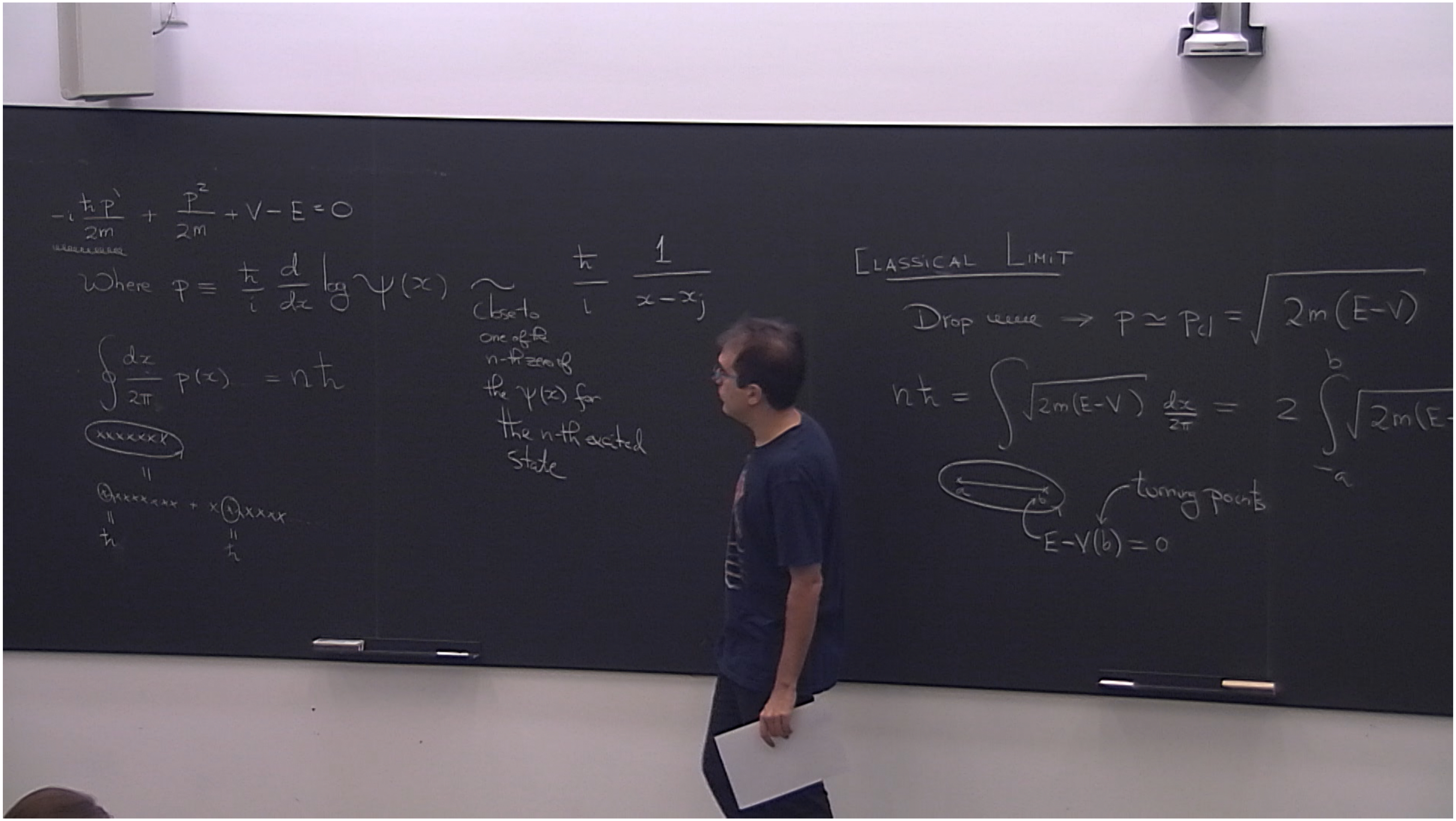
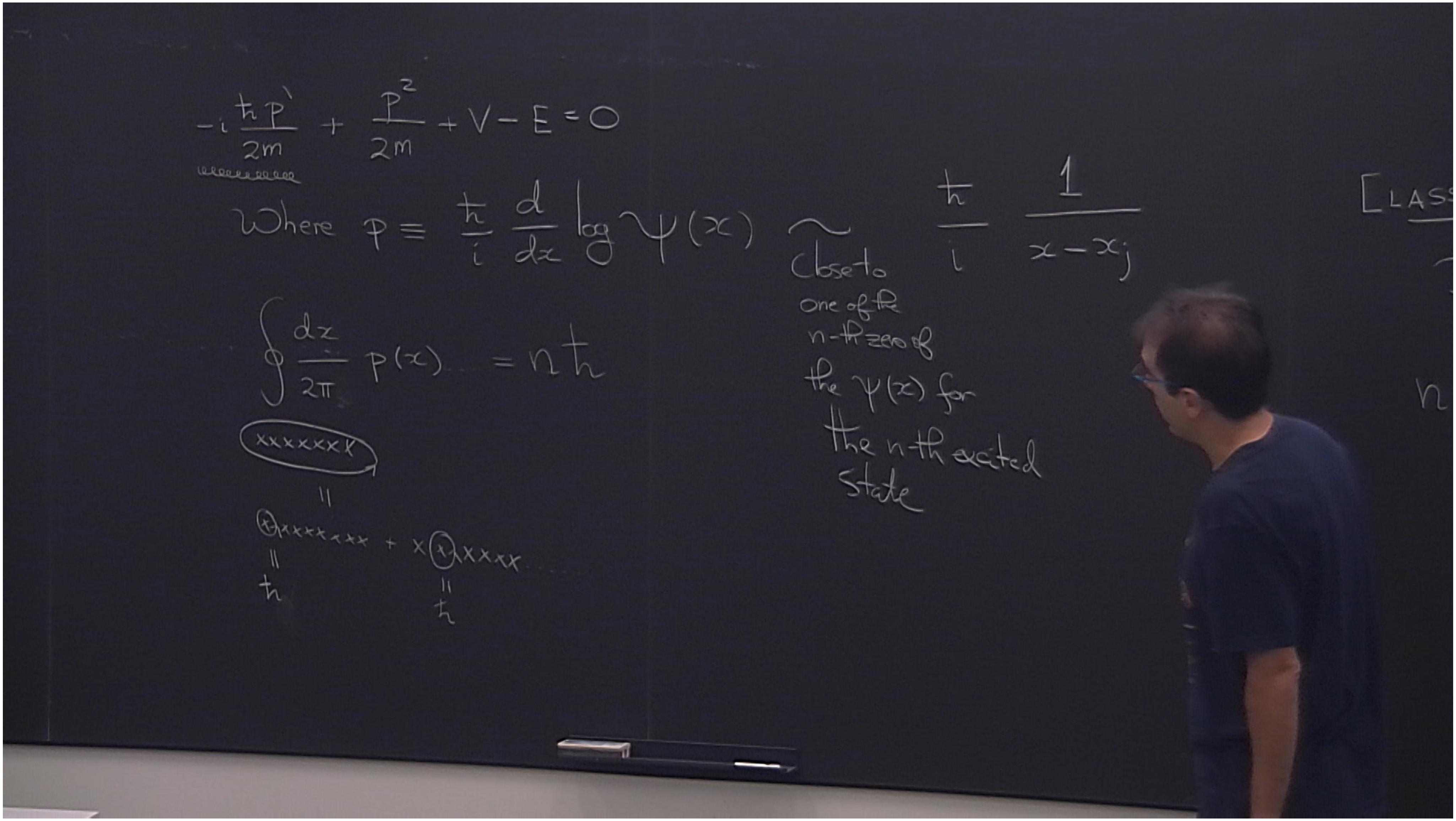
Title: PSI 2018/2019 - Statistical Mechanics - Lecture 3

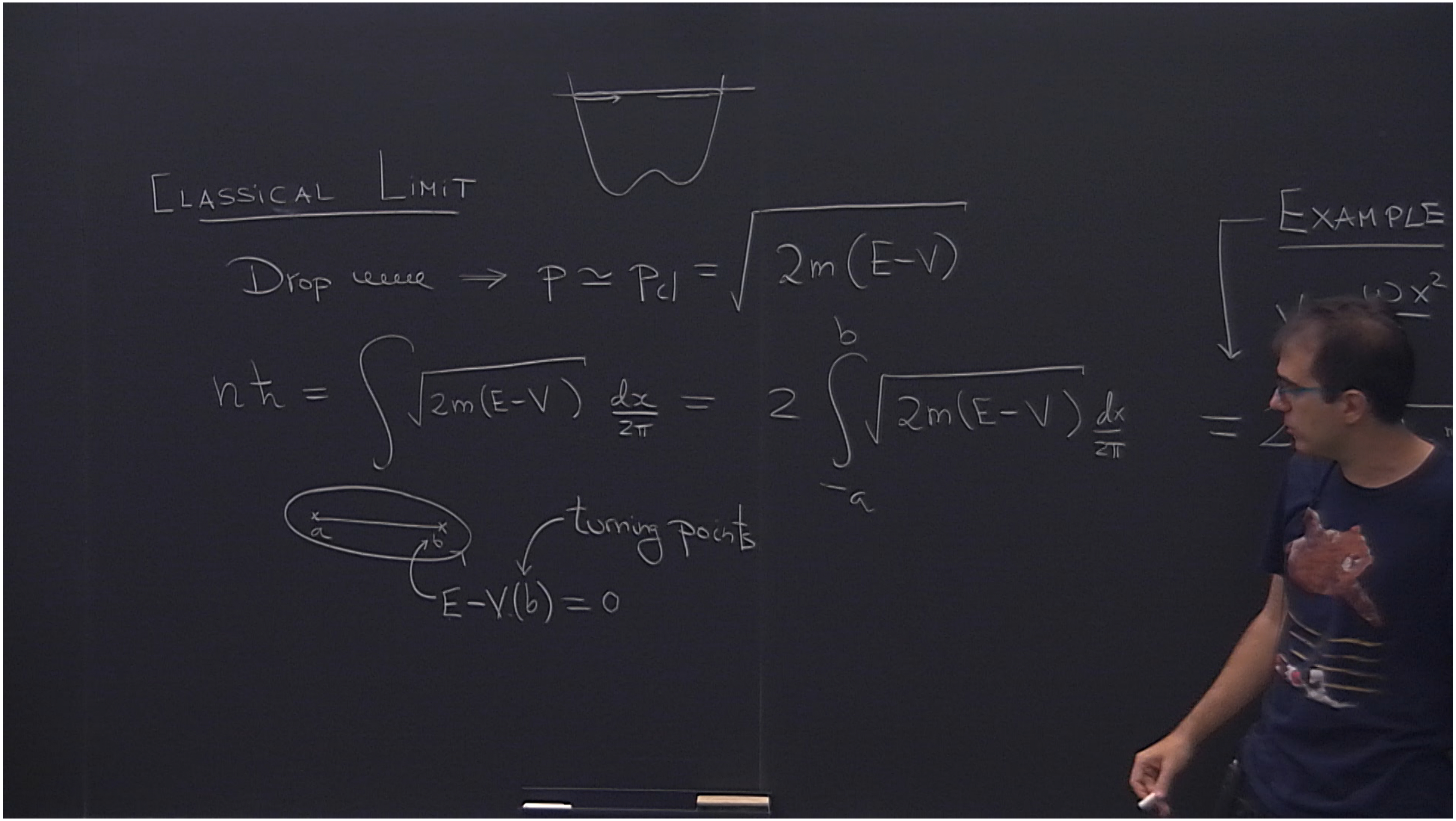
Date: Oct 11, 2018 10:45 AM

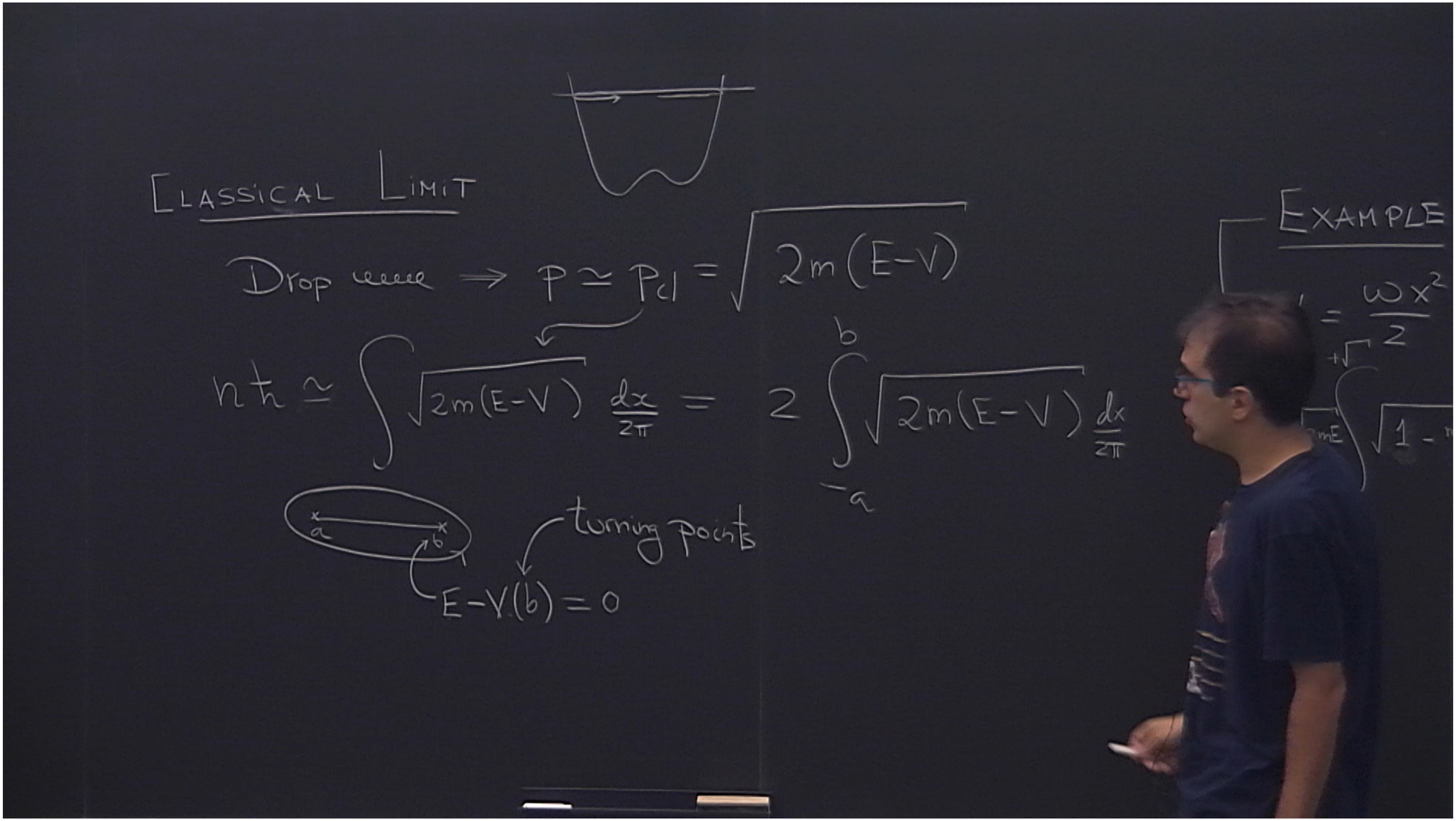
URL: http://pirsa.org/18100023

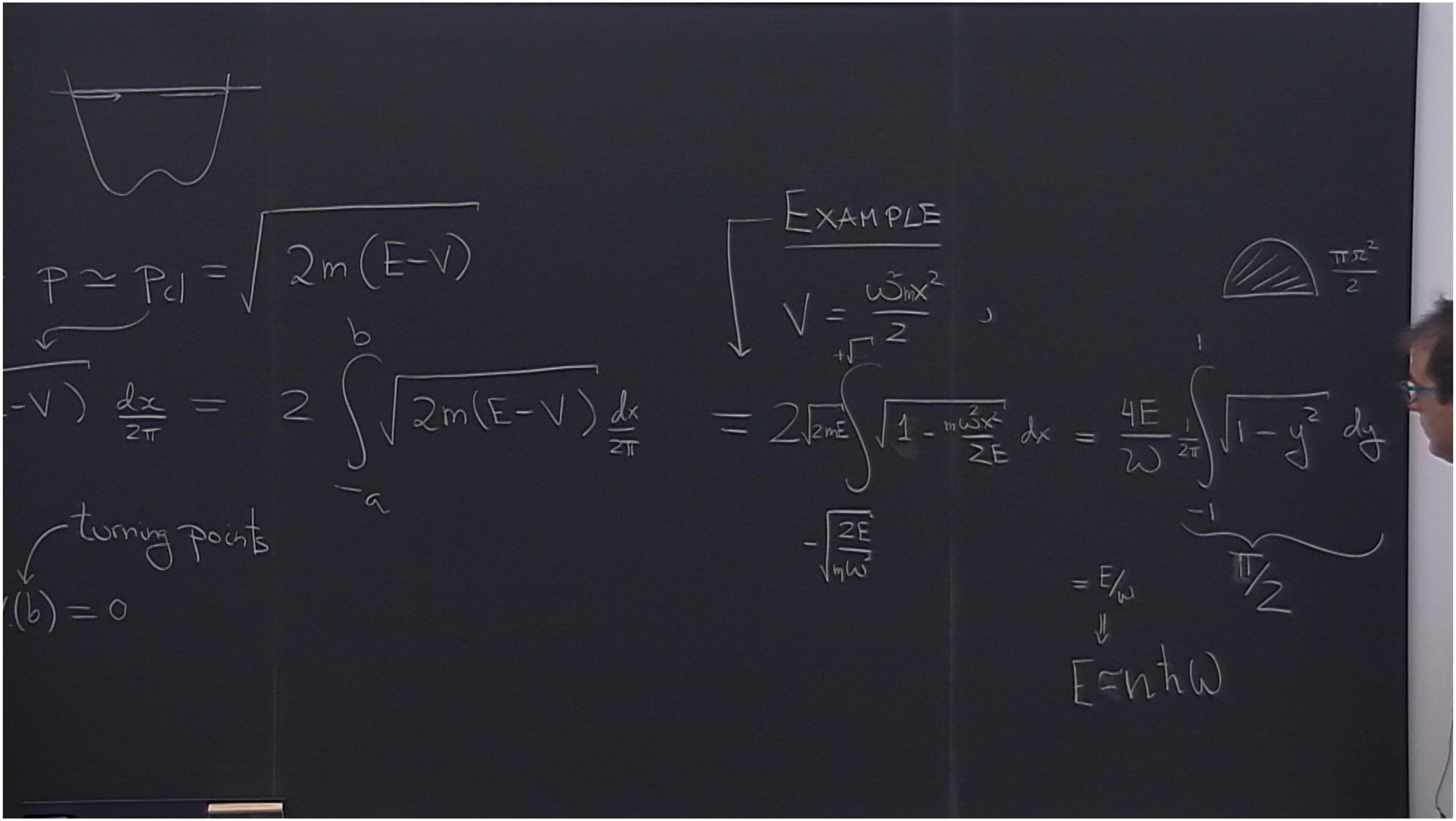
Abstract:

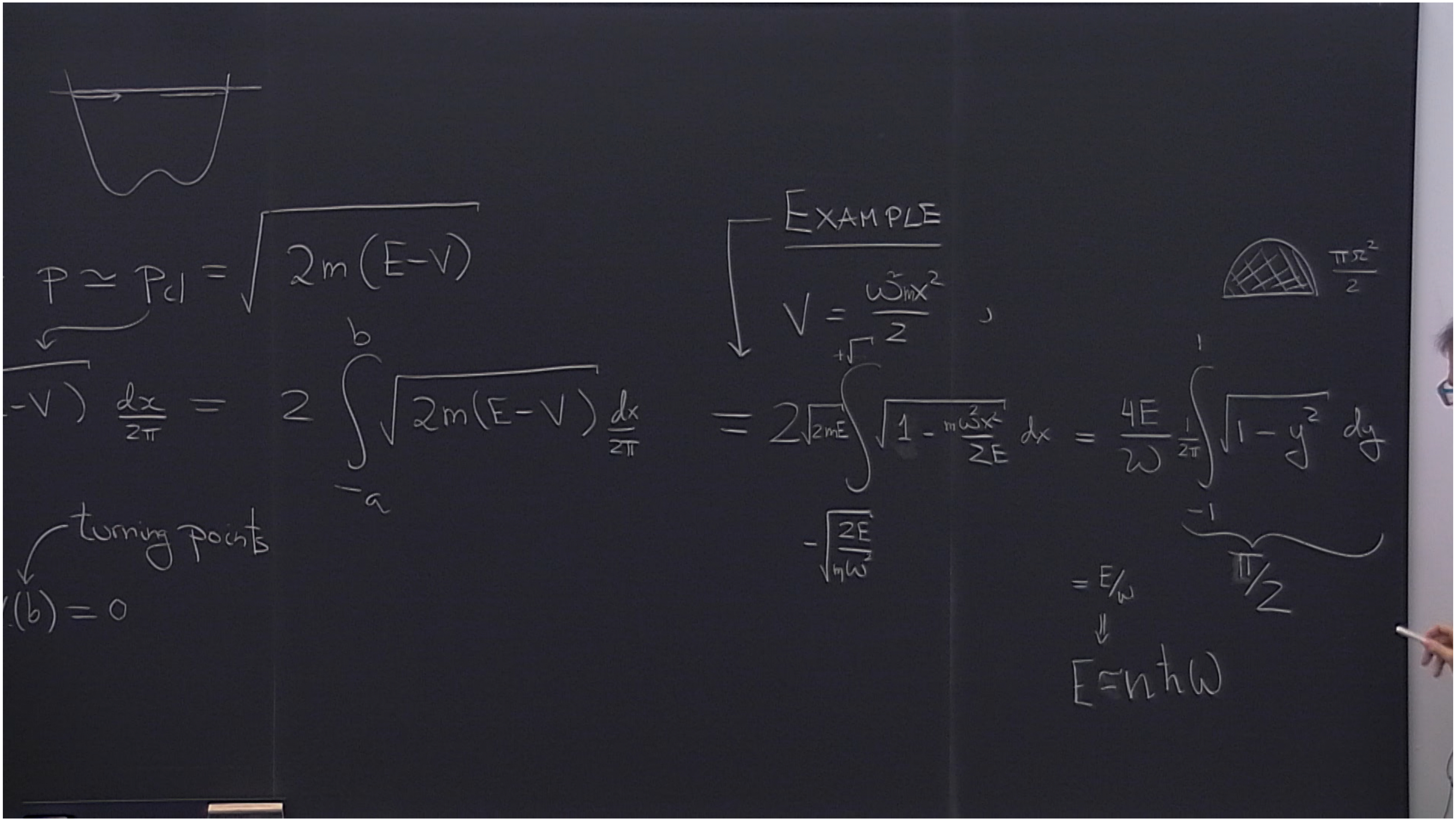

-i ħp'/2m + p²/2m + V - E = 0
Where p ≡ ħ/i d/dx log ψ(x) ~ ħ/i 1/(x - x_j)
Close to one of the n-th zeros of the ψ(x) for the n-th excited state
∮ dx/2π p(x) = nħ
CLASSICAL LIMIT
Drop → p ≃ p_cl = √(2m(E-V))
nħ = ∫√(2m(E-V)) dx/2π = 2 ∫_a^b √(2m(E-
turning points
E-V(b) = 0

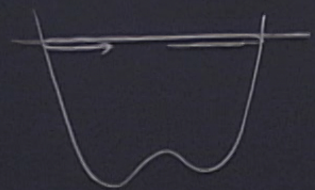

$-i\frac{\hbar p'}{2m} + \frac{p^2}{2m} + V - E = 0$
Where $p \equiv \frac{\hbar}{i}\frac{d}{dx}\log\psi(x) \sim \frac{\hbar}{i}\frac{1}{x-x_j}$
Close to one of the n-th zero of the $\psi(x)$ for the n-th excited state
$\oint \frac{dx}{2\pi} p(x) = n\hbar$


CLASSICAL LIMIT
Drop
p ≃ pcl = √2m(E−V)
nħ = ∫√2m(E−V) dx/2π = 2 ∫ √2m(E−V) dx/2π =
b
a
a
b
turning points
E−V(b) = 0
EXAMPLE


CLASSICAL LIMIT
Drop ℏ ⇒ p ≃ p_cl = √(2m(E−V))
nℏ ≃ ∫ √(2m(E−V)) dx/2π = 2 ∫_{−a}^{b} √(2m(E−V)) dx/2π
a
b
turning points
E−V(b) = 0
EXAMPLE


p ≃ p_cl = √(2m(E−V))
−V) dx/2π = 2 ∫_{−a}^{b} √(2m(E−V)) dx/2π
turning points
(b) = 0
EXAMPLE
V = ω²mx²/2
= 2√(2mE) ∫_{−√(2E/mω²)}^{+√} √(1 − mω²x²/2E) dx = (4E/ω)(1/2π) ∫_{−1}^{1} √(1 − y²) dy
π/2
πr²/2
= E/ω
⇓
E = nħω


EXAMPLE
V = ω²mx²/2
p ≃ p_cl = √(2m(E−V))
turning points
(b) = 0
= 2 ∫_{−a}^{b} √(2m(E−V)) dx/2π
= 2√(2mE) ∫ √(1 − mω²x²/2E) dx = 4E/ω · 1/2π ∫_{−1}^{1} √(1 − y²) dy
π/2
= E/ω
E = nħω
πr²/2



$$p \simeq p_{cl} = \sqrt{2m(E-V)}$$

$$\ldots -V)\,\frac{dx}{2\pi} = 2\int_{-a}^{b}\sqrt{2m(E-V)}\,\frac{dx}{2\pi}$$

turning points

$$\ldots(b) = 0$$

EXAMPLE

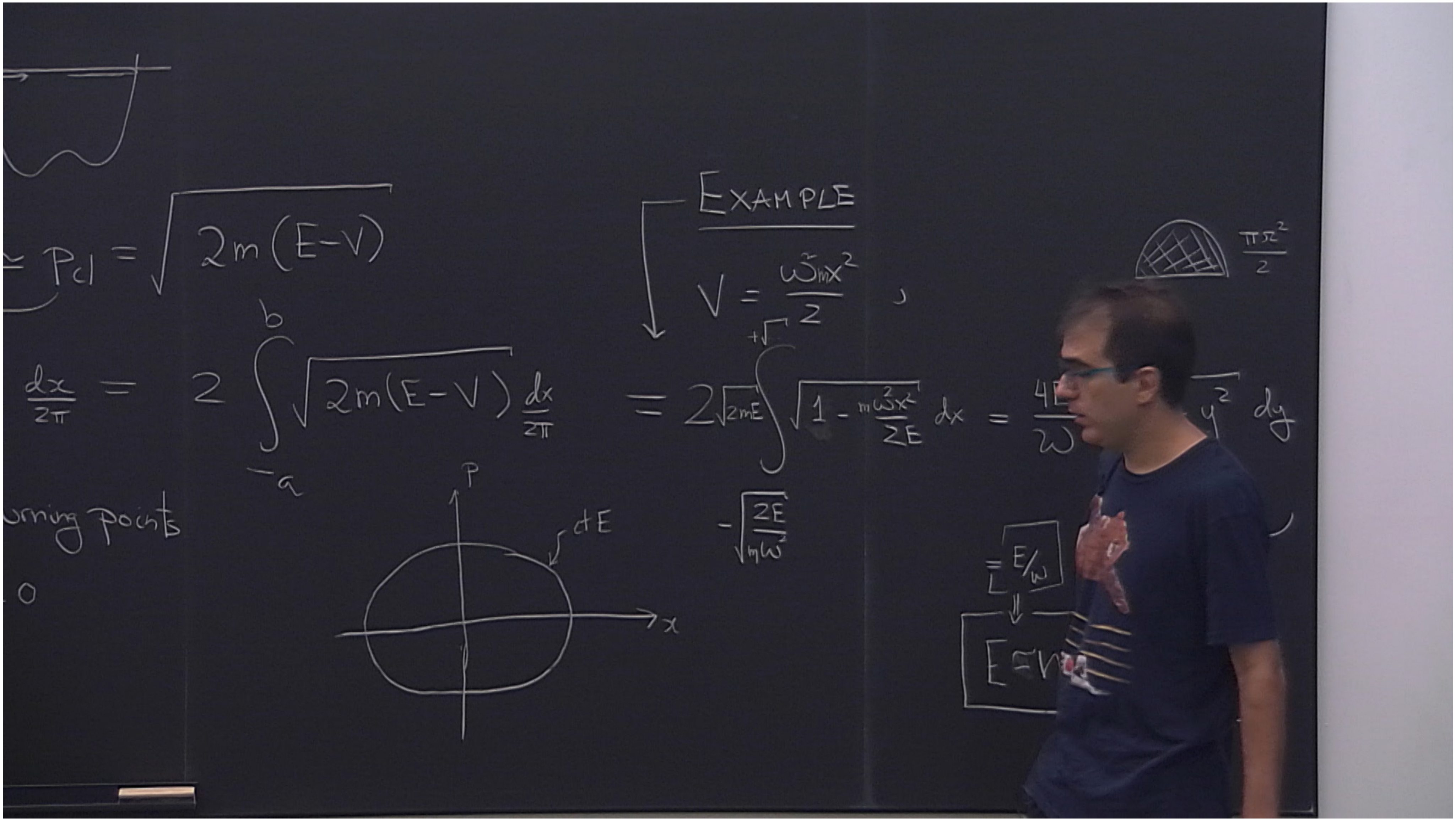
$$V = \frac{\omega^2 m x^2}{2},$$

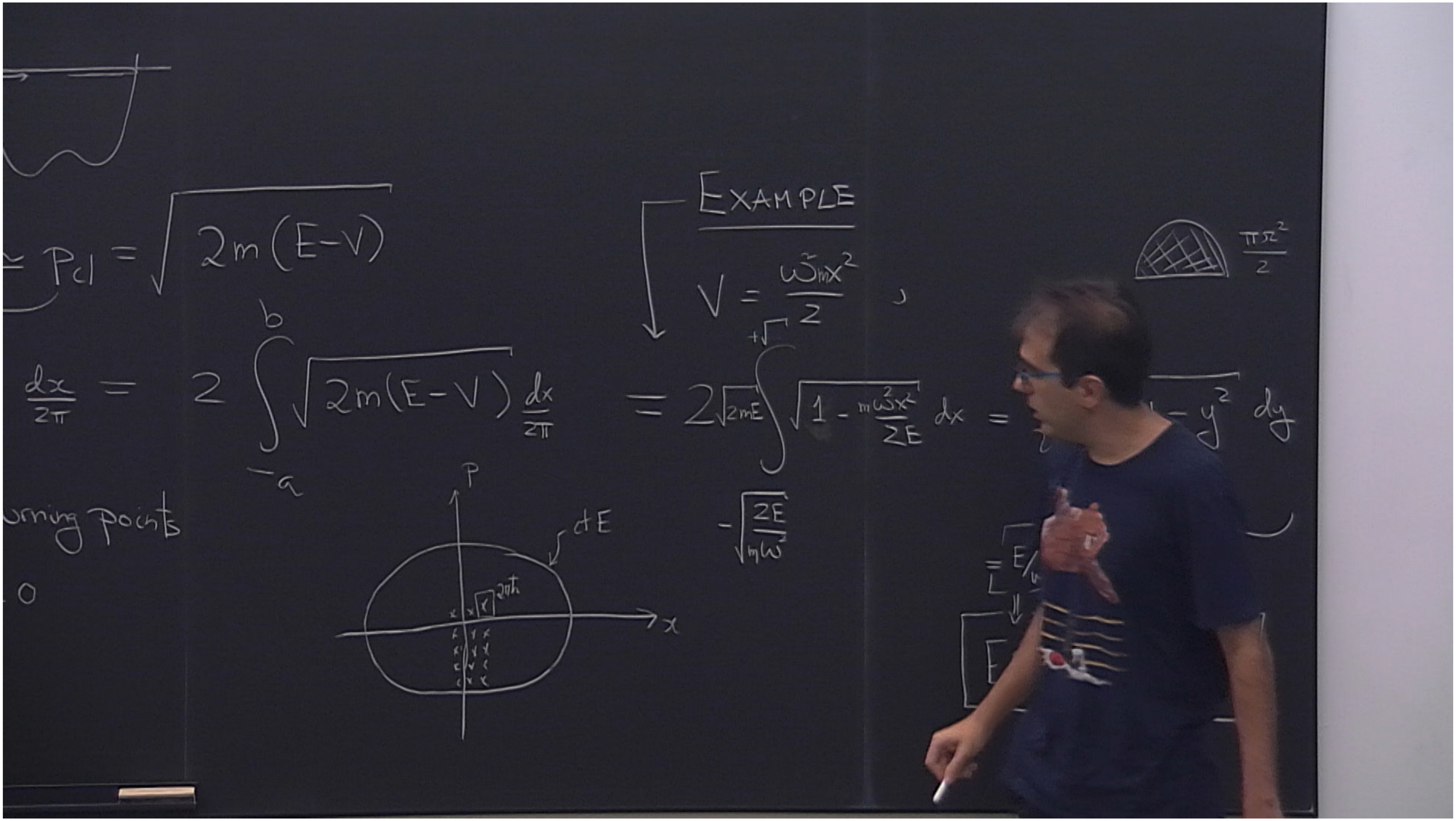
$$= 2\sqrt{2mE}\int_{-\sqrt{\frac{2E}{m\omega^2}}}^{+\sqrt{\ }}\sqrt{1-\frac{m\omega^2 x^2}{2E}}\,dx = \frac{4E}{\omega}\frac{1}{2\pi}\int_{-1}^{1}\sqrt{1-y^2}\,dy$$

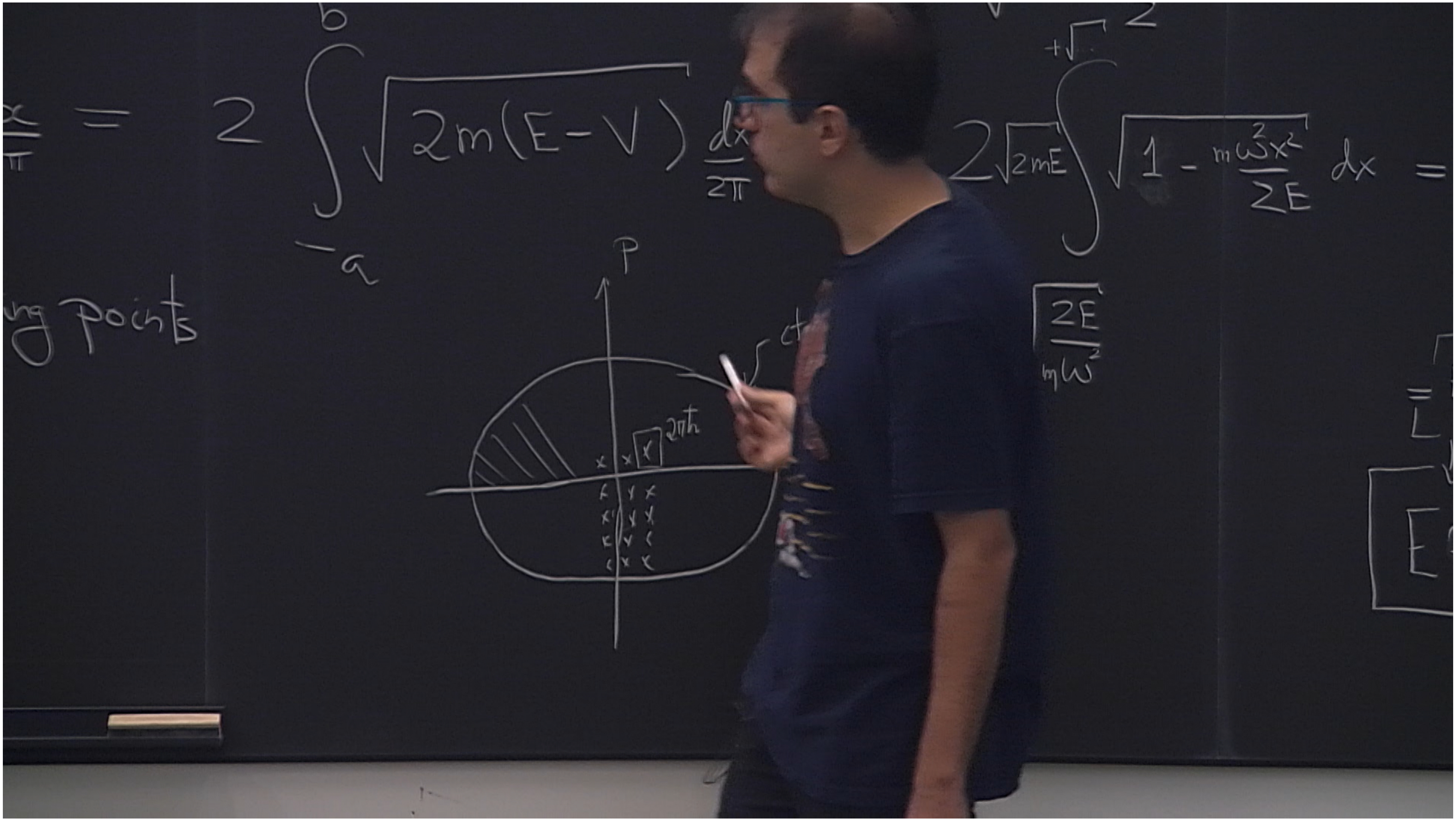
$\frac{\pi r^2}{2}$

$$= \frac{E}{\omega} \qquad \frac{\pi}{2}$$

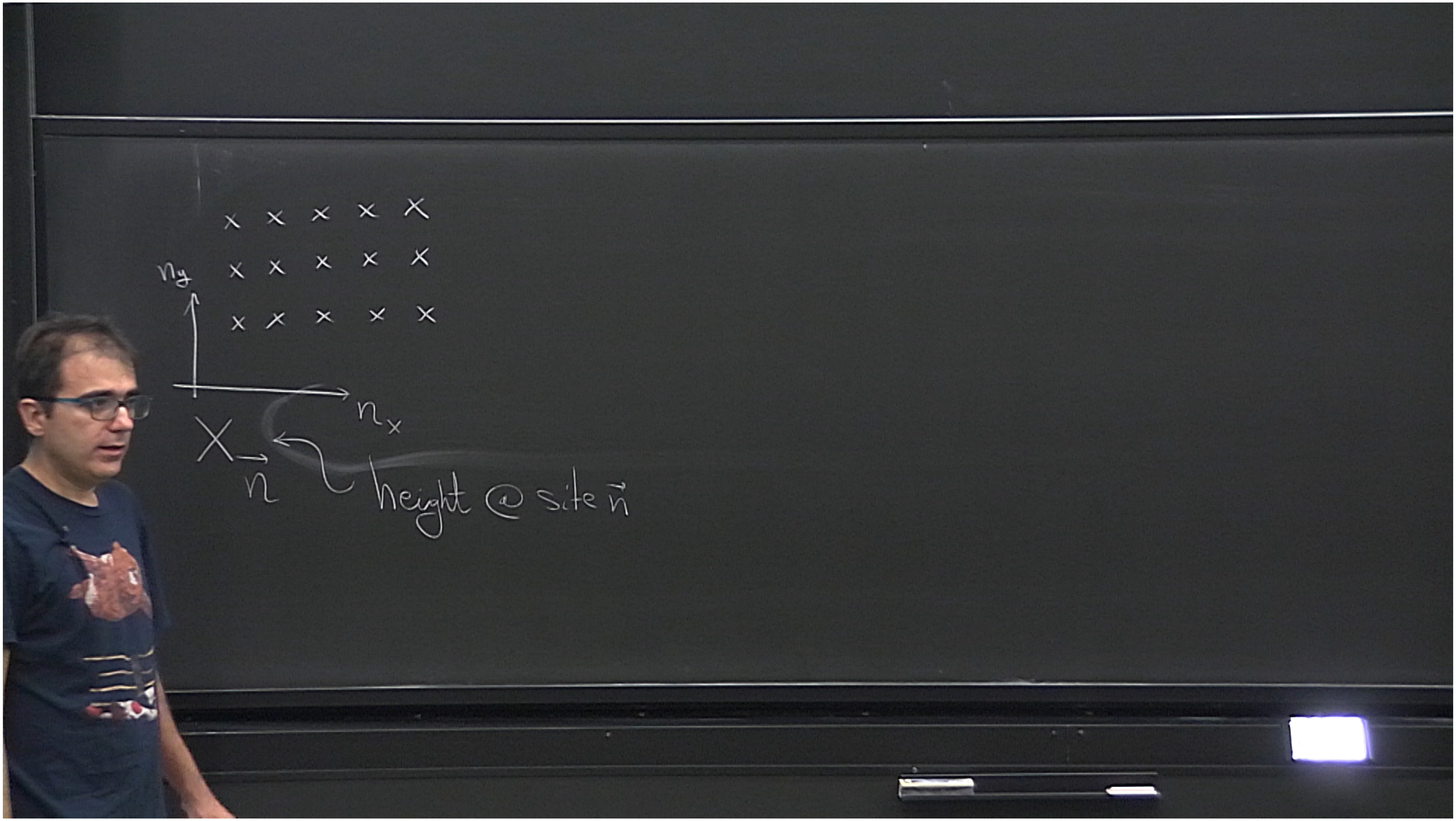
$$E = n\hbar\omega \quad \text{miss } 1/2$$

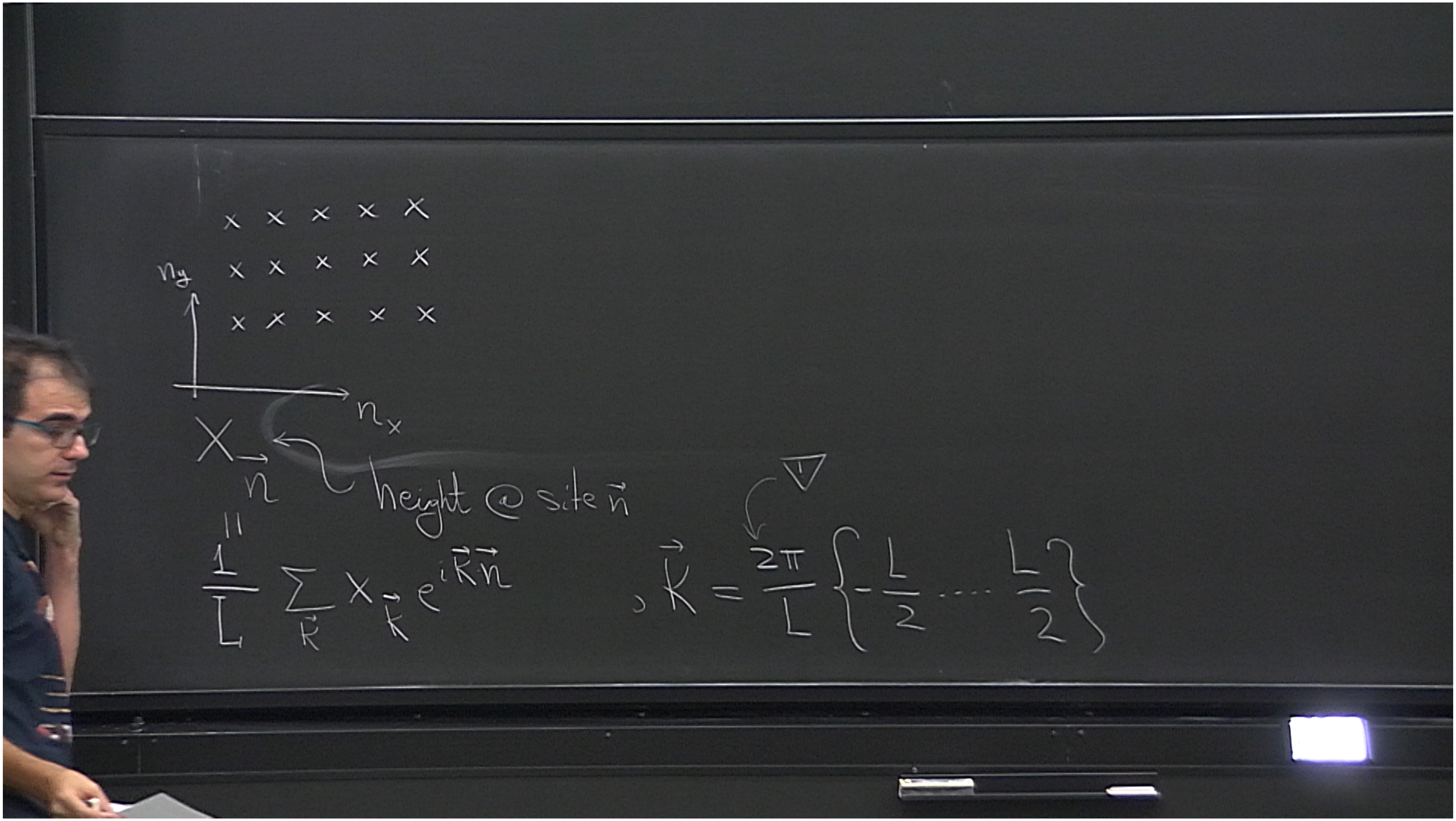

EXAMPLE
$p_{cl} = \sqrt{2m(E-V)}$
$V = \frac{\omega^2 m x^2}{2}$
$\frac{dx}{2\pi} = 2\int_{-a}^{b}\sqrt{2m(E-V)}\,\frac{dx}{2\pi}$
turning points
dE
p
x
$\frac{\pi r^2}{2}$


