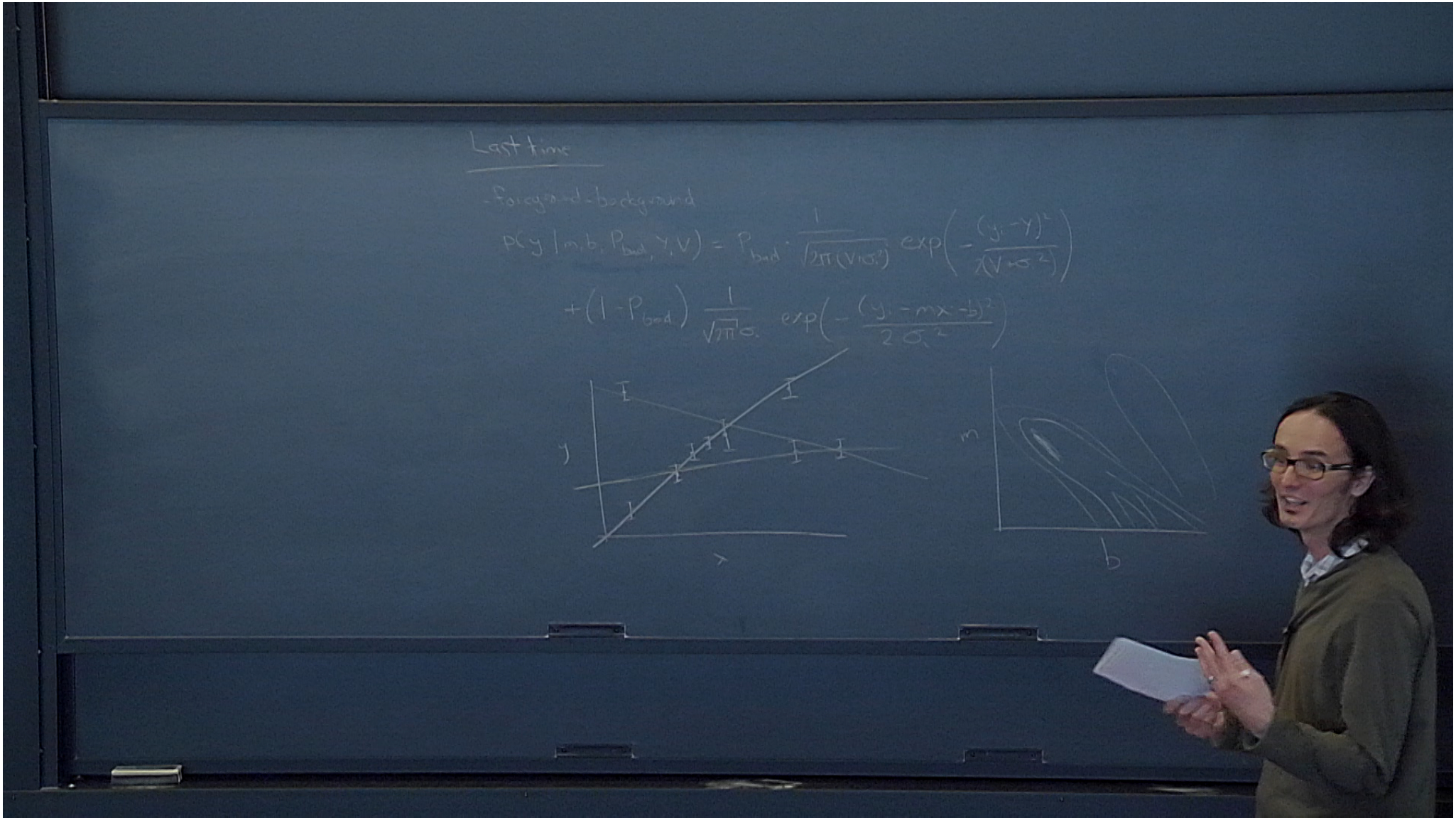
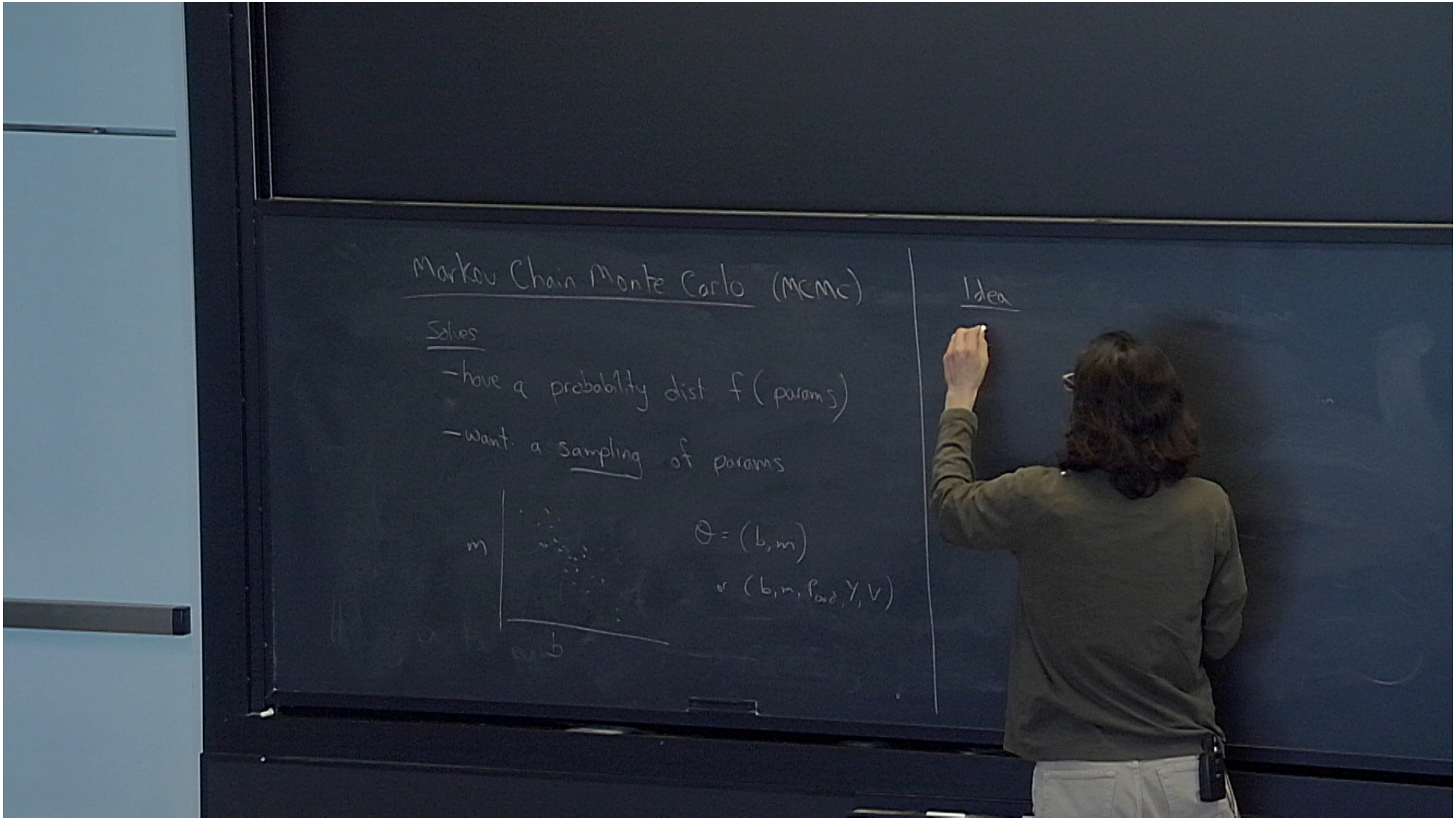
Title: Computational Physics - Lecture 11

