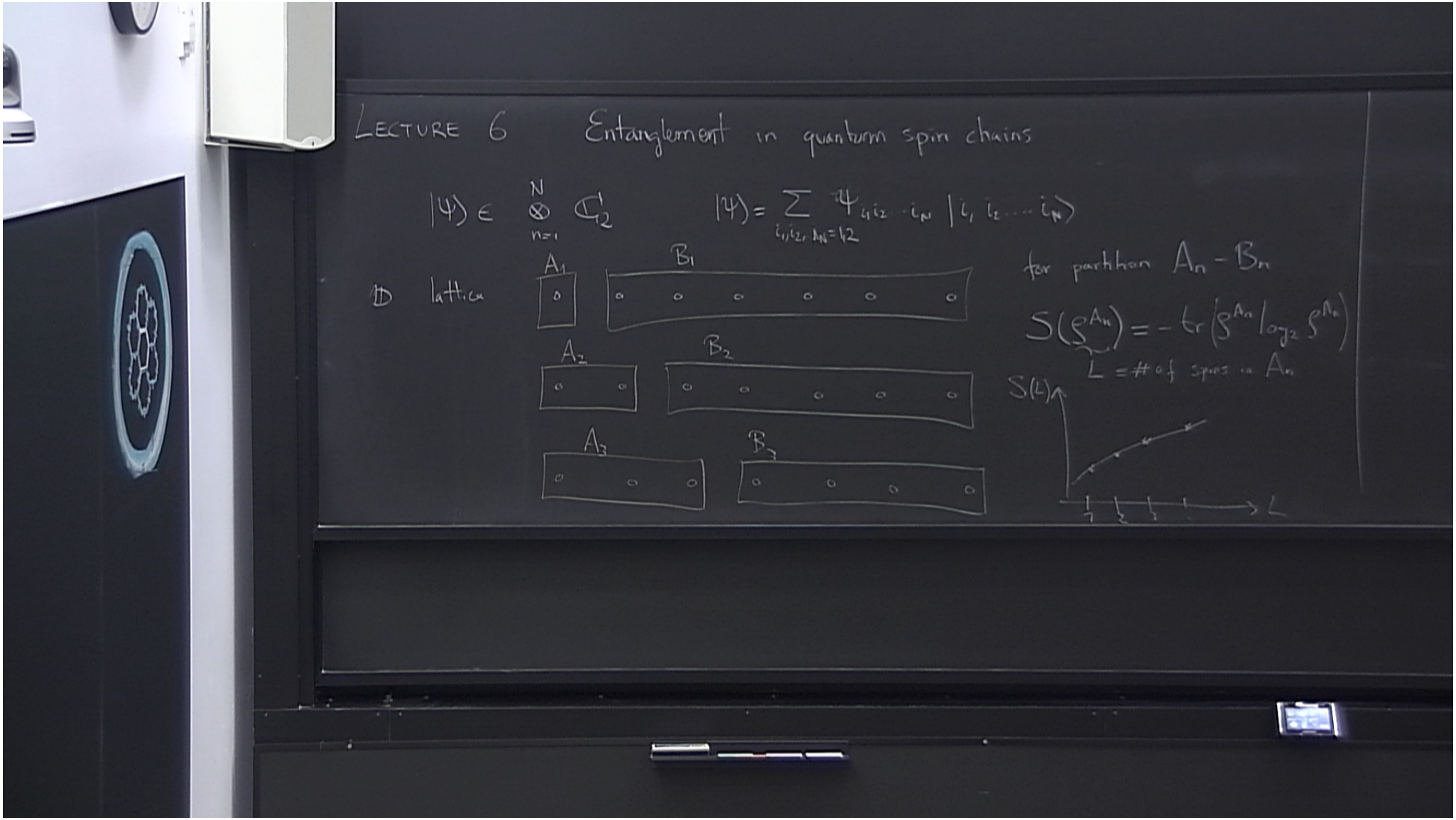
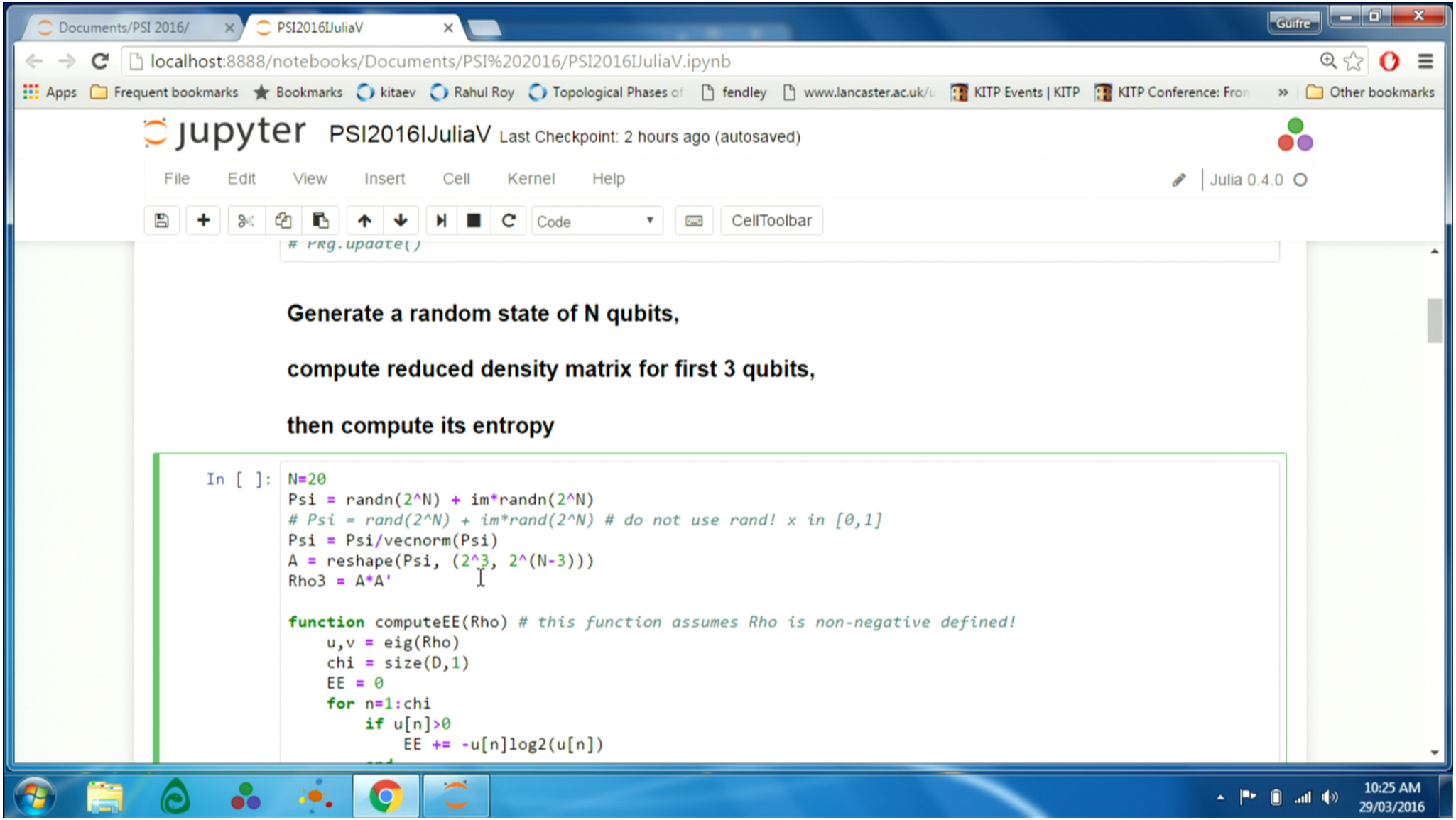
Title: PSI 2015/2016 Explorations in Condensed Matter - Guifre Vidal - 6

