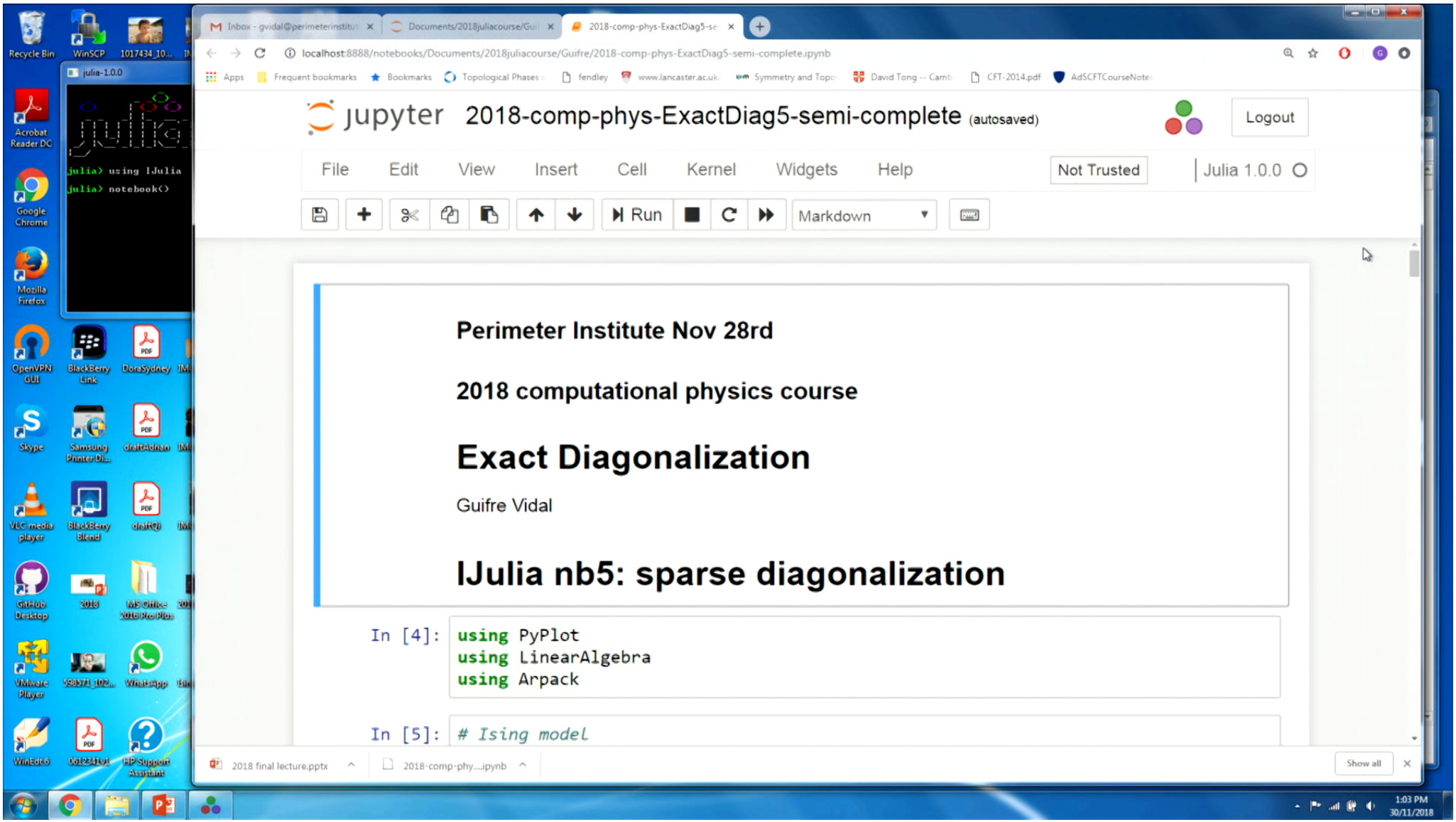
Title: Computational Physics - Lecture 24

Speakers: Erik Schnetter

Collection: Computational Physics (Schnetter)

Date: November 30, 2018 - 1:00 PM

URL: http://pirsa.org/18110050


jupyter 2018-comp-phys-ExactDiag5-semi-complete (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Julia 1.0.0
Run
Markdown
Perimeter Institute Nov 28rd
2018 computational physics course
Exact Diagonalization
Guifre Vidal
IJulia nb5: sparse diagonalization
In [4]:
using PyPlot
using LinearAlgebra
using Arpack
In [5]:
Ising model
julia> using IJulia
julia> notebook()

jupyter 2018-comp-phys-ExactDiag5-semi-complete (unsaved changes)

File Edit View Insert Cell Kernel Widgets Help Not Trusted Julia 1.0.0

In [8]:
```
# function for sparse multiplication

function multiplyHPsi(Psi, H2)
    L = length(Psi)  # Dimension of the vector space
    N = convert(Int64,log2(L))  # Number of spins
    HPsi = zeros(L)
    for n=1:N                                   # This multiplies by the Hamiltonian H
        Psi = reshape(Psi,(4,2^(N-2)))
        HPsi = reshape(HPsi,(4,2^(N-2)))
        HPsi += H2*Psi
        Psi = reshape(Psi,(2,2^(N-1)))
        HPsi = reshape(HPsi,(2,2^(N-1)))
        Psi = permutedims(Psi,(2,1))
        HPsi = permutedims(HPsi,(2,1))
    end
    Psi = reshape(Psi,L)
    HPsi = reshape(HPsi, L)
    return HPsi
end
```

Out[8]:
```
multiplyHPsi (generic function with 1 method)
```

jupyter 2018-comp-phys-ExactDiag5-semi-complete (unsaved changes)

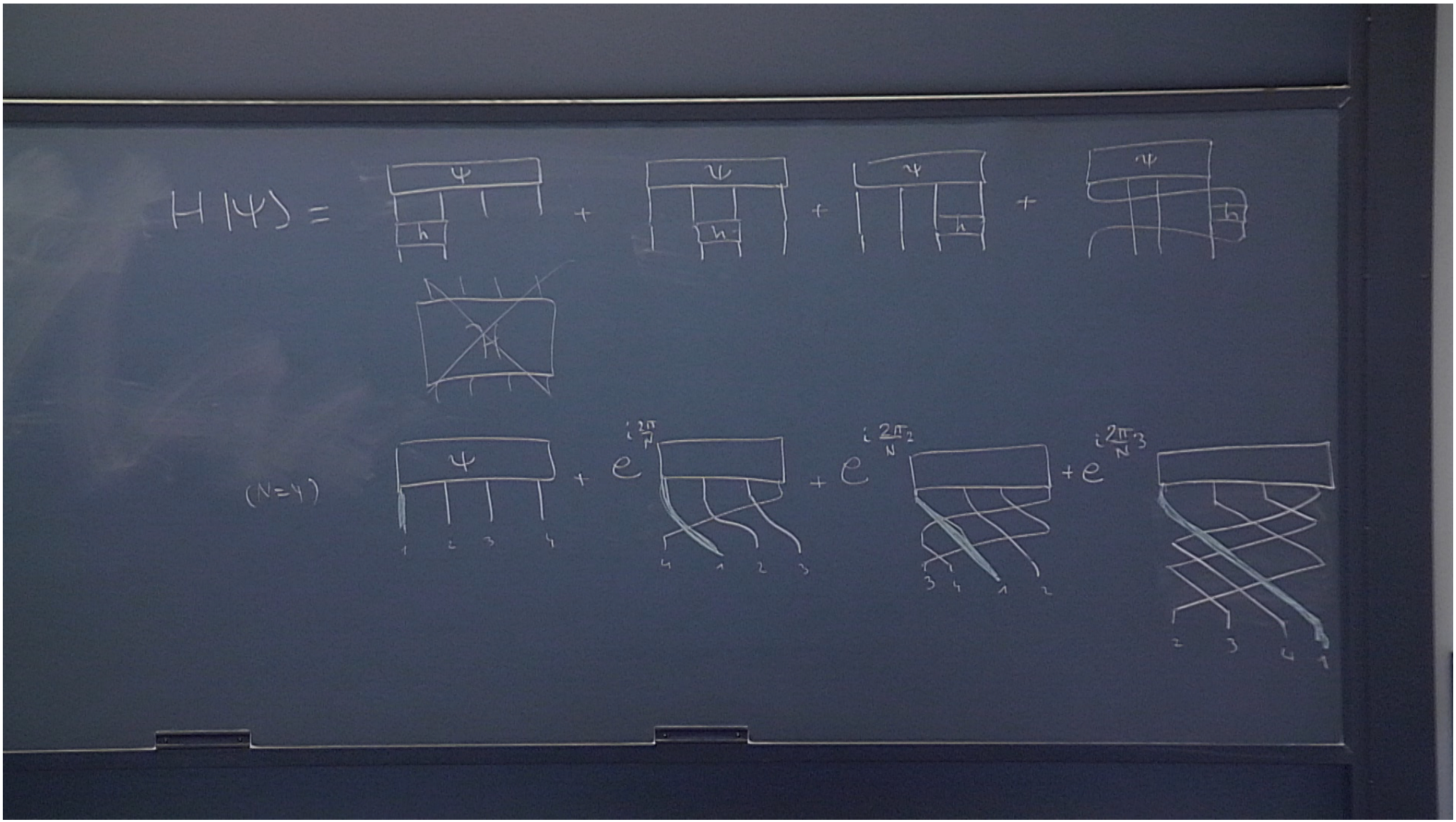
File Edit View Insert Cell Kernel Widgets Help Not Trusted Julia 1.0.0

Initialize Psi for N spins

In [12]:
```
N=18 #  N = 20     t < 0.5s
Psi = randn(2^N)

# Normalize Psi
Psi = Psi/sqrt(abs(Psi'*Psi))

@time multiplyHPsi(Psi,H2)

init_steps = 1
energy = ones(0)
initial_step=1
Nsteps = 0
;
```

```
0.086269 seconds (298 allocations: 146.012 MiB, 19.30% gc time)
```

Apply power method to Psi


H|Ψ⟩ =
Ψ
h
+
(N=4)
1 2 3 4
$e^{i\frac{2\pi}{N}}$
4 1 2 3
$e^{i\frac{2\pi}{N}2}$
3 4 1 2
$e^{i\frac{2\pi}{N}3}$
2 3 4 1

jupyter 2018-comp-phys-ExactDiag5-semi-complete (unsaved changes)

File Edit View Insert Cell Kernel Widgets Help Not Trusted Julia 1.0.0

Initialize Psi for N spins

In [6]:
```
N=18 #  N = 20     t < 0.5s
Psi = randn(2^N)

# Normalize Psi
Psi = Psi/sqrt(abs(Psi'*Psi))

@time multiplyHPsi(Psi,H2)

init_steps = 1
energy = ones(0)
initial_step=1
Nsteps = 0
;
```

```
0.641438 seconds (1.23 M allocations: 205.672 MiB, 7.16% gc time)
```

Apply power method to Psi

jupyter 2018-comp-phys-ExactDiag5-semi-complete (unsaved changes)

File Edit View Insert Cell Kernel Widgets Help Not Trusted Julia 1.0.0

Apply power method to Psi

In [*]:
```
# Use this to start or continue to apply power method

initial_step += Nsteps
Nsteps = 300
final_step = initial_step + Nsteps-1

for n=initial_step:final_step
    HPsi = multiplyHPsi(Psi,H2)
    newenergy = real(Psi'*HPsi) + shiftE*N
    energy = [energy; newenergy]
    Psi = HPsi/sqrt(abs(HPsi'*HPsi))
    print(n-initial_step+1, ":", Nsteps, " ")
end

bestenergy_ground = energy[end]
```

```
1:300 2:300 3:300 4:300 5:300 6:300 7:300 8:300 9:300 10:300 11:300 12:300 13:3
00 14:300 15:300 16:300 17:300 18:300 19:300 20:300 21:300 22:300 23:300 24:300
25:300 26:300 27:300 28:300 29:300 30:300 31:300 32:300 33:300 34:300 35:300 3
6:300 37:300 38:300 39:300 40:300 41:300 42:300 43:300 44:300 45:300 46:300 47:
300 48:300 49:300 50:300 51:300 52:300 53:300 54:300 55:300 56:300
```

jupyter 2018-comp-phys-ExactDiag5-semi-complete (unsaved changes)

File Edit View Insert Cell Kernel Widgets Help Not Trusted Julia 1.0.0

```
300 93:300 94:300 95:300 96:300 97:300 98:300 99:300 100:300 101:300 102:300 10
3:300 104:300 105:300 106:300 107:300 108:300 109:300 110:300 111:300 112:300 1
13:300 114:300 115:300 116:300 117:300 118:300 119:300 120:300 121:300 122:300
123:300 124:300 125:300 126:300 127:300 128:300 129:300 130:300 131:300 132:300
133:300 134:300 135:300 136:300 137:300 138:300 139:300 140:300 141:300 142:300
143:300 144:300 145:300 146:300 147:300 148:300 149:300 150:300 151:300 152:300
153:300 154:300 155:300 156:300 157:300 158:300 159:300 160:300 161:300 162:300
163:300 164:300 165:300 166:300 167:300 168:300 169:300 170:300 171:300 172:300
173:300 174:300 175:300 176:300 177:300 178:300 179:300 180:300 181:300 182:300
183:300 184:300 185:300 186:300 187:300 188:300 189:300 190:300 191:300 192:300
193:300 194:300 195:300 196:300 197:300 198:300 199:300 200:300 201:300 202:300
203:300 204:300 205:300 206:300 207:300 208:300 209:300 210:300 211:300 212:300
213:300 214:300 215:300 216:300 217:300 218:300 219:300 220:300 221:300 222:300
223:300 224:300 225:300 226:300 227:300 228:300 229:300 230:300 231:300 232:300
233:300 234:300 235:300 236:300 237:300 238:300 239:300 240:300 241:300 242:300
243:300 244:300 245:300 246:300 247:300 248:300
```

In [14]:

```
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")
```

jupyter 2018-comp-phys-ExactDiag5-semi-complete (unsaved changes)

File Edit View Insert Cell Kernel Widgets Help Not Trusted Julia 1.0.0

```
subplot(122) # Create the 1st axis of a 2x2 arrax of axes
grid("on") # Create a grid on the axis
title("Power method")
ax = gca()
xlabel("most recent iterations")
ylabel("Energy")
plot(initial_step:final_step, energy[initial_step:final_step], marker = "o");
```

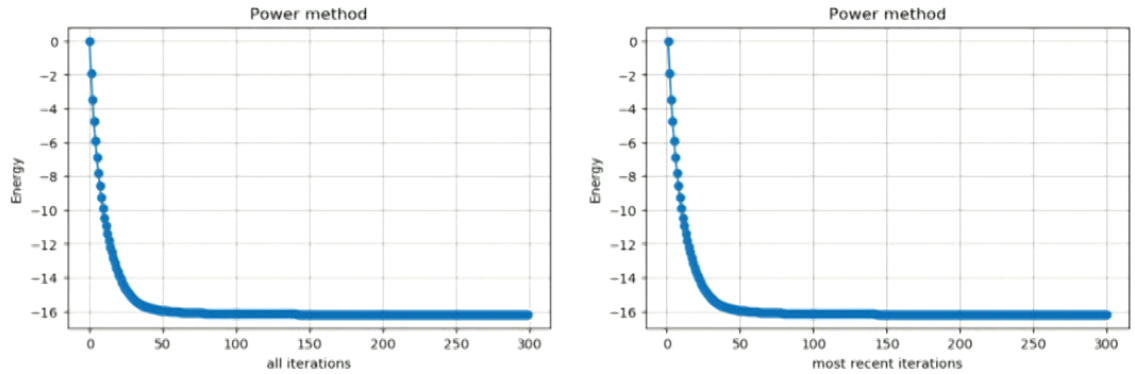
Exercise (1/2): For $N = 20$ spins, compute the lowest energy with momentum $k = n2\pi/N$ for $n = 0, 2, \cdots, N-1$

In [15]:

```
function normalizeVec(Vec)
    Vec = Vec/sqrt(abs(Vec'*Vec))
end

function give_momentum(Psi, k)
    L = length(Psi)  # Dimension of the vector space
    N = convert(Int64,log2(L))  # Number of spins
    newPsi = zeros(L);
```

jupyter 2018-comp-phys-ExactDiag5-semi-complete (unsaved changes)

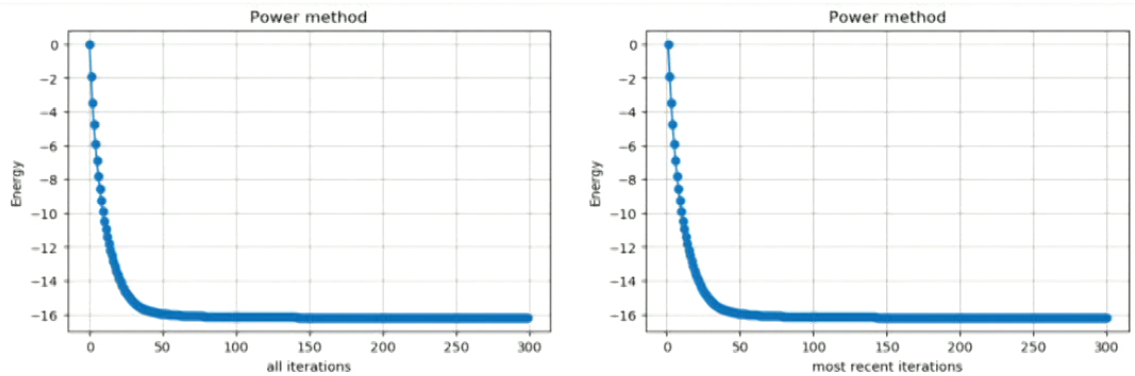
```
subplot(122) # Create the 1st axis of a 2x2 arrax of axes
grid("on") # Create a grid on the axis
title("Power method")
ax = gca()
xlabel("most recent iterations")
ylabel("Energy")
plot(initial_step:final_step, energy[initial_step:final_step], marker = "o");
```



jupyter 2018-comp-phys-ExactDiag5-semi-complete (unsaved changes)

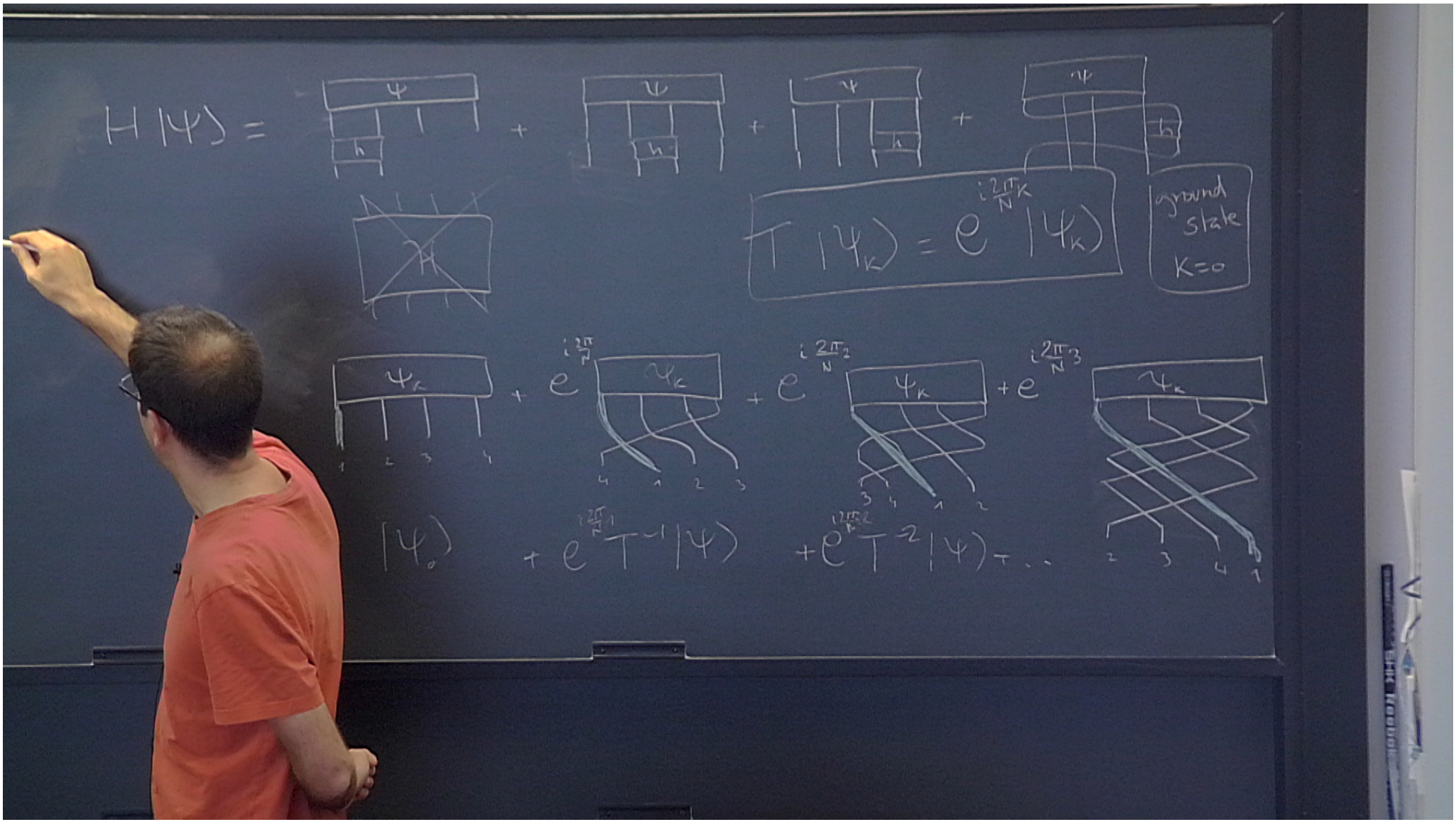
File Edit View Insert Cell Kernel Widgets Help Not Trusted Julia 1.0.0

Run Markdown



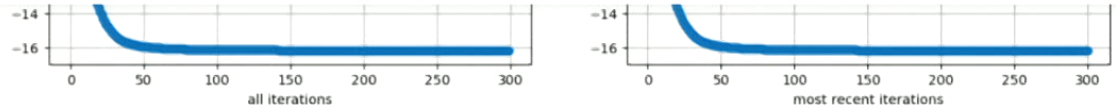
Exercise (1/2): For $N = 20$ spins, compute the lowest energy with momentum $k = n2\pi/N$ for $n = 0, 2, \cdots, N-1$

In [15]:
```
function normalizeVec(Vec)
    Vec = Vec/sqrt(abs(Vec'*Vec))
end
```


H |ψ⟩ =
T |ψ_k⟩ = e^{i 2π/N k} |ψ_k⟩
ground state
k=0
|ψ_0⟩ + e^{i 2π/N 1} T^{-1} |ψ⟩ + e^{i 2π/N 2} T^{-2} |ψ⟩ + . .

jupyter 2018-comp-phys-ExactDiag5-semi-complete (autosaved)

