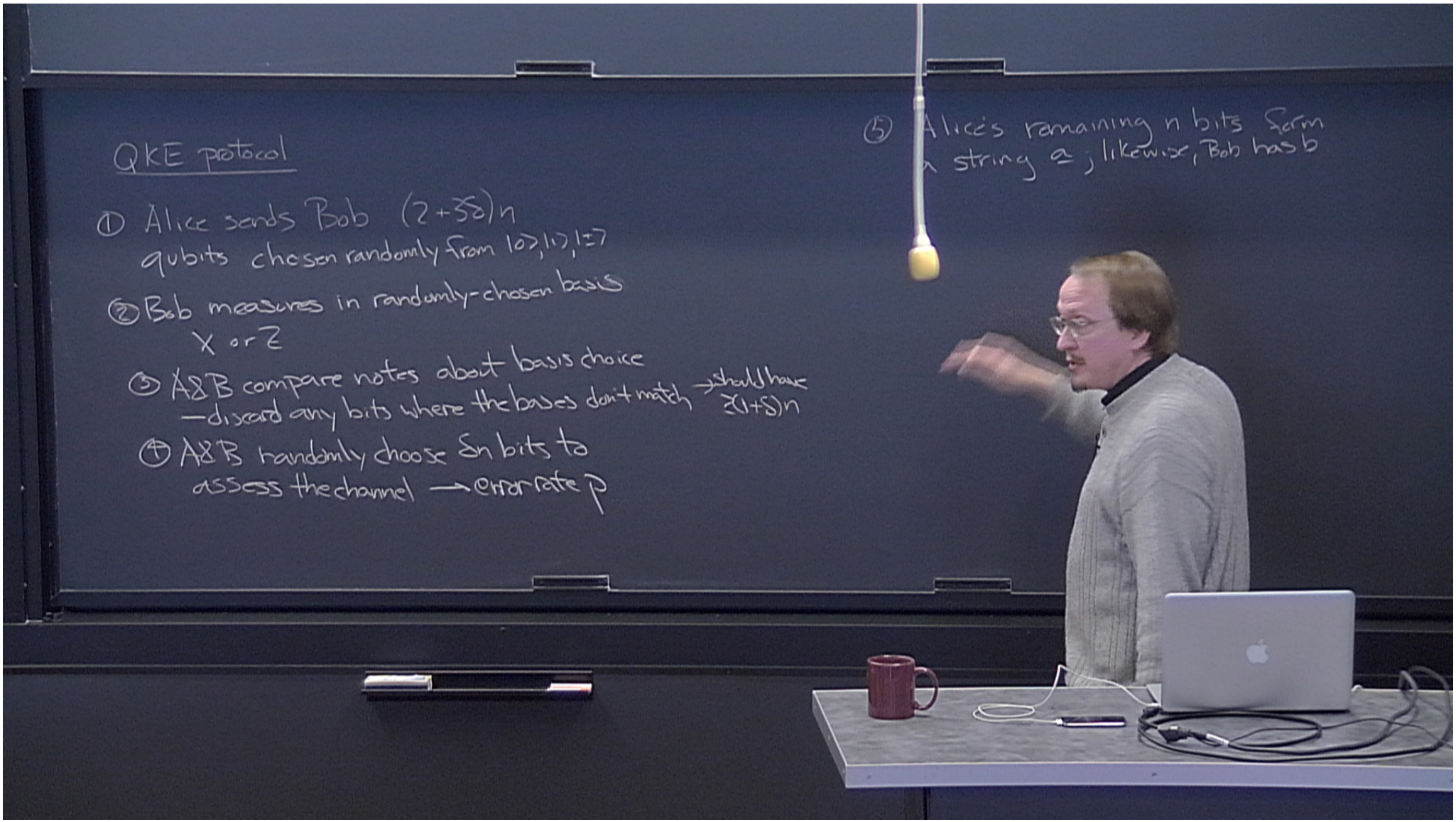

QKE protocol
① Alice sends Bob (2+δ)n
qubits chosen randomly from |0>, |1>, |±>
② Bob measures in randomly-chosen basis
X or Z
③

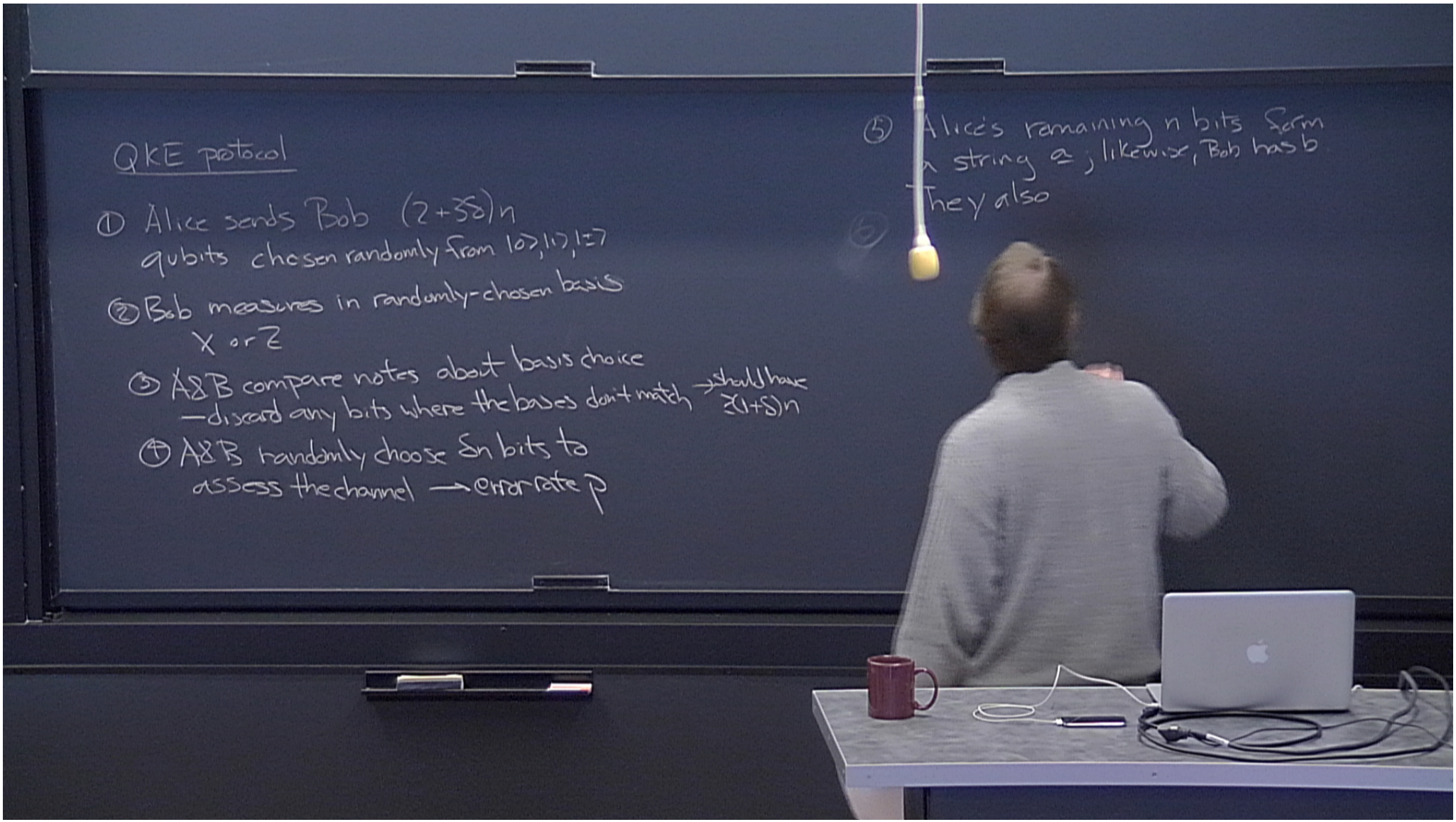

QKE protocol
① Alice sends Bob (2+δ)n
qubits chosen randomly from |0>, |1>, |±>
② Bob measures in randomly-chosen basis
X or Z
③ A&B compare notes about basis choice
—discard any bits where the bases don't match
④ A&B randomly choose

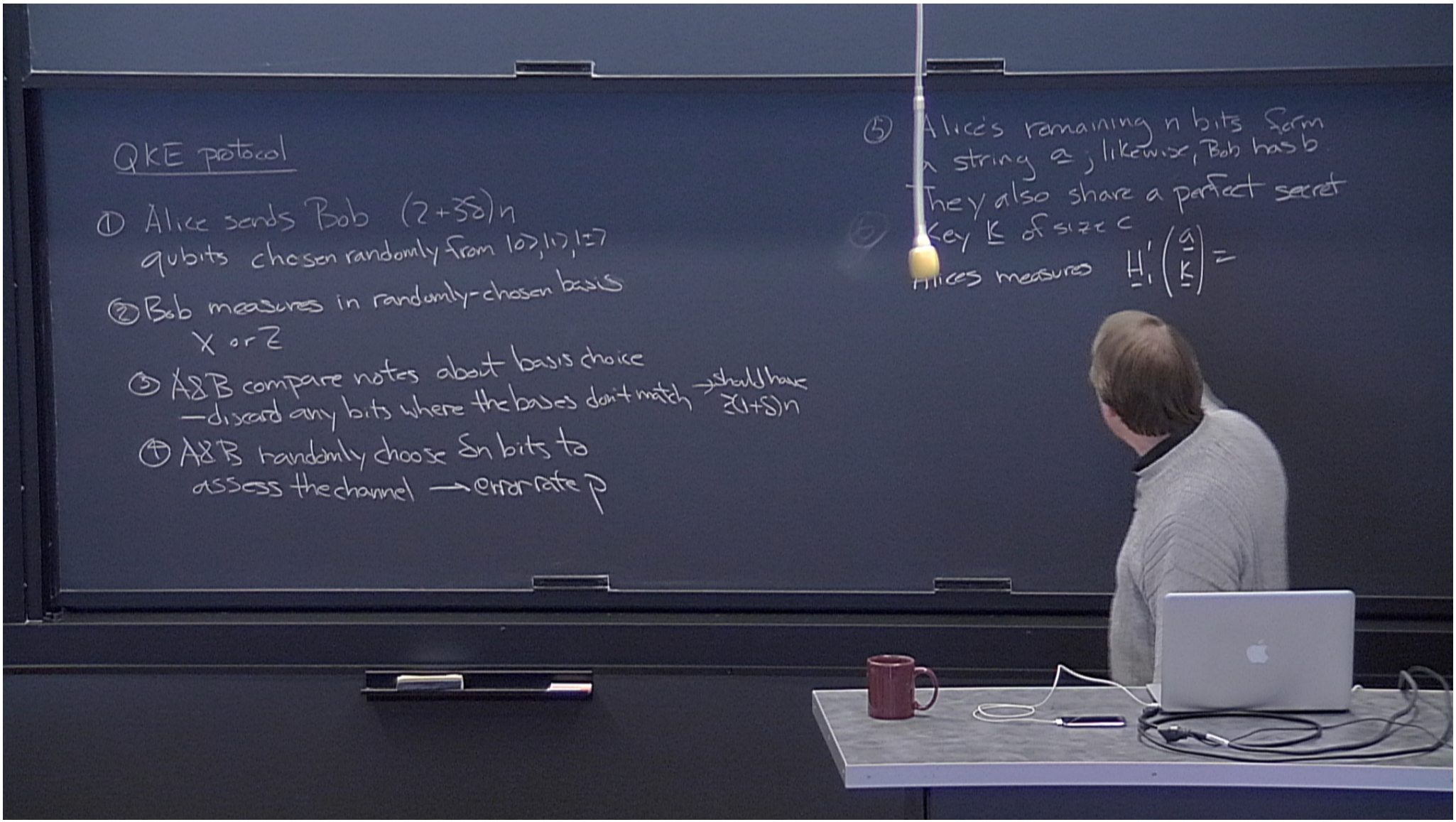

QKE protocol
① Alice sends Bob (2+3δ)n
qubits chosen randomly from |0>, |1>, |±>
② Bob measures in randomly-chosen basis
X or Z
③ A&B compare notes about basis choice
—discard any bits where the bases don't match → should have
2(1+δ)n
④ A&B randomly choose

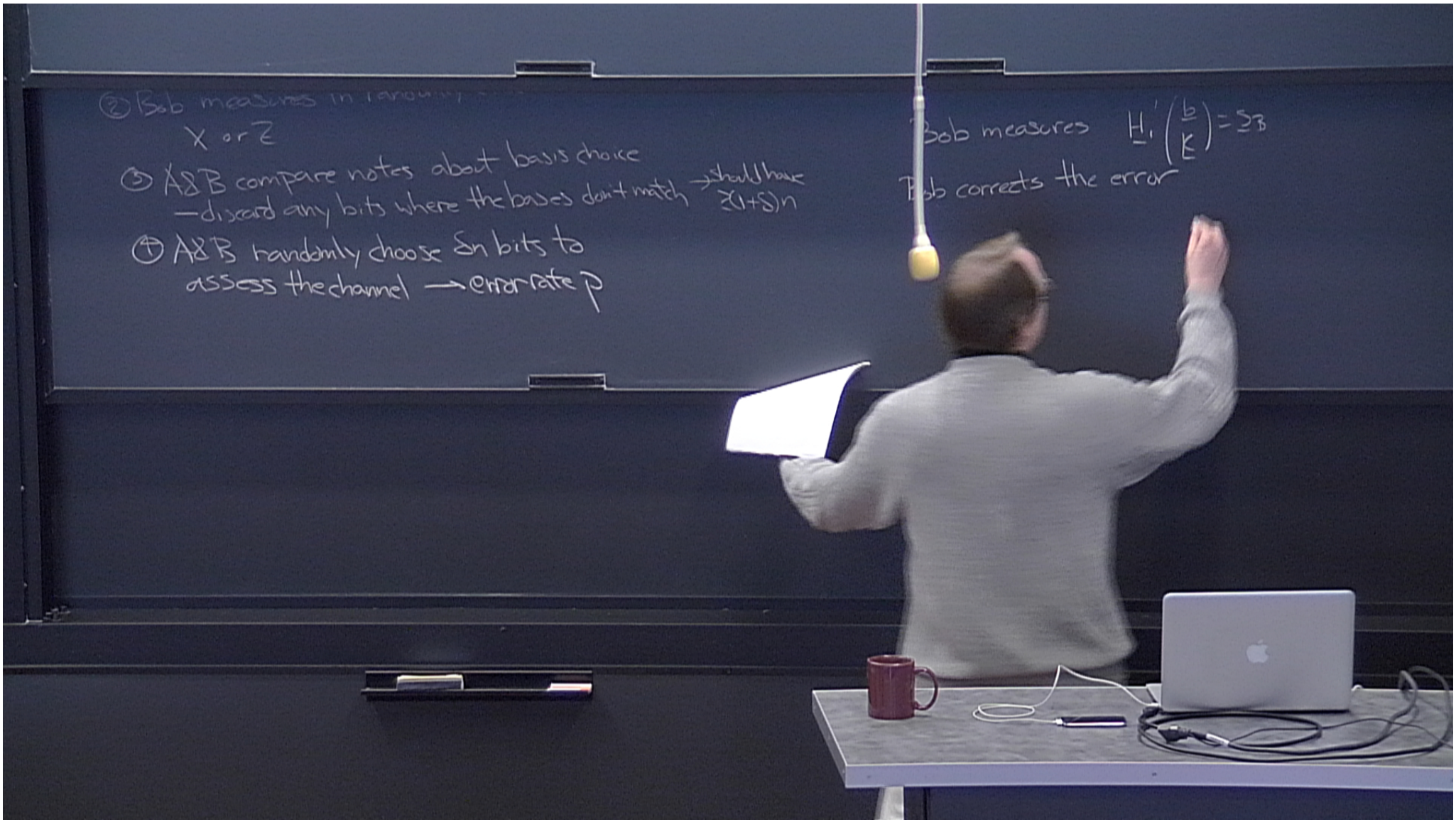

QKE protocol
① Alice sends Bob (2+3δ)n
qubits chosen randomly from |0>, |1>, |±>
② Bob measures in randomly-chosen basis
X or Z
③ A&B compare notes about basis choice
—discard any bits where the bases don't match → should have ≥(1+δ)n
④ A&B randomly choose δn bits to
assess the channel