Date: Oct 17, 2018 01:00 PM

URL: http://pirsa.org/18100052

Abstract:

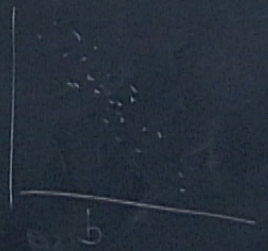

Last time
- foreground - background
$p(y_i \mid m, b, P_{bad}, Y, V) = P_{bad} \cdot \frac{1}{\sqrt{2\pi(V+\sigma_i^2)}} \exp\left(-\frac{(y_i - Y)^2}{2(V+\sigma_i^2)}\right)$
$+(1-P_{bad}) \frac{1}{\sqrt{2\pi}\sigma_i} \exp\left(-\frac{(y_i - mx_i - b)^2}{2\sigma_i^2}\right)$
y
x
m
b


Markov Chain Monte Carlo (MCMC)
Idea
Solves
-have a probability dist f(params)
-want a sampling of params
m
b
θ = (b, m)
or (b, m, ρ_bd, Y, V)

Markov Chain Monte Carlo (MCMC)

Solves

- have a probability dist $f(\text{params})$
- want a sampling of params



$\theta = (b, m)$

or (b, m, P_{bad}, Y, V)

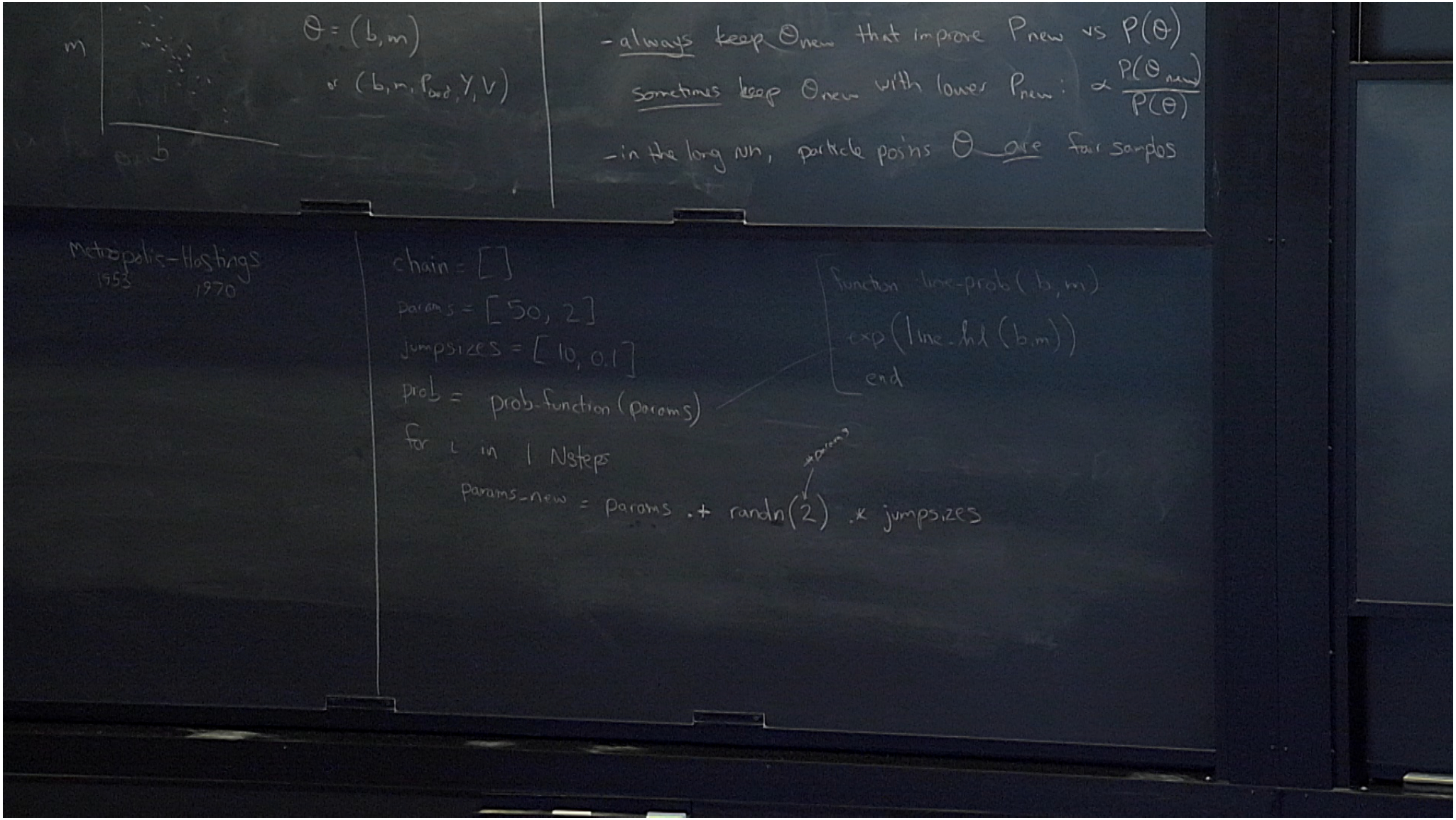
Idea

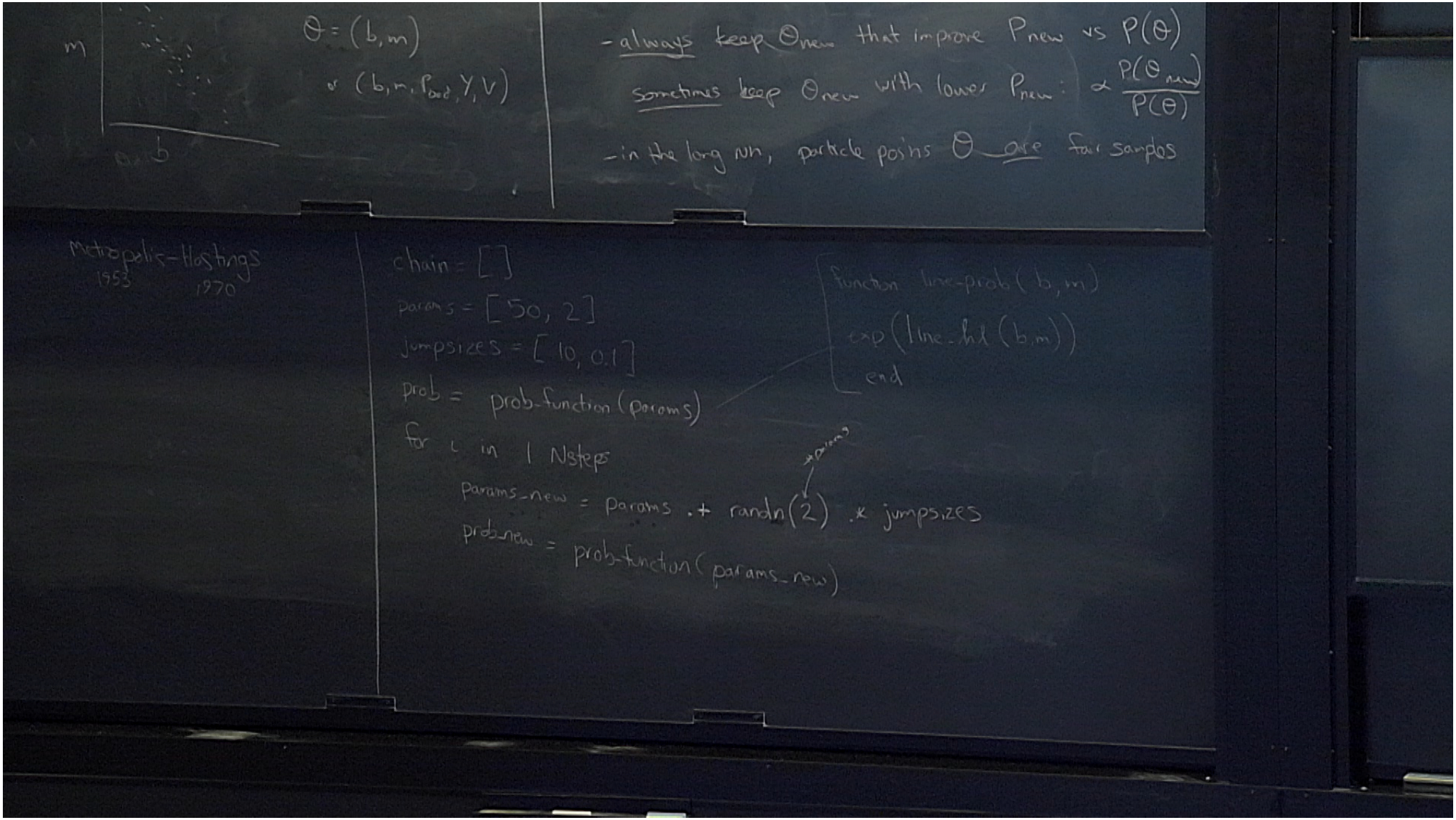
- more a "particle"