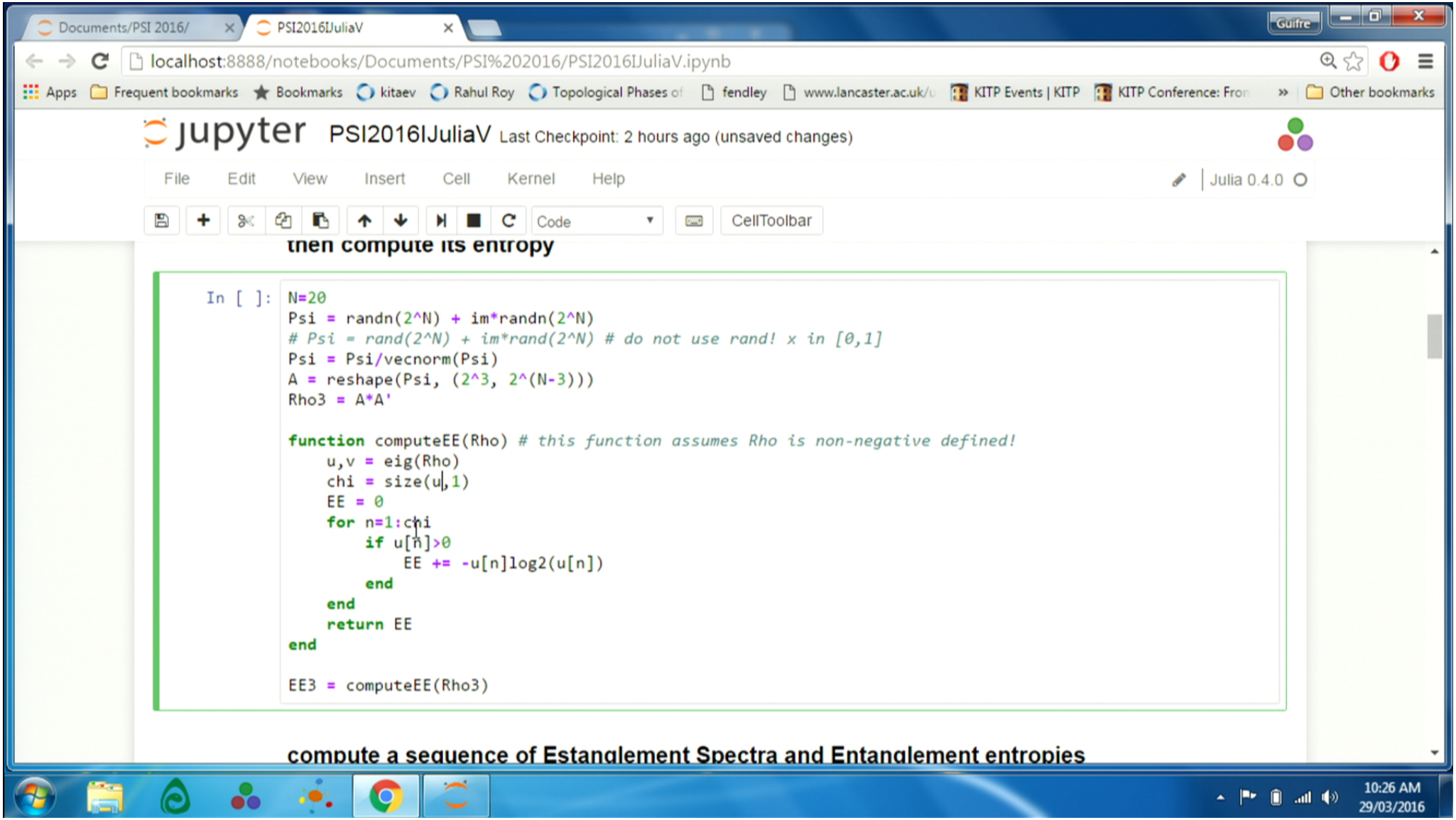
Date: Mar 29, 2016 10:15 AM

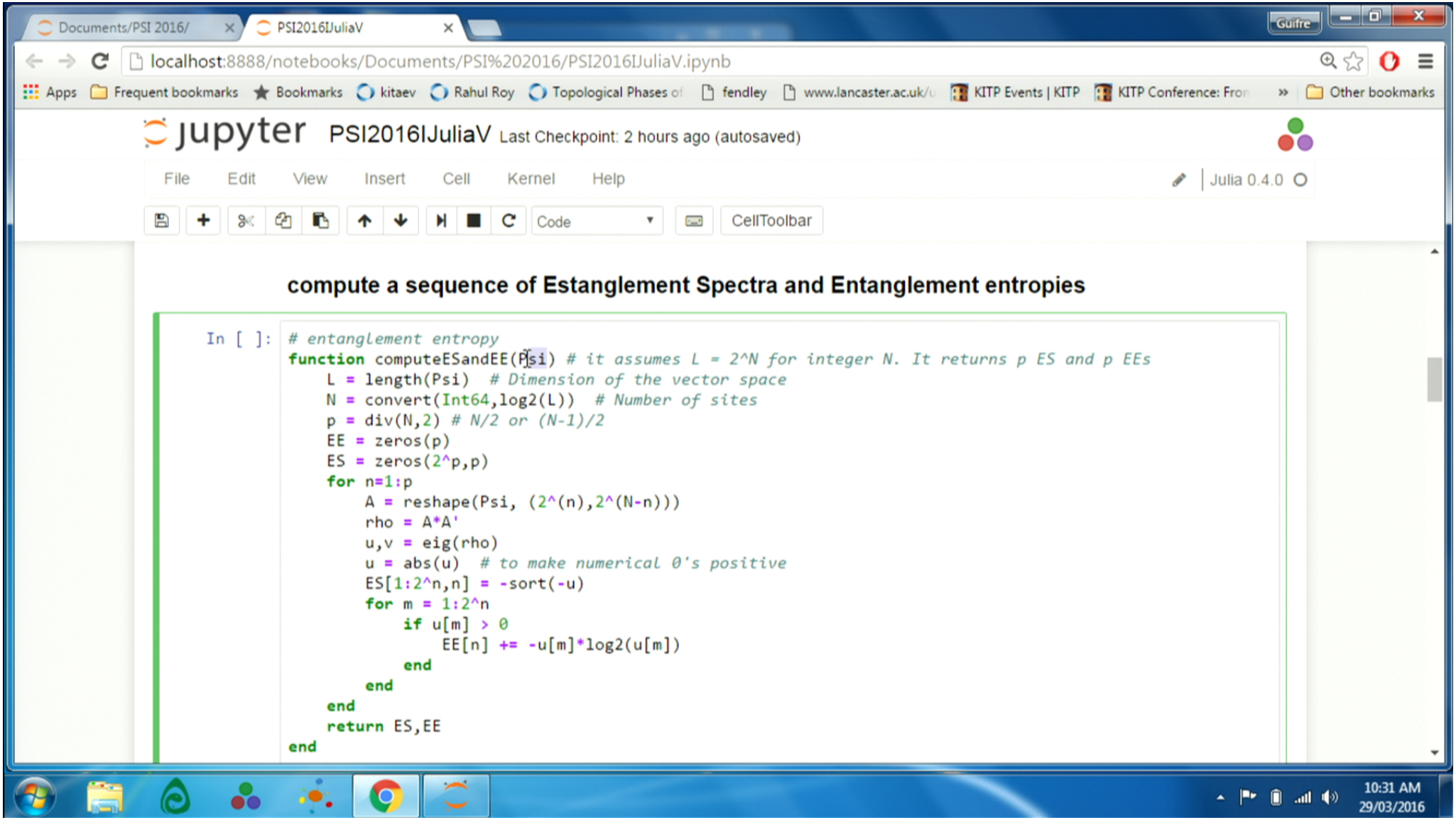
URL: http://pirsa.org/16030013

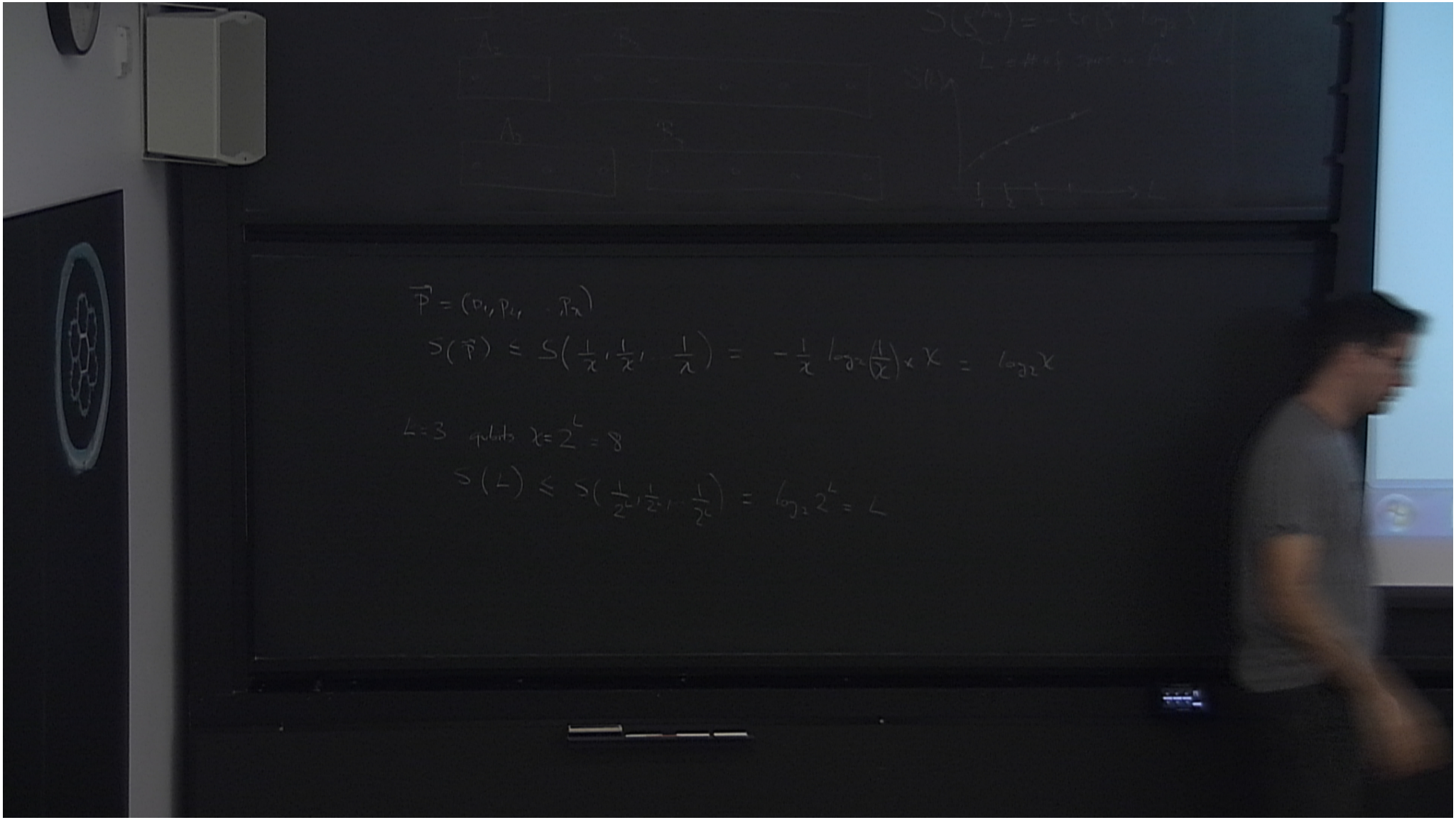
Abstract:

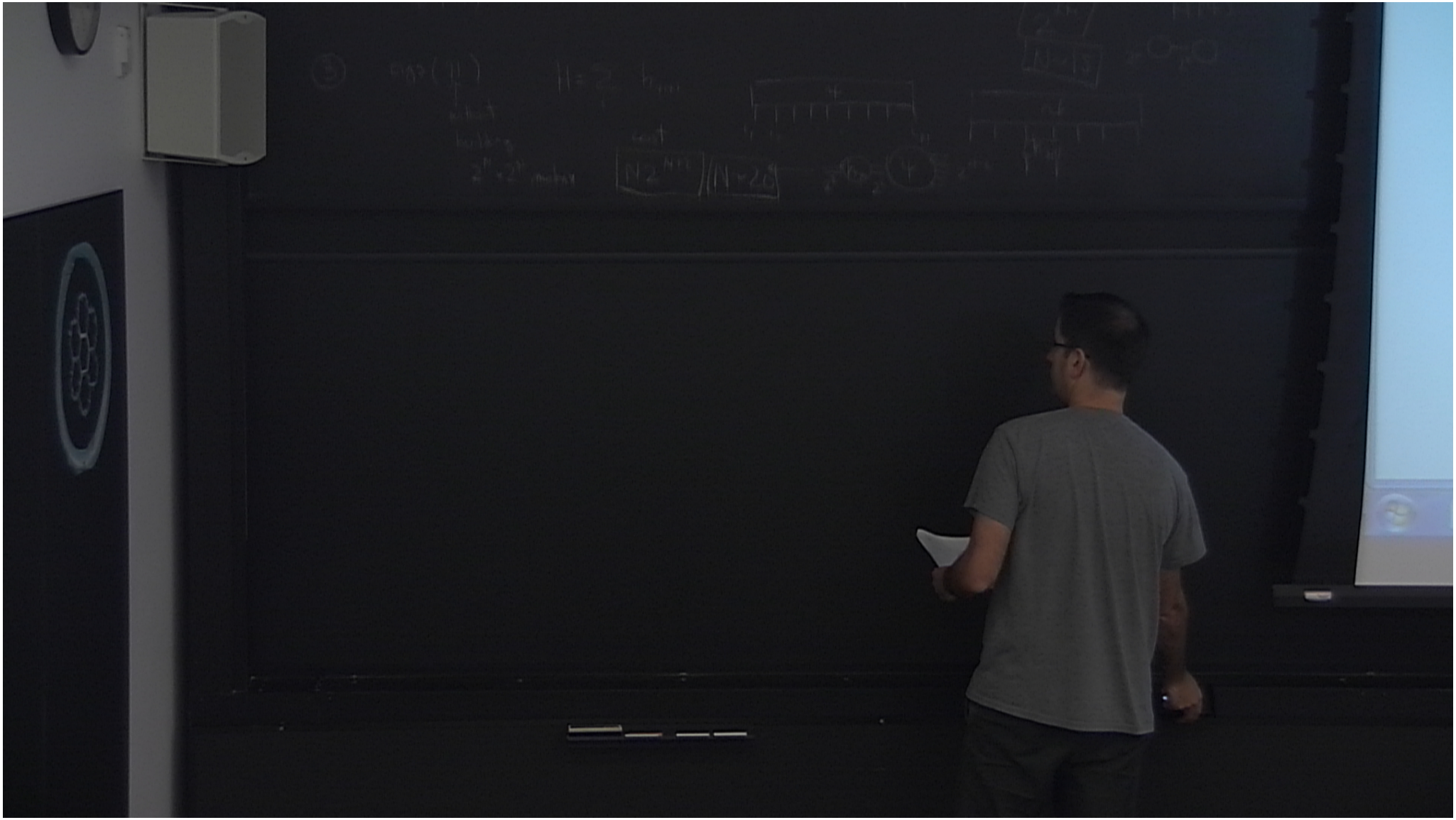

LECTURE 6 Entanglement in quantum spin chains
$|\psi\rangle \in \bigotimes_{n=1}^{N} \mathbb{C}_2^1$
$|\psi\rangle = \sum_{i_1,i_2,\ldots,i_N=\downarrow,\uparrow} \psi_{i_1 i_2 \cdots i_N} |i_1, i_2, \ldots, i_N\rangle$
1D lattice
A_1 B_1
A_2 B_2
A_3 B_3
for partition $A_n - B_n$
$S(\rho^{A_n}) = -\mathrm{tr}(\rho^{A_n} \log_2 \rho^{A_n})$
L = # of spins in A_n
$S(L)$
L

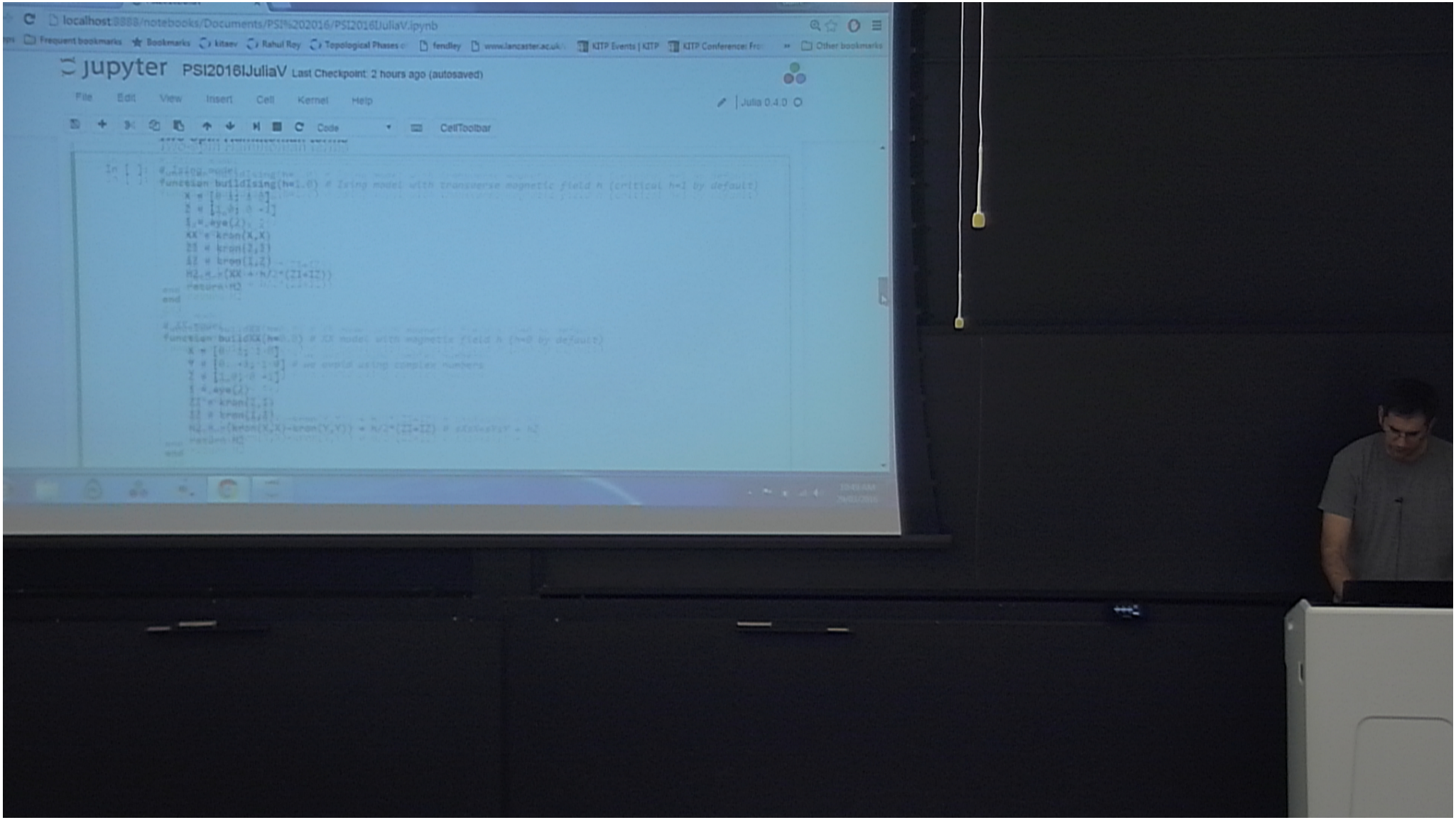

Documents/PSI 2016/
PSI2016IJuliaV
localhost:8888/notebooks/Documents/PSI%202016/PSI2016IJuliaV.ipynb
jupyter PSI2016IJuliaV Last Checkpoint: 2 hours ago (autosaved)
File Edit View Insert Cell Kernel Help
Julia 0.4.0
Code
CellToolbar
rng.update()
Generate a random state of N qubits,
compute reduced density matrix for first 3 qubits,
then compute its entropy
In [ ]:
N=20
Psi = randn(2^N) + im*randn(2^N)
# Psi = rand(2^N) + im*rand(2^N) # do not use rand! x in [0,1]
Psi = Psi/vecnorm(Psi)
A = reshape(Psi, (2^3, 2^(N-3)))
Rho3 = A*A'
function computeEE(Rho) # this function assumes Rho is non-negative defined!
u,v = eig(Rho)
chi = size(D,1)
EE = 0
for n=1:chi
if u[n]>0
EE += -u[n]log2(u[n])
10:25 AM
29/03/2016

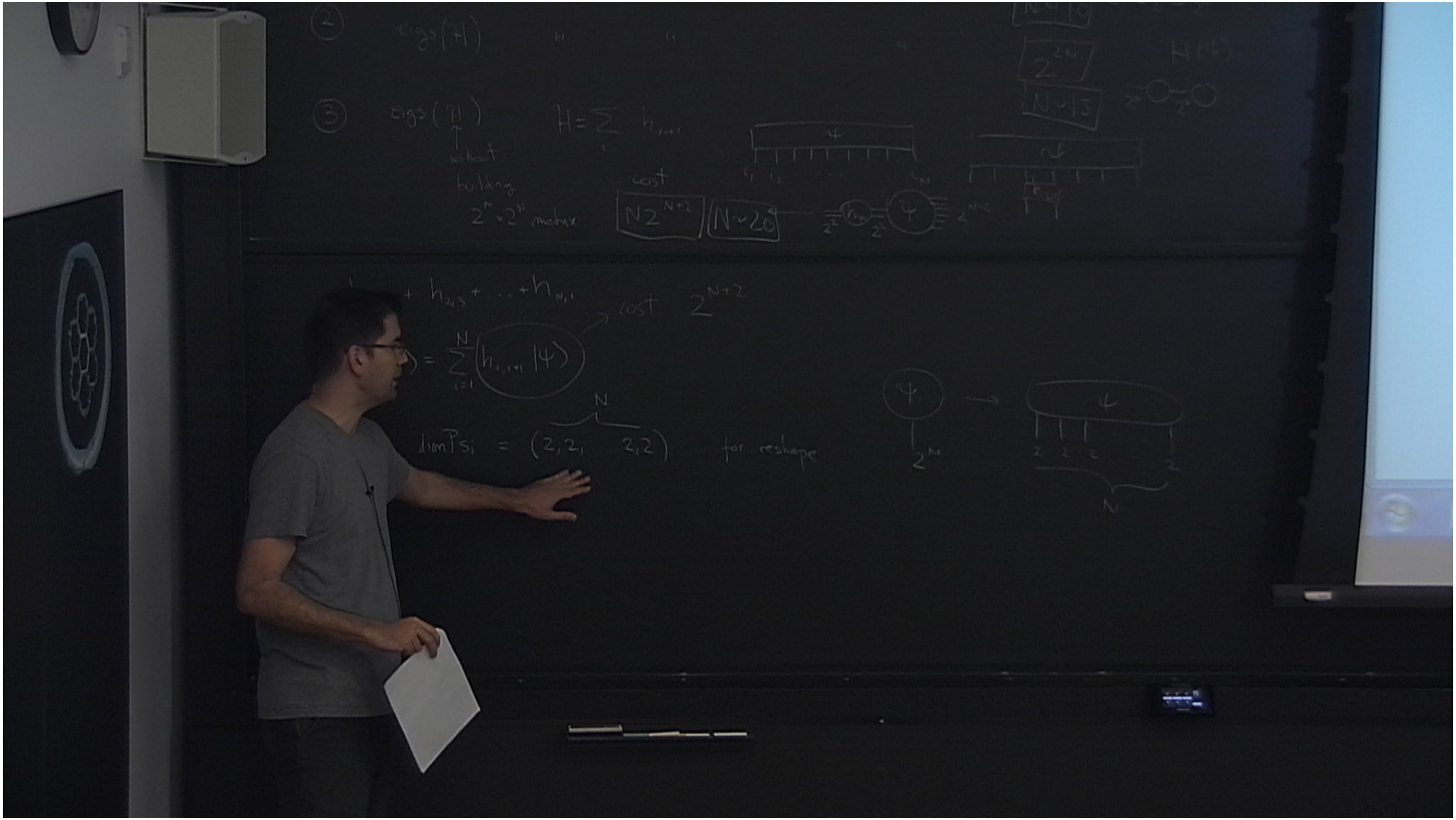

Documents/PSI 2016/
PSI2016IJuliaV
localhost:8888/notebooks/Documents/PSI%202016/PSI2016IJuliaV.ipynb
jupyter PSI2016IJuliaV Last Checkpoint: 2 hours ago (unsaved changes)
File Edit View Insert Cell Kernel Help
Julia 0.4.0
Code
CellToolbar
then compute its entropy
In [ ]:
N=20
Psi = randn(2^N) + im*randn(2^N)
# Psi = rand(2^N) + im*rand(2^N) # do not use rand! x in [0,1]
Psi = Psi/vecnorm(Psi)
A = reshape(Psi, (2^3, 2^(N-3)))
Rho3 = A*A'
function computeEE(Rho) # this function assumes Rho is non-negative defined!
u,v = eig(Rho)
chi = size(u,1)
EE = 0
for n=1:chi
if u[n]>0
EE += -u[n]log2(u[n])
end
end
return EE
end
EE3 = computeEE(Rho3)
compute a sequence of Estanglement Spectra and Entanglement entropies
10:26 AM
29/03/2016