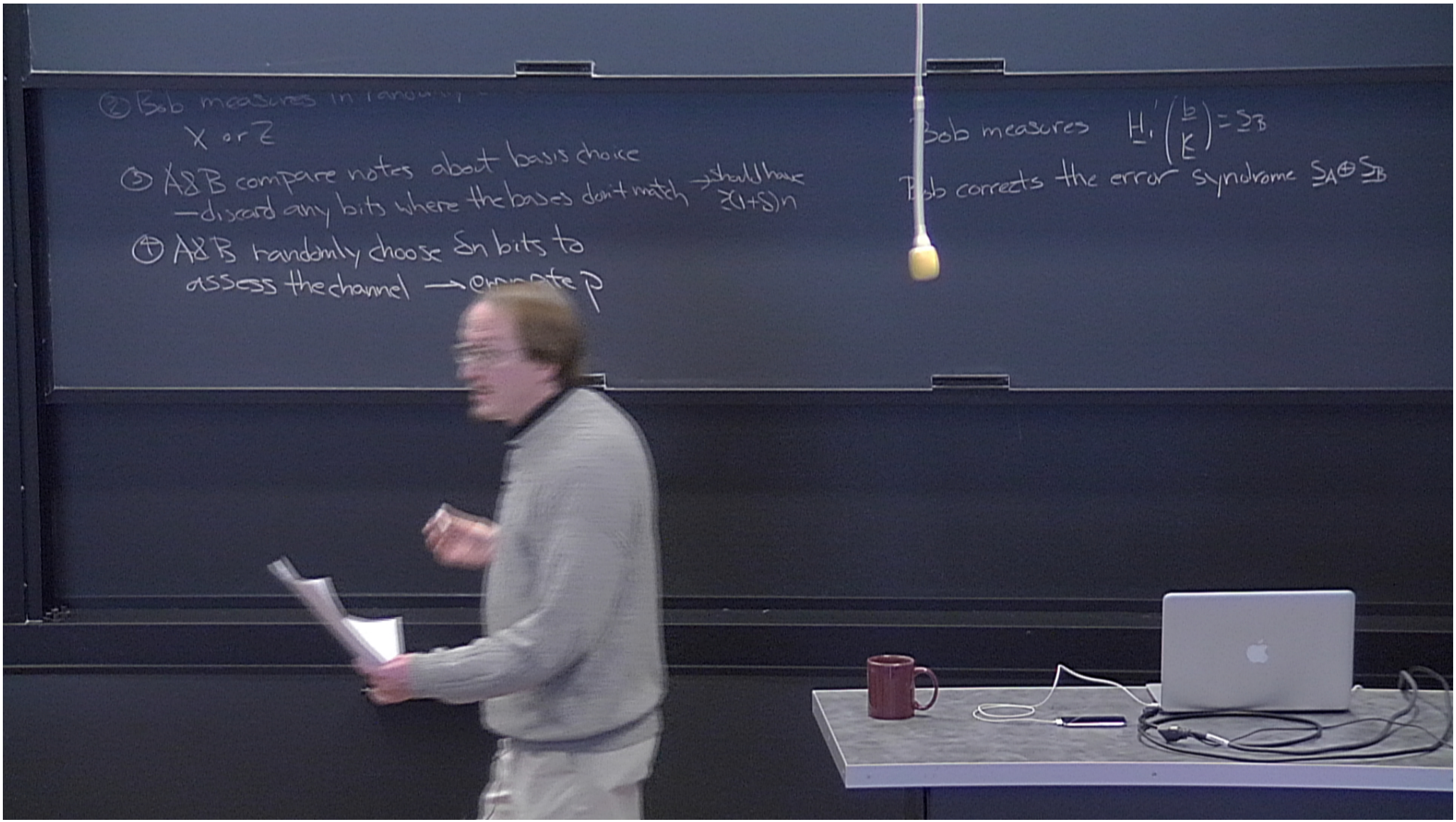

QKE protocol
① Alice sends Bob (2+3δ)n
qubits chosen randomly from |0>,|1>,|±>
② measures in randomly-chosen basis
compare notes about basis choice
discard any bits where the bases don't match → should have
(1+δ)n
randomly choose δn bits to
assess the channel → error rate p

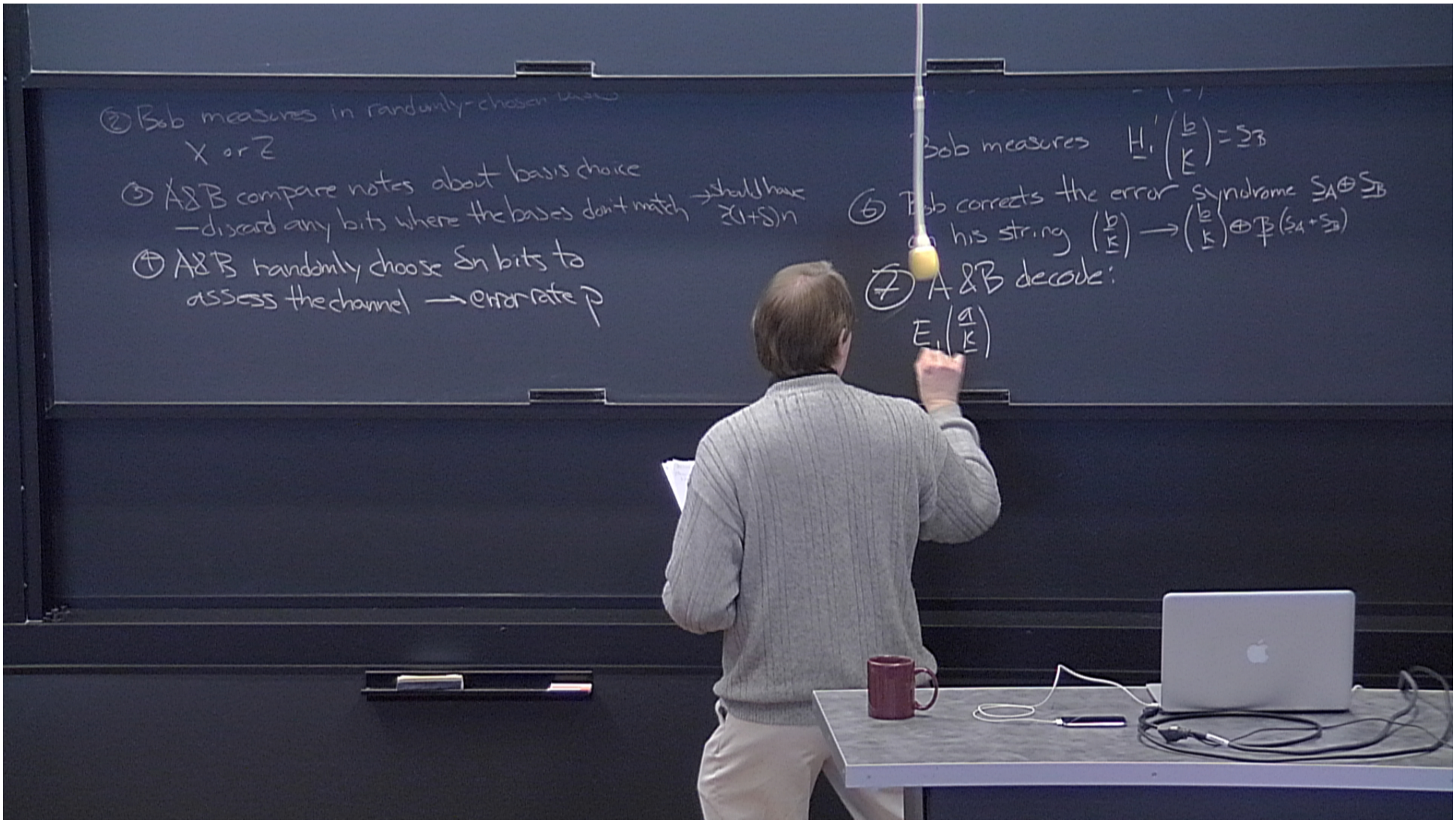

QKE protocol
① Alice sends Bob (2+3δ)n
qubits chosen randomly from |0⟩, |1⟩, |±⟩
② Bob measures in randomly-chosen basis
X or Z
③ A&B compare notes about basis choice
—discard any bits where the bases don't match → should have ≈(1+δ)n
④ A&B randomly choose δn bits to
assess the channel → error rate p
⑤ Alice's remaining n bits form
a string a; likewise, Bob has b


QKE protocol
① Alice sends Bob (2+3δ)n
qubits chosen randomly from |0⟩,|1⟩,|±⟩
② Bob measures in randomly-chosen basis
X or Z
③ A&B compare notes about basis choice
—discard any bits where the bases don't match → should have ≥(1+δ)n
④ A&B randomly choose δn bits to
assess the channel → error rate p
⑤ Alice's remaining n bits form
a string a; likewise, Bob has b
They also


QKE protocol
① Alice sends Bob (2+5δ)n
qubits chosen randomly from |0⟩,|1⟩,|±⟩
② Bob measures in randomly-chosen basis
X or Z
③ A&B compare notes about basis choice
—discard any bits where the bases don't match → should have ≥(1+δ)n
④ A&B randomly choose δn bits to
assess the channel → error rate p
⑤ Alice's remaining n bits form
a string a; likewise, Bob has b
They also share a perfect secret
key k of size c
Alices measures


② Bob measures in random
X or Z
③ A&B compare notes about basis choice
—discard any bits where the bases don't match → should have ½(1+δ)n
④ A&B randomly choose δn bits to
assess the channel → error rate p
Bob measures
Bob corrects the error


② Bob measures in randomly
X or Z
③ A&B compare notes about basis choice
—discard any bits where the bases don't match → should have ≈(1+δ)n
④ A&B randomly choose δn bits to
assess the channel → compute p
Bob measures $H_1'\begin{pmatrix} b \\ k \end{pmatrix} = S_B$
Bob corrects the error syndrome $S_A \oplus S_B$


② Bob measures in randomly-chosen basis
X or Z
③ A&B compare notes about basis choice
—discard any bits where the bases don't match → should have ≥(1+δ)n
④ A&B randomly choose δn bits to
assess the channel → error rate p
Bob measures
⑥ Bob corrects the error syndrome
his string
⑦ A&B decode:

QKE protocol

① Alice sends Bob $(2+3\delta)n$ qubits chosen randomly from $|0\rangle, |1\rangle, |\pm\rangle$

② Bob measures in randomly-chosen basis X or Z

③ A&B compare notes about basis choice — discard any bits where the bases don't match → should have $\geq (1+\delta)n$

④ A&B randomly choose δn bits to assess the channel → error rate p

⑤ Alice's remaining n bits form a string $\underline{a}$; likewise, Bob has b. They also share a perfect secret Key $\underline{k}$ of size c

⑥ Alices measures $\underline{H}_1'\begin{pmatrix}\underline{a}\\ \underline{k}\end{pmatrix} = \underline{s}_A \longrightarrow$ Bob

Bob measures $\underline{H}_1'\begin{pmatrix}\underline{b}\\ \underline{k}\end{pmatrix} = \underline{s}_B$

⑥ Bob corrects the error syndrome $\underline{s}_A \oplus \underline{s}_B$ on his string $\begin{pmatrix}\underline{b}\\ \underline{k}\end{pmatrix} \longrightarrow \begin{pmatrix}\underline{b}\\ \underline{k}\end{pmatrix} \oplus \underline{\beta}(\underline{s}_A + \underline{s}_B)$

⑦ A&B decode:

$\underline{E}_1\begin{pmatrix}\underline{a}\\ \underline{k}\end{pmatrix} \rightarrow \underline{k}_A$ $\quad \underline{E}_1\left(\begin{pmatrix}\underline{b}\\ \underline{k}\end{pmatrix} \oplus \underline{\beta}(\underline{s}_A \oplus \underline{s}_B)\right) = \underline{k}_B$

$$\underline{N}_1 = \begin{pmatrix}\underline{H}_1' \\ \underline{E}_1 \\ \underline{F}_1\end{pmatrix} \begin{matrix}\}\, n-k_1 \\ 2k_1+c \\ \}\, n-k_1-c\end{matrix} \qquad \underline{N}_2 = \begin{pmatrix}\underline{F}_2 \\ \underline{E}_2 \\ \underline{H}_2'\end{pmatrix}$$