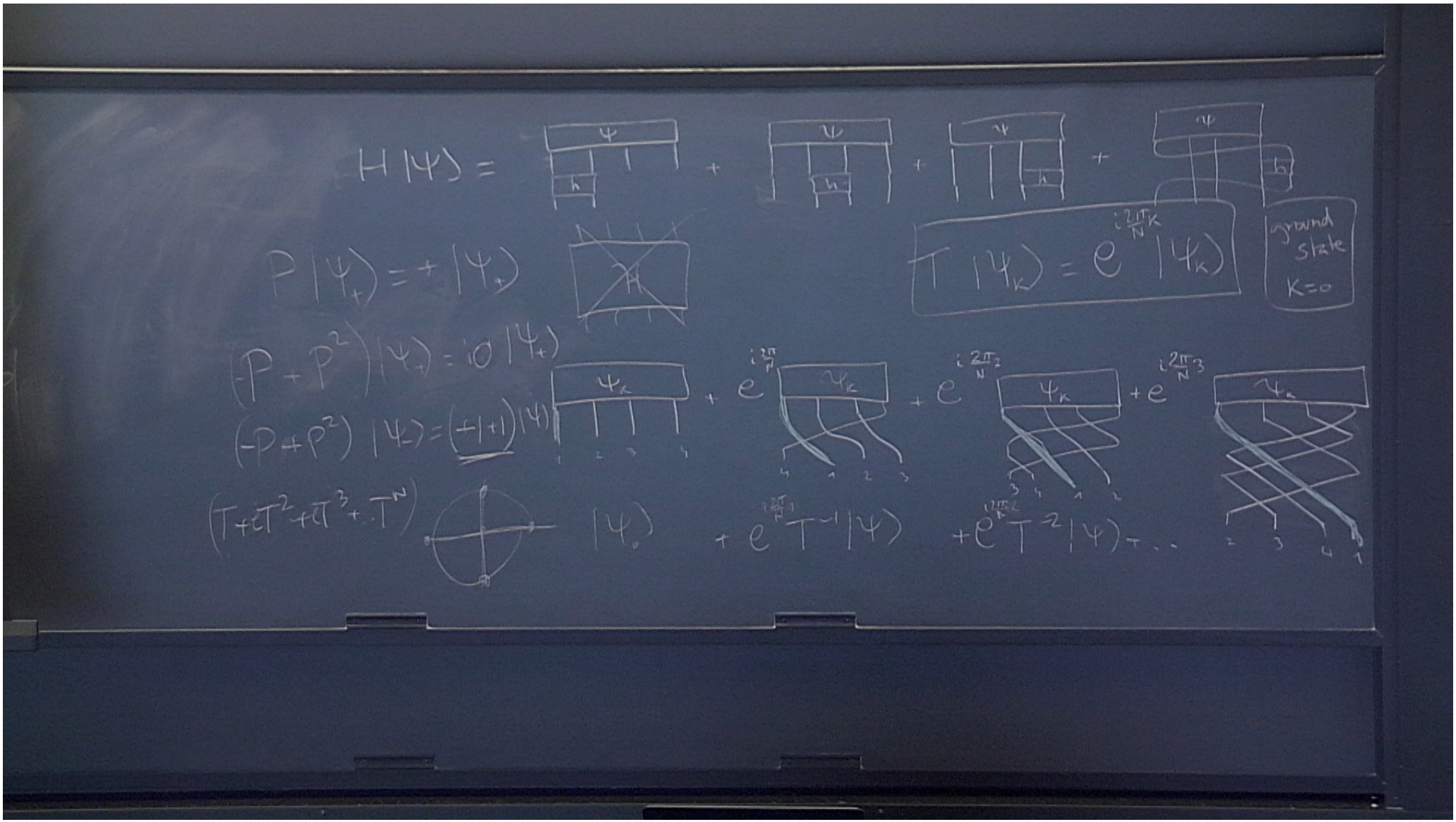
File Edit View Insert Cell Kernel Widgets Help Not Trusted Julia 1.0.0



Exercise (1/2): For $N = 20$ spins, compute the lowest energy with momentum $k = n2\pi/N$ for $n = 0, 2, \cdots, N-1$

In [15]:

```
function normalizeVec(Vec)
    Vec = Vec/sqrt(abs(Vec'*Vec))
end

function give_momentum(Psi, k)
    L = length(Psi)  # Dimension of the vector space
    N = convert(Int64,log2(L))  # Number of spins
    newPsi = zeros(L);
    for n in 1:N
        newPsi += Psi*exp(im*k*n*2pi/N)
        newPsi = reshape(newPsi, (2,2^(N-1)))
        newPsi = permutedims(newPsi, (2,1))
        newPsi = reshape(newPsi, (2^N))
```


H|ψ⟩ =
P|ψ+⟩ = +|ψ+⟩
(P + P²)|ψ+⟩ = 10|ψ+⟩
(−P + P²)|ψ−⟩ = (−1+1)|ψ⟩
(T + εT² + ε²T³ + . T^N)
|ψ0⟩
+ e^{i2π/N} T^{−1}|ψ⟩
+ e^{i2πk/N} T^{−2}|ψ⟩ + . .
T|ψk⟩ = e^{i2πk/N}|ψk⟩
ground state
k=0

jupyter 2018-comp-phys-ExactDiag5-semi-complete (autosaved)

File Edit View Insert Cell Kernel Widgets Help Not Trusted Julia 1.0.0

Exercise (1/2): For $N = 20$ spins, compute the lowest energy with momentum $k = n2\pi/N$ for $n = 0, 2, \cdots, N-1$

In [15]:

```
function normalizeVec(Vec)
    Vec = Vec/sqrt(abs(Vec'*Vec))
end

function give_momentum(Psi, k)
    L = length(Psi)  # Dimension of the vector space
    N = convert(Int64,log2(L))  # Number of spins
    newPsi = zeros(L);
    for n in 1:N
        newPsi += Psi*exp(im*k*n*2pi/N)
        newPsi = reshape(newPsi, (2,2^(N-1)))
        newPsi = permutedims(newPsi, (2,1))
        newPsi = reshape(newPsi, (2^N))
    end
    return normalizeVec(newPsi)
end
```

jupyter 2018-comp-phys-ExactDiag5-semi-complete (autosaved)

File Edit View Insert Cell Kernel Widgets Help Not Trusted Julia 1.0.0

```julia
    newPsi = zeros(L);
    for n in 1:N
        newPsi += Psi*exp(im*k*n*2pi/N)
        newPsi = reshape(newPsi, (2,2^(N-1)))
        newPsi = permutedims(newPsi, (2,1))
        newPsi = reshape(newPsi, (2^N))
    end
    return normalizeVec(newPsi)
end

function compute_momentum(Psi)
    L = length(Psi)  # Dimension of the vector space
    N = convert(Int64,log2(L))  # Number of spins
    TPsi = zeros(L)
    TPsi = reshape(Psi, (2,2^(N-1)))
    TPsi = permutedims(TPsi, (2,1))
    TPsi = reshape(TPsi, (2^N))
    scalarp = Psi'*TPsi
    return angle(scalarp)/(2pi)*N
end
```

Out[15]: compute_momentum (generic function with 1 method)

jupyter 2018-comp-phys-ExactDiag5-semi-complete (unsaved changes)

File Edit View Insert Cell Kernel Widgets Help Not Trusted Julia 1.0.0

In [16]:
```
# testing give_momentum and compute_momentum
q = zeros(N)
for n in 1:N
    print(n,":", N, " ")
    Q = give_momentum(randn(2^N)+im*randn(2^N), n)
    q[n] = compute_momentum(Q)
end
q
```

```
1:18 2:18 3:18 4:18 5:18 6:18 7:18 8:18 9:18 10:18 11:18 12:18 13:18 14:18 15:1
8 16:18 17:18 18:18
```

Out[16]:
```
18-element Array{Float64,1}:
  1.0000000000000038
  2.0000000000000018
  3.0000000000000044
  4.000000000000001
  4.999999999999999
  5.999999999999998
  6.9999999999999964
  8.000000000000002
 -9.0
 -7.999999999999997
```

jupyter 2018-comp-phys-ExactDiag5-semi-complete (unsaved changes)

File Edit View Insert Cell Kernel Widgets Help Not Trusted Julia 1.0.0

```
-1.9999999999999993
-1.0000000000000016
-2.4848083448933682e-17
```

In [17]:
```
N=18 #  N = 20    t < 0.5s

k = 1 # momentum k=0,1,...,N-1
Psik = give_momentum(randn(2^N)+im*randn(2^N), k) # normalized random vector with

@time multiplyHPsi(Psik,H2)

init_steps = 1
energy = ones(0)
initial_step=1
Nsteps = 0
;
```

```
0.638405 seconds (1.53 M allocations: 361.761 MiB, 17.40% gc time)
```

In [21]:
```
initial_step += Nsteps
Nsteps = 300
final_step = initial_step + Nsteps-1
```

jupyter 2018-comp-phys-ExactDiag5-semi-complete (unsaved changes)

File Edit View Insert Cell Kernel Widgets Help Not Trusted Julia 1.0.0

```
0.741472 seconds (1.53 M allocations: 361.796 MiB, 8.41% gc time)
```

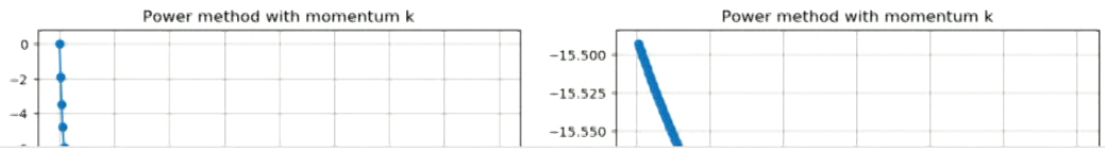
In [21]:

```julia
initial_step += Nsteps
Nsteps = 300
final_step = initial_step + Nsteps-1

for n=initial_step:final_step
    HPsik = multiplyHPsi(Psik,H2)
    newenergy = real(Psik'*HPsik) + shiftE*N
    energy = [energy; newenergy]
    Psik = normalizeVec(HPsik)
    print(n-initial_step+1, ":", Nsteps, " ")
end

bestenergy_k = energy[end]
```

```
1:300 2:300 3:300 4:300 5:300 6:300 7:300 8:300 9:300 10:300 11:300 12:300 13:3
00 14:300 15:300 16:300 17:300 18:300 19:300 20:300 21:300 22:300 23:300 24:300
25:300 26:300 27:300 28:300 29:300 30:300 31:300 32:300 33:300 34:300 35:300 3
6:300 37:300 38:300 39:300 40:300 41:300 42:300 43:300 44:300 45:300 46:300 47:
300 48:300 49:300 50:300 51:300 52:300 53:300 54:300 55:300 56:300 57:300 58:30
0 59:300 60:300 61:300 62:300 63:300 64:300 65:300 66:300 67:300 68:300 69:300
70:300 71:300 72:300 73:300 74:300 75:300 76:300 77:300 78:300 79:300 80:300 8
```

jupyter 2018-comp-phys-ExactDiag5-semi-complete (unsaved changes)

In [22]:
```
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 with momentum k")
ax = gca()
xlabel("all iterations")
ylabel("Energy")
plot(energy, marker = "o")

subplot(122) # Create the 1st axis of a 2x2 arrax of axes
grid("on") # Create a grid on the axis
title("Power method with momentum k")
ax = gca()
xlabel("most recent iterations")
ylabel("Energy")
plot(initial_step:final_step, energy[initial_step:final_step], marker = "o");
```



jupyter 2018-comp-phys-ExactDiag5-semi-complete (unsaved changes)

File Edit View Insert Cell Kernel Widgets Help Not Trusted Julia 1.0.0

```
final_step = initial_step + Nsteps-1

for n=initial_step:final_step
    HPsik = multiplyHPsi(Psik,H2)
    newenergy = real(Psik'*HPsik) + shiftE*N
    energy = [energy; newenergy]
    Psik = normalizeVec(HPsik)
    print(n-initial_step+1, ":", Nsteps, " ")
end

bestenergy_k = energy[end]
```

```
1:300 2:300 3:300 4:300 5:300 6:300 7:300 8:300 9:300 10:300 11:300 12:300 13:3
00 14:300 15:300 16:300 17:300 18:300 19:300 20:300 21:300 22:300 23:300 24:300
25:300 26:300 27:300 28:300 29:300 30:300 31:300 32:300 33:300 34:300
```

In [22]:

```
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 with momentum k")
ax = gca()
xlabel("all iterations")
```

jupyter 2018-comp-phys-ExactDiag5-semi-complete (unsaved changes)

File Edit View Insert Cell Kernel Widgets Help Not Trusted Julia 1.0.0

```
final_step = initial_step + Nsteps-1

for n=initial_step:final_step
    HPsik = multiplyHPsi(Psik,H2)
    newenergy = real(Psik'*HPsik) + shiftE*N
    energy = [energy; newenergy]
    Psik = normalizeVec(HPsik)
    print(n-initial_step+1, ":", Nsteps, " ")
end

bestenergy_k = energy[end]
```

```
1:300 2:300 3:300 4:300 5:300 6:300 7:300 8:300 9:300 10:300 11:300 12:300 13:3
00 14:300 15:300 16:300 17:300 18:300 19:300 20:300 21:300 22:300 23:300 24:300
25:300 26:300 27:300 28:300 29:300 30:300 31:300 32:300 33:300 34:300 35:300 3
6:300 37:300 38:300 39:300 40:300 41:300 42:300 43:300 44:300 45:300 46:300 47:
300 48:300 49:300 50:300 51:300 52:300 53:300 54:300 55:300 56:300 57:300 58:30
0 59:300 60:300 61:300 62:300 63:300 64:300 65:300 66:300 67:300 68:300 69:300
70:300 71:300 72:300 73:300 74:300 75:300 76:300 77:300 78:300 79:300 80:300 8
1:300 82:300 83:300 84:300 85:300 86:300
```

In [22]:
```
figure("Power_method",figsize=(14,4))
```

jupyter 2018-comp-phys-ExactDiag5-semi-complete (unsaved changes)

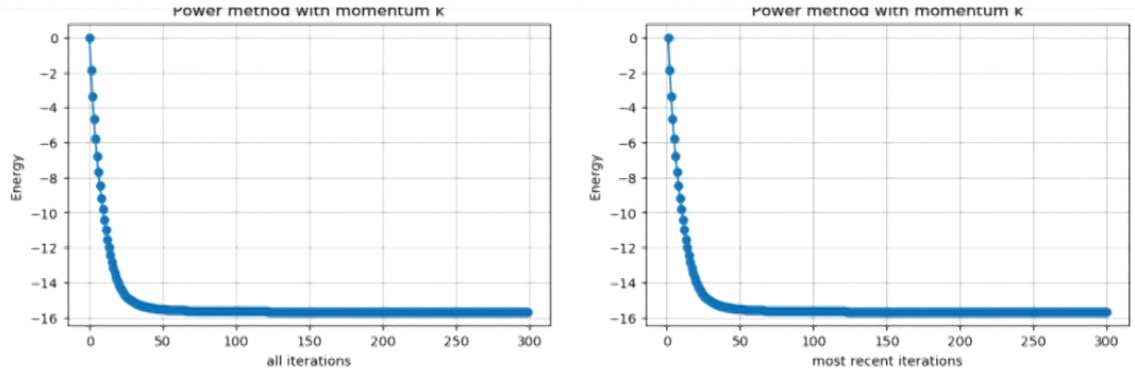
File Edit View Insert Cell Kernel Widgets Help Not Trusted Julia 1.0.0

```
70:300 71:300 72:300 73:300 74:300 75:300 76:300 77:300 78:300 79:300 80:300 8
1:300 82:300 83:300 84:300 85:300 86:300 87:300 88:300 89:300 90:300 91:300 92:
300 93:300 94:300 95:300 96:300 97:300 98:300 99:300 100:300 101:300 102:300 10
3:300 104:300 105:300 106:300 107:300 108:300 109:300 110:300 111:300 112:300 1
13:300 114:300 115:300 116:300 117:300 118:300 119:300 120:300 121:300 122:300
123:300 124:300 125:300 126:300 127:300 128:300 129:300 130:300 131:300 132:300
133:300 134:300 135:300 136:300 137:300 138:300 139:300 140:300 141:300 142:300
143:300 144:300 145:300 146:300 147:300 148:300 149:300 150:300 151:300 152:300
153:300 154:300 155:300 156:300 157:300 158:300 159:300 160:300 161:300 162:300
163:300 164:300 165:300 166:300 167:300 168:300 169:300 170:300 171:300 172:300
173:300 174:300 175:300 176:300 177:300 178:300 179:300 180:300 181:300 182:300
183:300 184:300 185:300 186:300 187:300 188:300 189:300 190:300 191:300 192:300
193:300 194:300 195:300 196:300 197:300 198:300 199:300 200:300 201:300 202:300
203:300 204:300 205:300 206:300 207:300 208:300 209:300 210:300 211:300 212:300
213:300 214:300 215:300 216:300 217:300 218:300 219:300 220:300 221:300 222:300
223:300 224:300 225:300 226:300 227:300 228:300 229:300 230:300 231:300 232:300
233:300 234:300 235:300 236:300 237:300 238:300 239:300 240:300 241:300 242:300
243:300 244:300 245:300 246:300 247:300 248:300 249:300 250:300 251:300 252:300
253:300 254:300 255:300 256:300 257:300 258:300 259:300 260:300
```

In [22]:
```
figure("Power_method",figsize=(14,4))

subplot(121) # Create the 1st axis of a 2x2 arrax of axes
```

jupyter 2018-comp-phys-ExactDiag5-semi-complete (unsaved changes)

File Edit View Insert Cell Kernel Widgets Help Not Trusted Julia 1.0.0



In [14]:
```
[bestenergy_ground bestenergy_k]
```

Out[14]:
```
1×2 Array{Float64,2}:
 -16.1761  -15.6733
```

Exercise (1/2): For $N = 20$ spins, compute first excited state with momentum $k = 0$

jupyter 2018-comp-phys-ExactDiag5-semi-complete (autosaved)

File Edit View Insert Cell Kernel Widgets Help Not Trusted Julia 1.0.0

Out[14]:
```
1×2 Array{Float64,2}:
 -16.1761  -15.6733
```

Exercise (1/2): For $N = 20$ spins, compute first excited state with momentum $k = 0$

In [24]:
```julia
function make_orthogonal(Excited,Ground) # it assumes Ground is normalized, that
    Excited = Excited - Ground *(Ground'*Excited)
    normalizeVec(Excited)
end

Ground = randn(2^N) + im*rand(2^N)
Ground = normalize(Ground)
Excited = rand(2^N) + im*rand(2^N)
Excited = normalizeVec(Excited)

display(Excited'*Ground)
Excited2 = make_orthogonal(Excited,Ground)
display(Excited2'*Ground)
```

jupyter 2018-comp-phys-ExactDiag5-semi-complete (autosaved)

File Edit View Insert Cell Kernel Widgets Help Not Trusted Julia 1.0.0

```
    Excited = Excited - Ground *(Ground'*Excited)
    normalizeVec(Excited)
end

Ground = randn(2^N) + im*rand(2^N)
Ground = normalize(Ground)
Excited = rand(2^N) + im*rand(2^N)
Excited = normalizeVec(Excited)

display(Excited'*Ground)
Excited2 = make_orthogonal(Excited,Ground)
display(Excited2'*Ground)
```

```
0.26346497919392153 + 0.26536685552068107im

1.609823385706477e-15 - 8.326672684688674e-17im
```

Note 1: Another possible application is the simulation of time evolution $|\Psi(t)\rangle \equiv \exp(-itH)|\Psi(0)\rangle$,

jupyter 2018-comp-phys-ExactDiag5-semi-complete (unsaved changes)

File Edit View Insert Cell Kernel Widgets Help Not Trusted Julia 1.0.0

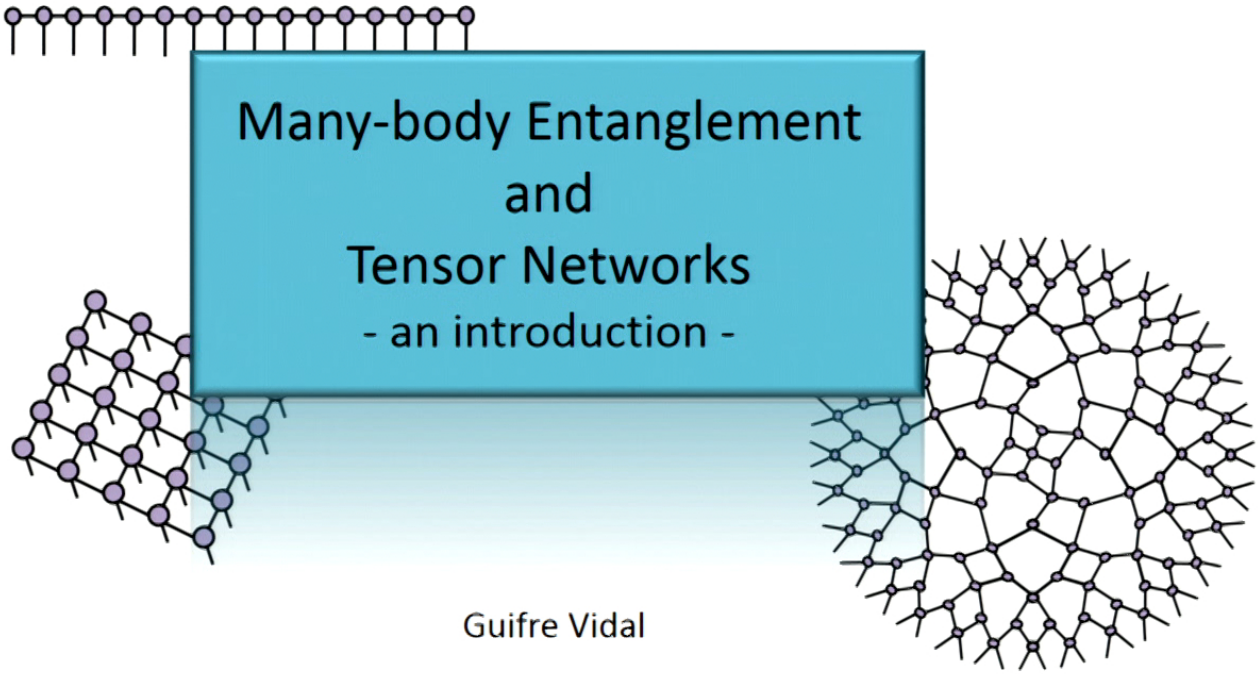
```
    Excited = Excited - Ground *(Ground'*Excited)
    normalizeVec(Excited)
end

Ground = randn(2^N) + im*rand(2^N)
Ground = normalizeVec(Ground)
Excited = rand(2^N) + im*rand(2^N)
Excited = normalizeVec(Excited)

display(Excited'*Ground)
Excited2 = make_orthogonal(Excited,Ground)
display(Excited2'*Ground)
```

```
0.2668777676382546 + 0.26402971759076915im

-2.7755575615628914e-16 + 3.0531133177191805e-16im
```

Note 1: Another possible application is the simulation of time evolution $|\Psi(t)\rangle \equiv \exp(-itH)|\Psi(0)\rangle$,

Perimeter Institute Erik Schnetter's Computational Physics Waterloo, November 30th 2018

last lecture of module 4



Many-body Entanglement and Tensor Networks

- an introduction -

Guifre Vidal

Instituto de Física Teórica UAM-CSIC

PERIMETER INSTITUTE FOR THEORETICAL PHYSICS

Tensor Networks Initiative

SIMONS FOUNDATION

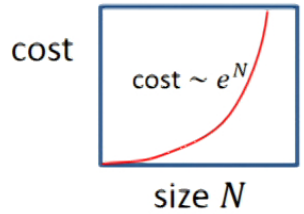
NSERC

compute canada | calcul canada

Outline

1 – Quantum mechanics and the many-body problem

$$i\hbar \frac{\partial}{\partial t}\Psi = \hat{H}\Psi$$

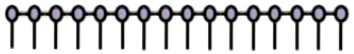


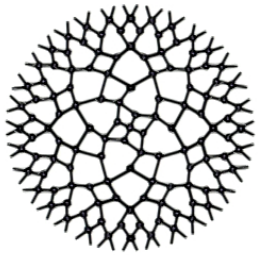
2 – Many-body entanglement

$$S(A) \sim |\partial A|$$

area law

3 – Tensor network states





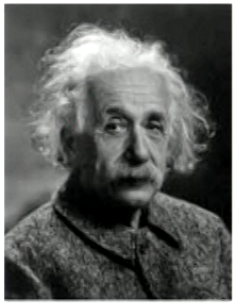
Part 1:

Quantum mechanics
and the many-body problem

$$i\hbar \frac{\partial}{\partial t}\Psi = \hat{H}\Psi$$

Quantum Mechanics 1920-1930

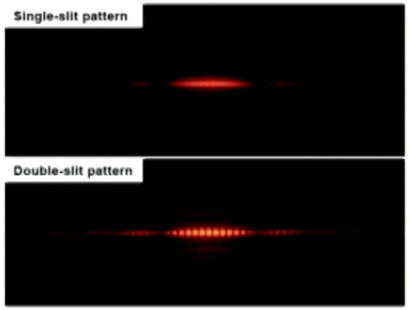
Niels Bohr



Albert Einstein

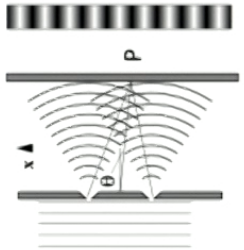
Wolfgang Pauli

Paul Dirac



$$i\hbar \frac{\partial}{\partial t}\Psi = \hat{H}\Psi$$

Schrodinger equation



Erwin Schrodinger

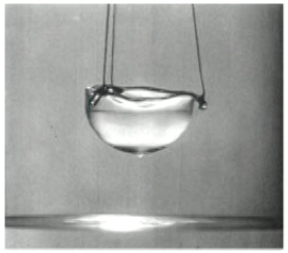
Enrico Fermi

Werner Heisenberg

Richard Feynman

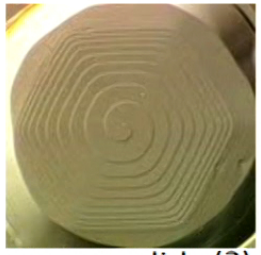
Exotic phases of quantum matter:

Collective quantum many-body phenomena

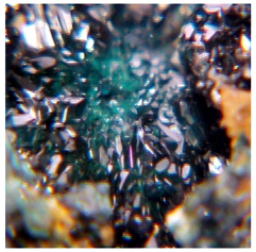


superfluids

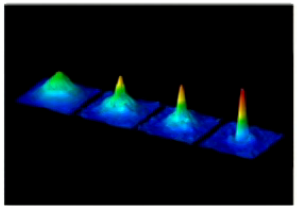
superconductors



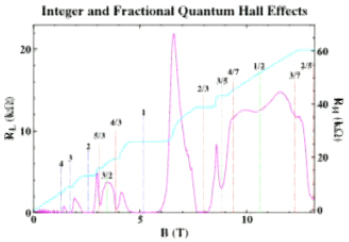
supersolids (?)