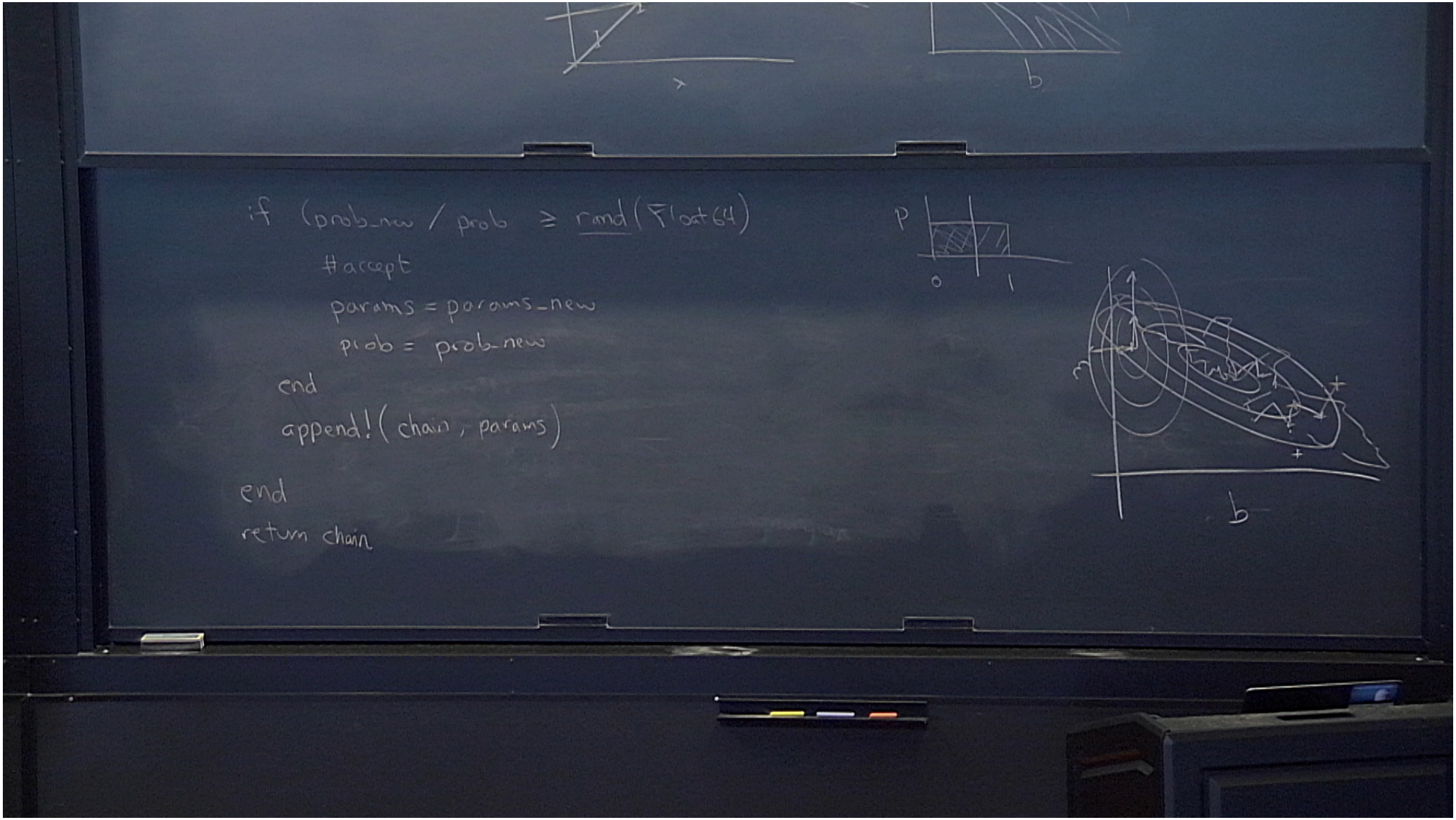
$$\theta_{new} = \underbrace{\theta + N(0, \sigma_\theta^2)}_{\text{proposal distrib}}$$