$\underbrace{\qquad}_{n+c}$

$\underline{E}_1 \leftarrow$ logical operators

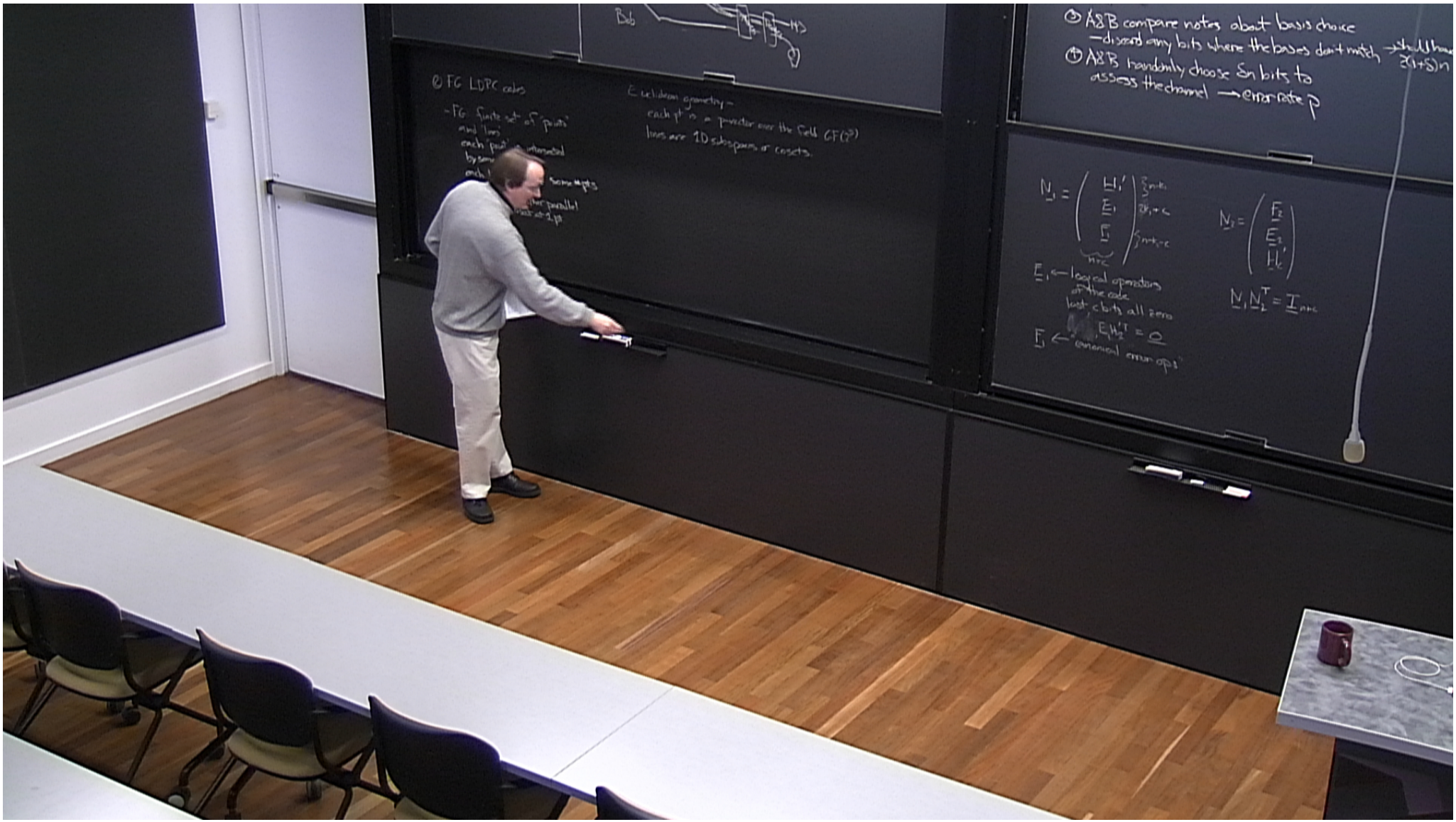
$\underline{N}_1 \underline{N}_2^T = \underline{I}_{n+c}$

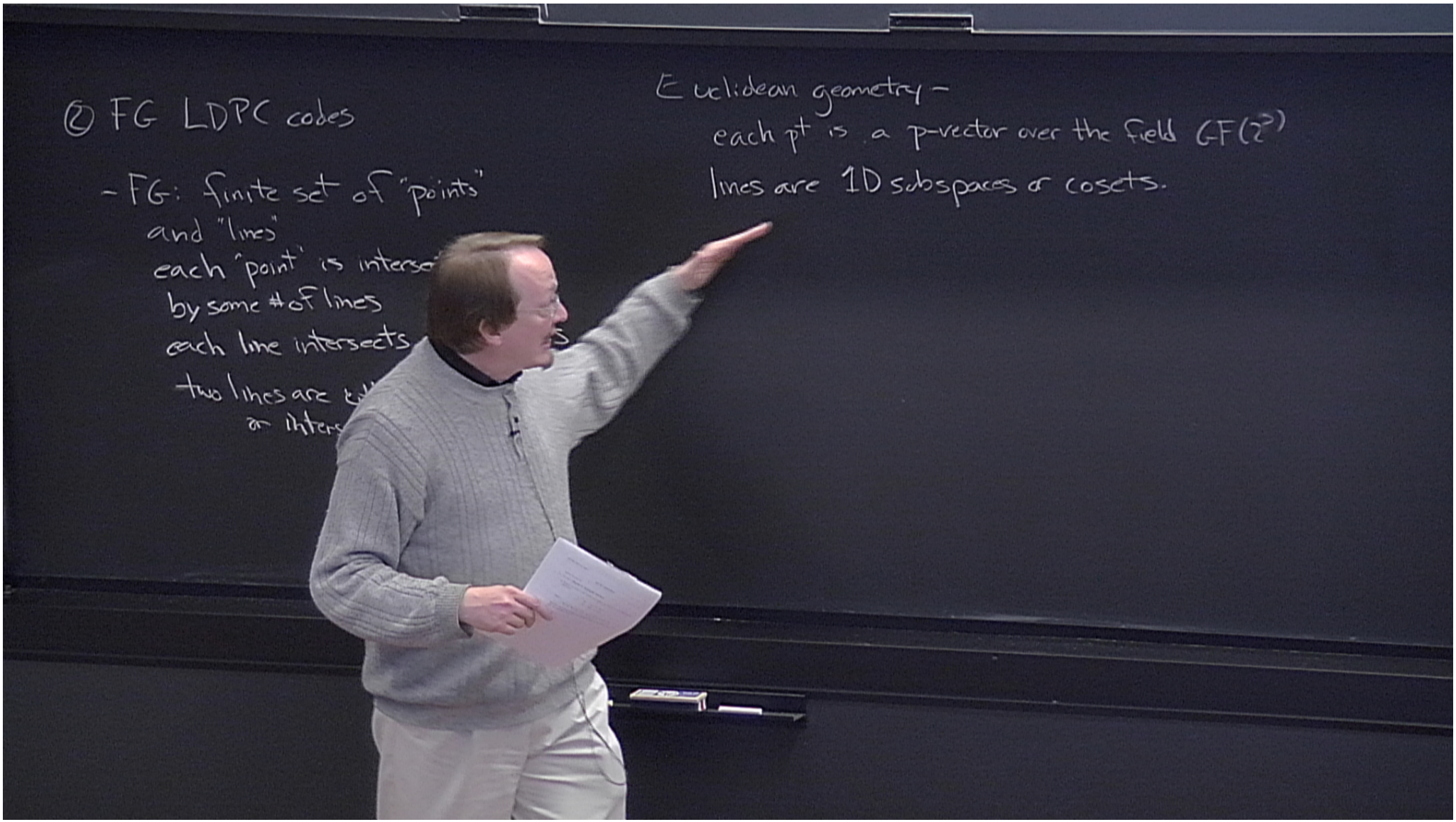
② FG LDPC codes

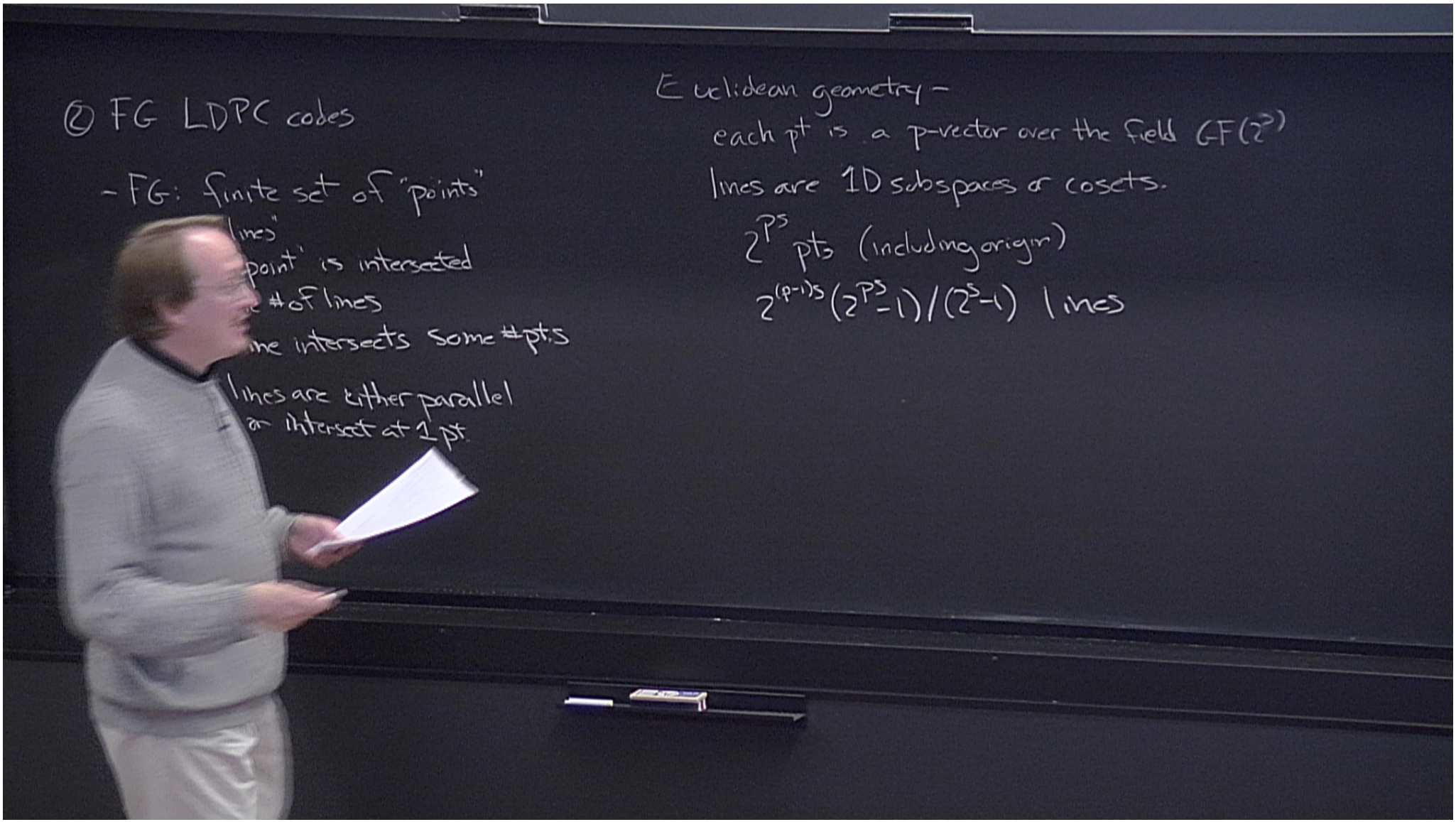
- FG: finite set of "points" and "lines"
  each "point" is intersected by some # of lines
  each line intersects some # pt.s
  two lines are either parallel or intersect at 1 pt.