EXAMPLE
V = ω²mx²/2
p_cl = √(2m(E−V))
dE
P
x



$= 2\int_{-a}^{b}\sqrt{2m(E-V)}\,\frac{dx}{2\pi}$
ng points
p
$2\pi\hbar$
$2\sqrt{2mE}\int\sqrt{1-\frac{m\omega^2x^2}{2E}}\,dx =$
$\sqrt{\frac{2E}{m\omega^2}}$


n_y
n_x
$X_{\vec{n}}$
height @ site $\vec{n}$



n_y
n_x
X_n height @ site n
= 1/L Σ_k X_k e^{ikn}
, k = 2π/L {-L/2 .... L/2}

$$H = \sum_n \frac{P_n^2}{2m} + \frac{m\omega^2}{2} x_n^2 + K/2 \sum_{\vec{l} = \hat{x}, \hat{y}} \left(x_{\vec{n}} - x_{\vec{n}+\vec{l}} \right)^2$$

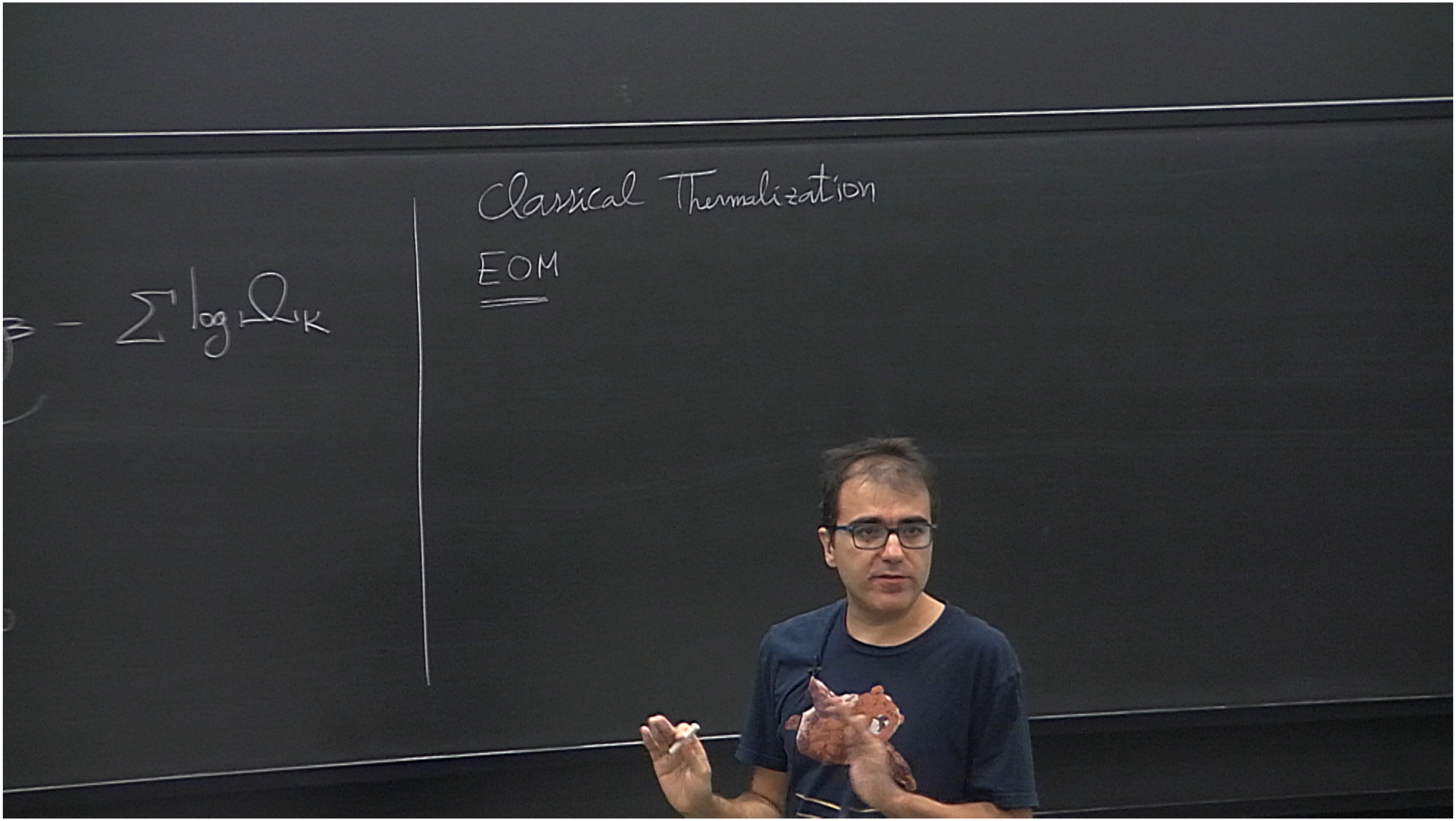
$$= \sum_k |\hat{x}_k|^2 \frac{m\Omega_k^2}{2} + \frac{P_k^2}{2m}, \quad \text{where } \Omega_k^2 = \omega^2 + \frac{8K}{m}\left(\sin^2 \frac{k_x}{2} + \sin^2 \frac{k_y}{2} \right)$$

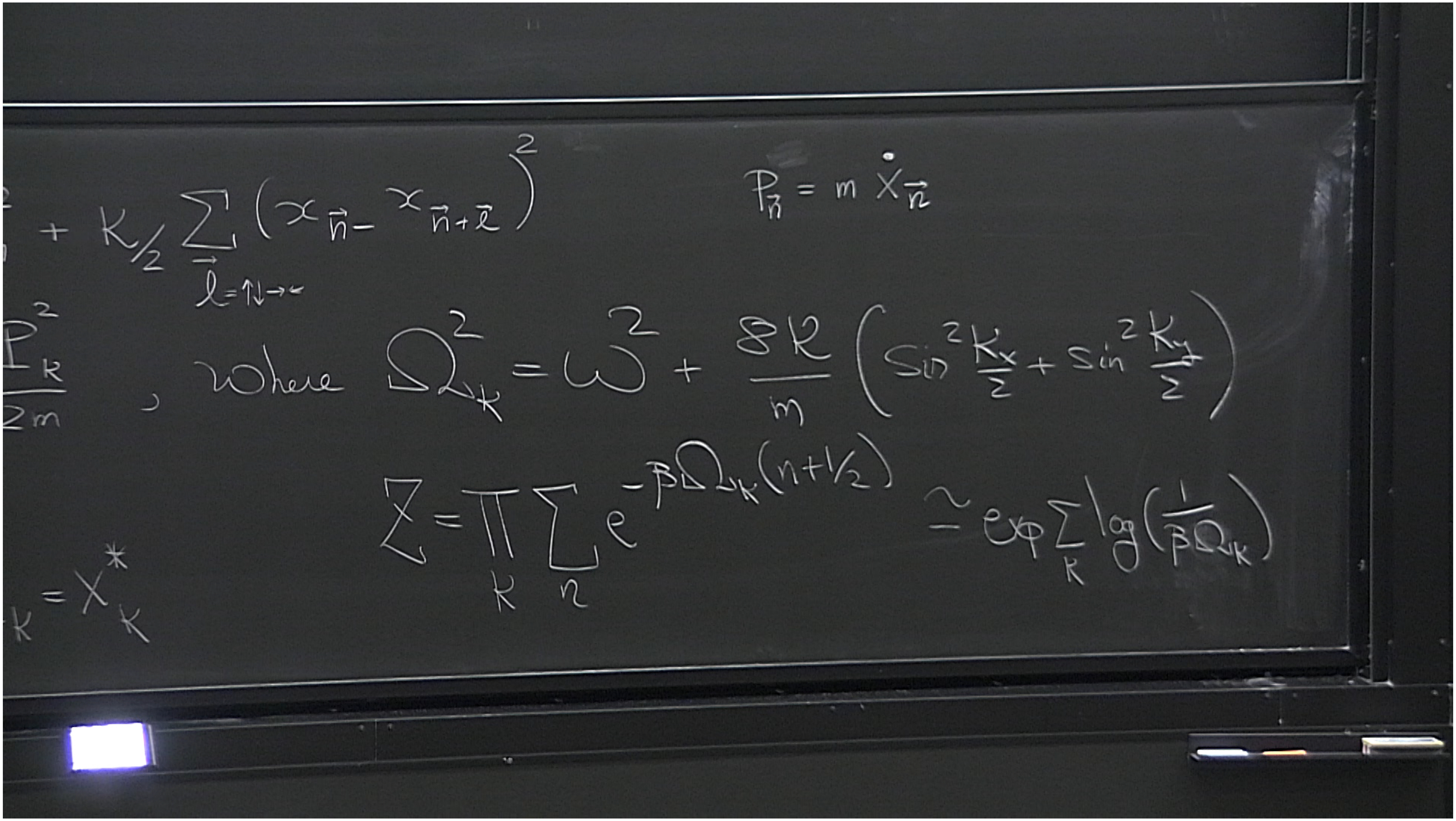
$$\vec{k} = \frac{2\pi}{L} \left\{ -\frac{L}{2} \dots \frac{L}{2} \right\}, \quad x_{-k} = x_k^*$$

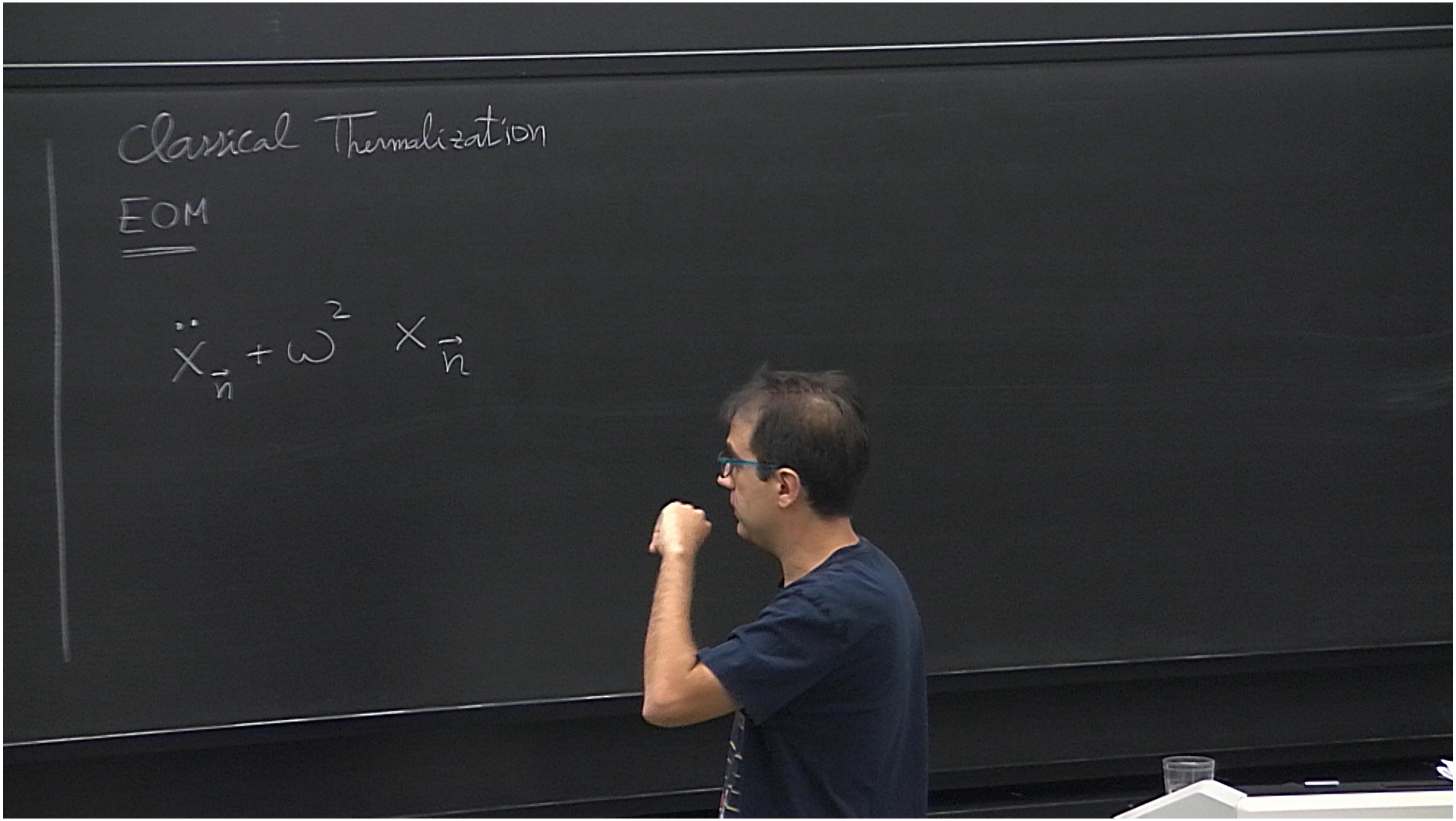
$$Z = \prod_k \sum_n e^{-\beta \Omega_k (n + 1/2)} \simeq \exp \sum_k \log\left(\frac{1}{\beta \Omega_k} \right)$$

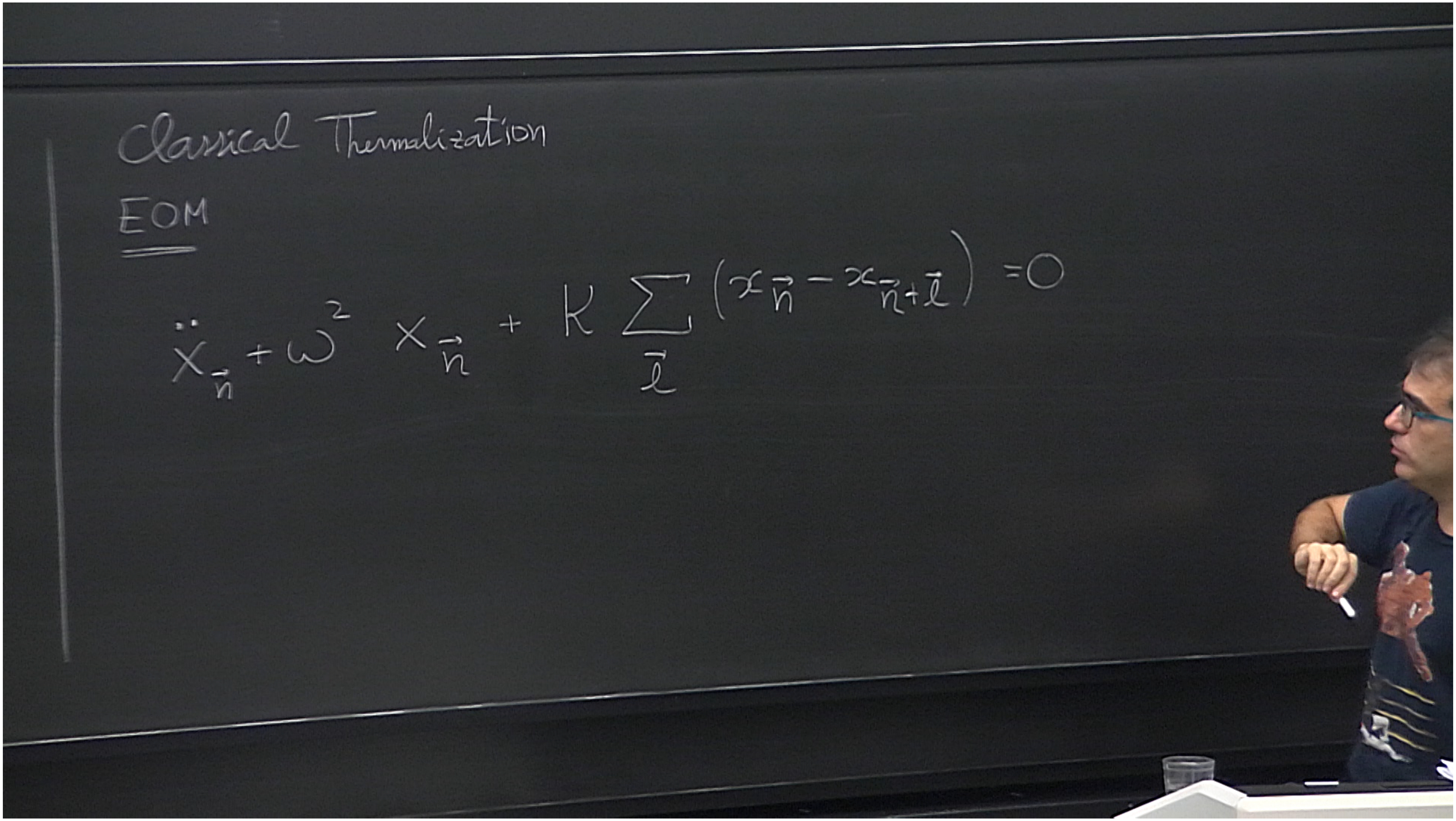
$$\log Z = \underbrace{-L^2 \cdot \log \beta}_{E = L^2/\beta} - \sum' \log \Omega_k$$

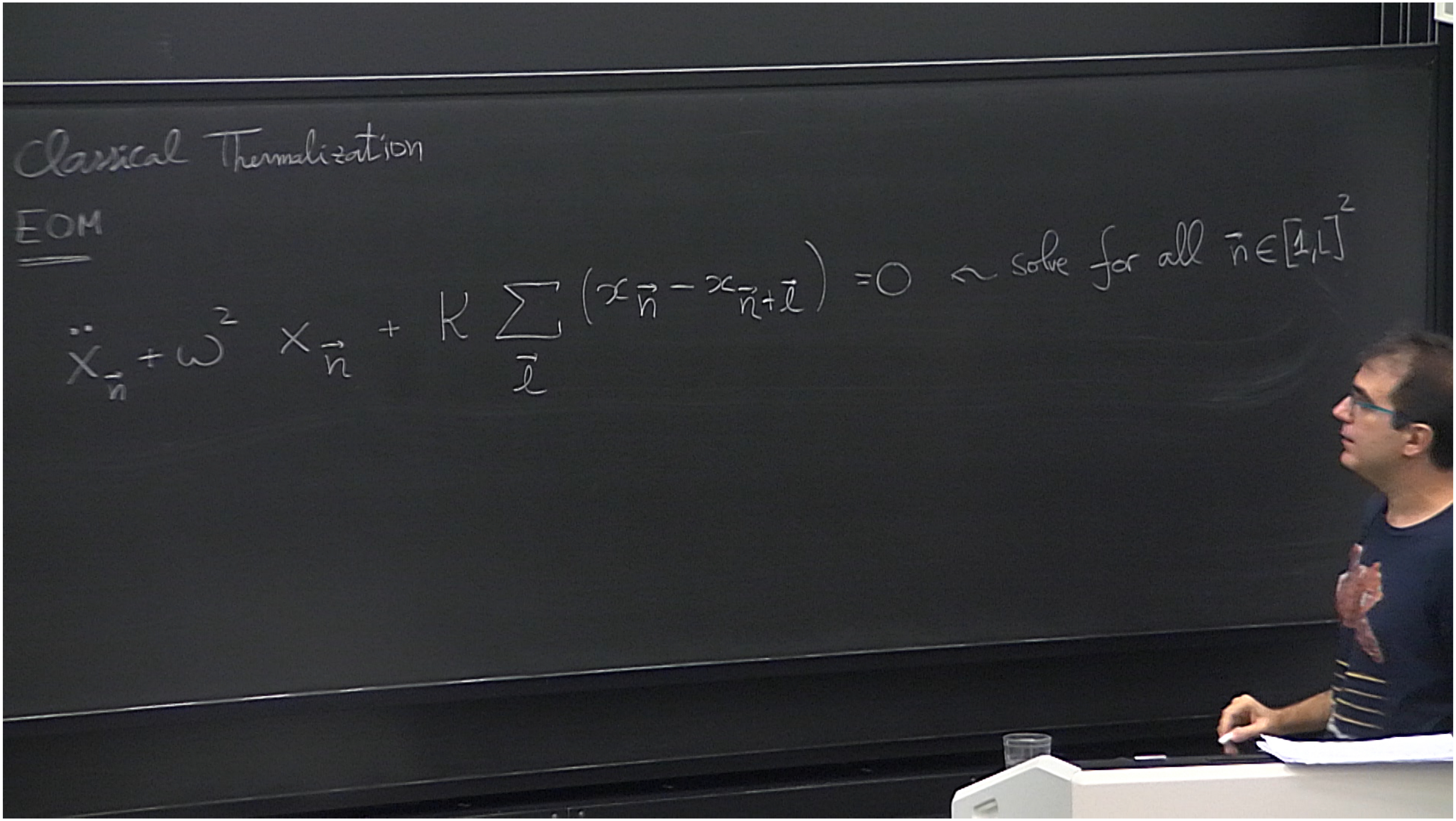
as we knew already.

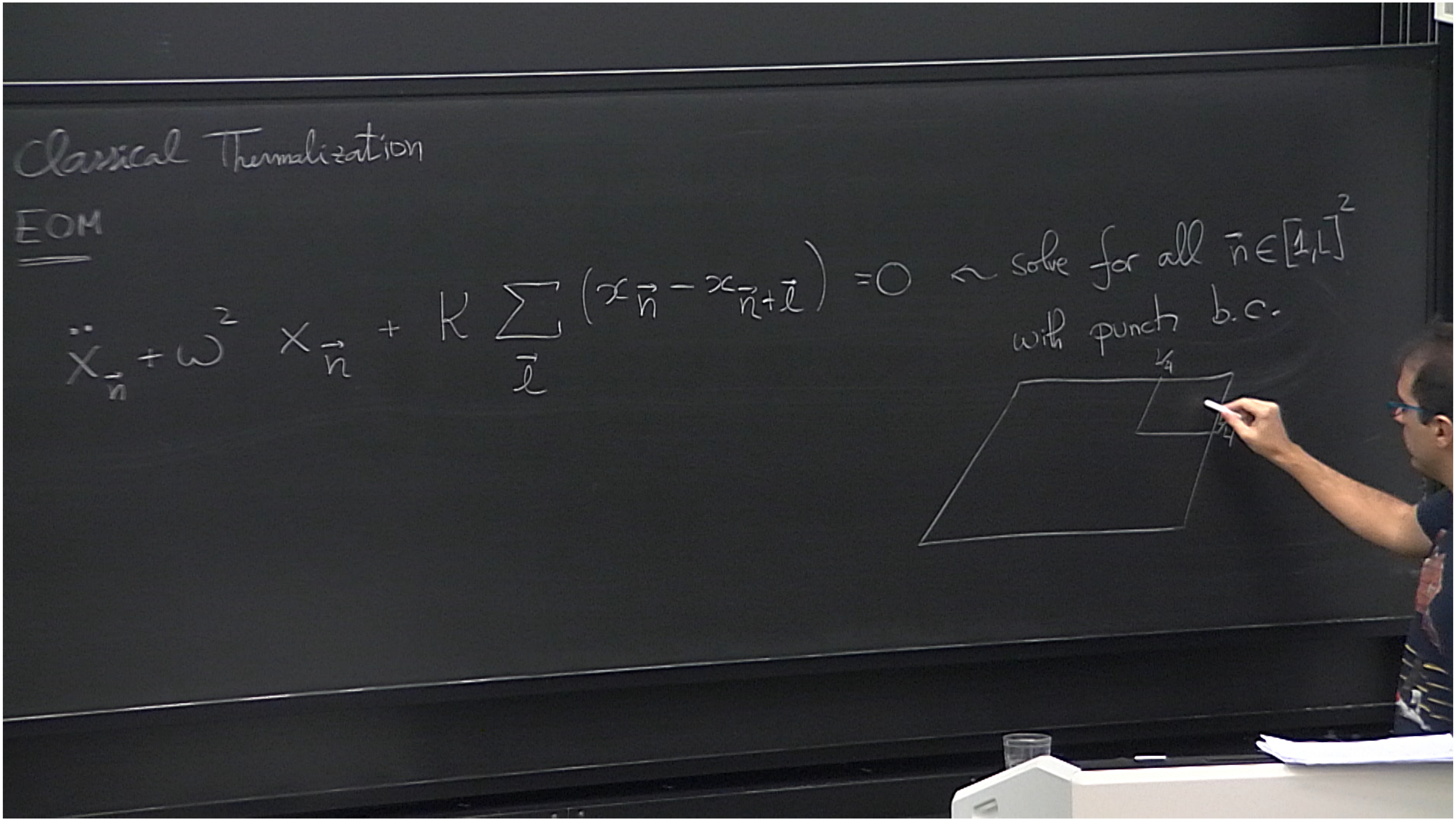

Classical Thermalization
EOM
$\beta - \sum \log \Omega_k$

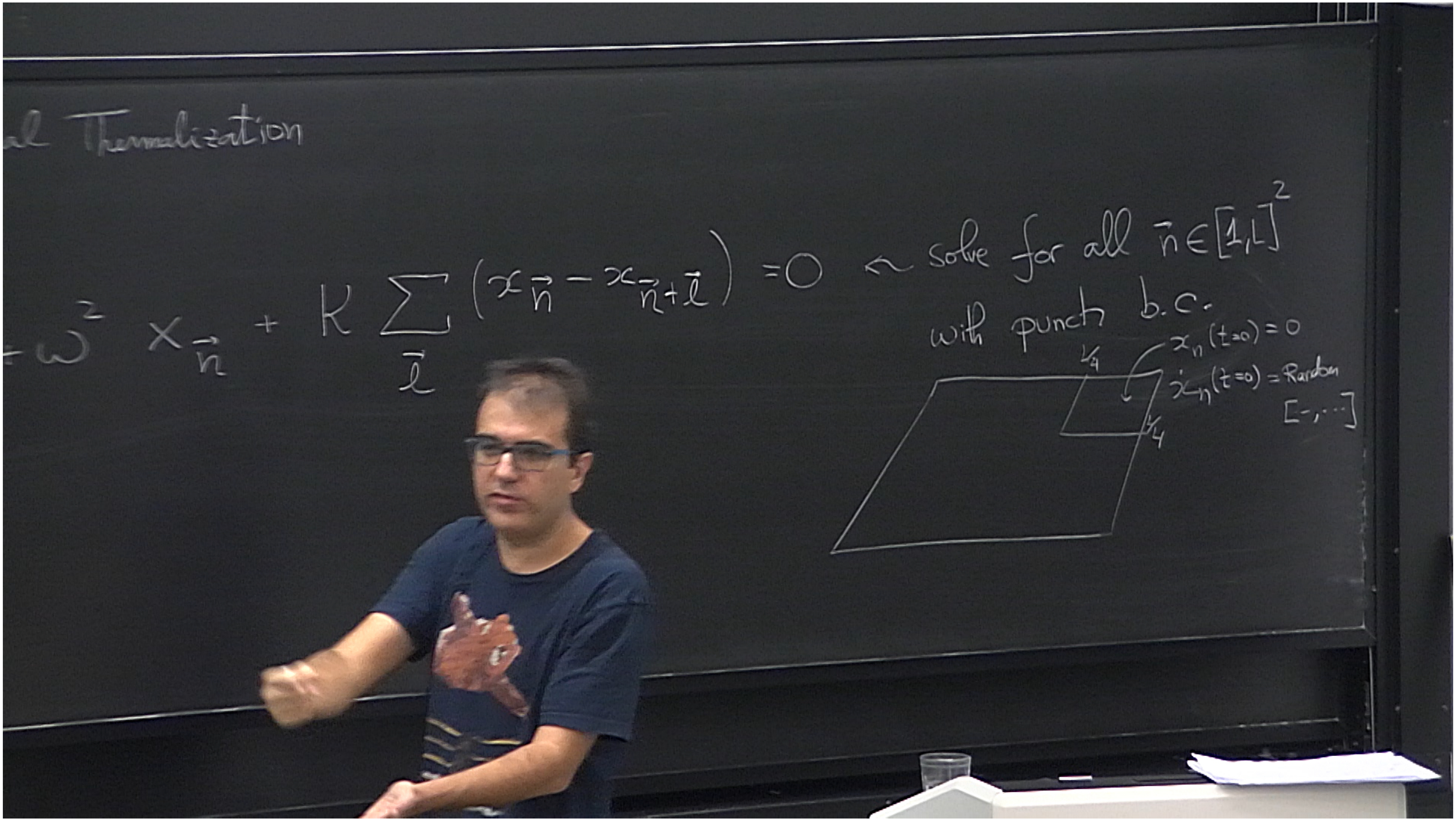

+ K/2 Σ_{l = ↑↓→←} (x_n − x_{n+l})²
P_n = m ẋ_n
P_k² / 2m , where Ω_k² = ω² + 8K/m (sin² k_x/2 + sin² k_y/2)
Z = Π_k Σ_n e^{−βΩ_k(n+1/2)} ≃ exp Σ_k log(1/βΩ_k)
= X_k*


Classical Thermalization
EOM
$\ddot{x}_{\vec{n}} + \omega^2 \; x_{\vec{n}}$


Classical Thermalization
EOM
$\ddot{x}_{\vec{n}} + \omega^2 x_{\vec{n}} + K \sum_{\vec{l}} (x_{\vec{n}} - x_{\vec{n}+\vec{l}}) = 0$


Classical Thermalization
EOM
$\ddot{x}_{\vec{n}} + \omega^2 x_{\vec{n}} + K \sum_{\vec{l}} (x_{\vec{n}} - x_{\vec{n}+\vec{l}}) = 0$ ~ solve for all $\vec{n} \in [1,L]^2$


Classical Thermalization
EOM
$\ddot{x}_{\vec{n}} + \omega^2 x_{\vec{n}} + K \sum_{\vec{l}} (x_{\vec{n}} - x_{\vec{n}+\vec{l}}) = 0$
solve for all $\vec{n} \in [1,L]^2$
with punch b.c.
1/4
3/4


al Thermalization
ω² x_n + K Σ_l (x_n − x_{n+l}) = 0
solve for all n ∈ [1,L]²
with punch b.c.
x_n (t=0) = 0
ẋ_n (t=0) = Random
[−, …]
1/4
1/4

bbjubk

200%

Punching the Table. Time Evolution.

200%

Punching the Table. Time Evolution.

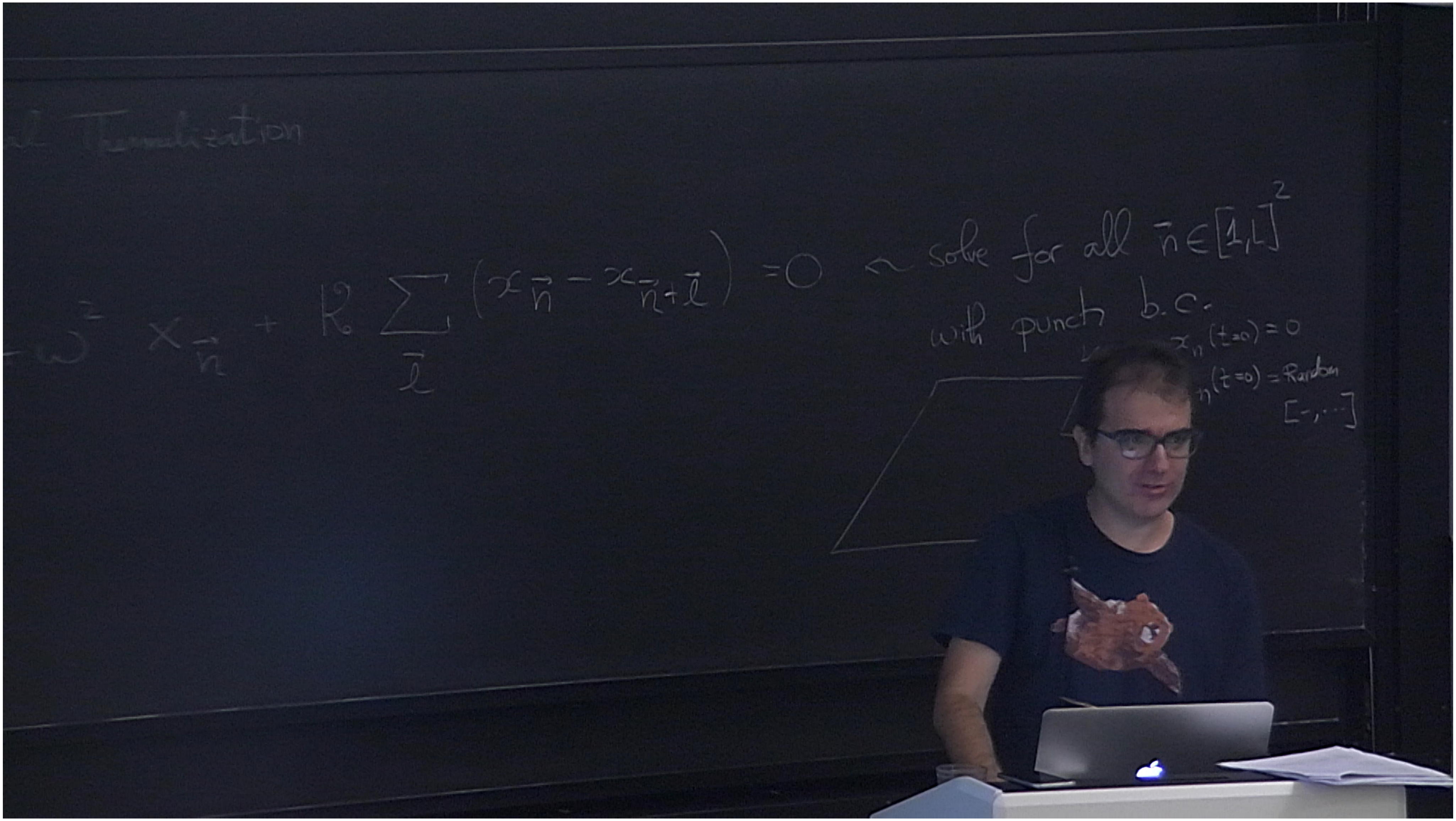
```
eqs = ∂{t,2} x[i, j][t] + ω^2 x[i, j][t] + κ Sum[x[i, j][t] - x[i + l, j][t], {l, -1, 1}]
```

200%

Punching the Table. Time Evolution.

```
eqs = ∂{t,2} x[i, j][t] + ω^2 x[i, j][t] + κ Sum[x[i, j][t] - x[i + l, j][t] + x[i, j][t] - x[i, j + l][t], {l, -1, 1}]
```

200%



Thermalization
solve for all $\vec{n} \in [1,L]^2$
with punch b.c.
$x_n(t=0) = 0$
$(t=0) =$ Random

Punching the Table. Time Evolution.

```
x[i_, j_][t_] /; i > L ⋁ i ≤ 0 ⋁ j > L ⋁ j ≤ 0 = x[Mod[i, L, 1], Mod[j, L, 1]]

eqs =
 Table[∂{t,2} x[i, j][t] + ω^2 x[i, j][t] + κ Sum[x[i, j][t] - x[i + l, j][t] + x[i, j][t] - x[i, j + l][t], {l, -1, 1}] == 0,
  {i, L}, {j, L}]
```

200%

Punching the Table. Time Evolution.

```
L = 60; κ = 5; ω = 2;
x[i_, j_][t_] /; i > L ⋁ i ≤ 0 ⋁ j > L ⋁ j ≤ 0 = x[Mod[i, L, 1], Mod[j, L, 1]]

eqs =
 Table[∂{t,2} x[i, j][t] + ω^2 x[i, j][t] + κ Sum[x[i, j][t] - x[i + l, j][t] + x[i, j][t] - x[i, j + l][t], {l, -1, 1}] == 0,
  {i, L}, {j, L}]
```

200%

Punching the Table. Time Evolution.

In[2591]:=

```
L = 60; ϰ = 5; ω = 2;
x[i_, j_][t_] /; i > L ⋁ i ≤ 0 ⋁ j > L ⋁ j ≤ 0 := x[Mod[i, L, 1], Mod[j, L, 1]]
```

In[2594]:=

```
x[25, 85][t]
```

Out[2594]=

```
x[25, 25][t]
```

```
eqs =
 Table[∂{t,2} x[i, j][t] + ω^2 x[i, j][t] + ϰ Sum[x[i, j][t] - x[i + l, j][t] + x[i, j][t] - x[i, j + l][t], {l, -1, 1}] == 0,
  {i, L}, {j, L}]
```

Punching the Table. Time Evolution.

In[2591]:=

```
L = 60; κ = 5; ω = 2;
x[i_, j_][t_] /; i > L ⋁ i ≤ 0 ⋁ j > L ⋁ j ≤ 0 := x[Mod[i, L, 1], Mod[j, L, 1]]
```

In[2595]:=

```
x[-25, 85][t]
```

Out[2595]=

```
x[35, 25][t]
```

```
eqs =
 Table[∂{t,2} x[i, j][t] + ω^2 x[i, j][t] + κ Sum[x[i, j][t] - x[i + l, j][t] + x[i, j][t] - x[i, j + l][t], {l, -1, 1}] == 0,
  {i, L}, {j, L}]
```

Punching the Table. Time Evolution.

In[2591]:=

```
L = 60; κ = 5; ω = 2;
x[i_, j_][t_] /; i > L ⋁ i ≤ 0 ⋁ j > L ⋁ j ≤ 0 := x[Mod[i, L, 1], Mod[j, L, 1]]
```

In[2596]:=

```
eqs =
 Table[∂{t,2} x[i, j][t] + ω^2 x[i, j][t] + κ Sum[x[i, j][t] - x[i + l, j][t] + x[i, j][t] - x[i, j + l][t], {l, -1, 1}] == 0,
  {i, L}, {j, L}]
```

Out[2596]=

```
{{4 x[1, 1][t] + 5 (4 x[1, 1][t] - x[1, 2][t] - x[1, 60][t] - x[2, 1][t] - x[60, 1][t]) + x[1, 1]''[t] == 0,
  ⋯1⋯ == 0, ⋯56⋯, ⋯1⋯ == 0, 4 x[1, 60][t] + 5 (⋯1⋯) + x[1, 60]''[t] == 0}, ⋯58⋯, {⋯1⋯}}
```

large output | show less | show more | show all | set size limit...

200%

Punching the Table. Time Evolution.

In[2591]:=

```
L = 60; ϰ = 5; ω = 2;
x[i_, j_][t_] /; i > L ⋁ i ≤ 0 ⋁ j > L ⋁ j ≤ 0 := x[Mod[i, L, 1], Mod[j, L, 1]]

eqs =
  Table[∂{t,2} x[i, j][t] + ω^2 x[i, j][t] + ϰ Sum[x[i, j][t] - x[i + l, j][t] + x[i, j][t] - x[i, j + l][t], {l, -1, 1}] == 0,
   {i, L}, {j, L}];
```

In[2598]:=

```
bc = Table[{x[i, j][0] == 0, x[i, j]'[0] == If[i > 3 L / 4 ⋀ j > 3 L / 4, RandomReal[{4, 6}], 0]}, {i, L}, {j, L}] // Flatten
```

Out[2598]=

```
{0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
```

200%

```
L = 60; κ = 5; ω = 2;
x[i_, j_][t_] /; i > L ∨ i ≤ 0 ∨ j > L ∨ j ≤ 0 := x[Mod[i, L, 1], Mod[j, L, 1]]
```

In[2603]:=

```
eqs =
  Table[∂{t,2} x[i, j][t] + ω^2 x[i, j][t] + κ Sum[x[i, j][t] - x[i + l, j][t] + x[i, j][t] - x[i, j + l][t], {l, -1, 1}] == 0,
   {i, L}, {j, L}];
```

In[2604]:=

```
bc = Table[{x[i, j][0] == 0, x[i, j]'[0] == If[i > 3 L / 4 ∧ j > 3 L / 4, RandomReal[{4, 6}], 0]}, {i, L}, {j, L}] //
  Flatten
```

Out[2604]=

```
{x[1, 1][0] == 0, True, x[1, 2][0] == 0, True, x[1, 3][0] == 0, True, x[1, 4][0] == 0, True, ... 7184 ...,
 x[60, 57][0] == 0, False, x[60, 58][0] == 0, False, x[60, 59][0] == 0, False, x[60, 60][0] == 0, False}
```

large output | show less | show more | show all | set size limit...

200%

Punching the Table. Time Evolution.

In[1]:=
```
ClearAll[x]
L = 60; ϰ = 5; ω = 2;
x[i_, j_][t_] /; i > L ∨ i ≤ 0 ∨ j > L ∨ j ≤ 0 := x[Mod[i, L, 1], Mod[j, L, 1]]
```

In[4]:=
```
eqs =
  Table[∂{t,2} x[i, j][t] + ω^2 x[i, j][t] + ϰ Sum[x[i, j][t] - x[i + l, j][t] + x[i, j][t] - x[i, j + l][t], {l, -1, 1}] == 0,
   {i, L}, {j, L}];
```

In[5]:=
```
bc = Table[{x[i, j][0] == 0, x[i, j]'[0] == If[i > 3 L / 4 ∧ j > 3 L / 4, RandomReal[{4, 6}], 0]}, {i, L}, {j, L}] //
  Flatten
```

Out[5]=
```
{x[1, 1][0] == 0, x[1, 1]′[0] == 0, x[1, 2][0] == 0, x[1, 2]′[0] == 0, x[1, 3][0] == 0,
 ··· 7191 ···, x[60, 59][0] == 0, x[60, 59]′[0] == 5.53107, x[60, 60][0] == 0, x[60, 60]′[0] == 4.08415}
```

large output | show less | show more | show all | set size limit...

200%

Punching the Table. Time Evolution.

In[1]:=
```
ClearAll[x]
L = 60; κ = 5; ω = 2;
x[i_, j_][t_] /; i > L ∨ i ≤ 0 ∨ j > L ∨ j ≤ 0 := x[Mod[i, L, 1], Mod[j, L, 1]]
```

In[4]:=
```
eqs =
  Table[∂{t,2} x[i, j][t] + ω^2 x[i, j][t] + κ Sum[x[i, j][t] - x[i + l, j][t] + x[i, j][t] - x[i, j + l][t], {l, -1, 1}] == 0,
   {i, L}, {j, L}];

bc = Table[{x[i, j][0] == 0, x[i, j]'[0] == If[i > 3 L / 4 ∧ j > 3 L / 4, RandomReal[{4, 6}], 0]}, {i, L}, {j, L}] //
   Flatten;
```

In[6]:=
```
var = Table[x[i, j][t], {i, L}, {j, L}] // Flatten
```

Out[6]=
```
{x[1, 1][t], x[1, 2][t], x[1, 3][t], x[1, 4][t], x[1, 5][t], x[1, 6][t], x[1, 7][t], ··· 3586 ···,
 x[60, 54][t], x[60, 55][t], x[60, 56][t], x[60, 57][t], x[60, 58][t], x[60, 59][t], x[60, 60][t]}
```

large output | show less | show more | show all | set size limit...

200%

Punching the Table. Time Evolution.

```
In[1]:= ClearAll[x]
        L = 60; κ = 5; ω = 2;
        x[i_, j_][t_] /; i > L ⋁ i ≤ 0 ⋁ j > L ⋁ j ≤ 0 := x[Mod[i, L, 1], Mod[j, L, 1]]

In[4]:= eqs =
          Table[∂{t,2} x[i, j][t] + ω^2 x[i, j][t] + κ Sum[x[i, j][t] - x[i + l, j][t] + x[i, j][t] - x[i, j + l][t], {l, -1, 1}] == 0,
           {i, L}, {j, L}];

        bc = Table[{x[i, j][0] == 0, x[i, j]'[0] == If[i > 3 L / 4 ⋀ j > 3 L / 4, RandomReal[{4, 6}], 0]}, {i, L}, {j, L}] //
           Flatten;

        var = Table[x[i, j][t], {i, L}, {j, L}] // Flatten;

In[7]:= var // Length

Out[7]= 3600
```

Punching the Table. Time Evolution.