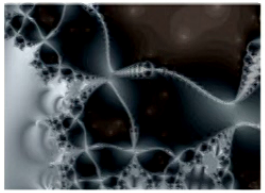
spin liquids



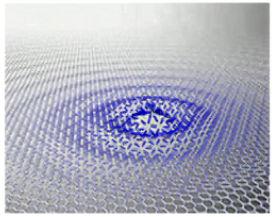
Bose-Einstein condensation



fractional quantum Hall effect



quantum criticality



topological order

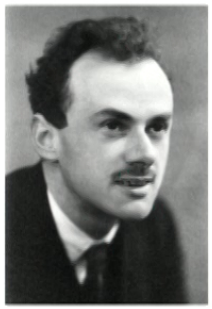
There is a problem…

Paul Dirac

"The fundamental laws necessary for the mathematical treatment of a large part of physics and the whole of chemistry are thus completely known,

and the difficulty lies only in the fact that application of these laws leads to equations that are too complex to be solved."

There is a problem…



Paul Dirac

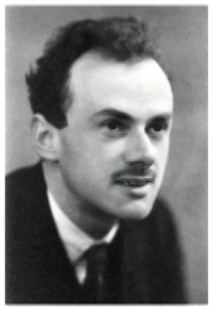
"The fundamental laws necessary for the mathematical treatment of a large part of physics and the whole of chemistry are thus completely known,

and the difficulty lies only in the fact that application of these laws leads to equations that are too complex to be solved."

$$i\hbar \frac{\partial}{\partial t}\Psi = \hat{H}\Psi$$

Schrodinger equation

There is a problem…



Paul Dirac

"The fundamental laws necessary for the mathematical treatment of a large part of physics and the whole of chemistry are thus completely known,

and the difficulty lies only in the fact that application of these laws leads to equations that are too complex to be solved."

$$i\hbar \frac{\partial}{\partial t}\Psi = \hat{H}\Psi$$

Schrodinger equation

There is a problem…

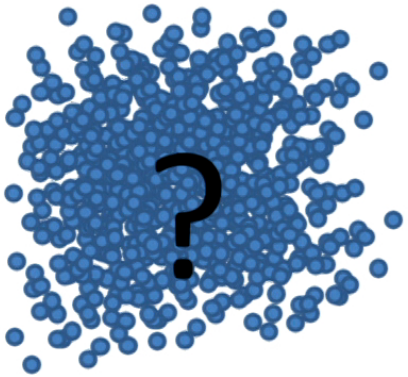
Paul Dirac

"The fundamental laws necessary for the mathematical treatment of a large part of physics and the whole of chemistry are thus completely known,

and the difficulty lies only in the fact that application of these laws leads to equations that are too complex to be solved."

$$i\hbar \frac{\partial}{\partial t}\Psi = \hat{H}\Psi$$

Schrodinger equation



There is a problem…

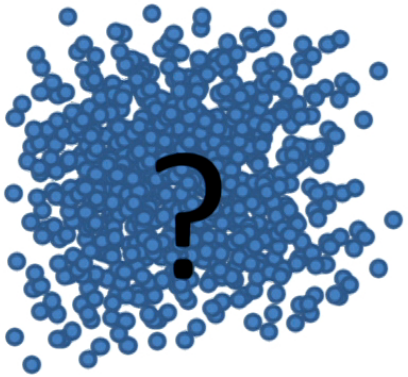
Paul Dirac

"The fundamental laws necessary for the mathematical treatment of a large part of physics and the whole of chemistry are thus completely known,

and the difficulty lies only in the fact that application of these laws leads to equations that are too complex to be solved."

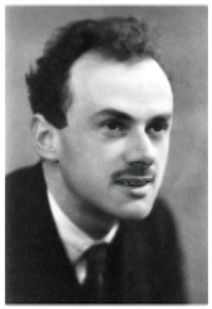
$$i\hbar \frac{\partial}{\partial t}\Psi = \hat{H}\Psi$$

Schrodinger equation



Emergence:
long-distance physics is often radically different from short-distance physics

There is a problem...



Paul Dirac

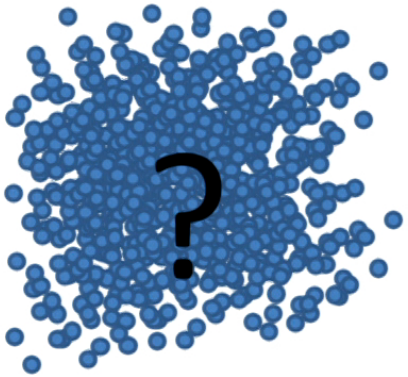
"The fundamental laws necessary for the mathematical treatment of a large part of physics and the whole of chemistry are thus completely known,

and the difficulty lies only in the fact that application of these laws leads to equations that are too complex to be solved."

1929!

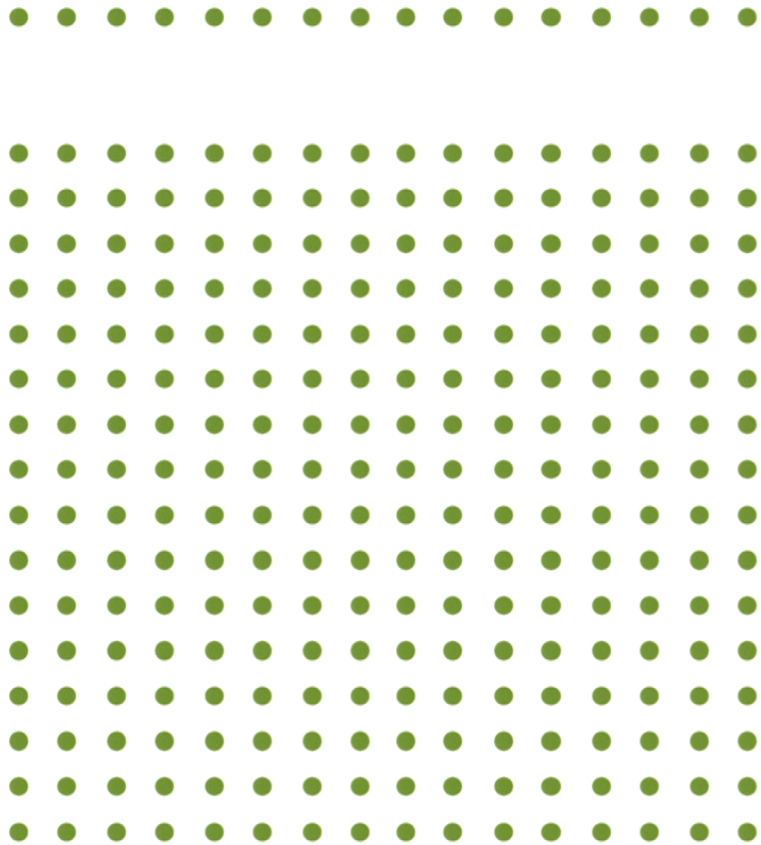
$$i\hbar \frac{\partial}{\partial t}\Psi = \hat{H}\Psi$$

Schrodinger equation



Emergence:
long-distance physics is often radically different from short-distance physics

lattice model of quantum spins

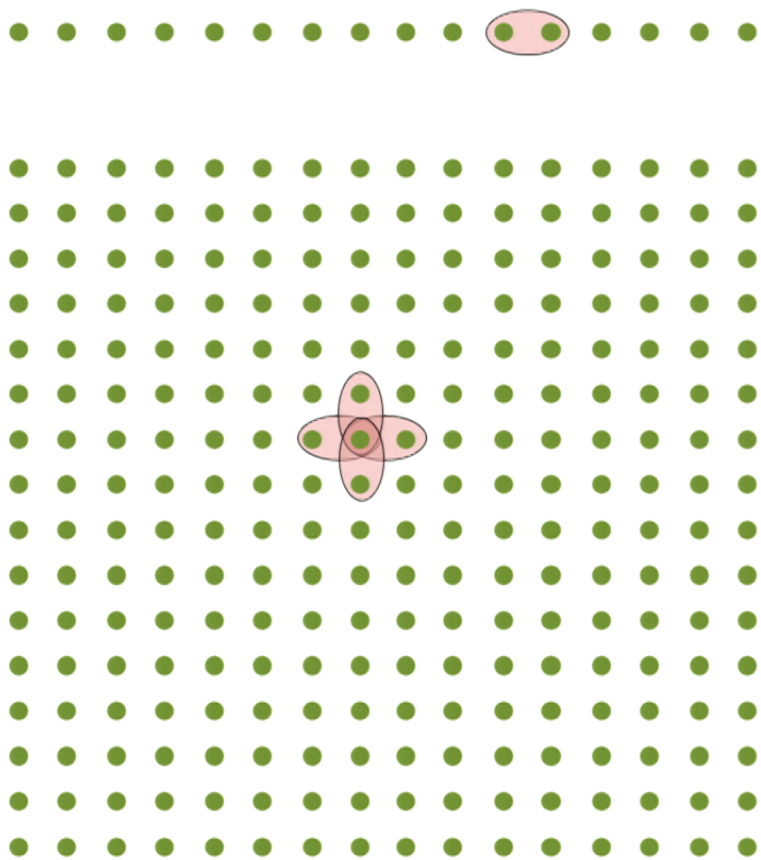


Hilbert space

$\mathcal{H} = \mathbb{C}_2$

spin ↑↓

lattice model of quantum spins



Hilbert space

$\mathcal{H} = \mathbb{C}_2$

spin ↑↓

local Hamiltonian

$$\hat{H} = \sum_{<i,j>} \hat{h}_{ij}$$

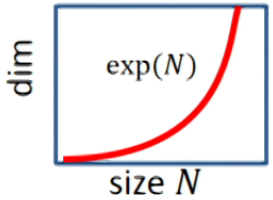
$$\hat{H} = \sum_{<ij>} \vec{S}_i \cdot \vec{S}_j$$

number of spins	Hilbert space dimension
$N = 1$	$2^1 = 2$

number of spins		Hilbert space dimension	
$N = 1$	●	$2^1 = 2$	$\lvert\Psi\rangle = c_1\lvert\uparrow\rangle + c_2\lvert\downarrow\rangle$

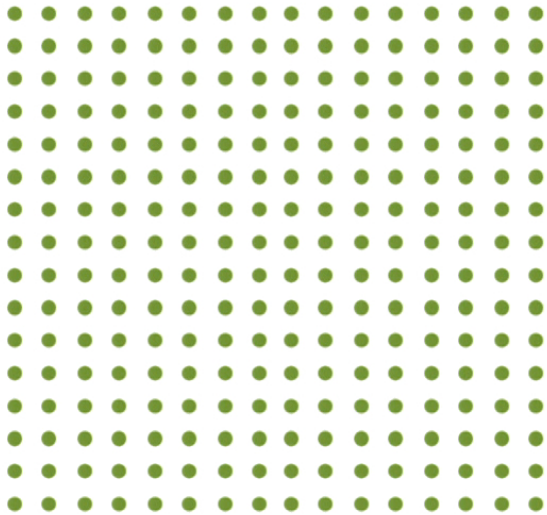
number of spins		Hilbert space dimension	
$N = 1$	●	$2^1 = 2$	$\lvert\Psi\rangle = c_1\lvert\uparrow\rangle + c_2\lvert\downarrow\rangle$
$N = 2$	● ●	$2^2 = 4$	$\lvert\Psi\rangle = c_1\lvert\uparrow\uparrow\rangle + c_2\lvert\uparrow\downarrow\rangle + c_3\lvert\downarrow\uparrow\rangle + c_4\lvert\downarrow\downarrow\rangle$
$N = 3$	● ● ●	$2^3 = 8$	$\lvert\Psi\rangle = c_1\lvert\uparrow\uparrow\uparrow\rangle + c_2\lvert\uparrow\uparrow\downarrow\rangle + c_3\lvert\uparrow\downarrow\uparrow\rangle + c_4\lvert\uparrow\downarrow\downarrow\rangle + c_5\lvert\downarrow\uparrow\uparrow\rangle + c_6\lvert\downarrow\uparrow\downarrow\rangle + c_7\lvert\downarrow\downarrow\uparrow\rangle + c_8\lvert\downarrow\downarrow\downarrow\rangle$
$N = 4$	● ● ● ●	$2^4 = 16$	$\lvert\Psi\rangle = c_1\lvert\uparrow\uparrow\uparrow\uparrow\rangle + c_2\lvert\uparrow\uparrow\uparrow\downarrow\rangle + c_3\lvert\uparrow\uparrow\downarrow\uparrow\rangle + c_4\lvert\uparrow\uparrow\downarrow\downarrow\rangle + c_5\lvert\uparrow\downarrow\uparrow\uparrow\rangle + c_6\lvert\uparrow\downarrow\uparrow\downarrow\rangle + c_7\lvert\uparrow\downarrow\downarrow\uparrow\rangle + c_8\lvert\uparrow\downarrow\downarrow\downarrow\rangle + c_9\lvert\downarrow\uparrow\uparrow\uparrow\rangle + c_{10}\lvert\downarrow\uparrow\uparrow\downarrow\rangle + c_{11}\lvert\downarrow\uparrow\downarrow\uparrow\rangle + c_{12}\lvert\downarrow\uparrow\downarrow\downarrow\rangle + c_{13}\lvert\downarrow\downarrow\uparrow\uparrow\rangle + c_{14}\lvert\downarrow\downarrow\uparrow\downarrow\rangle + c_{15}\lvert\downarrow\downarrow\downarrow\uparrow\rangle + c_{16}\lvert\downarrow\downarrow\downarrow\downarrow\rangle$

number of spins		Hilbert space dimension	
$N = 1$	●	$2^1 = 2$	$\lvert\Psi\rangle = c_1\lvert\uparrow\rangle + c_2\lvert\downarrow\rangle$
$N = 2$	● ●	$2^2 = 4$	$\lvert\Psi\rangle = c_1\lvert\uparrow\uparrow\rangle + c_2\lvert\uparrow\downarrow\rangle + c_3\lvert\downarrow\uparrow\rangle + c_4\lvert\downarrow\downarrow\rangle$
$N = 3$	● ● ●	$2^3 = 8$	$\lvert\Psi\rangle = c_1\lvert\uparrow\uparrow\uparrow\rangle + c_2\lvert\uparrow\uparrow\downarrow\rangle + c_3\lvert\uparrow\downarrow\uparrow\rangle + c_4\lvert\uparrow\downarrow\downarrow\rangle + c_5\lvert\downarrow\uparrow\uparrow\rangle + c_6\lvert\downarrow\uparrow\downarrow\rangle + c_7\lvert\downarrow\downarrow\uparrow\rangle + c_8\lvert\downarrow\downarrow\downarrow\rangle$
$N = 4$	● ● ● ●	$2^4 = 16$	$\lvert\Psi\rangle = c_1\lvert\uparrow\uparrow\uparrow\uparrow\rangle + c_2\lvert\uparrow\uparrow\uparrow\downarrow\rangle + c_3\lvert\uparrow\uparrow\downarrow\uparrow\rangle + c_4\lvert\uparrow\uparrow\downarrow\downarrow\rangle + c_5\lvert\uparrow\downarrow\uparrow\uparrow\rangle + c_6\lvert\uparrow\downarrow\uparrow\downarrow\rangle + c_7\lvert\uparrow\downarrow\downarrow\uparrow\rangle + c_8\lvert\uparrow\downarrow\downarrow\downarrow\rangle + c_9\lvert\downarrow\uparrow\uparrow\uparrow\rangle + c_{10}\lvert\downarrow\uparrow\uparrow\downarrow\rangle + c_{11}\lvert\downarrow\uparrow\downarrow\uparrow\rangle + c_{12}\lvert\downarrow\uparrow\downarrow\downarrow\rangle + c_{13}\lvert\downarrow\downarrow\uparrow\uparrow\rangle + c_{14}\lvert\downarrow\downarrow\uparrow\downarrow\rangle + c_{15}\lvert\downarrow\downarrow\downarrow\uparrow\rangle + c_{16}\lvert\downarrow\downarrow\downarrow\downarrow\rangle$
N	● ● ● ··· ●	$dim = 2^N$	