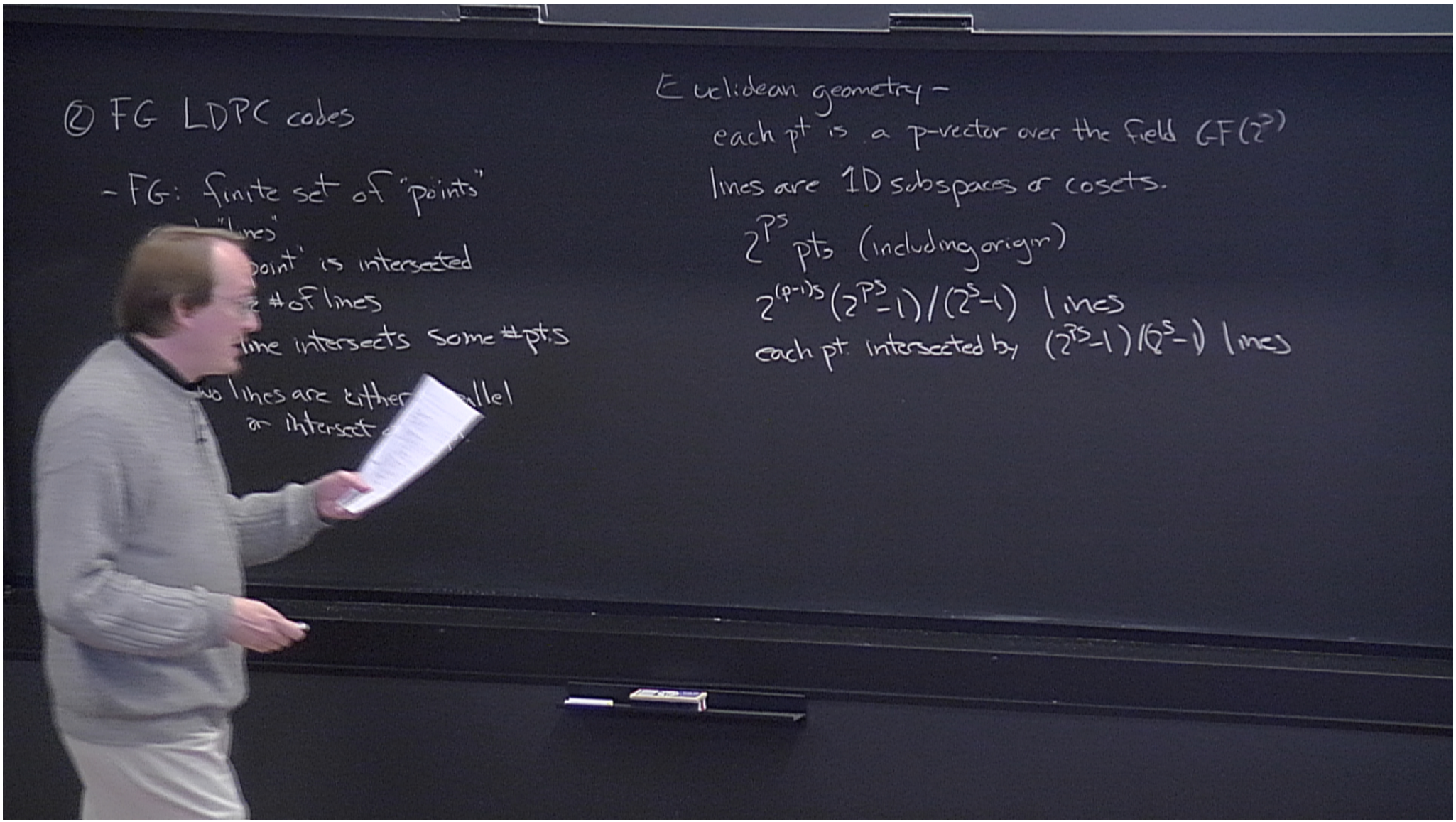
② FG LDPC codes

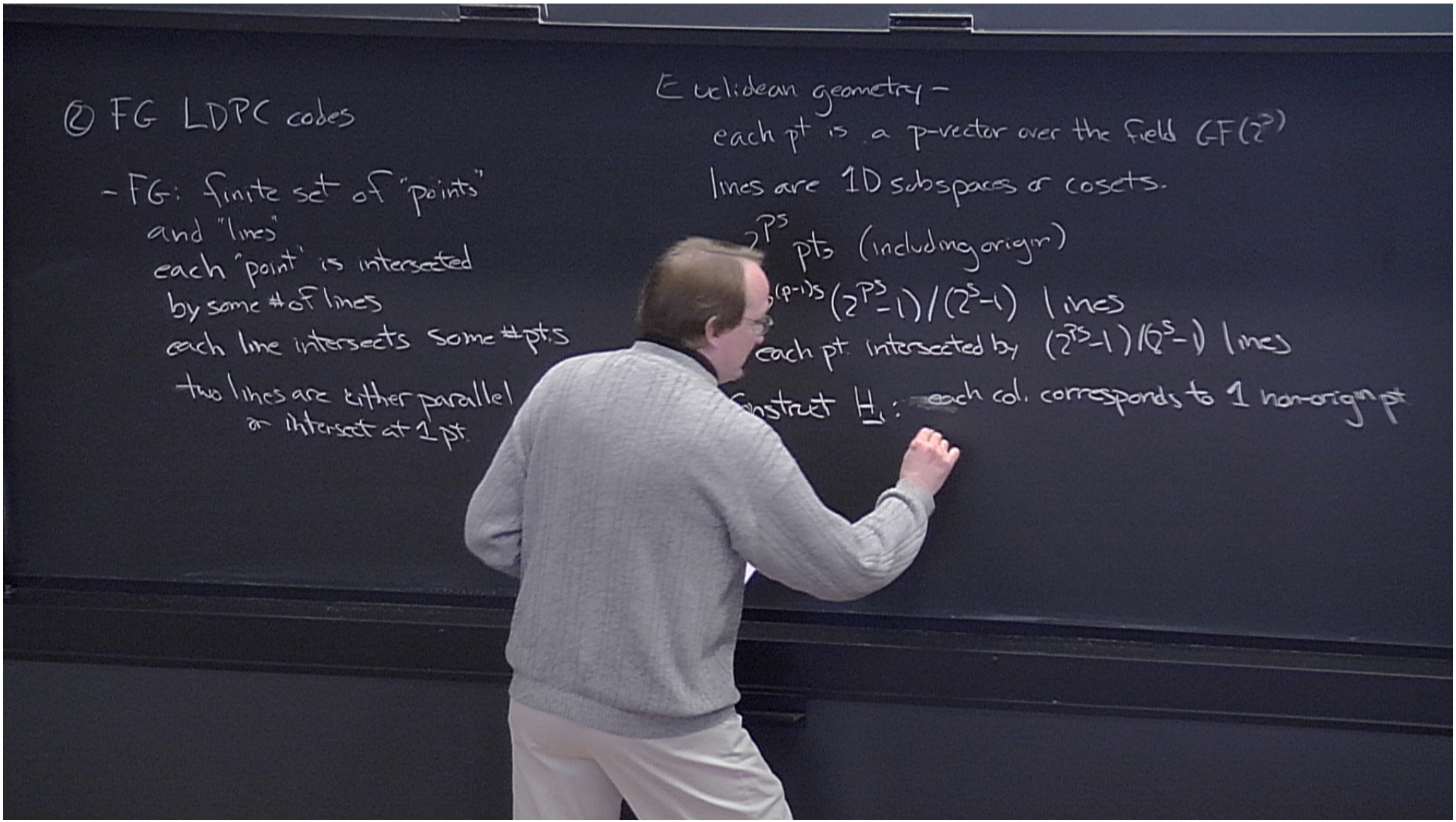
- FG: finite set of "points" and "lines"
  each "point" is intersected by some # of lines
  each line intersects some # pt.s
  two lines are either parallel or intersect at 1 pt.

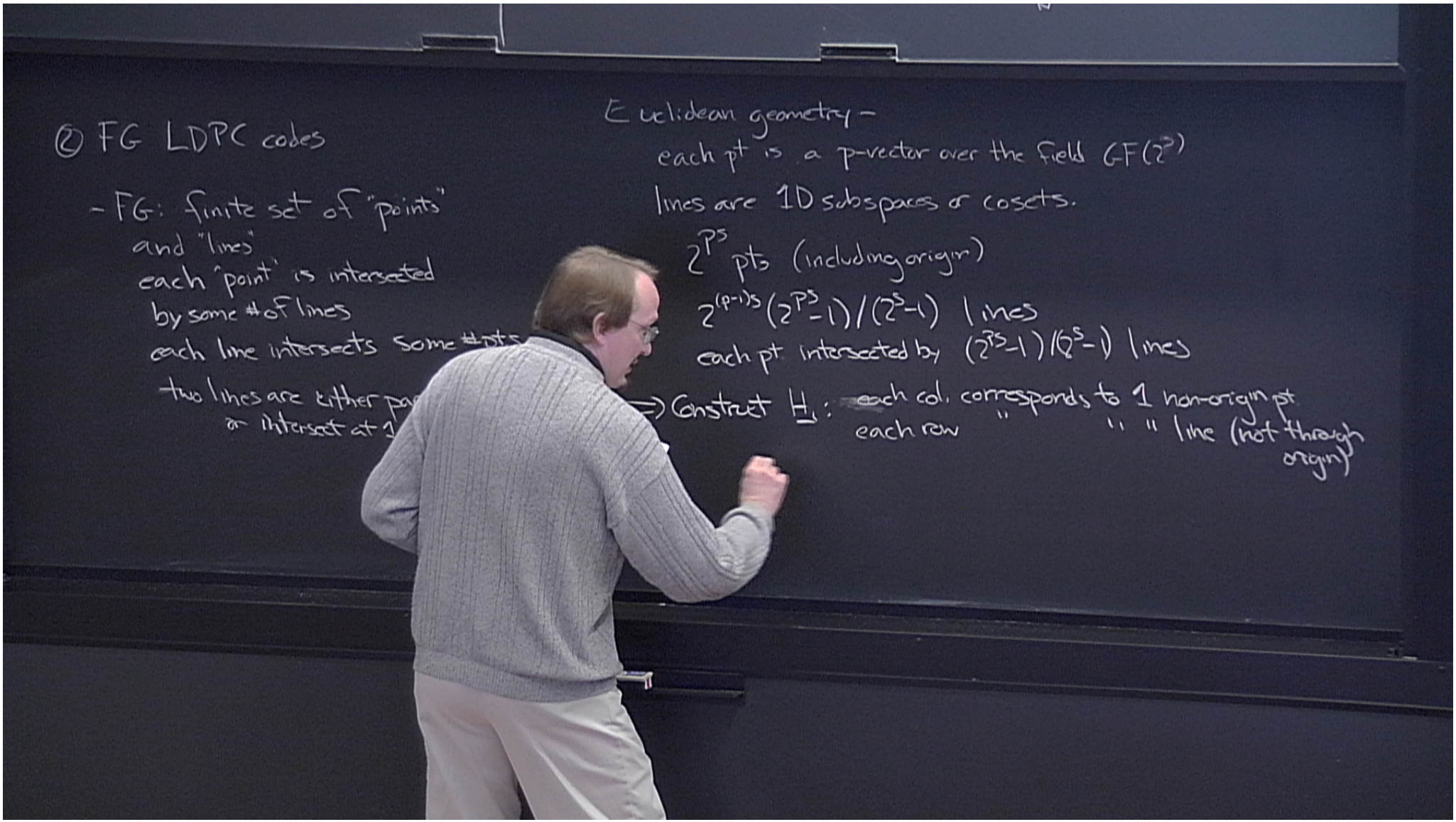

Bob
② FG LDPC codes
Euclidean geometry –
each pt is a projector over the field GF(q³)
lines are 1D subspaces or cosets.
① A&B compare notes about basis choice
– discard any bits where the bases don't match
② A&B randomly choose δn bits to
assess the channel → error rate p
E₁ ← logical operators on the code
lost c bits all zero
F₁ ← "canonical error ops"

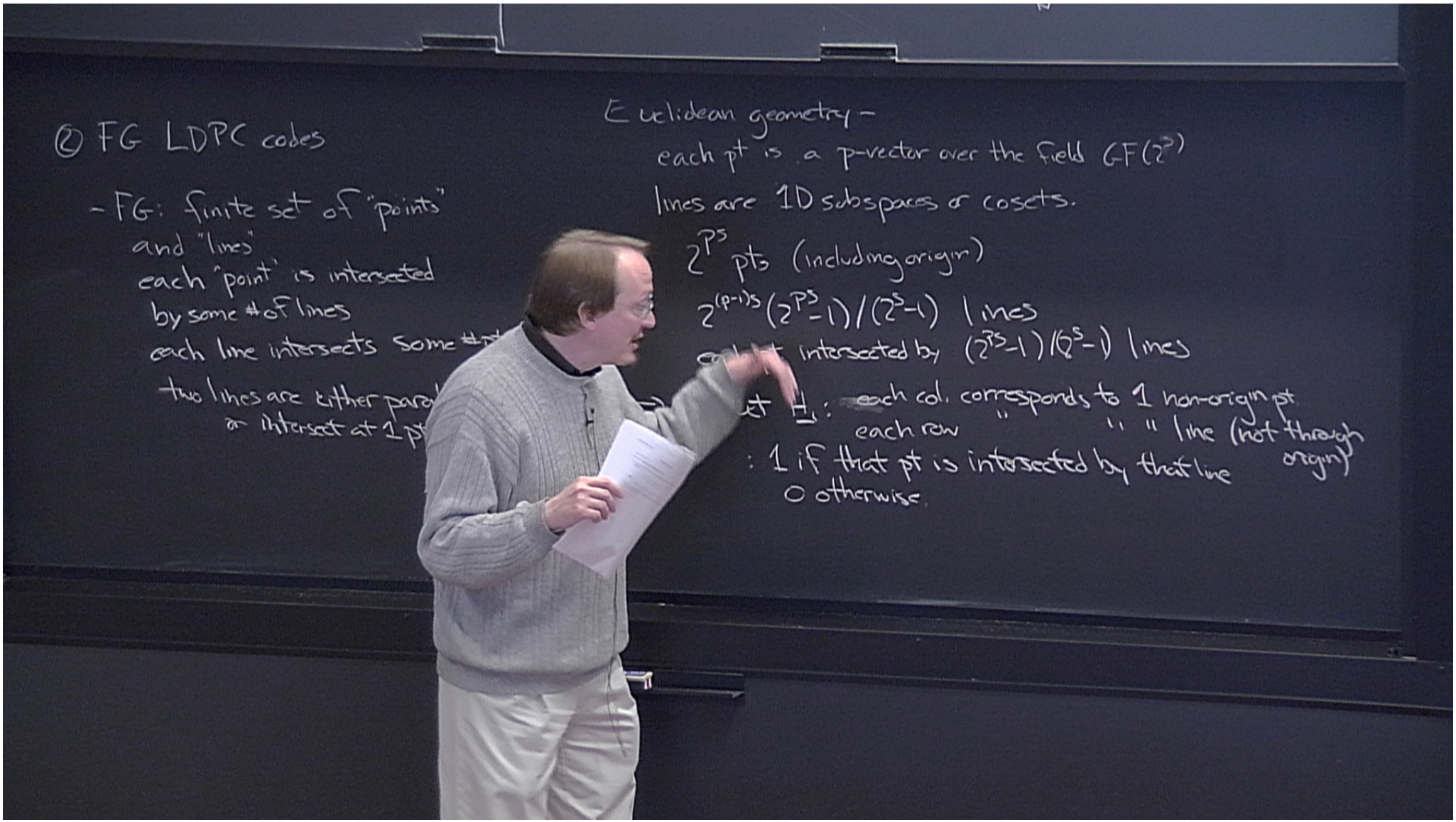

② FG LDPC codes
- FG: finite set of "points"
and "lines"
each "point" is intersec
by some # of lines
each line intersects
two lines are
an inters
Euclidean geometry -
each pt is a p-vector over the field GF(2^p)
lines are 1D subspaces or cosets.


② FG LDPC codes
- FG: finite set of "points"
lines"
point' is intersected
# of lines
ne intersects some # pt.s
lines are either parallel
or intersect at 1 pt.
Euclidean geometry -
each pt is a p-vector over the field GF(2^s)
lines are 1D subspaces or cosets.
2^ps pts (including origin)
2^(p-1)s (2^ps - 1)/(2^s - 1) lines


② FG LDPC codes
- FG: finite set of "points"
"lines"
point' is intersected
# of lines
line intersects some # pts
two lines are either parallel
or intersect
Euclidean geometry -
each pt is a p-vector over the field GF(2^s)
lines are 1D subspaces or cosets.
2^ps pts (including origin)
2^(p-1)s (2^ps - 1)/(2^s - 1) lines
each pt intersected by (2^ps - 1)/(2^s - 1) lines


② FG LDPC codes
- FG: finite set of "points" and "lines"
each "point" is intersected by some # of lines
each line intersects some # pts
two lines are either parallel or intersect at 1 pt.
Euclidean geometry -
each pt is a p-vector over the field GF(2^s)
lines are 1D subspaces or cosets.
2^ps pts (including origin)
(2^ps-1)/(2^s-1) lines
each pt. intersected by (2^ps-1)/(2^s-1) lines
Construct H: each col. corresponds to 1 non-origin pt


② FG LDPC codes
- FG: finite set of "points" and "lines"
each "point" is intersected by some # of lines
each line intersects some # pts
two lines are either par
or intersect at 1
Euclidean geometry -
each pt is a p-vector over the field GF(2^s)
lines are 1D subspaces or cosets.
2^ps pts (including origin)
2^(p-1)s (2^ps-1)/(2^s-1) lines
each pt intersected by (2^ps-1)/(2^s-1) lines
=> Construct H: each col. corresponds to 1 non-origin pt
each row " " " line (not through origin)

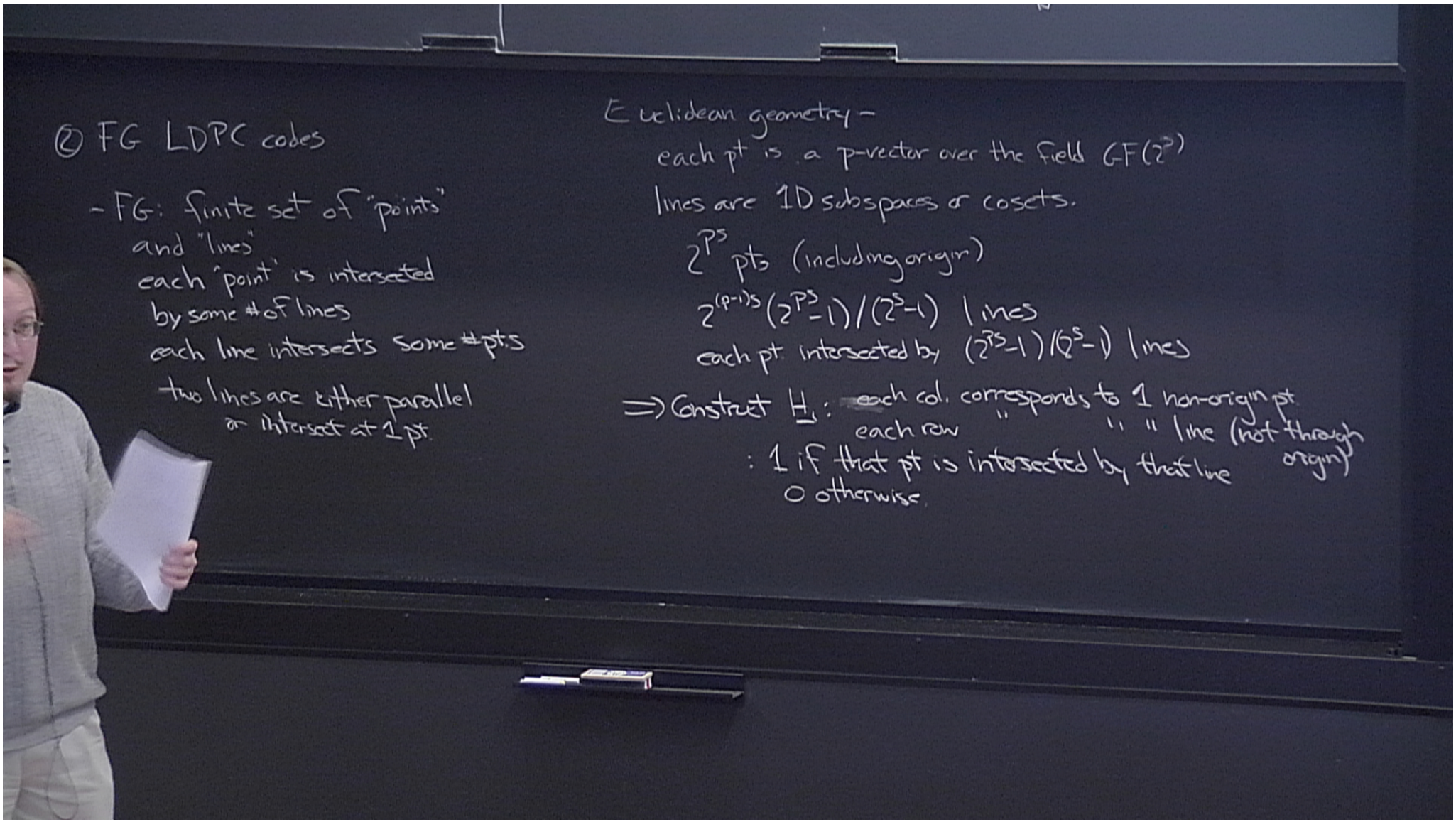

② FG LDPC codes
- FG: finite set of "points" and "lines"
each "point" is intersected by some # of lines
each line intersects some #
two lines are either para
or intersect at 1 pt
Euclidean geometry -
each pt is a p-vector over the field GF(2^s)
lines are 1D subspaces or cosets.
2^ps pts (including origin)
2^(p-1)s (2^ps-1)/(2^s-1) lines
intersected by (2^ps-1)/(2^s-1) lines
each col. corresponds to 1 non-origin pt
each row " " " line (not through origin)
: 1 if that pt is intersected by that line
0 otherwise.

