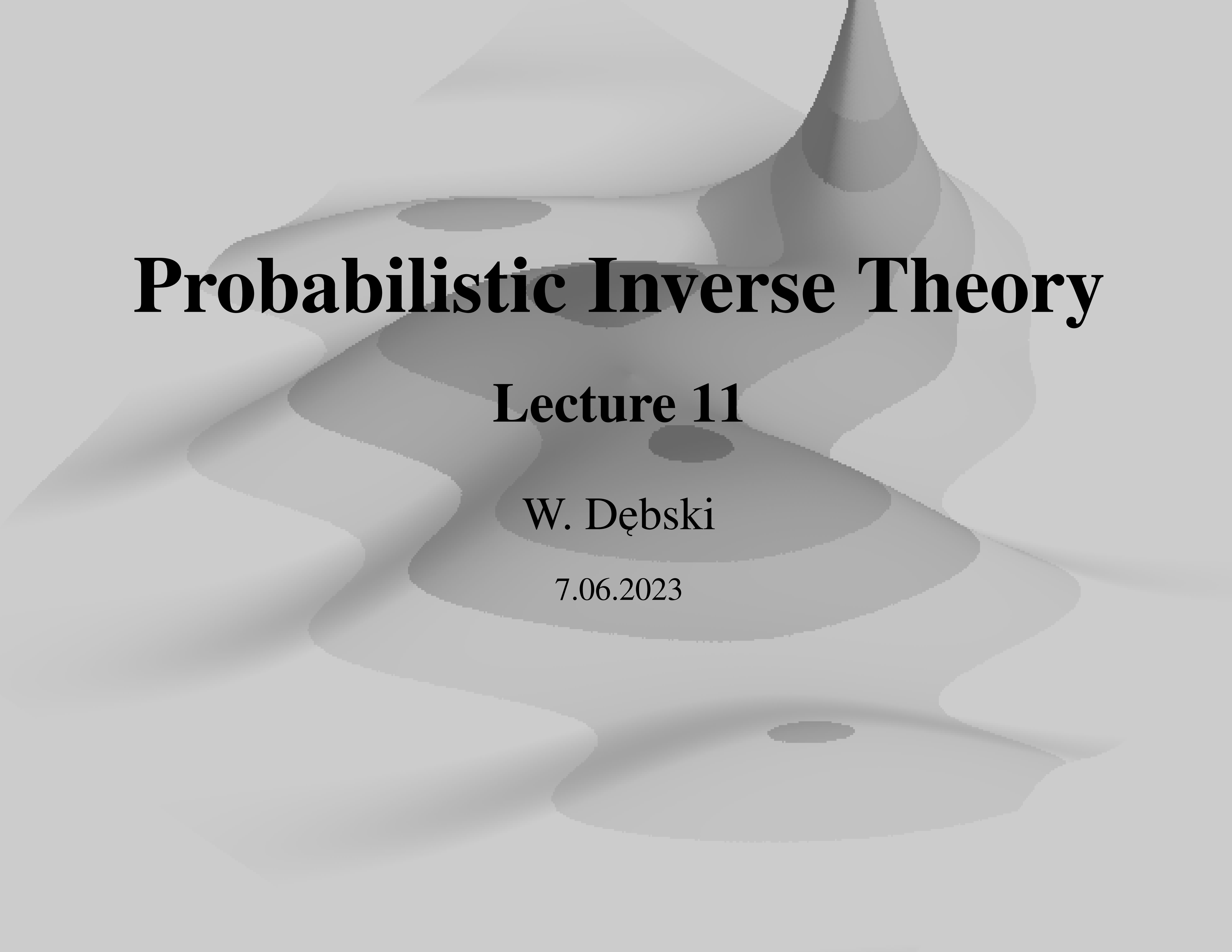


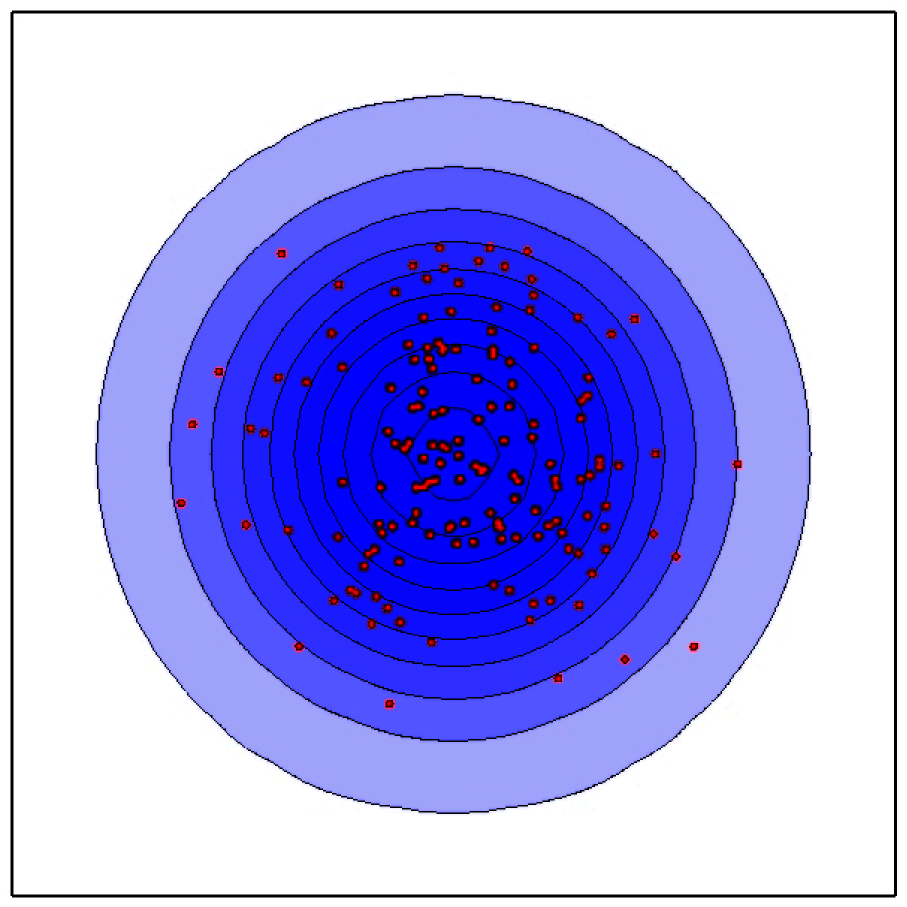
Probabilistic Inverse Theory

Lecture 11

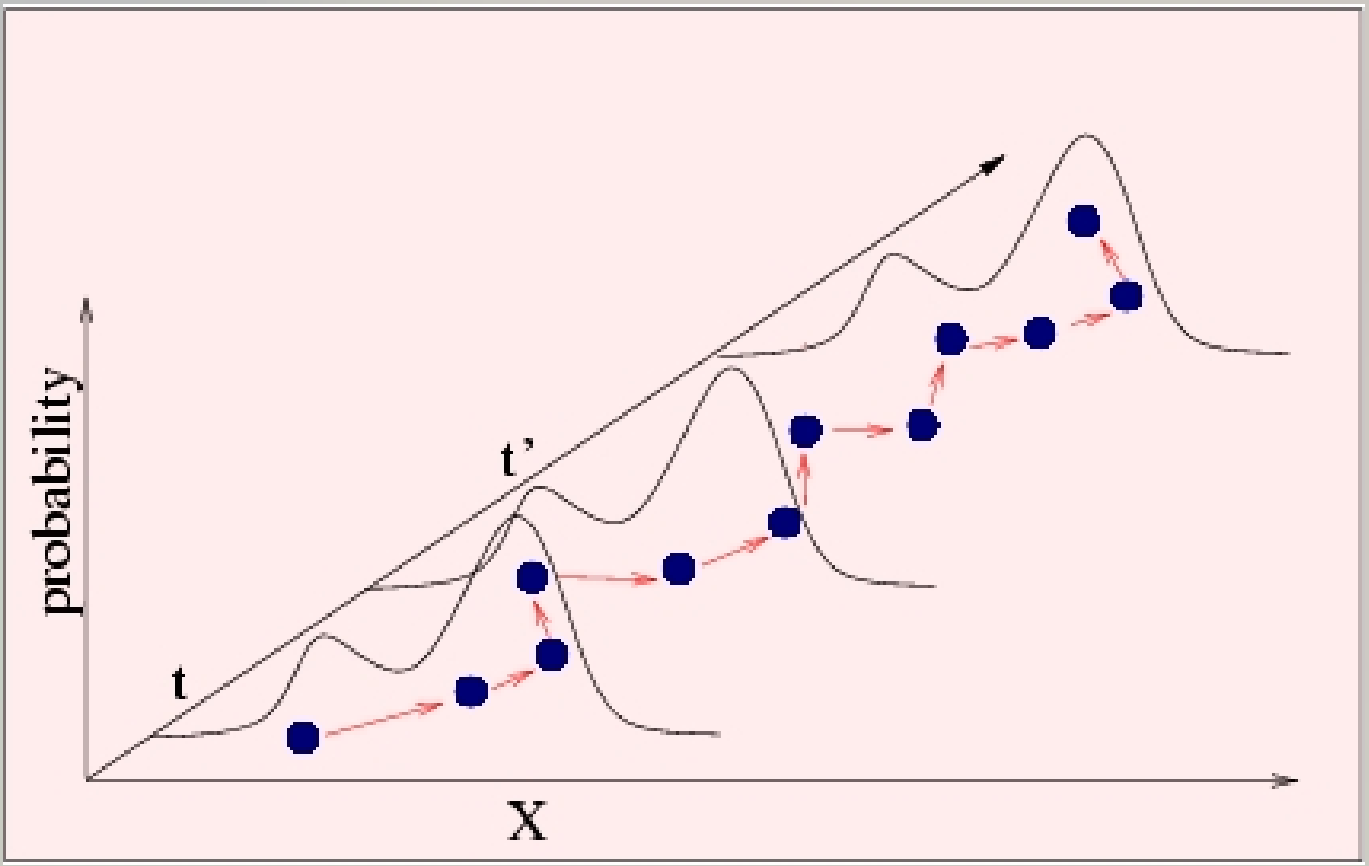
W. Dębski

7.06.2023

Direct sampling $sigma(\mathbf{m})$



Stationary Markov Chain



Direct vs. MCMC sampling

✦ Direct sampling:

generation of samples $\mathbf{m}^\alpha$ in such a way that all lies in a region where $\sigma(\mathbf{m})$ is large so they can be used for accurate estimation of integrals as follow

$$I = \int_{\mathcal{M}} h(\mathbf{m})\,\sigma(\mathbf{m}) d\mathbf{m} \approx \sum_\alpha h(\mathbf{m}^\alpha)\,\sigma(\mathbf{m}^\alpha)$$

Direct vs. MCMC sampling

✦ MCMC sampling generation of samples $\mathbf{m}^\alpha$ in such a way that

$$N(\mathbf{m}^\alpha \in [\mathbf{m} - h, \mathbf{m} + h]) \;\sim\; N^{tot}\sigma(\mathbf{m})$$

then

$$I \int_{\mathcal{M}} h(\mathbf{m})\sigma(\mathbf{m})d\mathbf{m} \approx \sum_\alpha h(\mathbf{m}^\alpha)$$

Direct vs. MCMC sampling - summarising

- ✦ Direct sampling:
  important are **values** $\sigma(\mathbf{m}^\alpha)$ not distribution of $\mathbf{m} \in \mathcal{M}$

- ✦ MCMC sampling:
  important is **distribution** of $\mathbf{m} \in \mathcal{M}$, not values $\sigma(\mathbf{m}^\alpha)$