example: $N = 16 \times 16 = 256$

$\dim = 2^{256} \approx 10^{77}$ complex numbers



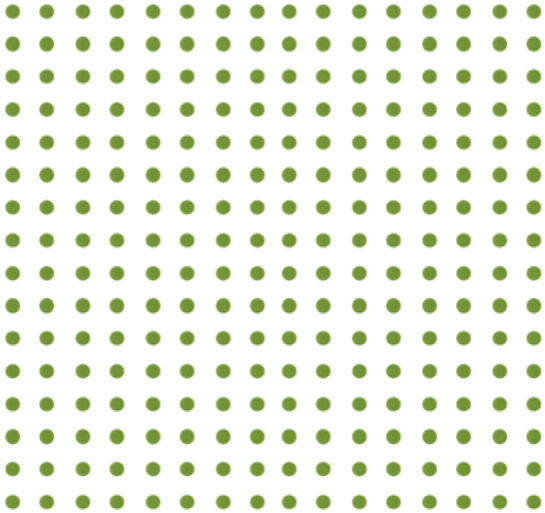
example: $N = 16 \times 16 = 256$

$\dim = 2^{256} \approx 10^{77}$ complex numbers

16GB laptop

10^9 complex numbers

$N = 32$ spins



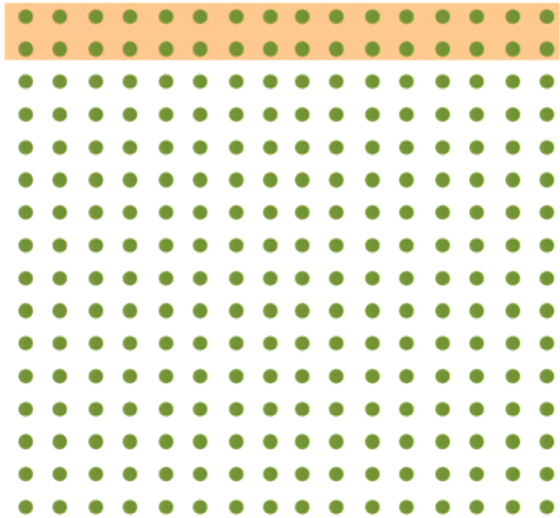
example: $N = 16 \times 16 = 256$

$\dim = 2^{256} \approx 10^{77}$ complex numbers

16GB laptop

10^9 complex numbers

$N = 32$ spins



example: $N = 16 \times 16 = 256$

$\dim = 2^{256} \approx 10^{77}$ complex numbers

16GB laptop

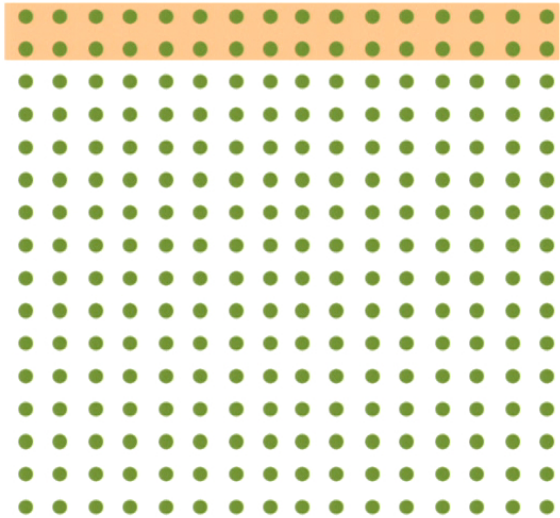
10^9 complex numbers

$N = 32$ spins

petabyte (10^{15} bytes) supercomputer

10^{14} complex numbers

$N = 48$ spins



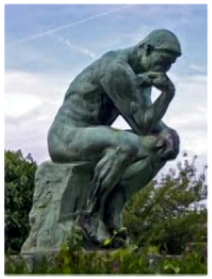
example: $N = 16 \times 16 = 256$

$\dim = 2^{256} \approx 10^{77}$ complex numbers

# of atoms in

human being

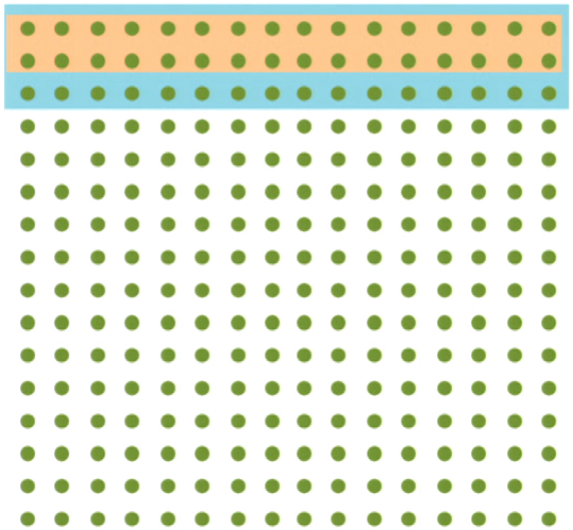
10^{28}



16GB laptop

10^9 complex numbers

$N = 32$ spins



petabyte (10^{15} bytes) supercomputer

10^{14} complex numbers

$N = 48$ spins

example: $N = 16 \times 16 = 256$

$\dim = 2^{256} \approx 10^{77}$ complex numbers

16GB laptop

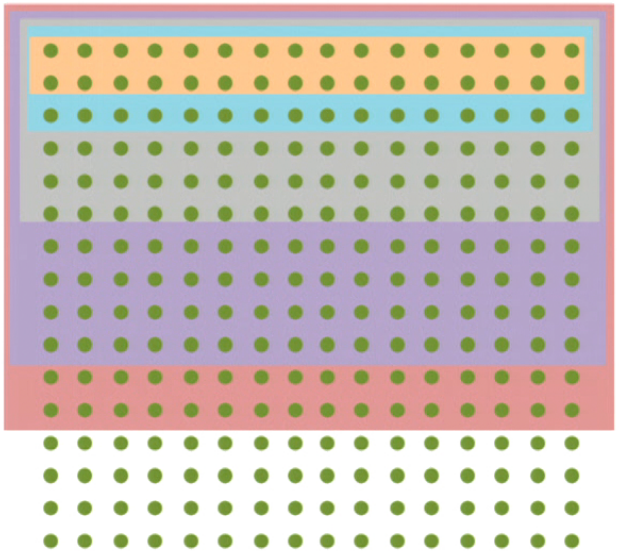
10^9 complex numbers

$N = 32$ spins

petabyte (10^{15} bytes) supercomputer

10^{14} complex numbers

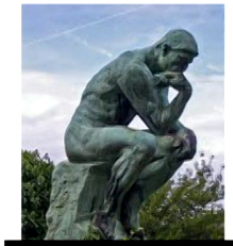
$N = 48$ spins



# of atoms in

human being

10^{28}



earth

10^{50}

solar system

10^{57}

example: $N = 16 \times 16 = 256$

$\dim = 2^{256} \approx 10^{77}$ complex numbers

16GB laptop

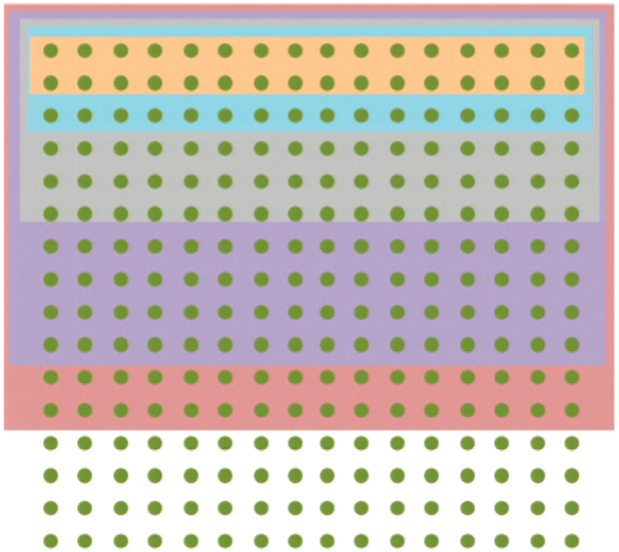
10^9 complex numbers

$N = 32$ spins

petabyte (10^{15} bytes) supercomputer

10^{14} complex numbers

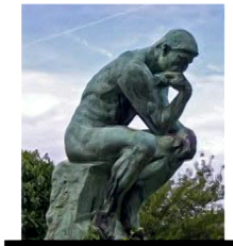
$N = 48$ spins



# of atoms in

human being

10^{28}



earth

10^{50}

solar system

10^{57}

milky way

10^{69}

example: $N = 16 \times 16 = 256$

$\dim = 2^{256} \approx 10^{77}$ complex numbers

16GB laptop

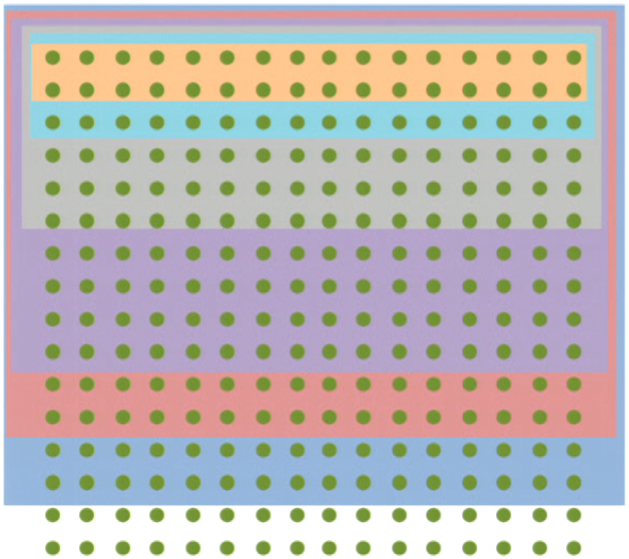
10^9 complex numbers

$N = 32$ spins

petabyte (10^{15} bytes) supercomputer

10^{14} complex numbers

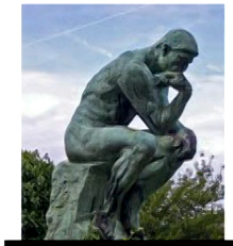
$N = 48$ spins



# of atoms in

human being

10^{28}



earth

10^{50}

solar system

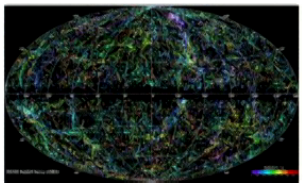
10^{57}

milky way

10^{69}

observable universe

10^{80}



example: $N = 16 \times 16 = 256$

$\dim = 2^{256} \approx 10^{77}$ complex numbers

16GB laptop

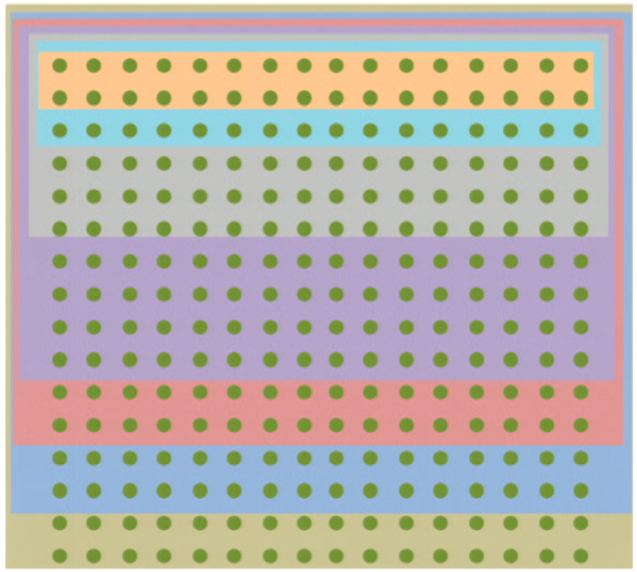
10^9 complex numbers

$N = 32$ spins

petabyte (10^{15} bytes) supercomputer

10^{14} complex numbers

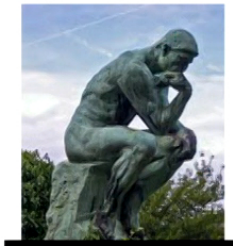
$N = 48$ spins



# of atoms in

human being

10^{28}



earth

10^{50}

solar system

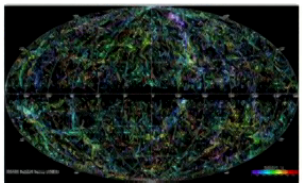
10^{57}

milky way

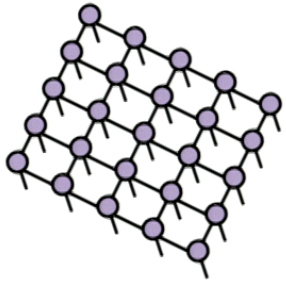
10^{69}

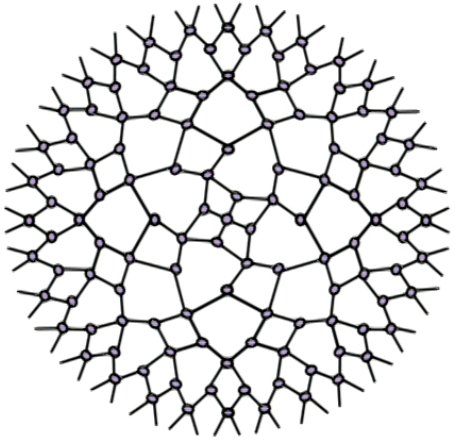
observable universe

10^{80}

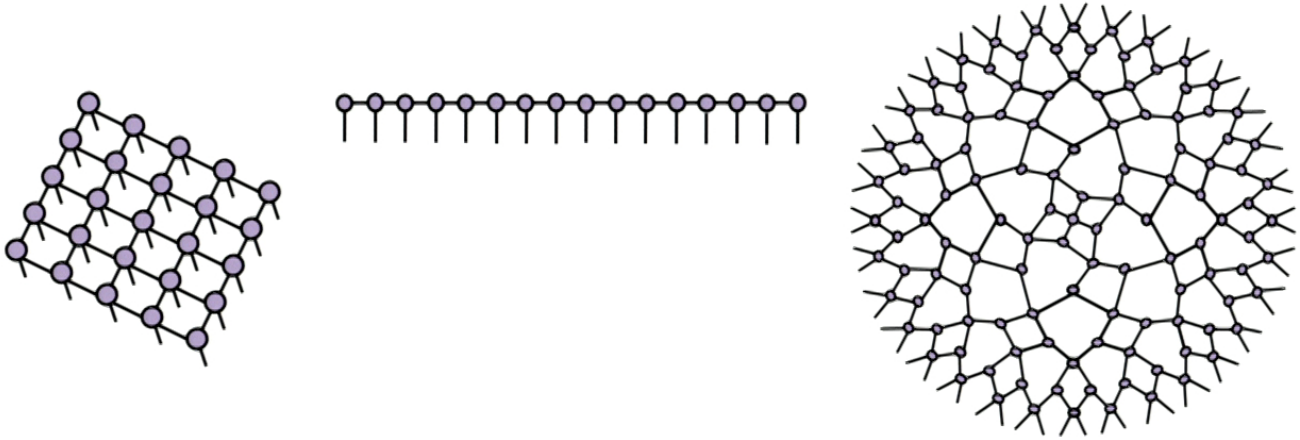


Today's talk: tensor network states





Today's talk: tensor network states



- Variational method

cost $\sim N$ | biased, but only weakly

(biased towards low entanglement)

Part 2:

Many-body entanglement

$$S(A) \sim |\partial A|$$

area law

Entanglement

$N = 2$

$\mathcal{H}^A \otimes \mathcal{H}^B$

Entanglement

$N = 2$ $\qquad \mathcal{H}^A \otimes \mathcal{H}^B$

Def.: product state

$$|\Psi^{AB}\rangle = |\psi^A\rangle \otimes |\varphi^B\rangle$$

example: $|\uparrow^A\rangle \otimes |\uparrow^B\rangle$

Entanglement

$N = 2$ $\qquad \mathcal{H}^A \otimes \mathcal{H}^B$

Def.: product state

$$|\Psi^{AB}\rangle = |\psi^A\rangle \otimes |\varphi^B\rangle$$

example: $|\uparrow^A\rangle \otimes |\uparrow^B\rangle$

Def.: entangled state

$$|\Psi^{AB}\rangle \neq |\psi^A\rangle \otimes |\varphi^B\rangle$$

example:

$$|\uparrow^A\rangle \otimes |\downarrow^B\rangle - |\downarrow^A\rangle \otimes |\uparrow^B\rangle$$

Entanglement

$N = 2$ $\qquad \mathcal{H}^A \otimes \mathcal{H}^B$

Def.: product state

$$|\Psi^{AB}\rangle = |\psi^A\rangle \otimes |\varphi^B\rangle$$

example: $|\uparrow^A\rangle \otimes |\uparrow^B\rangle$

Def.: entangled state

$$|\Psi^{AB}\rangle \neq |\psi^A\rangle \otimes |\varphi^B\rangle$$

example: $|\uparrow^A\rangle \otimes |\downarrow^B\rangle - |\downarrow^A\rangle \otimes |\uparrow^B\rangle$

product state or entangled state?

$$|\uparrow^A\rangle \otimes |\uparrow^B\rangle + |\uparrow^A\rangle \otimes |\downarrow^B\rangle + |\downarrow^A\rangle \otimes |\uparrow^B\rangle + |\downarrow^A\rangle \otimes |\downarrow^B\rangle$$

Is state $|\Psi^{AB}\rangle$ entangled?

$$\mathcal{H}^A \otimes \mathcal{H}^B = \mathbb{C}_d \otimes \mathbb{C}_d \qquad |\Psi^{AB}\rangle = \sum_{i,j=1}^{d} c_{ij}\, |i^A\rangle \otimes |j^B\rangle$$

$N = 2$

Is state $|\Psi^{AB}\rangle$ entangled?

$$\mathcal{H}^A \otimes \mathcal{H}^B = \mathbb{C}_d \otimes \mathbb{C}_d \qquad |\Psi^{AB}\rangle = \sum_{i,j=1}^{d} c_{ij}\, |i^A\rangle \otimes |j^B\rangle$$

$N = 2$

$$\rho^A \equiv tr_B\, [|\Psi^{AB}\rangle\langle\Psi^{AB}|]$$

A B

Is state $|\Psi^{AB}\rangle$ entangled?

$$\mathcal{H}^A \otimes \mathcal{H}^B = \mathbb{C}_d \otimes \mathbb{C}_d \qquad |\Psi^{AB}\rangle = \sum_{i,j=1}^{d} c_{ij}\, |i^A\rangle \otimes |j^B\rangle$$

$N = 2$

$$\rho^A \equiv tr_B\, [|\Psi^{AB}\rangle\langle\Psi^{AB}|]$$

A

Is state $|\Psi^{AB}\rangle$ entangled?

$$\mathcal{H}^A \otimes \mathcal{H}^B = \mathbb{C}_d \otimes \mathbb{C}_d \qquad |\Psi^{AB}\rangle = \sum_{i,j=1}^{d} c_{ij}\, |i^A\rangle \otimes |j^B\rangle$$

$N = 2$

$$\rho^A \equiv tr_B\, [|\Psi^{AB}\rangle\langle\Psi^{AB}|]$$

$|\Psi^{AB}\rangle$ is **entangled** $\Leftrightarrow$ ρ^A is **mixed**

$$\rho^A \equiv \sum_{\alpha} p_\alpha\, |\psi_\alpha^A\rangle\langle\psi_\alpha^A|$$

Is state $|\Psi^{AB}\rangle$ entangled?

$$\mathcal{H}^A \otimes \mathcal{H}^B = \mathbb{C}_d \otimes \mathbb{C}_d \qquad |\Psi^{AB}\rangle = \sum_{i,j=1}^{d} c_{ij}\, |i^A\rangle \otimes |j^B\rangle$$

$N = 2$

$$\rho^A \equiv tr_B\, [|\Psi^{AB}\rangle\langle\Psi^{AB}|]$$

$|\Psi^{AB}\rangle$ is **entangled** $\Leftrightarrow$ ρ^A is **mixed**

$$\rho^A \equiv \sum_{\alpha} p_\alpha\, |\psi_\alpha^A\rangle\langle\psi_\alpha^A|$$

How entangled is $|\Psi^{AB}\rangle$?

Is state $|\Psi^{AB}\rangle$ entangled?

$$\mathcal{H}^A \otimes \mathcal{H}^B = \mathbb{C}_d \otimes \mathbb{C}_d \qquad |\Psi^{AB}\rangle = \sum_{i,j=1}^{d} c_{ij}\, |i^A\rangle \otimes |j^B\rangle$$

$N = 2$

$$\rho^A \equiv tr_B\, [|\Psi^{AB}\rangle\langle\Psi^{AB}|]$$

$|\Psi^{AB}\rangle$ is **entangled** $\Leftrightarrow$ ρ^A is **mixed**

$$\rho^A \equiv \sum_{\alpha} p_\alpha\, |\psi_\alpha^A\rangle\langle\psi_\alpha^A|$$

How entangled is $|\Psi^{AB}\rangle$?

measure of entanglement

$$E(\Psi^{AB}) \equiv S(\rho^A) = -\sum_{\alpha} p_\alpha \log_2 p_\alpha$$

von Neumann entropy of ρ^A

Is state $|\Psi^{AB}\rangle$ entangled?

$$\mathcal{H}^A \otimes \mathcal{H}^B = \mathbb{C}_d \otimes \mathbb{C}_d \qquad |\Psi^{AB}\rangle = \sum_{i,j=1}^{d} c_{ij}\, |i^A\rangle \otimes |j^B\rangle$$

$N = 2$

$$\rho^A \equiv tr_B\, [|\Psi^{AB}\rangle\langle\Psi^{AB}|]$$

A

$|\Psi^{AB}\rangle$ is **entangled** $\Leftrightarrow$ ρ^A is **mixed**

$$\rho^A \equiv \sum_{\alpha} p_\alpha\, |\psi_\alpha^A\rangle\langle\psi_\alpha^A|$$

How entangled is $|\Psi^{AB}\rangle$?

measure of entanglement

$$E(\Psi^{AB}) \equiv S(\rho^A) = -\sum_{\alpha} p_\alpha \log_2 p_\alpha$$

von Neumann entropy of ρ^A

example 1: $\frac{1}{\sqrt{2}}|\uparrow^A\rangle \otimes |\uparrow^B\rangle - \frac{1}{\sqrt{2}}|\downarrow^A\rangle \otimes |\downarrow^B\rangle$

Is state $|\Psi^{AB}\rangle$ entangled?

$$\mathcal{H}^A \otimes \mathcal{H}^B = \mathbb{C}_d \otimes \mathbb{C}_d \qquad |\Psi^{AB}\rangle = \sum_{i,j=1}^{d} c_{ij}\, |i^A\rangle \otimes |j^B\rangle$$

$N = 2$

$$\rho^A \equiv tr_B\, [|\Psi^{AB}\rangle\langle\Psi^{AB}|]$$

A

$|\Psi^{AB}\rangle$ is **entangled** $\Leftrightarrow$ ρ^A is **mixed**

$$\rho^A \equiv \sum_{\alpha} p_\alpha\, |\psi_\alpha^A\rangle\langle\psi_\alpha^A|$$

How entangled is $|\Psi^{AB}\rangle$?

measure of entanglement

$$E(\Psi^{AB}) \equiv S(\rho^A) = -\sum_{\alpha} p_\alpha \log_2 p_\alpha$$

von Neumann entropy of ρ^A

example 1: $\frac{1}{\sqrt{2}}|\uparrow^A\rangle \otimes |\uparrow^B\rangle - \frac{1}{\sqrt{2}}|\downarrow^A\rangle \otimes |\downarrow^B\rangle$

$$\rho^A = \frac{1}{2}|\uparrow^A\rangle\langle\uparrow^A| + \frac{1}{2}|\downarrow^A\rangle\langle\downarrow^A|$$

Is state $|\Psi^{AB}\rangle$ entangled?

$$\mathcal{H}^A \otimes \mathcal{H}^B = \mathbb{C}_d \otimes \mathbb{C}_d \qquad |\Psi^{AB}\rangle = \sum_{i,j=1}^{d} c_{ij}\, |i^A\rangle \otimes |j^B\rangle$$

$N = 2$

$$\rho^A \equiv tr_B\, [|\Psi^{AB}\rangle\langle\Psi^{AB}|]$$

$|\Psi^{AB}\rangle$ is **entangled** $\Leftrightarrow$ ρ^A is **mixed**

$$\rho^A \equiv \sum_{\alpha} p_\alpha\, |\psi_\alpha^A\rangle\langle\psi_\alpha^A|$$

How entangled is $|\Psi^{AB}\rangle$?

measure of entanglement

$$E(\Psi^{AB}) \equiv S(\rho^A) = -\sum_{\alpha} p_\alpha \log_2 p_\alpha$$

von Neumann entropy of ρ^A

example 1: $\frac{1}{\sqrt{2}}|\uparrow^A\rangle \otimes |\uparrow^B\rangle - \frac{1}{\sqrt{2}}|\downarrow^A\rangle \otimes |\downarrow^B\rangle$

$$\rho^A = \frac{1}{2}|\uparrow^A\rangle\langle\uparrow^A| + \frac{1}{2}|\downarrow^A\rangle\langle\downarrow^A|$$

$$E(\Psi^{AB}) = 1$$

1 unit of entanglement

Is state $|\Psi^{AB}\rangle$ entangled?

$$\mathcal{H}^A \otimes \mathcal{H}^B = \mathbb{C}_d \otimes \mathbb{C}_d \qquad |\Psi^{AB}\rangle = \sum_{i,j=1}^{d} c_{ij}\, |i^A\rangle \otimes |j^B\rangle$$

$N = 2$

$$\rho^A \equiv tr_B\, [|\Psi^{AB}\rangle\langle\Psi^{AB}|]$$

A

$|\Psi^{AB}\rangle$ is **entangled** $\Leftrightarrow$ ρ^A is **mixed**

$$\rho^A \equiv \sum_\alpha p_\alpha\, |\psi_\alpha^A\rangle\langle\psi_\alpha^A|$$

How entangled is $|\Psi^{AB}\rangle$?