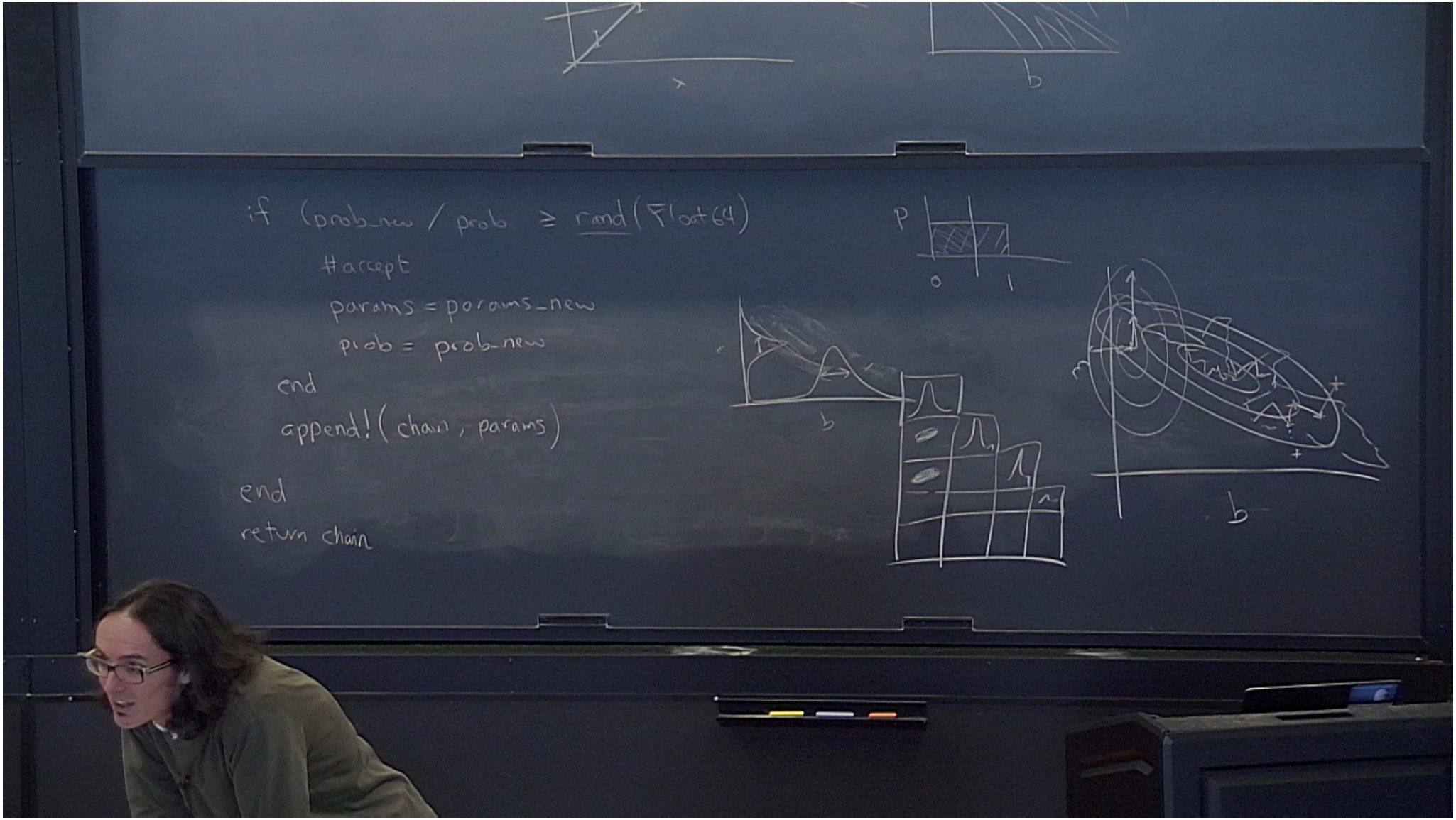
- compute prob at θ_{new} $\quad P_{new} = P(\theta_{new})$
- always keep θ_{new} that improve P_{new} vs $P(\theta)$

  sometimes keep θ_{new} with lower P_{new}: $\propto \frac{P(\theta_{new})}{P(\theta)}$
- in the long run, particle posns θ are fair samples


m
b
θ = (b, m)
or (b, m, P_bad, Y, V)
- always keep θ_new that improve P_new vs P(θ)
sometimes keep θ_new with lower P_new: ∝ P(θ_new)/P(θ)
- in the long run, particle posns θ are fair samples
Metropolis-Hastings
1953
1970
chain = []
params = [50, 2]
jumpsizes = [10, 0.1]
prob = prob-function(params)
for i in 1 Nsteps
params_new = params .+ randn(2) .* jumpsizes
function line-prob(b, m)
exp(line_hl(b,m))
end


m
b
θ = (b, m)
or (b, m, P_bad, Y, V)
- always keep θ_new that improve P_new vs P(θ)
sometimes keep θ_new with lower P_new: ∝ P(θ_new)/P(θ)
- in the long run, particle points θ are fair samples
Metropolis-Hastings
1953 1970
chain = []
params = [50, 2]
jumpsizes = [10, 0.1]
prob = prob_function(params)
for i in 1 Nsteps
params_new = params .+ randn(2) .* jumpsizes
prob_new = prob_function(params_new)
function line_prob(b, m)
exp(line_hl(b, m))
end


if (prob_new / prob ≥ rand(Float64)
#accept
params = params_new
prob = prob_new
end
append!(chain, params)
end
return chain
p
0
1
b
b


if (prob_new / prob ≥ rand(Float64)
#accept
params = porams_new
prob = prob_new
end
append!(chain, params)
end
return chain
p
0
1
b
b