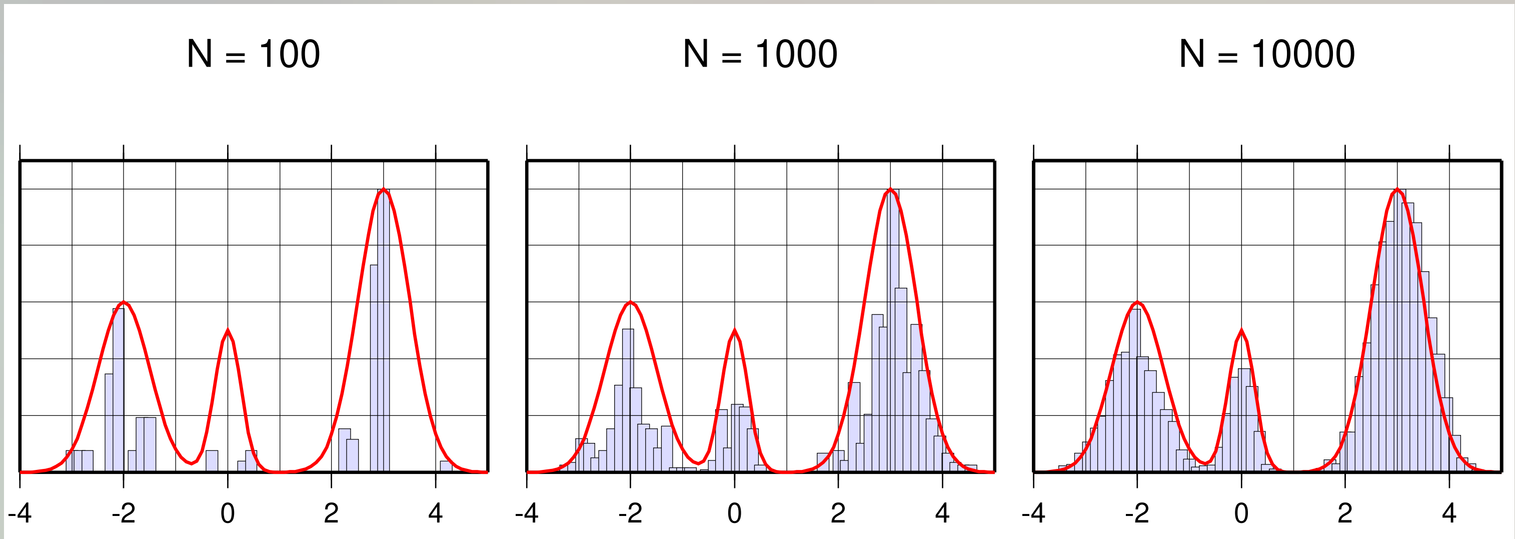
Metropolis pseudo-code for sampling from $p(\mathbf{m})$

```
✦ Initialize sampling $\mathbf{m}^\alpha = \mathbf{m}^0$

    ★ generate test sample $\mathbf{m}^\beta$:    $\mathbf{m}^\beta = \mathbf{m}^\alpha + \delta\mathbf{m}$
    ★ evaluate $\mathbf{m}^\beta$:    $p_\beta = p(\mathbf{m}^\beta, T_k)$
    ★ select a new chain state $\mathbf{m}^{\alpha+1}$
        ➡ accept $\mathbf{m}^\beta$ with probability $p = min\,(1, p_\beta/p_\alpha)$

                $\mathbf{m}^{\alpha+1} = \mathbf{m}^\beta$

        ➡ if $\mathbf{m}^\beta$ rejected duplicate

                $\mathbf{m}^{\alpha+1} = \mathbf{m}^\alpha$

✦ Repeat untill sufficient number of $\{\mathbf{m}^i\}$ is obtained
```