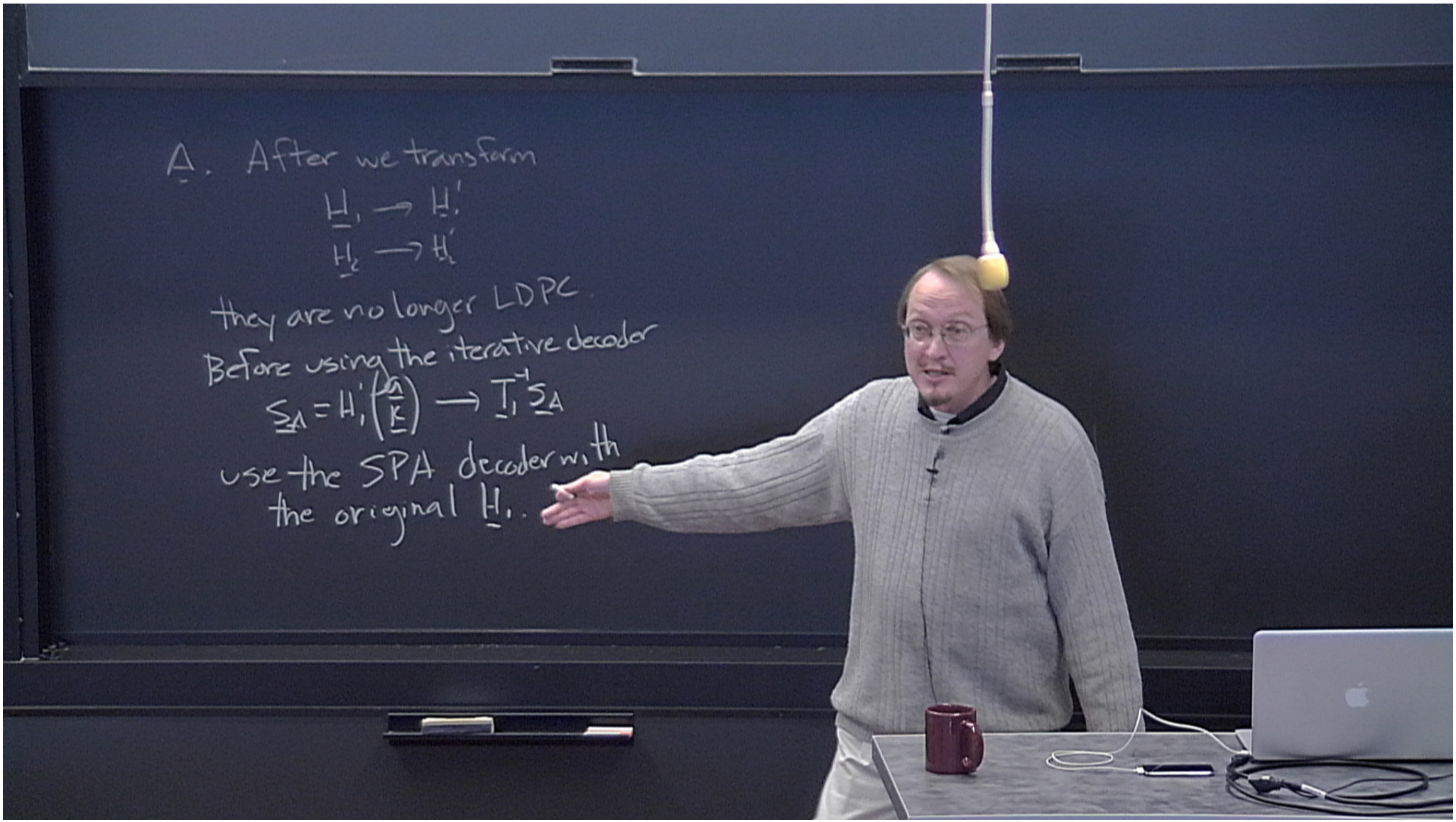
② FG LDPC codes

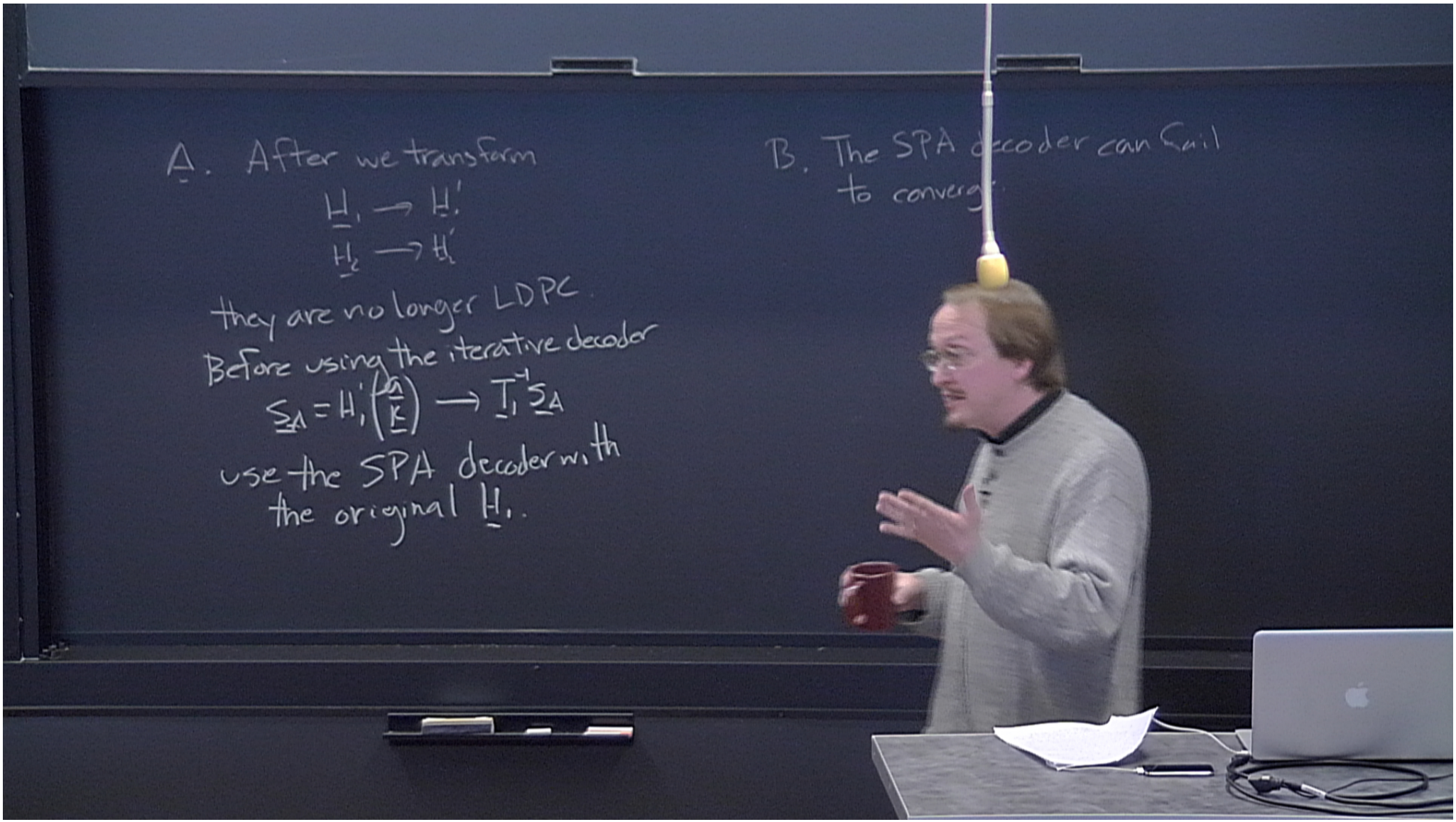
- FG: finite set of "points" and "lines"
  each "point" is intersected by some # of lines
  each line intersects some # pt.s
  two lines are either parallel or intersect at 1 pt.

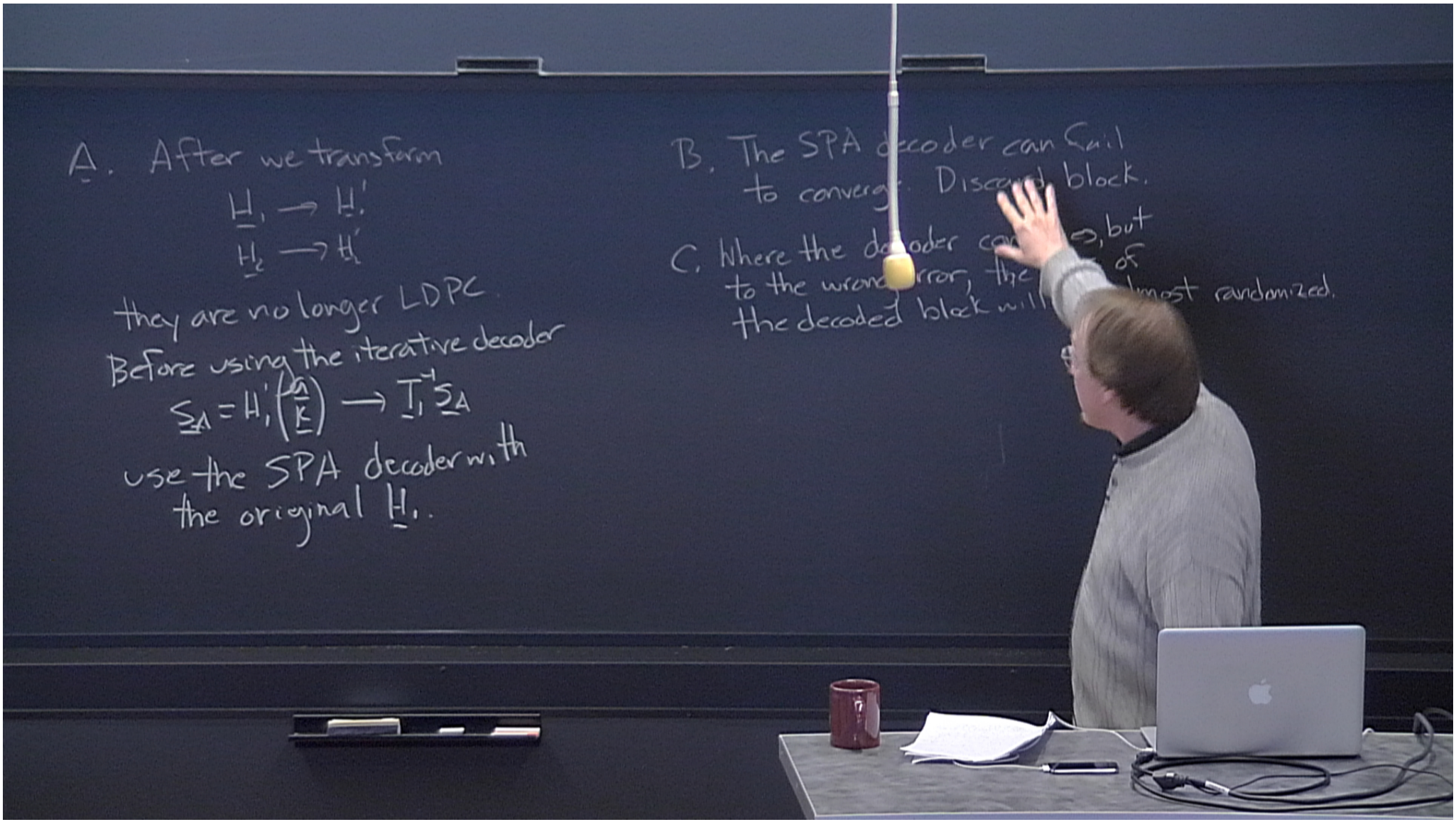
Euclidean geometry –
each pt is a p-vector over the field $GF(2^s)$
lines are 1D subspaces or cosets.
2^{ps} pts (including origin)
$2^{(p-1)s}(2^{ps}-1)/(2^s-1)$ lines
each pt intersected by $(2^{ps}-1)/(2^s-1)$ lines