In[8]:=
```
ClearAll[x]
L = 60; ϰ = 5; ω = 2; tmax = 500;
x[i_, j_][t_] /; i > L ⋁ i ≤ 0 ⋁ j > L ⋁ j ≤ 0 := x[Mod[i, L, 1], Mod[j, L, 1]]
```

In[4]:=
```
eqs =
  Table[∂{t,2} x[i, j][t] + ω^2 x[i, j][t] + ϰ Sum[x[i, j][t] - x[i + l, j][t] + x[i, j][t] - x[i, j + l][t], {l, -1, 1}] == 0,
   {i, L}, {j, L}];

bc = Table[{x[i, j][0] == 0, x[i, j]'[0] == If[i > 3 L / 4 ⋀ j > 3 L / 4, RandomReal[{4, 6}], 0]}, {i, L}, {j, L}] //
   Flatten;

var = Table[x[i, j][t], {i, L}, {j, L}] // Flatten;
```

Punching the Table. Time Evolution.

```
In[8]:= ClearAll[x]
L = 60; ϰ = 5; ω = 2; tmax = 500;
x[i_, j_][t_] /; i > L ∨ i ≤ 0 ∨ j > L ∨ j ≤ 0 := x[Mod[i, L, 1], Mod[j, L, 1]]

In[4]:= eqs =
  Table[∂{t,2} x[i, j][t] + ω^2 x[i, j][t] + ϰ Sum[x[i, j][t] - x[i + l, j][t] + x[i, j][t] - x[i, j + l][t], {l, -1, 1}] == 0,
   {i, L}, {j, L}];

bc = Table[{x[i, j][0] == 0, x[i, j]'[0] == If[i > 3 L / 4 ∧ j > 3 L / 4, RandomReal[{4, 6}], 0]}, {i, L}, {j, L}] //
   Flatten;

var = Table[x[i, j][t], {i, L}, {j, L}] // Flatten;

In[11]:= solution = NDSolve[eqs, bc, var, {t, 0, tmax}][[1]];
```

··· NDSolve: x[1, 1][0] == 0 cannot be used as a function.

Punching the Table. Time Evolution.

```
In[8]:= ClearAll[x]
        L = 60; ϰ = 5; ω = 2; tmax = 500;
        x[i_, j_][t_] /; i > L ⋁ i ≤ 0 ⋁ j > L ⋁ j ≤ 0 := x[Mod[i, L, 1], Mod[j, L, 1]]

In[12]:= eqs =
          Table[∂{t,2} x[i, j][t] + ω^2 x[i, j][t] + ϰ Sum[x[i, j][t] - x[i + l, j][t] + x[i, j][t] - x[i, j + l][t], {l, -1, 1}] == 0,
            {i, L}, {j, L}] // Flatten;

In[13]:= bc = Table[{x[i, j][0] == 0, x[i, j]'[0] == If[i > 3 L / 4 ⋀ j > 3 L / 4, RandomReal[{4, 6}], 0]}, {i, L}, {j, L}] //
           Flatten;

In[14]:= var = Table[x[i, j][t], {i, L}, {j, L}] // Flatten;

In[11]:= solution = NDSolve[eqs, bc, var, {t, 0, tmax}][[1]];

   ⋯ NDSolve: x[1, 1][0] == 0 cannot be used as a function.
```

200%

Punching the Table. Time Evolution.

```
In[8]:= ClearAll[x]
        L = 60; ϰ = 5; ω = 2; tmax = 500;
        x[i_, j_][t_] /; i > L ⋁ i ≤ 0 ⋁ j > L ⋁ j ≤ 0 := x[Mod[i, L, 1], Mod[j, L, 1]]

In[12]:= eqs =
          Table[∂{t,2} x[i, j][t] + ω^2 x[i, j][t] + ϰ Sum[x[i, j][t] - x[i + l, j][t] + x[i, j][t] - x[i, j + l][t], {l, -1, 1}] == 0,
            {i, L}, {j, L}] // Flatten;

In[13]:= bc = Table[{x[i, j][0] == 0, x[i, j]'[0] == If[i > 3 L / 4 ⋀ j > 3 L / 4, RandomReal[{4, 6}], 0]}, {i, L}, {j, L}] //
           Flatten;

In[14]:= var = Table[x[i, j][t], {i, L}, {j, L}] // Flatten;

         solution = NDSolve[Join[eqs, bc], var, {t, 0, tmax}][[1]];

         ··· NDSolve: x[1, 1][0] == 0 cannot be used as a function.
```

200%

Punching the Table. Time Evolution.

```
In[17]:= ClearAll[x]
         L = 60; ϰ = 5; ω = 2; tmax = 500;
         x[i_, j_][t_] /; i > L ∨ i ≤ 0 ∨ j > L ∨ j ≤ 0 := x[Mod[i, L, 1], Mod[j, L, 1]][t]

In[20]:= eqs =
           Table[∂{t,2} x[i, j][t] + ω^2 x[i, j][t] + ϰ Sum[x[i, j][t] - x[i + l, j][t] + x[i, j][t] - x[i, j + l][t], {l, -1, 1}] == 0,
             {i, L}, {j, L}] // Flatten;

In[21]:= bc = Table[{x[i, j][0] == 0, x[i, j]'[0] == If[i > 3 L / 4 ∧ j > 3 L / 4, RandomReal[{4, 6}], 0]}, {i, L}, {j, L}] //
           Flatten;

In[22]:= var = Table[x[i, j][t], {i, L}, {j, L}] // Flatten;

In[23]:= solution = NDSolve[Join[eqs, bc], var, {t, 0, tmax}][[1]];

         ··· NDSolve: The function x[60, 1] appears with no arguments.
```

Punching the Table. Time Evolution.

```
In[17]:= ClearAll[x]
         L = 60; κ = 5; ω = 2; tmax = 500;
         x[i_, j_][t_] /; i > L ∨ i ≤ 0 ∨ j > L ∨ j ≤ 0 := x[Mod[i, L, 1], Mod[j, L, 1]][t]

In[20]:= eqs =
           Table[∂{t,2} x[i, j][t] + ω^2 x[i, j][t] + κ Sum[x[i, j][t] - x[i + l, j][t] + x[i, j][t] - x[i, j + l][t], {l, -1, 1}] == 0,
             {i, L}, {j, L}] // Flatten;

In[21]:= bc = Table[{x[i, j][0] == 0, x[i, j]'[0] == If[i > 3 L / 4 ∧ j > 3 L / 4, RandomReal[{4, 6}], 0]}, {i, L}, {j, L}] //
           Flatten;

In[22]:= var = Table[x[i, j][t], {i, L}, {j, L}] // Flatten;

In[23]:= solution = NDSolve[Join[eqs, bc], var, {t, 0, tmax}][[1]];

In[24]:= var

Out[24]=
  {x[1, 1][t], x[1, 2][t], x[1, 3][t], x[1, 4][t], x[1, 5][t], x[1, 6][t], x[1, 7][t], ··· 3586 ···,
   x[60, 54][t], x[60, 55][t], x[60, 56][t], x[60, 57][t], x[60, 58][t], x[60, 59][t], x[60, 60][t]}

  large output   show less   show more   show all   set size limit...
```

Punching the Table. Time Evolution.

```
In[17]:= ClearAll[x]
L = 60; ϰ = 5; ω = 2; tmax = 500;
x[i_, j_][t_] /; i > L ∨ i ≤ 0 ∨ j > L ∨ j ≤ 0 := x[Mod[i, L, 1], Mod[j, L, 1]][t]

In[20]:= eqs =
  Table[∂{t,2} x[i, j][t] + ω^2 x[i, j][t] + ϰ Sum[x[i, j][t] - x[i + l, j][t] + x[i, j][t] - x[i, j + l][t], {l, -1, 1}] == 0,
    {i, L}, {j, L}] // Flatten;

In[21]:= bc = Table[{x[i, j][0] == 0, x[i, j]'[0] == If[i > 3 L / 4 ∧ j > 3 L / 4, RandomReal[{4, 6}], 0]}, {i, L}, {j, L}] //
   Flatten;

In[22]:= var = Table[x[i, j][t], {i, L}, {j, L}] // Flatten;

In[23]:= solution = NDSolve[Join[eqs, bc], var, {t, 0, tmax}][[1]];

In[25]:= var /. x[i_, j_][t] :> {i, j, x[i, j, t]}
Out[25]=
{{1, 1, x[1, 1, t]}, {1, 2, x[1, 2, t]}, {1, 3, x[1, 3, t]}, {1, 4, x[1, 4, t]},
 {1, 5, x[1, 5, t]}, … 3590 …, {60, 56, x[60, 56, t]}, {60, 57, x[60, 57, t]},
 {60, 58, x[60, 58, t]}, {60, 59, x[60, 59, t]}, {60, 60, x[60, 60, t]}}

large output   show less   show more   show all   set size limit...
```

200%

Punching the Table: Time Evolution.

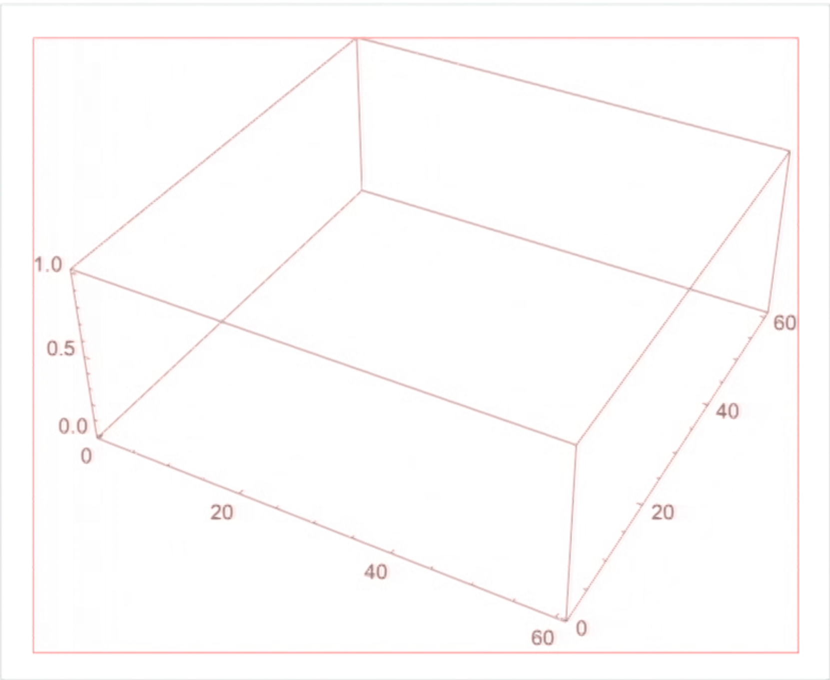
```
In[17]:= ClearAll[x]
L = 60; ϰ = 5; ω = 2; tmax = 500;
x[i_, j_][t_] /; i > L ∨ i ≤ 0 ∨ j > L ∨ j ≤ 0 := x[Mod[i, L, 1], Mod[j, L, 1]][t]

In[20]:= eqs =
  Table[∂{t,2} x[i, j][t] + ω^2 x[i, j][t] + ϰ Sum[x[i, j][t] - x[i + l, j][t] + x[i, j][t] - x[i, j + l][t], {l, -1, 1}] == 0,
    {i, L}, {j, L}] // Flatten;

In[21]:= bc = Table[{x[i, j][0] == 0, x[i, j]'[0] == If[i > 3 L / 4 ∧ j > 3 L / 4, RandomReal[{4, 6}], 0]}, {i, L}, {j, L}] //
    Flatten;

In[22]:= var = Table[x[i, j][t], {i, L}, {j, L}] // Flatten;

In[23]:= solution = NDSolve[Join[eqs, bc], var, {t, 0, tmax}][[1]];

var /. x[i_, j_][t] :> {i, j, x[i, j, t]} /. solution;

Out[25]=

{{1, 1, x[1, 1, t]}, {1, 2, x[1, 2, t]}, {1, 3, x[1, 3, t]}, {1, 4, x[1, 4, t]},
 {1, 5, x[1, 5, t]}, ..., {60, 56, x[60, 56, t]}, {60, 57, x[60, 57, t]},
 {60, 58, x[60, 58, t]}, {60, 59, x[60, 59, t]}, {60, 60, x[60, 60, t]}}

large output    show less    show more    show all    set size limit...
```

```
In[17]:= ClearAll[x]
         L = 60; κ = 5; ω = 2; tmax = 500;
         x[i_, j_][t_] /; i > L ∨ i ≤ 0 ∨ j > L ∨ j ≤ 0 := x[Mod[i, L, 1], Mod[j, L, 1]][t]

In[20]:= eqs =
           Table[∂{t,2} x[i, j][t] + ω^2 x[i, j][t] + κ Sum[x[i, j][t] - x[i + l, j][t] + x[i, j][t] - x[i, j + l][t], {l, -1, 1}] == 0,
             {i, L}, {j, L}] // Flatten;

In[21]:= bc = Table[{x[i, j][0] == 0, x[i, j]'[0] == If[i > 3 L / 4 ∧ j > 3 L / 4, RandomReal[{4, 6}], 0]}, {i, L}, {j, L}] //
           Flatten;

In[22]:= var = Table[x[i, j][t], {i, L}, {j, L}] // Flatten;

In[23]:= solution = NDSolve[Join[eqs, bc], var, {t, 0, tmax}][[1]];

In[26]:= ToPlot[t_] = var /. x[i_, j_][t] :→ {i, j, x[i, j, t]} /. solution;

Out[25]=
    {{1, 1, x[1, 1, t]}, {1, 2, x[1, 2, t]}, {1, 3, x[1, 3, t]}, {1, 4, x[1, 4, t]},
     {1, 5, x[1, 5, t]}, ..., {60, 56, x[60, 56, t]}, {60, 57, x[60, 57, t]},
     {60, 58, x[60, 58, t]}, {60, 59, x[60, 59, t]}, {60, 60, x[60, 60, t]}}

    large output    show less    show more    show all    set size limit...
```

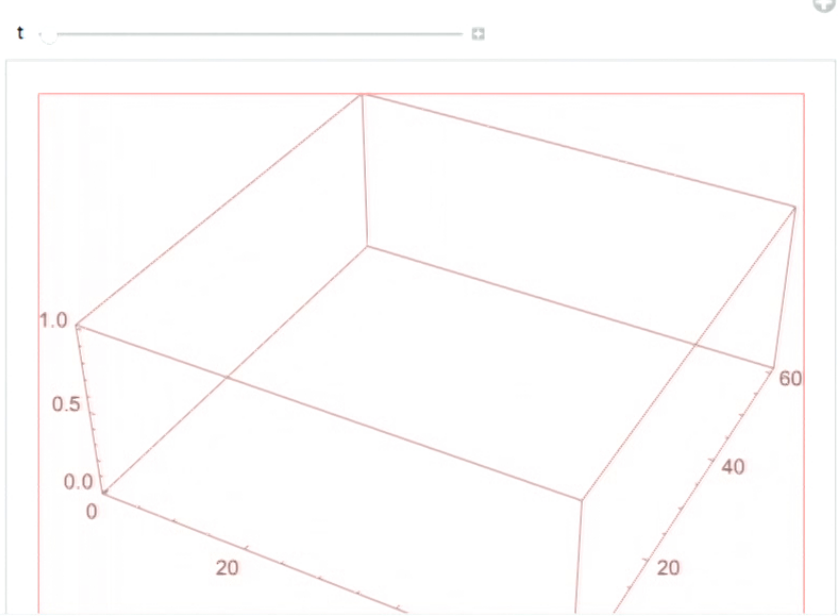
```
In[22]:= var = Table[x[i, j][t], {i, L}, {j, L}] // Flatten;

In[23]:= solution = NDSolve[Join[eqs, bc], var, {t, 0, tmax}][[1]];

In[26]:= ToPlot[t_] = var /. x[i_, j_][t] :> {i, j, x[i, j, t]} /. solution;

In[27]:= Manipulate[ListPointPlot3D[ToPlot[t]], {t, 0, 500, 1}]
```

Out[27]=

t



```
{i, L}, {j, L}] // Flatten;

In[21]:= bc = Table[{x[i, j][0] == 0, x[i, j]'[0] == If[i > 3 L / 4 ∧ j > 3 L / 4, RandomReal[{4, 6}], 0]}, {i, L}, {j, L}] //
    Flatten;

In[22]:= var = Table[x[i, j][t], {i, L}, {j, L}] // Flatten;

In[23]:= solution = NDSolve[Join[eqs, bc], var, {t, 0, tmax}][[1]];

ToPlot[t_] = var /. x[i_, j_][t] -> {i, j, x[i, j, t]} /. solution;

In[27]:= Manipulate[ListPointPlot3D[ToPlot[t]], {t, 0, 500, 1}]
Out[27]=
```



In[22]:=
```
var = Table[x[i, j][t], {i, L}, {j, L}] // Flatten;
```

In[23]:=
```
solution = NDSolve[Join[eqs, bc], var, {t, 0, tmax}][[1]];
```

In[29]:=
```
ToPlot[t_] = var /. x[i_, j_][t] -> {i, j, x[i, j][t]} /. solution;
```

In[27]:=
```
Manipulate[ListPointPlot3D[ToPlot[t]], {t, 0, 500, 1}]
```

200%

In[22]:=
```
var = Table[x[i, j][t], {i, L}, {j, L}] // Flatten;
```

In[23]:=
```
solution = NDSolve[Join[eqs, bc], var, {t, 0, tmax}][[1]];
```

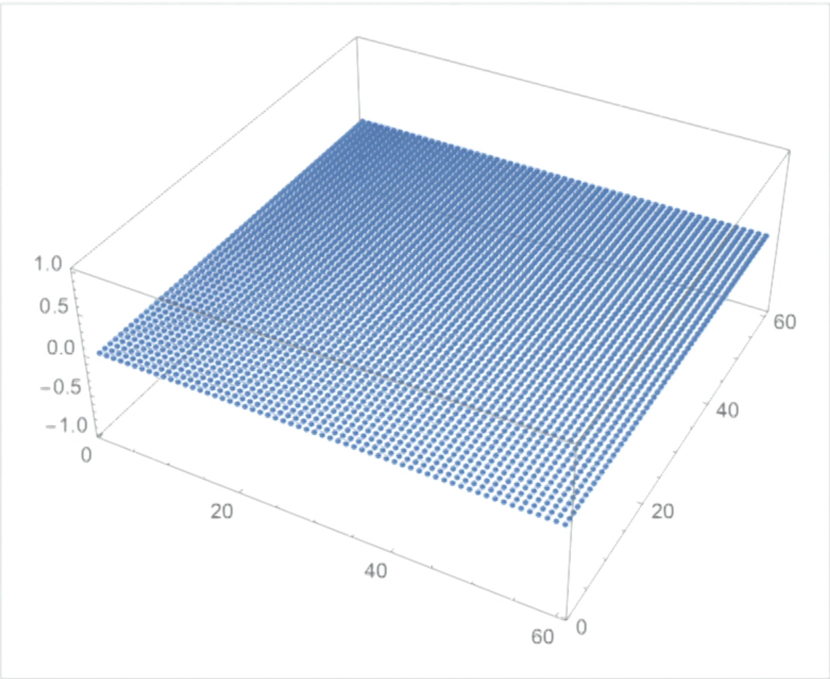
In[29]:=
```
ToPlot[t_] = var /. x[i_, j_][t] -> {i, j, x[i, j][t]} /. solution;
```

In[30]:=
```
Manipulate[ListPointPlot3D[ToPlot[t]], {t, 0, 500, 1}]
```

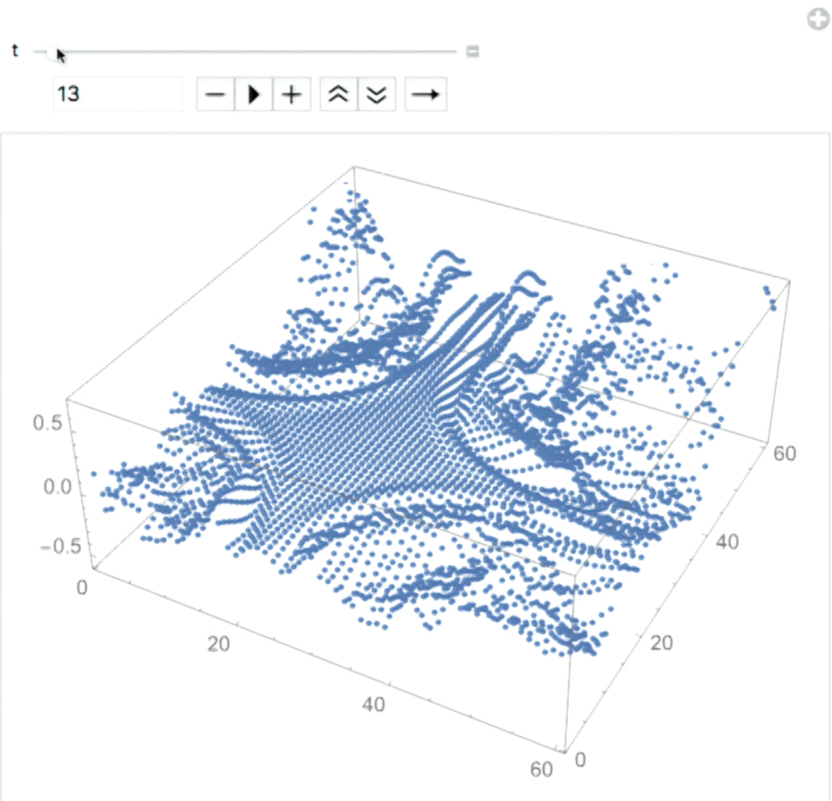
Out[30]=

t

0



In[22]:=
```
var = Table[x[i, j][t], {i, L}, {j, L}] // Flatten;
```
In[23]:=
```
solution = NDSolve[Join[eqs, bc], var, {t, 0, tmax}][[1]];
```
In[29]:=
```
ToPlot[t_] = var /. x[i_, j_][t] -> {i, j, x[i, j][t]} /. solution;
```
In[30]:=
```
Manipulate[ListPointPlot3D[ToPlot[t]], {t, 0, 500, 1}]
```
Out[30]=



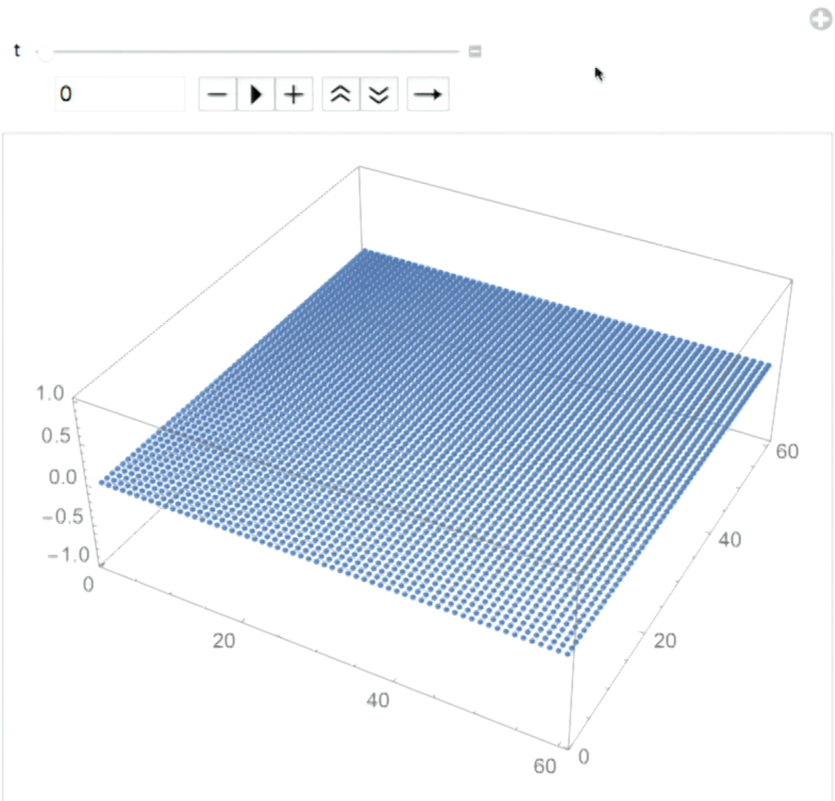
In[22]:=
```
var = Table[x[i, j][t], {i, L}, {j, L}] // Flatten;
```

In[23]:=
```
solution = NDSolve[Join[eqs, bc], var, {t, 0, tmax}][[1]];
```

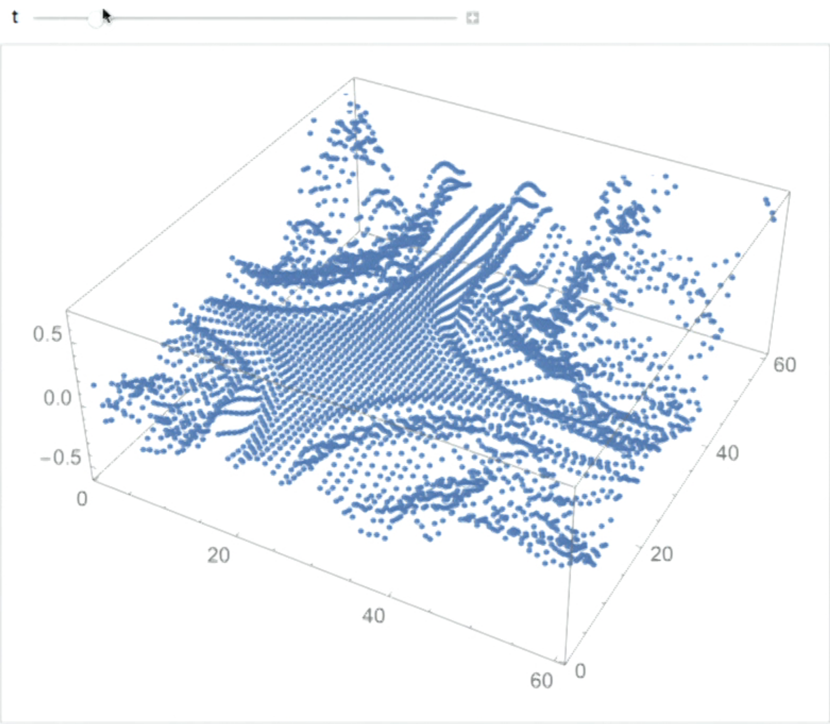
In[29]:=
```
ToPlot[t_] = var /. x[i_, j_][t] -> {i, j, x[i, j][t]} /. solution;
```

In[30]:=
```
Manipulate[ListPointPlot3D[ToPlot[t]], {t, 0, 500, 1}]
```

Out[30]=

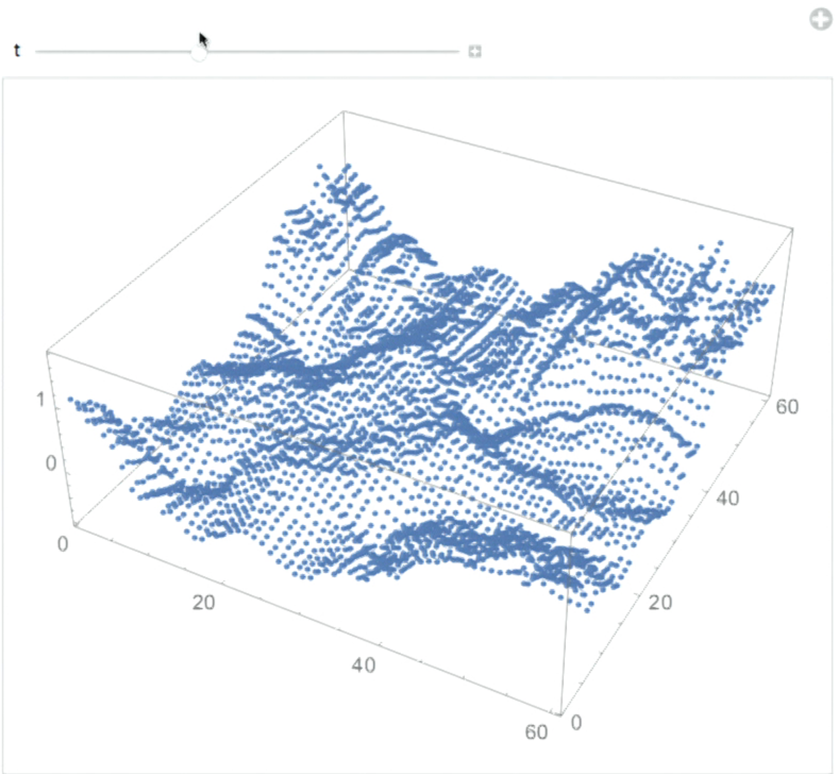


```
In[22]:= var = Table[x[i, j][t], {i, L}, {j, L}] // Flatten;

In[23]:= solution = NDSolve[Join[eqs, bc], var, {t, 0, tmax}][[1]];

In[29]:= ToPlot[t_] = var /. x[i_, j_][t] -> {i, j, x[i, j][t]} /. solution;

In[31]:= Manipulate[ListPointPlot3D[ToPlot[t]], {t, 0, 100, 1}]

Out[31]=
```

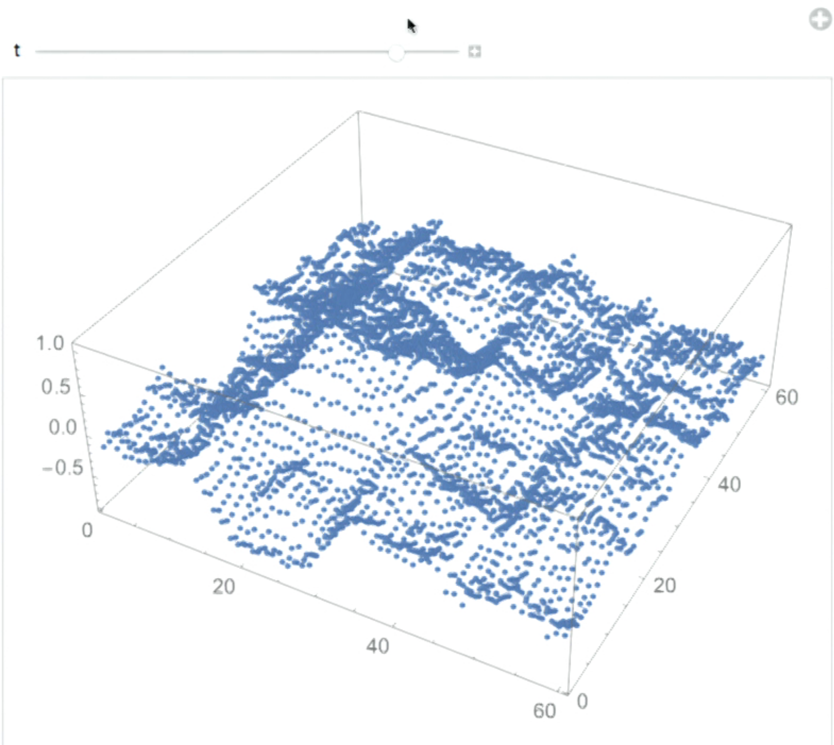


```
In[22]:= var = Table[x[i, j][t], {i, L}, {j, L}] // Flatten;

In[23]:= solution = NDSolve[Join[eqs, bc], var, {t, 0, tmax}][[1]];

In[29]:= ToPlot[t_] = var /. x[i_, j_][t] -> {i, j, x[i, j][t]} /. solution;

In[31]:= Manipulate[ListPointPlot3D[ToPlot[t]], {t, 0, 100, 1}]
```

Out[31]=



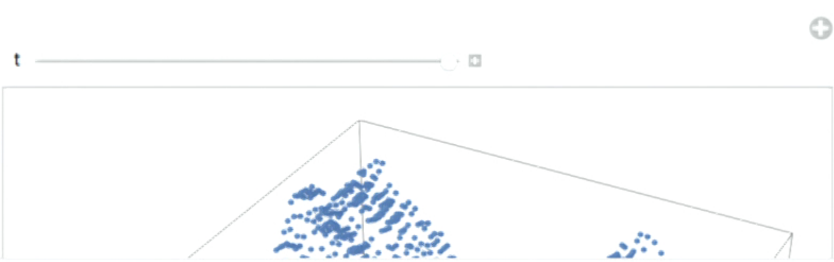
In[22]:=
```
var = Table[x[i, j][t], {i, L}, {j, L}] // Flatten;
```
In[23]:=
```
solution = NDSolve[Join[eqs, bc], var, {t, 0, tmax}][[1]];
```
In[29]:=
```
ToPlot[t_] = var /. x[i_, j_][t] -> {i, j, x[i, j][t]} /. solution;
```
In[31]:=
```
Manipulate[ListPointPlot3D[ToPlot[t]], {t, 0, 100, 1}]
```
Out[31]=



Punching the Table. Time Evolution.

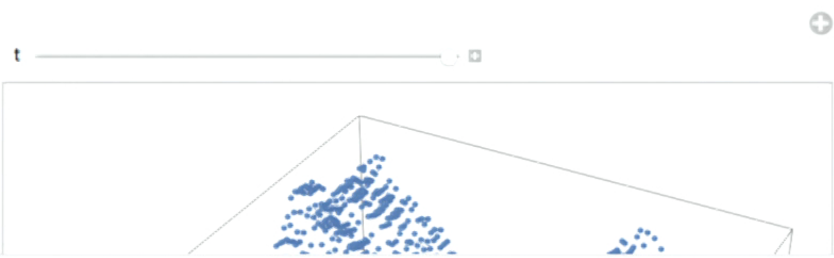
```
ClearAll[x]
L = 60; κ = 5; ω = 2; tmax = 500; μ = 5;
x[i_, j_][t_] /; i > L ∨ i ≤ 0 ∨ j > L ∨ j ≤ 0 := x[Mod[i, L, 1], Mod[j, L, 1]][t]
```

```
In[20]:= eqs =
  Table[∂{t,2} x[i, j][t] + ω^2 x[i, j][t] + κ Sum[x[i, j][t] - x[i + l, j][t] + x[i, j][t] - x[i, j + l][t], {l, -1, 1}] == 0,
    {i, L}, {j, L}] // Flatten;

In[21]:= bc = Table[{x[i, j][0] == 0, x[i, j]'[0] == If[i > 3 L / 4 ∧ j > 3 L / 4, RandomReal[{4, 6}], 0]}, {i, L}, {j, L}] //
   Flatten;

In[22]:= var = Table[x[i, j][t], {i, L}, {j, L}] // Flatten;

In[23]:= solution = NDSolve[Join[eqs, bc], var, {t, 0, tmax}][[1]];

In[29]:= ToPlot[t_] = var /. x[i_, j_][t] -> {i, j, x[i, j][t]} /. solution;

In[31]:= Manipulate[ListPointPlot3D[ToPlot[t]], {t, 0, 100, 1}]
Out[31]=
```



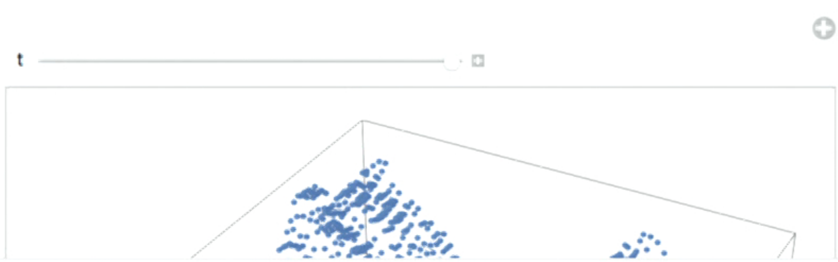
Punching the Table. Time Evolution.

```
ClearAll[x]
L = 60; ϰ = 5; ω = 2; tmax = 500; μ = 5;
x[i_, j_][t_] /; i > L ∨ i ≤ 0 ∨ j > L ∨ j ≤ 0 := x[Mod[i, L, 1], Mod[j, L, 1]][t]

eqs =
  Table[∂{t,2} x[i, j][t] + ω^2 x[i, j][t] + μ^2 x[i, j][t]^3 +
     ϰ Sum[x[i, j][t] - x[i + l, j][t] + x[i, j][t] - x[i, j + l][t], {l, -1, 1}] == 0, {i, L}, {j, L}] // Flatten;

In[21]:= bc = Table[{x[i, j][0] == 0, x[i, j]'[0] == If[i > 3 L / 4 ∧ j > 3 L / 4, RandomReal[{4, 6}], 0]}, {i, L}, {j, L}] //
   Flatten;

In[22]:= var = Table[x[i, j][t], {i, L}, {j, L}] // Flatten;

In[23]:= solution = NDSolve[Join[eqs, bc], var, {t, 0, tmax}][[1]];

In[29]:= ToPlot[t_] = var /. x[i_, j_][t] -> {i, j, x[i, j][t]} /. solution;

In[31]:= Manipulate[ListPointPlot3D[ToPlot[t]], {t, 0, 100, 1}]
Out[31]=
```



Punching the Table. Time Evolution.

```
ClearAll[x]
L = 60; ϰ = 5; ω = 2; tmax = 500; μ = 5;
x[i_, j_][t_] /; i > L ∨ i ≤ 0 ∨ j > L ∨ j ≤ 0 := x[Mod[i, L, 1], Mod[j, L, 1]][t]

eqs =
  Table[∂{t,2} x[i, j][t] + ω^2 x[i, j][t] + μ^2 x[i, j][t]^3 +
     ϰ Sum[x[i, j][t] - x[i + l, j][t] + x[i, j][t] - x[i, j + l][t], {l, -1, 1}] == 0, {i, L}, {j, L}] // Flatten;

In[21]:= bc = Table[{x[i, j][0] == 0, x[i, j]'[0] == If[i > 3 L / 4 ∧ j > 3 L / 4, RandomReal[{4, 6}], 0]}, {i, L}, {j, L}] //
   Flatten;

In[22]:= var = Table[x[i, j][t], {i, L}, {j, L}] // Flatten;

In[23]:= solution = NDSolve[Join[eqs, bc], var, {t, 0, tmax}][[1]];

In[29]:= ToPlot[t_] = var /. x[i_, j_][t] -> {i, j, x[i, j][t]} /. solution;

In[31]:= Manipulate[ListPointPlot3D[ToPlot[t]], {t, 0, 100, 1}]
Out[31]=
```



Punching the Table. Time Evolution.

```
ClearAll[x]
L = 60; κ = 5; ω = 2; tmax = 500; μ = 5;
x[i_, j_][t_] /; i > L ∨ i ≤ 0 ∨ j > L ∨ j ≤ 0 := x[Mod[i, L, 1], Mod[j, L, 1]][t]
```

```
In[32]:= eqs =
  Table[∂{t,2} x[i, j][t] + ω^2 x[i, j][t] + μ^2 x[i, j][t]^3 +
     κ Sum[x[i, j][t] - x[i + l, j][t] + x[i, j][t] - x[i, j + l][t], {l, -1, 1}] == 0, {i, L}, {j, L}] // Flatten;
```

```
In[33]:= bc = Table[{x[i, j][0] == 0, x[i, j]'[0] == If[i > 3 L / 4 ∧ j > 3 L / 4, RandomReal[{4, 6}], 0]}, {i, L}, {j, L}] //
   Flatten;
```

```
In[34]:= var = Table[x[i, j][t], {i, L}, {j, L}] // Flatten;
```

```
In[35]:= solution = NDSolve[Join[eqs, bc], var, {t, 0, tmax}][[1]];
```

```
··· NDSolve: The function value
  {0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., ≪3≫, 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., ≪
     7150≫} is not a list of numbers with dimensions {7200} at
  {t, x[1, 1][t], x[1, 1]′[t], x[1, 2][t], x[1, 2]′[t], x[1, 3][t], x[1, 3]′[t], x[1, 4][t], x[1, 4]′[t], ≪34≫, x[1, 22][t], x[1, 22]′[t], x[1, 23][t], x[1, 23]′[t], x[1, 24][t], x[1, 24]′[t], x[1, 25][t],
    ≪7151≫} =
  {0.000114591, 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., ≪9≫, 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., ≪7151≫}.
```

```
In[29]:= ToPlot[t_] = var /. x[i_, j_][t] -> {i, j, x[i, j][t]} /. solution;
```

```
In[31]:= Manipulate[ListPointPlot3D[ToPlot[t]], {t, 0, 100, 1}]
Out[31]=
```

Punching the Table. Time Evolution.

```
In[36]:= ClearAll[x]
L = 60; κ = 5; ω = 2; tmax = 500; μ = 5;
x[i_, j_][t_] /; i > L ∨ i ≤ 0 ∨ j > L ∨ j ≤ 0 := x[Mod[i, L, 1], Mod[j, L, 1]][t]

In[39]:= eqs =
  Table[∂{t,2} x[i, j][t] + ω^2 x[i, j][t] + μ^2 x[i, j][t]^3 +
     κ Sum[x[i, j][t] - x[i + l, j][t] + x[i, j][t] - x[i, j + l][t], {l, -1, 1}] == 0, {i, L}, {j, L}] // Flatten;

In[40]:= bc = Table[{x[i, j][0] == 0, x[i, j]'[0] == If[i > 3 L / 4 ∧ j > 3 L / 4, RandomReal[{4, 6}], 0]}, {i, L}, {j, L}] //
   Flatten;

In[41]:= var = Table[x[i, j][t], {i, L}, {j, L}] // Flatten;

In[42]:= solution = NDSolve[Join[eqs, bc], var, {t, 0, tmax}][[1]];

In[29]:= ToPlot[t_] = var /. x[i_, j_][t] -> {i, j, x[i, j][t]} /. solution;

legs = Graphics3D[Cylinder[{{1, 1, 0}, {1, 1, -8}}], Cylinder[{{1, L, 0}, {1, L, -8}}],
   Cylinder[{{L, 1, 0}, {L, 1, -8}}], Cylinder[{{L, L, 0}, {L, L, -8}}]];

Manipulate[Show[legs, ListPointPlot3D[ToPlot[t]]], {t, 0, 100, 1}]
Out[31]=
```

t

Graphics3D

Search for all pages containing Graphics3D.

Wolfram Language > Symbolic Graphics Language > Graphics3D
Wolfram Language > Maps & Cartography > Symbolic Graphics Language > Graphics3D
Wolfram Language > Charting and Information Visualization > Maps & Cartography > Symbolic Graphics Language > Graphics3D

BUILT-IN WOLFRAM LANGUAGE SYMBOL

Tutorials ▾ See Also ▾ Related Guides ▾ URL ▾

Graphics3D

Graphics3D[*primitives*, *options*]
represents a three-dimensional graphical image.