measure of entanglement

$$E(\Psi^{AB}) \equiv S(\rho^A) = -\sum_\alpha p_\alpha \log_2 p_\alpha$$

von Neumann entropy of ρ^A

example 1: $\frac{1}{\sqrt{2}}|\uparrow^A\rangle \otimes |\uparrow^B\rangle - \frac{1}{\sqrt{2}}|\downarrow^A\rangle \otimes |\downarrow^B\rangle$

$$\rho^A = \frac{1}{2}|\uparrow^A\rangle\langle\uparrow^A| + \frac{1}{2}|\downarrow^A\rangle\langle\downarrow^A|$$

$$E(\Psi^{AB}) = 1$$

1 unit of entanglement

example 2: $|\uparrow^A\rangle \otimes |\uparrow^B\rangle$

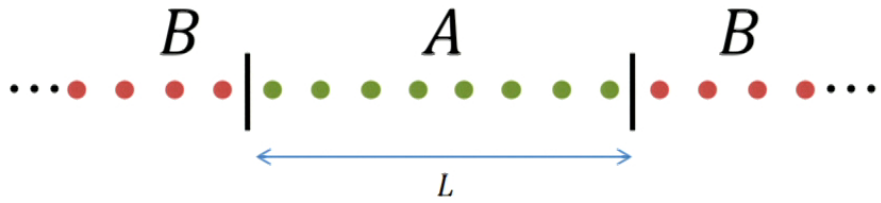
$$\rho^A = |\uparrow^A\rangle\langle\uparrow^A|$$

$$E(\Psi^{AB}) = 0$$

0 units of entanglement

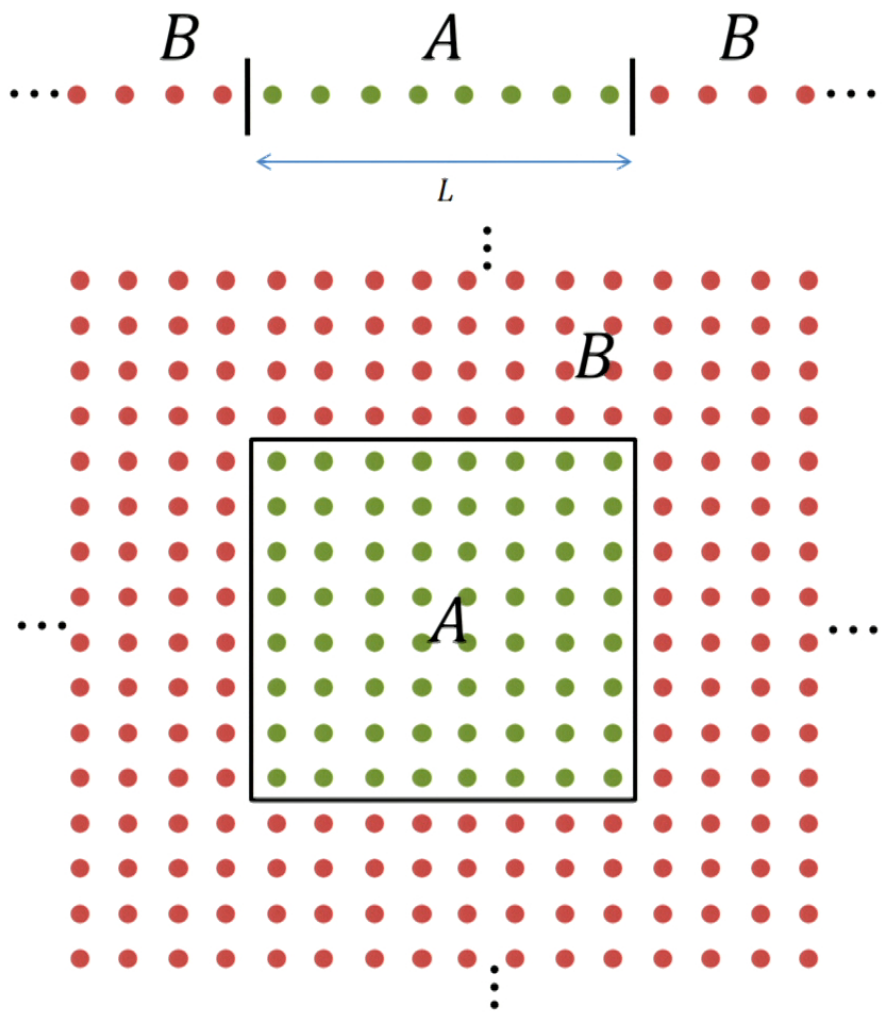
Scaling of entanglement in many-body systems

$\mathcal{H}^A \otimes \mathcal{H}^B$



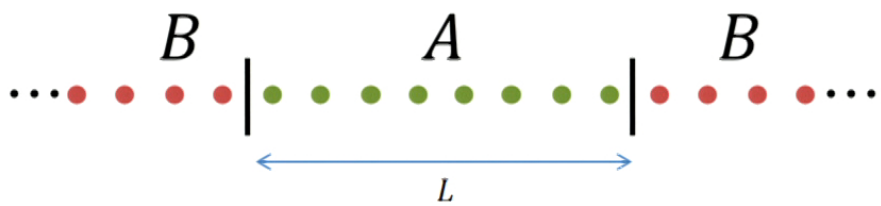
Scaling of entanglement in many-body systems

$\mathcal{H}^A \otimes \mathcal{H}^B$



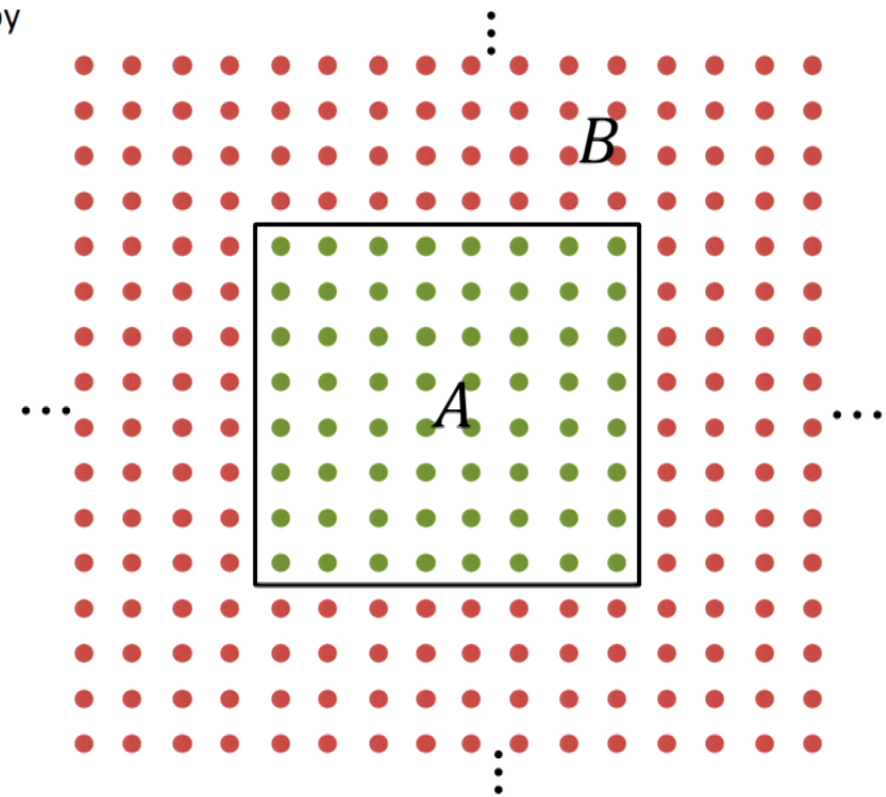
Scaling of entanglement in many-body systems

$\mathcal{H}^A \otimes \mathcal{H}^B$



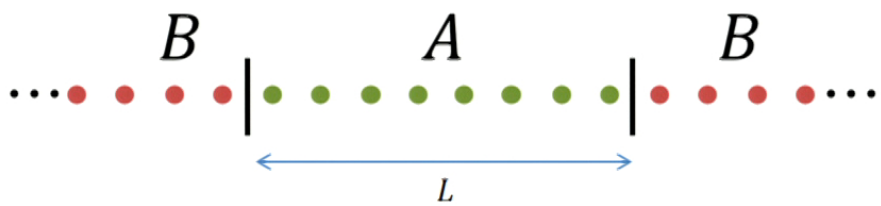
How does entanglement entropy scale with L ?

- generic state of $\mathcal{H}^A \otimes \mathcal{H}^B$



Scaling of entanglement in many-body systems

$\mathcal{H}^A \otimes \mathcal{H}^B$

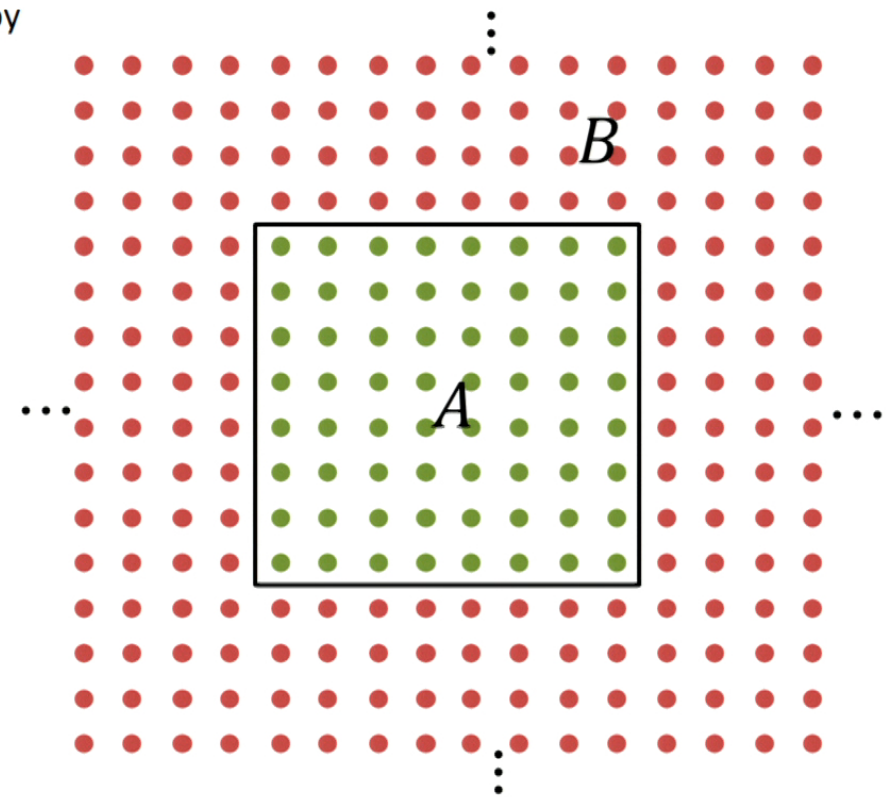


How does entanglement entropy scale with L ?

- generic state of $\mathcal{H}^A \otimes \mathcal{H}^B$

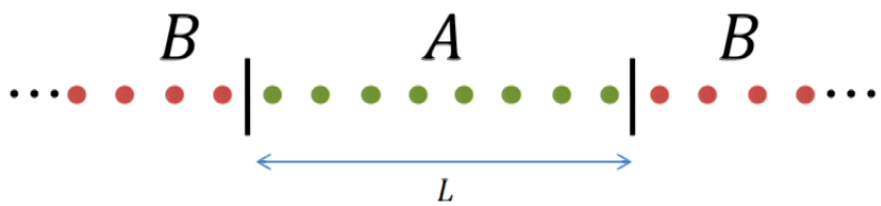
$S(A) \sim |A| = L^D$

volume law



Scaling of entanglement in many-body systems

$\mathcal{H}^A \otimes \mathcal{H}^B$



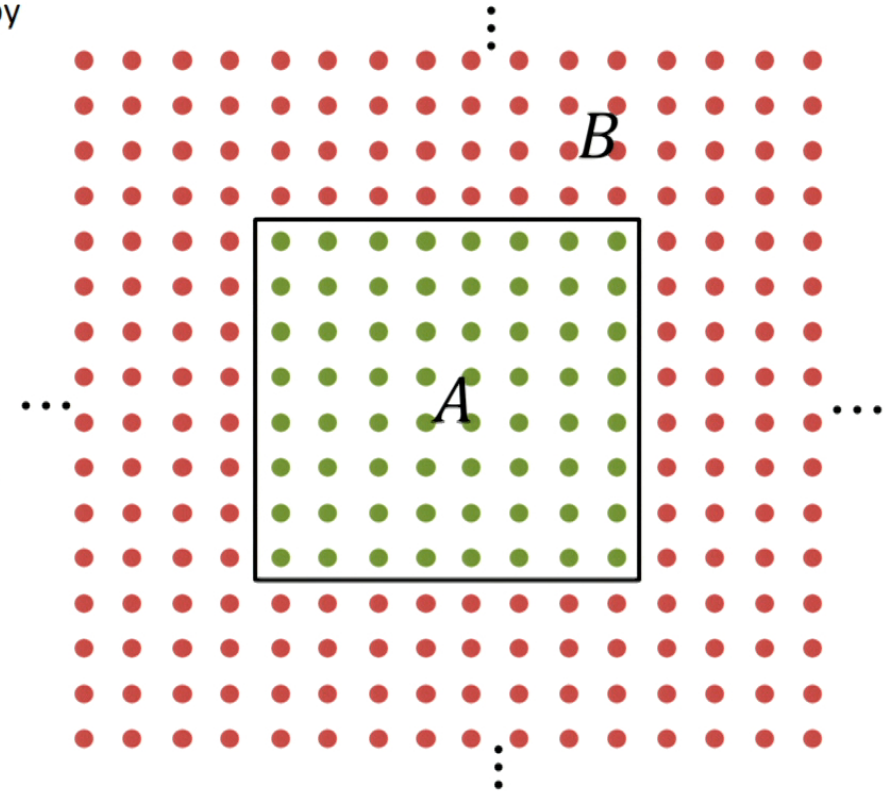
How does entanglement entropy scale with L ?

- generic state of $\mathcal{H}^A \otimes \mathcal{H}^B$

$S(A) \sim |A| = L^D$

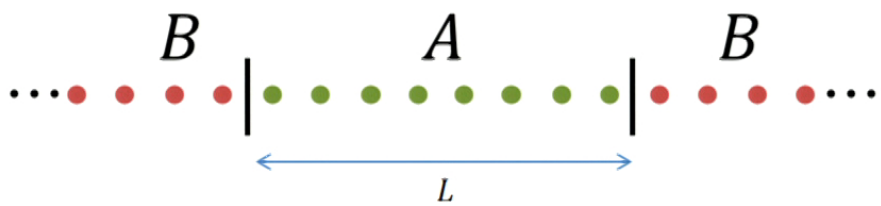
volume law

- ground state of local Hamiltonian



Scaling of entanglement in many-body systems

$\mathcal{H}^A \otimes \mathcal{H}^B$



How does entanglement entropy scale with L ?

- generic state of $\mathcal{H}^A \otimes \mathcal{H}^B$

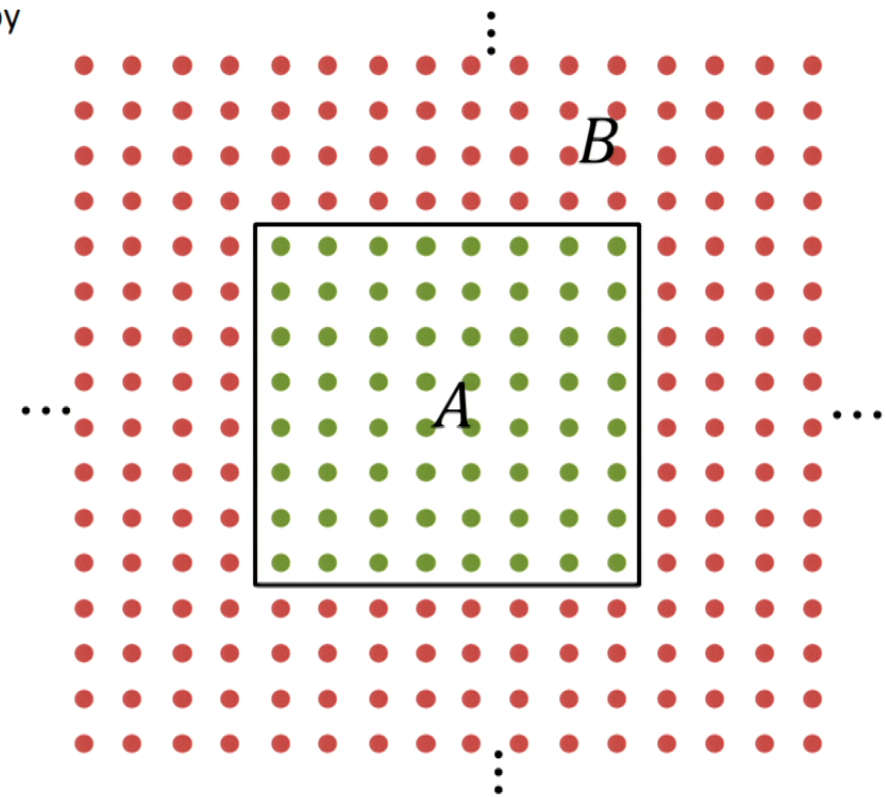
$S(A) \sim |A| = L^D$

volume law

- ground state of local Hamiltonian

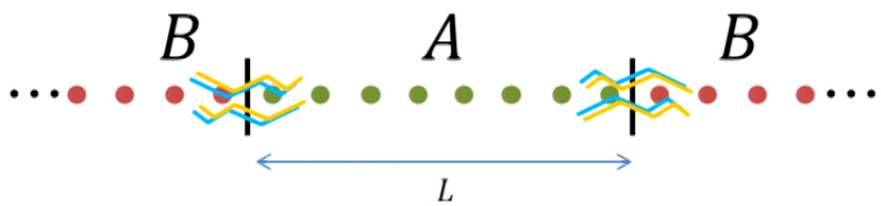
$S(A) \sim |\partial A| = L^{D-1}$

area law



Scaling of entanglement in many-body systems

$\mathcal{H}^A \otimes \mathcal{H}^B$



How does entanglement entropy scale with L ?

- generic state of $\mathcal{H}^A \otimes \mathcal{H}^B$

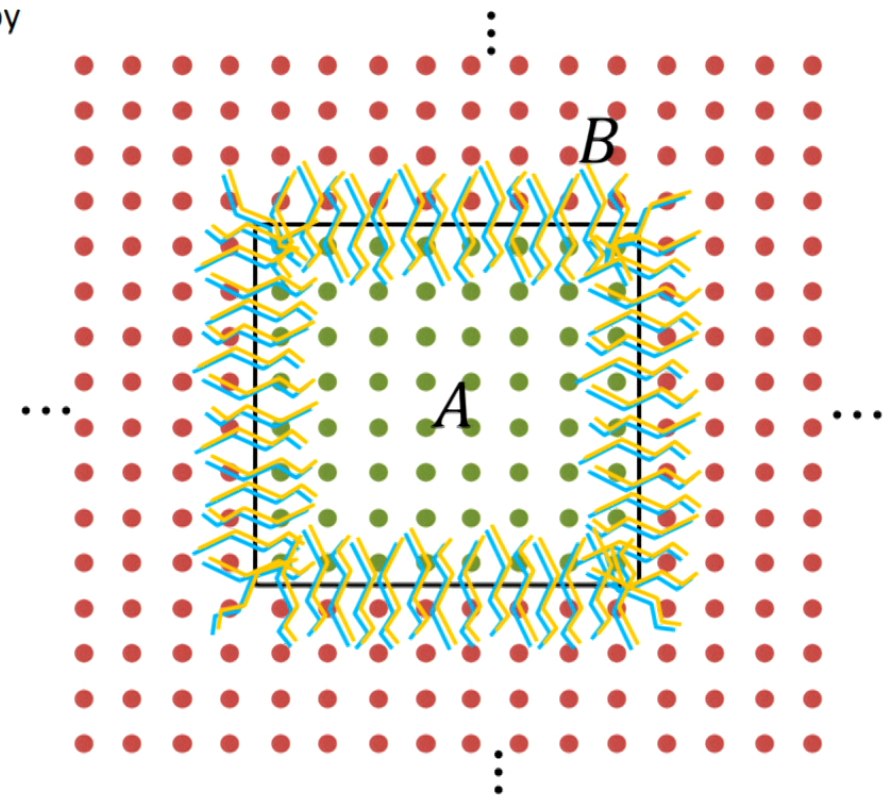
$S(A) \sim |A| = L^D$

volume law

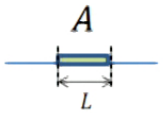
- ground state of local Hamiltonian

$S(A) \sim |\partial A| = L^{D-1}$

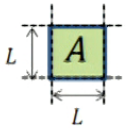
area law



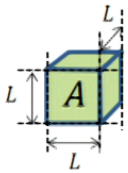
D=1



D=2



D=3



area law
for entanglement entropy

$$S(A) \sim |\partial A| \sim L^{D-1}$$

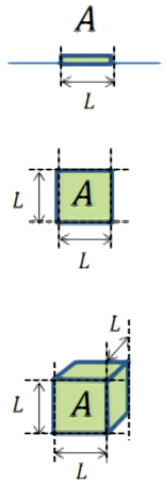
instead of
volume law

$$S(A) \sim |A| \sim L^{D}$$

D=1

D=2

D=3



area law
for entanglement entropy

$$S(A) \sim |\partial A| \sim L^{D-1}$$

sometimes,
logarithmic corrections

$$S(A) \sim L^{D-1} \log(L)$$

instead of
volume law

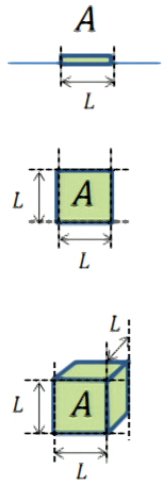
$$S(A) \sim |A| \sim L^{D}$$

Ground states of local Hamiltonians are special/non-generic states

D=1

D=2

D=3



area law
for entanglement entropy

$$S(A) \sim |\partial A| \sim L^{D-1}$$

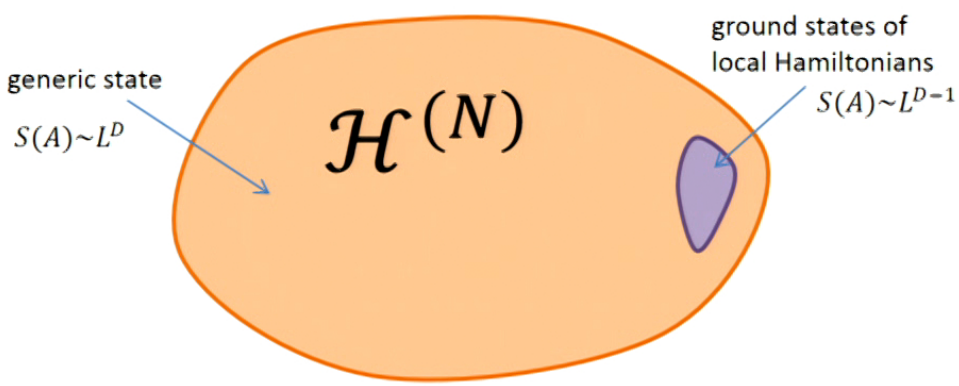
instead of
volume law

$$S(A) \sim |A| \sim L^{D}$$

sometimes,
logarithmic corrections

$$S(A) \sim L^{D-1} \log(L)$$

Ground states of local Hamiltonians are special/non-generic states

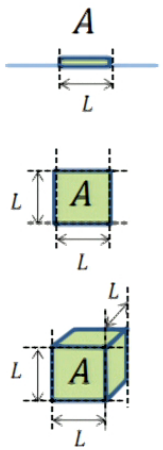


Scaling of entanglement in many-body ground states (complete list?)

$D = 1$

$D = 2$

$D = 3$

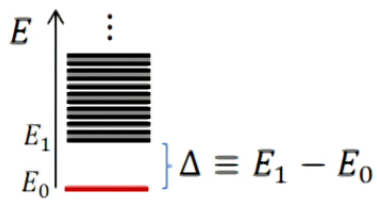


$S(A) \sim L^{D-1}$

area law

$S(A) \sim L^{D-1} \log(L)$

Scaling of entanglement in many-body ground states (complete list?)

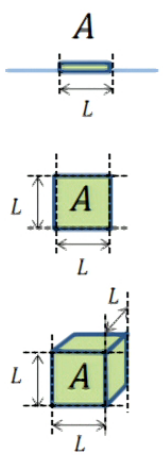


gapped system

$D = 1$

$D = 2$

$D = 3$

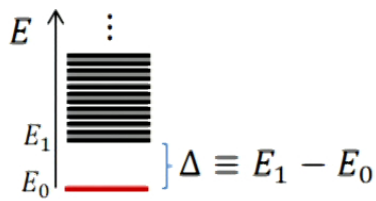


$S(A) \sim L^{D-1}$

area law

$S(A) \sim L^{D-1} \log(L)$

Scaling of entanglement in many-body ground states (complete list?)