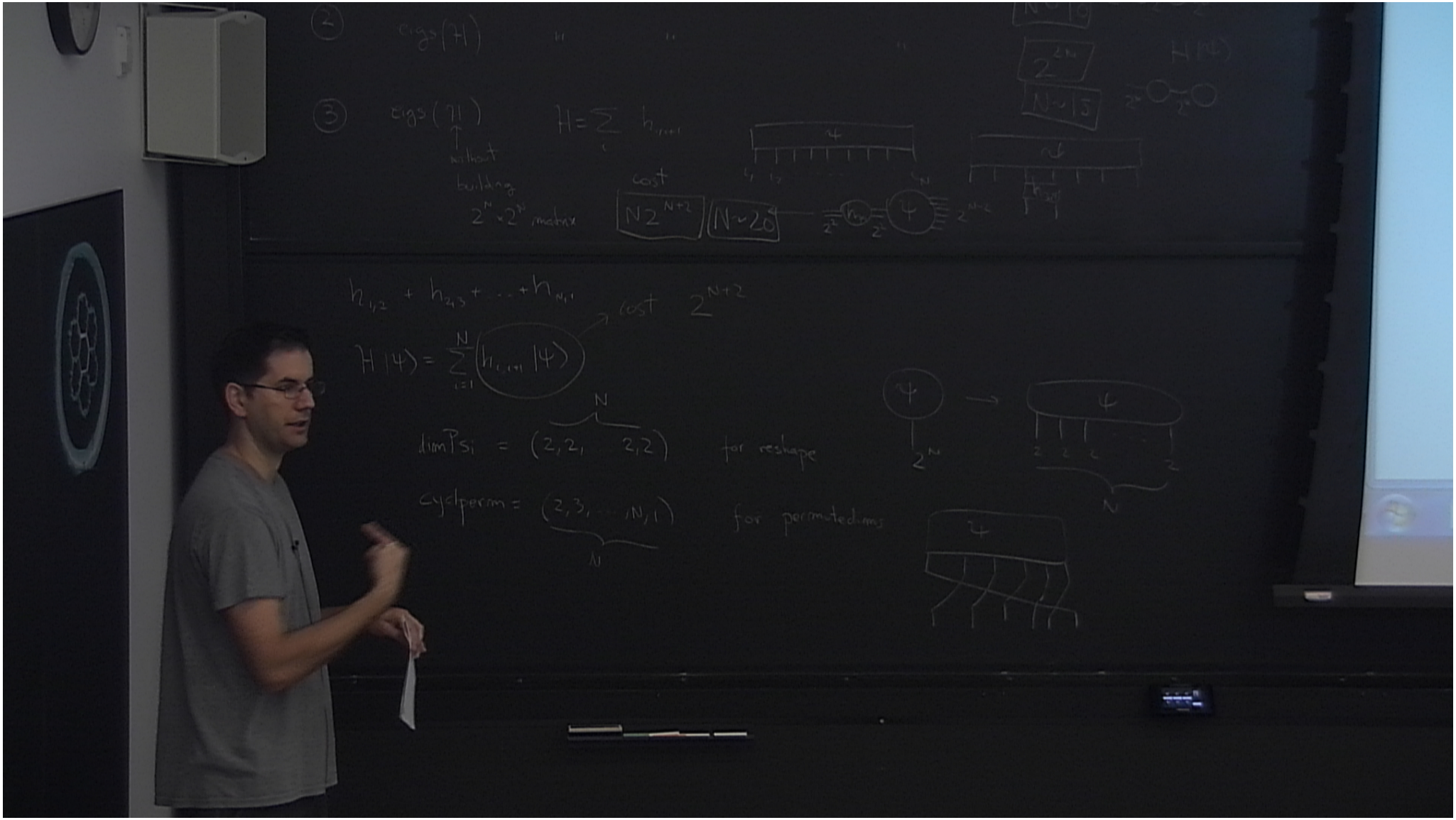

Documents/PSI 2016/
PSI2016IJuliaV
localhost:8888/notebooks/Documents/PSI%202016/PSI2016IJuliaV.ipynb
jupyter PSI2016IJuliaV Last Checkpoint: 2 hours ago (autosaved)
File Edit View Insert Cell Kernel Help
Julia 0.4.0
Code
CellToolbar
compute a sequence of Estanglement Spectra and Entanglement entropies
In [ ]:
entanglement entropy
function computeESandEE(Psi) # it assumes L = 2^N for integer N. It returns p ES and p EEs
L = length(Psi) # Dimension of the vector space
N = convert(Int64,log2(L)) # Number of sites
p = div(N,2) # N/2 or (N-1)/2
EE = zeros(p)
ES = zeros(2^p,p)
for n=1:p
A = reshape(Psi, (2^(n),2^(N-n)))
rho = A*A'
u,v = eig(rho)
u = abs(u) # to make numerical 0's positive
ES[1:2^n,n] = -sort(-u)
for m = 1:2^n
if u[m] > 0
EE[n] += -u[m]*log2(u[m])
end
end
end
return ES,EE
end
10:31 AM
29/03/2016

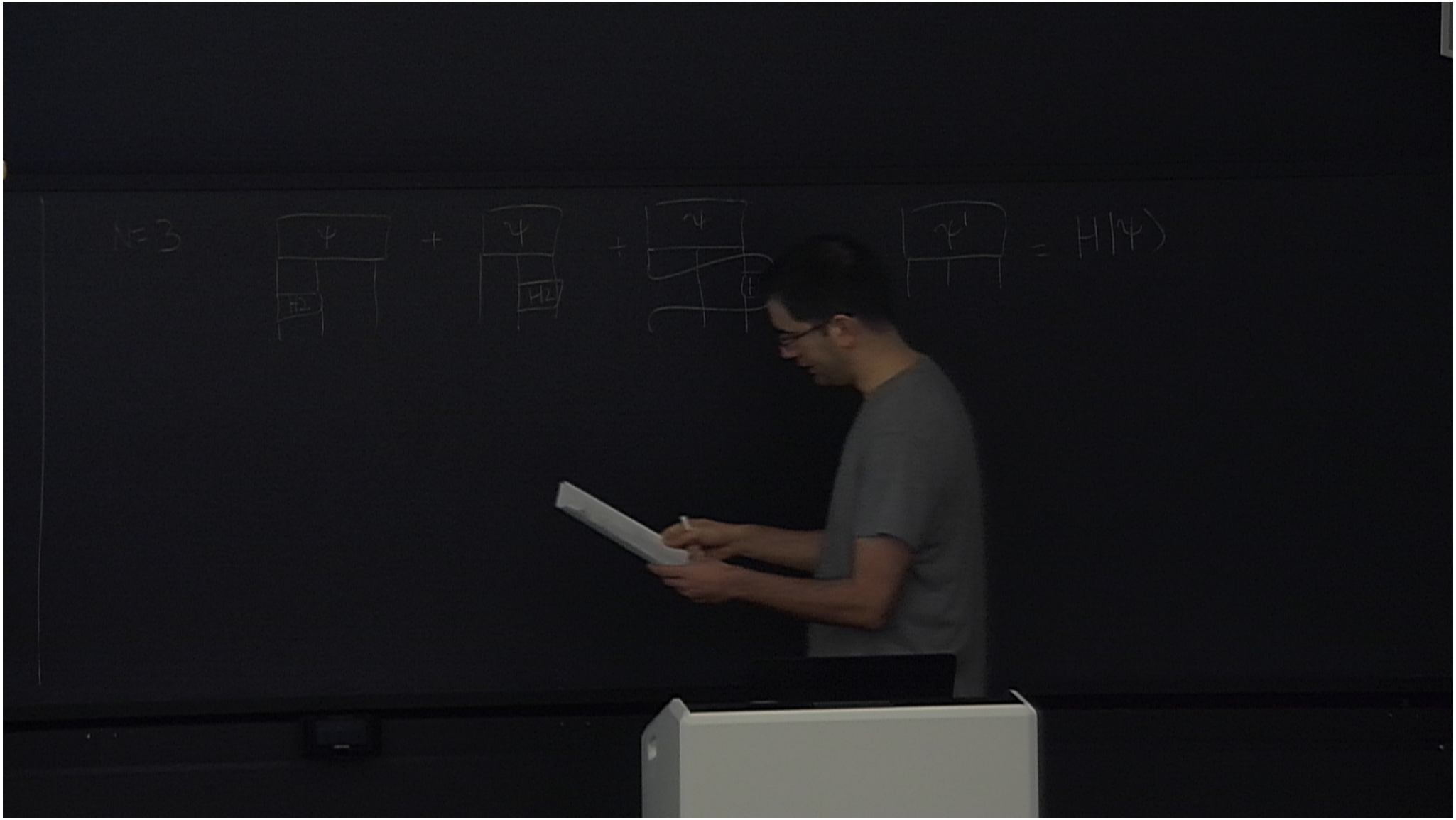

$\vec{P} = (p_1, p_2, \ldots, p_n)$
$S(\vec{P}) = S\left(\frac{1}{n}, \frac{1}{n}, \ldots, \frac{1}{n}\right) = -\frac{1}{n} \log_2 \left(\frac{1}{n}\right) \times n = \log_2 n$
$L = 3$ qubits $n = 2^L = 8$
$S(L) \leq S\left(\frac{1}{2^L}, \frac{1}{2^L}, \ldots, \frac{1}{2^L}\right) = \log_2 2^L = L$

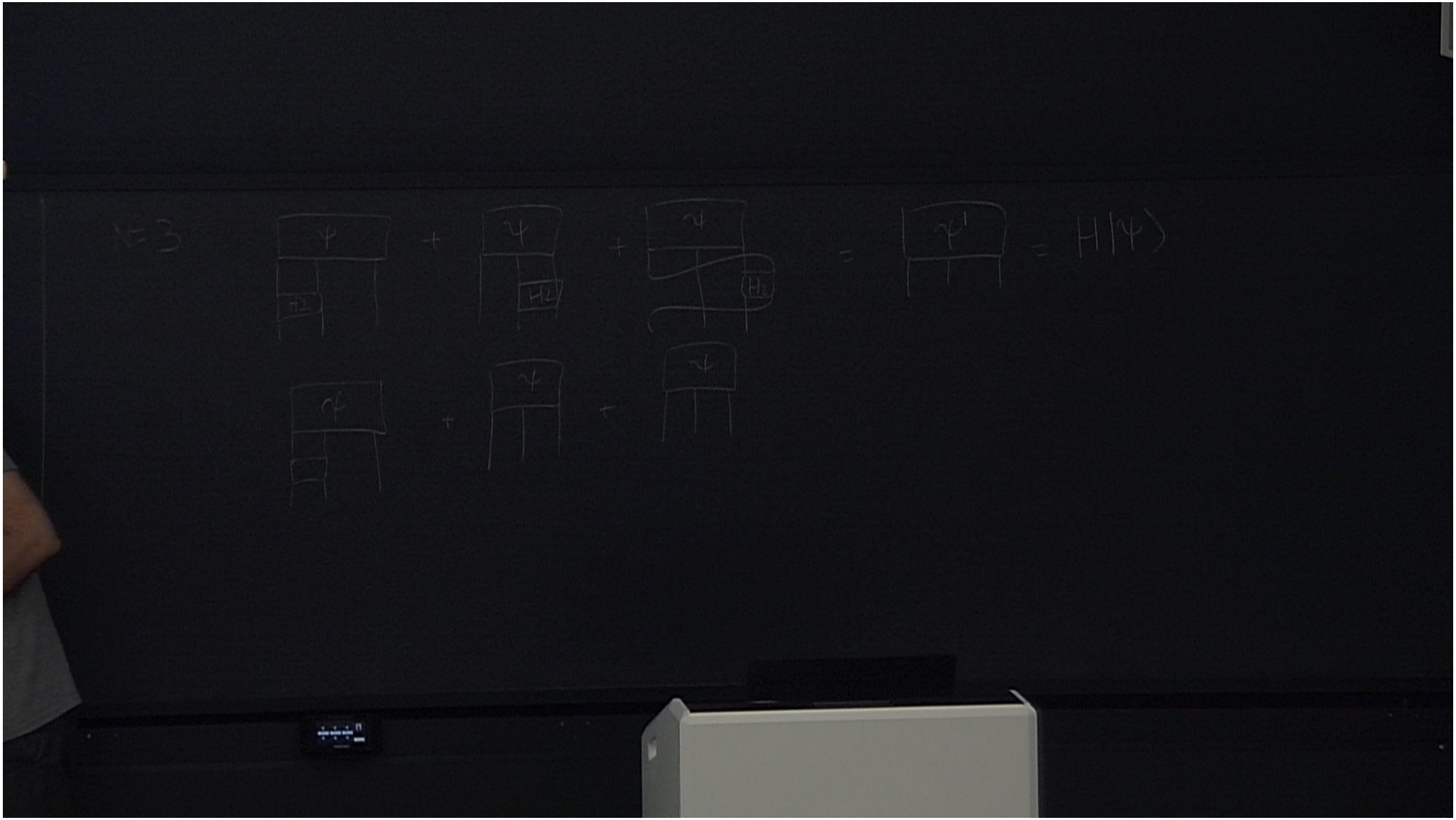


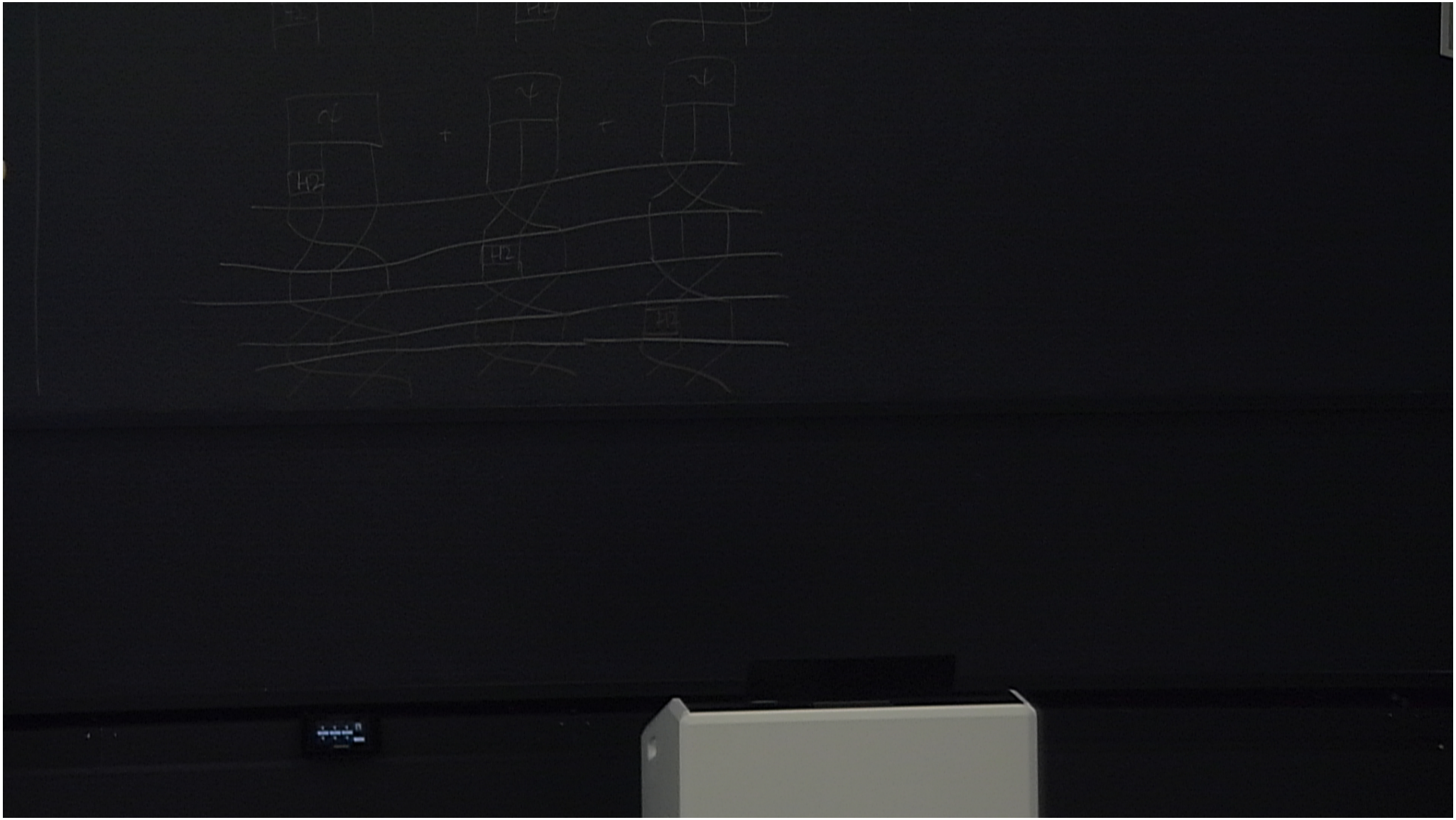

localhost:8888/notebooks/Documents/PSI%202016/PSI2016IJuliaV.ipynb
jupyter PSI2016IJuliaV Last Checkpoint: 2 hours ago (autosaved)
File Edit View Insert Cell Kernel Help
Julia 0.4.0
Code
CellToolbar