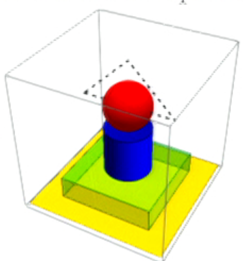
▸ **Details and Options**

▾ **Examples** (98)

▾ **Basic Examples** (3)

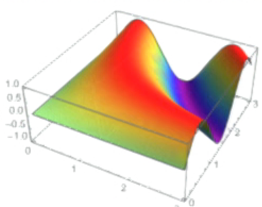
Use lines, polygons, cylinders, spheres, etc. to build up a 3D graphics scene:

In[1]:=
```
Graphics3D[{Blue, Cylinder[], Red, Sphere[{0, 0, 2}], Black, Thick, Dashed, Line[{{-2, 0, 2}, {2, 0, 2}, {0, 0, 4}, {-2, 0, 2}}], Yellow, Polygon[{{-3, -3, -2}, {-3, 3, -2}, {3, 3, -2}, {3, -3, -2}}], Green, Opacity[.3], Cuboid[{-2, -2, -2}, {2, 2, -1}]}]
```

Out[1]=


Use plot functions to automatically create Graphics3D from different types of data:

In[1]:=
```
Plot3D[Sin[x y], {x, 0, 3}, {y, 0, 3}, ColorFunction → "Rainbow", Mesh → None]
```

Out[1]=


In[2]:=
```
ParametricPlot3D[{(3 + Cos[v]) Cos[u], (3 + Cos[v]) Sin[u], Sin[v]}, {u, 0, 2 Pi}, {v, 0, 2 Pi}, Mesh → None]
```

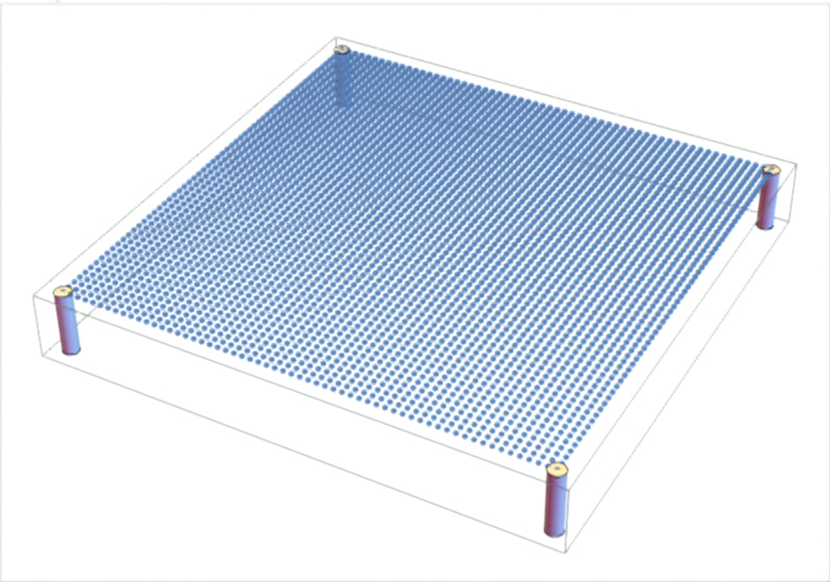
100%

```
legs = Graphics3D[{Cylinder[{{1, 1, 0}, {1, 1, -8}}], Cylinder[{{1, L, 0}, {1, L, -8}}],
    Cylinder[{{L, 1, 0}, {L, 1, -8}}], Cylinder[{{L, L, 0}, {L, L, -8}}]}];
```

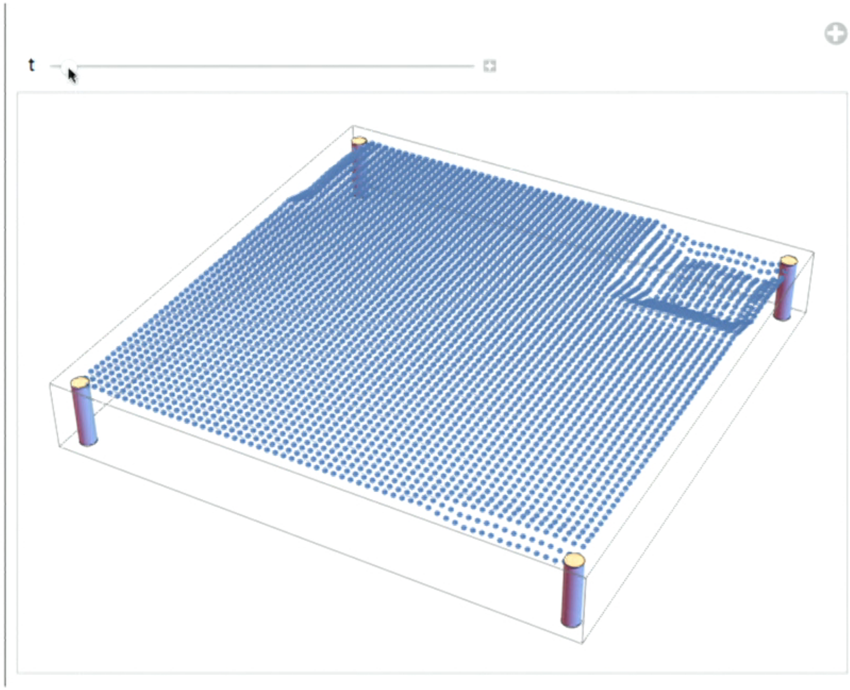
In[44]:=
```
Manipulate[Show[legs, ListPointPlot3D[ToPlot[t]]], {t, 0, 100, 1}]
```

Out[44]=

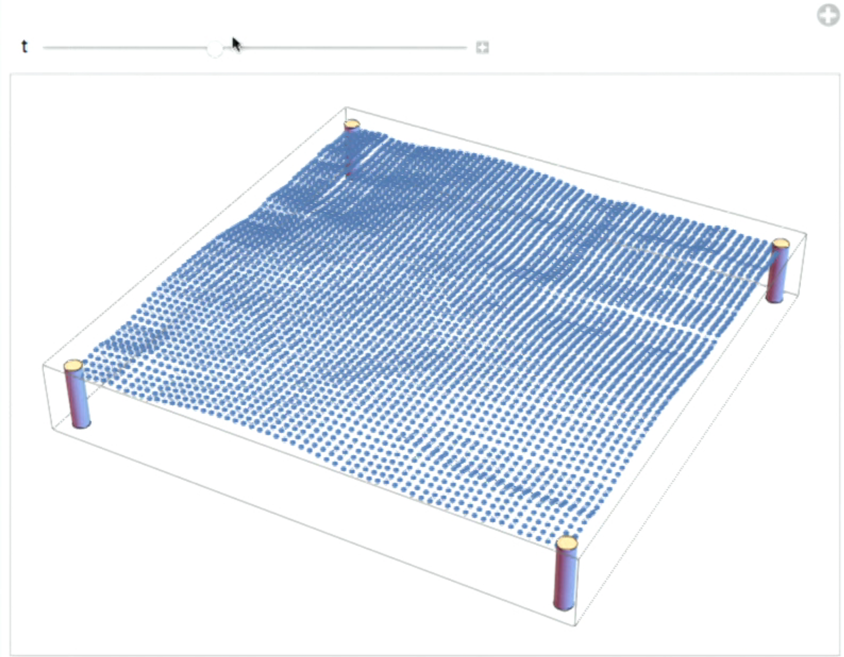
t



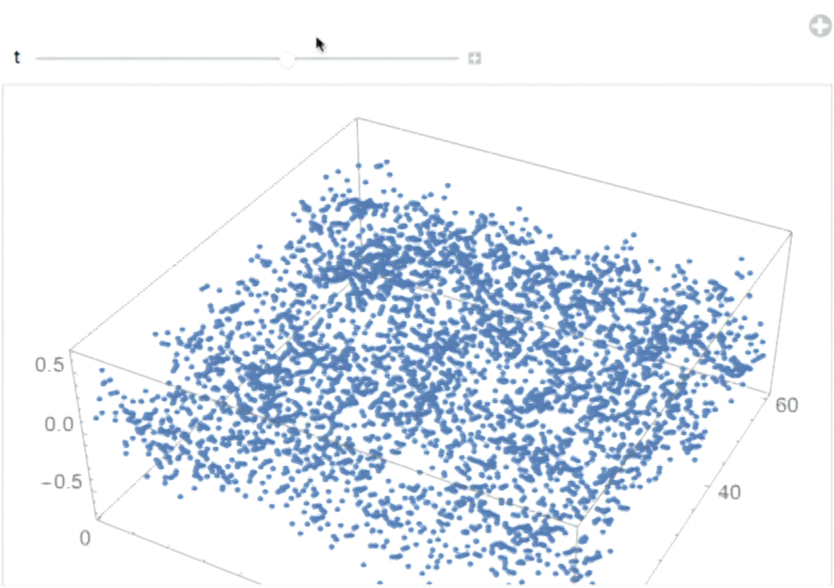
```
legs = Graphics3D[{Cylinder[{{1, 1, 0}, {1, 1, -8}}], Cylinder[{{1, L, 0}, {1, L, -8}}],
    Cylinder[{{L, 1, 0}, {L, 1, -8}}], Cylinder[{{L, L, 0}, {L, L, -8}}]}];

In[44]:= Manipulate[Show[legs, ListPointPlot3D[ToPlot[t]]], {t, 0, 100, 1}]
Out[44]=
```



```
legs = Graphics3D[{Cylinder[{{1, 1, 0}, {1, 1, -8}}], Cylinder[{{1, L, 0}, {1, L, -8}}],
    Cylinder[{{L, 1, 0}, {L, 1, -8}}], Cylinder[{{L, L, 0}, {L, L, -8}}]}];

In[44]:= Manipulate[Show[legs, ListPointPlot3D[ToPlot[t]]], {t, 0, 100, 1}]
Out[44]=
```



```
In[40]:= bc = Table[{x[i, j][0] == 0, x[i, j]'[0] == If[i > 3 L / 4 ∧ j > 3 L / 4, RandomReal[{4, 6}], 0]}, {i, L}, {j, L}] //
    Flatten;

In[41]:= var = Table[x[i, j][t], {i, L}, {j, L}] // Flatten;

In[42]:= solution = NDSolve[Join[eqs, bc], var, {t, 0, tmax}][[1]];

In[50]:= ToPlot[t_] = var /. x[i_, j_][t] -> {i, j, x[i, j][t]} /. solution;

In[47]:= legs = Graphics3D[{Cylinder[{{1, 1, 0}, {1, 1, -3}}], Cylinder[{{1, L, 0}, {1, L, -3}}],
     Cylinder[{{L, 1, 0}, {L, 1, -3}}], Cylinder[{{L, L, 0}, {L, L, -3}}]}];

In[51]:= Manipulate[ListPointPlot3D[ToPlot[t]], {t, 0, 100, 1}]
Out[51]=
```

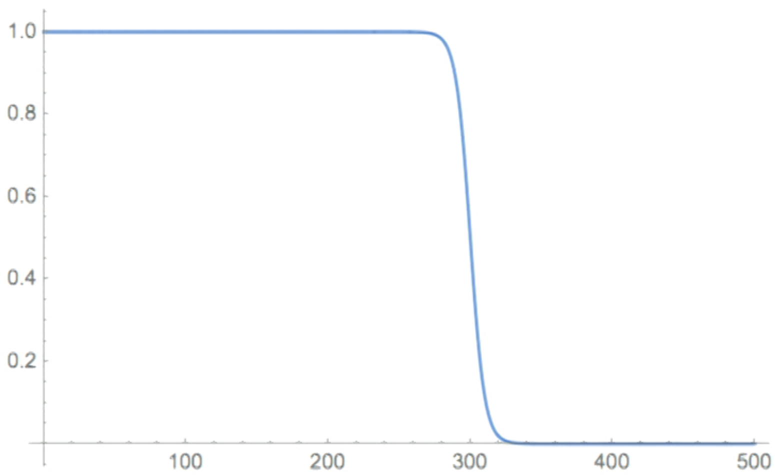


Punching the Table. Time Evolution.

In[36]:=
```
ClearAll[x]
L = 60; ϰ = 5; ω = 2; tmax = 500; μ = 5;
x[i_, j_][t_] /; i > L ⋁ i ≤ 0 ⋁ j > L ⋁ j ≤ 0 := x[Mod[i, L, 1], Mod[j, L, 1]][t]
```

In[52]:=
```
dial[t_] = (1 - Tanh[(t-300)/10])/2;
Plot[dial[t], {t, 0, 500}]
```

Out[53]=

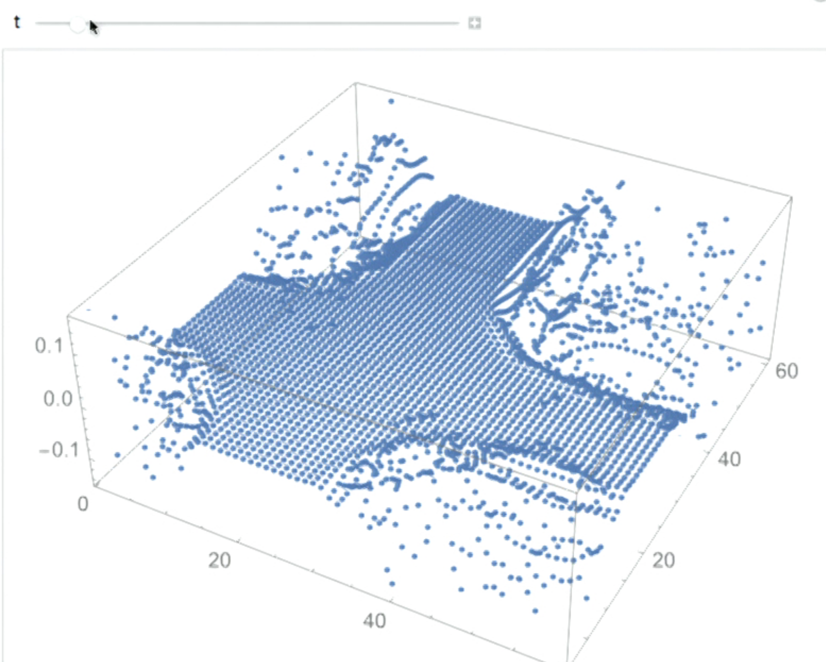


In[39]:=
```
eqs =
  Table[∂{t,2} x[i, j][t] + ω^2 x[i, j][t] + μ^2 x[i, j][t]^3 +
    ϰ Sum[x[i, j][t] - x[i + l, j][t] + x[i, j][t] - x[i, j + l][t], {l, -1, 1}] == 0, {i, L}, {j, L}] // Flatten;
```

Punching the Table. Time Evolution.

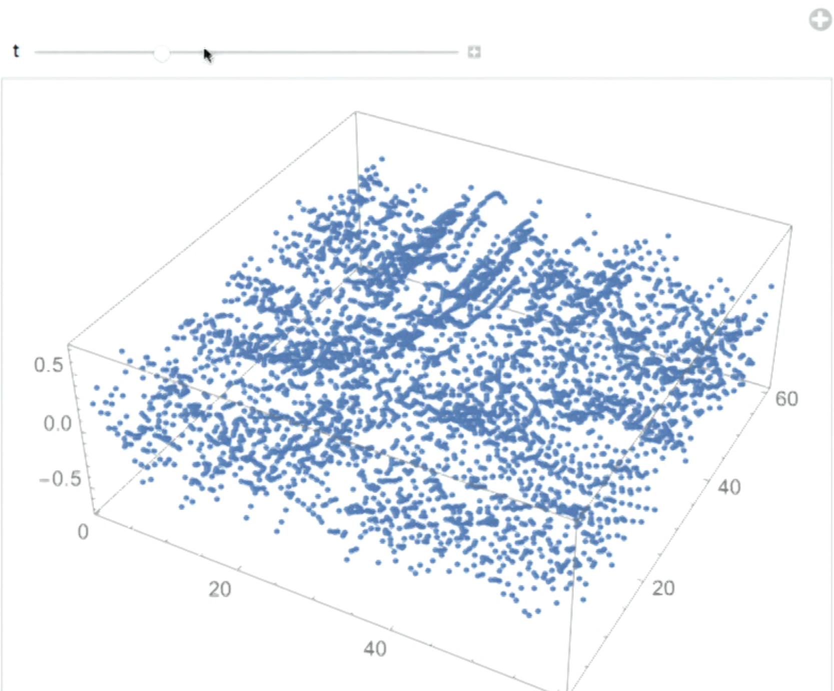
```
In[36]:= ClearAll[x]
L = 60; κ = 5; ω = 2; tmax = 500; μ = 5;
x[i_, j_][t_] /; i > L ∨ i ≤ 0 ∨ j > L ∨ j ≤ 0 := x[Mod[i, L, 1], Mod[j, L, 1]][t]

dial[t_] = (1 - Tanh[(t-300)/10])/2;
Plot[dial[t], {t, 0, 500}];

eqs =
  Table[∂{t,2} x[i, j][t] + ω^2 x[i, j][t] + dial[t] μ^2 x[i, j][t]^3 +
     κ Sum[x[i, j][t] - x[i + l, j][t] + x[i, j][t] - x[i, j + l][t], {l, -1, 1}] == 0, {i, L}, {j, L}] // Flatten;

In[40]:= bc = Table[{x[i, j][0] == 0, x[i, j]'[0] == If[i > 3 L / 4 ∧ j > 3 L / 4, RandomReal[{4, 6}], 0]}, {i, L}, {j, L}] //
   Flatten;

In[41]:= var = Table[x[i, j][t], {i, L}, {j, L}] // Flatten;

In[42]:= solution = NDSolve[Join[eqs, bc], var, {t, 0, tmax}][[1]];

In[50]:= ToPlot[t_] = var /. x[i_, j_][t] -> {i, j, x[i, j][t]} /. solution;

In[47]:= legs = Graphics3D[{Cylinder[{{1, 1, 0}, {1, 1, -3}}], Cylinder[{{1, L, 0}, {1, L, -3}}],
    Cylinder[{{L, 1, 0}, {L, 1, -3}}], Cylinder[{{L, L, 0}, {L, L, -3}}]}];

Manipulate[ListPointPlot3D[ToPlot[t]], {t, 0, 100, 1}];
```

```
In[58]:= var = Table[x[i, j][t], {i, L}, {j, L}] // Flatten;

In[59]:= solution = NDSolve[Join[eqs, bc], var, {t, 0, tmax}][[1]];

In[60]:= ToPlot[t_] = var /. x[i_, j_][t] -> {i, j, x[i, j][t]} /. solution;

In[61]:= legs = Graphics3D[{Cylinder[{{1, 1, 0}, {1, 1, -3}}], Cylinder[{{1, L, 0}, {1, L, -3}}],
    Cylinder[{{L, 1, 0}, {L, 1, -3}}], Cylinder[{{L, L, 0}, {L, L, -3}}]}];

In[64]:= Manipulate[ListPointPlot3D[ToPlot[t]], {t, 0, 100, 1}]
Out[64]=
```

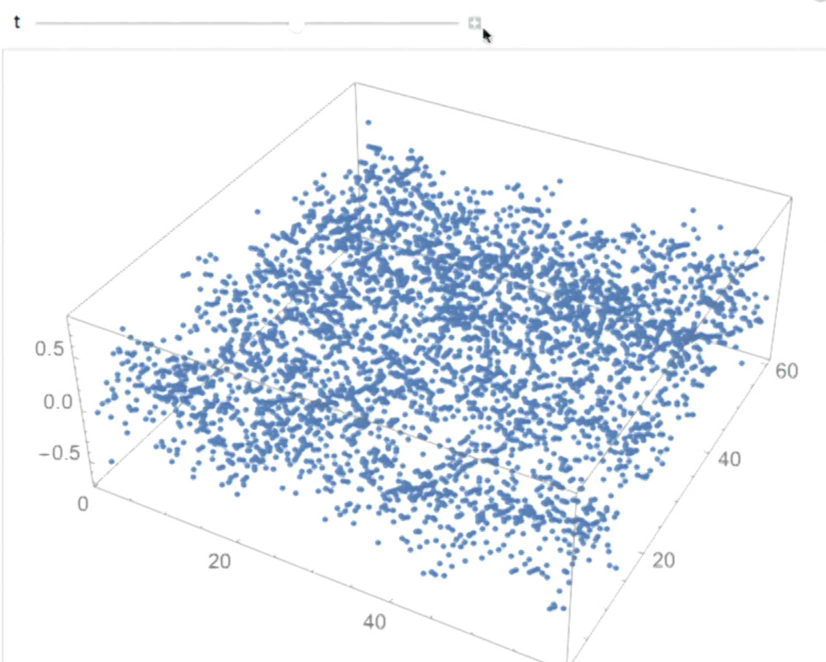


```
In[58]:= var = Table[x[i, j][t], {i, L}, {j, L}] // Flatten;

In[59]:= solution = NDSolve[Join[eqs, bc], var, {t, 0, tmax}][[1]];

In[60]:= ToPlot[t_] = var /. x[i_, j_][t] -> {i, j, x[i, j][t]} /. solution;

In[61]:= legs = Graphics3D[{Cylinder[{{1, 1, 0}, {1, 1, -3}}], Cylinder[{{1, L, 0}, {1, L, -3}}],
    Cylinder[{{L, 1, 0}, {L, 1, -3}}], Cylinder[{{L, L, 0}, {L, L, -3}}]}];

In[64]:= Manipulate[ListPointPlot3D[ToPlot[t]], {t, 0, 100, 1}]
```

Out[64]=



```
In[58]:= var = Table[x[i, j][t], {i, L}, {j, L}] // Flatten;

In[59]:= solution = NDSolve[Join[eqs, bc], var, {t, 0, tmax}][[1]];

In[60]:= ToPlot[t_] = var /. x[i_, j_][t] -> {i, j, x[i, j][t]} /. solution;

In[61]:= legs = Graphics3D[{Cylinder[{{1, 1, 0}, {1, 1, -3}}], Cylinder[{{1, L, 0}, {1, L, -3}}],
    Cylinder[{{L, 1, 0}, {L, 1, -3}}], Cylinder[{{L, L, 0}, {L, L, -3}}]}];

In[65]:= Manipulate[ListPointPlot3D[ToPlot[t]], {t, 0, 500, 1}]
Out[65]=
```



```
In[54]:= dial[t_] = ─────────── ;
                        2
         Plot[dial[t], {t, 0, 500}];

In[56]:= eqs =
           Table[∂{t,2} x[i, j][t] + ω^2 x[i, j][t] + dial[t] μ^2 x[i, j][t]^3 +
               κ Sum[x[i, j][t] - x[i + l, j][t] + x[i, j][t] - x[i, j + l][t], {l, -1, 1}] == 0, {i, L}, {j, L}] // Flatten;

In[57]:= bc = Table[{x[i, j][0] == 0, x[i, j]'[0] == If[i > 3 L / 4 ∧ j > 3 L / 4, RandomReal[{4, 6}], 0]}, {i, L}, {j, L}] //
            Flatten;

In[58]:= var = Table[x[i, j][t], {i, L}, {j, L}] // Flatten;

In[59]:= solution = NDSolve[Join[eqs, bc], var, {t, 0, tmax}][[1]];

In[60]:= ToPlot[t_] = var /. x[i_, j_][t] -> {i, j, x[i, j][t]} /. solution;

In[61]:= legs = Graphics3D[{Cylinder[{{1, 1, 0}, {1, 1, -3}}], Cylinder[{{1, L, 0}, {1, L, -3}}],
             Cylinder[{{L, 1, 0}, {L, 1, -3}}], Cylinder[{{L, L, 0}, {L, L, -3}}]}];

In[65]:= Manipulate[ListPointPlot3D[ToPlot[t]], {t, 0, 500, 1}]
```

Saving the evolution for easy manipulation

```
In[66]:= solution[[1]]
Out[66]=
         x[1, 1][t] → InterpolatingFunction[ Domain: {{0., 500.}}
                                             Output: scalar ][t]
```

200%

```
  Flatten;

In[58]:= var = Table[x[i, j][t], {i, L}, {j, L}] // Flatten;

In[59]:= solution = NDSolve[Join[eqs, bc], var, {t, 0, tmax}][[1]];

In[60]:= ToPlot[t_] = var /. x[i_, j_][t] -> {i, j, x[i, j][t]} /. solution;

In[61]:= legs = Graphics3D[{Cylinder[{{1, 1, 0}, {1, 1, -3}}], Cylinder[{{1, L, 0}, {1, L, -3}}],
     Cylinder[{{L, 1, 0}, {L, 1, -3}}], Cylinder[{{L, L, 0}, {L, L, -3}}]}];

In[65]:= Manipulate[ListPointPlot3D[ToPlot[t]], {t, 0, 500, 1}]
```

Saving the evolution for easy manipulation

```
In[66]:= solution[[1]]

Out[66]=
x[1, 1][t] → InterpolatingFunction[ Domain: {{0., 500.}}
                                    Output: scalar ][t]

In[67]:= POSITIONS[t_] = Table[{a, b} → x[a, b][t], {a, L}, {b, L}]

Out[67]=
{{{1, 1} → x[1, 1][t], {1, 2} → x[1, 2][t], {1, 3} → x[1, 3][t],
  {1, 4} → x[1, 4][t], ... 52 ..., {1, 57} → x[1, 57][t], {1, 58} → x[1, 58][t],
  {1, 59} → x[1, 59][t], {1, 60} → x[1, 60][t]}, ... 58 ..., {... 1 ...}}

large output   show less   show more   show all   set size limit...
```

```
In[60]:= ToPlot[t_] = var /. x[i_, j_][t] -> {i, j, x[i, j][t]} /. solution;

In[61]:= legs = Graphics3D[{Cylinder[{{1, 1, 0}, {1, 1, -3}}], Cylinder[{{1, L, 0}, {1, L, -3}}],
     Cylinder[{{L, 1, 0}, {L, 1, -3}}], Cylinder[{{L, L, 0}, {L, L, -3}}]}];

In[65]:= Manipulate[ListPointPlot3D[ToPlot[t]], {t, 0, 500, 1}]
```

Saving the evolution for easy manipulation

```
In[66]:= solution[[1]]

Out[66]=
x[1, 1][t] → InterpolatingFunction[ Domain: {{0., 500.}} Output: scalar ][t]

POSITIONS[t_] = Table[{a, b} → x[a, b][t], {a, L}, {b, L}] /. solutions;
```

200%

```
In[60]:= ToPlot[t_] = var /. x[i_, j_][t] -> {i, j, x[i, j][t]} /. solution;

In[61]:= legs = Graphics3D[{Cylinder[{{1, 1, 0}, {1, 1, -3}}], Cylinder[{{1, L, 0}, {1, L, -3}}],
         Cylinder[{{L, 1, 0}, {L, 1, -3}}], Cylinder[{{L, L, 0}, {L, L, -3}}]}];

In[65]:= Manipulate[ListPointPlot3D[ToPlot[t]], {t, 0, 500, 1}]
```

Saving the evolution for easy manipulation

```
In[66]:= solution[[1]]

Out[66]=
x[1, 1][t] → InterpolatingFunction[ Domain: {{0., 500.}} Output: scalar ][t]

POSITIONS[t_] = Table[{a, b} → x[a, b][t], {a, L}, {b, L}] /. solution;
```

200%

In[60]:=
```
ToPlot[t_] = var /. x[i_, j_][t] -> {i, j, x[i, j][t]} /. solution;
```

In[61]:=
```
legs = Graphics3D[{Cylinder[{{1, 1, 0}, {1, 1, -3}}], Cylinder[{{1, L, 0}, {1, L, -3}}],
    Cylinder[{{L, 1, 0}, {L, 1, -3}}], Cylinder[{{L, L, 0}, {L, L, -3}}]}];
```

In[65]:=
```
Manipulate[ListPointPlot3D[ToPlot[t]], {t, 0, 500, 1}]
```

Saving the evolution for easy manipulation

In[66]:=
```
solution[[1]]
```

Out[66]=

```
x[1, 1][t] → InterpolatingFunction[ Domain: {{0., 500.}} Output: scalar ][t]
```

```
POSITIONS[t_] = Table[{a, b} → x[a, b][t], {a, L}, {b, L}] /. solution;

ALLPOSITIONS = Table[POSITIONS[t], {t, 0, 500}]
```

200%

```
In[60]:= ToPlot[t_] = var /. x[i_, j_][t] -> {i, j, x[i, j][t]} /. solution;

In[61]:= legs = Graphics3D[{Cylinder[{{1, 1, 0}, {1, 1, -3}}], Cylinder[{{1, L, 0}, {1, L, -3}}],
       Cylinder[{{L, 1, 0}, {L, 1, -3}}], Cylinder[{{L, L, 0}, {L, L, -3}}]}];

In[65]:= Manipulate[ListPointPlot3D[ToPlot[t]], {t, 0, 500, 1}]
```

Saving the evolution for easy manipulation

```
In[66]:= solution[[1]]

Out[66]=
x[1, 1][t] → InterpolatingFunction[ Domain: {{0., 500.}} Output: scalar ][t]

POSITIONS[t_] = Table[{a, b} → x[a, b][t], {a, L}, {b, L}] /. solution;

ALLPOSITIONS = Table[POSITIONS[t], {t, 0, 500, 1}]
```

200%

```
In[60]:= ToPlot[t_] = var /. x[i_, j_][t] -> {i, j, x[i, j][t]} /. solution;

In[61]:= legs = Graphics3D[{Cylinder[{{1, 1, 0}, {1, 1, -3}}], Cylinder[{{1, L, 0}, {1, L, -3}}],
      Cylinder[{{L, 1, 0}, {L, 1, -3}}], Cylinder[{{L, L, 0}, {L, L, -3}}]}];

In[65]:= Manipulate[ListPointPlot3D[ToPlot[t]], {t, 0, 500, 1}]
```

Saving the evolution for easy manipulation

```
In[66]:= solution[[1]]

Out[66]=
x[1, 1][t] → InterpolatingFunction[ Domain: {{0., 500.}} Output: scalar ][t]

POSITIONS[t_] = Table[{a, b} → x[a, b][t], {a, L}, {b, L}] /. solution;

ALLPOSITIONS = Table[POSITIONS[t], {t, 0, tmax, 1}]
```

```
In[60]:= ToPlot[t_] = var /. x[i_, j_][t] -> {i, j, x[i, j][t]} /. solution;

In[61]:= legs = Graphics3D[{Cylinder[{{1, 1, 0}, {1, 1, -3}}], Cylinder[{{1, L, 0}, {1, L, -3}}],
            Cylinder[{{L, 1, 0}, {L, 1, -3}}], Cylinder[{{L, L, 0}, {L, L, -3}}]}];

In[65]:= Manipulate[ListPointPlot3D[ToPlot[t]], {t, 0, 500, 1}]
```

Saving the evolution for easy manipulation

```
In[66]:= solution[[1]]

Out[66]=
         x[1, 1][t] → InterpolatingFunction[ Domain: {{0., 500.}}
                                             Output: scalar ][t]

In[69]:= POSITIONS[t_] = Flatten@Table[{a, b} → x[a, b][t], {a, L}, {b, L}] /. solution;

In[71]:= ALLPOSITIONS = Table[POSITIONS[t], {t, 0, tmax, 1}];

In[72]:= ALLPOSITIONS // Dimensions

Out[72]=
         {501, 3600}
```

```
In[60]:= ToPlot[t_] = var /. x[i_, j_][t] -> {i, j, x[i, j][t]} /. solution;

In[61]:= legs = Graphics3D[{Cylinder[{{1, 1, 0}, {1, 1, -3}}], Cylinder[{{1, L, 0}, {1, L, -3}}],
     Cylinder[{{L, 1, 0}, {L, 1, -3}}], Cylinder[{{L, L, 0}, {L, L, -3}}]}];

In[65]:= Manipulate[ListPointPlot3D[ToPlot[t]], {t, 0, 500, 1}]
```

Saving the evolution for easy manipulation

```
In[66]:= solution[[1]]

Out[66]=
    x[1, 1][t] → InterpolatingFunction[ Domain: {{0., 500.}}
                                        Output: scalar ][t]

In[69]:= POSITIONS[t_] = Flatten@Table[{a, b} → x[a, b][t], {a, L}, {b, L}] /. solution;

In[71]:= ALLPOSITIONS = Table[POSITIONS[t], {t, 0, tmax, 1}];

In[72]:= ALLPOSITIONS // Dimensions
```

200%

```
In[60]:= ToPlot[t_] = var /. x[i_, j_][t] -> {i, j, x[i, j][t]} /. solution;

In[61]:= legs = Graphics3D[{Cylinder[{{1, 1, 0}, {1, 1, -3}}], Cylinder[{{1, L, 0}, {1, L, -3}}],
        Cylinder[{{L, 1, 0}, {L, 1, -3}}], Cylinder[{{L, L, 0}, {L, L, -3}}]}];

In[65]:= Manipulate[ListPointPlot3D[ToPlot[t]], {t, 0, 500, 1}]
```

Saving the evolution for easy manipulation

```
In[66]:= solution[[1]]

Out[66]=
   x[1, 1][t] → InterpolatingFunction[ Domain: {{0., 500.}}
                                       Output: scalar ][t]

In[69]:= POSITIONS[t_] = Flatten@Table[{a, b} → x[a, b][t], {a, L}, {b, L}] /. solution;

In[71]:= ALLPOSITIONS = Table[POSITIONS[t], {t, 0, tmax, 1}];

In[73]:= ALLPOSITIONS >> (NotebookDirectory[] <> "Data/positions.txt");
```

200%

```
In[60]:= ToPlot[t_] = var /. x[i_, j_][t] -> {i, j, x[i, j][t]} /. solution;

In[61]:= legs = Graphics3D[{Cylinder[{{1, 1, 0}, {1, 1, -3}}], Cylinder[{{1, L, 0}, {1, L, -3}}],
     Cylinder[{{L, 1, 0}, {L, 1, -3}}], Cylinder[{{L, L, 0}, {L, L, -3}}]}];

In[65]:= Manipulate[ListPointPlot3D[ToPlot[t]], {t, 0, 500, 1}]
```

Saving the evolution for easy manipulation

```
In[66]:= solution[[1]]

Out[66]=
x[1, 1][t] → InterpolatingFunction[ Domain: {{0., 500.}} Output: scalar ][t]

In[69]:= POSITIONS[t_] = Flatten@Table[{a, b} → x[a, b][t], {a, L}, {b, L}] /. solution;

In[71]:= ALLPOSITIONS = Table[POSITIONS[t], {t, 0, tmax, 1}];

In[73]:= ALLPOSITIONS >> (NotebookDirectory[] <> "Data/positions.txt");
```

Plots

```
L = 60;
ALLPOSITIONS = << (NotebookDirectory[] <> "Data/positions.txt");
```

Saving the evolution for easy manipulation

In[66]:=
```
solution[[1]]
```

Out[66]=
```
x[1, 1][t] → InterpolatingFunction[ Domain: {{0., 500.}} Output: scalar ][t]
```

In[69]:=
```
POSITIONS[t_] = Flatten@Table[{a, b} → x[a, b][t], {a, L}, {b, L}] /. solution;
```

In[71]:=
```
ALLPOSITIONS = Table[POSITIONS[t], {t, 0, tmax, 1}];
```

In[73]:=
```
ALLPOSITIONS >> (NotebookDirectory[] <> "Data/positions.txt");
```

Plots

In[74]:=
```
L = 60;
ALLPOSITIONS = << (NotebookDirectory[] <> "Data/positions.txt");
```

In[76]:=
```
ALLPOSITIONS[[-1]]
```

Out[76]=
```
{{1, 1} → -0.0364219, {1, 2} → 0.248445, {1, 3} → 0.281665,
 ··· 3594 ···, {60, 58} → 0.0300734, {60, 59} → 0.252991, {60, 60} → 0.122429}
```

large output | show less | show more | show all | set size limit...

Saving the evolution for easy manipulation

In[66]:=
```
solution[[1]]
```

Out[66]=
```
x[1, 1][t] → InterpolatingFunction[ Domain: {{0., 500.}} Output: scalar ][t]
```

```
POSITIONS[t_] = Flatten@Table[{a, b} → x[a, b][t], {a, L}, {b, L}] /. solution;
VELOCITIES[t_] = Flatten@Table[{a, b} → x[a, b][t], {a, L}, {b, L}] /. solution;
ENERGIES[t_] = Flatten@Table[{a, b} → x[a, b][t], {a, L}, {b, L}] /. solution;
```

In[71]:=
```
ALLPOSITIONS = Table[POSITIONS[t], {t, 0, tmax, 1}];
```

In[73]:=
```
ALLPOSITIONS >> (NotebookDirectory[] <> "Data/positions.txt");
```

Plots

```
L = 60;
ALLPOSITIONS = << (NotebookDirectory[] <> "Data/positions.txt");
ALLVELOCITIES = << (NotebookDirectory[] <> "Data/velocities.txt")
ALLENERGIES = << (NotebookDirectory[] <> "Data/energies.txt")
```

In[76]:=
```
ALLPOSITIONS[[-1]]
```

Out[76]=
```
{{1, 1} → -0.0364219, {1, 2} → 0.248445, {1, 3} → 0.281665,
```

```
In[66]:= solution[[1]]

Out[66]=
x[1, 1][t] → InterpolatingFunction[ Domain: {{0., 500.}} Output: scalar ][t]
```

```
POSITIONS[t_] = Flatten@Table[{a, b} → x[a, b][t], {a, L}, {b, L}] /. solution;
VELOCITIES[t_] = Flatten@Table[{a, b} → x[a, b]'[t], {a, L}, {b, L}] /. solution;
ENERGIES[t_] =
  Flatten@
   Table[{a, b} → 1/2 x[a, b]'[t]^2 + ω^2 x[a, b][t]^2/2 + dial[t] μ^2/4 x[a, b][t]^4 +
      κ/4 Sum[(x[a, b][t] - x[a + l, b][t])^2 + (x[a, b][t] - x[a, b + l][t])^2, {l, -1, 1}], {a, L}, {b, L}] /. solution;
```

```
In[71]:= ALLPOSITIONS = Table[POSITIONS[t], {t, 0, tmax, 1}];

In[73]:= ALLPOSITIONS >> (NotebookDirectory[] <> "Data/positions.txt");
```

Plots

```
L = 60;
ALLPOSITIONS = << (NotebookDirectory[] <> "Data/positions.txt");
ALLVELOCITIES = << (NotebookDirectory[] <> "Data/velocities.txt")
ALLENERGIES = << (NotebookDirectory[] <> "Data/energies.txt")

In[76]:= ALLPOSITIONS[[-1]]
```

```
Cylinder[{{L, 1, 0}, {L, 1, -3}}], Cylinder[{{L, L, 0}, {L, L, -3}}]}];
```

```
In[88]:= Manipulate[ListPointPlot3D[ToPlot[t]], {t, 0, 500, 1}]
```

Saving the evolution for easy manipulation

```
solution[[1]];

POSITIONS[t_] = Flatten@Table[{a, b} → x[a, b][t], {a, L}, {b, L}] /. solution;
VELOCITIES[t_] = Flatten@Table[{a, b} → x[a, b]'[t], {a, L}, {b, L}] /. solution;
ENERGIES[t_] =
  Flatten@
```

$$\text{Table}\Big[\{a, b\} \to \frac{1}{2}\, x[a, b]'[t]^2 + \omega^2\, \frac{x[a, b][t]^2}{2} + \text{dial}[t]\, \frac{\mu^2}{4}\, x[a, b][t]^4 + \frac{\kappa}{4}\, \text{Sum}\big[(x[a, b][t] - x[a + l, b][t])^2 + (x[a, b][t] - x[a, b + l][t])^2, \{l, \{-1, 1\}\}\big], \{a, L\}, \{b, L\}\Big] \;/.$$

```
    solution;

In[71]:= ALLPOSITIONS = Table[POSITIONS[t], {t, 0, tmax, 1}];

In[73]:= ALLPOSITIONS >> (NotebookDirectory[] <> "Data/positions.txt");
```

Plots

```
L = 60;
ALLPOSITIONS = << (NotebookDirectory[] <> "Data/positions.txt");
ALLVELOCITIES = << (NotebookDirectory[] <> "Data/velocities.txt")
ALLENERGIES = << (NotebookDirectory[] <> "Data/energies.txt")
```

```
In[88]:= Manipulate[ListPointPlot3D[ToPlot[t]], {t, 0, 500, 1}]
```

Saving the evolution for easy manipulation

```
solution[[1]];

In[89]:= POSITIONS[t_] = Flatten@Table[{a, b} → x[a, b][t], {a, L}, {b, L}] /. solution;
VELOCITIES[t_] = Flatten@Table[{a, b} → x[a, b]'[t], {a, L}, {b, L}] /. solution;
ENERGIES[t_] =
  Flatten@
   Table[{a, b} → 1/2 x[a, b]'[t]^2 + ω^2 x[a, b][t]^2/2 + dial[t] μ^2/4 x[a, b][t]^4 +
     κ/4 Sum[(x[a, b][t] - x[a + l, b][t])^2 + (x[a, b][t] - x[a, b + l][t])^2, {l, {-1, 1}}], {a, L}, {b, L}] /.
    solution;

In[71]:= ALLPOSITIONS = Table[POSITIONS[t], {t, 0, tmax, 1}];

ALLVELOCITIES = Table[VELOCITIES[t], {t, 0, tmax, 1}];

ALLENERGIES = Table[ENERGIES[t], {t, 0, tmax, 1}];

In[73]:= ALLPOSITIONS >> (NotebookDirectory[] <> "Data/positions.txt");
```

Plots

```
Cylinder[{{L, 1, 0}, {L, 1, -3}}], Cylinder[{{L, L, 0}, {L, L, -3}}]}],
```

```
In[88]:= Manipulate[ListPointPlot3D[ToPlot[t]], {t, 0, 500, 1}]
```

Saving the evolution for easy manipulation

```
solution[[1]];
```

```
In[89]:= POSITIONS[t_] = Flatten@Table[{a, b} → x[a, b][t], {a, L}, {b, L}] /. solution;
         VELOCITIES[t_] = Flatten@Table[{a, b} → x[a, b]'[t], {a, L}, {b, L}] /. solution;
         ENERGIES[t_] =
           Flatten@
             Table[{a, b} → 1/2 x[a, b]'[t]^2 + ω^2 x[a, b][t]^2/2 + dial[t] μ^2/4 x[a, b][t]^4 +
               κ/4 Sum[(x[a, b][t] - x[a + l, b][t])^2 + (x[a, b][t] - x[a, b + l][t])^2, {l, {-1, 1}}], {a, L}, {b, L}] /.
            solution;
```

```
In[71]:= ALLPOSITIONS = Table[POSITIONS[t], {t, 0, tmax, 1}];

         ALLVELOCITIES = Table[VELOCITIES[t], {t, 0, tmax, 1}];

         ALLENERGIES = Monitor[Table[ENERGIES[t], {t, 0, tmax, 1}], ProgressIndicator[t, {0, tmax}]]
```

```
In[73]:= ALLPOSITIONS >> (NotebookDirectory[] <> "Data/positions.txt");
```

Plots

```
In[88]:= Manipulate[ListPointPlot3D[ToPlot[t]], {t, 0, 500, 1}]
```

Saving the evolution for easy manipulation

```
solution[[1]];

In[89]:= POSITIONS[t_] = Flatten@Table[{a, b} → x[a, b][t], {a, L}, {b, L}] /. solution;
VELOCITIES[t_] = Flatten@Table[{a, b} → x[a, b]'[t], {a, L}, {b, L}] /. solution;
ENERGIES[t_] =
  Flatten@
   Table[{a, b} → 1/2 x[a, b]'[t]^2 + ω^2 x[a, b][t]^2/2 + dial[t] μ^2/4 x[a, b][t]^4 +
      κ/4 Sum[(x[a, b][t] - x[a + l, b][t])^2 + (x[a, b][t] - x[a, b + l][t])^2, {l, {-1, 1}}], {a, L}, {b, L}] /.
    solution;

In[71]:= ALLPOSITIONS = Table[POSITIONS[t], {t, 0, tmax, 1}];

ALLVELOCITIES = Table[VELOCITIES[t], {t, 0, tmax, 1}];

ALLENERGIES = Monitor[Table[ENERGIES[t], {t, 0, tmax, 1}], ProgressIndicator[t, {0, tmax}]];

In[73]:= ALLPOSITIONS >> (NotebookDirectory[] <> "Data/positions.txt");
```

Plots

```
Flatten@
  Table[{a, b} → 1/2 x[a, b]'[t]^2 + ω^2 x[a, b][t]^2/2 + dial[t] μ^2/4 x[a, b][t]^4 +
     κ/4 Sum[(x[a, b][t] - x[a + l, b][t])^2 + (x[a, b][t] - x[a, b + l][t])^2, {l, {-1, 1}}], {a, L}, {b, L}] /.
   solution;
```

```
In[71]:= ALLPOSITIONS = Table[POSITIONS[t], {t, 0, tmax, 1}];

ALLVELOCITIES = Table[VELOCITIES[t], {t, 0, tmax, 1}];

ALLENERGIES = Monitor[Table[ENERGIES[t], {t, 0, tmax, 1}], ProgressIndicator[t, {0, tmax}]];

ALLPOSITIONS >> (NotebookDirectory[] <> "Data/positions.txt");
ALLVELOCITIES >> (NotebookDirectory[] <> "Data/velocities.txt");
ALLENERGIES >> (NotebookDirectory[] <> "Data/energies.txt");
```

Plots

```
L = 60;
ALLPOSITIONS = << (NotebookDirectory[] <> "Data/positions.txt");
ALLVELOCITIES = << (NotebookDirectory[] <> "Data/velocities.txt")
ALLENERGIES = << (NotebookDirectory[] <> "Data/energies.txt")

In[76]:= ALLPOSITIONS[[-1]]

Out[76]=
{{1, 1} → -0.0364219, {1, 2} → 0.248445, {1, 3} → 0.281665,
```

```
ALLENERGIES >> (NotebookDirectory[] <> "Data/energies.txt");
```

Plots

```
L = 60;
ALLPOSITIONS = << (NotebookDirectory[] <> "Data/positions.txt");
ALLVELOCITIES = << (NotebookDirectory[] <> "Data/velocities.txt")
ALLENERGIES = << (NotebookDirectory[] <> "Data/energies.txt")
```

```
solution;
```

In[93]:=
```
ALLPOSITIONS = Table[POSITIONS[t], {t, 0, tmax, 1}];
```

In[92]:=
```
ALLVELOCITIES = Table[VELOCITIES[t], {t, 0, tmax, 1}];
```

In[94]:=
```
ALLENERGIES = Monitor[Table[ENERGIES[t], {t, 0, tmax, 1}], ProgressIndicator[t, {0, tmax}]];
```

In[95]:=
```
ALLPOSITIONS >> (NotebookDirectory[] <> "Data/positions.txt");
ALLVELOCITIES >> (NotebookDirectory[] <> "Data/velocities.txt");
ALLENERGIES >> (NotebookDirectory[] <> "Data/energies.txt");
```

Plots

```
L = 60;
ALLPOSITIONS = << (NotebookDirectory[] <> "Data/positions.txt");
ALLVELOCITIES = << (NotebookDirectory[] <> "Data/velocities.txt")
ALLENERGIES = << (NotebookDirectory[] <> "Data/energies.txt")
```

In[98]:=
```
ALLENERGIES[[-1]]
```

Out[98]=

$\{\{1, 1\} \to 0.712968 + \frac{1}{2} x[1, 1]'[500]^2, \{1, 2\} \to 1.08788 + \frac{1}{2}$ ⋯1⋯ ^{2}, ⋯3596⋯, $\{60, 59\} \to$ ⋯19⋯ + ⋯1⋯, $\{60, 60\} \to 0.21426 + \frac{1}{2}$ ⋯1⋯ $'[500]^2\}$

large output | show less | show more | show all | set size limit...

```
    4
  solution;
```

In[93]:=
```
ALLPOSITIONS = Table[POSITIONS[t], {t, 0, tmax, 1}];
```

In[92]:=
```
ALLVELOCITIES = Table[VELOCITIES[t], {t, 0, tmax, 1}];
```

In[94]:=
```
ALLENERGIES = Monitor[Table[ENERGIES[t], {t, 0, tmax, 1}], ProgressIndicator[t, {0, tmax}]];
```

In[95]:=
```
ALLPOSITIONS >> (NotebookDirectory[] <> "Data/positions.txt");
ALLVELOCITIES >> (NotebookDirectory[] <> "Data/velocities.txt");
ALLENERGIES >> (NotebookDirectory[] <> "Data/energies.txt");
```

Plots

```
L = 60;
ALLPOSITIONS = << (NotebookDirectory[] <> "Data/positions.txt");
ALLVELOCITIES = << (NotebookDirectory[] <> "Data/velocities.txt")
ALLENERGIES = << (NotebookDirectory[] <> "Data/energies.txt")
```

In[102]:=
```
ALLVELOCITIES[[-2]] /. solution
```

Out[102]=
```
{{1, 1} → x[1, 1]'[499], {1, 2} → x[1, 2]'[499], {1, 3} → x[1, 3]'[499], ... 3594 ...,
 {60, 58} → x[60, 58]'[499], {60, 59} → x[60, 59]'[499], {60, 60} → x[60, 60]'[499]}
```

large output | show less | show more | show all | set size limit...