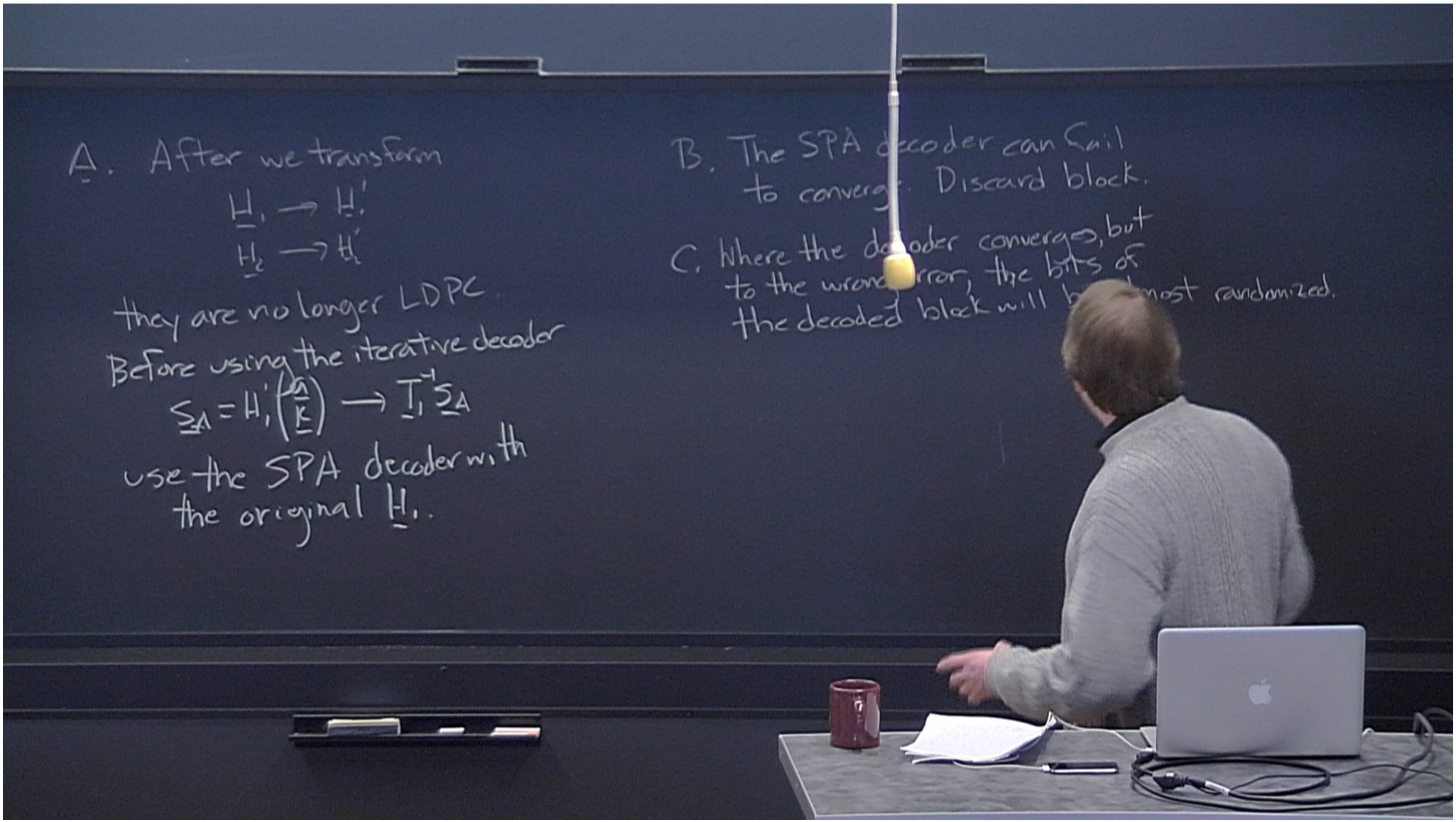
$\Rightarrow$ Construct $\underline{H}$: each col. corresponds to 1 non-origin pt
each row " " " line (not through origin)
: 1 if that pt is intersected by that line
0 otherwise.

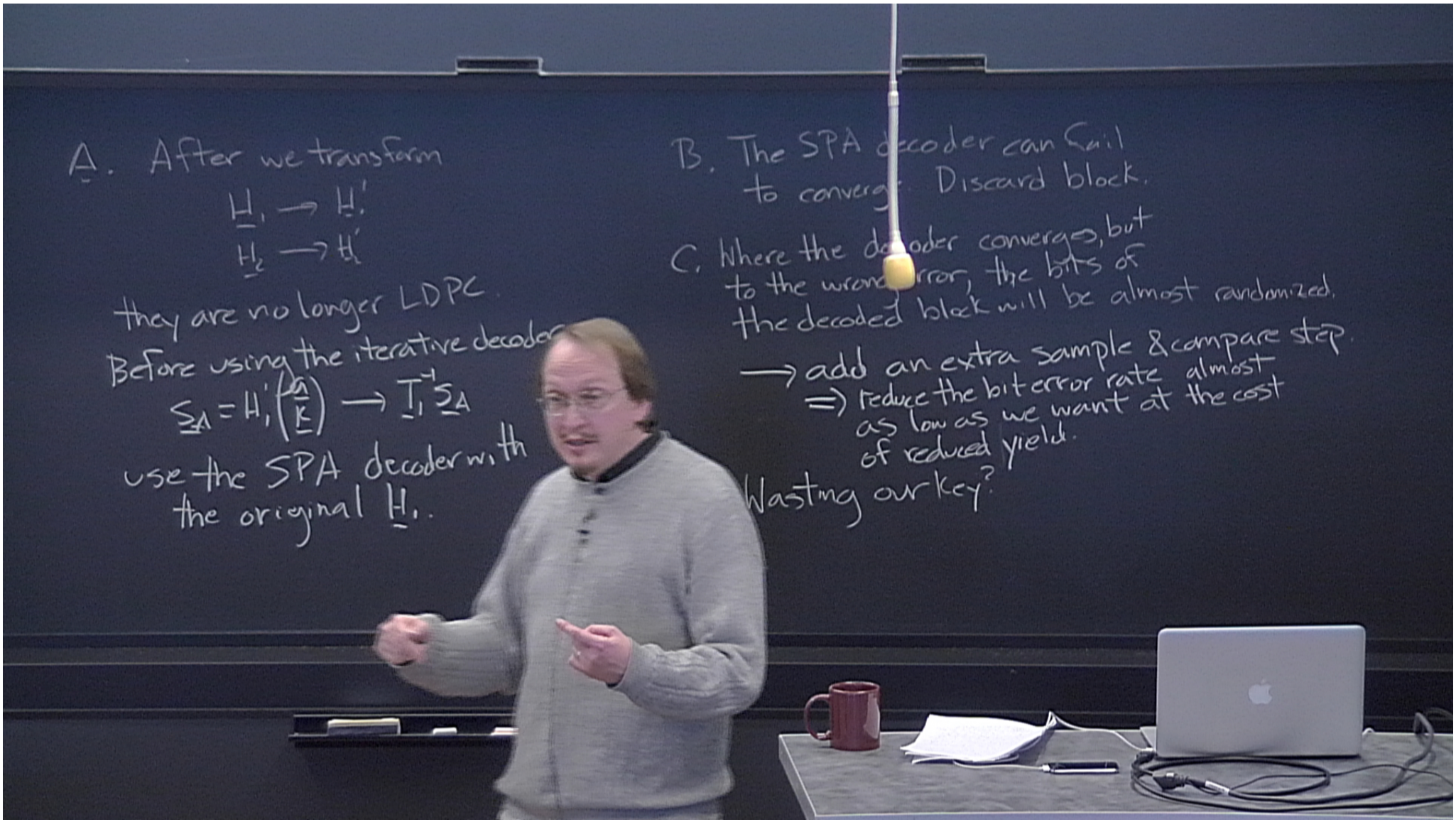

② FG LDPC codes
- FG: finite set of "points" and "lines"
each "point" is intersected by some # of lines
each line intersects some # pt.s
two lines are either parallel or intersect at 1 pt.
Euclidean geometry -
each pt is a p-vector over the field GF(2^s)
lines are 1D subspaces or cosets.
2^ps pts (including origin)
2^(p-1)s (2^ps-1)/(2^s-1) lines
each pt intersected by (2^ps-1)/(2^s-1) lines
⇒ Construct H: each col. corresponds to 1 non-origin pt
each row " " " line (not through origin)
: 1 if that pt is intersected by that line
0 otherwise

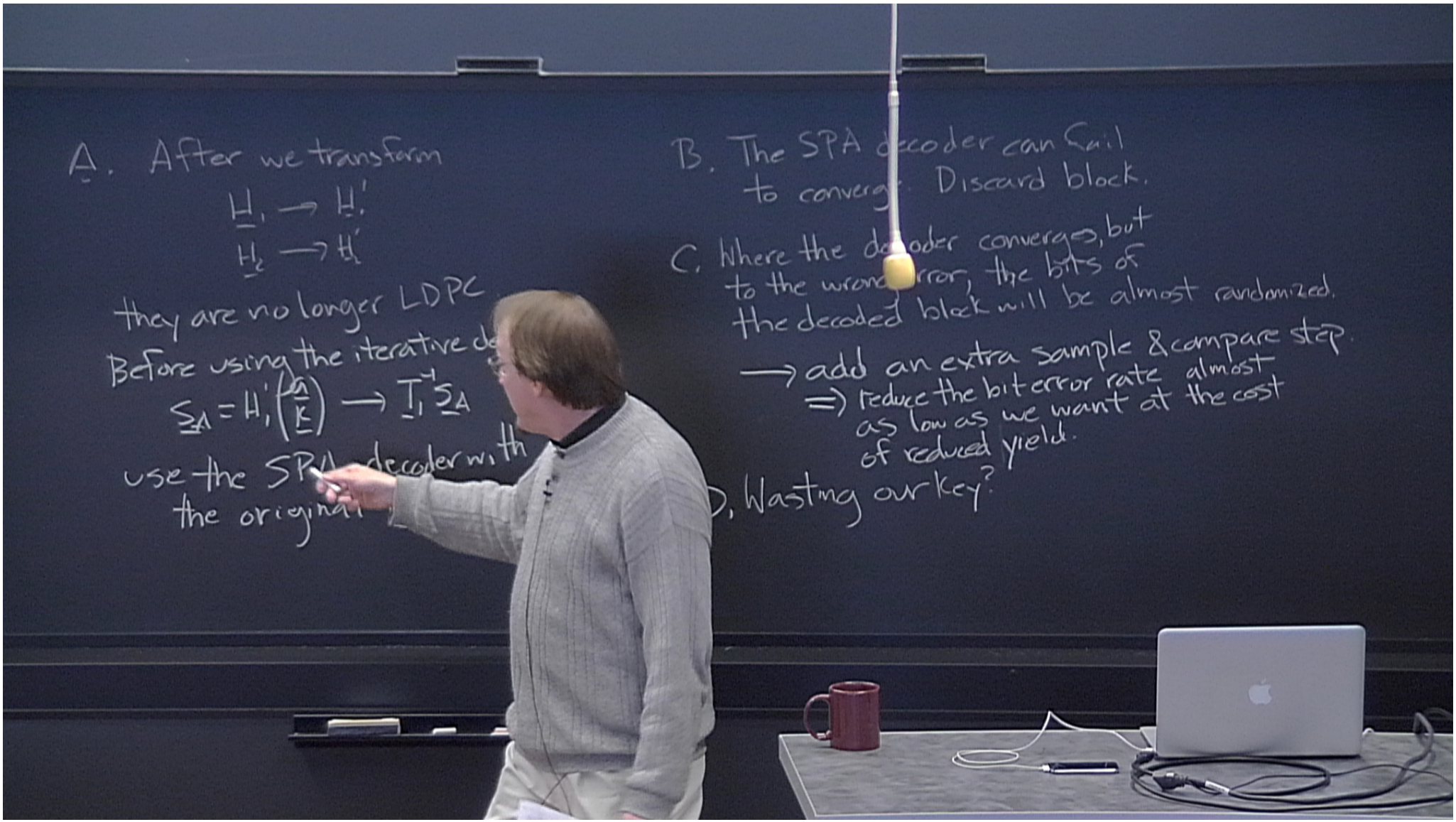

A. After we transform
$\underline{H}_1 \rightarrow \underline{H}_1'$
$\underline{H}_2 \rightarrow \underline{H}_2'$
they are no longer LDPC
Before using the iterative decoder
$\underline{S}_A = H_1' \begin{pmatrix} \underline{A} \\ \underline{K} \end{pmatrix} \rightarrow \underline{T}_1^{-1} \underline{S}_A$
use the SPA decoder with
the original $\underline{H}_1$.


A. After we transform
$\underline{H}_1 \to \underline{H}_1'$
$\underline{H}_2 \to \underline{H}_2'$
they are no longer LDPC.
Before using the iterative decoder
$\underline{S}_A = H_1' \begin{pmatrix} \underline{s}_A \\ \underline{k} \end{pmatrix} \to \underline{T}_1^{-1} \underline{S}_A$
use the SPA decoder with
the original $\underline{H}_1$.
B. The SPA decoder can fail
to converge.


A. After we transform
$\underline{H}_1 \rightarrow \underline{H}_1'$
$\underline{H}_2 \rightarrow \underline{H}_2'$
they are no longer LDPC.
Before using the iterative decoder
$\underline{S}_A = H_1' \begin{pmatrix} \underline{a} \\ \underline{k} \end{pmatrix} \rightarrow \underline{T}_1^{-1} \underline{S}_A$
use the SPA decoder with
the original $\underline{H}_1$.
B. The SPA decoder can fail
to converg. Discard block.
C. Where the decoder
to the wrong error, the
the decoded block will
most randomized.


A. After we transform
$\underline{H}_1 \rightarrow \underline{H}_1'$
$\underline{H}_2 \rightarrow \underline{H}_2'$
they are no longer LDPC.
Before using the iterative decoder
$\underline{S}_A = \underline{H}_1' \begin{pmatrix} \underline{A} \\ \underline{K} \end{pmatrix} \rightarrow \underline{T}_1^{-1} \underline{S}_A$
use the SPA decoder with
the original $\underline{H}_1$.
B. The SPA decoder can fail
to converge. Discard block.
C. Where the decoder converges, but
to the wrong error, the bits of
the decoded block will be almost randomized.


A. After we transform
$\underline{H}_1 \rightarrow \underline{H}_1'$
$\underline{H}_2 \rightarrow \underline{H}_2'$
they are no longer LDPC.
Before using the iterative decoder
$\underline{S}_A = \underline{H}_1' \begin{pmatrix} \underline{A} \\ \underline{K} \end{pmatrix} \rightarrow \underline{T}_1^{-1} \underline{S}_A$
use the SPA decoder with
the original $\underline{H}_1$.
B. The SPA decoder can fail
to converge. Discard block.
C. Where the decoder converges, but
to the wrong error, the bits of
the decoded block will be almost randomized.
→ add an extra sample & compare step.
⇒ reduce the bit error rate almost
as low as we want at the cost
of reduced yield.
Wasting our key?


A. After we transform
$H_1 \rightarrow H_1'$
$H_2 \rightarrow H_2'$
they are no longer LDPC
Before using the iterative de
$S_A = H_1' \begin{pmatrix} A \\ K \end{pmatrix} \rightarrow T_1^{-1} S_A$
use the SPA decoder with
the original
B. The SPA decoder can fail
to converge. Discard block.
C. Where the decoder converges, but
to the wrong error, the bits of
the decoded block will be almost randomized.
→ add an extra sample & compare step.
⇒ reduce the bit error rate almost
as low as we want at the cost
of reduced yield.
D. Wasting our key?

D. Soln — don't use preshared Key until the last step.

① Bob uses $\underline{s}_A \oplus \underline{s}_B$

$$\underline{s}_A \oplus \underline{s}_B = H_1'\left[\begin{pmatrix}\underline{a}\\ \underline{k}\end{pmatrix} \oplus \begin{pmatrix}\underline{b}\\ \underline{k}\end{pmatrix}\right]$$

$$= H_1'\left[\begin{pmatrix}\underline{a}\\ \underline{0}\end{pmatrix} \oplus \begin{pmatrix}\underline{b}\\ \underline{0}\end{pmatrix}\right]$$

Euclidean geometry —

each pt is a p-vector over the field $GF(2^s)$

lines are 1D subspaces or cosets.