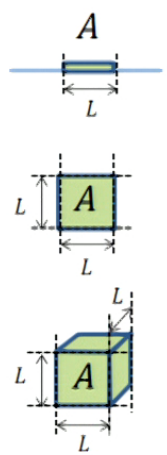


gapped system

$D = 1$

$D = 2$

$D = 3$



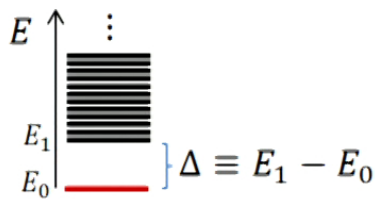
const.	
L	
L^2	

$S(A) \sim L^{D-1}$

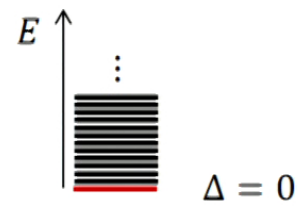
area law

$S(A) \sim L^{D-1} \log(L)$

Scaling of entanglement in many-body ground states (complete list?)



gapped system

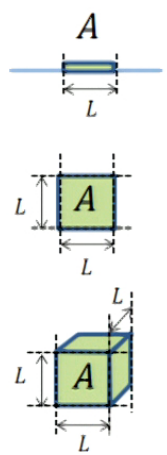


gapless system

$D = 1$

$D = 2$

$D = 3$



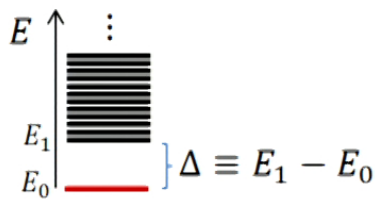
const.	
L	
L^2	

$S(A) \sim L^{D-1}$

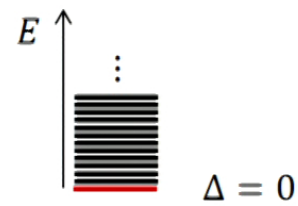
area law

$S(A) \sim L^{D-1} \log(L)$

Scaling of entanglement in many-body ground states (complete list?)

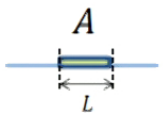
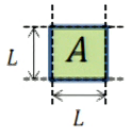
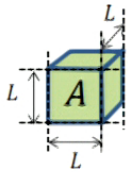


gapped system



gapless system

$D - D_\Gamma \geq 2$

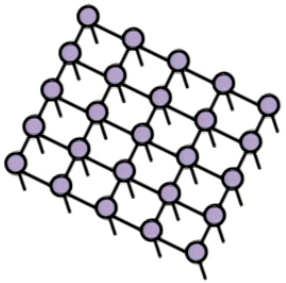
$D = 1$		const.	-
$D = 2$		L	L
$D = 3$		L^2	L^2

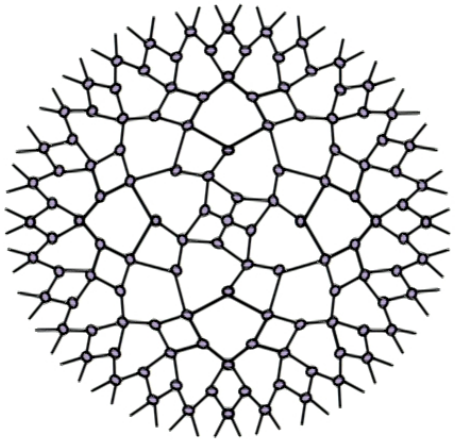
$S(A) \sim L^{D-1}$
area law

$S(A) \sim L^{D-1} \log(L)$

Part 3:

Tensor network states





Tensor network state: variational many-body wavefunction

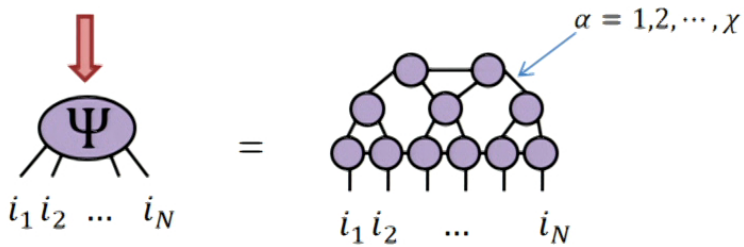
many-body wave-function

$$|\Psi\rangle = \sum_{i_1, i_2, \cdots, i_N} \Psi_{i_1 i_2 \cdots i_N} \, |i_1 i_2 \cdots i_N\rangle$$

Tensor network state: variational many-body wavefunction

many-body wave-function

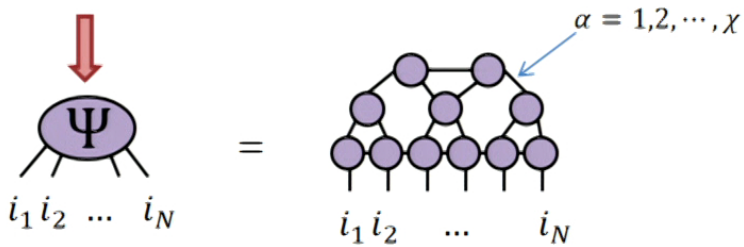
$$|\Psi\rangle = \sum_{i_1, i_2, \cdots, i_N} \Psi_{i_1 i_2 \cdots i_N} |i_1 i_2 \cdots i_N\rangle$$



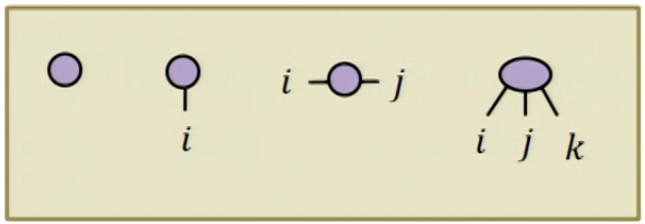
Tensor network state: variational many-body wavefunction

many-body wave-function

$$|\Psi\rangle = \sum_{i_1, i_2, \cdots, i_N} \Psi_{i_1 i_2 \cdots i_N} |i_1 i_2 \cdots i_N\rangle$$



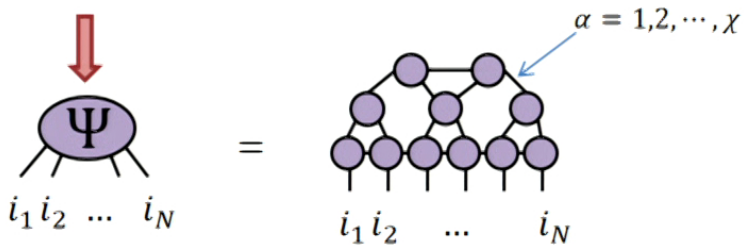
graphical notation



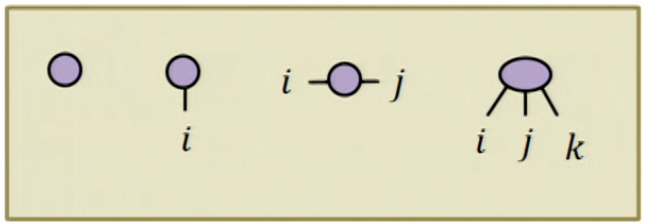
Tensor network state: variational many-body wavefunction

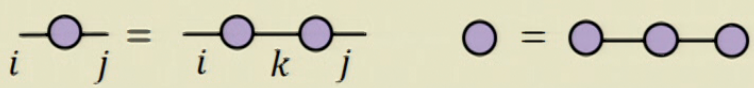
many-body wave-function

$$|\Psi\rangle = \sum_{i_1, i_2, \cdots, i_N} \Psi_{i_1 i_2 \cdots i_N} |i_1 i_2 \cdots i_N\rangle$$



graphical notation



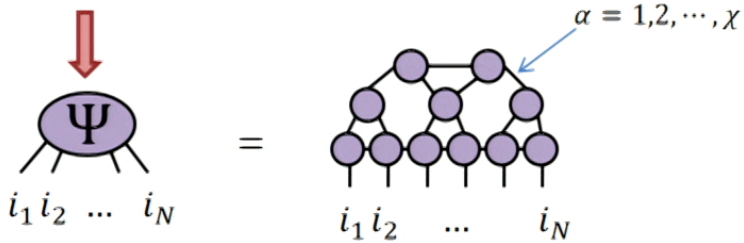


$$T_{ij} = \sum_k R_{ik} S_{kj} \qquad a = \vec{y}^{\dagger} \cdot M \cdot \vec{x}$$

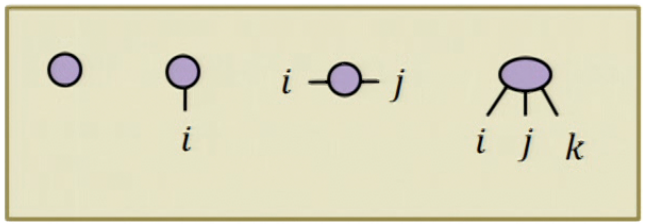
Tensor network state: variational many-body wavefunction

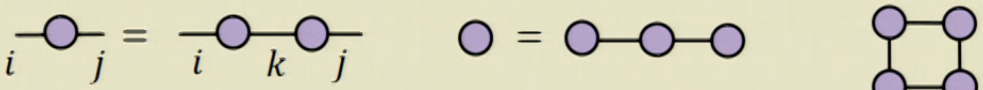
many-body wave-function

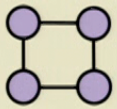
$$|\Psi\rangle = \sum_{i_1, i_2, \cdots, i_N} \Psi_{i_1 i_2 \cdots i_N} |i_1 i_2 \cdots i_N\rangle$$



graphical notation





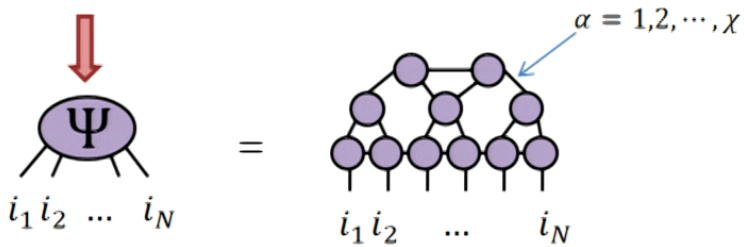


$$T_{ij} = \sum_k R_{ik} S_{kj} \qquad a = \vec{y}^{\dagger} \cdot M \cdot \vec{x}$$

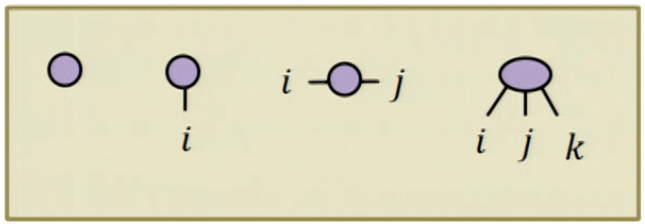
Tensor network state: variational many-body wavefunction

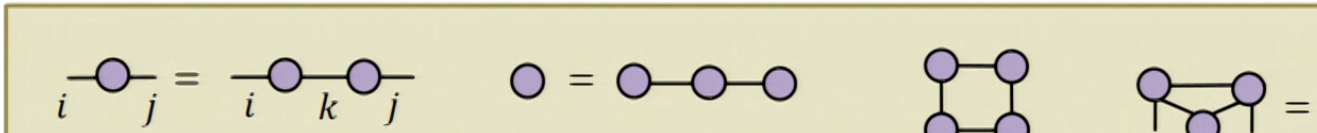
many-body wave-function

$$|\Psi\rangle = \sum_{i_1, i_2, \cdots, i_N} \Psi_{i_1 i_2 \cdots i_N} |i_1 i_2 \cdots i_N\rangle$$



graphical notation





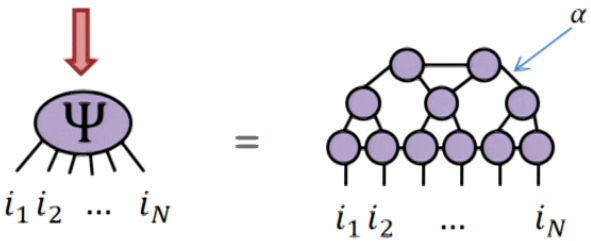
$$T_{ij} = \sum_k R_{ik} S_{kj}$$

$$a = \vec{y}^{\dagger} \cdot M \cdot \vec{x}$$

$$tr(ABCD)$$

$$|\Psi\rangle = \sum_{i_1, i_2, \cdots, i_N} \Psi_{i_1 i_2 \cdots i_N} |i_1 i_2 \cdots i_N\rangle$$

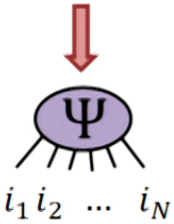
1 tensor
of size 2^N



$O(N)$ tensors
of size χ^p independent of N

$$|\Psi\rangle = \sum_{i_1, i_2, \cdots, i_N} \Psi_{i_1 i_2 \cdots i_N} |i_1 i_2 \cdots i_N\rangle$$

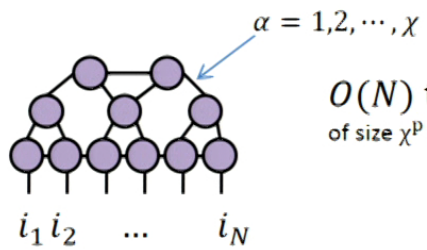
1 tensor
of size 2^N



2^N
parameters

inefficient

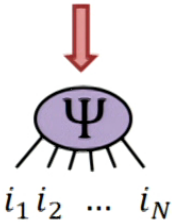
=



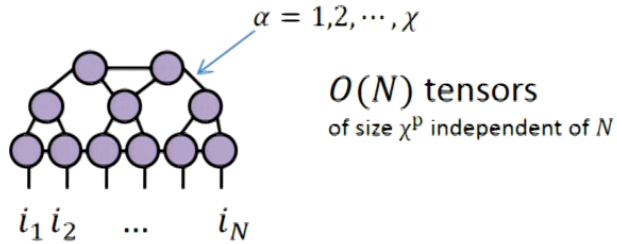
$O(N)$ tensors
of size χ^p independent of N

$$|\Psi\rangle = \sum_{i_1, i_2, \cdots, i_N} \Psi_{i_1 i_2 \cdots i_N} |i_1 i_2 \cdots i_N\rangle$$

1 tensor
of size 2^N



=



$O(N)$ tensors
of size χ^p independent of N

2^N
parameters

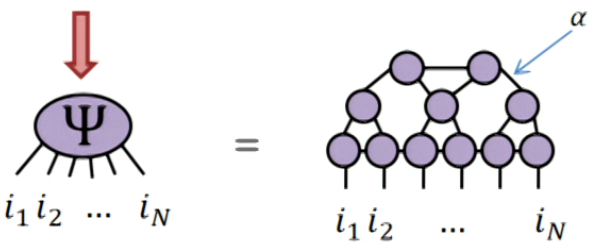
inefficient

$O(N)$
parameters

efficient

$$|\Psi\rangle = \sum_{i_1, i_2, \cdots, i_N} \Psi_{i_1 i_2 \cdots i_N} |i_1 i_2 \cdots i_N\rangle$$

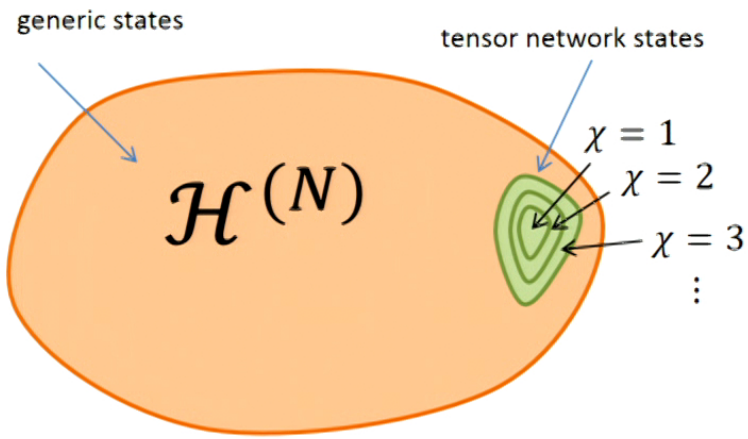
1 tensor
of size 2^N



$O(N)$ tensors
of size χ^p independent of N

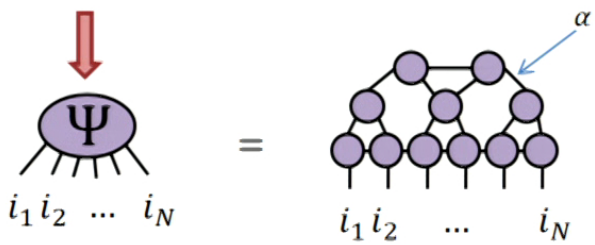
2^N
parameters
inefficient

$O(N)$
parameters
efficient



$$|\Psi\rangle = \sum_{i_1, i_2, \cdots, i_N} \Psi_{i_1 i_2 \cdots i_N} |i_1 i_2 \cdots i_N\rangle$$

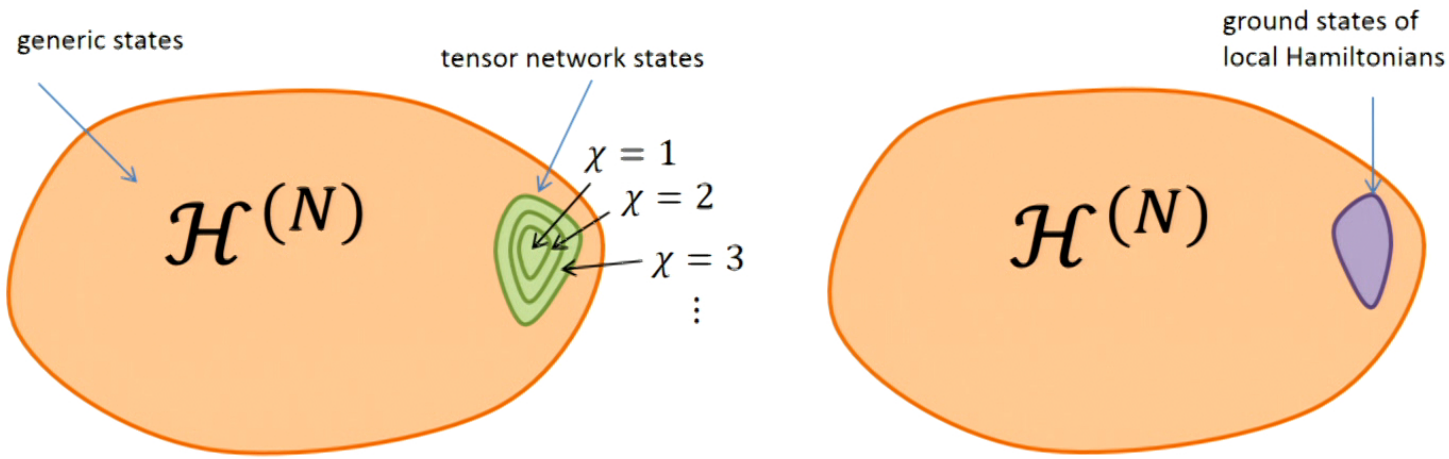
1 tensor
of size 2^N



$O(N)$ tensors
of size χ^p independent of N

2^N
parameters
inefficient

$O(N)$
parameters
efficient



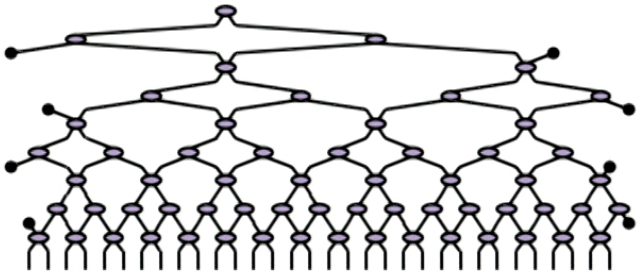
Examples of tensor networks in $D = 1$ space dimensions

matrix product state

(MPS)

multi-scale entanglement renormalization ansatz

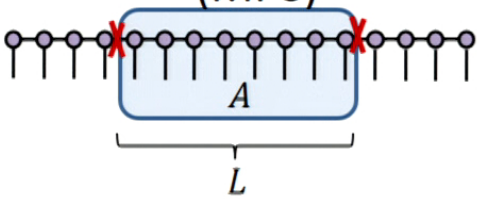
(MERA)



scaling of entanglement

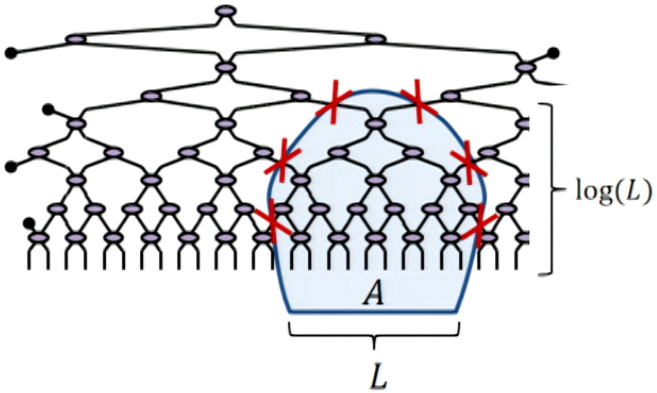
matrix product state

(MPS)



multi-scale entanglement renormalization ansatz

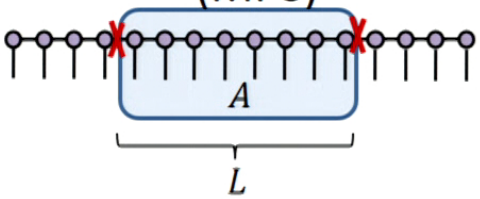
(MERA)



scaling of entanglement

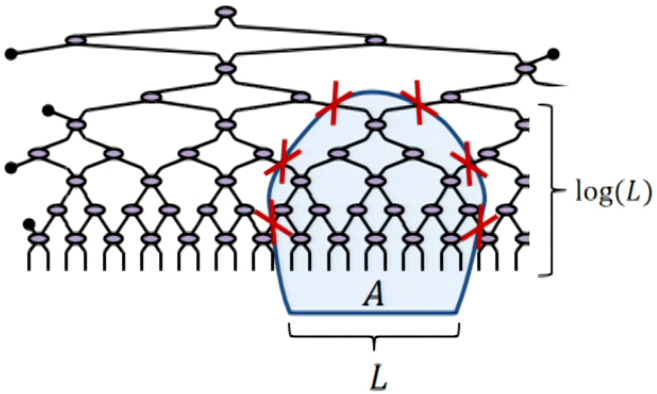
matrix product state

(MPS)



multi-scale entanglement renormalization ansatz

(MERA)



network connectivity:

$S(A) \approx const$

area law!

just as ground state of

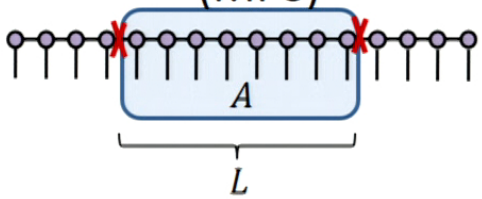
massive/gapped

1d system

scaling of entanglement

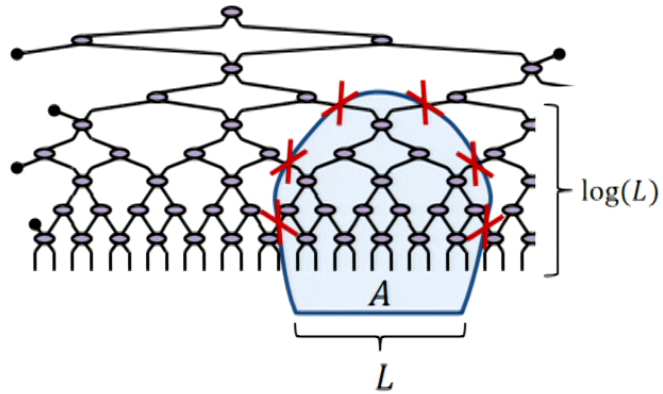
matrix product state

(MPS)



multi-scale entanglement renormalization ansatz

(MERA)



network connectivity:

$S(A) \approx const$

area law!

just as ground state of
massive/gapped
1d system

$S(A) \approx \log L$

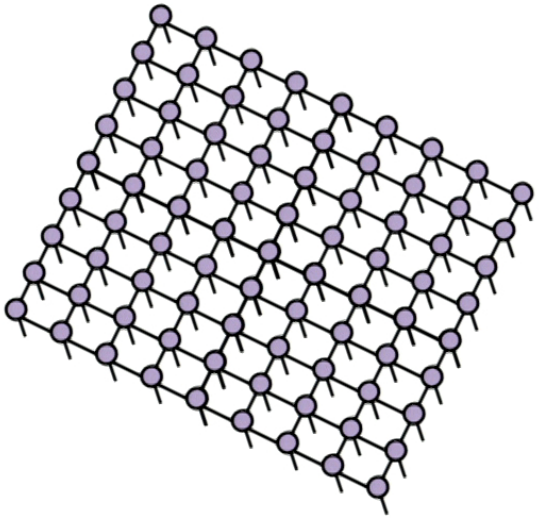
logarithmic correction!

just as ground state of
critical/gapless
1d system

Example of tensor network in $D = 2$ space dimensions

Matrix Product State
MPS
$D = 1$ dimensions

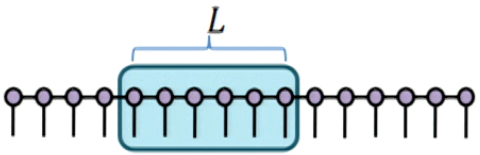
Projected entangled pair states
PEPS
$D = 2$ dimensions



Example of tensor network in $D = 2$ space dimensions

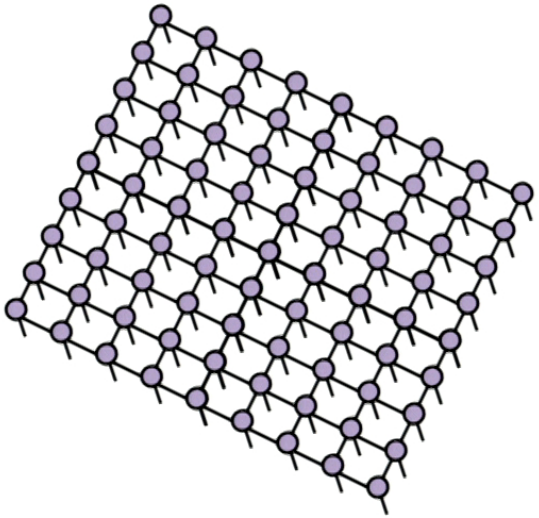
Matrix Product State
MPS

$D = 1$ dimensions



Projected entangled pair states
PEPS

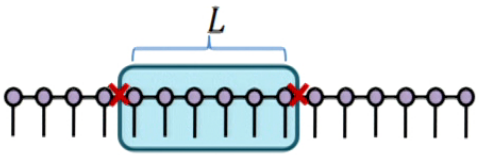
$D = 2$ dimensions



Example of tensor network in $D = 2$ space dimensions

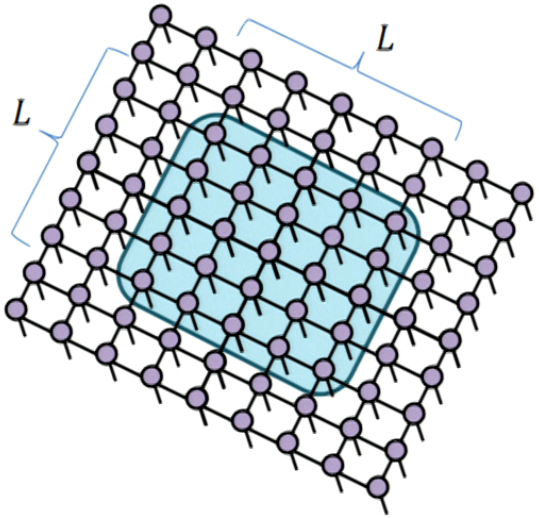
Matrix Product State
MPS

$D = 1$ dimensions



Projected entangled pair states
PEPS

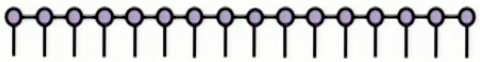
$D = 2$ dimensions



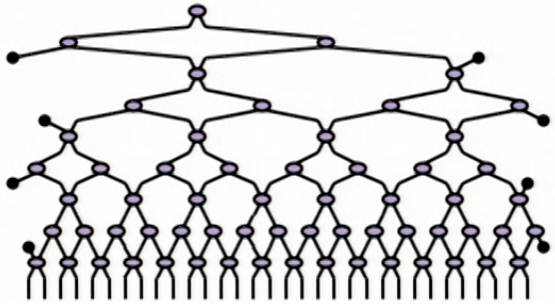
$$S(A) \sim L^0 = const.$$

$D = 1$ space dimensions

matrix product state
(MPS)



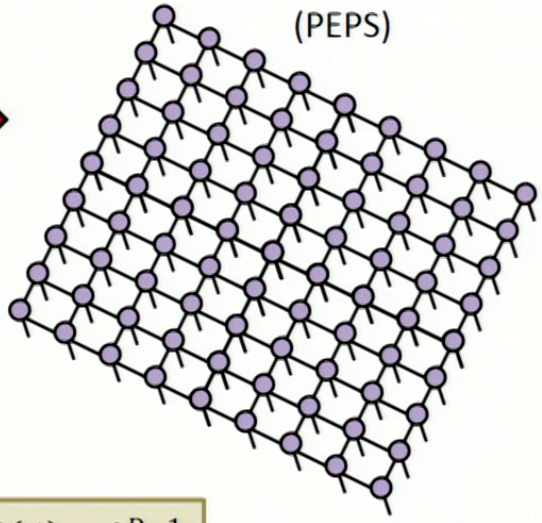
$S(A) \sim const$
area law



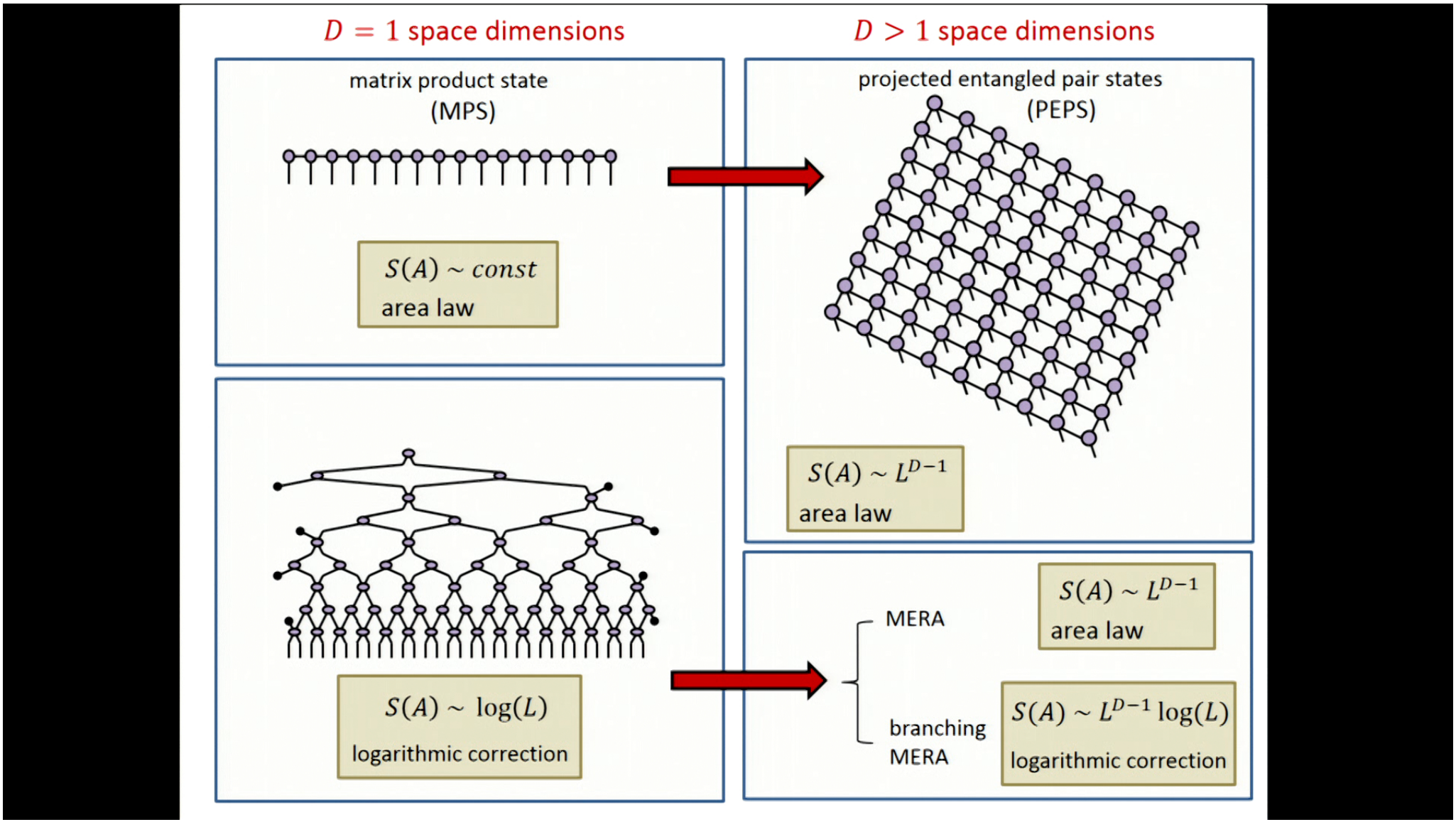
$S(A) \sim \log(L)$
logarithmic correction

$D > 1$ space dimensions

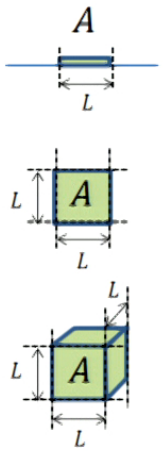
projected entangled pair states
(PEPS)



$S(A) \sim L^{D-1}$
area law

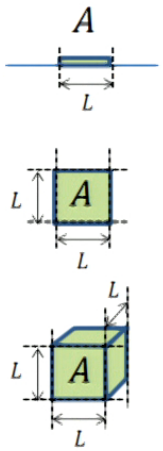

$D = 1$ space dimensions
$D > 1$ space dimensions
matrix product state
(MPS)
$S(A) \sim const$
area law
projected entangled pair states
(PEPS)
$S(A) \sim L^{D-1}$
area law
$S(A) \sim \log(L)$
logarithmic correction
MERA
$S(A) \sim L^{D-1}$
area law
branching
MERA
$S(A) \sim L^{D-1} \log(L)$
logarithmic correction

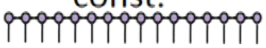
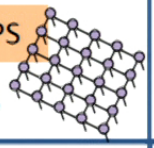
Scaling of entanglement and tensor network states



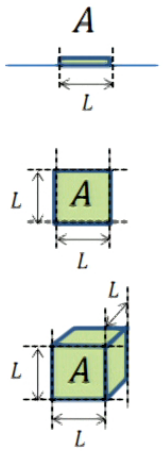
	gapped system	gapless system	
		$D - D_\Gamma \geq 2$	$D - D_\Gamma = 1$
$D = 1$	const.	-	$\log(L)$
$D = 2$	L	L	$L \log(L)$
$D = 3$	L^2	L^2	$L^2 \log(L)$

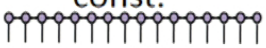
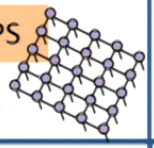
Scaling of entanglement and tensor network states



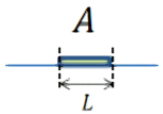
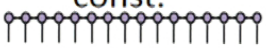
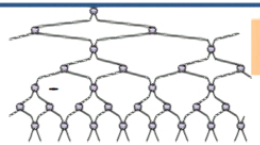
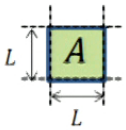
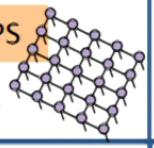
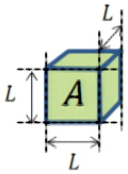
	gapped system	gapless system	
		$D - D_\Gamma \geq 2$	$D - D_\Gamma = 1$
$D = 1$	MPS const. 	-	$\log(L)$
$D = 2$	PEPS  L	L	$L \log(L)$
$D = 3$	PEPS L^2	L^2	$L^2 \log(L)$

Scaling of entanglement and tensor network states

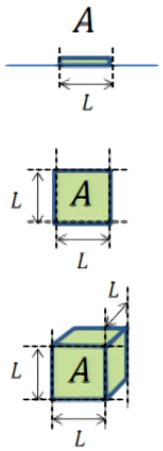


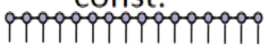
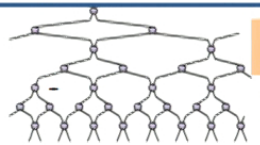
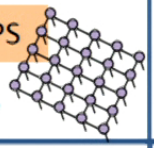
	gapped system	gapless system	
		$D - D_\Gamma \geq 2$	$D - D_\Gamma = 1$
$D = 1$	MPS const. 	-	$\log(L)$
$D = 2$	PEPS  L	L	$L \log(L)$
$D = 3$	PEPS L^2	L^2	$L^2 \log(L)$

Scaling of entanglement and tensor network states

		gapped system	gapless system $D - D_\Gamma \geq 2$	gapless system $D - D_\Gamma = 1$
$D = 1$		MPS const. 	 MERA $\log(L)$	
$D = 2$		PEPS L 	MERA L	$L \log(L)$
$D = 3$		PEPS L^2	MERA L^2	$L^2 \log(L)$

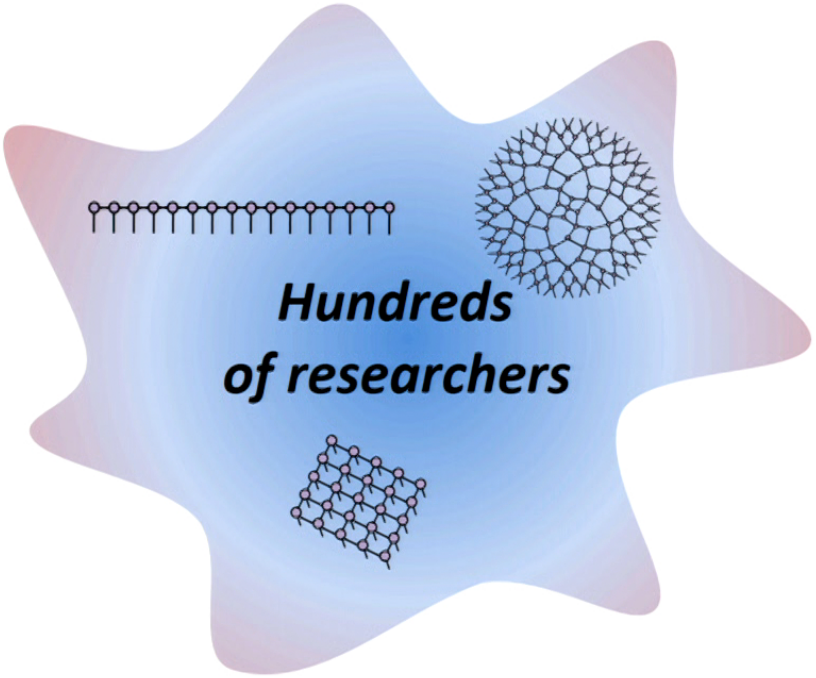
Scaling of entanglement and tensor network states



	gapped system	gapless system $D - D_\Gamma \geq 2$	gapless system $D - D_\Gamma = 1$
$D = 1$	MPS const. 	 MERA $\log(L)$	
$D = 2$	PEPS  L	MERA L	branching MERA $L \log(L)$
$D = 3$	PEPS L^2	MERA L^2	branching MERA $L^2 \log(L)$

CURRENT APPLICATIONS

tensor network = sparse data structure

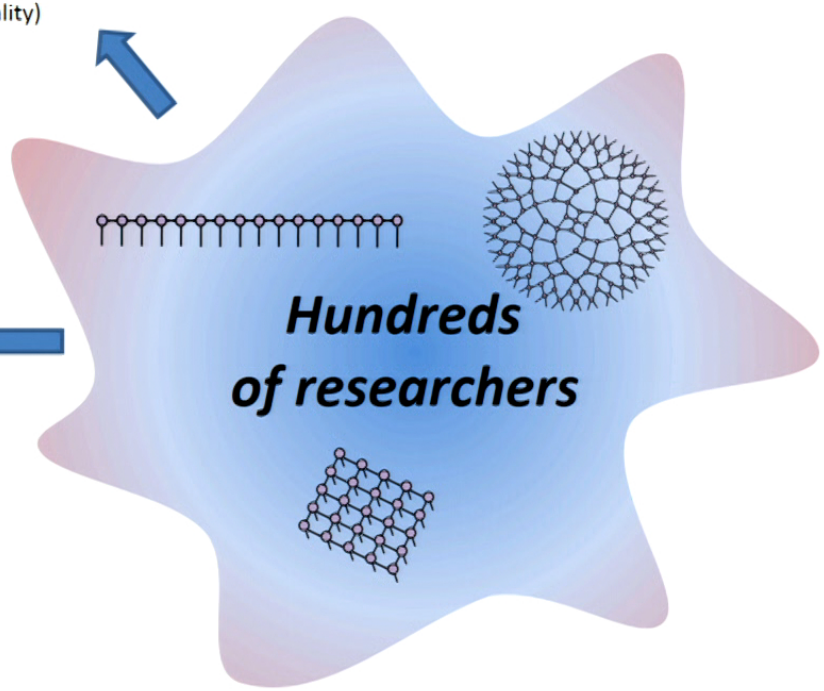


CURRENT APPLICATIONS

tensor network = sparse data structure

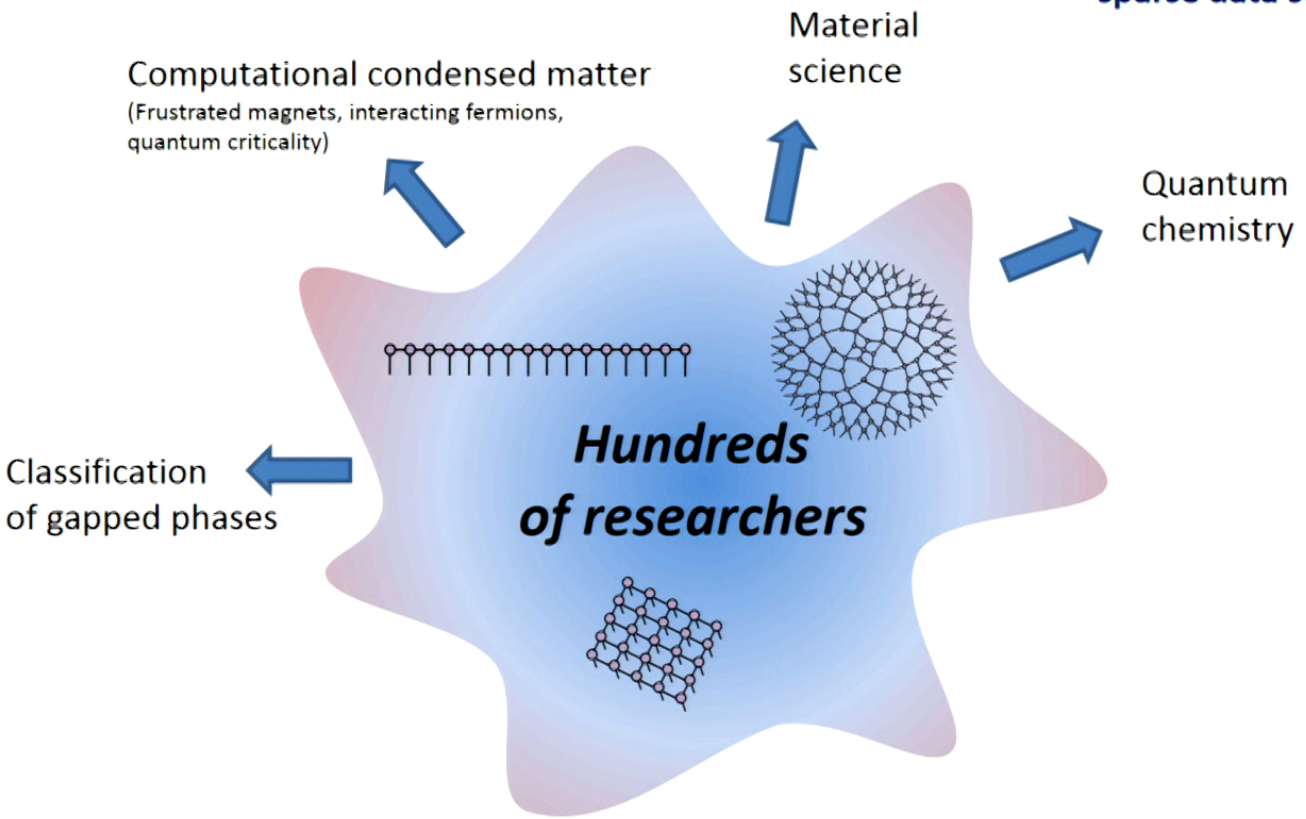
Computational condensed matter
(Frustrated magnets, interacting fermions, quantum criticality)