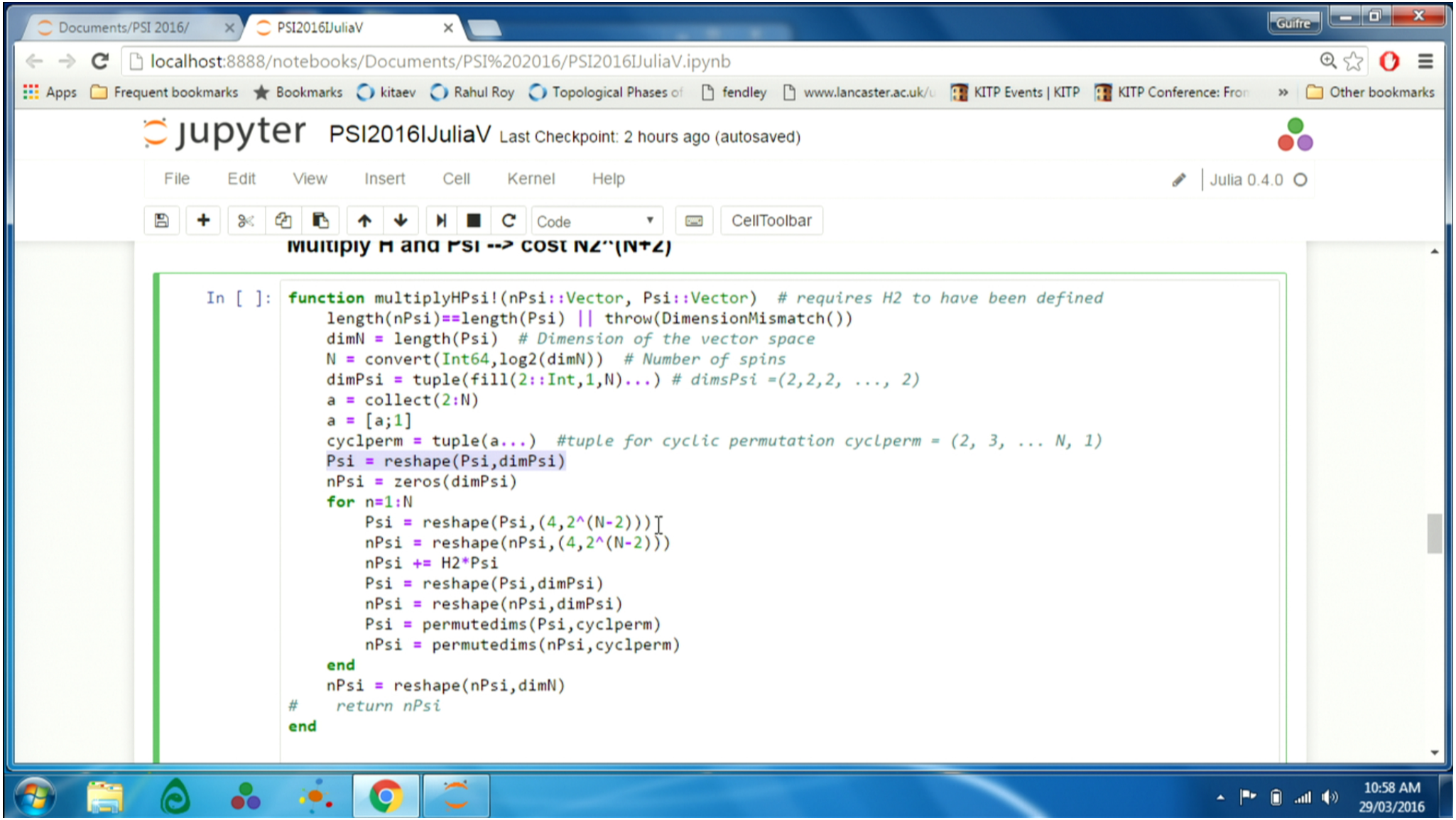

eigs(H)
eigs(H)
without building
$2^N \times 2^N$ matrix
$H = \sum_i h_{i,i+1}$
cost
$N2^{N+2}$
$N \sim 20$
$h_{1,2} + h_{2,3} + \ldots + h_{N,1}$
cost 2^{N+2}
$\sum_{i=1}^{N} (h_{i,i+1}|\psi\rangle)$
$N \sim 15$
2^{2N}
$\dim Psi = (2,2, \ldots, 2,2)$
N
for reshape
2^N

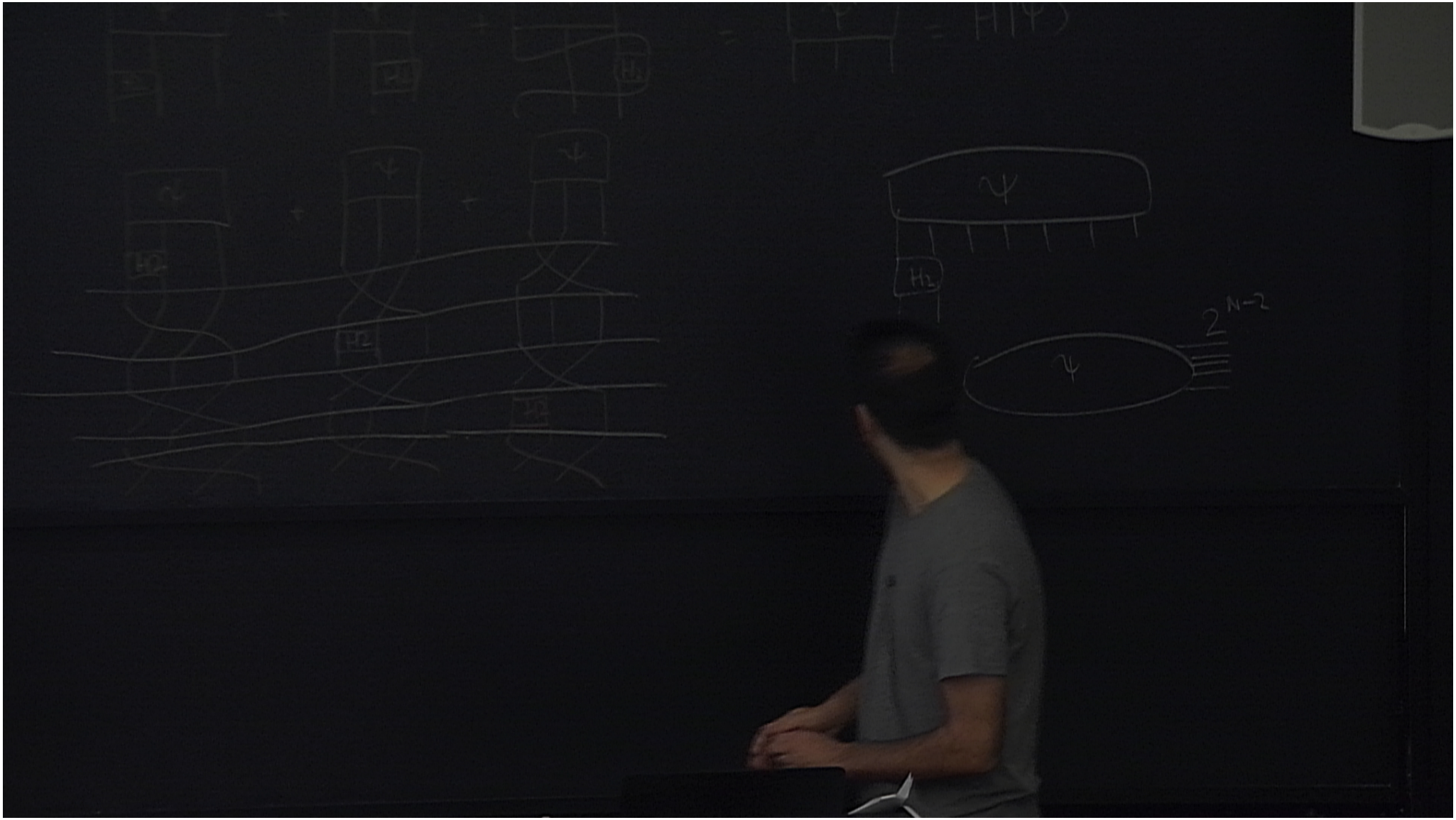

eigs(H)
eigs(H)
without building $2^N \times 2^N$ matrix
$H = \sum_i h_{i,i+1}$
cost
$N 2^{N+2}$
$N \sim 20$
$h_{1,2} + h_{2,3} + \ldots + h_{N,1}$
cost 2^{N+2}
$H|\psi\rangle = \sum_{i=1}^{N} h_{i,i+1}|\psi\rangle$
dimPsi = (2,2, ..., 2,2)
N
for reshape
cyclperm = (2,3, ..., N, 1)
N
for permutedims
2^N
N