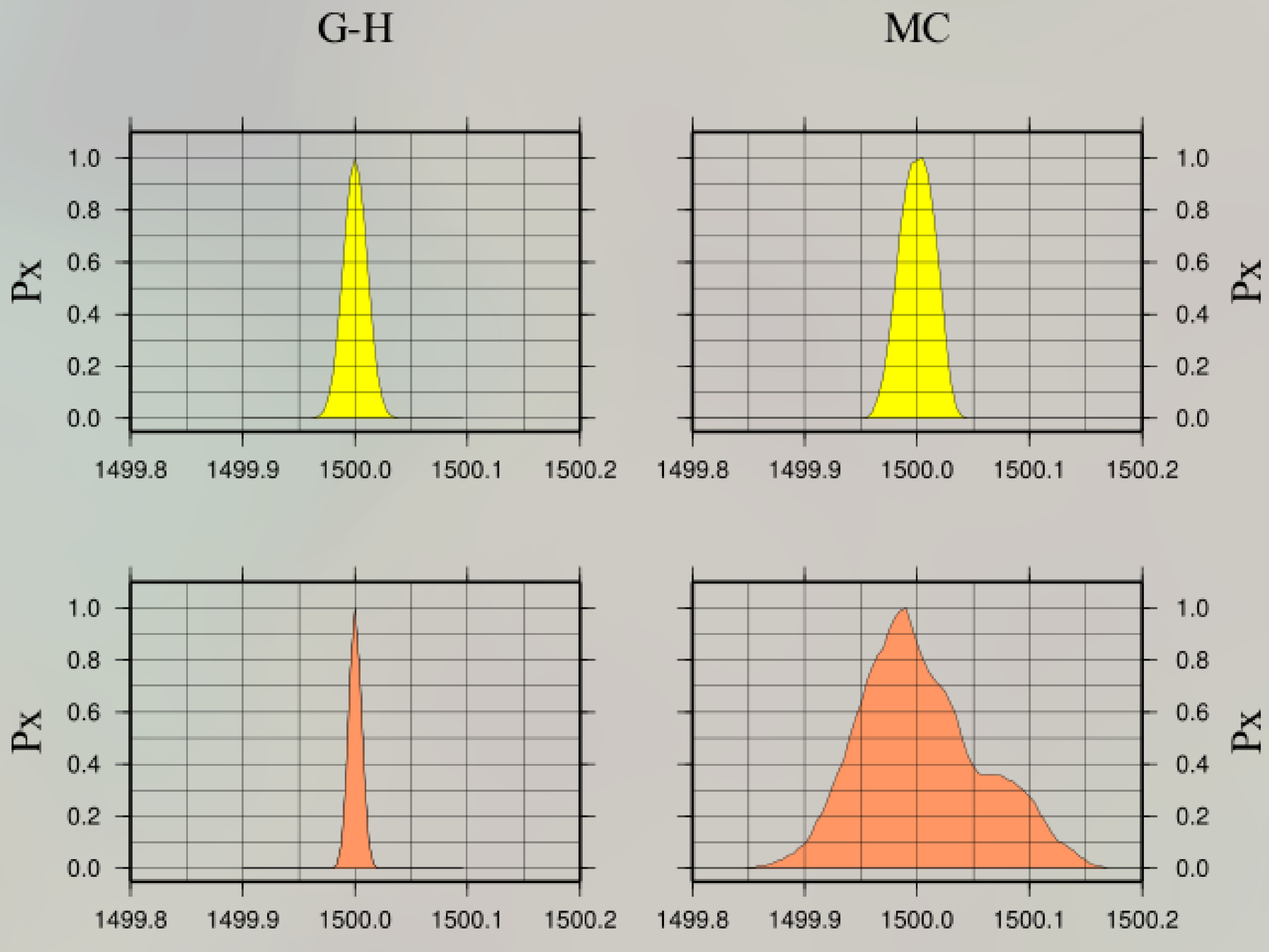
Metropolis algorithm -features

- after so called burn-in initial time generated samples follow $\sigma(\mathbf{m})$ probability distribution
- subsequent samples are strongly correlated - the chain must be run for a long time
- MH is optimum sampling algorithm if only $\sigma(\mathbf{m})$ is available.
- How many samples should be generated ?
- problem with generating “proper” test samples ($\mathbf{m}^{\beta}$)