200%

```
In[87]:= legs = Graphics3D[{Cylinder[{{1, 1, 0}, {1, 1, -3}}], Cylinder[{{1, L, 0}, {1, L, -3}}],
     Cylinder[{{L, 1, 0}, {L, 1, -3}}], Cylinder[{{L, L, 0}, {L, L, -3}}]}];

In[88]:= Manipulate[ListPointPlot3D[ToPlot[t]], {t, 0, 500, 1}]
```

Saving the evolution for easy manipulation

```
solution[[1]];

POSITIONS[t_] = Flatten@Table[{a, b} → x[a, b][t], {a, L}, {b, L}] /. solution;
VELOCITIES[t_] = Flatten@Table[{a, b} → x[a, b]'[t], {a, L}, {b, L}] /. solution;
ENERGIES[t_] =
  Flatten@
   Table[{a, b} → 1/2 x[a, b]'[t]^2 + ω^2 x[a, b][t]^2/2 + dial[t] μ^2/4 x[a, b][t]^4 +
     κ/4 Sum[(x[a, b][t] - x[a + l, b][t])^2 + (x[a, b][t] - x[a, b + l][t])^2, {l, {-1, 1}}], {a, L}, {b, L}] /.
   solution;

In[93]:= ALLPOSITIONS = Table[POSITIONS[t], {t, 0, tmax, 1}];

In[92]:= ALLVELOCITIES = Table[VELOCITIES[t], {t, 0, tmax, 1}];

In[94]:= ALLENERGIES = Monitor[Table[ENERGIES[t], {t, 0, tmax, 1}], ProgressIndicator[t, {0, tmax}]];

In[95]:= ALLPOSITIONS >> (NotebookDirectory[] <> "Data/positions.txt");
ALLVELOCITIES >> (NotebookDirectory[] <> "Data/velocities.txt");
ALLENERGIES >> (NotebookDirectory[] <> "Data/energies.txt");
```

```
In[87]:= legs = Graphics3D[{Cylinder[{{1, 1, 0}, {1, 1, -3}}], Cylinder[{{1, L, 0}, {1, L, -3}}],
     Cylinder[{{L, 1, 0}, {L, 1, -3}}], Cylinder[{{L, L, 0}, {L, L, -3}}]}];

In[88]:= Manipulate[ListPointPlot3D[ToPlot[t]], {t, 0, 500, 1}]
```

Saving the evolution for easy manipulation

```
solution[[1]];

In[104]:=
VELOCITIES[t][[1]]

Out[104]=
{1, 1} → x[1, 1]′[t]
```

```
POSITIONS[t_] = Flatten@Table[{a, b} → x[a, b][t], {a, L}, {b, L}] /. solution;
VELOCITIES[t_] = Flatten@Table[{a, b} → x[a, b]'[t], {a, L}, {b, L}] /. solution;
ENERGIES[t_] =
  Flatten@
   Table[{a, b} → 1/2 x[a, b]'[t]^2 + ω^2 x[a, b][t]^2/2 + dial[t] μ^2/4 x[a, b][t]^4 +
     κ/4 Sum[(x[a, b][t] - x[a + l, b][t])^2 + (x[a, b][t] - x[a, b + l][t])^2, {l, {-1, 1}}], {a, L}, {b, L}] /.
    solution;

In[93]:= ALLPOSITIONS = Table[POSITIONS[t], {t, 0, tmax, 1}];

In[92]:= ALLVELOCITIES = Table[VELOCITIES[t], {t, 0, tmax, 1}];

In[94]:= ALLENERGIES = Monitor[Table[ENERGIES[t], {t, 0, tmax, 1}], ProgressIndicator[t, {0, tmax}]];
```

```
In[87]:= legs = Graphics3D[{Cylinder[{{1, 1, 0}, {1, 1, -3}}], Cylinder[{{1, L, 0}, {1, L, -3}}],
      Cylinder[{{L, 1, 0}, {L, 1, -3}}], Cylinder[{{L, L, 0}, {L, L, -3}}]}];

In[88]:= Manipulate[ListPointPlot3D[ToPlot[t]], {t, 0, 500, 1}]
```

Saving the evolution for easy manipulation

```
solution[[1]];

In[105]:=
VELOCITIES[t][[1]] /. solution

Out[105]=
{1, 1} → x[1, 1]′[t]

In[106]:=
solution[[1]]

Out[106]=
x[1, 1][t] → InterpolatingFunction[ Domain: {{0., 500.}} Output: scalar ][t]
```

```
POSITIONS[t_] = Flatten@Table[{a, b} → x[a, b][t], {a, L}, {b, L}] /. solution;
VELOCITIES[t_] = Flatten@Table[{a, b} → x[a, b]'[t], {a, L}, {b, L}] /. solution;
ENERGIES[t_] =
  Flatten@
    Table[{a, b} → 1/2 x[a, b]'[t]^2 + ω^2 x[a, b][t]^2/2 + dial[t] μ^2/4 x[a, b][t]^4 +
      κ/4 Sum[(x[a, b][t] - x[a + l, b][t])^2 + (x[a, b][t] - x[a, b + l][t])^2, {l, {-1, 1}}], {a, L}, {b, L}] /.
```

```
In[87]:= legs = Graphics3D[{Cylinder[{{1, 1, 0}, {1, 1, -3}}], Cylinder[{{1, L, 0}, {1, L, -3}}],
     Cylinder[{{L, 1, 0}, {L, 1, -3}}], Cylinder[{{L, L, 0}, {L, L, -3}}]}];

In[88]:= Manipulate[ListPointPlot3D[ToPlot[t]], {t, 0, 500, 1}]
```

Saving the evolution for easy manipulation

```
solution[[1]];

In[105]:=
VELOCITIES[t][[1]] /. solution
Out[105]=
{1, 1} → x[1, 1]′[t]

In[114]:=
solution[[1]]
Out[114]=
x[1, 1][t] → InterpolatingFunction[ Domain: {{0., 500.}} Output: scalar ][t]

In[113]:=
D[x[1, 1][t], t] /. solution[[1]]
Out[113]=
x[1, 1]′[t]

POSITIONS[t_] = Flatten@Table[{a, b} → x[a, b][t], {a, L}, {b, L}] /. solution;
VELOCITIES[t_] = Flatten@Table[{a, b} → x[a, b]'[t], {a, L}, {b, L}] /. solution;
ENERGIES[t_] =
```

```
In[87]:= legs = Graphics3D[{Cylinder[{{1, 1, 0}, {1, 1, -3}}], Cylinder[{{1, L, 0}, {1, L, -3}}],
      Cylinder[{{L, 1, 0}, {L, 1, -3}}], Cylinder[{{L, L, 0}, {L, L, -3}}]}];

In[88]:= Manipulate[ListPointPlot3D[ToPlot[t]], {t, 0, 500, 1}]
```

Saving the evolution for easy manipulation

```
solution[[1]];

In[105]:=
VELOCITIES[t][[1]] /. solution
Out[105]=
{1, 1} → x[1, 1]′[t]

In[114]:=
solution[[1]]

x[1,

In[113]:=
D[x[1, 1][t], t] /. solution[[1]]
Out[113]=
x[1, 1]′[t]

POSITIONS[t_] = Flatten@Table[{a, b} → x[a, b][t], {a, L}, {b, L}] /. solution;
VELOCITIES[t_] = Flatten@Table[{a, b} → x[a, b]'[t], {a, L}, {b, L}] /. solution;
ENERGIES[t_] =
  Flatten@
    Table[{a, b} → 1/2 x[a, b]'[t]^2 + μ^2 x[a, b][t]^2/... + dial[t] μ^2/... x[a, b][t]^4 + ...
```

```
In[87]:= legs = Graphics3D[{Cylinder[{{1, 1, 0}, {1, 1, -3}}], Cylinder[{{1, L, 0}, {1, L, -3}}],
     Cylinder[{{L, 1, 0}, {L, 1, -3}}], Cylinder[{{L, L, 0}, {L, L, -3}}]}];

In[88]:= Manipulate[ListPointPlot3D[ToPlot[t]], {t, 0, 500, 1}]
```

Saving the evolution for easy manipulation

```
solution[[1]];

In[105]:=
VELOCITIES[t][[1]] /. solution

Out[105]=
{1, 1} → x[1, 1]′[t]

In[114]:=
solution[[1]]

x[1, 1][t] /. solution //

Out[115]=
InterpolatingFunction[ Domain: {{0., 500.}} Output: scalar ][t]

In[113]:=
D[x[1, 1][t], t] /. solution[[1]]

Out[113]=
x[1, 1]′[t]

POSITIONS[t_] = Flatten@Table[{a, b} → x[a, b][t], {a, L}, {b, L}] /. solution;
VELOCITIES[t_] = Flatten@Table[{a, b} → x[a, b]'[t], {a, L}, {b, L}] /. solution;
```

```
In[87]:= legs = Graphics3D[{Cylinder[{{1, 1, 0}, {1, 1, -3}}], Cylinder[{{1, L, 0}, {1, L, -3}}],
      Cylinder[{{L, 1, 0}, {L, 1, -3}}], Cylinder[{{L, L, 0}, {L, L, -3}}]}];

In[88]:= Manipulate[ListPointPlot3D[ToPlot[t]], {t, 0, 500, 1}]
```

Saving the evolution for easy manipulation

```
solution[[1]];

In[105]:=
VELOCITIES[t][[1]] /. solution

Out[105]=
{1, 1} → x[1, 1]′[t]

In[114]:=
solution[[1]]

In[116]:=
x[1, 1][t] /. solution // D[#, t] &

Out[116]=
InterpolatingFunction[ Domain: {{0., 500.}}
                       Output: scalar ][t]

In[113]:=
D[x[1, 1][t], t] /. solution[[1]]

Out[113]=
x[1, 1]′[t]

POSITIONS[t_] = Flatten@Table[{a, b} → x[a, b][t], {a, L}, {b, L}] /. solution;
```

In[87]:=
```
legs = Graphics3D[{Cylinder[{{1, 1, 0}, {1, 1, -3}}], Cylinder[{{1, L, 0}, {1, L, -3}}],
    Cylinder[{{L, 1, 0}, {L, 1, -3}}], Cylinder[{{L, L, 0}, {L, L, -3}}]}];
```

In[88]:=
```
Manipulate[ListPointPlot3D[ToPlot[t]], {t, 0, 500, 1}]
```

Saving the evolution for easy manipulation

```
solution[[1]];
```

In[105]:=
```
VELOCITIES[t][[1]] /. solution
```

Out[105]=
```
{1, 1} → x[1, 1]′[t]
```

In[114]:=
```
solution[[1]]
```

In[117]:=
```
x[1, 1]'[t] /. solution
```

Out[117]=
```
x[1, 1]′[t]
```

In[116]:=
```
x[1, 1][t] /. solution // D[#, t] &
```

Out[116]=
```
InterpolatingFunction[ Domain: {{0., 500.}} Output: scalar ][t]
```

In[113]:=
```
D[x[1, 1][t], t] /. solution[[1]]
```

200%

In[114]:=

```
solution[[1]]

x[1, 1][t] /. solution
```

Out[117]=

```
x[1, 1]'[t]
```

In[116]:=

```
x[1, 1][t] /. solution // D[#, t] &
```

Out[116]=

```
InterpolatingFunction[ Domain: {{0., 500.}} Output: scalar ][t]
```

In[120]:=

```
solution[[1]] /. g_[t] :> g
```

Out[120]=

```
x[1, 1] → InterpolatingFunction[ Domain: {{0., 500.}} Output: scalar ]
```

```
POSITIONS[t_] = Flatten@Table[{a, b} → x[a, b][t], {a, L}, {b, L}] /. solution;
VELOCITIES[t_] = Flatten@Table[{a, b} → x[a, b]'[t], {a, L}, {b, L}] /. solution;
ENERGIES[t_] =
  Flatten@
   Table[{a, b} → 1/2 x[a, b]'[t]^2 + ω^2 x[a, b][t]^2/2 + dial[t] μ^2/4 x[a, b][t]^4 +
     κ/4 Sum[(x[a, b][t] - x[a + l, b][t])^2 + (x[a, b][t] - x[a, b + l][t])^2, {l, {-1, 1}}], {a, L}, {b, L}] /.
    solution;
```

```
In[114]:=
solution[[1]]

x[1, 1][t] /. solution
Out[117]=
x[1, 1]′[t]

In[116]:=
x[1, 1][t] /. solution // D[#, t] &
Out[116]=
InterpolatingFunction[ Domain: {{0., 500.}}
                       Output: scalar ][t]

In[121]:=
solution[[1]] /. g_[t] :→ g
x[1, 1]'[t] /. %
Out[121]=
x[1, 1] → InterpolatingFunction[ Domain: {{0., 500.}}
                                 Output: scalar ]

Out[122]=
InterpolatingFunction[ Domain: {{0., 500.}}
                       Output: scalar ][t]

POSITIONS[t_] = Flatten@Table[{a, b} → x[a, b][t], {a, L}, {b, L}] /. solution;
VELOCITIES[t_] = Flatten@Table[{a, b} → x[a, b]'[t], {a, L}, {b, L}] /. solution;
ENERGIES[t_] =
  Flatten@
       1                  x[a, b][t]^2          μ^2
```

```
2
Plot[dial[t], {t, 0, 500}];

eqs =
  Table[∂{t,2} x[i, j][t] + ω^2 x[i, j][t] + dial[t] μ^2 x[i, j][t]^3 +
     κ Sum[x[i, j][t] - x[i + l, j][t] + x[i, j][t] - x[i, j + l][t], {l, {-1, 1}}] == 0, {i, L}, {j, L}] // Flatten;

In[83]:= bc = Table[{x[i, j][0] == 0, x[i, j]'[0] == If[i > 3 L / 4 ∧ j > 3 L / 4, RandomReal[{4, 6}], 0]}, {i, L}, {j, L}] //
   Flatten;

In[84]:= var = Table[x[i, j][t], {i, L}, {j, L}] // Flatten;

solution = NDSolve[Join[eqs, bc], var, {t, 0, tmax}][[1]] /. g_[t] :> g;

In[86]:= ToPlot[t_] = var /. x[i_, j_][t] -> {i, j, x[i, j][t]} /. solution;

In[87]:= legs = Graphics3D[{Cylinder[{{1, 1, 0}, {1, 1, -3}}], Cylinder[{{1, L, 0}, {1, L, -3}}],
    Cylinder[{{L, 1, 0}, {L, 1, -3}}], Cylinder[{{L, L, 0}, {L, L, -3}}]}];

In[88]:= Manipulate[ListPointPlot3D[ToPlot[t]], {t, 0, 500, 1}]
```

Saving the evolution for easy manipulation

```
solution[[1]];

In[114]:=
solution[[1]]

x[1, 1][t] /. solution

Out[117]=
```

200%

```
ALLVELOCITIES >> (NotebookDirectory[] <> "Data/velocities.txt");
ALLENERGIES >> (NotebookDirectory[] <> "Data/energies.txt");
```

Plots

```
L = 60;
ALLPOSITIONS = << (NotebookDirectory[] <> "Data/positions.txt");
ALLVELOCITIES = << (NotebookDirectory[] <> "Data/velocities.txt")
ALLENERGIES = << (NotebookDirectory[] <> "Data/energies.txt")
```

In[102]:=

```
ALLVELOCITIES[[-2]] /. solution
```

Out[102]=

```
{{1, 1} → x[1, 1]′[499], {1, 2} → x[1, 2]′[499], {1, 3} → x[1, 3]′[499], ··· 3594 ···,
 {60, 58} → x[60, 58]′[499], {60, 59} → x[60, 59]′[499], {60, 60} → x[60, 60]′[499]}
```

large output | show less | show more | show all | set size limit...

```
ALLENERGIES[[-1]]
```

200%

```
ALLVELOCITIES >> (NotebookDirectory[] <> "Data/velocities.txt");
ALLENERGIES >> (NotebookDirectory[] <> "Data/energies.txt");
```

Plots

```
L = 60;
ALLPOSITIONS = << (NotebookDirectory[] <> "Data/positions.txt");
ALLVELOCITIES = << (NotebookDirectory[] <> "Data/velocities.txt")
ALLENERGIES = << (NotebookDirectory[] <> "Data/energies.txt")
```

In[102]:=

```
ALLVELOCITIES[[-2]] /. solution
```

Out[102]=

```
{{1, 1} → x[1, 1]′[499], {1, 2} → x[1, 2]′[499], {1, 3} → x[1, 3]′[499], ··· 3594 ···,
 {60, 58} → x[60, 58]′[499], {60, 59} → x[60, 59]′[499], {60, 60} → x[60, 60]′[499]}
```

large output | show less | show more | show all | set size limit...

```
ALLENERGIES[[-1]]

ALLENERGIES[[-1]] // SparseArray
```

Plots

```
L = 60;
ALLPOSITIONS = << (NotebookDirectory[] <> "Data/positions.txt");
ALLVELOCITIES = << (NotebookDirectory[] <> "Data/velocities.txt")
ALLENERGIES = << (NotebookDirectory[] <> "Data/energies.txt")
```

In[102]:=

```
ALLVELOCITIES[[-2]] /. solution
```

Out[102]=

```
{{1, 1} → x[1, 1]'[499], {1, 2} → x[1, 2]'[499], {1, 3} → x[1, 3]'[499], ··· 3594 ··· ,
 {60, 58} → x[60, 58]'[499], {60, 59} → x[60, 59]'[499], {60, 60} → x[60, 60]'[499]}
```

large output | show less | show more | show all | set size limit...

```
ALLENERGIES[[-1]]

ALLENERGIES[[-1]] // SparseArray

SparseArray /@ ALLENERGIES
```

200%

Plots

```
L = 60;
ALLPOSITIONS = << (NotebookDirectory[] <> "Data/positions.txt");
ALLVELOCITIES = << (NotebookDirectory[] <> "Data/velocities.txt")
ALLENERGIES = << (NotebookDirectory[] <> "Data/energies.txt")
```

In[102]:=

```
ALLVELOCITIES[[-2]] /. solution
```

Out[102]=

```
{{1, 1} → x[1, 1]′[499], {1, 2} → x[1, 2]′[499], {1, 3} → x[1, 3]′[499], … 3594 …,
 {60, 58} → x[60, 58]′[499], {60, 59} → x[60, 59]′[499], {60, 60} → x[60, 60]′[499]}
```

large output | show less | show more | show all | set size limit...

```
ALLENERGIES[[-1]]

ALLENERGIES[[-1]] // SparseArray

energiesMatrices = SparseArray /@ ALLENERGIES
```

200%

Plots

```
L = 60;
ALLPOSITIONS = << (NotebookDirectory[] <> "Data/positions.txt");
ALLVELOCITIES = << (NotebookDirectory[] <> "Data/velocities.txt")
ALLENERGIES = << (NotebookDirectory[] <> "Data/energies.txt")
```

In[102]:=

```
ALLVELOCITIES[[-2]] /. solution
```

Out[102]=

```
{{1, 1} → x[1, 1]′[499], {1, 2} → x[1, 2]′[499], {1, 3} → x[1, 3]′[499], ··· 3594 ···,
 {60, 58} → x[60, 58]′[499], {60, 59} → x[60, 59]′[499], {60, 60} → x[60, 60]′[499]}
```

large output | show less | show more | show all | set size limit...

```
ALLENERGIES[[-1]]

ALLENERGIES[[-1]] // SparseArray

energiesMatrices = SparseArray /@ ALLENERGIES;

Table[Mean@Mean@(energiesMatrices[[j]]), {j, 1, tmax}]
```

tmax

200%

Plots

In[134]:=

```
L = 60;
ALLPOSITIONS = << (NotebookDirectory[] <> "Data/positions.txt");
ALLVELOCITIES = << (NotebookDirectory[] <> "Data/velocities.txt")
ALLENERGIES = << (NotebookDirectory[] <> "Data/energies.txt")
```

In[102]:=

```
ALLVELOCITIES[[-2]] /. solution
```

Out[102]=

```
{{1, 1} → x[1, 1]′[499], {1, 2} → x[1, 2]′[499], {1, 3} → x[1, 3]′[499], ··· 3594 ···,
 {60, 58} → x[60, 58]′[499], {60, 59} → x[60, 59]′[499], {60, 60} → x[60, 60]′[499]}
```

large output | show less | show more | show all | set size limit...

```
ALLENERGIES[[-1]]

ALLENERGIES[[-1]] // SparseArray

energiesMatrices = SparseArray /@ ALLENERGIES;

Table[Mean@Mean@(energiesMatrices[[j]]), {j, 1, tmax}] (* == 1/β *)
```

200%

Plots

In[134]:=

```
L = 60;
ALLPOSITIONS = << (NotebookDirectory[] <> "Data/positions.txt");
ALLVELOCITIES = << (NotebookDirectory[] <> "Data/velocities.txt")
ALLENERGIES = << (NotebookDirectory[] <> "Data/energies.txt")
```

Out[136]=

```
$Aborted
```

Out[137]=

```
$Aborted
```

In[138]:=

```
ALLVELOCITIES[[-2]] /. solution
```

Out[138]=

```
{{1, 1} → 2.12164, {1, 2} → -0.557653, {1, 3} → -0.723602,
 ··· 3594 ···, {60, 58} → -0.702605, {60, 59} → -0.292339, {60, 60} → 1.64309}
```

large output | show less | show more | show all | set size limit...

```
ALLENERGIES[[-1]]

ALLENERGIES[[-1]] // SparseArray

energiesMatrices = SparseArray /@ ALLENERGIES;
```

200%

In[134]:=

```
L = 60;
ALLPOSITIONS = << (NotebookDirectory[] <> "Data/positions.txt");
ALLVELOCITIES = << (NotebookDirectory[] <> "Data/velocities.txt")
ALLENERGIES = << (NotebookDirectory[] <> "Data/energies.txt")
```

Out[136]=

```
$Aborted
```

Out[137]=

```
$Aborted
```

In[144]:=

```
ALLENERGIES[[-4]]
```

Out[144]=

```
{{1, 1} → 1.148, {1, 2} → 0.937023, {1, 3} → 0.0683552,
 … 3594 …, {60, 58} → 0.162234, {60, 59} → 0.841085, {60, 60} → 1.53002}
```

large output | show less | show more | show all | set size limit...

```
ALLENERGIES[[-1]] // SparseArray

energiesMatrices = SparseArray /@ ALLENERGIES;

Table[Mean@Mean@(energiesMatrices[[j]]), {j, 1, tmax}] (* == 1/β *)
```

```
··· 3594 ···, {60, 58} → 0.162234, {60, 59} → 0.841085, {60, 60} → 1.53002}
```

large output | show less | show more | show all | set size limit...

In[145]:=

```
ALLENERGIES[[-1]] // SparseArray
```

Out[145]=

```
SparseArray[ ⊞  Specified elements: 3600
                Dimensions: {60, 60} ]
```

In[146]:=

```
energiesMatrices = SparseArray /@ ALLENERGIES;
```

In[147]:=

```
Table[Mean@Mean@(energiesMatrices[[j]]), {j, 1, tmax}] (* ≡ 1/β *)
```

Out[147]=

```
{0.809413, 0.809413, 0.809413, 0.809413, 0.809412, 0.809412, 0.809412, 0.809412, 0.809412, 0.809412, 0.809412,
 0.809412, 0.809412, 0.809413, 0.809413, 0.809413, 0.809413, 0.809413, 0.809413, 0.809413, 0.809413, 0.809413,
 0.809413, 0.809413, 0.809413, 0.809413, 0.809413, 0.809413, 0.809413, 0.809413, 0.809413, 0.809413, 0.809413,
 0.809413, 0.809413, 0.809413, 0.809413, 0.809413, 0.809413, 0.809413, 0.809413, 0.809413, 0.809413,
 0.809413, 0.809413, 0.809413, 0.809413, 0.809413, 0.809413, 0.809413, 0.809413, 0.809413, 0.809413,
 0.809413, 0.809413, 0.809413, 0.809413, 0.809413, 0.809413, 0.809413, 0.809413, 0.809413, 0.809413,
 0.809413, 0.809413, 0.809413, 0.809413, 0.809413, 0.809413, 0.809413, 0.809413, 0.809413, 0.809413,
 0.809413, 0.809413, 0.809413, 0.809413, 0.809413, 0.809413, 0.809413, 0.809413, 0.809414, 0.809414,
 0.809414, 0.809414, 0.809414, 0.809414, 0.809414, 0.809414, 0.809414, 0.809414, 0.809414, 0.809414,
 0.809414, 0.809414, 0.809414, 0.809414, 0.809414, 0.809414, 0.809414, 0.809414, 0.809414, 0.809414,
 0.809414, 0.809414, 0.809414, 0.809414, 0.809414, 0.809414, 0.809414, 0.809414, 0.809414, 0.809414,
 0.809414, 0.809414, 0.809414, 0.809414, 0.809414, 0.809414, 0.809414, 0.809414, 0.809414, 0.809414,
```

```
··· 3594 ···, {60, 58} → 0.162234, {60, 59} → 0.841085, {60, 60} → 1.53002}
```

large output | show less | show more | show all | set size limit...

In[145]:=

```
ALLENERGIES[[-1]] // SparseArray
```

Out[145]=

SparseArray[ Specified elements: 3600 Dimensions: {60, 60}]

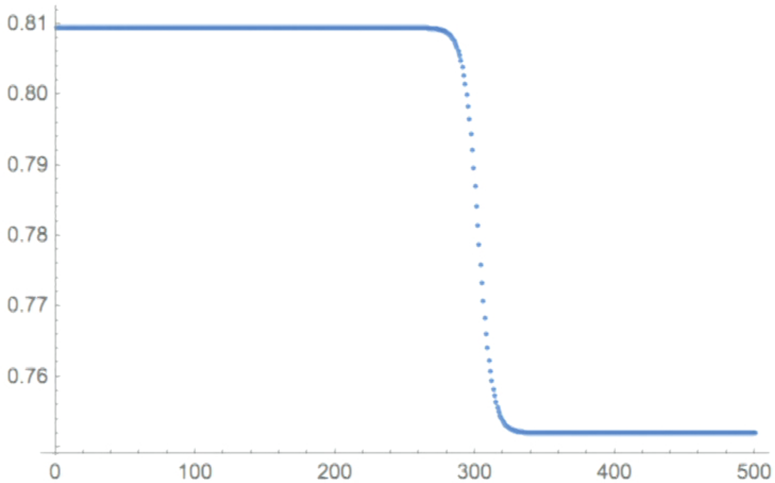
In[146]:=

```
energiesMatrices = SparseArray /@ ALLENERGIES;
```

In[148]:=

```
Table[Mean@Mean@(energiesMatrices[[j]]), {j, 1, tmax}] (* == 1/β *) // ListPlot
```

Out[148]=



```
In[123]:=
solution = NDSolve[Join[eqs, bc], var, {t, 0, tmax}][[1]] /. g_[t] :> g;

In[86]:= ToPlot[t_] = var /. x[i_, j_][t] -> {i, j, x[i, j][t]} /. solution;

In[87]:= legs = Graphics3D[{Cylinder[{{1, 1, 0}, {1, 1, -3}}], Cylinder[{{1, L, 0}, {1, L, -3}}],
     Cylinder[{{L, 1, 0}, {L, 1, -3}}], Cylinder[{{L, L, 0}, {L, L, -3}}]}];

Manipulate[ListPointPlot3D[ToPlot[t]], {t, 0, 500, 1}];
```

Saving the evolution for easy manipulation

```
In[125]:=
POSITIONS[t_] = Flatten@Table[{a, b} → x[a, b][t], {a, L}, {b, L}] /. solution;
VELOCITIES[t_] = Flatten@Table[{a, b} → x[a, b]'[t], {a, L}, {b, L}] /. solution;
ENERGIES[t_] =
  Flatten@
   Table[{a, b} → 1/2 x[a, b]'[t]^2 + ω^2 x[a, b][t]^2/2 + dial[t] μ^2/4 x[a, b][t]^4 +
      κ/4 Sum[(x[a, b][t] - x[a + l, b][t])^2 + (x[a, b][t] - x[a, b + l][t])^2, {l, {-1, 1}}], {a, L}, {b, L}] /.
    solution;

In[128]:=
ALLPOSITIONS = Table[POSITIONS[t], {t, 0, tmax, 1}];

In[129]:=
ALLVELOCITIES = Table[VELOCITIES[t], {t, 0, tmax, 1}];

In[130]:=
ALLENERGIES = Monitor[Table[ENERGIES[t], {t, 0, tmax, 1}], ProgressIndicator[t, {0, tmax}]];
```

large output | show less | show more | show all | set size limit...

In[145]:=

```
ALLENERGIES[[-1]] // SparseArray
```

Out[145]=

SparseArray[⊞ Specified elements: 3600 Dimensions: {60, 60}]

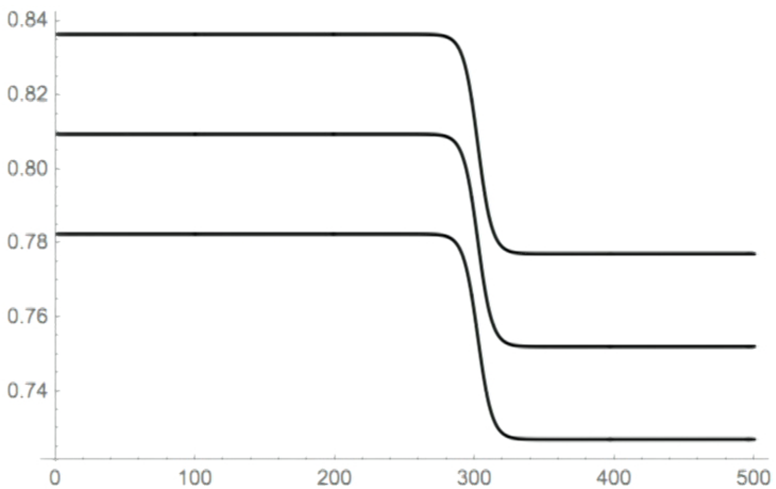
In[146]:=

```
energiesMatrices = SparseArray /@ ALLENERGIES;
```

In[149]:=

```
Table[Mean@Mean@(energiesMatrices[[j]]), {j, 1, tmax}] (* == 1/β *) //
 ListPlot[{#, # (1 + 2/L), # (1 - 2/L)}, PlotStyle → Black, Joined → True] &
```

Out[149]=



In[145]:=

```
ALLENERGIES[[-1]] // SparseArray
```

Out[145]=

```
SparseArray[ Specified elements: 3600
             Dimensions: {60, 60} ]
```

In[146]:=

```
energiesMatrices = SparseArray /@ ALLENERGIES;
```

In[150]:=

```
energyConservation = Table[Mean@Mean@(energiesMatrices[[j]]), {j, 1, tmax}] (* == 1/β *) //
  ListPlot[{#, # (1 + 2/L), # (1 - 2/L)}, PlotStyle → Black, Joined → True] &;
```

In[145]:=

```
ALLENERGIES[[-1]] // SparseArray
```

Out[145]=

```
SparseArray[ Specified elements: 3600
             Dimensions: {60, 60} ]
```

In[146]:=

```
energiesMatrices = SparseArray /@ ALLENERGIES;
```

In[150]:=

```
energyConservation = Table[Mean@Mean@(energiesMatrices[[j]]), {j, 1, tmax}] (* == 1/β *) //
   ListPlot[{#, # (1 + 2/L), # (1 - 2/L)}, PlotStyle → Black, Joined → True] &;

energiesMatrices[[t, l (j - 1) + 1 ;; l (i - 1) + 1]], {i, 1, L / l}, {j, 1, L / l}
```

In[145]:=

```
ALLENERGIES[[-1]] // SparseArray
```

Out[145]=

```
SparseArray[ Specified elements: 3600
             Dimensions: {60, 60} ]
```

In[146]:=

```
energiesMatrices = SparseArray /@ ALLENERGIES;
```

In[150]:=

```
energyConservation = Table[Mean@Mean@(energiesMatrices[[j]]), {j, 1, tmax}] (* == 1/β *) //
   ListPlot[{#, # (1 + 2/L), # (1 - 2/L)}, PlotStyle → Black, Joined → True] &;

energiesMatrices[[t, l (j - 1) + 1 ;; l j, l (i - 1) + 1]], {i, 1, L / l}, {j, 1, L / l}
```

In[145]:=

```
ALLENERGIES[[-1]] // SparseArray
```

Out[145]=

```
SparseArray[ Specified elements: 3600
             Dimensions: {60, 60} ]
```

In[146]:=

```
energiesMatrices = SparseArray /@ ALLENERGIES;
```

In[150]:=

```
energyConservation = Table[Mean@Mean@(energiesMatrices[[j]]), {j, 1, tmax}] (* == 1/β *) //
  ListPlot[{#, # (1 + 2/L), # (1 - 2/L)}, PlotStyle → Black, Joined → True] &;

energiesMatrices[[t, l (j - 1) + 1 ;; l j, l (i - 1) + 1, l i]], {i, 1, L / l}, {j, 1, L / l}
```

In[145]:=

```
ALLENERGIES[[-1]] // SparseArray
```

Out[145]=

```
SparseArray[ Specified elements: 3600
             Dimensions: {60, 60} ]
```

In[146]:=

```
energiesMatrices = SparseArray /@ ALLENERGIES;
```

In[150]:=

```
energyConservation = Table[Mean@Mean@(energiesMatrices[[j]]), {j, 1, tmax}] (* == 1/β *) //
   ListPlot[{#, # (1 + 2/L), # (1 - 2/L)}, PlotStyle → Black, Joined → True] &;

energiesFourCorners =
 Table[Table[Mean@Mean@(energiesMatrices[[t, l (j - 1) + 1 ;; l j, l (i - 1) + 1, l i]]), {i, 1, L / l}, {j, 1, L / l}],
  {n, 1, 500}]
```

In[145]:=

```
ALLENERGIES[[-1]] // SparseArray
```

Out[145]=

SparseArray[⊞ Specified elements: 3600 Dimensions: {60, 60}]

In[146]:=

```
energiesMatrices = SparseArray /@ ALLENERGIES;
```

In[150]:=

```
energyConservation = Table[Mean@Mean@(energiesMatrices[[j]]), {j, 1, tmax}] (* == 1/β *) //
   ListPlot[{#, # (1 + 2/L), # (1 - 2/L)}, PlotStyle → Black, Joined → True] &;
```

In[151]:=

```
energiesFourCorners =
 Table[Table[Mean@Mean@(energiesMatrices[[t, l (j - 1) + 1 ;; l j, l (i - 1) + 1, l i]]), {i, 1, L / l}, {j, 1, L / l}],
  {n, 1, 500}]
```

··· **Table**: Iterator $\{i, 1, \frac{60}{l}\}$ does not have appropriate bounds.

··· **Table**: Iterator $\{i, 1, \frac{60}{l}\}$ does not have appropriate bounds.

··· **Table**: Iterator $\{i, 1, \frac{60}{l}\}$ does not have appropriate bounds.

··· **General**: Further output of Table::iterb will be suppressed during this calculation.

··· **Table**: Iterator $\{i, 1, \frac{60}{l}\}$ does not have appropriate bounds.

200%

Out[145]=

SparseArray[Specified elements: 3600 Dimensions: {60, 60}]

In[146]:=

```
energiesMatrices = SparseArray /@ ALLENERGIES;
```

In[150]:=

```
energyConservation = Table[Mean@Mean@(energiesMatrices[[j]]), {j, 1, tmax}] (* == 1/β *) //
   ListPlot[{#, # (1 + 2/L), # (1 - 2/L)}, PlotStyle → Black, Joined → True] &;
```

In[152]:=

```
l = L / 2;
energiesFourCorners =
 Table[Table[Mean@Mean@(energiesMatrices[[t, l (j - 1) + 1 ;; l j, l (i - 1) + 1, l i]]), {i, 1, L / l}, {j, 1, L / l}],
  {n, 1, 500}]
```

··· Part: The expression t cannot be used as a part specification.

··· Part: The expression t cannot be used as a part specification.

··· Part: The expression t cannot be used as a part specification.

··· General: Further output of Part::pkspec1 will be suppressed during this calculation.

Out[153]=

$Aborted[]

```
SparseArray[ Specified elements: 3600
             Dimensions: {60, 60} ]

In[146]:=
energiesMatrices = SparseArray /@ ALLENERGIES;

In[150]:=
energyConservation = Table[Mean@Mean@(energiesMatrices[[j]]), {j, 1, tmax}] (* == 1/β *) //
  ListPlot[{#, # (1 + 2/L), # (1 - 2/L)}, PlotStyle → Black, Joined → True] &;

In[156]:=
l = L / 2;
energiesFourCorners =
 Table[Table[Mean@Mean@(energiesMatrices[[t, l (j - 1) + 1 ;; l j, l (i - 1) + 1 ;; l i]]), {i, 1, L / l}, {j, 1, L / l}],
  {t, 1, 500}]

Out[157]=
{{{0, 8.17424 × 10^-48}, {1.12882 × 10^-47, 3.23765}}, {{0.000510291, 0.0532233}, {0.0536415, 3.13028}},
 {{0.00452592, 0.0871111}, {0.0864609, 3.05955}}, {{0.00931268, 0.152758}, {0.146991, 2.92859}},
 {{0.0176598, 0.221099}, {0.211958, 2.78693}}, {{0.0247266, 0.290279}, {0.284581, 2.63806}},
 {{0.0323115, 0.223526}, {0.217742, 2.76407}}, {{0.0347856, 0.207599}, {0.189443, 2.80582}},
 {{0.0413473, 0.267199}, {0.260929, 2.66817}}, {{0.0337515, 0.282457}, {0.288265, 2.63318}},
 {{0.0301335, 0.300389}, {0.301258, 2.60587}}, {{0.0376081, 0.35912}, {0.361653, 2.47927}},
 {{0.0584431, 0.39465}, {0.417086, 2.36747}}, {{0.083077, 0.397541}, {0.452569, 2.30446}},
 {{0.104014, 0.393917}, {0.424829, 2.31489}}, {{0.114826, 0.385}, {0.397761, 2.34006}},
 {{0.124252, 0.370103}, {0.389537, 2.35376}}, {{0.134591, 0.382354}, {0.374094, 2.34661}},
 {{0.145081, 0.39757}, {0.359803, 2.3352}}, {{0.153643, 0.412836}, {0.367134, 2.30404}},
 {{0.138356, 0.454799}, {0.390894, 2.2536}}, {{0.126844, 0.488459}, {0.401327, 2.22102}},
 {{0.131048, 0.520616}, {0.414783, 2.1712}}, {{0.155491, 0.528052}, {0.444597, 2.10951}},
```

```
SparseArray[ Specified elements: 3600  Dimensions: {60, 60} ]

In[146]:=
energiesMatrices = SparseArray /@ ALLENERGIES;

In[150]:=
energyConservation = Table[Mean@Mean@(energiesMatrices[[j]]), {j, 1, tmax}] (* == 1/β *) //
   ListPlot[{#, # (1 + 2/L), # (1 - 2/L)}, PlotStyle → Black, Joined → True] &;

In[158]:=
l = L / 2;
energiesFourCorners =
 Table[Flatten@Table[Mean@Mean@(energiesMatrices[[t, l (j - 1) + 1 ;; l j, l (i - 1) + 1 ;; l i]]),
    {i, 1, L / l}, {j, 1, L / l}], {t, 1, 500}]

Out[159]=
{{0, 8.17424 × 10^-48, 1.12882 × 10^-47, 3.23765}, {0.000510291, 0.0532233, 0.0536415, 3.13028},
 {0.00452592, 0.0871111, 0.0864609, 3.05955}, {0.00931268, 0.152758, 0.146991, 2.92859},
 {0.0176598, 0.221099, 0.211958, 2.78693}, {0.0247266, 0.290279, 0.284581, 2.63806},
 {0.0323115, 0.223526, 0.217742, 2.76407}, {0.0347856, 0.207599, 0.189443, 2.80582},
 {0.0413473, 0.267199, 0.260929, 2.66817}, {0.0337515, 0.282457, 0.288265, 2.63318},
 {0.0301335, 0.300389, 0.301258, 2.60587}, {0.0376081, 0.35912, 0.361653, 2.47927},
 {0.0584431, 0.39465, 0.417086, 2.36747}, {0.083077, 0.397541, 0.452569, 2.30446},
 {0.104014, 0.393917, 0.424829, 2.31489}, {0.114826, 0.385, 0.397761, 2.34006},
 {0.124252, 0.370103, 0.389537, 2.35376}, {0.134591, 0.382354, 0.374094, 2.34661},
 {0.145081, 0.39757, 0.359803, 2.3352}, {0.153643, 0.412836, 0.367134, 2.30404},
 {0.138356, 0.454799, 0.390894, 2.2536}, {0.126844, 0.488459, 0.401327, 2.22102},
 {0.131048, 0.520616, 0.414783, 2.1712}, {0.155491, 0.528052, 0.444597, 2.10951},
```

200%

In[158]:=

```
l = L / 2;
energiesFourCorners =
 Table[Flatten@Table[Mean@Mean@(energiesMatrices[[t, l (j - 1) + 1 ;; l j, l (i - 1) + 1 ;; l i]]),
   {i, 1, L / l}, {j, 1, L / l}], {t, 1, 500}]
```

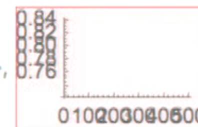
200%

In[150]:=

```
energyConservation = Table[Mean@Mean@(energiesMatrices[[j]]), {j, 1, tmax}] (* == 1/β *) //
   ListPlot[{#, # (1 + 2/L), # (1 - 2/L)}, PlotStyle → Black, Joined → True] &;

l = L / 2;
energiesFourCorners =
  Table[Flatten@Table[Mean@Mean@(energiesMatrices[[t, l (j - 1) + 1 ;; l j, l (i - 1) + 1 ;; l i]]),
      {i, 1, L / l}, {j, 1, L / l}], {t, 1, 500}] // ListPlot;
```

In[160]:=

```
Show[energiesFourCorners, energyConservation]
```

··· Show: Could not combine the graphics objects in Show[{{0, 8.17424 × 10^-48, 1.12882 × 10^-47, 3.23765}, ≪49≫, ≪450≫}, ].