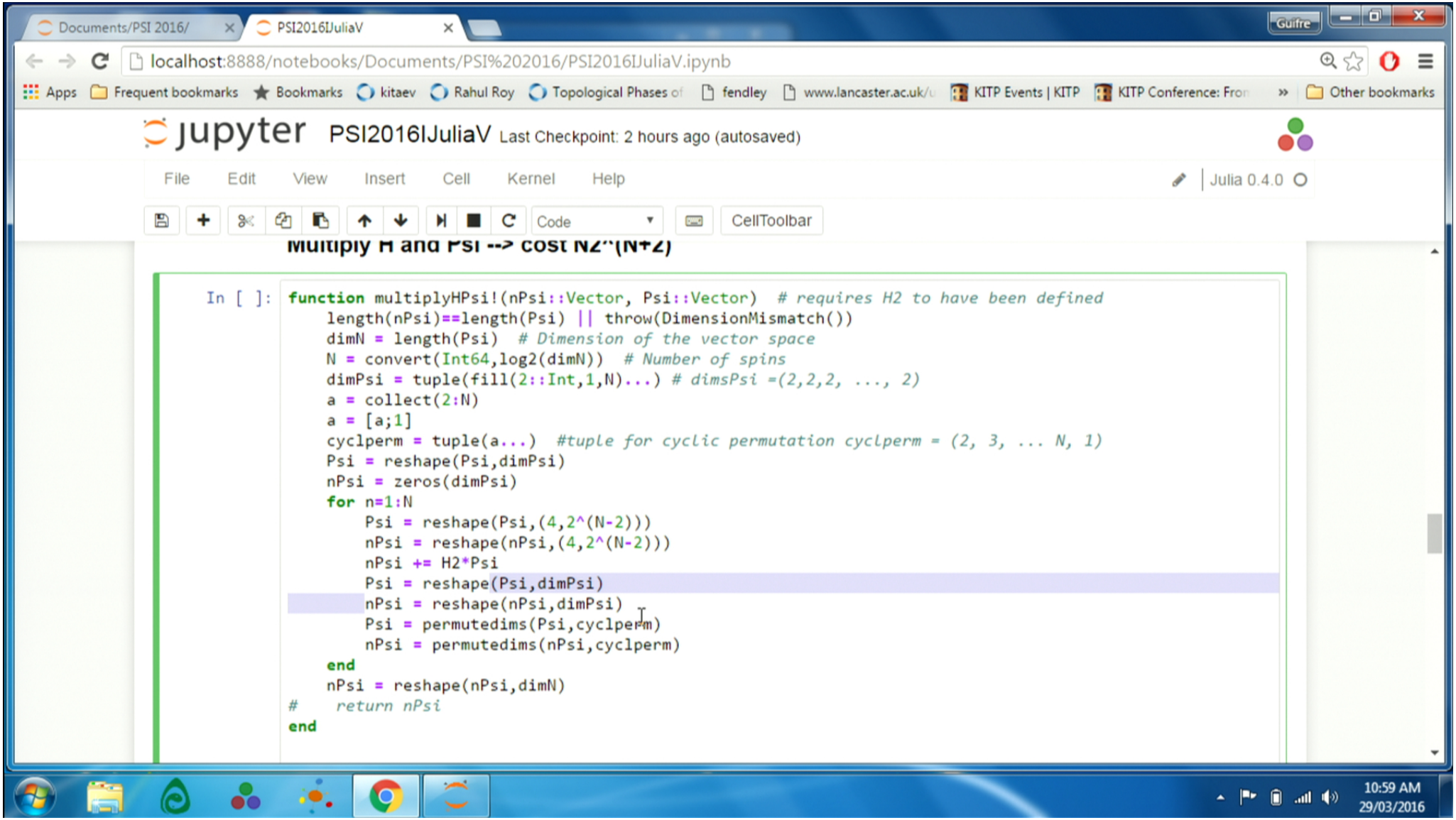

N=3
ψ
H2
+
ψ
H2
+
ψ
H
ψ'
= H|ψ⟩

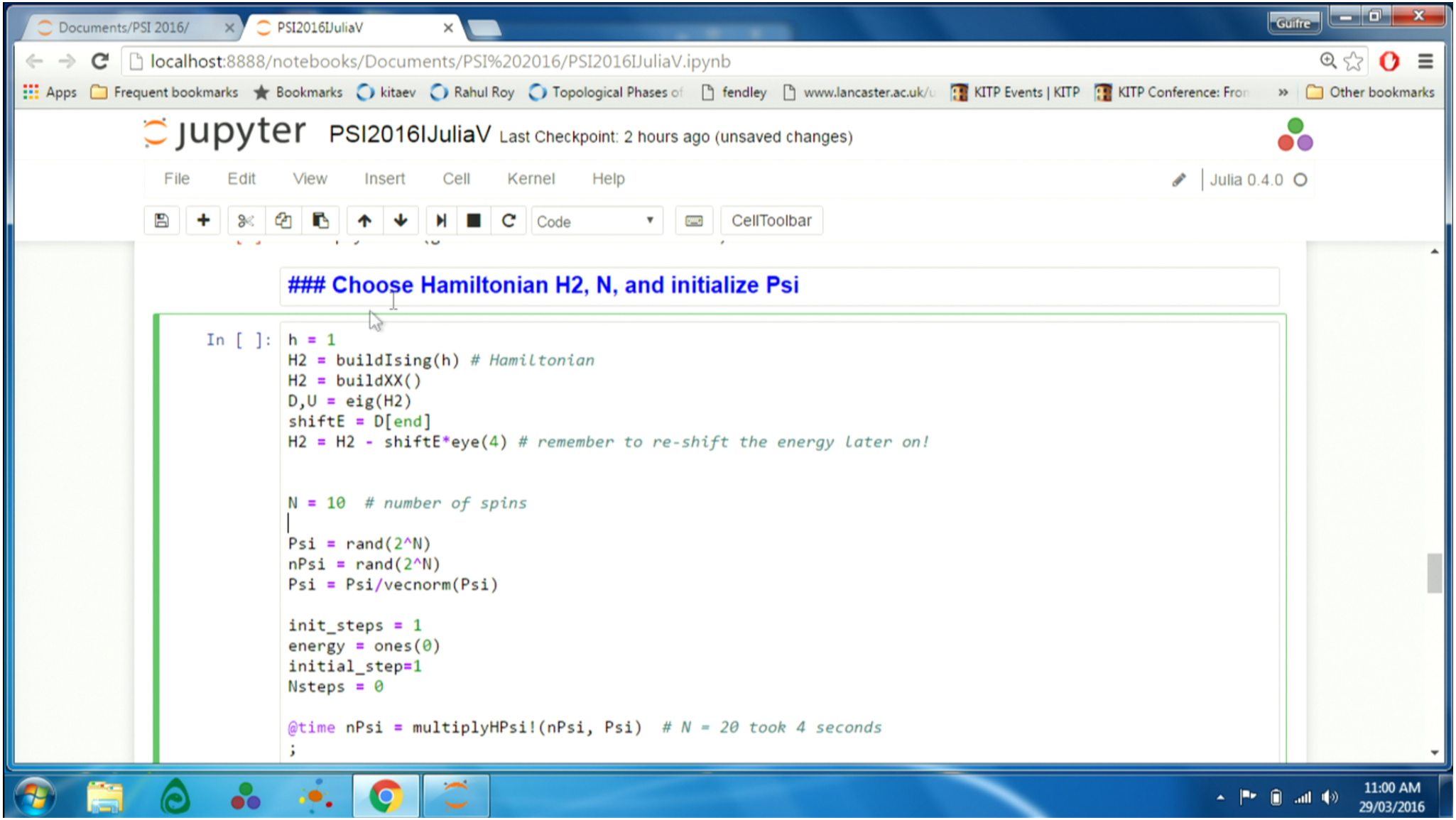

N=3
H1
H2
H1
= H|ψ⟩

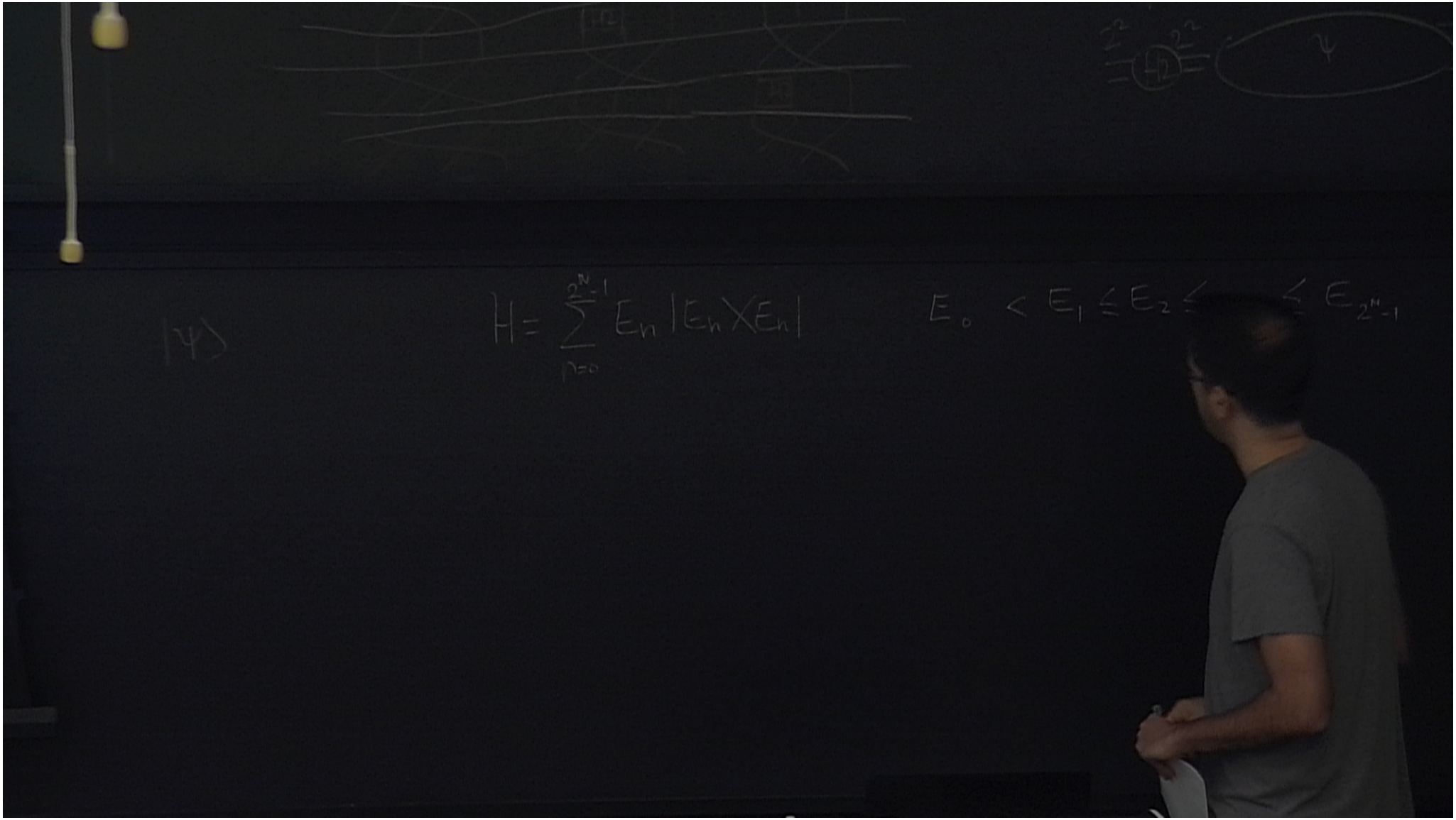


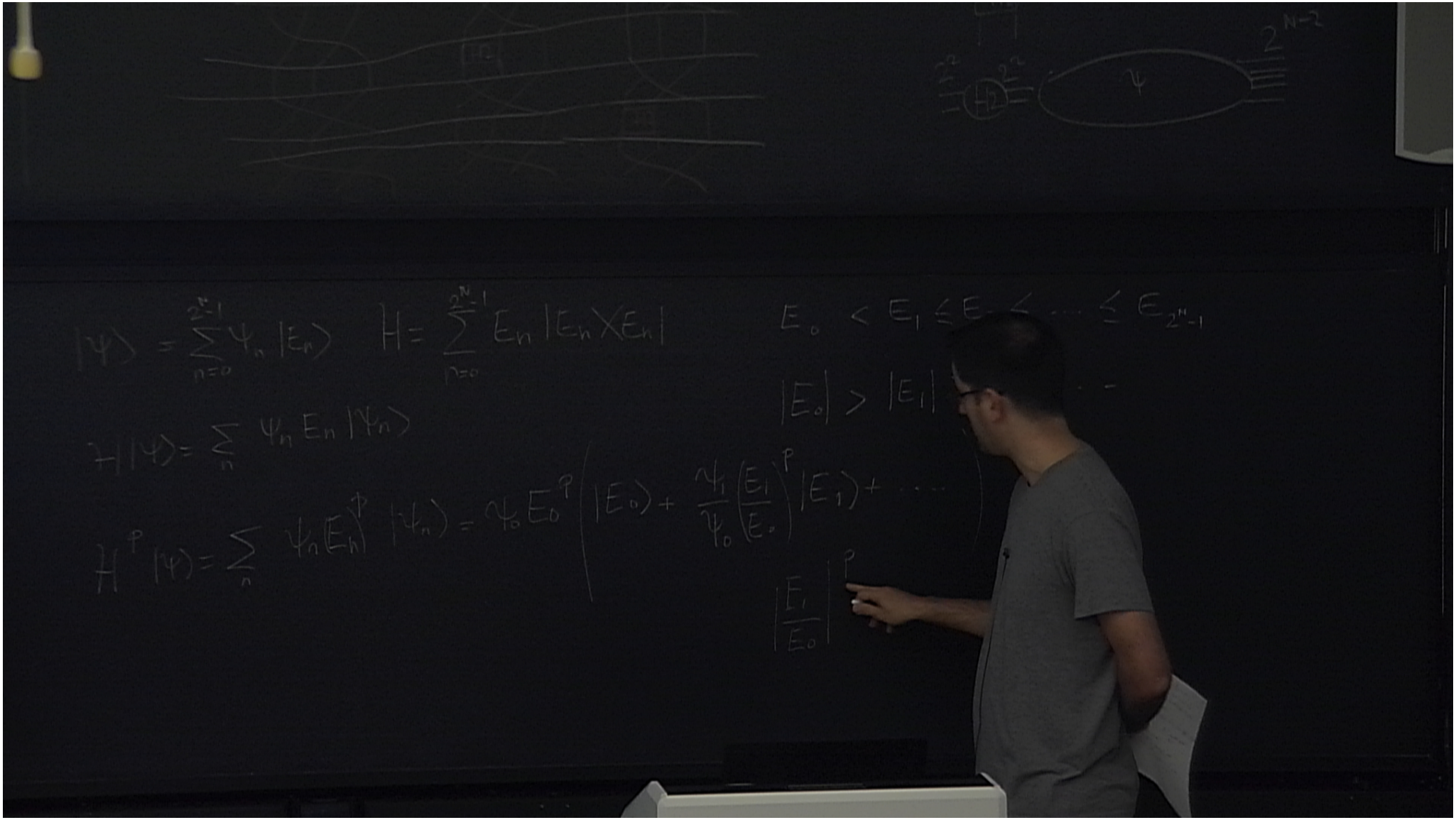

Documents/PSI 2016/
PSI2016IJuliaV
localhost:8888/notebooks/Documents/PSI%202016/PSI2016IJuliaV.ipynb
jupyter PSI2016IJuliaV Last Checkpoint: 2 hours ago (autosaved)
File Edit View Insert Cell Kernel Help
Julia 0.4.0
Code
CellToolbar
Multiply H and Psi --> cost N2^(N+2)
In [ ]:
function multiplyHPsi!(nPsi::Vector, Psi::Vector) # requires H2 to have been defined
length(nPsi)==length(Psi) || throw(DimensionMismatch())
dimN = length(Psi) # Dimension of the vector space
N = convert(Int64,log2(dimN)) # Number of spins
dimPsi = tuple(fill(2::Int,1,N)...) # dimsPsi =(2,2,2, ..., 2)
a = collect(2:N)
a = [a;1]
cyclperm = tuple(a...) #tuple for cyclic permutation cyclperm = (2, 3, ... N, 1)
Psi = reshape(Psi,dimPsi)
nPsi = zeros(dimPsi)
for n=1:N
Psi = reshape(Psi,(4,2^(N-2)))
nPsi = reshape(nPsi,(4,2^(N-2)))
nPsi += H2*Psi
Psi = reshape(Psi,dimPsi)
nPsi = reshape(nPsi,dimPsi)
Psi = permutedims(Psi,cyclperm)
nPsi = permutedims(nPsi,cyclperm)
end
nPsi = reshape(nPsi,dimN)
# return nPsi
end
10:58 AM
29/03/2016