MH algorithm - how many samples



MH algorithm - generating proposal samples

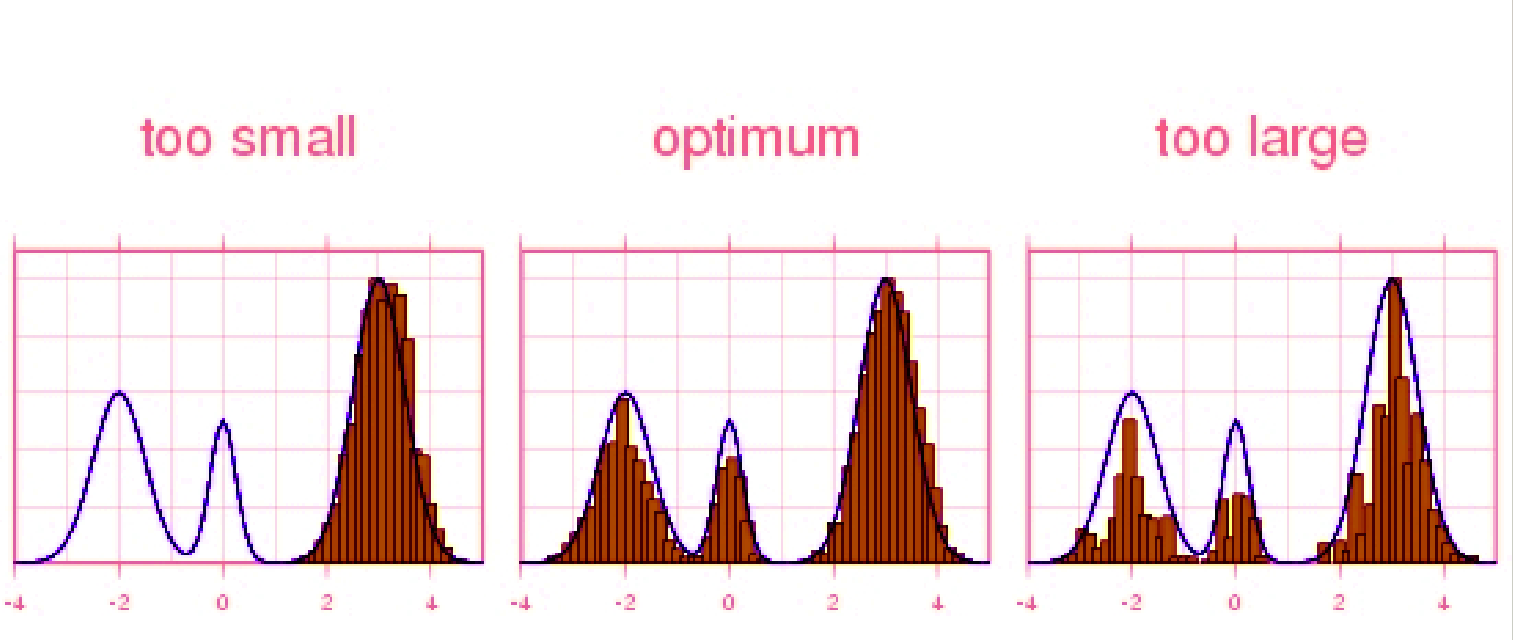


MH algorithm - generating proposal samples

“Narrow” $\sigma(\mathbf{m})$ requires small update of $\mathbf{m}^\alpha$

$$\mathbf{m}^\beta = \mathbf{m}^\alpha + \delta\mathbf{m}$$

but then we diminishing mixing property of the chain



Metropolis Hasting algorithm

$$\sigma(\mathbf{m}) \;=\; f(\mathbf{m})L(\mathbf{m})$$

```
✦ Initialize $\mathbf{m}^0$

✦ Repeat

    ★ generate test sample $\mathbf{m}^\beta$ from $f(\mathbf{m})$
    ★ generate uniform random number $u \sim U(0,1)$
    ★ if $u < P(\mathbf{m}^\beta, \mathbf{m}^\alpha) \;=\; \min\left[1, \frac{L(\mathbf{m}^\beta)}{L(\mathbf{m}^\alpha)}\right]$
          $\mathbf{m}^{\alpha+1} = \mathbf{m}^\beta$
      otherwise
          $\mathbf{m}^{\alpha+1} = \mathbf{m}^\alpha$

✦ Continue until sufficient number of samples $\{\mathbf{m}^\alpha\}$ is generated
```

Mixture of two MH chains

$$\mathbf{K}_m = \nu \mathbf{K}_1 + (1-\nu)\mathbf{K}_2$$

- Generate initial state $\mathbf{m}^0$
- Repeat
  - get random number $u \sim U(0,1)$
  - if $u < \nu$

        run the chain with kernel $\mathbf{K}_1$

    otherwise

        run the chain with kernel $\mathbf{K}_2$
  - reselect chain $\mathbf{K}_1, \mathbf{K}_2$

    and restart from $\mathbf{m}^\alpha = \mathbf{m}^{\mathbf{K}_i^{final}}$

The Markov chain mixture algorithm is built from two chains with the $\mathbf{K}_1$ and $\mathbf{K}_2$ kernels having the same invariant distribution.