Out[160]=

```
Show[{{0, 8.17424 × 10^-48, 1.12882 × 10^-47, 3.23765}, {0.000510291, 0.0532233, 0.0536415, 3.13028},
  {0.00452592, 0.0871111, 0.0864609, 3.05955}, {0.00931268, 0.152758, 0.146991, 2.92859},
  {0.0176598, 0.221099, 0.211958, 2.78693}, {0.0247266, 0.290279, 0.284581, 2.63806},
  {0.0323115, 0.223526, 0.217742, 2.76407}, {0.0347856, 0.207599, 0.189443, 2.80582},
  {0.0413473, 0.267199, 0.260929, 2.66817}, {0.0337515, 0.282457, 0.288265, 2.63318},
  {0.0301335, 0.300389, 0.301258, 2.60587}, {0.0376081, 0.35912, 0.361653, 2.47927},
  {0.0584431, 0.39465, 0.417086, 2.36747}, {0.083077, 0.397541, 0.452569, 2.30446},
  {0.104014, 0.393917, 0.424829, 2.31489}, {0.114826, 0.385, 0.397761, 2.34006},
  {0.124252, 0.370103, 0.389537, 2.35376}, {0.134591, 0.382354, 0.374094, 2.34661},
  {0.145081, 0.39757, 0.359803, 2.3352}, {0.153643, 0.412836, 0.367134, 2.30404},
  {0.138356, 0.454799, 0.390894, 2.2536}, {0.126844, 0.488459, 0.401327, 2.22102},
  {0.131048, 0.520616, 0.414783, 2.1712}, {0.155491, 0.528052, 0.444597, 2.10951},
```

200%

In[150]:=

```
energyConservation = Table[Mean@Mean@(energiesMatrices[[j]]), {j, 1, tmax}] (* ≡ 1/β *) //
    ListPlot[{#, # (1 + 2/L), # (1 - 2/L)}, PlotStyle → Black, Joined → True] &;
```

In[161]:=

```
l = L / 2;
energiesFourCorners =
 Table[Flatten@Table[Mean@Mean@(energiesMatrices[[t, l (j - 1) + 1 ;; l j, l (i - 1) + 1 ;; l i]]),
     {i, 1, L / l}, {j, 1, L / l}], {t, 1, 500}]ᵀ
```

Out[162]=

```
{{0, 0.000510291, 0.00452592, 0.00931268, 0.0176598, 0.0247266, 0.0323115, 0.0347856, 0.0413473, 0.0337515,
  0.0301335, 0.0376081, 0.0584431, 0.083077, 0.104014, 0.114826, 0.124252, 0.134591, 0.145081, 0.153643,
  0.138356, 0.126844, 0.131048, 0.155491, 0.170487, 0.174996, 0.193464, 0.213356, 0.233087, 0.261776, 0.2943,
  0.323565, 0.344327, 0.363035, 0.381296, 0.401461, 0.426978, 0.454972, 0.484933, 0.521359, 0.565183,
  0.594297, 0.62576, 0.647628, 0.679077, 0.714002, 0.736047, 0.747974, 0.767465, 0.789836, 0.816781, 0.83848,
  0.854331, 0.872294, 0.878009, 0.872108, 0.876, 0.890441, 0.898691, 0.904711, 0.905306, 0.897464, 0.898185,
  0.89493, 0.883591, 0.884083, 0.89177, 0.890249, 0.887698, 0.88518, 0.875782, 0.86703, 0.869273, 0.85804,
  0.847388, 0.844703, 0.84544, 0.847148, 0.840557, 0.834735, 0.842831, 0.848345, 0.834347, 0.821633, 0.818968,
  0.820964, 0.830265, 0.833658, 0.831439, 0.830181, 0.825906, 0.82286, 0.821737, 0.816199, 0.8117, 0.813909,
  0.814163, 0.813212, 0.815771, 0.817389, 0.80726, 0.7969, 0.78967, 0.79035, 0.786516, 0.770617, 0.761266,
  0.754115, 0.766417, 0.779823, 0.78375, 0.782449, 0.790528, 0.794505, 0.789543, 0.7813, 0.787674, 0.800933,
  0.80718, 0.812292, 0.824281, 0.830593, 0.830036, 0.824815, 0.817058, 0.813824, 0.818758, 0.827181,
  0.828489, 0.834383, 0.84265, 0.840203, 0.833537, 0.826868, 0.829228, 0.830723, 0.837707, 0.841715,
  0.844771, 0.844601, 0.837161, 0.834929, 0.834658, 0.827435, 0.816421, 0.808458, 0.798854, 0.795972,
  0.791197, 0.786954, 0.783379, 0.783185, 0.786774, 0.797092, 0.802984, 0.80402, 0.798894, 0.799096,
  0.807688, 0.810853, 0.807936, 0.80386, 0.814181, 0.821945, 0.81554, 0.805242, 0.803238, 0.80179, 0.80666,
  0.809233, 0.801977, 0.798529, 0.800809, 0.802632, 0.803106, 0.805944, 0.803992, 0.810268, 0.818756,
```

200%

```
energiesMatrices = SparseArray /@ALLENERGIES;
```

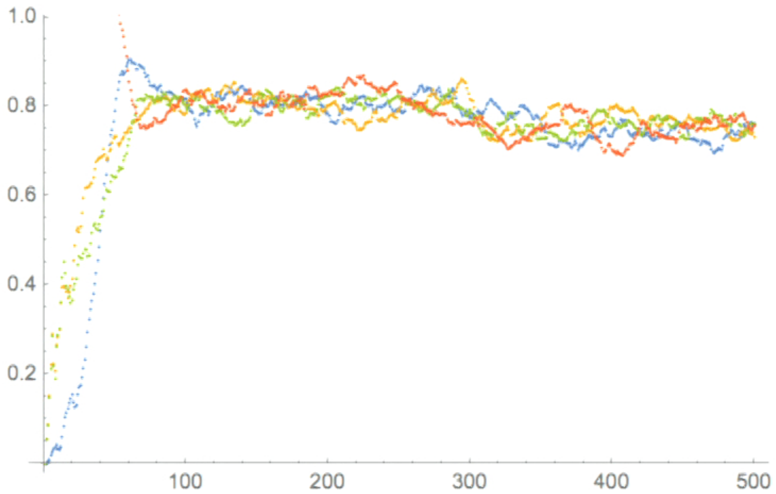
In[150]:=

```
energyConservation = Table[Mean@Mean@(energiesMatrices[[j]]), {j, 1, tmax}] (* == 1/β *) //
   ListPlot[{#, # (1 + 2/L), # (1 - 2/L)}, PlotStyle → Black, Joined → True] &;
```

In[163]:=

```
l = L / 2;
energiesFourCorners =
 Table[Flatten@Table[Mean@Mean@(energiesMatrices[[t, l (j - 1) + 1 ;; l j, l (i - 1) + 1 ;; l i]]),
     {i, 1, L / l}, {j, 1, L / l}], {t, 1, 500}]ᵀ // ListPlot
```

Out[164]=



In[160]:=

```
Show[energiesFourCorners, energyConservation]
```

```
energiesMatrices = SparseArray /@ ALLENERGIES;
```

In[150]:=

```
energyConservation = Table[Mean@Mean@(energiesMatrices[[j]]), {j, 1, tmax}] (* == 1/β *) //
   ListPlot[{#, # (1 + 2/L), # (1 - 2/L)}, PlotStyle → Black, Joined → True] &;
```

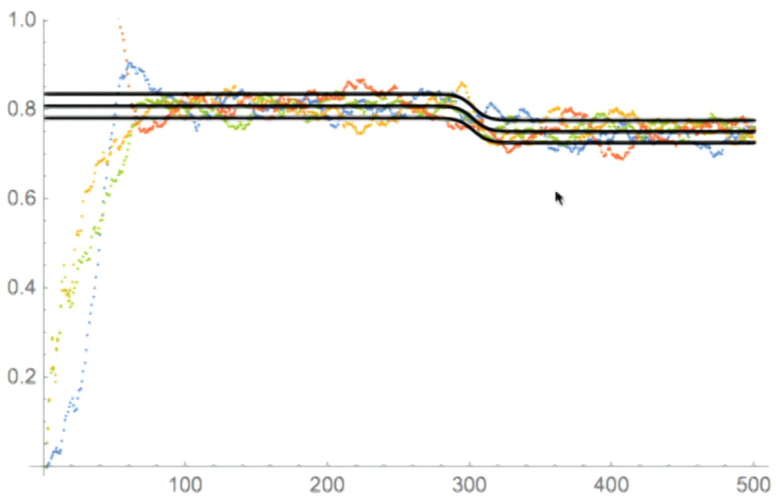
In[165]:=

```
l = L / 2;
energiesFourCorners =
  Table[Flatten@Table[Mean@Mean@(energiesMatrices[[t, l (j - 1) + 1 ;; l j, l (i - 1) + 1 ;; l i]]),
       {i, 1, L / l}, {j, 1, L / l}], {t, 1, 500}]ᵀ // ListPlot;
```

In[167]:=

```
Show[energiesFourCorners, energyConservation]
```

Out[167]=

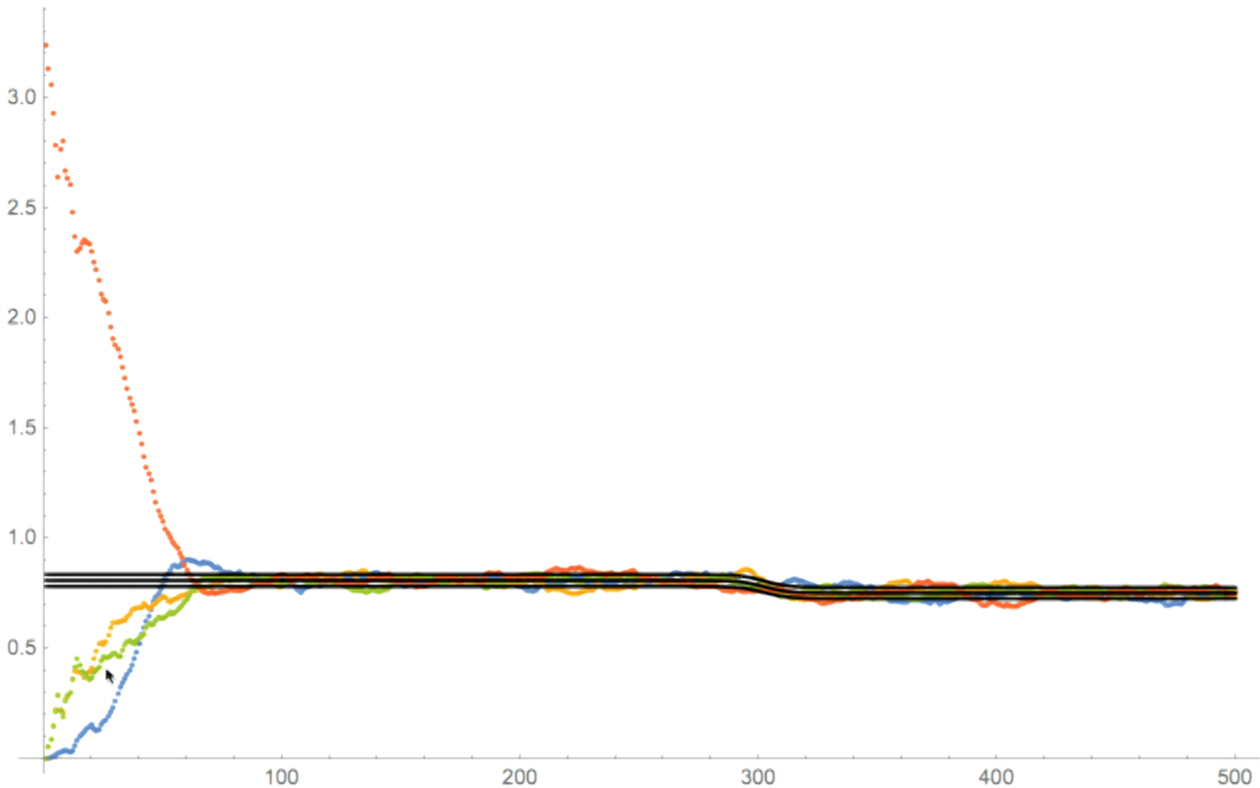


```
l = L / 2;
energiesFourCorners =
  Table[Flatten@Table[Mean@Mean@(energiesMatrices[[t, l (j - 1) + 1 ;; l j, l (i - 1) + 1 ;; l i]]),
      {i, 1, L / l}, {j, 1, L / l}], {t, 1, 500}]ᵀ // ListPlot;
```

In[168]:=

```
Show[energiesFourCorners, energyConservation, PlotRange → All]
```

Out[168]=



Punching the Table. Time Evolution.

```
ClearAll[x]
L = 60; ϰ = 4; ω = 1; tmax = 500; μ = 0 × 5;
x[i_, j_][t_] /; i > L ⋁ i ≤ 0 ⋁ j > L ⋁ j ≤ 0 := x[Mod[i, L, 1], Mod[j, L, 1]][t]

In[80]:= dial[t_] = (1 - Tanh[(t-300)/10])/2;
Plot[dial[t], {t, 0, 500}];

eqs =
  Table[∂{t,2} x[i, j][t] + ω^2 x[i, j][t] + dial[t] μ^2 x[i, j][t]^3 +
     ϰ Sum[x[i, j][t] - x[i + l, j][t] + x[i, j][t] - x[i, j + l][t], {l, {-1, 1}}] == 0, {i, L}, {j, L}] // Flatten;

In[83]:= bc = Table[{x[i, j][0] == 0, x[i, j]'[0] == If[i > 3 L / 4 ⋀ j > 3 L / 4, RandomReal[{4, 6}], 0]}, {i, L}, {j, L}] //
    Flatten;

In[84]:= var = Table[x[i, j][t], {i, L}, {j, L}] // Flatten;

In[123]:=
solution = NDSolve[Join[eqs, bc], var, {t, 0, tmax}][[1]] /. g_[t] ⧴ g;

In[86]:= ToPlot[t_] = var /. x[i_, j_][t] -> {i, j, x[i, j][t]} /. solution;

In[87]:= legs = Graphics3D[{Cylinder[{{1, 1, 0}, {1, 1, -3}}], Cylinder[{{1, L, 0}, {1, L, -3}}],
     Cylinder[{{L, 1, 0}, {L, 1, -3}}], Cylinder[{{L, L, 0}, {L, L, -3}}]}];

Manipulate[ListPointPlot3D[ToPlot[t]], {t, 0, 500, 1}];
```

200%

Saving the evolution for easy manipulation

In[125]:=

```
POSITIONS[t_] = Flatten@Table[{a, b} → x[a, b][t], {a, L}, {b, L}] /. solution;
VELOCITIES[t_] = Flatten@Table[{a, b} → x[a, b]'[t], {a, L}, {b, L}] /. solution;
ENERGIES[t_] =
  Flatten@
   Table[{a, b} → 1/2 x[a, b]'[t]^2 + ω^2 x[a, b][t]^2/2 + dial[t] μ^2/4 x[a, b][t]^4 +
      κ/4 Sum[(x[a, b][t] - x[a + l, b][t])^2 + (x[a, b][t] - x[a, b + l][t])^2, {l, {-1, 1}}], {a, L}, {b, L}] /.
   solution;
```

In[128]:=

```
ALLPOSITIONS = Table[POSITIONS[t], {t, 0, tmax, 1}];
```

In[129]:=

```
ALLVELOCITIES = Table[VELOCITIES[t], {t, 0, tmax, 1}];
```

In[130]:=

```
ALLENERGIES = Monitor[Table[ENERGIES[t], {t, 0, tmax, 1}], ProgressIndicator[t, {0, tmax}]];
```

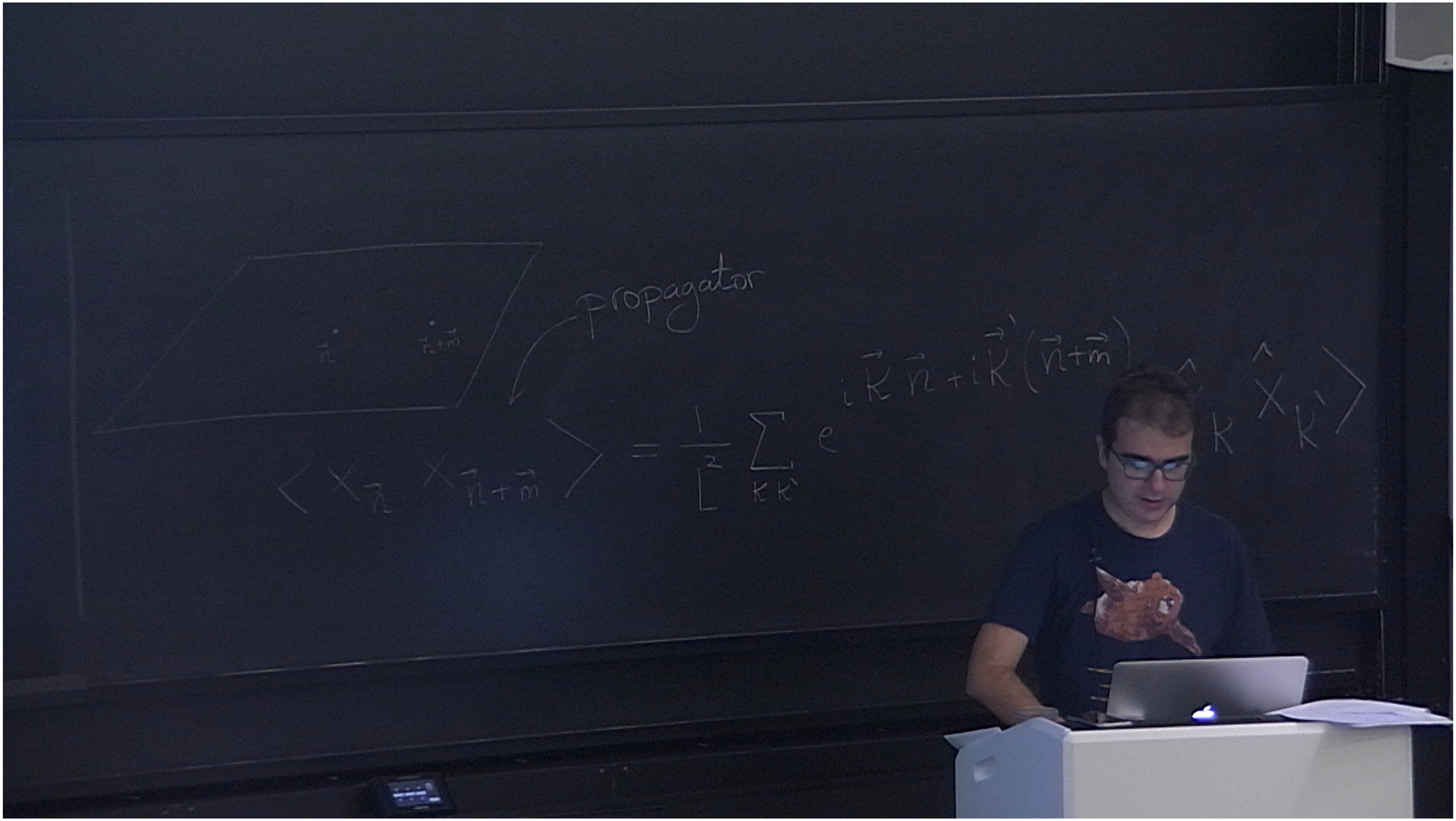
In[131]:=

```
ALLPOSITIONS >> (NotebookDirectory[] <> "Data/positions.txt");
ALLVELOCITIES >> (NotebookDirectory[] <> "Data/velocities.txt");
ALLENERGIES >> (NotebookDirectory[] <> "Data/energies.txt");
```

Plots

200%


propagator
$\vec{n}$
$\vec{n}+\vec{m}$
$\langle X_{\vec{n}} X_{\vec{n}+\vec{m}} \rangle$


propagator
$\langle X_{\vec{n}} X_{\vec{n}+\vec{m}} \rangle = \frac{1}{L^2} \sum_{k k'} e^{i\vec{k}\vec{n} + i\vec{k}'(\vec{n}+\vec{m})} \langle \hat{X}_{k} \hat{X}_{k'} \rangle$

In[181]:=

```
energyConservation = Table[Mean@Mean@(energiesMatrices[[j]]), {j, 1, tmax}] (* == 1/β *) //
   ListPlot[{#, # (1 + 2/L), # (1 - 2/L)}, PlotStyle → Black, Joined → True] &;
```

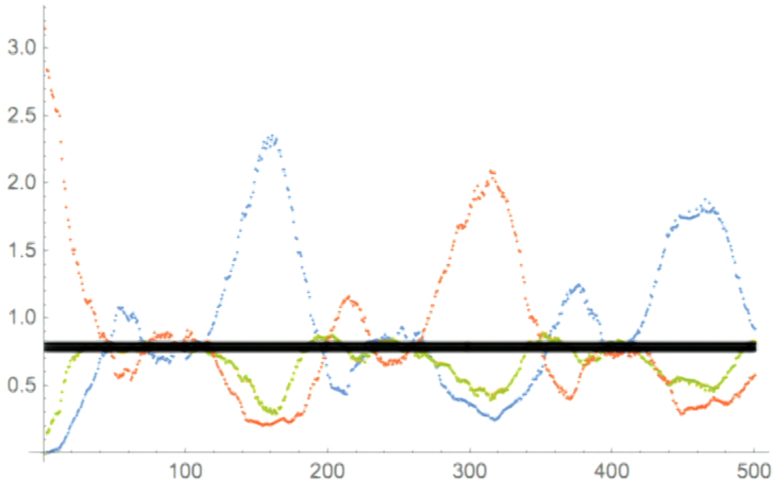
In[182]:=

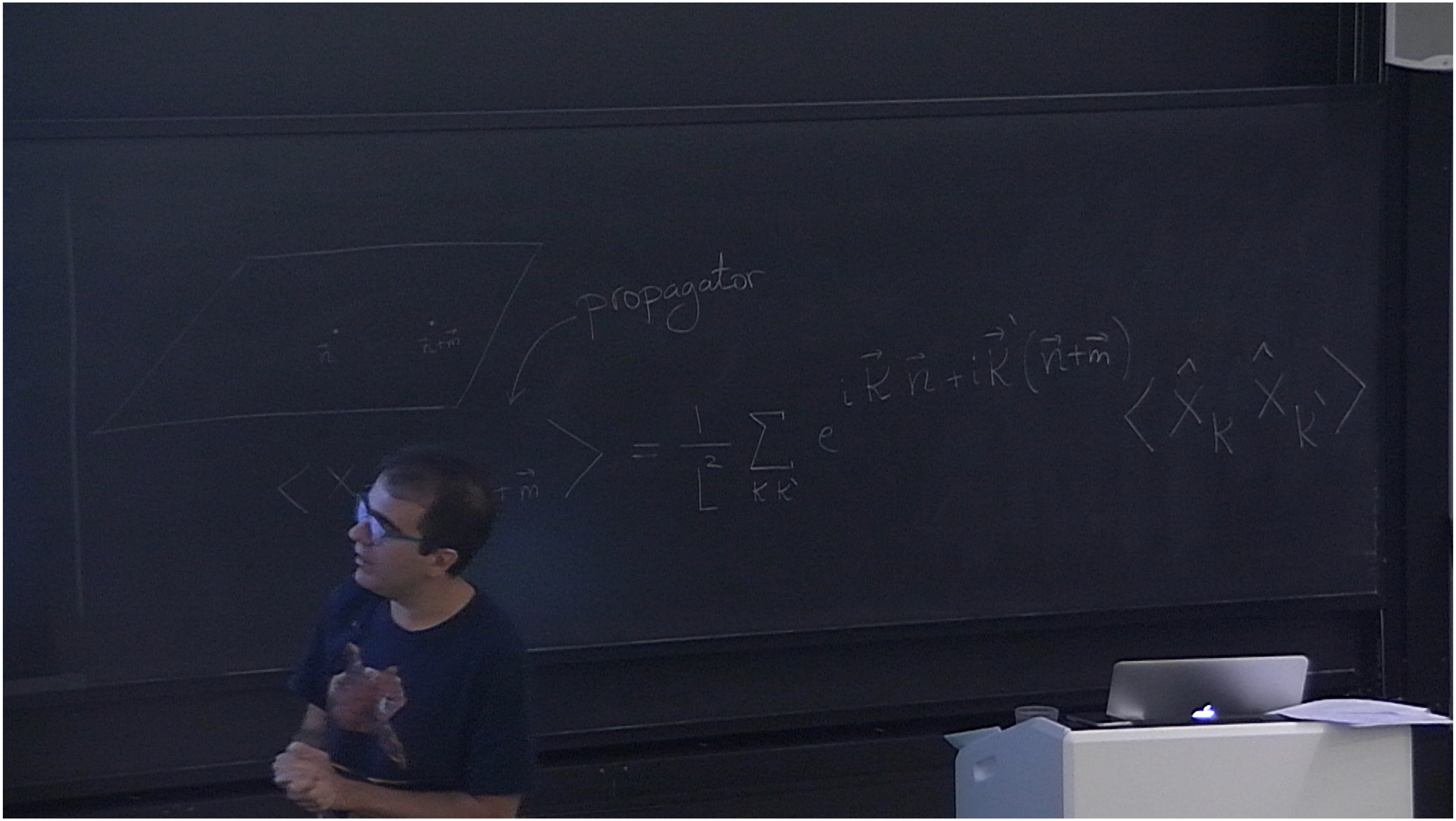
```
l = L / 2;
energiesFourCorners =
  Table[Flatten@Table[Mean@Mean@(energiesMatrices[[t, l (j - 1) + 1 ;; l j, l (i - 1) + 1 ;; l i]]),
      {i, 1, L / l}, {j, 1, L / l}], {t, 1, 500}]ᵀ // ListPlot;
```

In[184]:=

```
Show[energiesFourCorners, energyConservation, PlotRange → All]
```

Out[184]=

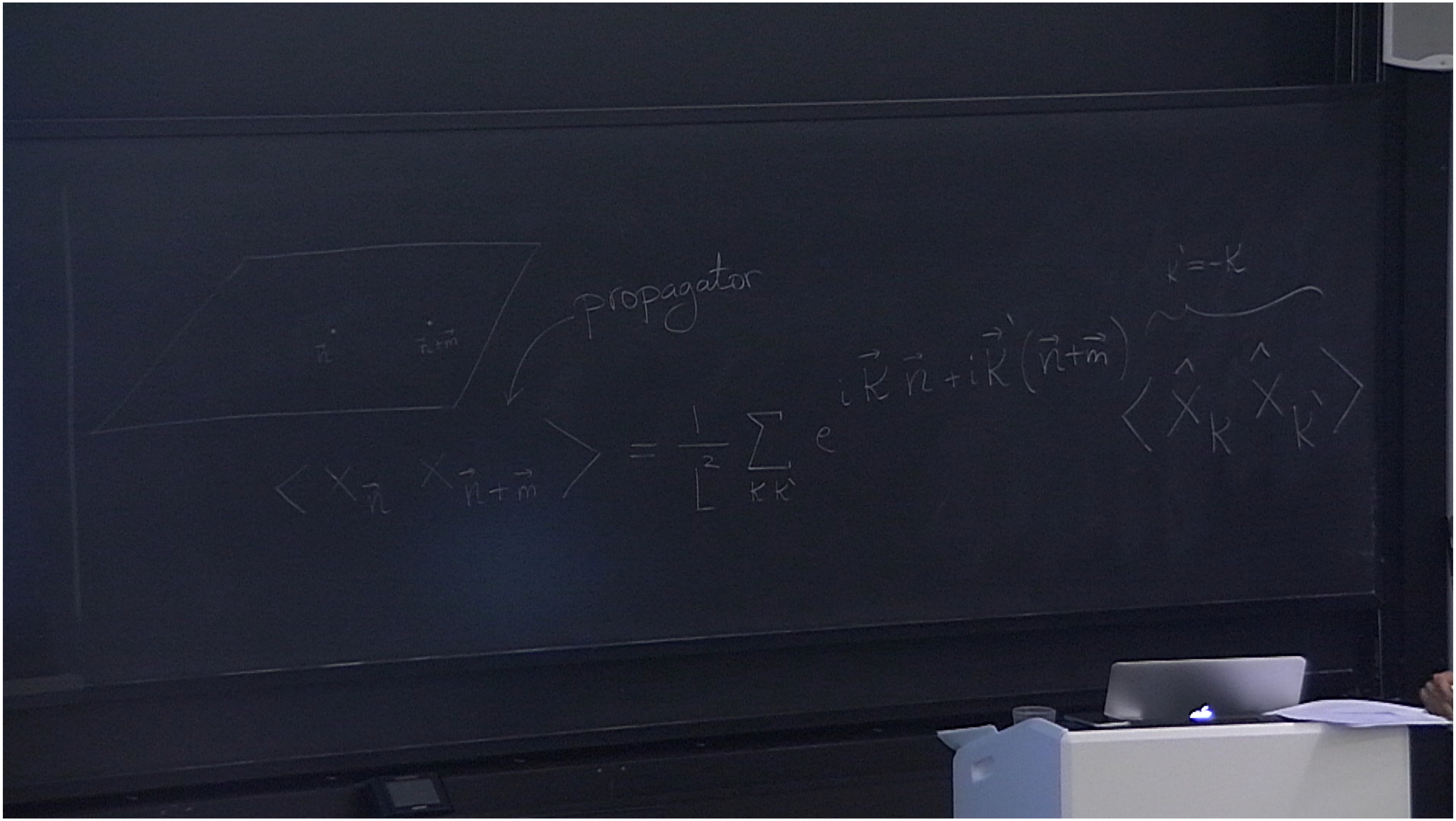


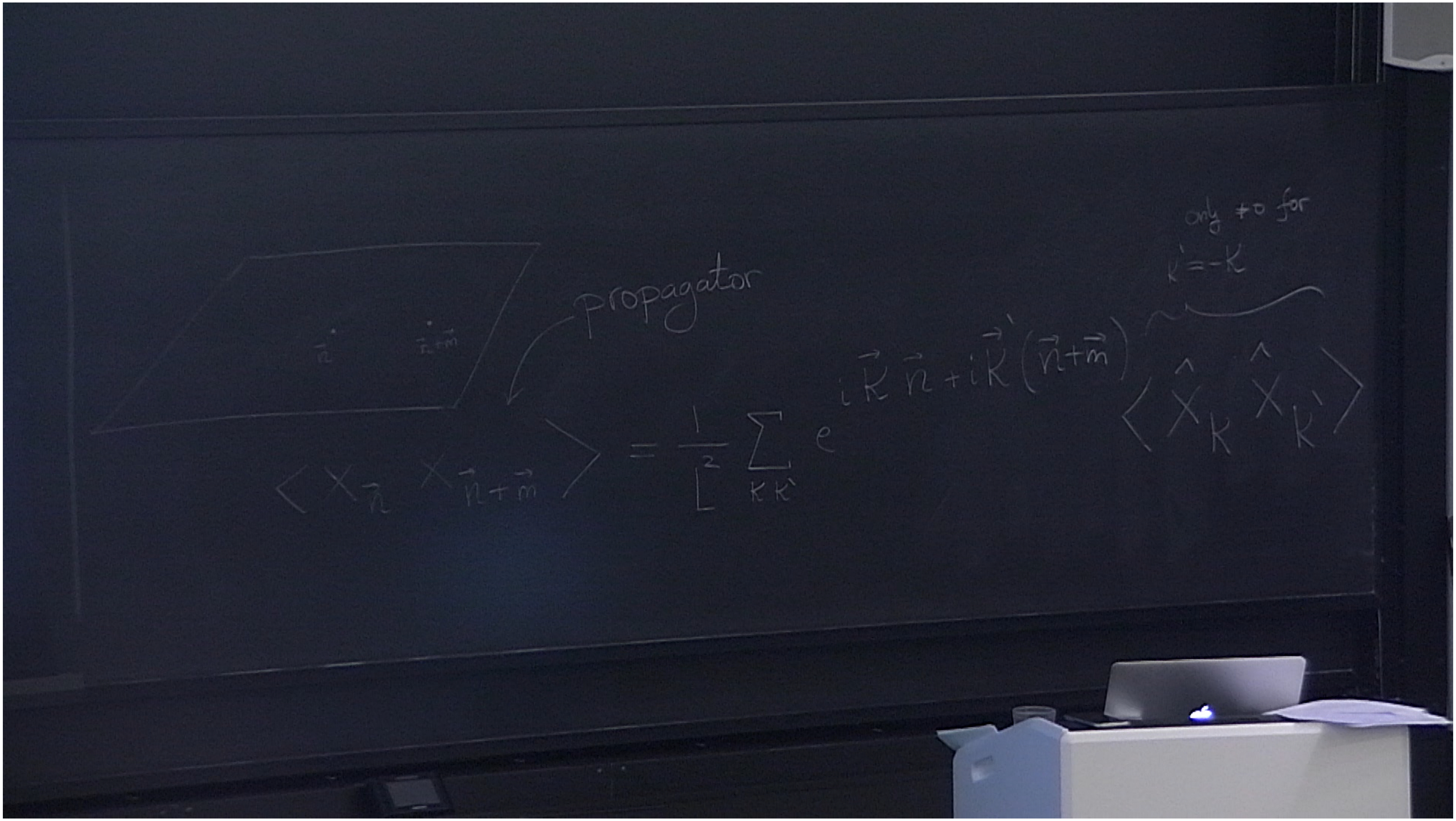


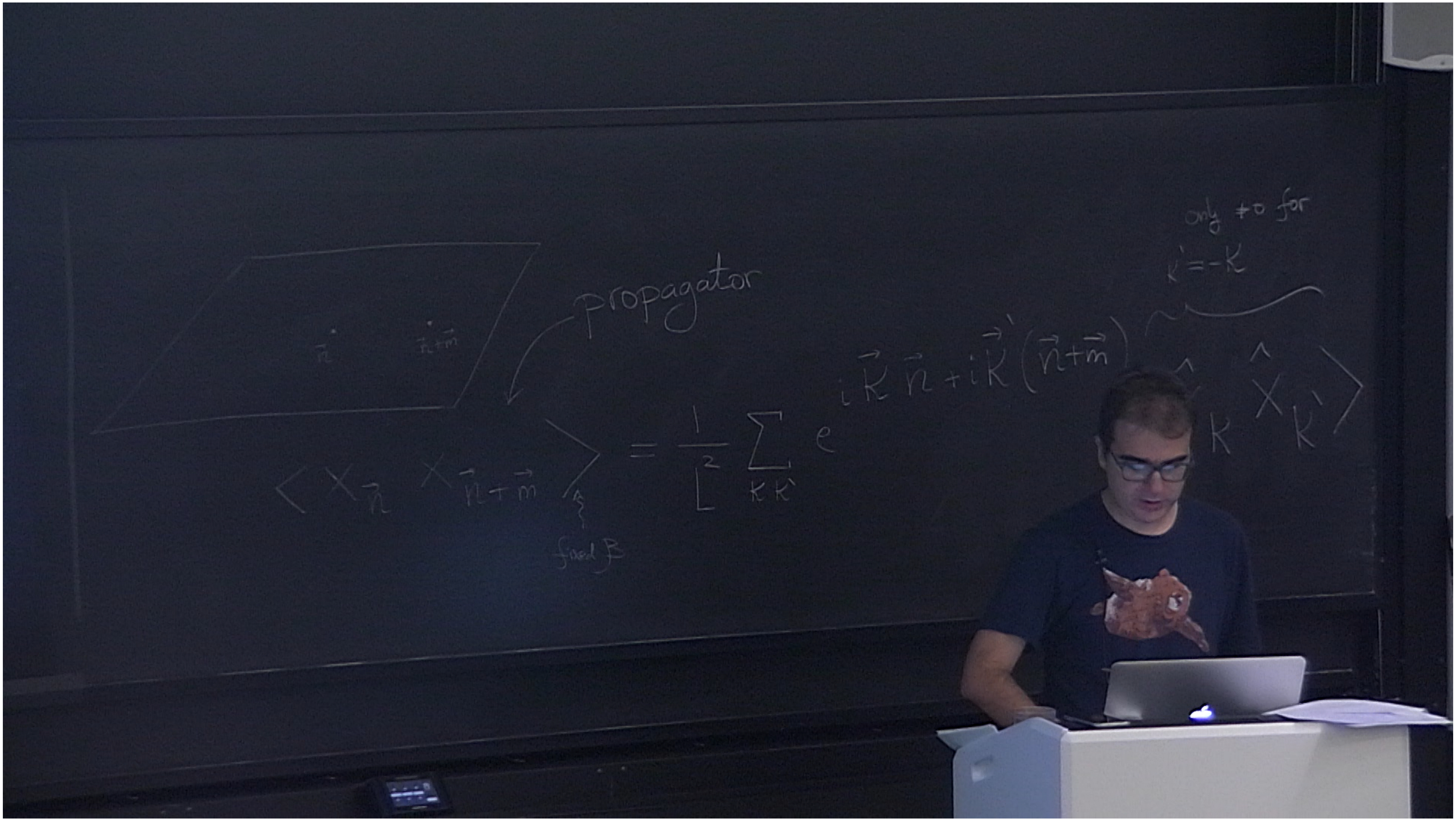
propagator

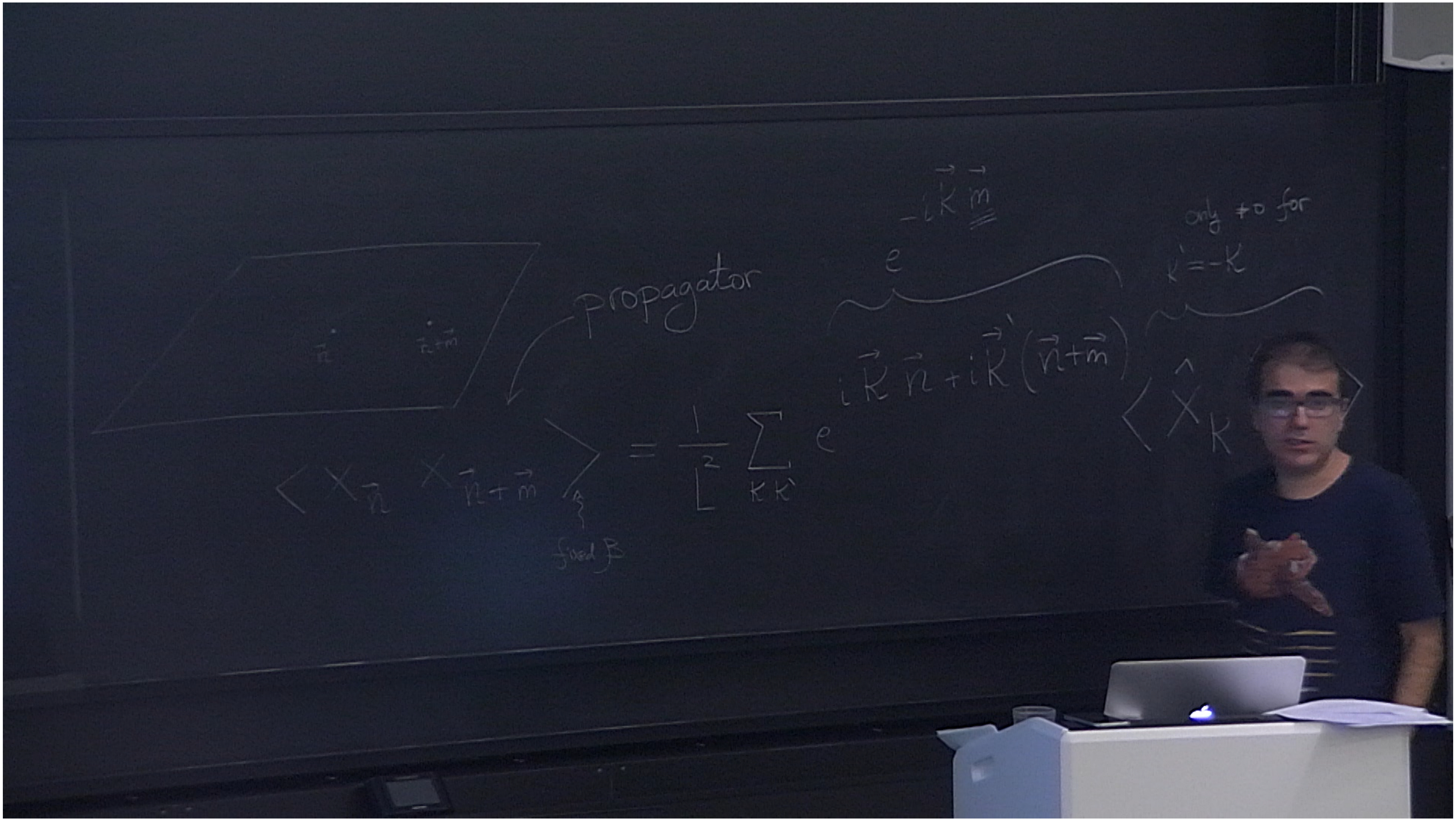
Punching the Table. Time Evolution.