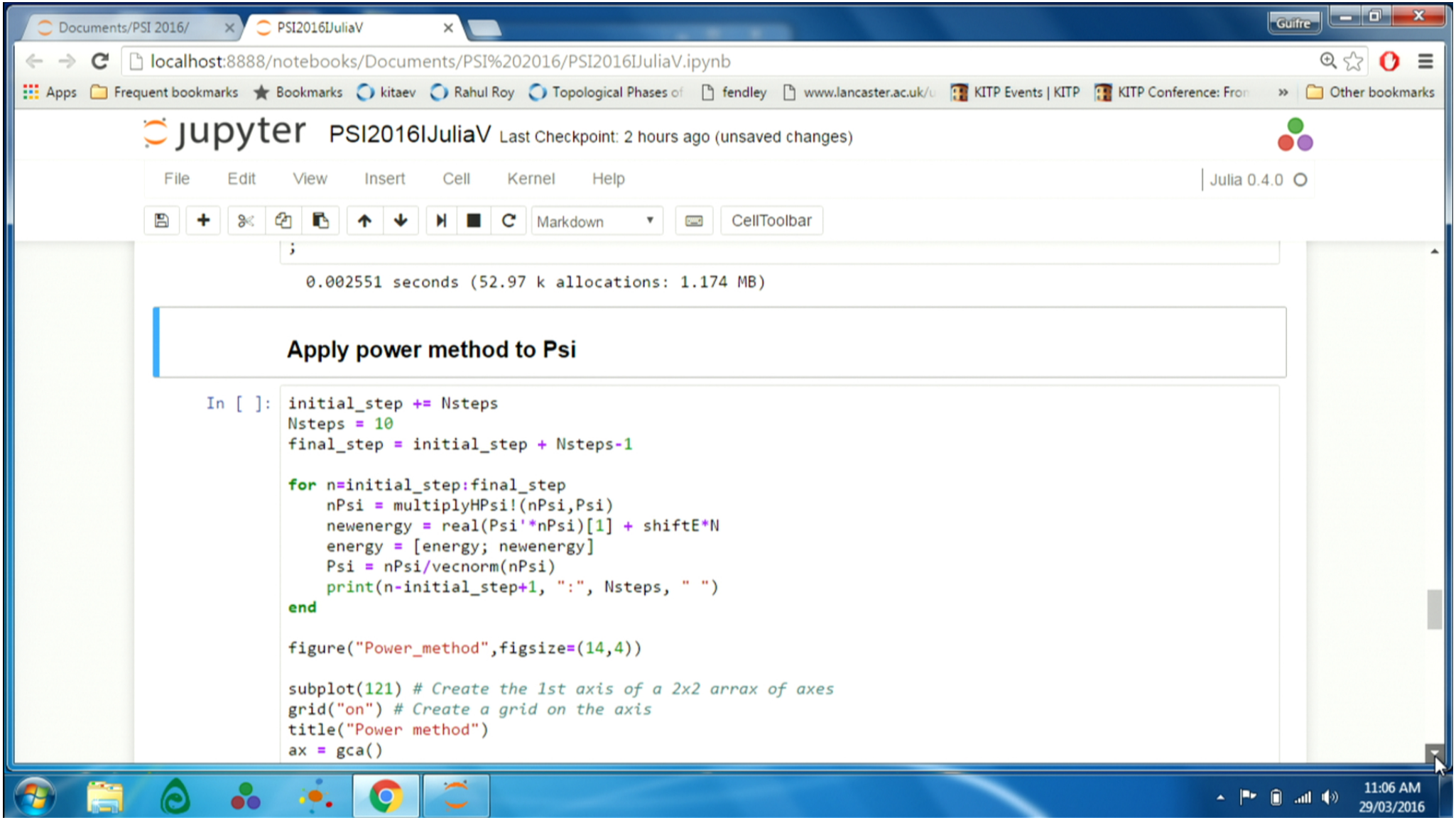

ψ
H₂
2^{N-2}
ψ

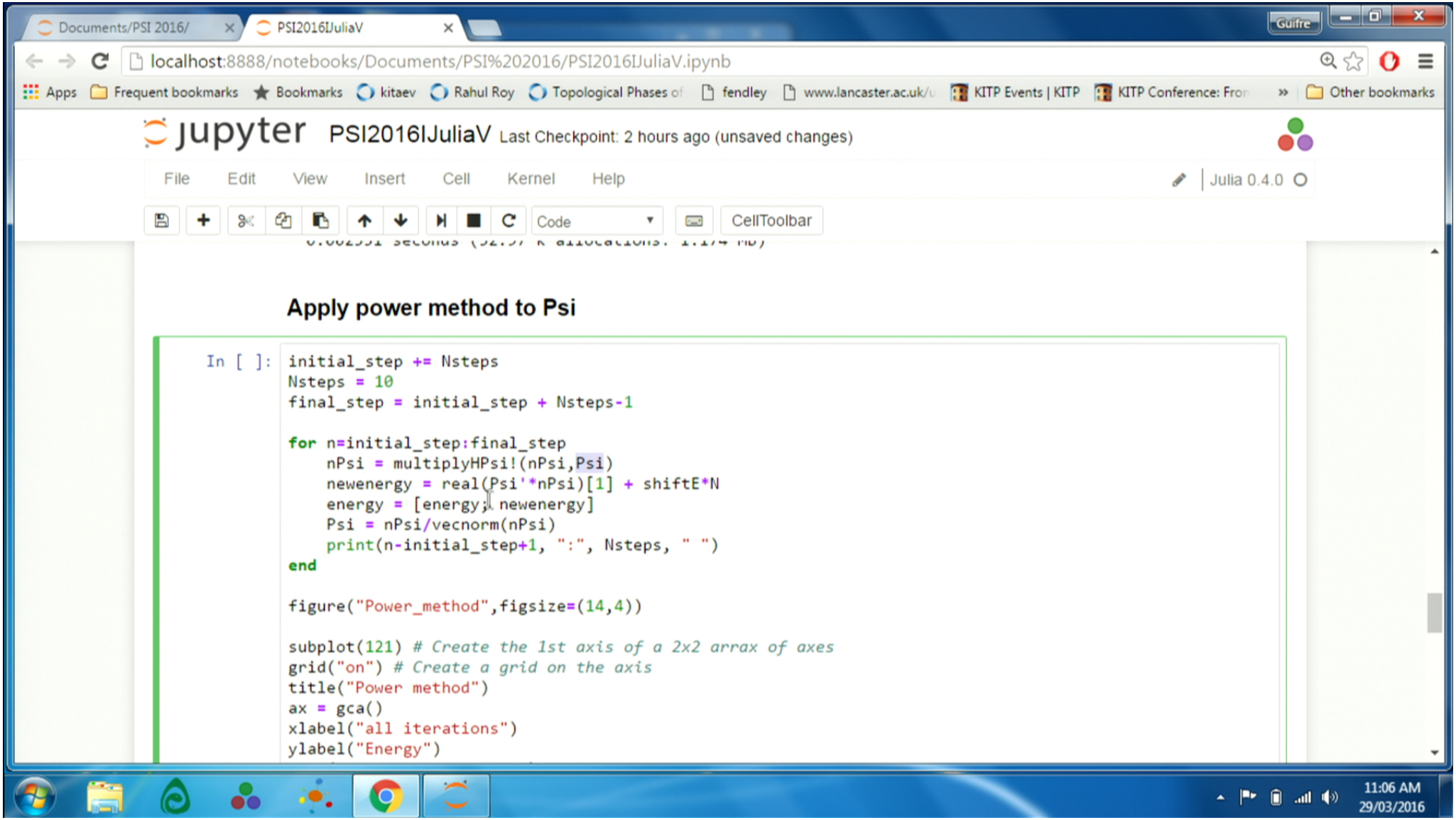

Documents/PSI 2016/
PSI2016IJuliaV
localhost:8888/notebooks/Documents/PSI%202016/PSI2016IJuliaV.ipynb
jupyter PSI2016IJuliaV Last Checkpoint: 2 hours ago (autosaved)
File Edit View Insert Cell Kernel Help
Julia 0.4.0
Code
CellToolbar
Multiply H and Psi --> cost N2^(N+2)
In [ ]:
function multiplyHPsi!(nPsi::Vector, Psi::Vector) # requires H2 to have been defined
length(nPsi)==length(Psi) || throw(DimensionMismatch())
dimN = length(Psi) # Dimension of the vector space
N = convert(Int64,log2(dimN)) # Number of spins
dimPsi = tuple(fill(2::Int,1,N)...) # dimsPsi =(2,2,2, ..., 2)
a = collect(2:N)
a = [a;1]
cyclperm = tuple(a...) #tuple for cyclic permutation cyclperm = (2, 3, ... N, 1)
Psi = reshape(Psi,dimPsi)
nPsi = zeros(dimPsi)
for n=1:N
Psi = reshape(Psi,(4,2^(N-2)))
nPsi = reshape(nPsi,(4,2^(N-2)))
nPsi += H2*Psi
Psi = reshape(Psi,dimPsi)
nPsi = reshape(nPsi,dimPsi)
Psi = permutedims(Psi,cyclperm)
nPsi = permutedims(nPsi,cyclperm)
end
nPsi = reshape(nPsi,dimN)
# return nPsi
end
10:59 AM
29/03/2016


Documents/PSI 2016/
PSI2016IJuliaV
localhost:8888/notebooks/Documents/PSI%202016/PSI2016IJuliaV.ipynb
jupyter PSI2016IJuliaV Last Checkpoint: 2 hours ago (unsaved changes)
File Edit View Insert Cell Kernel Help
Julia 0.4.0
Code
CellToolbar
Choose Hamiltonian H2, N, and initialize Psi
In [ ]:
h = 1
H2 = buildIsing(h) # Hamiltonian
H2 = buildXX()
D,U = eig(H2)
shiftE = D[end]
H2 = H2 - shiftE*eye(4) # remember to re-shift the energy later on!
N = 10 # number of spins
Psi = rand(2^N)
nPsi = rand(2^N)
Psi = Psi/vecnorm(Psi)
init_steps = 1
energy = ones(0)
initial_step=1
Nsteps = 0
@time nPsi = multiplyHPsi!(nPsi, Psi) # N = 20 took 4 seconds
;
11:00 AM
29/03/2016


$|\psi\rangle$
$H = \sum_{n=0}^{2^N-1} E_n |E_n X E_n|$
$E_0 < E_1 \leq E_2 \leq \cdots \leq E_{2^N-1}$


$|\psi\rangle = \sum_{n=0}^{2^N-1} \psi_n |E_n\rangle$
$H = \sum_{n=0}^{2^N-1} E_n |E_n\rangle\langle E_n|$
$E_0 < E_1 \leq E_2 \leq \cdots \leq E_{2^N-1}$
$|E_0| > |E_1|$
$H|\psi\rangle = \sum_n \psi_n E_n |\psi_n\rangle$
$H^p|\psi\rangle = \sum_n \psi_n (E_n)^p |\psi_n\rangle = \psi_0 E_0^p \left(|E_0\rangle + \frac{\psi_1}{\psi_0}\left(\frac{E_1}{E_0}\right)^p |E_1\rangle + \cdots \right)$
$\left|\frac{E_1}{E_0}\right|^p$
2^{N-2}
ψ


Documents/PSI 2016/
PSI2016IJuliaV
localhost:8888/notebooks/Documents/PSI%202016/PSI2016IJuliaV.ipynb
jupyter PSI2016IJuliaV Last Checkpoint: 2 hours ago (unsaved changes)
File Edit View Insert Cell Kernel Help
Julia 0.4.0
Markdown
CellToolbar
;
0.002551 seconds (52.97 k allocations: 1.174 MB)
Apply power method to Psi
In [ ]:
initial_step += Nsteps
Nsteps = 10
final_step = initial_step + Nsteps-1
for n=initial_step:final_step
nPsi = multiplyHPsi!(nPsi,Psi)
newenergy = real(Psi'*nPsi)[1] + shiftE*N
energy = [energy; newenergy]
Psi = nPsi/vecnorm(nPsi)
print(n-initial_step+1, ":", Nsteps, " ")
end
figure("Power_method",figsize=(14,4))
subplot(121) # Create the 1st axis of a 2x2 arrax of axes
grid("on") # Create a grid on the axis
title("Power method")
ax = gca()
11:06 AM
29/03/2016


Documents/PSI 2016/
PSI2016IJuliaV
localhost:8888/notebooks/Documents/PSI%202016/PSI2016IJuliaV.ipynb
jupyter PSI2016IJuliaV Last Checkpoint: 2 hours ago (unsaved changes)
File Edit View Insert Cell Kernel Help
Julia 0.4.0
Code
CellToolbar
Apply power method to Psi
In [ ]:
initial_step += Nsteps
Nsteps = 10
final_step = initial_step + Nsteps-1
for n=initial_step:final_step
nPsi = multiplyHPsi!(nPsi,Psi)
newenergy = real(Psi'*nPsi)[1] + shiftE*N
energy = [energy; newenergy]
Psi = nPsi/vecnorm(nPsi)
print(n-initial_step+1, ":", Nsteps, " ")
end
figure("Power_method",figsize=(14,4))
subplot(121) # Create the 1st axis of a 2x2 arrax of axes
grid("on") # Create a grid on the axis
title("Power method")
ax = gca()
xlabel("all iterations")
ylabel("Energy")
11:06 AM
29/03/2016

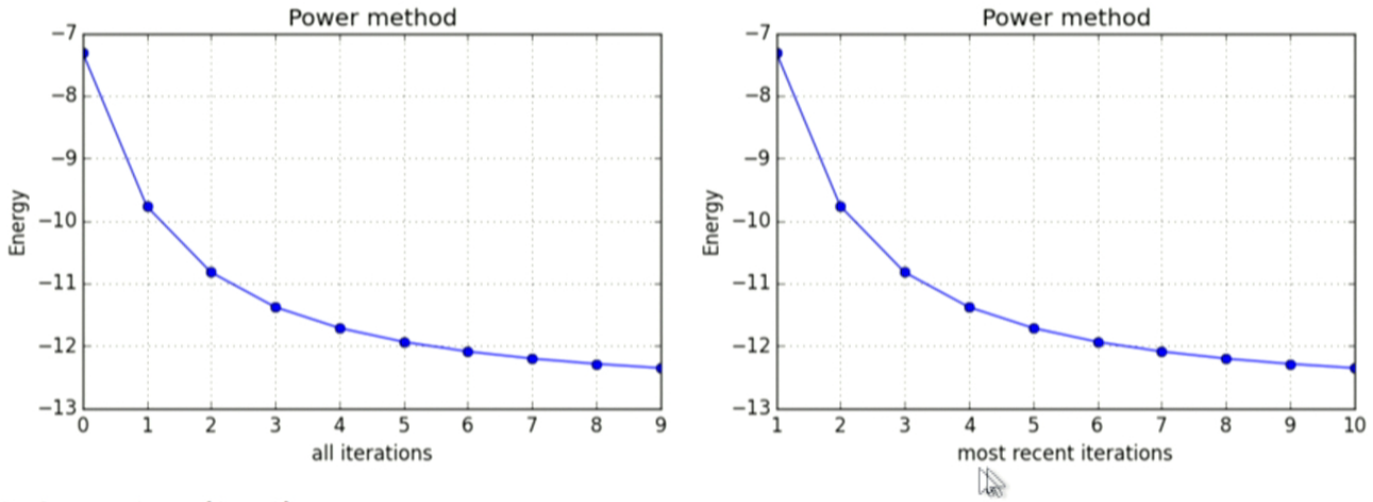
jupyter PSI2016IJuliaV Last Checkpoint: 2 hours ago (unsaved changes)

File Edit View Insert Cell Kernel Help Julia 0.4.0

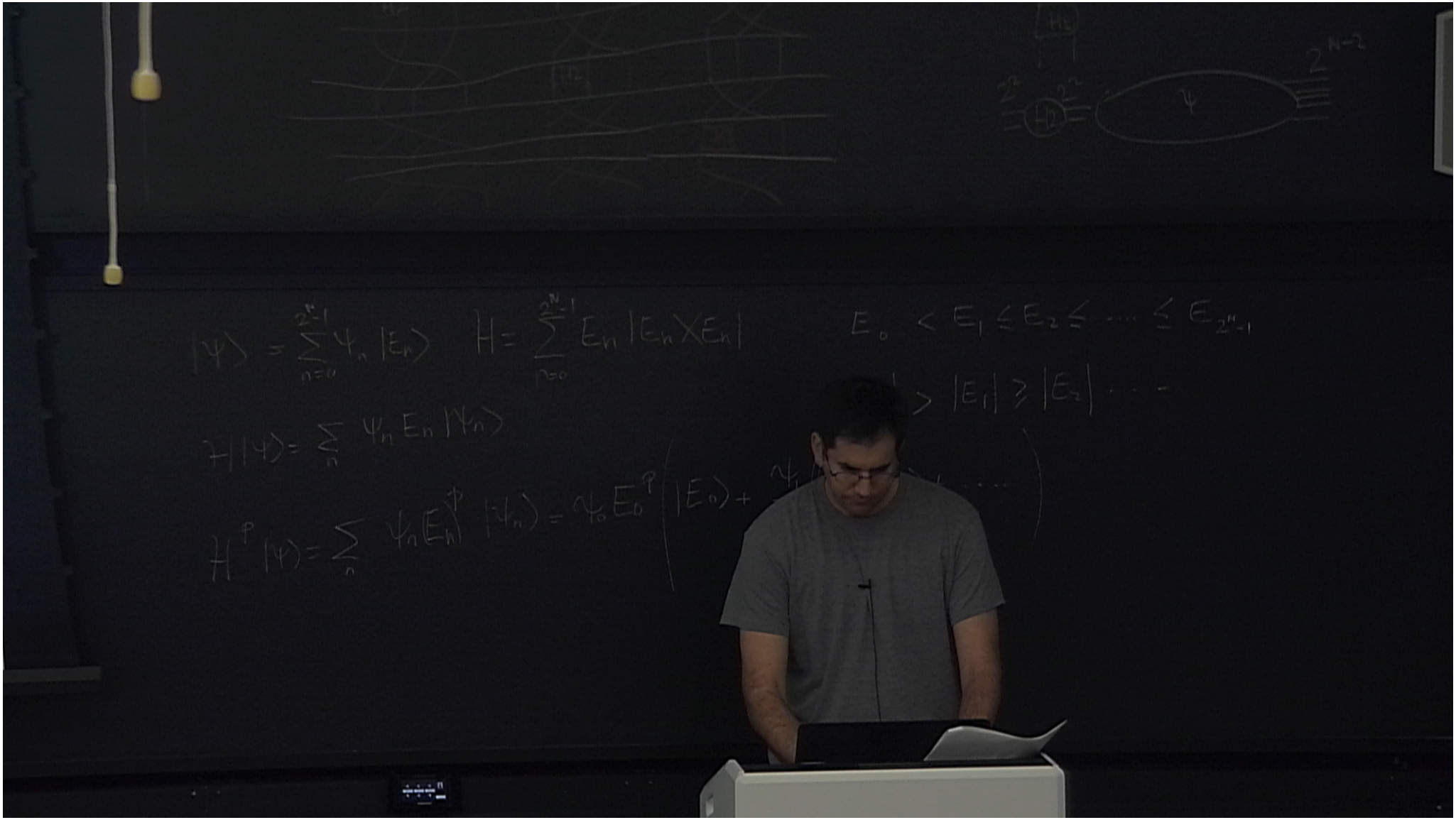
Markdown CellToolbar

```
ylabel("Energy")
plot(initial_step:final_step, energy[initial_step:final_step], marker = "o")
```

1



Out[12]:
```
1-element Array{Any,1}:
 PyObject <matplotlib.lines.Line2D object at 0x000000000F202320>
```


$|\psi\rangle = \sum_{n=0}^{2^N-1} \psi_n |E_n\rangle$
$H = \sum_{n=0}^{2^N-1} E_n |E_n\rangle\langle E_n|$
$E_0 < E_1 \leq E_2 \leq \cdots \leq E_{2^N-1}$
$H|\psi\rangle = \sum_n \psi_n E_n |\psi_n\rangle$
$H^p|\psi\rangle = \sum_n \psi_n (E_n)^p |\psi_n\rangle = \psi_0 E_0^p \left(|E_0\rangle + \right.$
$\left. \cdots \right)$
$|E_1| \geq |E_2|$
H_2
H_2
H_2
ψ
2^{N-2}