Non stationary MCMC

✦ discrete time: $t_1, t_2, \cdots$

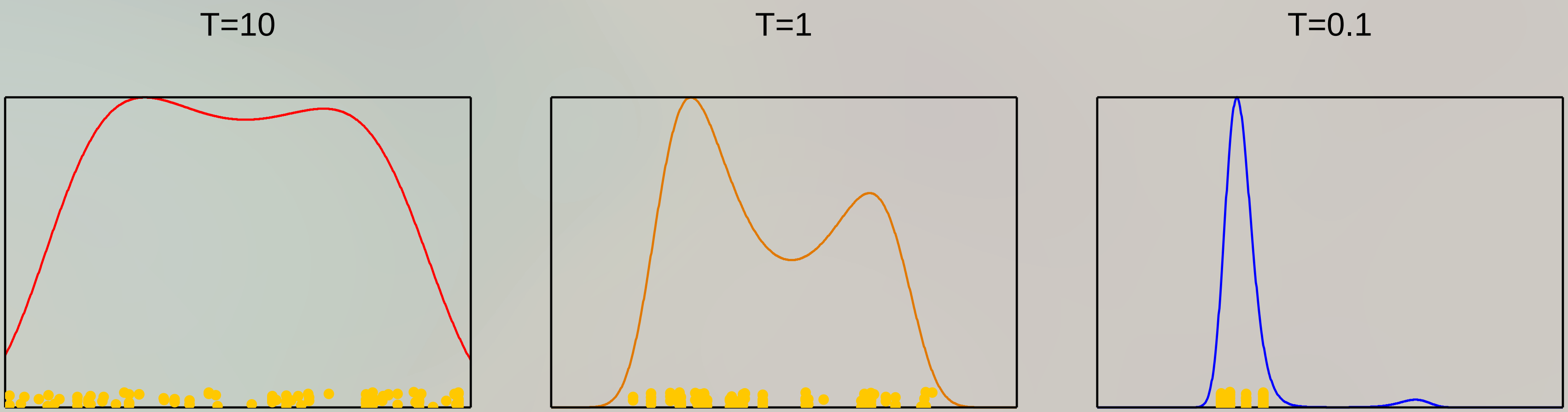
✦ short memory $K(t_i, x; t_{i-1}, x_{i-1})$

✦ ergodic process:

- ★ stationary $K(t, x; t', x') = K(x, x')$
- ★ irreducible
- ★ aperiodic

*** NO Stationary probability distribution:***

Non stationary chain evolution



Example of (slow) MC chain “stationary” distribution evolution in time due to a particular time dependences of $K(\cdot,\cdot)$ kernel.

Non stationary MCMC

$$p(\mathbf{m}) \;=\; \exp\left(-\frac{S(\mathbf{m})}{kT}\right)$$

$$K(\mathbf{m}^\beta,\mathbf{m}^\alpha): \quad P(\mathbf{m}^\beta,\mathbf{m}^\alpha) \;=\; \min\left[1, exp\left(-\frac{S(\mathbf{m}^\beta)-S(\mathbf{m}^\alpha)}{kT}\right)\right]$$

where, e.g.