2^{ps} pts (including origin)

$2^{(p-1)s}(2^{ps}-1)/(2^s-1)$ lines

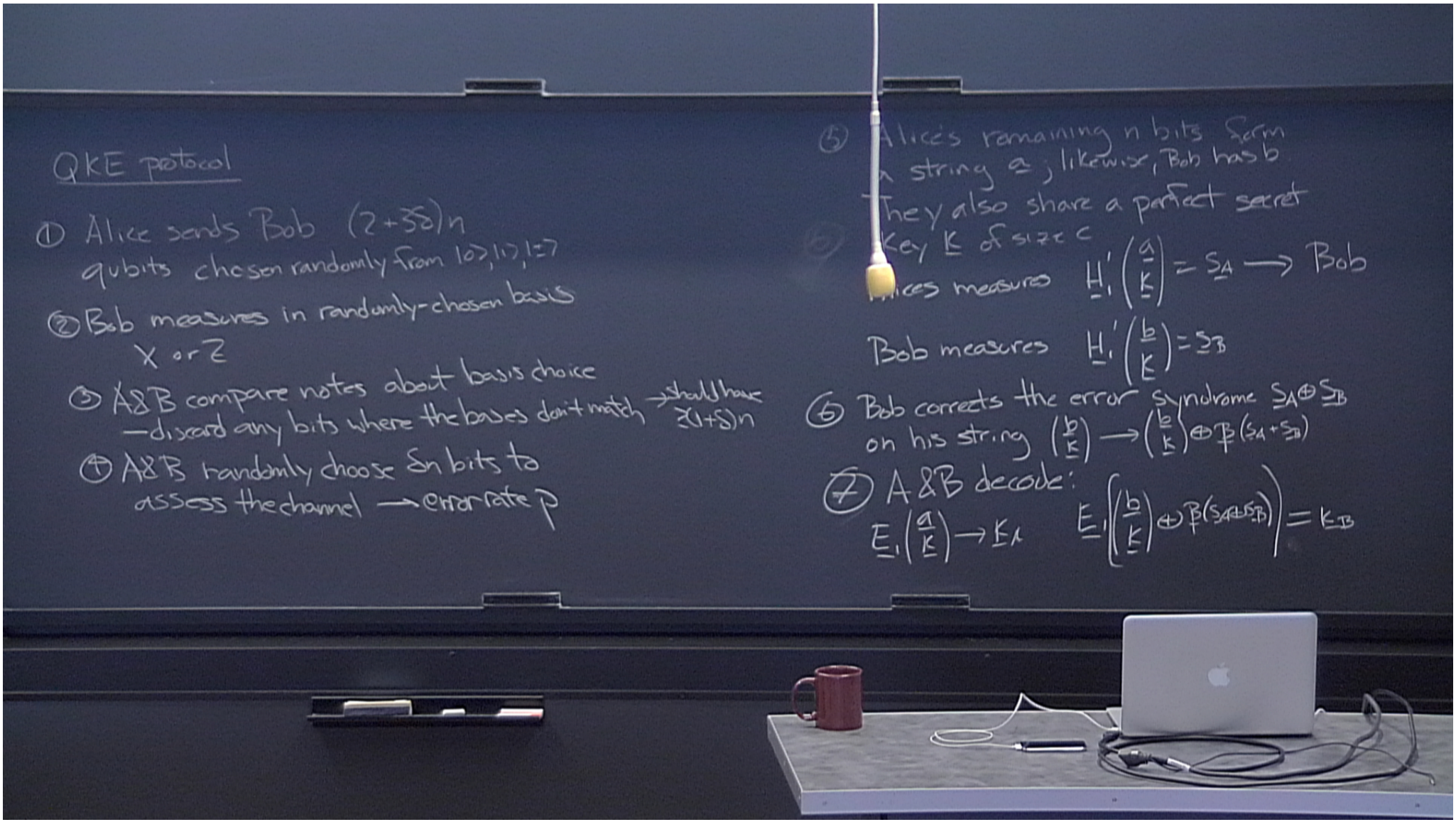
each pt intersected by $(2^{ps}-1)/(2^s-1)$ lines

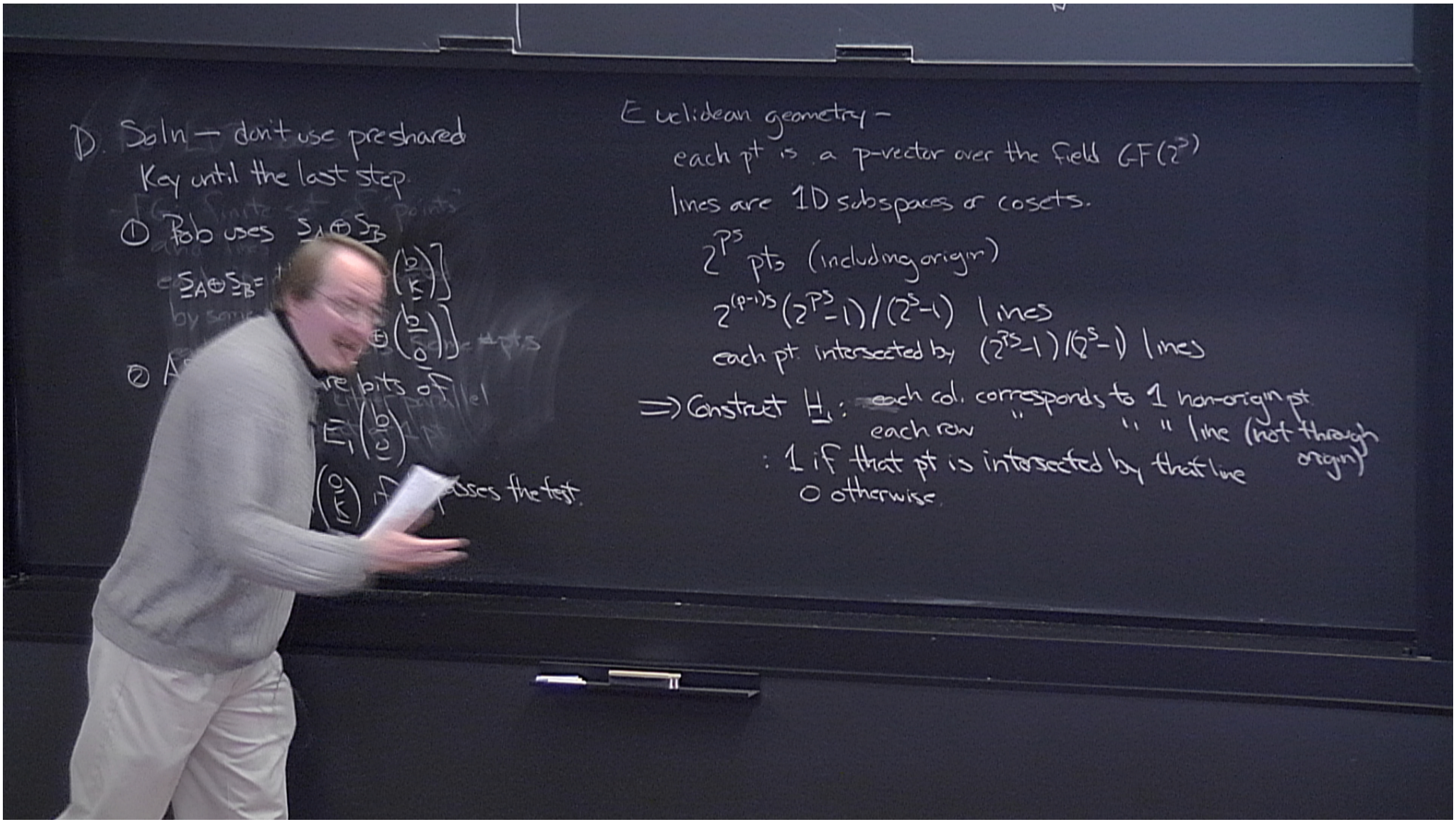
$\Rightarrow$ Construct $\underline{H}_1$: each col. corresponds to 1 non-origin pt.

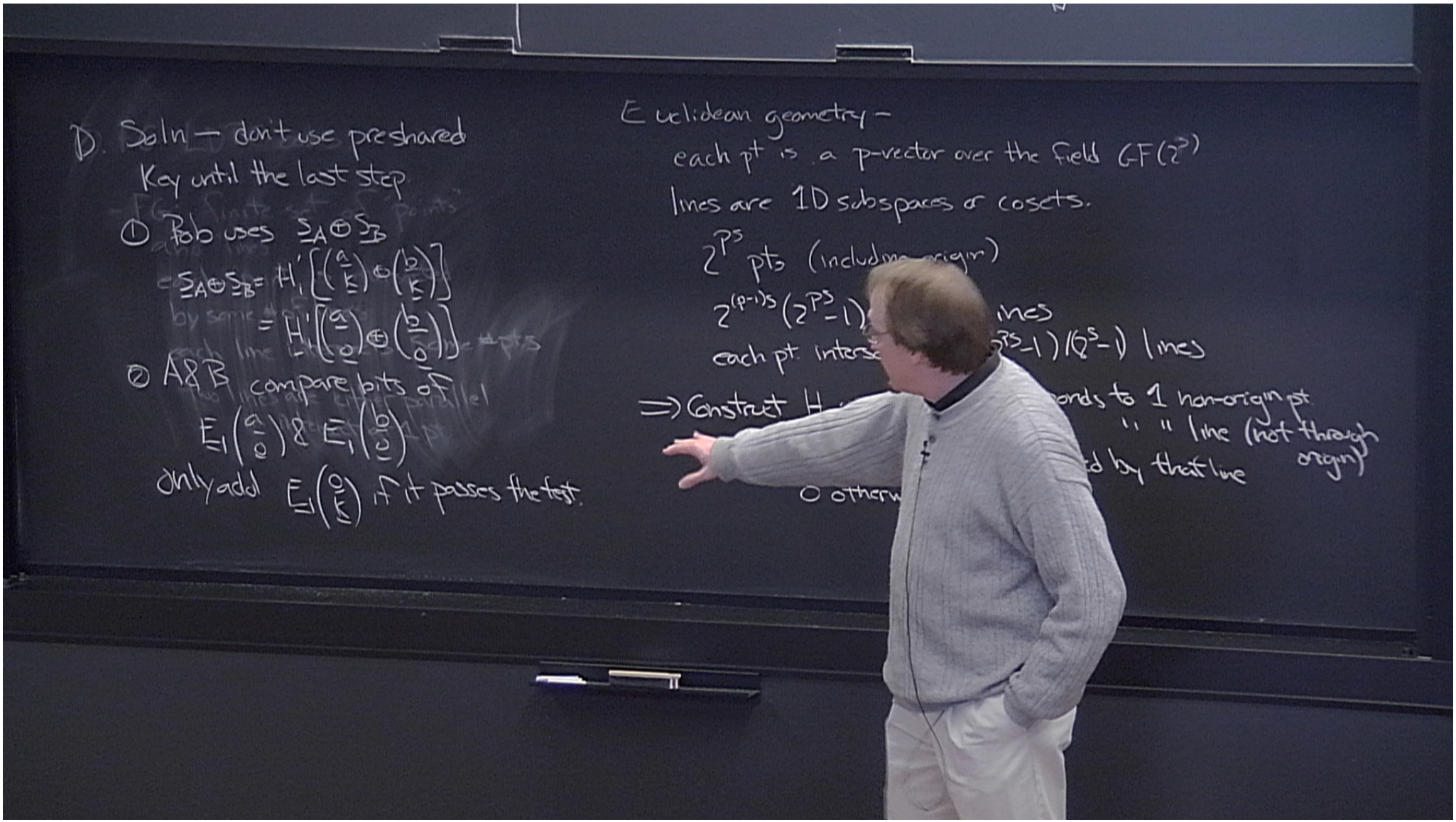
each row " " " line (not through origin)

: 1 if that pt is intersected by that line

0 otherwise.


QKE protocol
① Alice sends Bob (2+5δ)n qubits chosen randomly from |0>, |1>, |+>
② Bob measures in randomly-chosen basis X or Z
③ A&B compare notes about basis choice
—discard any bits where the bases don't match → should have ≈(1+δ)n
④ A&B randomly choose δn bits to assess the channel → error rate p
⑤ Alice's remaining n bits form a string a; likewise, Bob has b.
They also share a perfect secret key k of size c
⑥ Alice measures H'₁(a k) = s_A → Bob
Bob measures H'₁(b k) = s_B
⑥ Bob corrects the error syndrome s_A ⊕ s_B on his string (b k) → (b k) ⊕ P(s_A + s_B)
⑦ A&B decode:
E₁(a k) → k_A
E₁((b k) ⊕ P(s_A ⊕ s_B)) = k_B


D. Soln — don't use preshared
Key until the last step.
① Bob uses $S_A \oplus S_B$
② A
are bits of
$E_1 \begin{pmatrix} b \\ 0 \end{pmatrix}$
$\begin{pmatrix} 0 \\ k \end{pmatrix}$ uses the test.
Euclidean geometry –
each pt is a p-vector over the field $GF(2^s)$
lines are 1D subspaces or cosets.
2^{ps} pts (including origin)
$2^{(p-1)s}(2^{ps}-1)/(2^s-1)$ lines
each pt intersected by $(2^{ps}-1)/(2^s-1)$ lines
⇒ Construct H_1: each col. corresponds to 1 non-origin pt
each row " " " line (not through origin)
: 1 if that pt is intersected by that line
0 otherwise.


D. Soln — don't use preshared
Key until the last step
① Bob uses $\underline{s}_A \oplus \underline{s}_B$
$= \underline{s}_A \oplus \underline{s}_B = H_1'\left[\begin{pmatrix}\underline{a}\\ \underline{k}\end{pmatrix} \oplus \begin{pmatrix}\underline{b}\\ \underline{k}\end{pmatrix}\right]$
$= H_1'\left[\begin{pmatrix}\underline{a}\\ \underline{0}\end{pmatrix} \oplus \begin{pmatrix}\underline{b}\\ \underline{0}\end{pmatrix}\right]$
② A & B compare bits of
$E_1\begin{pmatrix}\underline{a}\\ \underline{0}\end{pmatrix}$ & $E_1\begin{pmatrix}\underline{b}\\ \underline{0}\end{pmatrix}$
only add $E_1\begin{pmatrix}\underline{0}\\ \underline{k}\end{pmatrix}$ if it passes the test.
Euclidean geometry –
each pt is a p-vector over the field $GF(2^s)$
lines are 1D subspaces or cosets.
2^{ps} pts (includ
$2^{(p-1)s}(2^{ps}-1)$ lines
each pt. inters
$(2^{ps}-1)/(2^s-1)$ lines
⇒ Construct H
ords to 1 non-origin pt
" " line (not through origin)
d by that line
0 otherw

D. Soln — don't use preshared Key until the last step

① Bob uses $\underline{s}_A \oplus \underline{s}_B$

$$\underline{s}_A \oplus \underline{s}_B = H_1'\left[\begin{pmatrix}\underline{a}\\ \underline{k}\end{pmatrix} \oplus \begin{pmatrix}\underline{b}\\ \underline{k}\end{pmatrix}\right]$$

$$= H_1'\left[\begin{pmatrix}\underline{a}\\ \underline{0}\end{pmatrix} \oplus \begin{pmatrix}\underline{b}\\ \underline{0}\end{pmatrix}\right]$$

② A & B compare bits of

$$E_1\begin{pmatrix}\underline{a}\\ \underline{0}\end{pmatrix} \;\&\; E_1\begin{pmatrix}\underline{b}\\ \underline{0}\end{pmatrix}$$

only add $E_1\begin{pmatrix}\underline{0}\\ \underline{k}\end{pmatrix}$ if it passes the test.