Classification of gapped phases



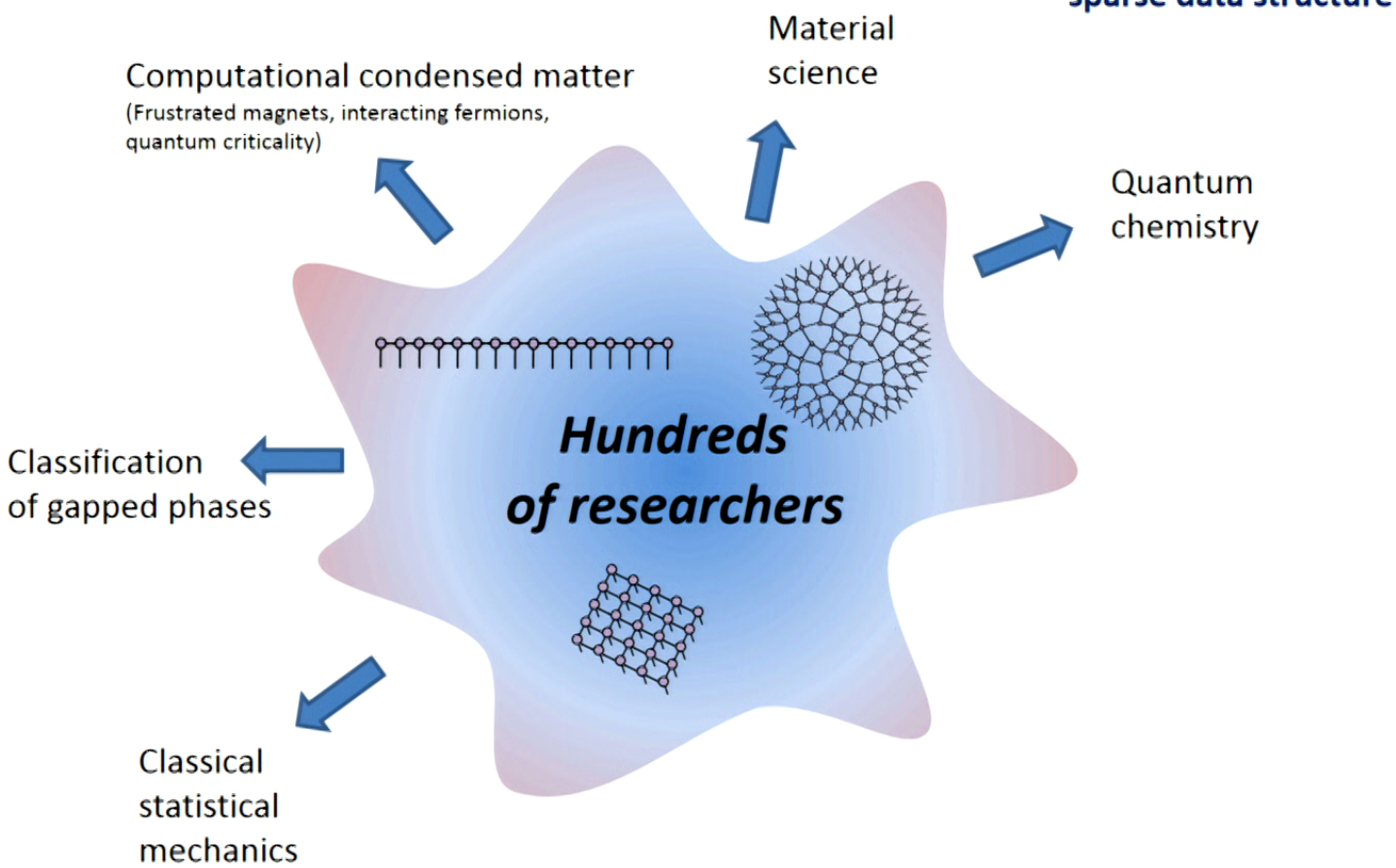
CURRENT APPLICATIONS

tensor network = sparse data structure



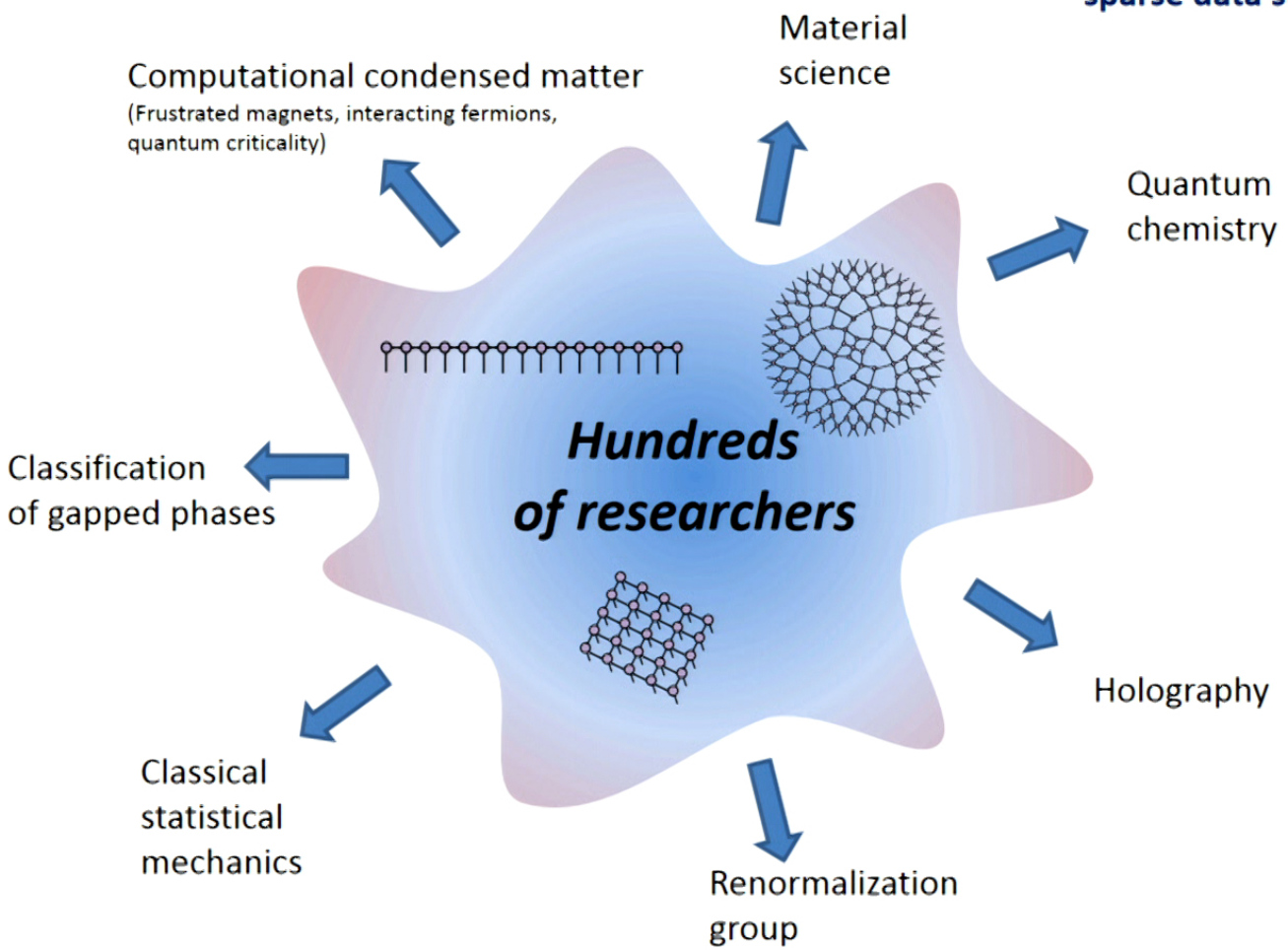
CURRENT APPLICATIONS

tensor network = sparse data structure



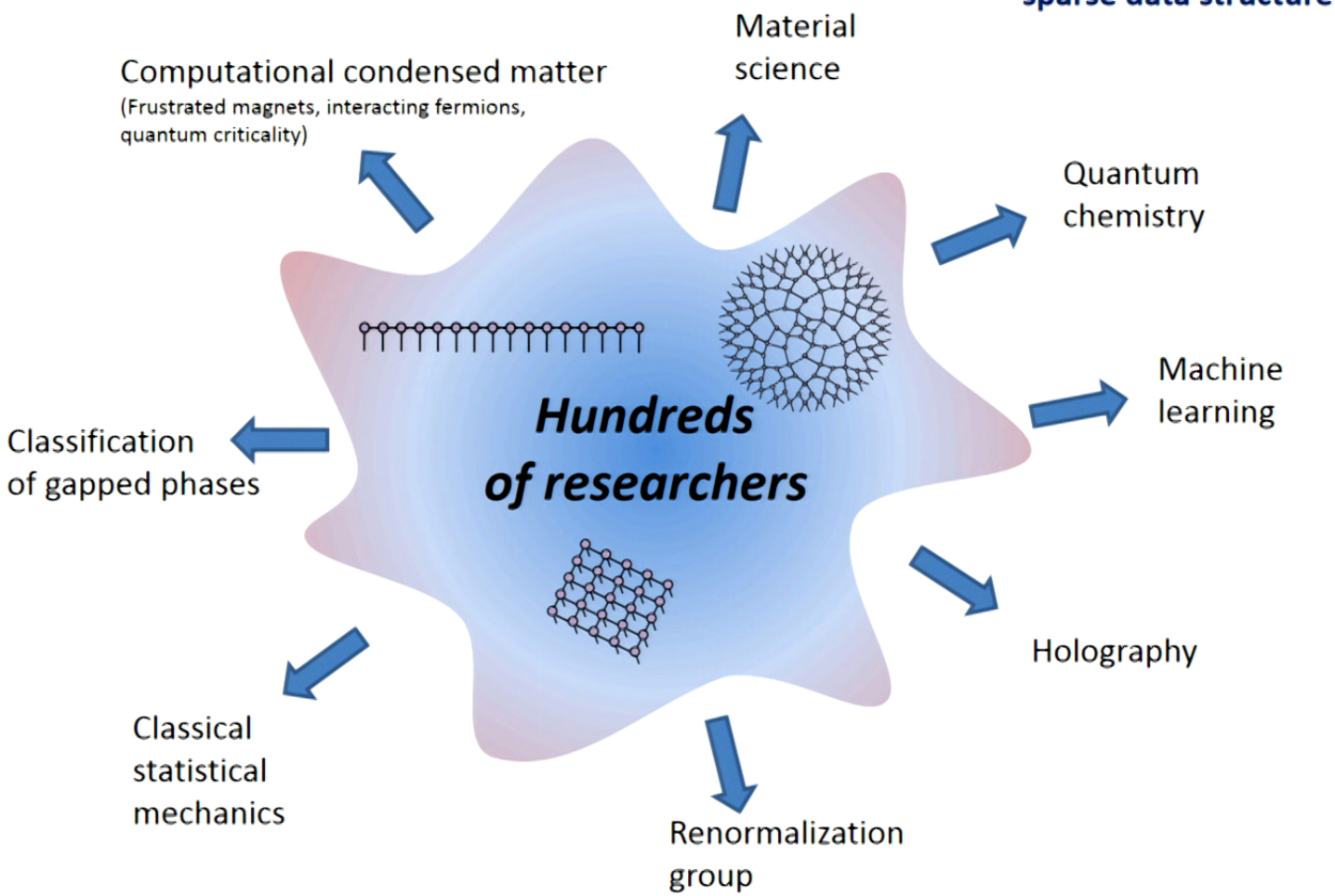
CURRENT APPLICATIONS

tensor network = sparse data structure



CURRENT APPLICATIONS

tensor network = sparse data structure



Current applications

Example 1: Variational approach to many-body systems

Current applications

Example 1: Variational approach to many-body systems

- frustrated antiferromagnets (2D)
- interacting fermions (2D)
- topologically ordered phases (2D)
- quantum criticality/phase transitions
- quantum chemistry

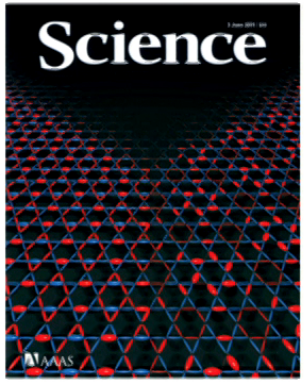
no sign problem!

Current applications

Example 1: Variational approach to many-body systems

- frustrated antiferromagnets (2D)
- interacting fermions (2D)
- topologically ordered phases (2D)
- quantum criticality/phase transitions
- quantum chemistry

no sign problem!

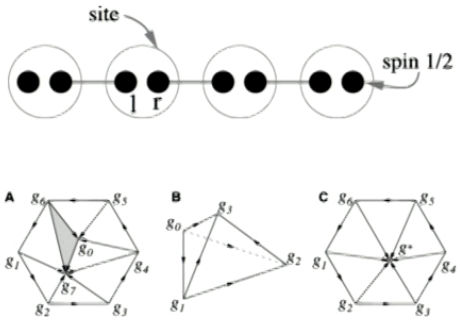


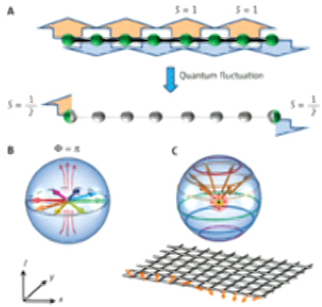
"Spin-liquid ground state of the $s = 1/2$ Kagome Heisenberg Antiferromagnet"
S. Yan, D. A. Huse, S. R. White, **Science (2011)**

Current applications

Example 2: Classification of phases of quantum matter

Theoretical framework to characterize/classify many-body states



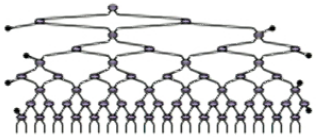


"Symmetry protected topological orders in interacting bosonic systems"
X. Chen, Z.-C. Gu, Z.-X. Liu, X.-G. Wen, **Science (2012)**

Current applications

Example 3: Holographic principle

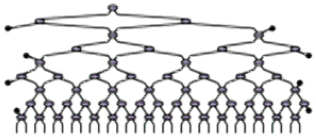
(my HoloClub talk on Tuesday)



Current applications

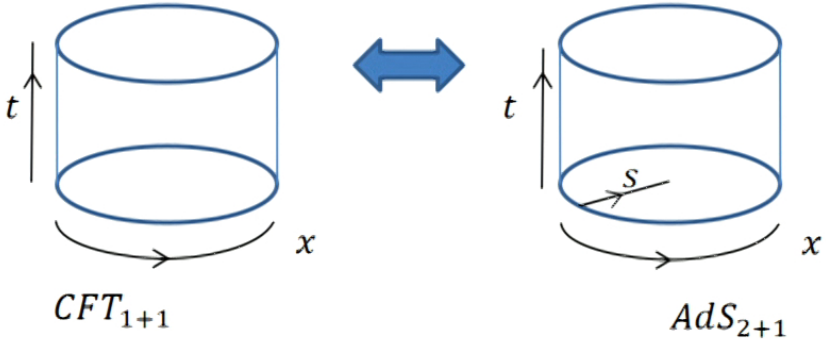
Example 3: Holographic principle

(my HoloClub talk on Tuesday)



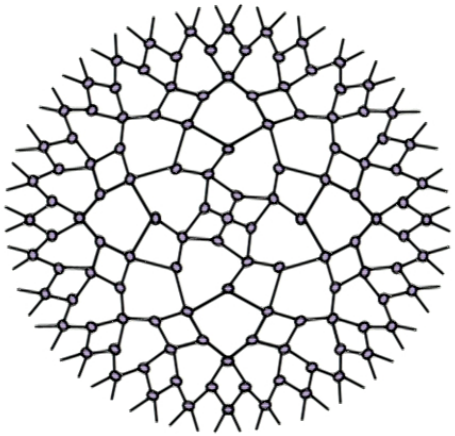
MERA =? lattice realization of the famous AdS/CFT correspondence of string theory

AdS/CFT correspondence, J. Maldacena, 1997



quantum field theory

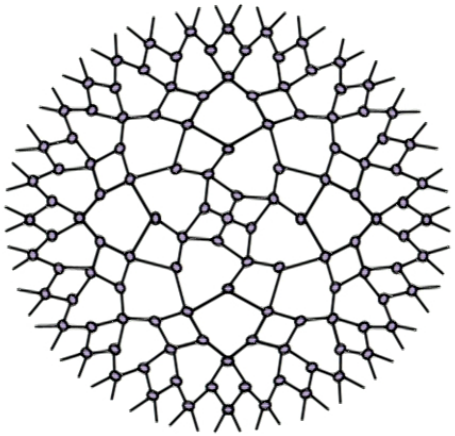
quantum gravity

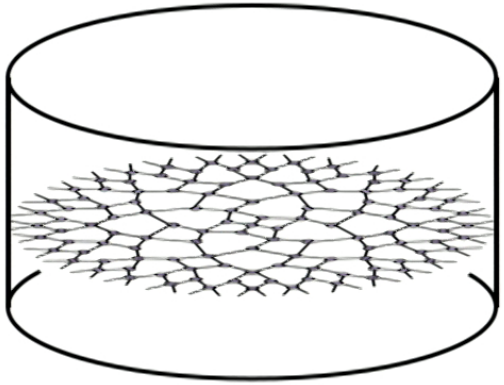


AdS_3/CFT_2

Swingle 2009, 2012

time slice of AdS_3 (hyperbolic plane H_2)

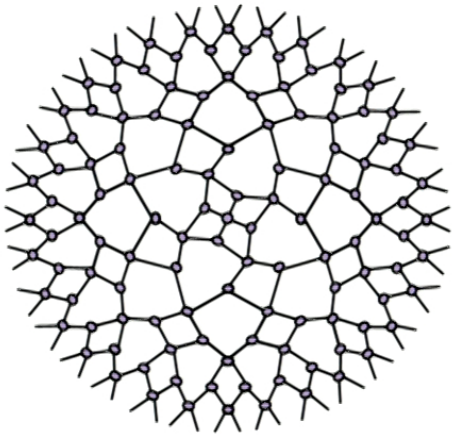


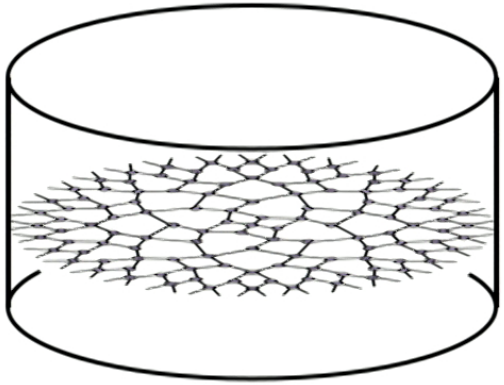


AdS_3/CFT_2

Swingle 2009, 2012

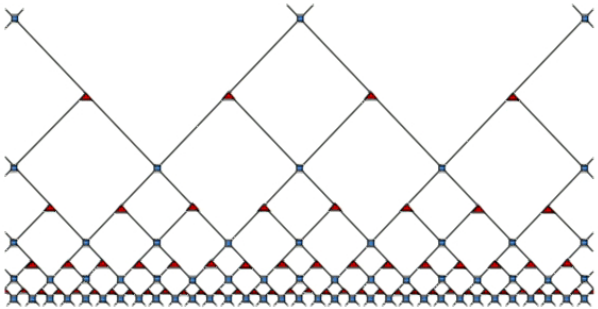
time slice of AdS_3 (hyperbolic plane H_2)

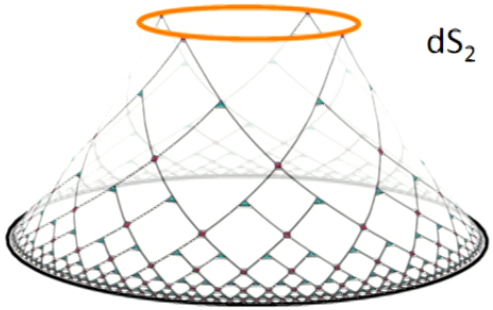




Czech, Lamprou, McCandlish, Sully, 2015-2016

kinematic space (integral transform of H_2)





Current applications

Example 4: Machine Learning

Current applications

Example 4: Machine Learning

Pattern recognition:

A neural network can be taught to e.g. recognize digits 0-9

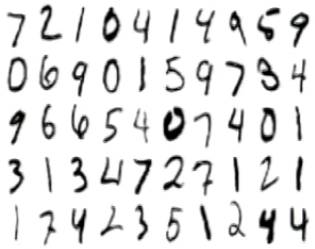
Current applications

Example 4: Machine Learning

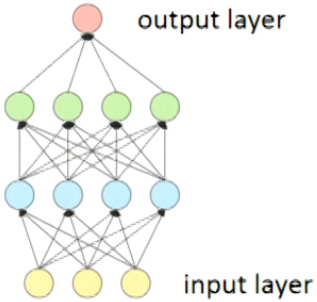
Pattern recognition:

A neural network can be taught to e.g. recognize digits 0-9

learning dataset



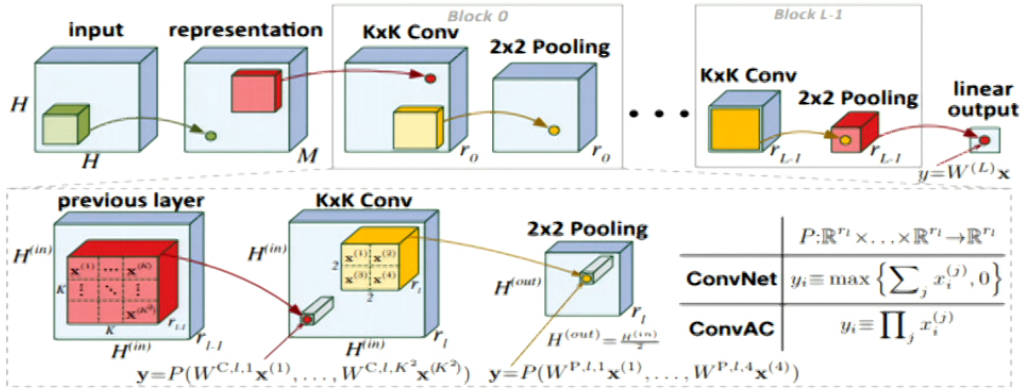
deep neural network



Current applications

Example 4: Machine Learning

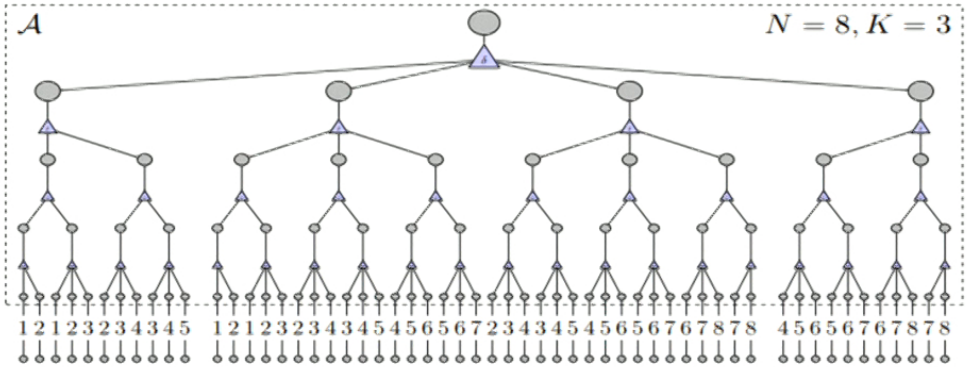
Deep Convolutional Network



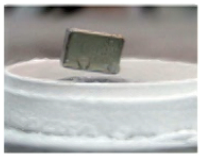
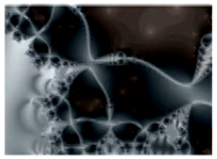
⇕

ConvAC ⇕

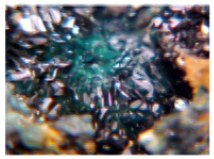
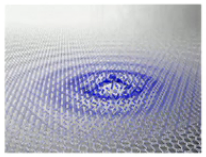
Tree Tensor Network



Y. Levine, O. Sharir, N. Cohen, A. Shashua
Bridging Many-Body Quantum Physics and Deep Learning via Tensor Networks
arXiv:1803.09780

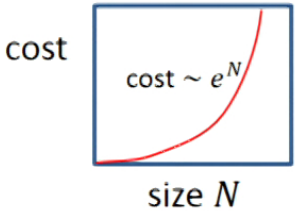



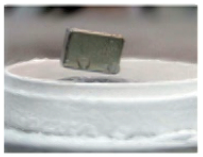
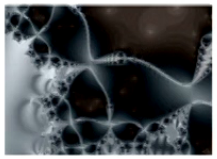
Summary

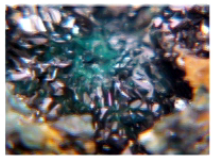
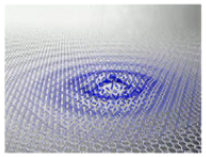
1 – Quantum mechanics and the many-body problem

$$i\hbar \frac{\partial}{\partial t}\Psi = \hat{H}\Psi$$



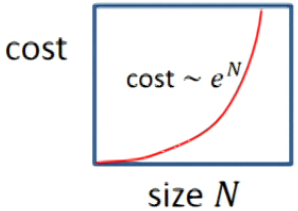



Summary

1 – Quantum mechanics and the many-body problem

$$i\hbar \frac{\partial}{\partial t}\Psi = \hat{H}\Psi$$



2 – Many-body entanglement

$$S(A) \sim |\partial A|$$

area law