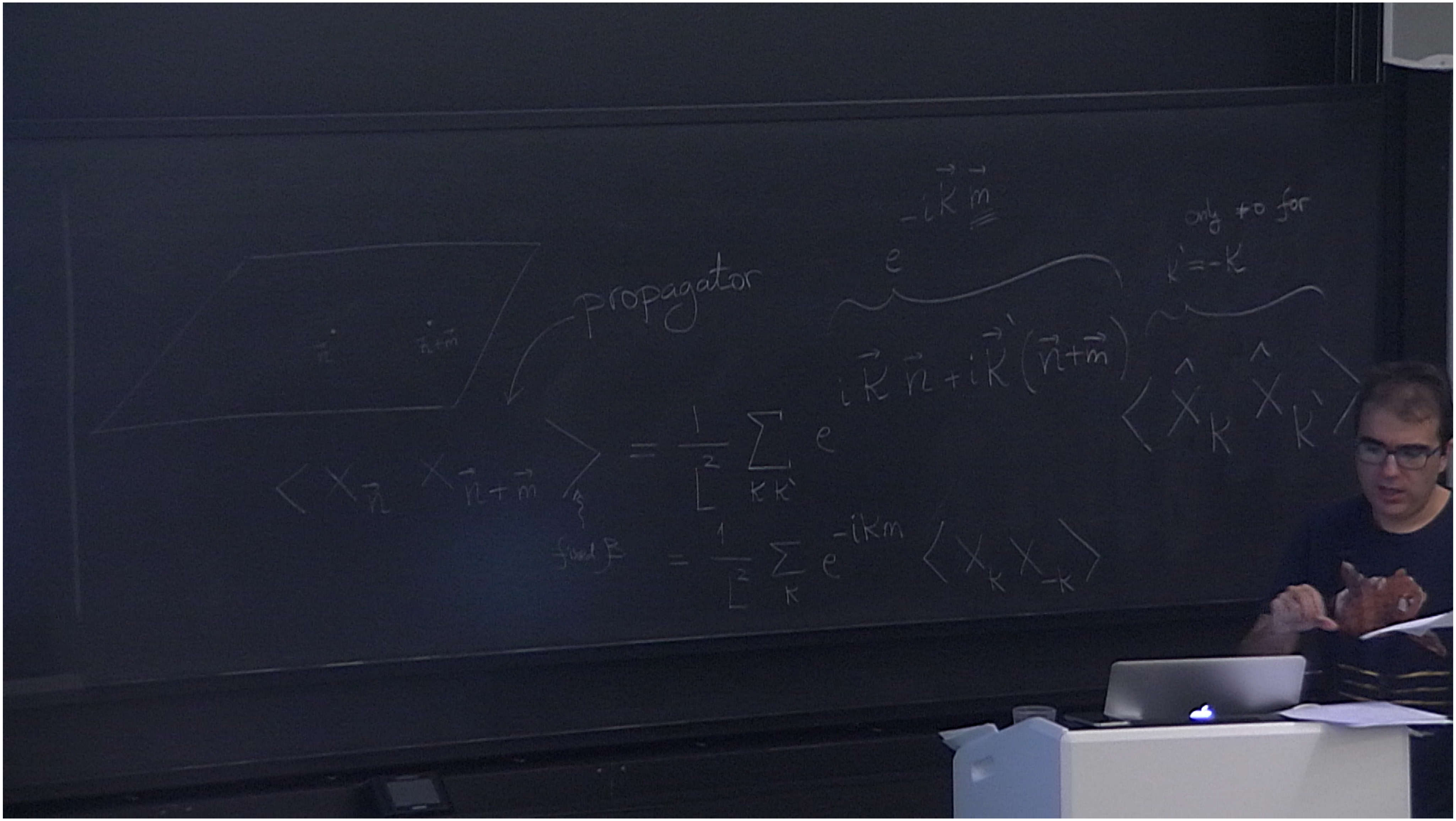
```
ClearAll[x]
L = 60; κ = 4; ω = 1; tmax = 500; μ = 5;
x[i_, j_][t_] /; i > L ⋁ i ≤ 0 ⋁ j > L ⋁ j ≤ 0 := x[Mod[i, L, 1], Mod[j, L, 1]][t]

In[172]:=
dial[t_] = (1 - Tanh[(t-300)/10]) / 2;
Plot[dial[t], {t, 0, 500}];

In[174]:=
eqs =
  Table[∂{t,2} x[i, j][t] + ω^2 x[i, j][t] + dial[t] μ^2 x[i, j][t]^3 +
     κ Sum[x[i, j][t] - x[i + l, j][t] + x[i, j][t] - x[i, j + l][t], {l, {-1, 1}}] == 0, {i, L}, {j, L}] // Flatten;

In[175]:=
bc = Table[{x[i, j][0] == 0, x[i, j]'[0] == If[i > 3 L / 4 ⋀ j > 3 L / 4, RandomReal[{4, 6}], 0]}, {i, L}, {j, L}] //
   Flatten;

In[176]:=
var = Table[x[i, j][t], {i, L}, {j, L}] // Flatten;

In[177]:=
solution = NDSolve[Join[eqs, bc], var, {t, 0, tmax}][[1]] /. g_[t] :→ g;

In[86]:= ToPlot[t_] = var /. x[i_, j_][t] -> {i, j, x[i, j][t]} /. solution;

In[87]:= legs = Graphics3D[{Cylinder[{{1, 1, 0}, {1, 1, -3}}], Cylinder[{{1, L, 0}, {1, L, -3}}],
```

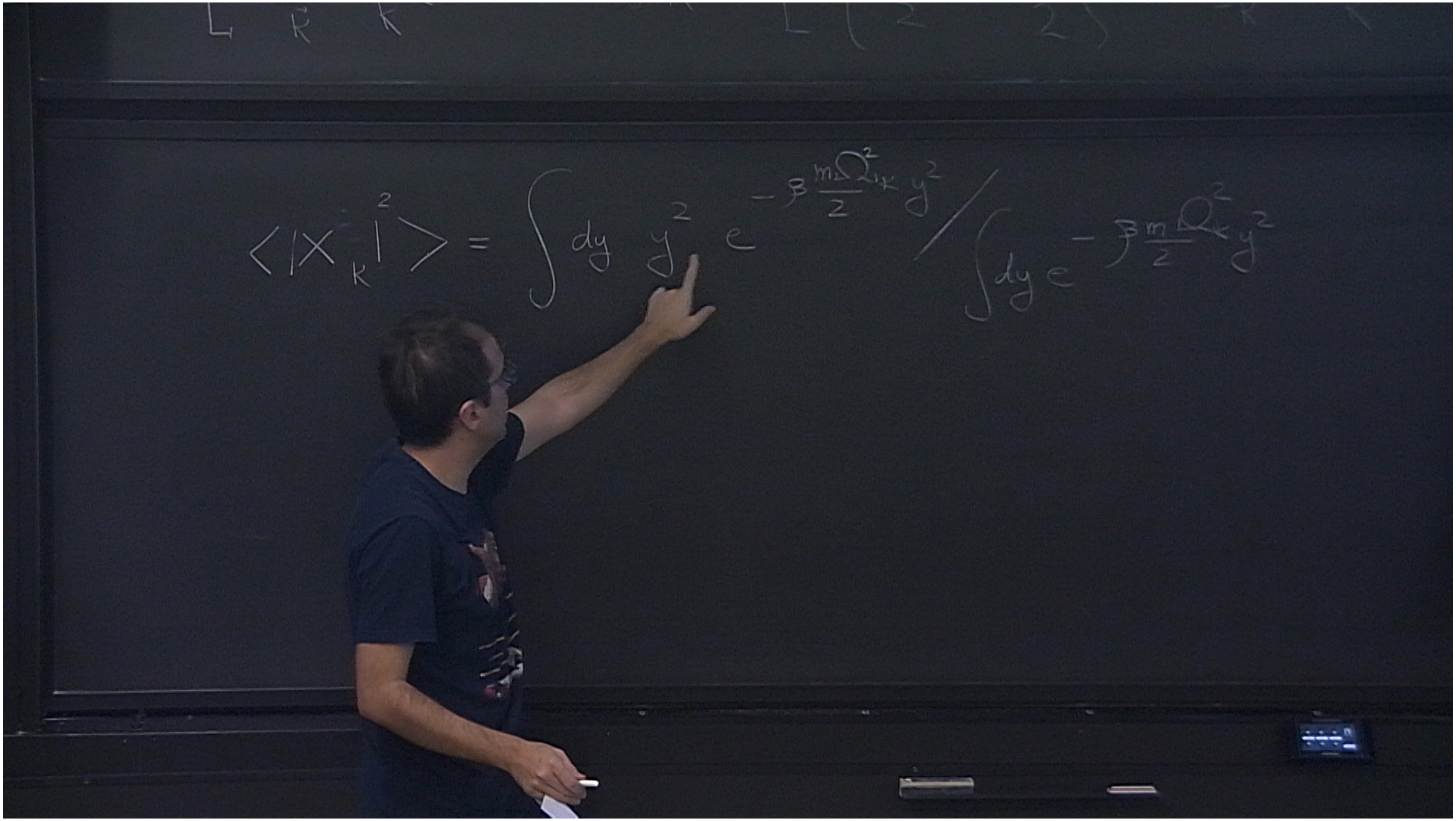

propagator
k' = -k
$\langle x_{\vec{n}} \, x_{\vec{n}+\vec{m}} \rangle = \frac{1}{L^2} \sum_{k k'} e^{i\vec{k}\vec{n} + i\vec{k}'(\vec{n}+\vec{m})} \langle \hat{x}_k \hat{x}_{k'} \rangle$

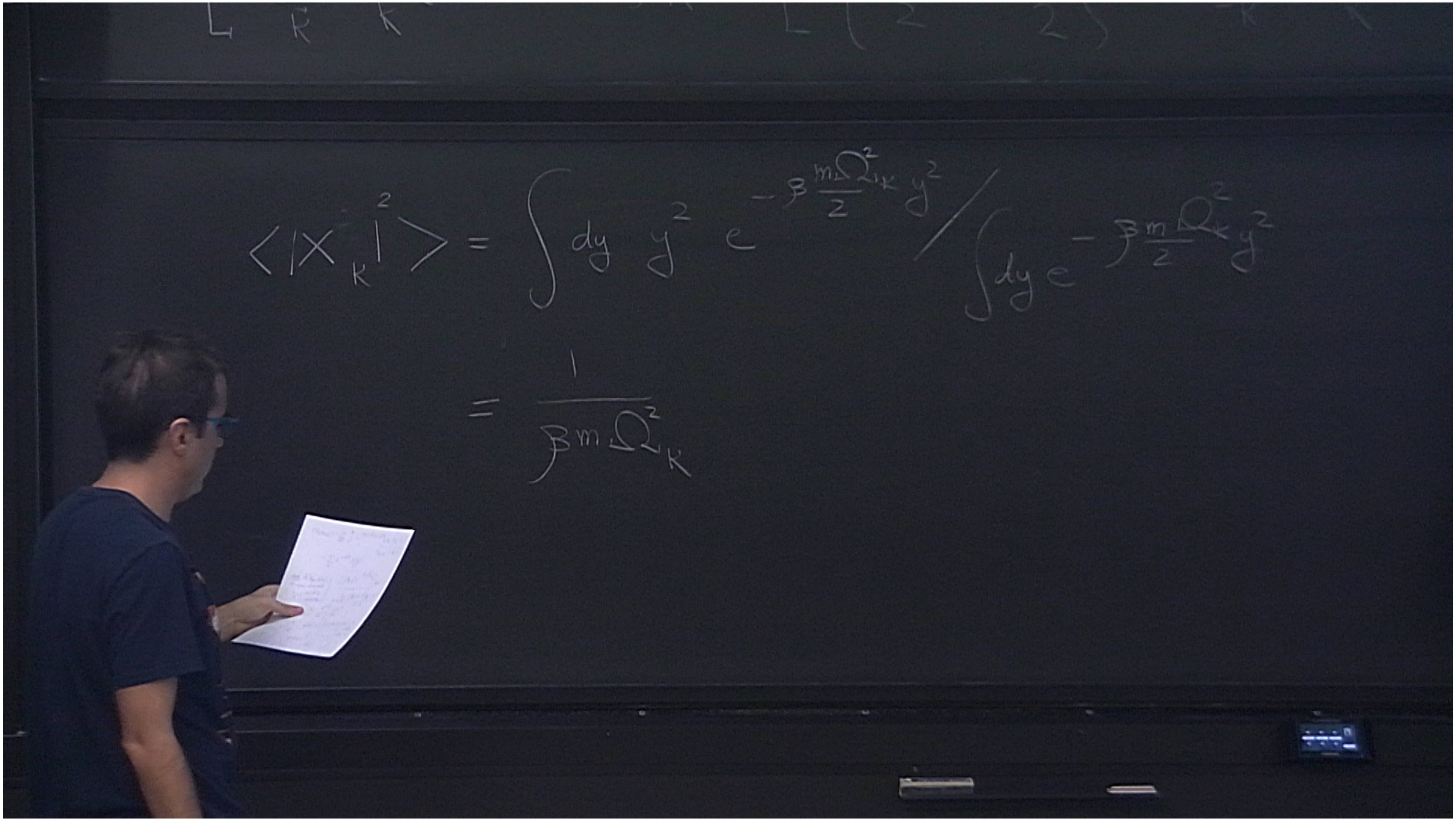

propagator
only ≠0 for
$\vec{k}' = -\vec{k}$
$\vec{n}$
$\vec{n}+\vec{m}$
$\left\langle X_{\vec{n}} \, X_{\vec{n}+\vec{m}} \right\rangle = \frac{1}{L^2} \sum_{k k'} e^{i\vec{k}\vec{n} + i\vec{k}'(\vec{n}+\vec{m})} \left\langle \hat{X}_{k} \, \hat{X}_{k'} \right\rangle$


propagator
only ≠0 for
fixed β

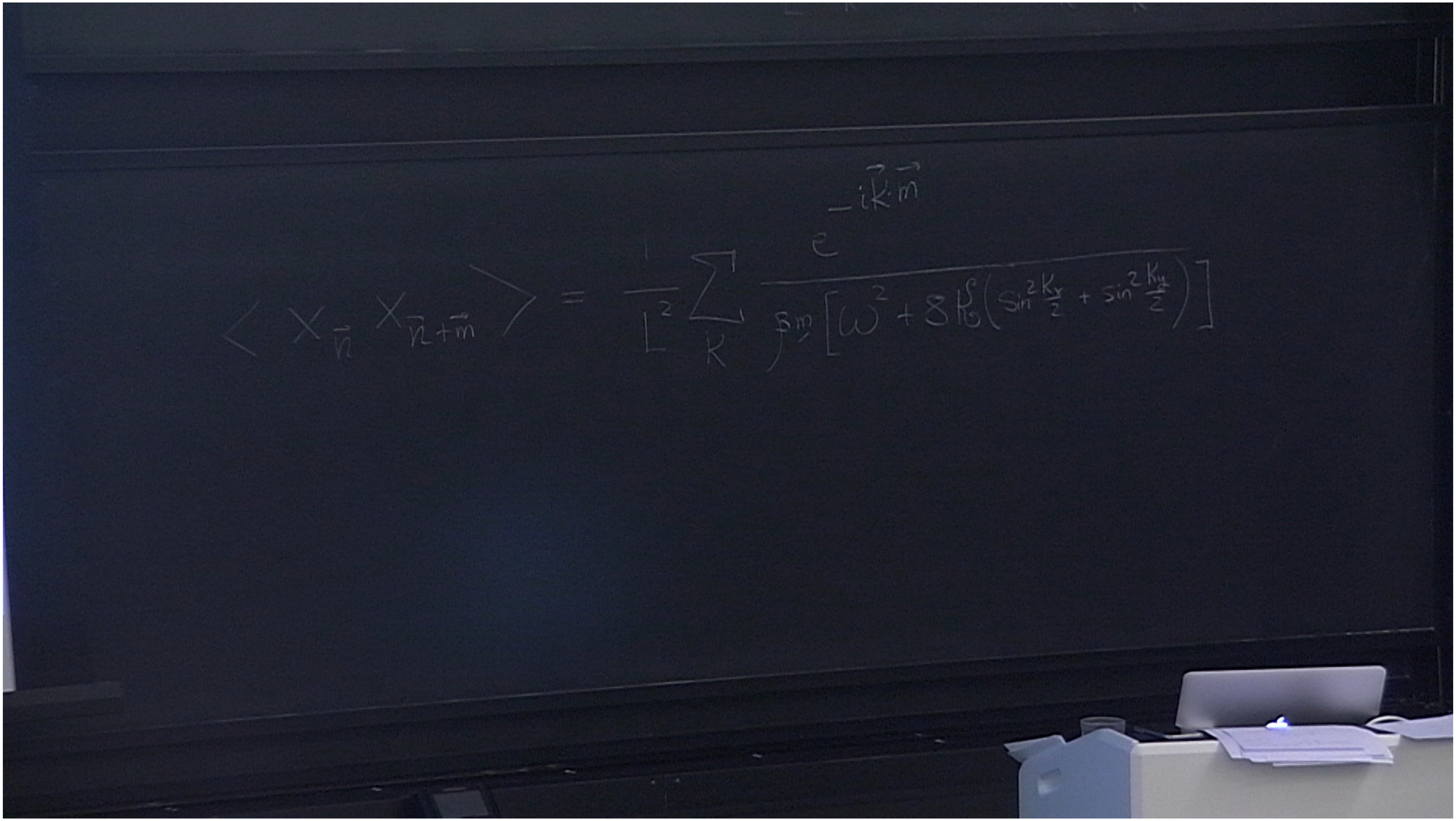

propagator
fixed β
only ≠ 0 for
k' = -k


propagator
fixed k
only ≠ 0 for
k' = -k


$\langle |X_k|^2 \rangle = \int dy \, y^2 e^{-\beta \frac{m\Omega_k^2}{2} y^2} \Big/ \int dy \, e^{-\beta \frac{m\Omega_k^2}{2} y^2}$


$\langle |X_k|^2 \rangle = \int dy\, y^2\, e^{-\beta \frac{m \Omega_k^2}{2} y^2} \Big/ \int dy\, e^{-\beta \frac{m \Omega_k^2}{2} y^2}$
$= \frac{1}{\beta m \Omega_k^2}$

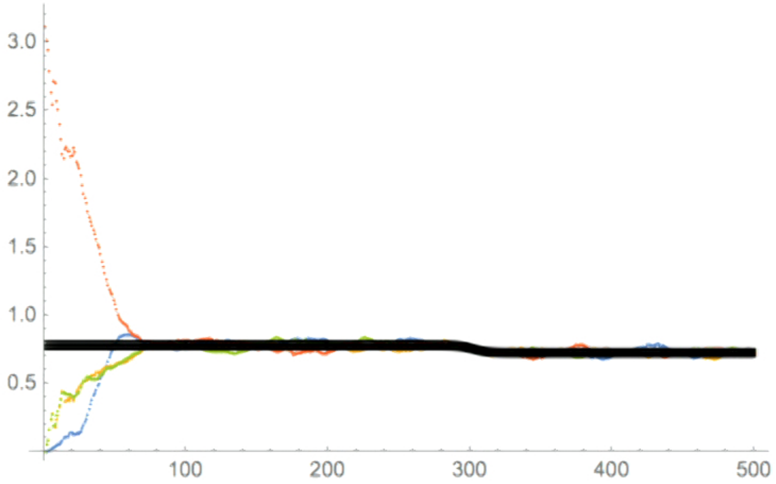
$$\langle X_{\vec{n}} X_{\vec{n}+\vec{m}} \rangle = \frac{1}{L^2} \sum_{k} \frac{e^{-i\vec{k}m}}{\beta m \left[\omega^2 + 8 k_0 \left(\sin^2 \frac{k_x}{2} + \sin^2 \frac{k_y}{2} \right) \right]}$$


$\langle X_{\vec{n}} X_{\vec{n}+\vec{m}} \rangle = \frac{1}{L^2} \sum_{K} \frac{e^{-i\vec{K}\cdot\vec{m}}}{\rho m \left[\omega^2 + 8 K_s^{\rho}\left(\sin^2\frac{K_x}{2} + \sin^2\frac{K_y}{2}\right)\right]}$

In[204]:=

```
Show[energiesFourCorners, energyConservation, PlotRange → All]
```

Out[204]=



Propagator

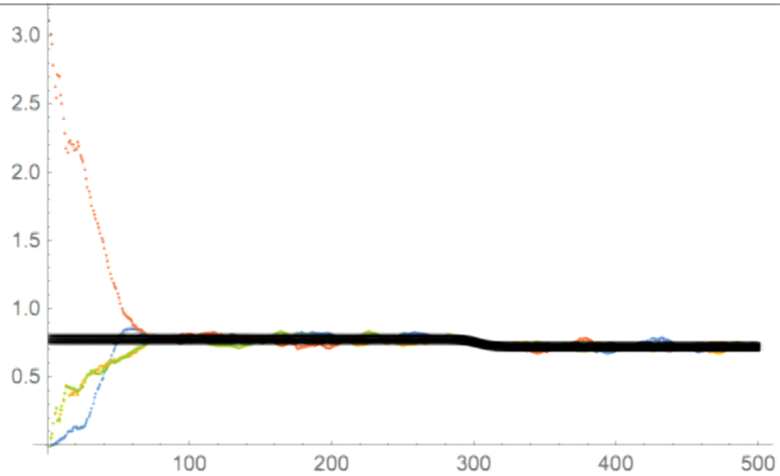
In[205]:=

```
ALLPOSITIONS[[500]]
```

Out[205]=

```
{{1, 1} → -0.0528715, {1, 2} → 0.305811, {1, 3} → 0.332093,
 ... 3594 ..., {60, 58} → -0.0778283, {60, 59} → -0.322094, {60, 60} → 0.149499}
```

large output | show less | show more | show all | set size limit...



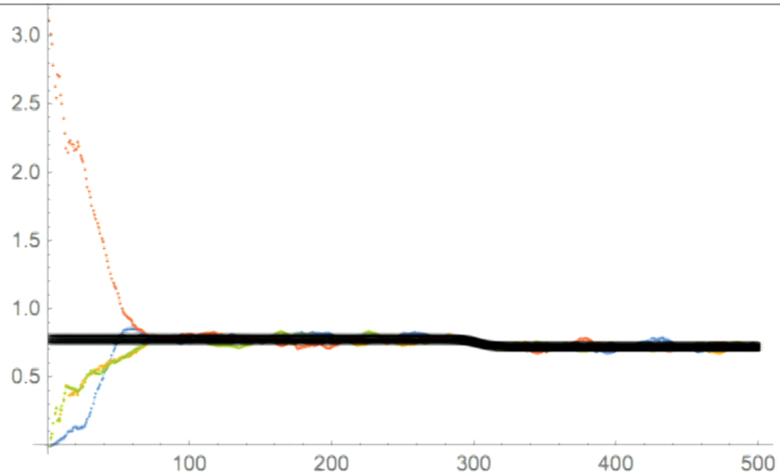
Propagator

```
{i, j} /.ALLPOSITIONS[[500]]
```

Out[205]=

{{1, 1} → -0.0528715, {1, 2} → 0.305811, {1, 3} → 0.332093,
, {60, 58} → -0.0778283, {60, 59} → -0.322094, {60, 60} → 0.149499}

large output   show less   show more   show all   set size limit...



Propagator

```
({i, j} /. ALLPOSITIONS[[500]]) ({i + nx, j + ny} /. ALLPOSITIONS[[500]])
```

Out[205]=

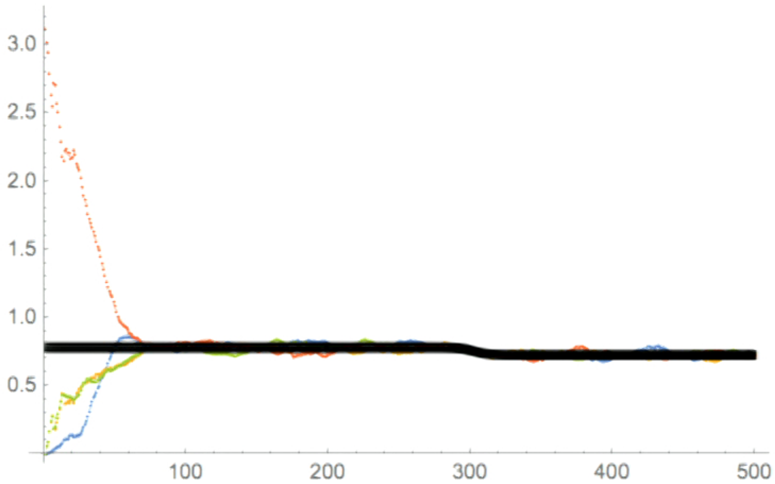
```
{{1, 1} → -0.0528715, {1, 2} → 0.305811, {1, 3} → 0.332093,
 , {60, 58} → -0.0778283, {60, 59} → -0.322094, {60, 60} → 0.149499}
```

large output   show less   show more   show all   set size limit...

In[204]:=

```
Show[energiesFourCorners, energyConservation, PlotRange → All]
```

Out[204]=




```
Mean@Mean@(energiesMatrices[[-50]])
```

Propagator

```
({i, j} /. ALLPOSITIONS[[500]]) ({i + nx, j + ny} /. ALLPOSITIONS[[500]])
```

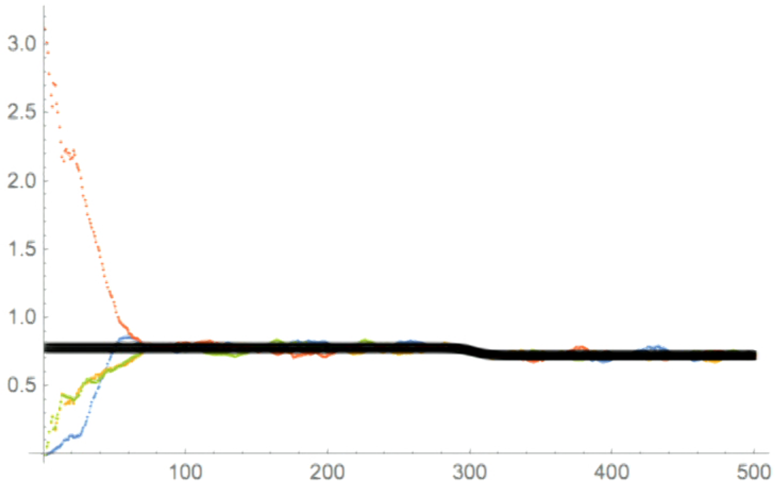
Out[205]=

```
{{1, 1} → -0.0528715, {1, 2} → 0.305811, {1, 3} → 0.332093,
, {60, 58} → -0.0778283, {60, 59} → -0.322094, {60, 60} → 0.149499}
```

In[204]:=

```
Show[energiesFourCorners, energyConservation, PlotRange → All]
```

Out[204]=



In[206]:=

```
Mean@Mean@(energiesMatrices[[-1]])
```

Out[206]=

```
0.72462
```

Propagator

```
({i, j} /. ALLPOSITIONS[[500]]) ({i + nx, j + ny} /. ALLPOSITIONS[[500]])
```

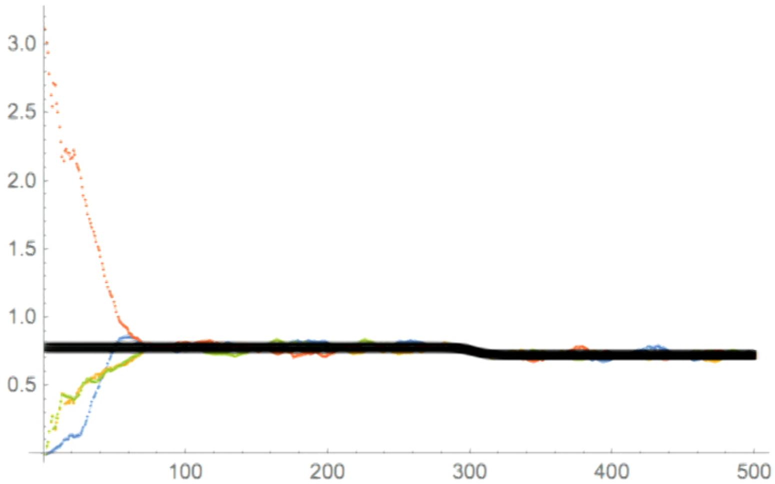
Out[205]=

```
{{1, 1} → -0.0528715, {1, 2} → 0.305811, {1, 3} → 0.332093,
```

In[204]:=

```
Show[energiesFourCorners, energyConservation, PlotRange → All]
```

Out[204]=




```
βeff = 1 / (Mean@Mean@(energiesMatrices[[-1]]))
```

Out[206]=

```
0.72462
```

Propagator

```
({i, j} /. ALLPOSITIONS[[500]]) ({i + nx, j + ny} /. ALLPOSITIONS[[500]])
```

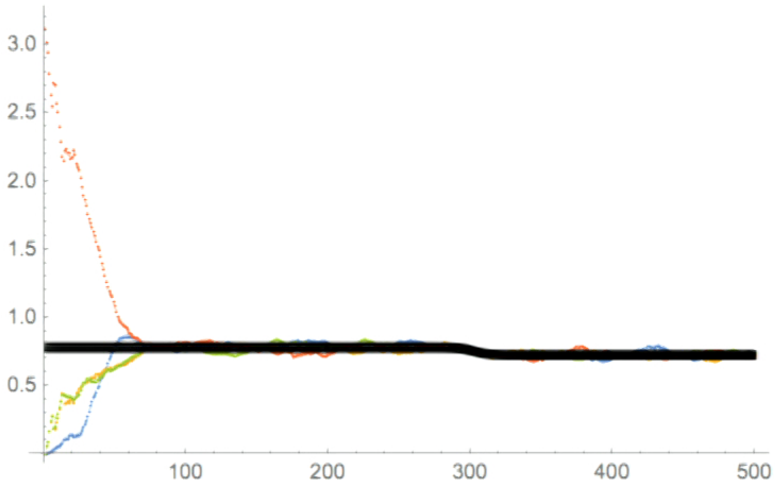
Out[205]=

```
{{1, 1} → -0.0528715, {1, 2} → 0.305811, {1, 3} → 0.332093,
```

In[204]:=

```
Show[energiesFourCorners, energyConservation, PlotRange → All]
```

Out[204]=



In[207]:=

```
βeff = 1 / (Mean@Mean@(energiesMatrices[[-1]]))
```

Out[207]=

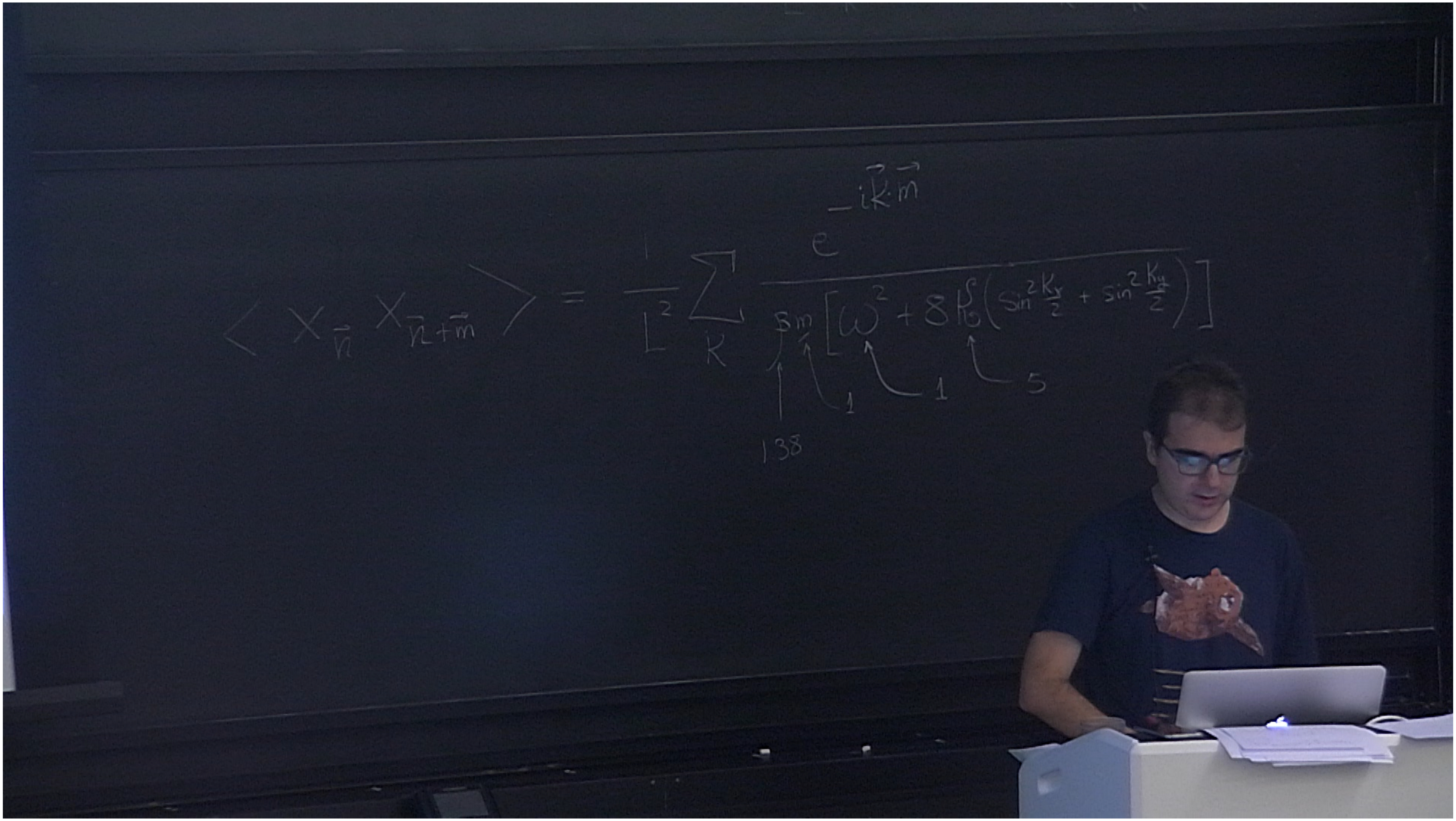
```
1.38003
```

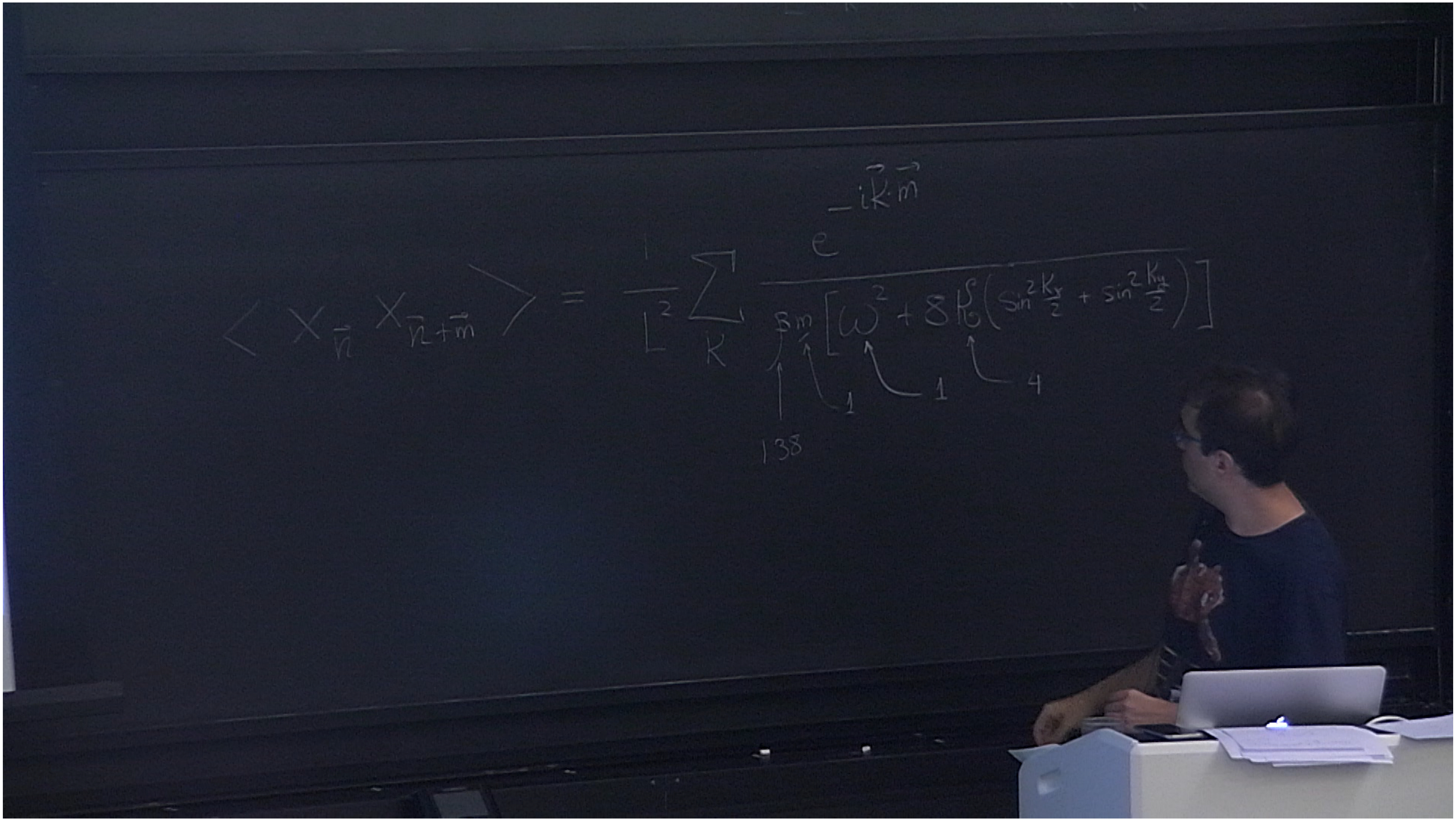
Propagator

```
({i, j} /. ALLPOSITIONS[[500]]) ({i + nx, j + ny} /. ALLPOSITIONS[[500]])
```

Out[205]=

```
{{1, 1} → -0.0528715, {1, 2} → 0.305811, {1, 3} → 0.332093,
```


1
1
5
138


$\langle x_{\vec{n}} x_{\vec{n}+\vec{m}} \rangle = \frac{1}{L^2} \sum_{K} \frac{e^{-i\vec{K}\cdot\vec{m}}}{8m\left[\omega^2 + 8k_0\left(\sin^2\frac{K_x}{2} + \sin^2\frac{K_y}{2}\right)\right]}$
138
1
1
4

In[207]:=

```
βeff = 1 / (Mean@Mean@(energiesMatrices[[-1]]))
```

Out[207]=

```
1.38003
```

Propagator

```
Table[
 ({i, j} /. ALLPOSITIONS[[500]]) ({Mod[i + nx, L, 1], Mod[j + ny, L, 1]} /. ALLPOSITIONS[[500]])
 , {i, L}, {j, L}, {nx, -L / 6, L / 6}, {xy
```

Out[205]=

```
{{1, 1} → -0.0528715, {1, 2} → 0.305811, {1, 3} → 0.332093,
 , {60, 58} → -0.0778283, {60, 59} → -0.322094, {60, 60} → 0.149499}
```

large output   show less   show more   show all   set size limit...

200%

In[207]:=

```
βeff = 1 / (Mean@Mean@(energiesMatrices[[-1]]))
```

Out[207]=

```
1.38003
```

Propagator

```
Table[
 ({i, j} /. ALLPOSITIONS[[500]]) ({Mod[i + nx, L, 1], Mod[j + ny, L, 1]} /. ALLPOSITIONS[[500]])
 , {i, L}, {j, L}, {nx, -L / 6, L / 6}, {ny, -L / 6, L / 6}]
```

Out[205]=

```
{{1, 1} → -0.0528715, {1, 2} → 0.305811, {1, 3} → 0.332093,
 , {60, 58} → -0.0778283, {60, 59} → -0.322094, {60, 60} → 0.149499}
```

large output   show less   show more   show all   set size limit...

200%

In[207]:=

```
βeff = 1 / (Mean@Mean@(energiesMatrices[[-1]]))
```

Out[207]=

```
1.38003
```

Propagator

```
Monitor[Table[
  ({i, j} /. ALLPOSITIONS[[500]]) ({Mod[i + nx, L, 1], Mod[j + ny, L, 1]} /. ALLPOSITIONS[[500]])
  , {i, L}, {j, L}, {nx, -L / 6, L / 6}, {ny, -L / 6, L / 6}], {i, j}]
```

Out[205]=

```
{{1, 1} → -0.0528715, {1, 2} → 0.305811, {1, 3} → 0.332093,
 ..., {60, 58} → -0.0778283, {60, 59} → -0.322094, {60, 60} → 0.149499}
```

large output   show less   show more   show all   set size limit...