Euclidean geometry –

each pt is a p-vector over the field $GF(2^s)$

lines are 1D subspaces or cosets.

2^{ps} pts (including origin)

$2^{(p-1)s}(2^{ps}-1)/(2^s-1)$ lines

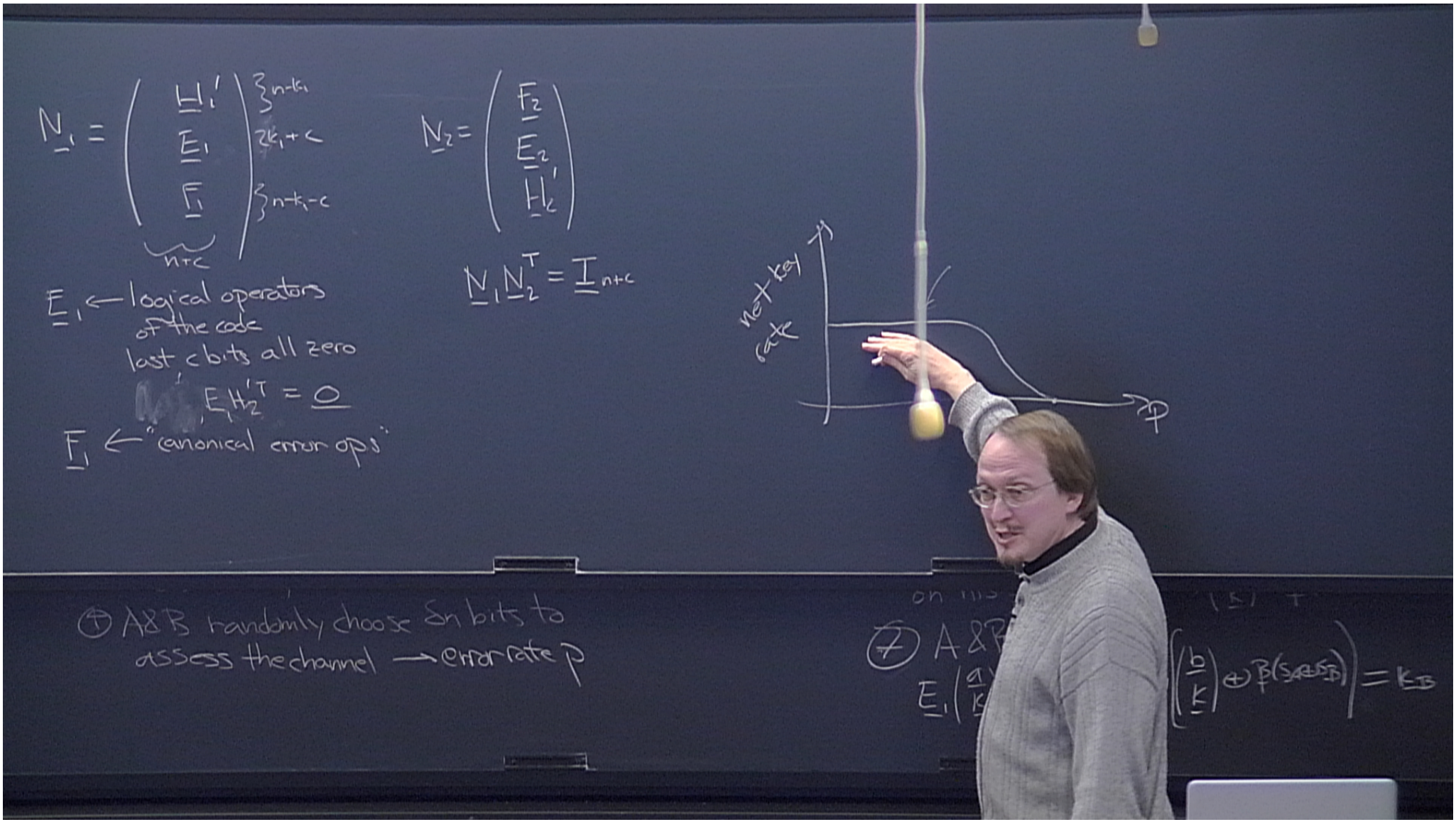
each pt intersected by $(2^{ps}-1)/(2^s-1)$ lines

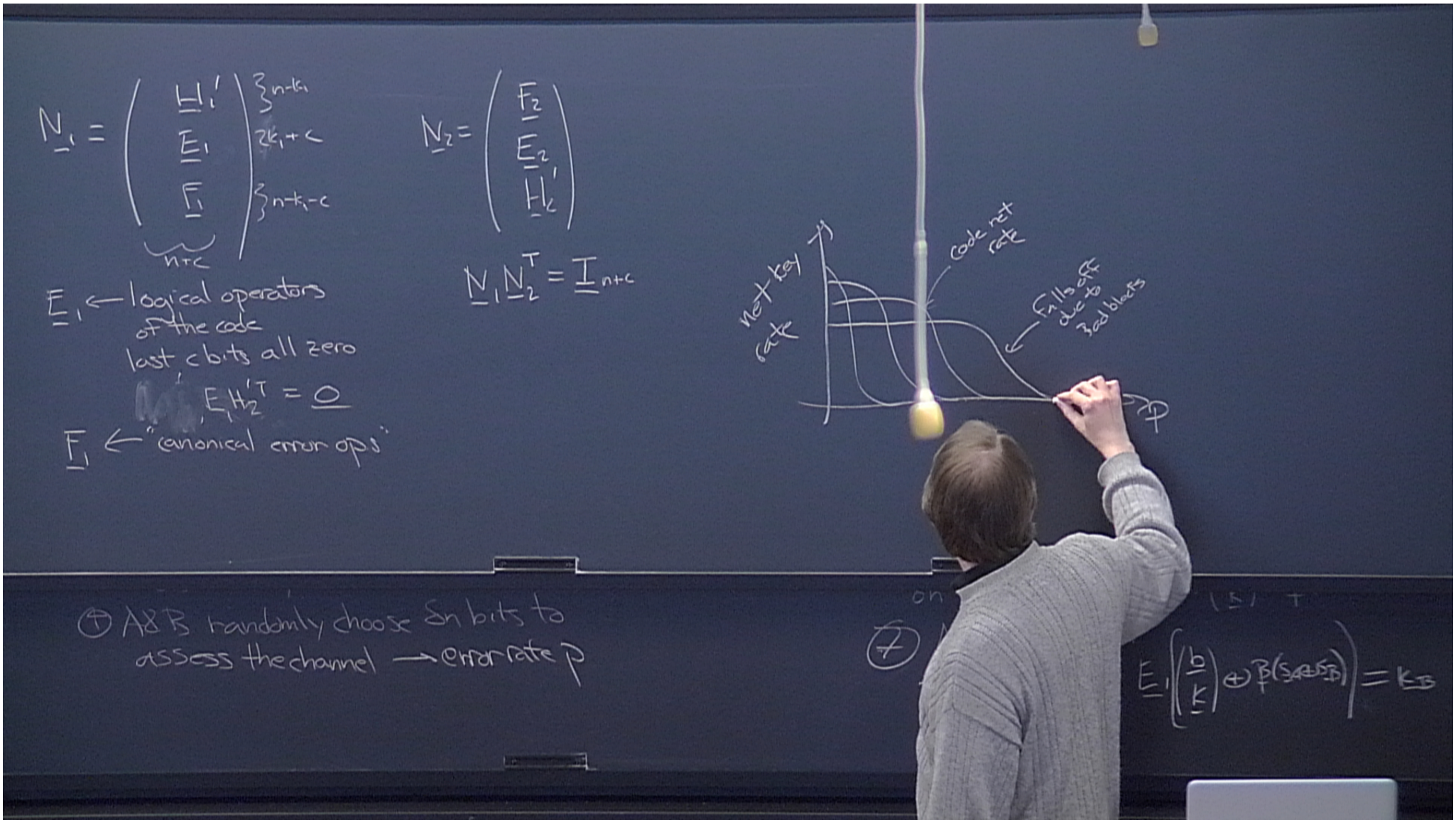
$\Rightarrow$ Construct $\underline{H}_1$: each col. corresponds to 1 non-origin pt

each row " " " line (not through origin)

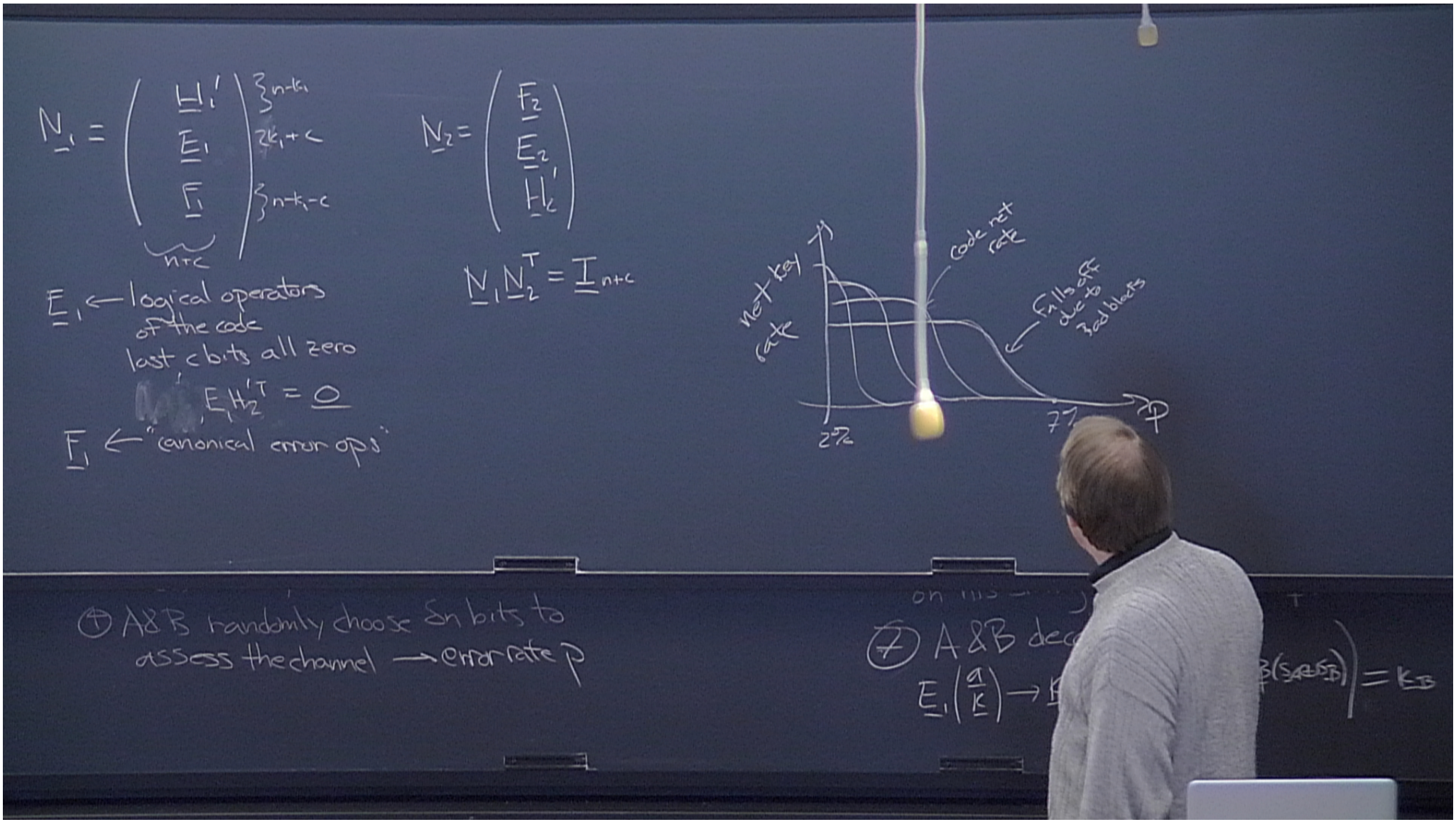
: 1 if that pt is intersected by that line

0 otherwise.


$N_1 = \begin{pmatrix} H_1' \\ E_1 \\ F_1 \end{pmatrix}$ $\begin{matrix} \}n-k_1 \\ 2k_1+c \\ \}n-k_1-c \end{matrix}$
$n+c$
$N_2 = \begin{pmatrix} F_2 \\ E_2 \\ H_2' \end{pmatrix}$
$N_1 N_2^T = I_{n+c}$
E_1 ← logical operators of the code
last c bits all zero
$E_1 H_2'^T = 0$
F_1 ← "canonical error ops"
net key rate
p
① A&B randomly choose δn bits to assess the channel → error rate p
⑦ A&B
$E_1 \begin{pmatrix} a \\ k \end{pmatrix}$
$\begin{pmatrix} b \\ k \end{pmatrix} \oplus \beta(s_A \oplus s_B)$
$= k_B$



$\underline{N}_1 = \begin{pmatrix} \underline{H}_1' \\ \underline{E}_1 \\ \underline{F}_1 \end{pmatrix}$ $\begin{matrix} \}n-k_1 \\ 2k_1+c \\ \}n-k_1-c \end{matrix}$
$n+c$
$\underline{N}_2 = \begin{pmatrix} \underline{F}_2 \\ \underline{E}_2 \\ \underline{H}_2' \end{pmatrix}$
$\underline{N}_1 \underline{N}_2^T = \underline{I}_{n+c}$
$\underline{E}_1 \leftarrow$ logical operators of the code
last c bits all zero
$E_1 H_2'^T = \underline{0}$
$\underline{F}_1 \leftarrow$ "canonical error ops"
net key rate
code net rate
falls off due to bad blocks
p
A&B randomly choose δn bits to assess the channel → error rate p



$\underline{N}_1 = \begin{pmatrix} \underline{H}_1' \\ \underline{E}_1 \\ \underline{F}_1 \end{pmatrix}$ $\begin{matrix} \}n-k_1 \\ 2k_1+c \\ \}n-k_1-c \end{matrix}$
$n+c$
$\underline{N}_2 = \begin{pmatrix} \underline{F}_2 \\ \underline{E}_2 \\ \underline{H}_2' \end{pmatrix}$
$\underline{N}_1 \underline{N}_2^T = \underline{I}_{n+c}$
$\underline{E}_1 \leftarrow$ logical operators of the code
last c bits all zero
$E_1 H_2'^T = \underline{0}$
$\underline{F}_1 \leftarrow$ "canonical error ops"
net key rate
code net rate
falls off due to bad blocks
2%
7%
p
④ A&B randomly choose δn bits to assess the channel → error rate p
⑦ A&B dec
$\underline{E}_1 \begin{pmatrix} \underline{a} \\ \underline{k} \end{pmatrix} \rightarrow$
$= \underline{k}_B$