Documents/PSI 2016/ | PSI2016IJuliaV

localhost:8888/notebooks/Documents/PSI%202016/PSI2016IJuliaV.ipynb

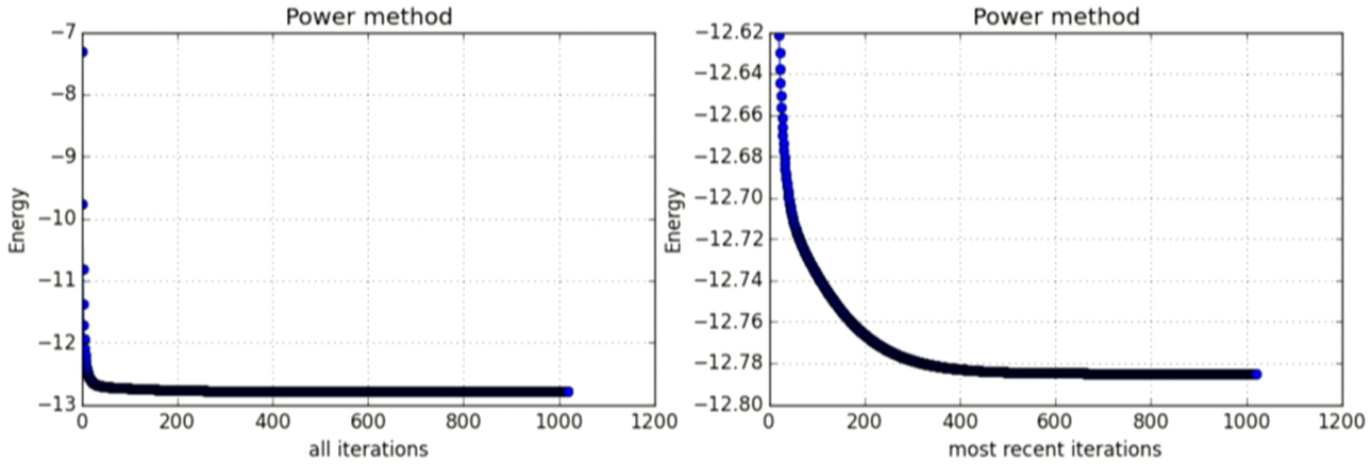
Apps | Frequent bookmarks | Bookmarks | kitaev | Rahul Roy | Topological Phases of | fendley | www.lancaster.ac.uk/ | KITP Events | KITP | KITP Conference: Fron | Other bookmarks

jupyter PSI2016IJuliaV Last Checkpoint: 2 hours ago (unsaved changes)

File Edit View Insert Cell Kernel Help Julia 0.4.0

Markdown CellToolbar

```
1000 969:1000 970:1000 971:1000 972:1000 973:1000 974:1000 975:1000 976:1000 977:1000 978:1000 979:100
0 920:1000 921:1000 922:1000 923:1000 924:1000 925:1000 926:1000 927:1000 928:1000 929:1000 930:1000 9
31:1000 932:1000 933:1000 934:1000 935:1000 936:1000 937:1000 938:1000 939:1000 940:1000 941:1000 942:
1000 943:1000 944:1000 945:1000 946:1000 947:1000 948:1000 949:1000 950:1000 951:1000 952:1000 953:100
0 954:1000 955:1000 956:1000 957:1000 958:1000 959:1000 960:1000 961:1000 962:1000 963:1000 964:1000 9
65:1000 966:1000 967:1000 968:1000 969:1000 970:1000 971:1000 972:1000
```



Out[14]:
```
1-element Array{Any,1}:
 PyObject <matplotlib.lines.Line2D object at 0x0000000020A0C080>
```

11:08 AM 29/03/2016

Documents/PSI 2016/ | PSI2016IJuliaV

localhost:8888/notebooks/Documents/PSI%202016/PSI2016IJuliaV.ipynb

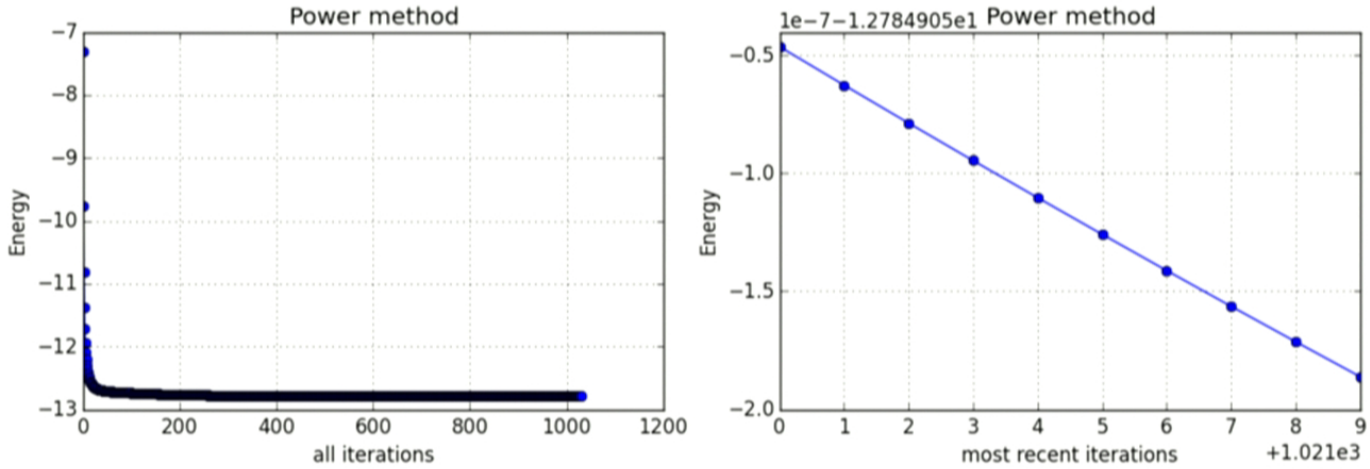
jupyter PSI2016IJuliaV Last Checkpoint: 2 hours ago (unsaved changes)

File Edit View Insert Cell Kernel Help Julia 0.4.0

Markdown CellToolbar

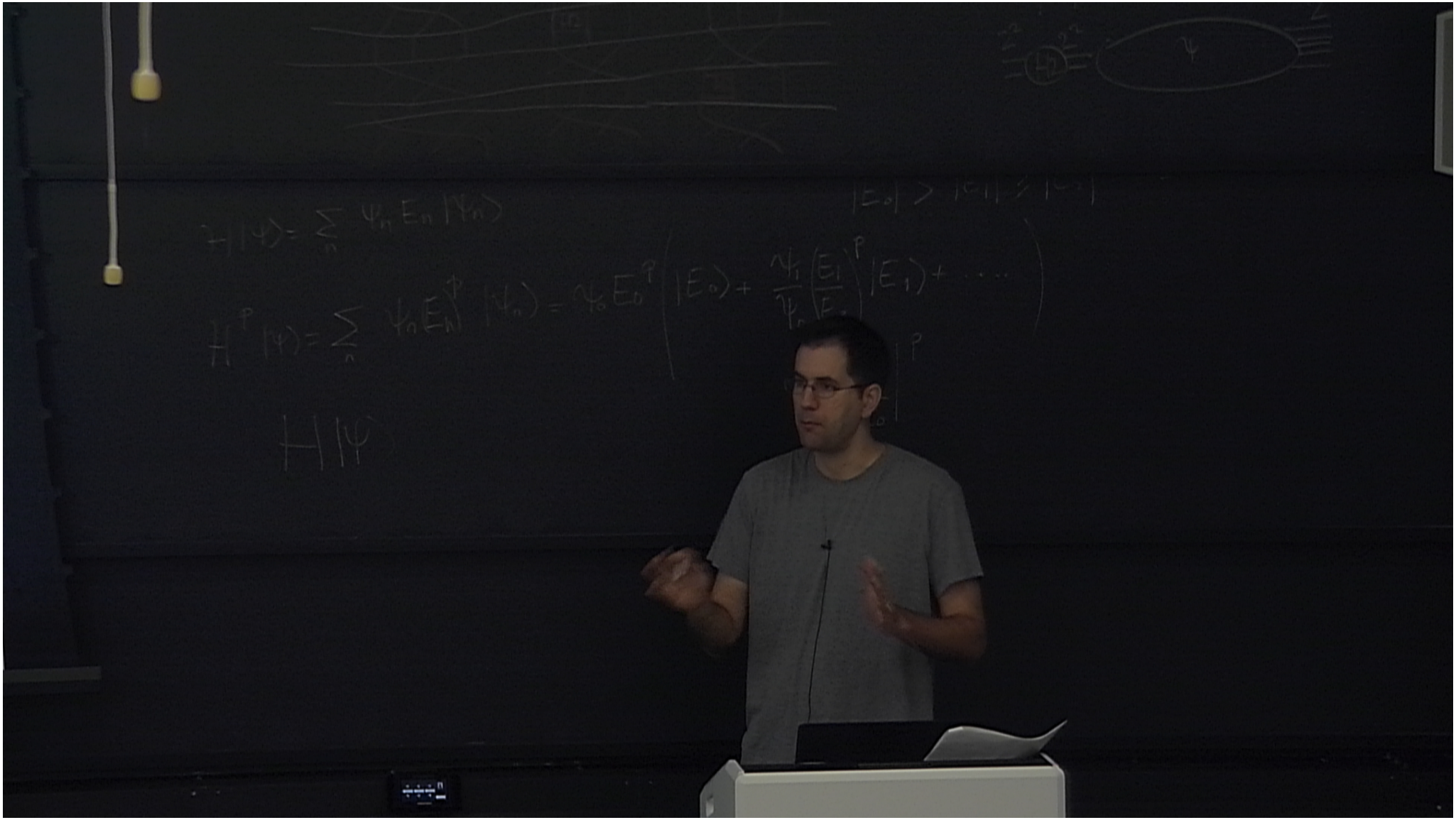
```
plot(initial_step:final_step, energy[initial_step:final_step], marker = "o")
```