200%

In[207]:=

```
βeff = 1 / (Mean@Mean@(energiesMatrices[[-1]]))
```

Out[207]=

```
1.38003
```

Propagator

In[208]:=

```
end = Monitor[Table[
   ({i, j} /. ALLPOSITIONS[[-1]]) ({Mod[i + nx, L, 1], Mod[j + ny, L, 1]} /. ALLPOSITIONS[[500]])
   , {i, L}, {j, L}, {nx, -L / 6, L / 6}, {ny, -L / 6, L / 6}], {i, j}]
```

```
{1, 22}
```

200%

In[207]:=

```
βeff = 1 / (Mean@Mean@(energiesMatrices[[-1]]))
```

Out[207]=

```
1.38003
```

Propagator

In[208]:=

```
end = Monitor[Table[
  ({i, j} /. ALLPOSITIONS[[-1]]) ({Mod[i + nx, L, 1], Mod[j + ny, L, 1]} /. ALLPOSITIONS[[500]])
  , {i, L}, {j, L}, {nx, -L / 6, L / 6}, {ny, -L / 6, L / 6}], {i, j}]
```

Out[208]=

```
$Aborted
```

```
Mean@Mean@end
```

In[207]:=

```
βeff = 1 / (Mean@Mean@(energiesMatrices[[-1]]))
```

Out[207]=

```
1.38003
```

Propagator

In[209]:=

```
end = Monitor[Table[
   {nx, ny, ({i, j} /. ALLPOSITIONS[[-1]]) ({Mod[i + nx, L, 1], Mod[j + ny, L, 1]} /. ALLPOSITIONS[[500]])}
   , {i, L}, {j, L}, {nx, -L / 6, L / 6}, {ny, -L / 6, L / 6}], {i, j}]
```

```
{1, 28}
```

```
Mean@Mean@end
```

200%

In[207]:=

```
βeff = 1 / (Mean@Mean@(energiesMatrices[[-1]]))
```

Out[207]=

```
1.38003
```

Propagator

In[210]:=

```
end = Monitor[Table[
   {nx, ny, ({i, j} /. ALLPOSITIONS[[-1]]) ({Mod[i + nx, L, 1], Mod[j + ny, L, 1]} /. ALLPOSITIONS[[-1]])}
   , {i, L}, {j, L}, {nx, -L / 6, L / 6}, {ny, -L / 6, L / 6}], {i, j}]
```

```
{5, 45}
```

```
Mean@Mean@end // Flatten[#, 1] &
```

200%

In[207]:=

```
βeff = 1 / (Mean@Mean@(energiesMatrices[[-1]]))
```

Out[207]=

```
1.38003
```

Propagator

In[210]:=

```
end = Monitor[Table[
   {nx, ny, ({i, j} /. ALLPOSITIONS[[-1]]) ({Mod[i + nx, L, 1], Mod[j + ny, L, 1]} /. ALLPOSITIONS[[-1]])}
   , {i, L}, {j, L}, {nx, -L / 6, L / 6}, {ny, -L / 6, L / 6}], {i, j}]
```

```
{11, 42}
```

```
Mean@Mean@end // Flatten[#, 1] & // ListPlot3D[#, PlotRange → All, PlotStyle → Blue] &
```

200%

In[207]:=

```
βeff = 1 / (Mean@Mean@(energiesMatrices[[-1]]))
```

Out[207]=

```
1.38003
```

Propagator

In[210]:=

```
end = Monitor[Table[
   {nx, ny, ({i, j} /. ALLPOSITIONS[[-1]]) ({Mod[i + nx, L, 1], Mod[j + ny, L, 1]} /. ALLPOSITIONS[[-1]])}
   , {i, L}, {j, L}, {nx, -L / 6, L / 6}, {ny, -L / 6, L / 6}], {i, j}]
```

```
{16, 30}
```

```
experiment = Mean@Mean@end // Flatten[#, 1] & // ListPlot3D[#, PlotRange → All, PlotStyle → Blue] &
```

theory = $\frac{1}{L^2}$

200%

In[207]:=

```
βeff = 1 / (Mean@Mean@(energiesMatrices[[-1]]))
```

Out[207]=

```
1.38003
```

Propagator

In[210]:=

```
end = Monitor[Table[
   {nx, ny, ({i, j} /. ALLPOSITIONS[[-1]]) ({Mod[i + nx, L, 1], Mod[j + ny, L, 1]} /. ALLPOSITIONS[[-1]])}
   , {i, L}, {j, L}, {nx, -L / 6, L / 6}, {ny, -L / 6, L / 6}], {i, j}]
```

{25, 28}

```
experiment = Mean@Mean@end // Flatten[#, 1] & // ListPlot3D[#, PlotRange → All, PlotStyle → Blue] &
```

$$\text{theory} = \frac{1}{L^2}\frac{1}{\beta\text{eff}}\,\text{Sum}\left[\frac{\text{Exp}\left[-I\left(nx\,\frac{2\pi}{L}\,ix + ny\,\frac{2\pi}{L}\,iy\right)\right]}{\square}\right.$$

In[207]:=

```
βeff = 1 / (Mean@Mean@(energiesMatrices[[-1]]))
```

Out[207]=

```
1.38003
```

Propagator

In[210]:=

```
end = Monitor[Table[
   {nx, ny, ({i, j} /. ALLPOSITIONS[[-1]]) ({Mod[i + nx, L, 1], Mod[j + ny, L, 1]} /. ALLPOSITIONS[[-1]])}
   , {i, L}, {j, L}, {nx, -L / 6, L / 6}, {ny, -L / 6, L / 6}], {i, j}]
```

{35, 36}

```
experiment = Mean@Mean@end // Flatten[#, 1] & // ListPlot3D[#, PlotRange → All, PlotStyle → Blue] &
```

theory = Table[{nx, ny, $\frac{1}{L^2}\frac{1}{\beta eff}$ Sum[$\frac{\text{Exp}\left[-I\left(nx\frac{2\pi}{L}ix + ny\frac{2\pi}{L}iy\right)\right]}{\omega^2 + 4\kappa\left(\text{Sin}\left[\frac{2\pi}{L}\frac{ix}{2}\right]^2 + \text{Sin}\left[\frac{2\pi}{L}\frac{iy}{2}\right]^2\right)}$, {ix, -L / 2, L / 2}, {iy, -L / 2, L / 2}]},

200%

In[207]:=

```
βeff = 1 / (Mean@Mean@(energiesMatrices[[-1]]))
```

Out[207]=

```
1.38003
```

Propagator

In[210]:=

```
end = Monitor[Table[
   {nx, ny, ({i, j} /. ALLPOSITIONS[[-1]]) ({Mod[i + nx, L, 1], Mod[j + ny, L, 1]} /. ALLPOSITIONS[[-1]])}
   , {i, L}, {j, L}, {nx, -L / 6, L / 6}, {ny, -L / 6, L / 6}], {i, j}]
```

{35, 48}

```
experiment = Mean@Mean@end // Flatten[#, 1] & // ListPlot3D[#, PlotRange → All, PlotStyle → Blue] &
```

$$\text{theory} = \text{Table}\left[\left\{nx,\ ny,\ \frac{1}{L^2}\ \frac{1}{\beta eff}\ \text{Sum}\left[\frac{\text{Exp}\left[-I\left(nx\ \frac{2\pi}{L}\ ix + ny\ \frac{2\pi}{L}\ iy\right)\right]}{\omega^2 + 4\kappa\left(\text{Sin}\left[\frac{2\pi}{L}\ \frac{ix}{2}\right]^2 + \text{Sin}\left[\frac{2\pi}{L}\ \frac{iy}{2}\right]^2\right)},\ \{ix, -L/2, L/2\},\ \{iy, -L/2, L/2\}\right]\right\},\ \{nx, -L/6, L/6\},\ \{ny, -L/6, L/6\}\right]$$

200%

In[207]:=

```
βeff = 1 / (Mean@Mean@(energiesMatrices[[-1]]))
```

Out[207]=

```
1.38003
```

Propagator

In[210]:=

```
end = Monitor[Table[
   {nx, ny, ({i, j} /. ALLPOSITIONS[[-1]]) ({Mod[i + nx, L, 1], Mod[j + ny, L, 1]} /. ALLPOSITIONS[[-1]])}
   , {i, L}, {j, L}, {nx, -L / 6, L / 6}, {ny, -L / 6, L / 6}], {i, j}]
{38, 44}

experiment = Mean@Mean@end // Flatten[#, 1] & // ListPlot3D[#, PlotRange → All, PlotStyle → Blue] &

theory =
```

$$\text{Monitor}\left[\text{Table}\left[\left\{nx,\ ny,\ \frac{1}{L^2}\,\frac{1}{\beta eff}\,\text{Sum}\left[\frac{\text{Exp}\left[-I\left(nx\,\frac{2\pi}{L}\,ix + ny\,\frac{2\pi}{L}\,iy\right)\right]}{\omega^2 + 4\,\kappa\left(\text{Sin}\left[\frac{2\pi}{L}\,\frac{ix}{2}\right]^2 + \text{Sin}\left[\frac{2\pi}{L}\,\frac{iy}{2}\right]^2\right)},\ \{ix, -L/2, L/2\},\ \{iy, -L/2, L/2\}\right]\right\},\right.\right.$$

$$\left.\left.\{nx, -L/6, L/6\},\ \{ny, -L/6, L/6\}\right],\ \{nx, ny\}\right]\ //\ \text{Flatten}[\#, 1]\ \&\ //$$

```
ListPlot3D[#, PlotRange → All, PlotStyle → Blue] &
```

In[207]:=

```
βeff = 1 / (Mean@Mean@(energiesMatrices[[-1]]))
```

Out[207]=

```
1.38003
```

Propagator

In[210]:=

```
end = Monitor[Table[
   {nx, ny, ({i, j} /. ALLPOSITIONS[[-1]]) ({Mod[i + nx, L, 1], Mod[j + ny, L, 1]} /. ALLPOSITIONS[[-1]])}
   , {i, L}, {j, L}, {nx, -L / 6, L / 6}, {ny, -L / 6, L / 6}], {i, j}]
```

{41, 52}

```
experiment = Mean@Mean@end // Flatten[#, 1] & // ListPlot3D[#, PlotRange → All, PlotStyle → Blue] &
```

```
theory =
 Monitor[Table[{nx, ny, 1/L^2 1/βeff Sum[Exp[-I (nx 2π/L ix + ny 2π/L iy)] / (ω^2 + 4 κ (Sin[2π/L ix/2]^2 + Sin[2π/L iy/2]^2)), {ix, -L / 2, L / 2}, {iy, -L / 2, L / 2}]},
     {nx, -L / 6, L / 6}, {ny, -L / 6, L / 6}], {nx, ny}] // Flatten[#, 1] & //
  ListPlot3D[#, PlotRange → All, PlotStyle → Directive[Yellow, Specularity[White, 20], Opacity[0.5]]] &
```

$$\text{theory} = \text{Monitor}\left[\text{Table}\left[\left\{nx, ny, \frac{1}{L^2}\frac{1}{\beta eff}\text{Sum}\left[\frac{\text{Exp}\left[-I\left(nx\frac{2\pi}{L}ix + ny\frac{2\pi}{L}iy\right)\right]}{\omega^2 + 4\kappa\left(\text{Sin}\left[\frac{2\pi}{L}\frac{ix}{2}\right]^2 + \text{Sin}\left[\frac{2\pi}{L}\frac{iy}{2}\right]^2\right)}, \{ix, -L/2, L/2\}, \{iy, -L/2, L/2\}\right]\right\}, \ldots\right]\right]$$

200%

In[207]:=

```
βeff = 1 / (Mean@Mean@(energiesMatrices[[-1]]))
```

Out[207]=

```
1.38003
```

Propagator

In[210]:=

```
end = Monitor[Table[
   {nx, ny, ({i, j} /. ALLPOSITIONS[[-1]]) ({Mod[i + nx, L, 1], Mod[j + ny, L, 1]} /. ALLPOSITIONS[[-1]])}
   , {i, L}, {j, L}, {nx, -L / 6, L / 6}, {ny, -L / 6, L / 6}], {i, j}]
```

```
{45, 41}
```

```
experiment = Mean@Mean@end // Flatten[#, 1] & // ListPlot3D[#, PlotRange → All, PlotStyle → Blue] &
```

```
theory =
```

$$\text{Monitor}\left[\text{Table}\left[\left\{nx,\ ny,\ \frac{1}{L^2}\ \frac{1}{\beta eff}\ \text{Sum}\left[\frac{\text{Exp}\left[-I\left(nx\ \frac{2\pi}{L}\ ix + ny\ \frac{2\pi}{L}\ iy\right)\right]}{\omega^2 + 4\,\kappa\left(\text{Sin}\left[\frac{2\pi}{L}\ \frac{ix}{2}\right]^2 + \text{Sin}\left[\frac{2\pi}{L}\ \frac{iy}{2}\right]^2\right)},\ \{ix,\ -L/2,\ L/2\},\ \{iy,\ -L/2,\ L/2\}\right]\right\},\right.\right.$$

```
    {nx, -L / 6, L / 6}, {ny, -L / 6, L / 6}], {nx, ny}] // Flatten[#, 1] & //
  ListPlot3D[#, PlotRange → All, PlotStyle → Directive[Yellow, Specularity[White, 20], Opacity[0.5]]] &;
```

```
Show[theory, experiment]
```

200%

Propagator

In[210]:=

```
end = Monitor[Table[
   {nx, ny, ({i, j} /. ALLPOSITIONS[[-1]]) ({Mod[i + nx, L, 1], Mod[j + ny, L, 1]} /. ALLPOSITIONS[[-1]])}
   , {i, L}, {j, L}, {nx, -L / 6, L / 6}, {ny, -L / 6, L / 6}], {i, j}]
```

Out[210]=

```
{{⋯1⋯}, ⋯58⋯,
 {{{{-10, -10, 0.0253081}, {-10, -9, -0.0664338}, ⋯18⋯, {-10, 10, -0.029583}}, ⋯19⋯, {⋯1⋯}},
  ⋯58⋯, {⋯1⋯}}}
```

large output | show less | show more | show all | set size limit...

```
experiment = Mean@Mean@end // Flatten[#, 1] & // ListPlot3D[#, PlotRange → All, PlotStyle → Blue] &

theory =
 Monitor[Table[{nx, ny, 1/L^2 1/βeff Sum[Exp[-I (nx 2π/L ix + ny 2π/L iy)] / (ω^2 + 4 κ (Sin[2π/L ix/2]^2 + Sin[2π/L iy/2]^2)), {ix, -L / 2, L / 2}, {iy, -L / 2, L / 2}]},
     {nx, -L / 6, L / 6}, {ny, -L / 6, L / 6}], {nx, ny}] // Flatten[#, 1] & //
  ListPlot3D[#, PlotRange → All, PlotStyle → Directive[Yellow, Specularity[White, 20], Opacity[0.5]]] &;

Show[theory, experiment]
```

In[210]:=

```
end = Monitor[Table[
  {nx, ny, ({i, j} /. ALLPOSITIONS[[-1]]) ({Mod[i + nx, L, 1], Mod[j + ny, L, 1]} /. ALLPOSITIONS[[-1]])}
  , {i, L}, {j, L}, {nx, -L / 6, L / 6}, {ny, -L / 6, L / 6}], {i, j}]
```

Out[210]=

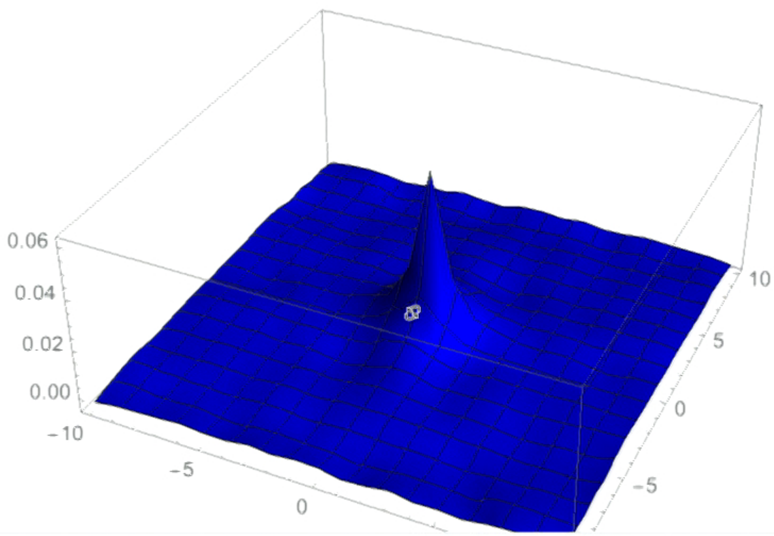
```
{{ ⋯ 1 ⋯ }, ⋯ 58 ⋯ ,
 {{{{-10, -10, 0.0253081}, {-10, -9, -0.0664338}, ⋯ 18 ⋯ , {-10, 10, -0.029583}}, ⋯ 19 ⋯ , { ⋯ 1 ⋯ }},
  ⋯ 58 ⋯ , { ⋯ 1 ⋯ }}}
```

large output | show less | show more | show all | set size limit...

In[211]:=

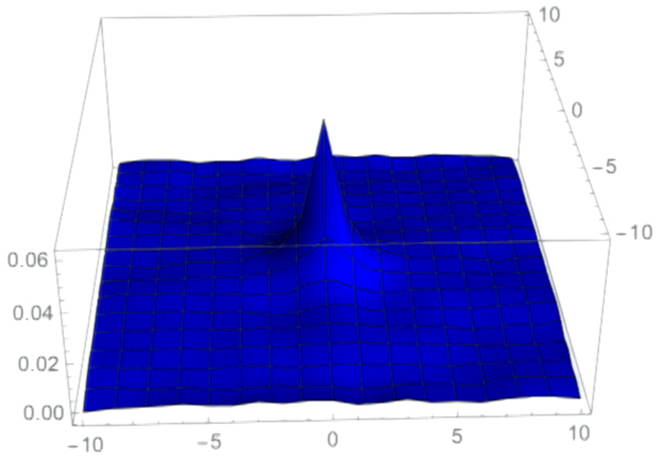
```
experiment = Mean@Mean@end // Flatten[#, 1] & // ListPlot3D[#, PlotRange → All, PlotStyle → Blue] &
```

Out[211]=



```
experiment = Mean@Mean@end // Flatten[#, 1] & // ListPlot3D[#, PlotRange → All, PlotStyle → Blue] &
```

Out[211]=

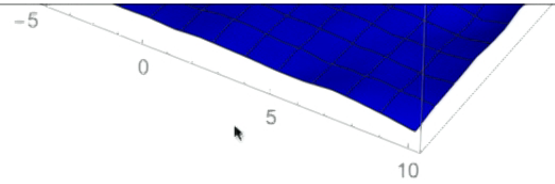


In[212]:=

```
theory =
  Monitor[Table[{nx, ny, 1/L^2 1/βeff Sum[Exp[-I (nx 2π/L ix + ny 2π/L iy)] / (ω^2 + 4 κ (Sin[2π/L ix/2]^2 + Sin[2π/L iy/2]^2)), {ix, -L/2, L/2}, {iy, -L/2, L/2}]},
      {nx, -L/6, L/6}, {ny, -L/6, L/6}], {nx, ny}] // Flatten[#, 1] & //
   ListPlot3D[#, PlotRange → All, PlotStyle → Directive[Yellow, Specularity[White, 20], Opacity[0.5]]] &;
```

{-10, -5}

```
Show[theory, experiment]
```

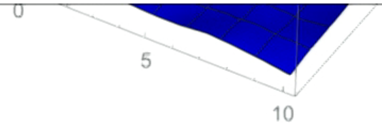


In[212]:=

```
theory =
  Monitor[Table[{nx, ny, 1/L^2 1/βeff Sum[Exp[-I (nx 2π/L ix + ny 2π/L iy)] / (ω^2 + 4 κ (Sin[2π/L ix/2]^2 + Sin[2π/L iy/2]^2)), {ix, -L/2, L/2}, {iy, -L/2, L/2}]},
      {nx, -L/6, L/6}, {ny, -L/6, L/6}], {nx, ny}] // Flatten[#, 1] & //
    ListPlot3D[#, PlotRange → All, PlotStyle → Directive[Yellow, Specularity[White, 20], Opacity[0.5]]] &;
```

{7, -9}

```
Show[theory, experiment]
```



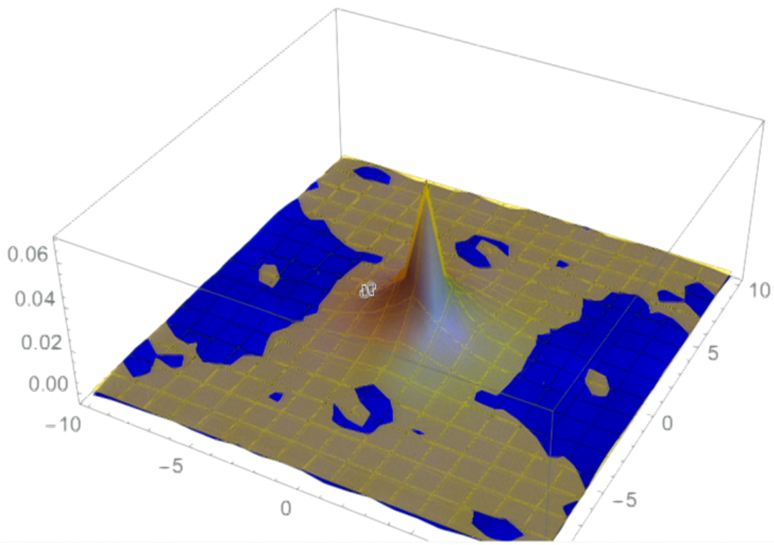
In[212]:=

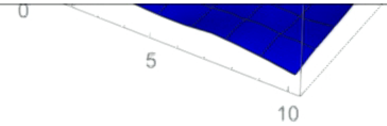
```
theory =
  Monitor[Table[{nx, ny, 1/L^2 1/βeff Sum[Exp[-I (nx 2π/L ix + ny 2π/L iy)] / (ω^2 + 4 κ (Sin[2π/L ix/2]^2 + Sin[2π/L iy/2]^2)), {ix, -L/2, L/2}, {iy, -L/2, L/2}]},
      {nx, -L/6, L/6}, {ny, -L/6, L/6}], {nx, ny}] // Flatten[#, 1] & //
    ListPlot3D[#, PlotRange → All, PlotStyle → Directive[Yellow, Specularity[White, 20], Opacity[0.5]]] &;
```

In[213]:=

```
Show[theory, experiment]
```

Out[213]=





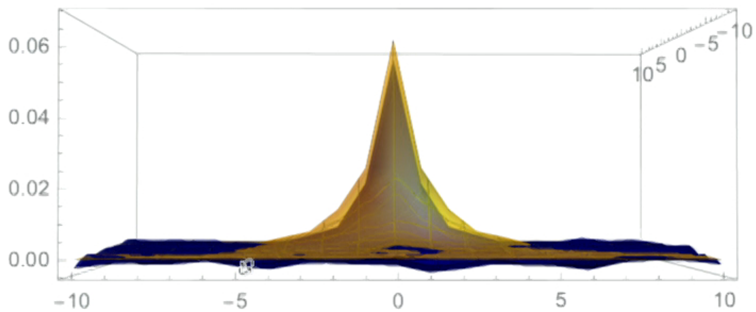
In[212]:=

```
theory =
  Monitor[Table[{nx, ny, 1/L^2 1/βeff Sum[Exp[-I (nx 2π/L ix + ny 2π/L iy)]/(ω^2 + 4 κ (Sin[2π/L ix/2]^2 + Sin[2π/L iy/2]^2)), {ix, -L/2, L/2}, {iy, -L/2, L/2}]},
      {nx, -L/6, L/6}, {ny, -L/6, L/6}], {nx, ny}] // Flatten[#, 1] & //
    ListPlot3D[#, PlotRange → All, PlotStyle → Directive[Yellow, Specularity[White, 20], Opacity[0.5]]] &;
```

In[213]:=

```
Show[theory, experiment]
```

Out[213]=

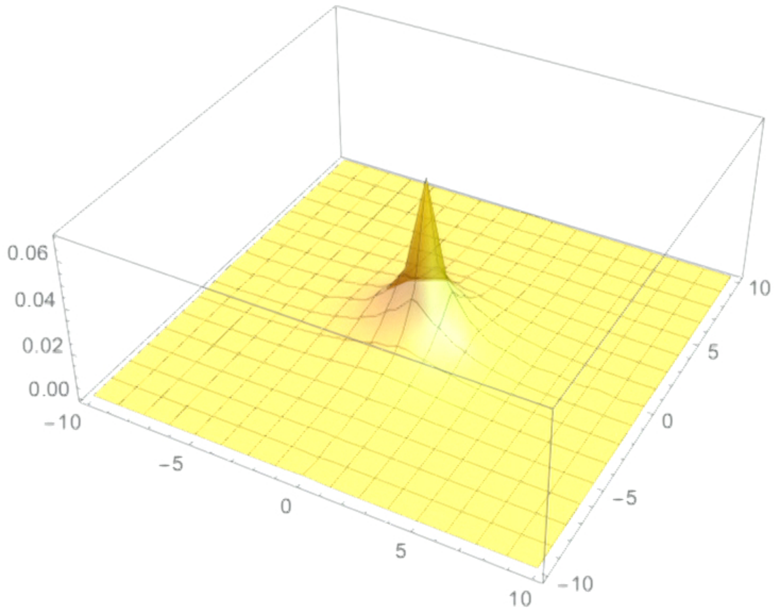


```
theory =
  Monitor[Table[{nx, ny, 1/L^2 1/βeff Sum[Exp[-I (nx 2π/L ix + ny 2π/L iy)] / (ω^2 + 4 κ (Sin[2π/L ix/2]^2 + Sin[2π/L iy/2]^2)), {ix, -L/2, L/2}, {iy, -L/2, L/2}],
     {nx, -L/6, L/6}, {ny, -L/6, L/6}], {nx, ny}] // Flatten[#, 1] & //
   ListPlot3D[#, PlotRange → All, PlotStyle → Directive[Yellow, Specularity[White, 20], Opacity[0.5]]] &;
```

In[214]:=

```
theory
```

Out[214]=

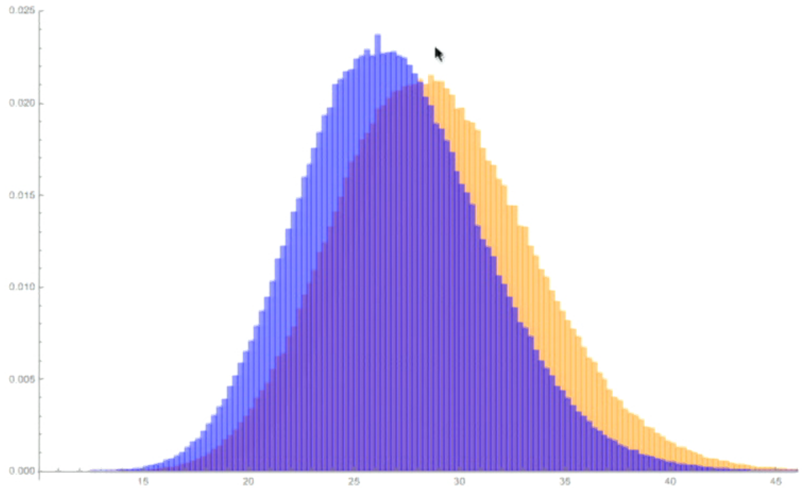


In[213]:=

```
l = L / 4;
energiesMatricesBig = Table[Table[Mean@Flatten@(energiesMatrices[[n, l (j - 1) + 1 ;; l j, l (i - 1) + 1 ;; l i]]), {i, L / l}, {j, L / l}], {n, Length[energiesMatrices]}];

(Flatten /@ energiesMatricesBig)' // ListPlot;
```

Bins

```
dL = 6;
tableBins = Monitor[Table[Flatten[Table[Sum[energiesMatrices[[t, Mod[i, L, 1], Mod[j, L, 1]]], {i, ii, ii + dL - 1}, {j, jj, jj + dL - 1}], {ii, L}, {jj, L}]], {t, 1, 501}], t];

intermediateAndFinal = (tableBins // Take[#, {100, 250}] & // Flatten, tableBins // Drop[#, 350] & // Flatten) // Histogram[#, {1 / 4}, "Probability", ChartStyle → {Orange, Blue}] &
```




```
(energyMean = (Drop[tableBins, 350] // Flatten // Mean) / dL^2) dL^2
βeff = 1 / energyMean

26.5835

1.35422

(energyMean = (Take[tableBins, {100, 250}] // Flatten // Mean) / dL^2) dL ^ 2
βintermediate = 1 / energyMean

28.6181

1.25795

Manipulate[tableBins[[j]] // Histogram[#, {1 / 2}, "Probability", PlotRange → {{0, 50}, {0, 0.06}}] &, {j, 1, 500, 1}]
```

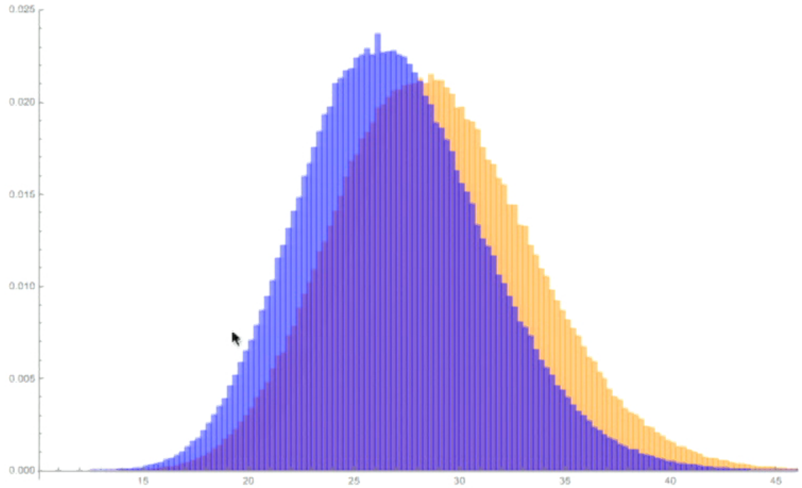
```
(Flatten /@ energiesMatricesBig)' // ListPlot;
```

Bins

```
In[215]:= dL = 6;
tableBins = Monitor[Table[Flatten[Table[Sum[energiesMatrices[[t, Mod[i, L, 1], Mod[j, L, 1]]], {i, ii, ii + dL - 1}, {j, jj, jj + dL - 1}], {ii, L}, {jj, L}]], {t, 1, 501}], t];
```

24

```
intermediateAndFinal = (tableBins // Take[#, {100, 250}] & // Flatten, tableBins // Drop[#, 350] & // Flatten) // Histogram[#, (1/4), "Probability", ChartStyle → (Orange, Blue)] &
```




```
(energyMean = (Drop[tableBins, 350] // Flatten // Mean) / dL^2) dL^2
βeff = 1 / energyMean
```

26.5835

1.35422

```
(energyMean = (Take[tableBins, {100, 250}] // Flatten // Mean) / dL^2) dL^2
βintermediate = 1 / energyMean
```

28.6181

1.25795

```
Manipulate[tableBins[[j]] // Histogram[#, {1/2}, "Probability", PlotRange → {{0, 50}, {0, 0.06}}] &, {j, 1, 500, 1}]
```

··· Part: Part specification tableBins⟦500⟧ is longer than depth of object.

··· Histogram: tableBins⟦500⟧ is not a valid dataset or list of datasets.

··· Histogram: tableBins⟦500⟧ is not a valid dataset or list of datasets.

··· Histogram: tableBins⟦500⟧ is not a valid dataset or list of datasets.

··· General: Further output of Histogram::ldata will be suppressed during this calculation.

Partition function and predictions

Define density of states (computed in tutorial 2) as a function of the systems energy 𝕌 and size 𝕃

```
g[𝕌_, 𝕃_] := 𝕌^(𝕃^2-1) ;
```

Define $\mathbb{P}_{\text{theory}}[\mathbb{U}]$

```
ℙt[𝕌_, 𝕃_] := (g[𝕌, 𝕃] Exp[-β 𝕌]) / (β^(-𝕃^2) Gamma[𝕃^2]);
```

This is normalized:

```
Integrate[g[𝕌, 𝕃] Exp[-β 𝕌], {𝕌, 0, Infinity}, GenerateConditions → False]
```

$\beta^{-\mathbb{L}^2}$ Gamma$[\mathbb{L}^2]$

Integrate $\mathbb{P}_{\text{theory}}[\mathbb{U}]$ from average energy per particle from α to $\alpha + d\alpha$ (for future comparison with histogram)

```
Intℙ[α_, dα_, 𝕃_, β0_] = Integrate[ℙt[𝕌, 𝕃] /. β → β0, {𝕌, α, α + dα}, GenerateConditions → False]
```

```
g[𝕌_, 𝕃_] := 𝕌^( … ) ;
```

Define $\mathbb{P}_{theory}[\mathbb{U}]$

```
ℙt[𝕌_, 𝕃_] := (g[𝕌, 𝕃] Exp[-β 𝕌]) / (β^(-𝕃^2) Gamma[𝕃^2]);
```

This is normalized:

```
Integrate[g[𝕌, 𝕃] Exp[-β 𝕌], {𝕌, 0, Infinity}, GenerateConditions → False]
```

$$\beta^{-\mathbb{L}^2}\,\text{Gamma}\left[\mathbb{L}^2\right]$$

Integrate $\mathbb{P}_{theory}[\mathbb{U}]$ from average energy per particle from α to $\alpha + d\alpha$ (for future comparison with histogram)

```
IntℙP[α_, dα_, 𝕃_, β0_] = Integrate[ℙt[𝕌, 𝕃] /. β → β0, {𝕌, α, α + dα}, GenerateConditions → False]
```

$$\frac{\beta 0^{\mathbb{L}^2}\left(\alpha^{\mathbb{L}^2}\,\text{ExpIntegralE}\left[1-\mathbb{L}^2,\ \alpha\,\beta 0\right]-(d\alpha+\alpha)^{\mathbb{L}^2}\,\text{ExpIntegralE}\left[1-\mathbb{L}^2,\ (d\alpha+\alpha)\,\beta 0\right]\right)}{\text{Gamma}\left[\mathbb{L}^2\right]}$$

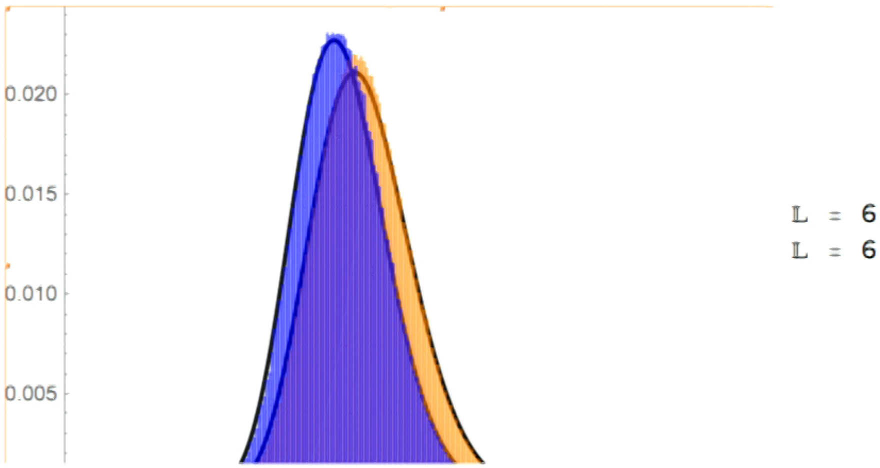
Comparison

```
Plot[IntℙP[α, 1 / 4, 6, βintermediate], {α, 1 / 10, 36 + 40}, PlotLegends → "𝕃 = " <> ToString[6],
  PlotStyle → {Thick, Black}, PlotRange → All];
Plot[IntℙP[α, 1 / 4, 6, βeff], {α, 1 / 10, 36 + 40}, PlotLegends → "𝕃 = " <> ToString[6],
  PlotStyle → {Thick, Black}, PlotRange → All];
Show[%, %%, intermediateAndFinal]
```

Gamma[$\mathbb{L}^2$]

Comparison

```
Plot[IntP[α, 1/4, 6, βintermediate], {α, 1/10, 36 + 40}, PlotLegends → "L = " <> ToString[6],
  PlotStyle → {Thick, Black}, PlotRange → All];
Plot[IntP[α, 1/4, 6, βeff], {α, 1/10, 36 + 40}, PlotLegends → "L = " <> ToString[6],
  PlotStyle → {Thick, Black}, PlotRange → All];
Show[%, %%, intermediateAndFinal]
```



Both work ?!

Propagator

Size: 822 x 490

```
26.5835

1.35422

(energyMean = (Take[tableBins, {100, 250}] // Flatten // Mean) / dL^2) dL^2

βintermediate = 1 / energyMean

28.6181

1.25795

Manipulate[tableBins[[j]] // Histogram[#, {1 / 2}, "Probability", PlotRange → {{0, 50}, {0, 0.06}}] &,
 {j, 1, 500, 1}]
```

j

```
Histogram[tableBins[[500]], {1/2}, Probability, PlotRange → {{0, 50}, {0, 0.06}}]
```

··· Part: Part specification tableBins[[500]] is longer than depth of object.

··· Histogram: tableBins[[500]] is not a valid dataset or list of datasets.

··· Histogram: tableBins[[500]] is not a valid dataset or list of datasets.

··· Histogram: tableBins[[500]] is not a valid dataset or list of datasets.

··· General: Further output of Histogram::ldata will be suppressed during this calculation.

Partition function and predictions

```
l = L / 4;
energiesMatricesBig =
  Table[Table[Mean@Flatten@(energiesMatrices[[n, l (j - 1) + 1 ;; l j, l (i - 1) + 1 ;; l i]]), {i, L / l}, {j, L / l}],
   {n, Length[energiesMatrices]}];

(Flatten /@ energiesMatricesBig)ᵀ // ListPlot;
```

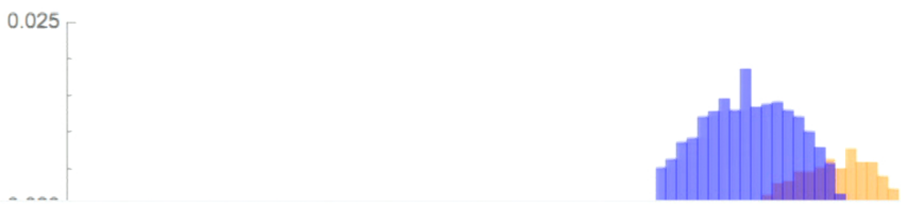
Bins

In[217]:=

```
dL = 6;
tableBins =
  Monitor[
   Table[
    Flatten[Table[Sum[energiesMatrices[[t, Mod[i, L, 1], Mod[j, L, 1]]], {i, ii, ii + dL - 1}, {j, jj, jj + dL - 1}],
      {ii, L}, {jj, L}]], {t, 1, 501}], t];
```

10

```
intermediateAndFinal = {tableBins // Take[#, {100, 250}] & // Flatten, tableBins // Drop[#, 350] & // Flatten} //
  Histogram[#, {1 / 4}, "Probability", ChartStyle → {Orange, Blue}] &
```



26.5835

1.35422

$$\left(\text{energyMean} = \frac{\text{Take[tableBins, \{100, 250\}] // Flatten // Mean}}{dL^2}\right) \text{dL}\wedge 2$$

```
βintermediate = 1 / energyMean
```

28.6181

1.25795

```
Manipulate[tableBins[[j]] // Histogram[#, {1 / 2}, "Probability", PlotRange → {{0, 50}, {0, 0.06}}] &,
 {j, 1, 500, 1}]
```

Partition function and predictions

Define density of states (computed in tutorial 2) as a function of the systems energy $\mathbb{U}$ and size $\mathbb{L}$

$$g[\mathbb{U}\_, \mathbb{L}\_] := \mathbb{U}^{(\mathbb{L}^2-1)};$$

Define $\mathbb{P}_{theory}[\mathbb{U}]$

$$\mathbb{P}_t[\mathbb{U}\_, \mathbb{L}\_] := \frac{g[\mathbb{U}, \mathbb{L}]\ \text{Exp}[-\beta\ \mathbb{U}]}{\beta^{-\mathbb{L}^2}\ \text{Gamma}[\mathbb{L}^2]};$$

This is normalized:

```
Integrate[g[U, L] Exp[-β U], {U, 0, Infinity}, GenerateConditions → False]
```

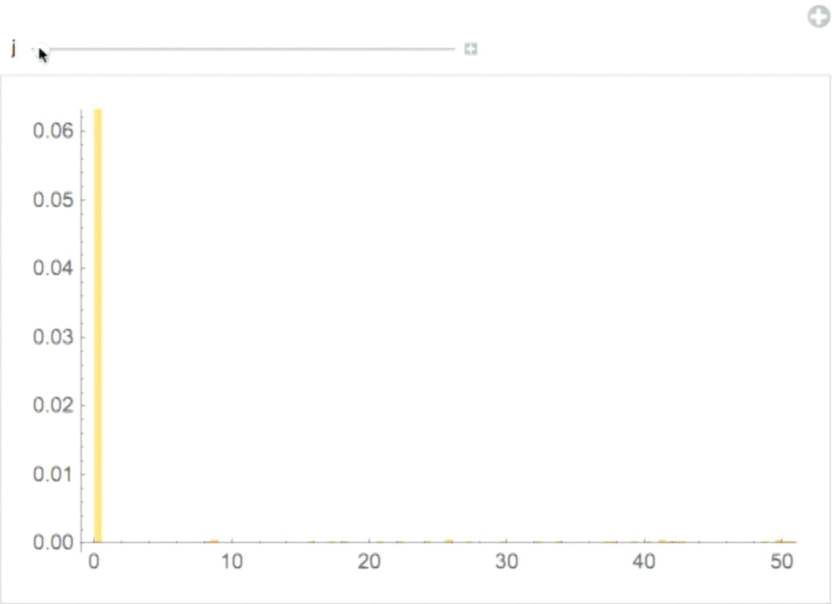
200%

1.25795

In[219]:=

```
Manipulate[tableBins[[j]] // Histogram[#, {1 / 2}, "Probability", PlotRange → {{0, 50}, {0, 0.06}}] &,
 {j, 1, 500, 1}]
```

Out[219]=



Partition function and predictions

Define density of states (computed in tutorial 2) as a function of the systems energy 𝕌 and size 𝕃

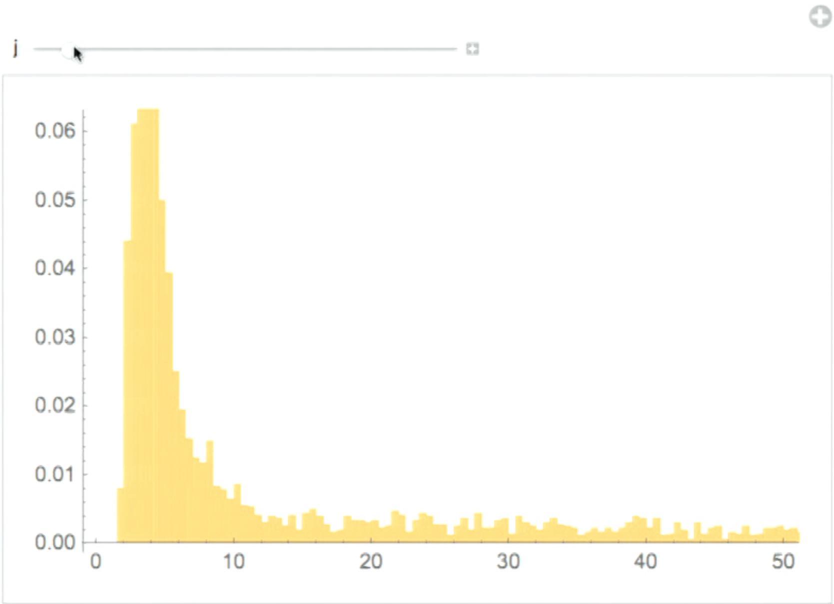
```
g[𝕌 , 𝕃 ] := 𝕌^(𝕃^2-1) ;
```

1.25795

In[219]:=

```
Manipulate[tableBins[[j]] // Histogram[#, {1 / 2}, "Probability", PlotRange → {{0, 50}, {0, 0.06}}] &,
 {j, 1, 500, 1}]
```

Out[219]=



Partition function and predictions

Define density of states (computed in tutorial 2) as a function of the systems energy 𝕌 and size 𝕃

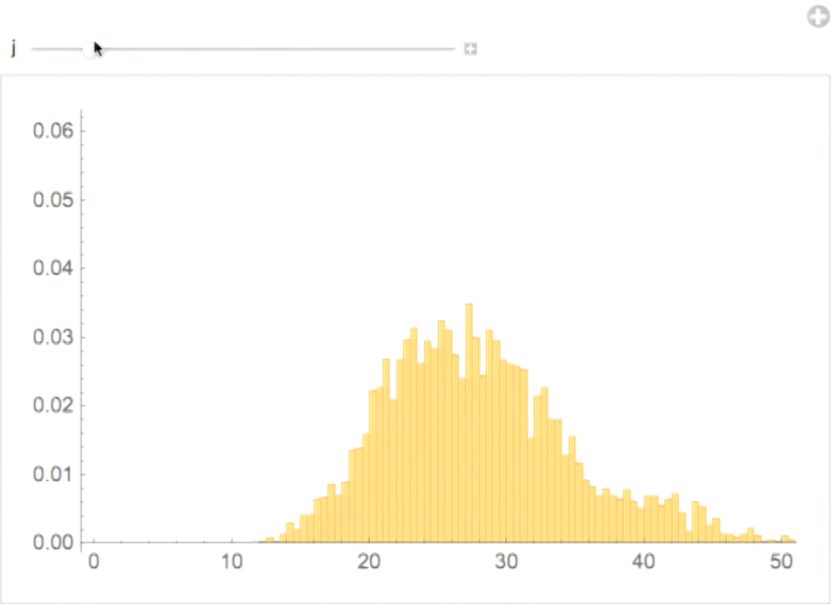
$g[\mathbb{U}, \mathbb{L}] := \mathbb{U}^{(\mathbb{L}^2-1)}$;

1.25795

In[219]:=

```
Manipulate[tableBins[[j]] // Histogram[#, {1 / 2}, "Probability", PlotRange → {{0, 50}, {0, 0.06}}] &,
 {j, 1, 500, 1}]
```

Out[219]=



Partition function and predictions

Define density of states (computed in tutorial 2) as a function of the systems energy 𝕌 and size 𝕃

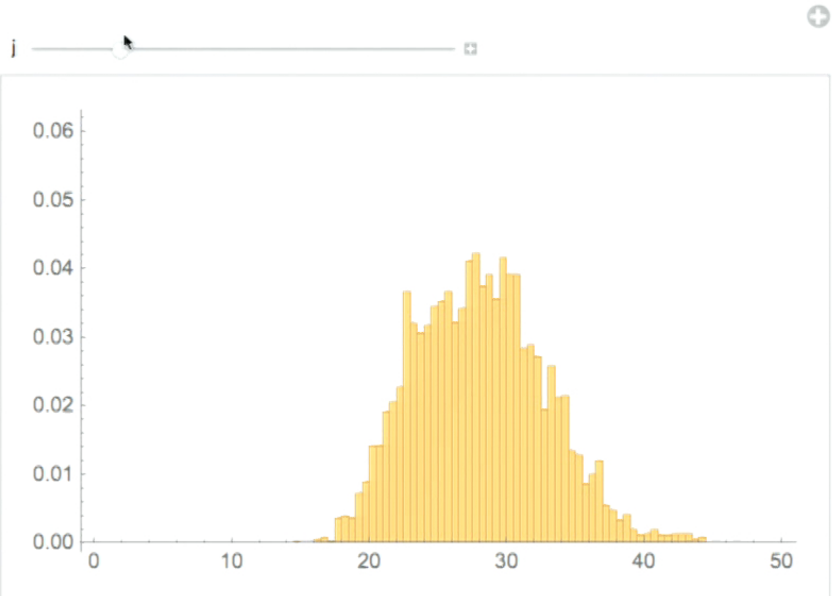
```
g[𝕌 , 𝕃 ] := 𝕌^(𝕃^2-1) ;
```

1.25795

In[219]:=

```
Manipulate[tableBins[[j]] // Histogram[#, {1 / 2}, "Probability", PlotRange → {{0, 50}, {0, 0.06}}] &,
 {j, 1, 500, 1}]
```

Out[219]=



Partition function and predictions

Define density of states (computed in tutorial 2) as a function of the systems energy 𝕌 and size 𝕃

```
g[𝕌 , 𝕃 ] := 𝕌^(𝕃^2-1) ;
```

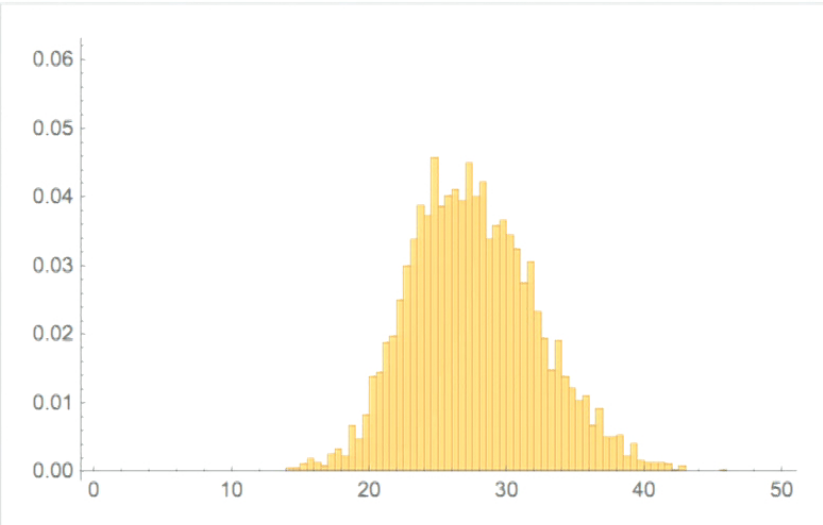
1.25795

In[219]:=

```
Manipulate[tableBins[[j]] // Histogram[#, {1 / 2}, "Probability", PlotRange → {{0, 50}, {0, 0.06}}] &,
 {j, 1, 500, 1}]
```

Out[219]=

j



Partition function and predictions

Define density of states (computed in tutorial 2) as a function of the systems energy 𝕌 and size 𝕃

```
g[𝕌_, 𝕃_] := 𝕌^(𝕃^2-1);
```

```
intermediateAndFinal = {tableBins // Take[#, {100, 250}] & // Flatten, tableBins // Drop[#, 350] & // Flatten} //
  Histogram[#, {1 / 4}, "Probability", ChartStyle → {Orange, Blue}] &
```