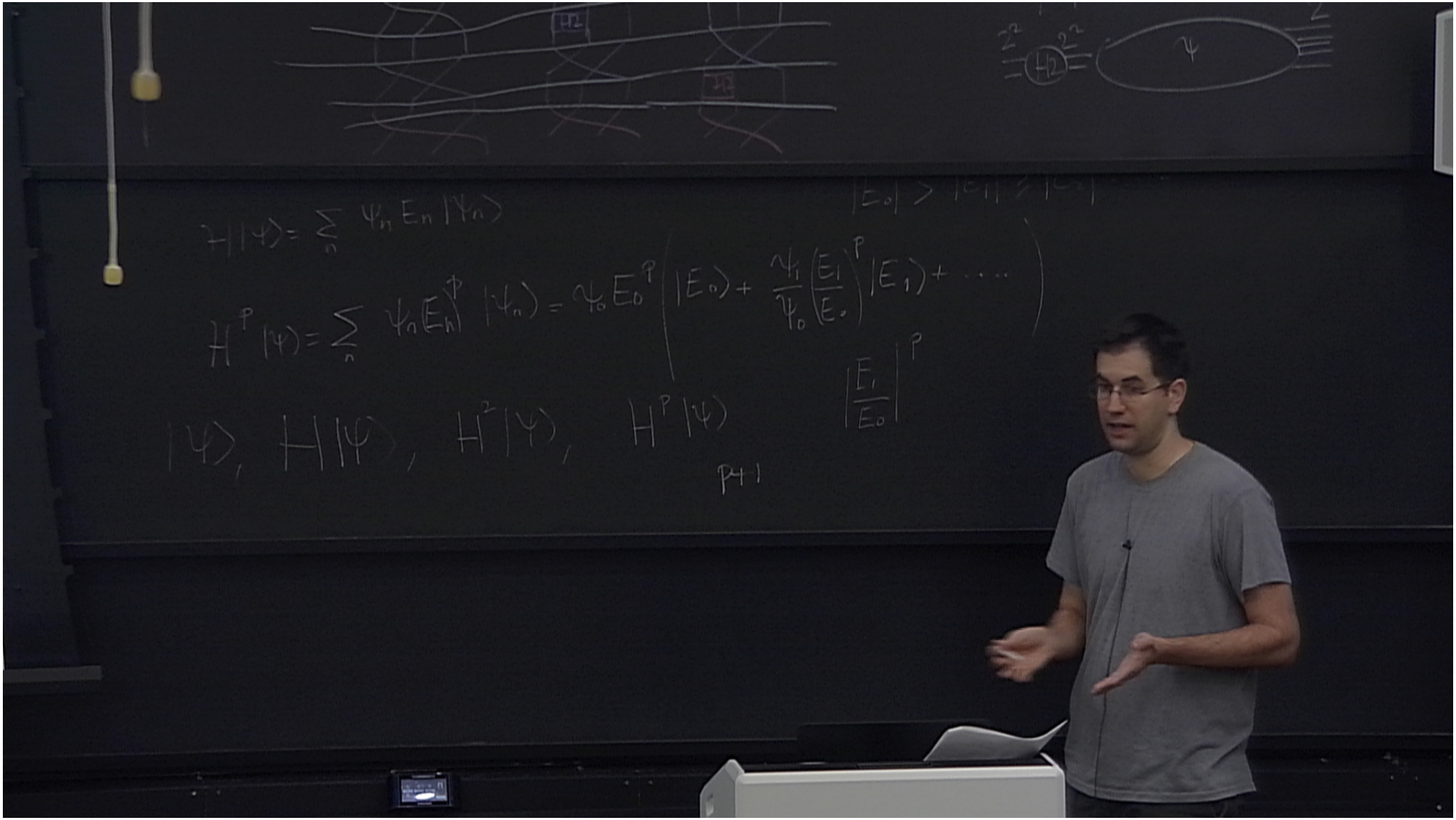
1

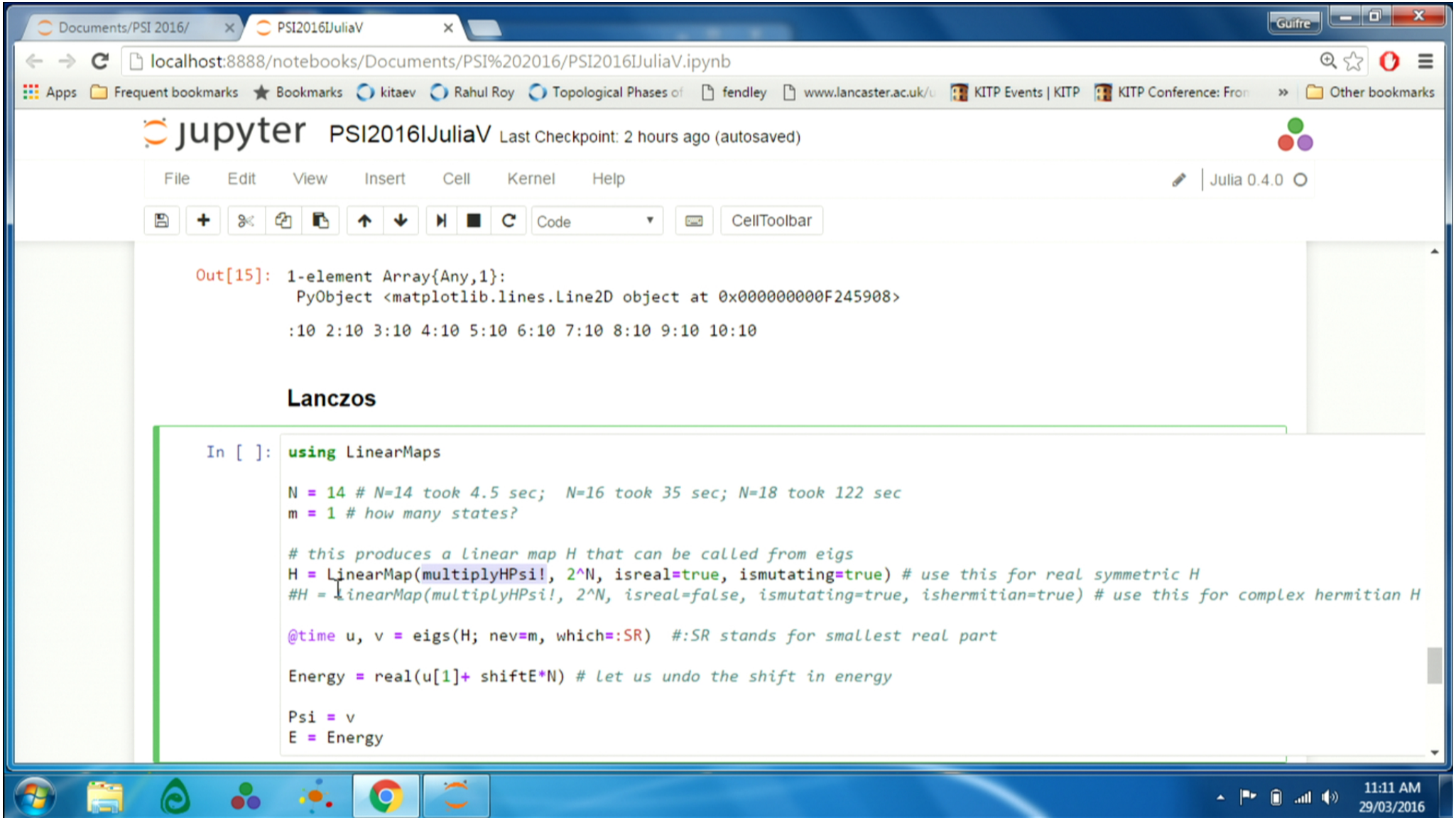


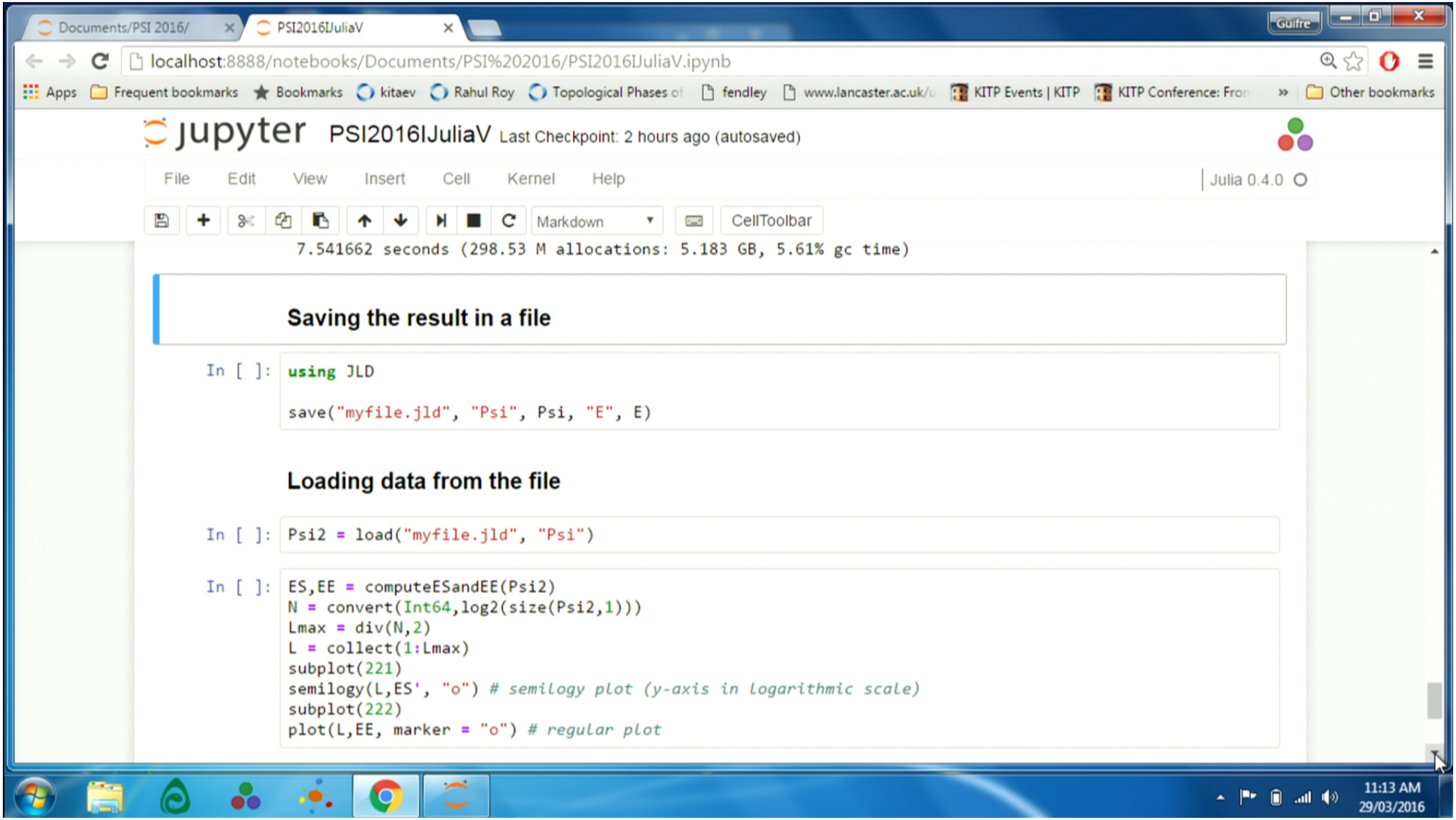
Out[15]:
```
1-element Array{Any,1}:
 PyObject <matplotlib.lines.Line2D object at 0x000000000F245908>
```

11:09 AM 29/03/2016


H|ψ⟩
H^p|ψ⟩


$H|\psi\rangle = \sum_n \psi_n E_n |\psi_n\rangle$
$H^p|\psi\rangle = \sum_n \psi_n E_n^p |\psi_n\rangle = \psi_0 E_0^p \left(|E_0\rangle + \frac{\psi_1}{\psi_0} \left(\frac{E_1}{E_0}\right)^p |E_1\rangle + \cdots \right)$
$|E_0| > |E_1| \geq |E_2|$
$\left|\frac{E_1}{E_0}\right|^p$
$|\psi\rangle, H|\psi\rangle, H^2|\psi\rangle, H^p|\psi\rangle$
p+1


Documents/PSI 2016/
PSI2016IJuliaV
localhost:8888/notebooks/Documents/PSI%202016/PSI2016IJuliaV.ipynb
jupyter PSI2016IJuliaV Last Checkpoint: 2 hours ago (autosaved)
File Edit View Insert Cell Kernel Help
Julia 0.4.0
Code
CellToolbar
Out[15]: 1-element Array{Any,1}:
PyObject <matplotlib.lines.Line2D object at 0x000000000F245908>
:10 2:10 3:10 4:10 5:10 6:10 7:10 8:10 9:10 10:10
Lanczos
In [ ]:
using LinearMaps
N = 14 # N=14 took 4.5 sec; N=16 took 35 sec; N=18 took 122 sec
m = 1 # how many states?
# this produces a linear map H that can be called from eigs
H = LinearMap(multiplyHPsi!, 2^N, isreal=true, ismutating=true) # use this for real symmetric H
#H = LinearMap(multiplyHPsi!, 2^N, isreal=false, ismutating=true, ishermitian=true) # use this for complex hermitian H
@time u, v = eigs(H; nev=m, which=:SR) #:SR stands for smallest real part
Energy = real(u[1]+ shiftE*N) # let us undo the shift in energy
Psi = v
E = Energy
11:11 AM
29/03/2016


Documents/PSI 2016/
PSI2016IJuliaV
localhost:8888/notebooks/Documents/PSI%202016/PSI2016IJuliaV.ipynb
jupyter PSI2016IJuliaV Last Checkpoint: 2 hours ago (autosaved)
File Edit View Insert Cell Kernel Help
Julia 0.4.0
Markdown
CellToolbar
7.541662 seconds (298.53 M allocations: 5.183 GB, 5.61% gc time)
Saving the result in a file
In [ ]:
using JLD
save("myfile.jld", "Psi", Psi, "E", E)
Loading data from the file
In [ ]:
Psi2 = load("myfile.jld", "Psi")
In [ ]:
ES,EE = computeESandEE(Psi2)
N = convert(Int64,log2(size(Psi2,1)))
Lmax = div(N,2)
L = collect(1:Lmax)
subplot(221)
semilogy(L,ES', "o") # semilogy plot (y-axis in logarithmic scale)
subplot(222)
plot(L,EE, marker = "o") # regular plot
11:13 AM
29/03/2016


Documents/PSI 2016/
PSI2016IJuliaV
localhost:8888/notebooks/Documents/PSI%202016/PSI2016IJuliaV.ipynb
jupyter PSI2016IJuliaV Last Checkpoint: 2 hours ago (autosaved)
File Edit View Insert Cell Kernel Help
Julia 0.4.0
Code
CellToolbar
Psi = v
E = Energy
Out[17]: -17.862808540760923
7.541662 seconds (298.53 M allocations: 5.183 GB, 5.61% gc time)
Saving the result in a file
In [ ]: using JLD
save("myfile.jld", "Psi", Psi, "E", E)
Loading data from the file
In [ ]: Psi2 = load("myfile.jld", "Psi")
In [ ]: ES,EE = computeESandEE(Psi2)
11:13 AM
29/03/2016

Documents/PSI 2016/ | PSI2016IJuliaV

localhost:8888/notebooks/Documents/PSI%202016/PSI2016IJuliaV.ipynb

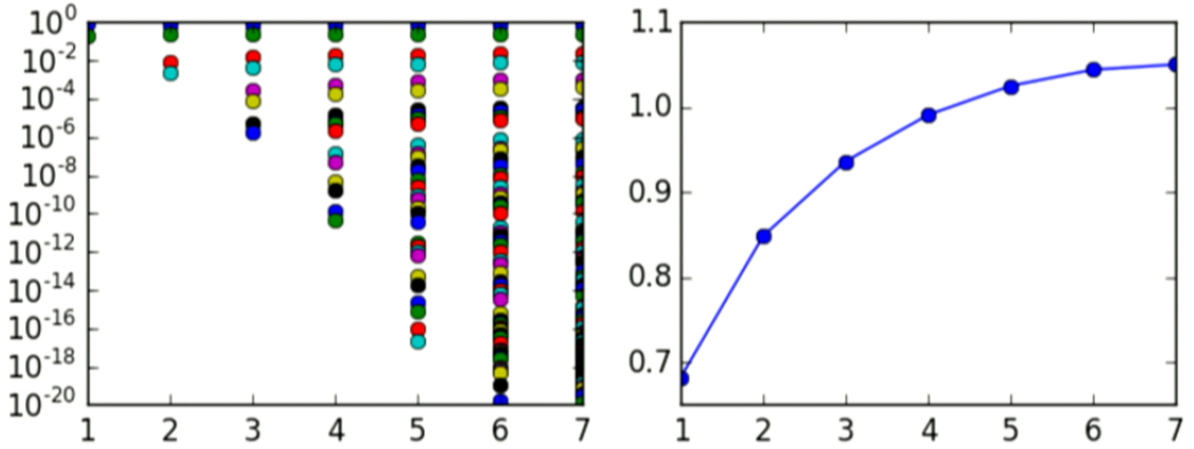
Apps | Frequent bookmarks | Bookmarks | kitaev | Rahul Roy | Topological Phases of | fendley | www.lancaster.ac.uk/ | KITP Events | KITP | KITP Conference: Fro | Other bookmarks

jupyter PSI2016IJuliaV Last Checkpoint: 2 hours ago (unsaved changes)

File Edit View Insert Cell Kernel Help Julia 0.4.0

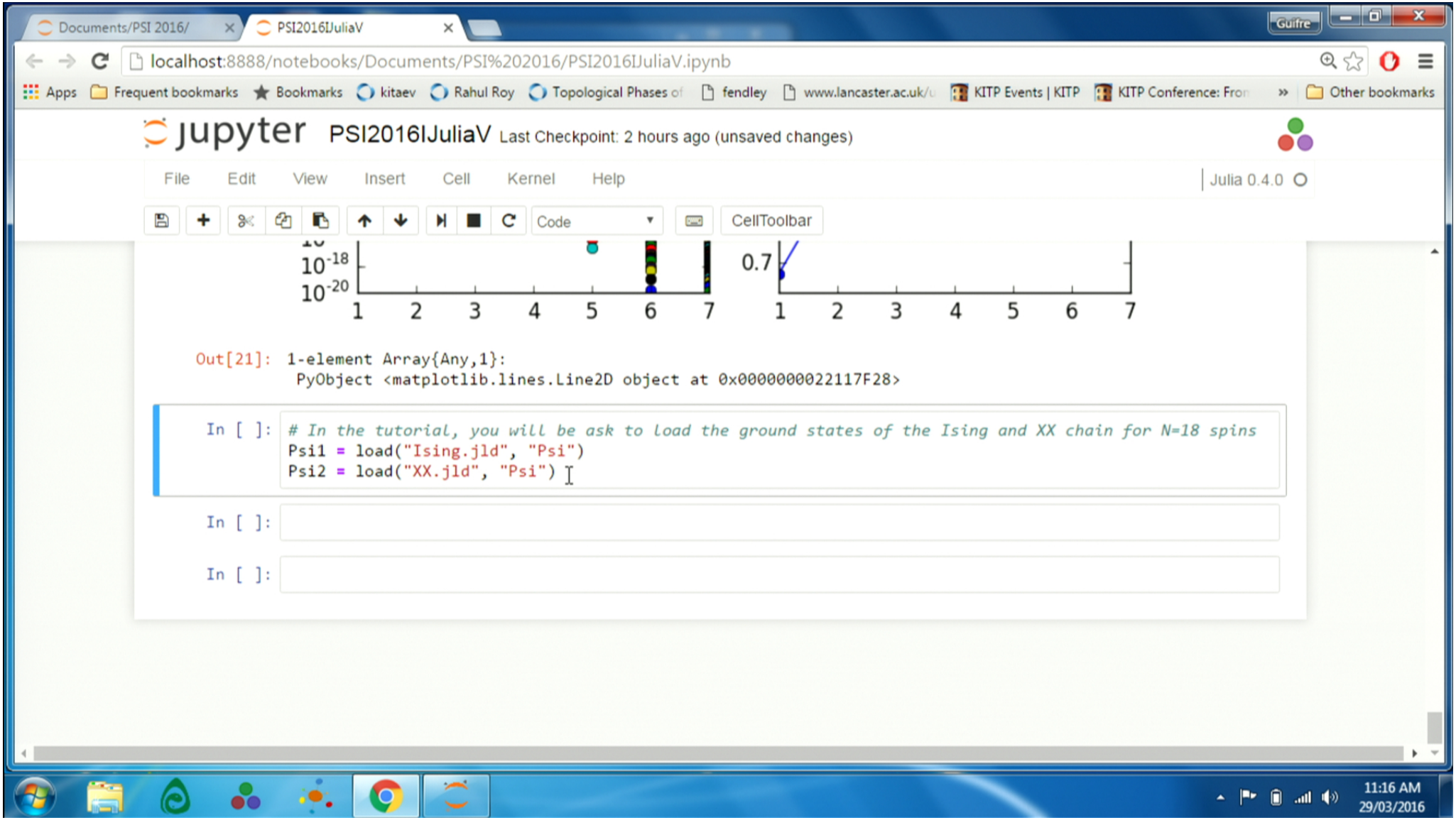
Code CellToolbar

```
subplot(221)
semilogy(L,ES', "o") # semilogy plot (y-axis in logarithmic scale)
subplot(222)
plot(L,EE, marker = "o") # regular plot
```



Out[21]: 1-element Array{Any,1}:
PyObject <matplotlib.lines.Line2D object at 0x0000000022117F28>

11:15 AM
29/03/2016


Documents/PSI 2016/
PSI2016IJuliaV
Guifre
localhost:8888/notebooks/Documents/PSI%202016/PSI2016IJuliaV.ipynb
Apps
Frequent bookmarks
Bookmarks
kitaev
Rahul Roy
Topological Phases of
fendley
www.lancaster.ac.uk/
KITP Events | KITP
KITP Conference: Fron
Other bookmarks
jupyter
PSI2016IJuliaV
Last Checkpoint: 2 hours ago (unsaved changes)
File
Edit
View
Insert
Cell
Kernel
Help
Julia 0.4.0
Code
CellToolbar
10-18
10-20
0.7
1 2 3 4 5 6 7
1 2 3 4 5 6 7
Out[21]: 1-element Array{Any,1}:
PyObject <matplotlib.lines.Line2D object at 0x0000000022117F28>
In [ ]:
# In the tutorial, you will be ask to load the ground states of the Ising and XX chain for N=18 spins
Psi1 = load("Ising.jld", "Psi")
Psi2 = load("XX.jld", "Psi")
In [ ]:
In [ ]:
11:16 AM
29/03/2016