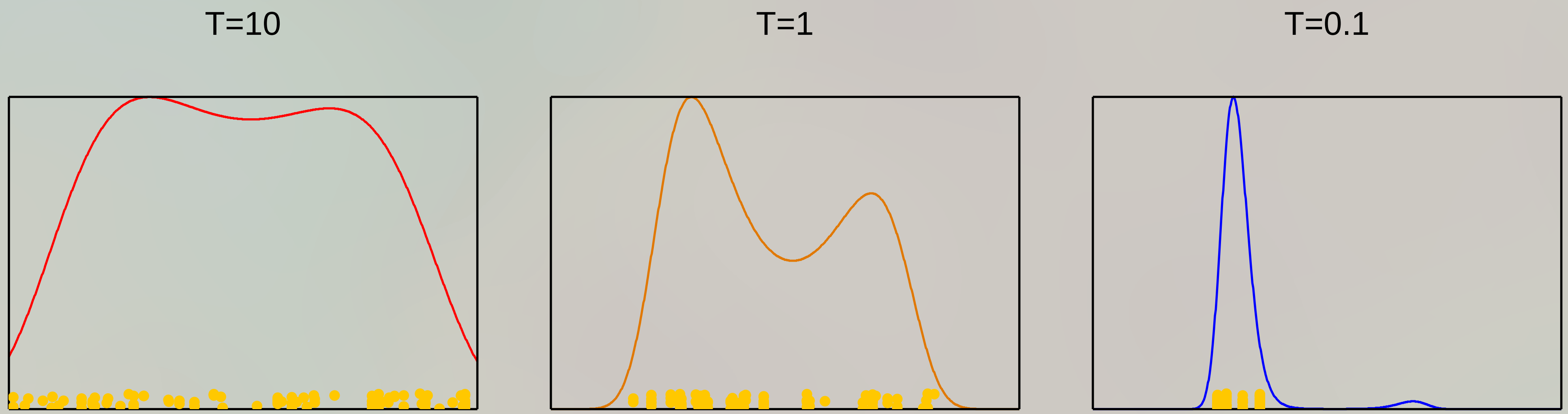
$$S(\mathbf{m}) \;=\; ||\mathbf{d}^o \;-\; \mathbf{d}^{th}(\mathbf{m})||$$

and allow decreasing of T:

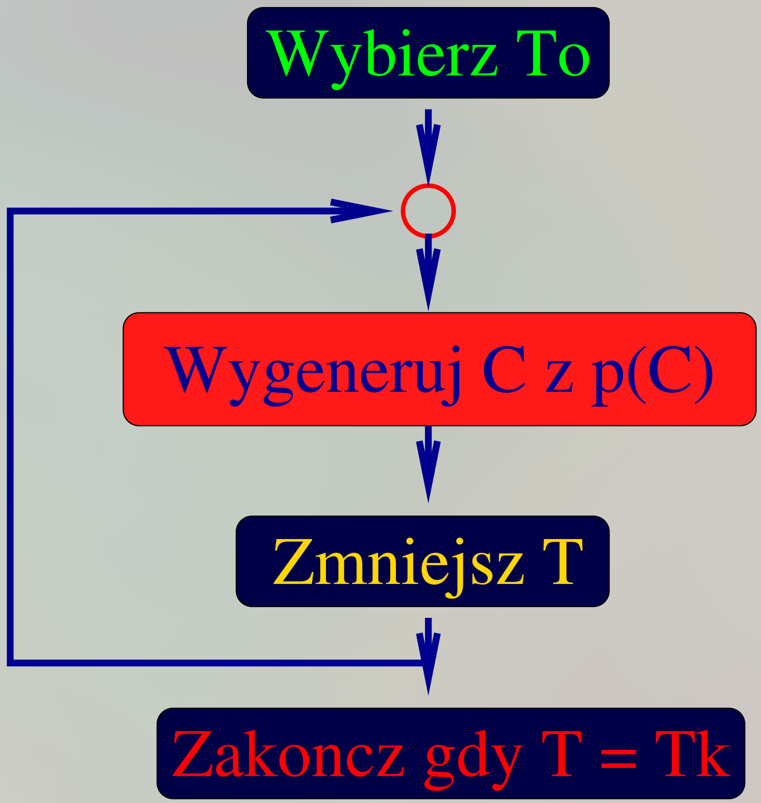
$$T(t_i) = \frac{T_o}{\ln(t_i)}$$

Non stationary MCMC

$$p(\mathbf{m}) \;=\; \exp\left(-\frac{S(\mathbf{m})}{kT}\right)$$



Simulated annealing - an idea

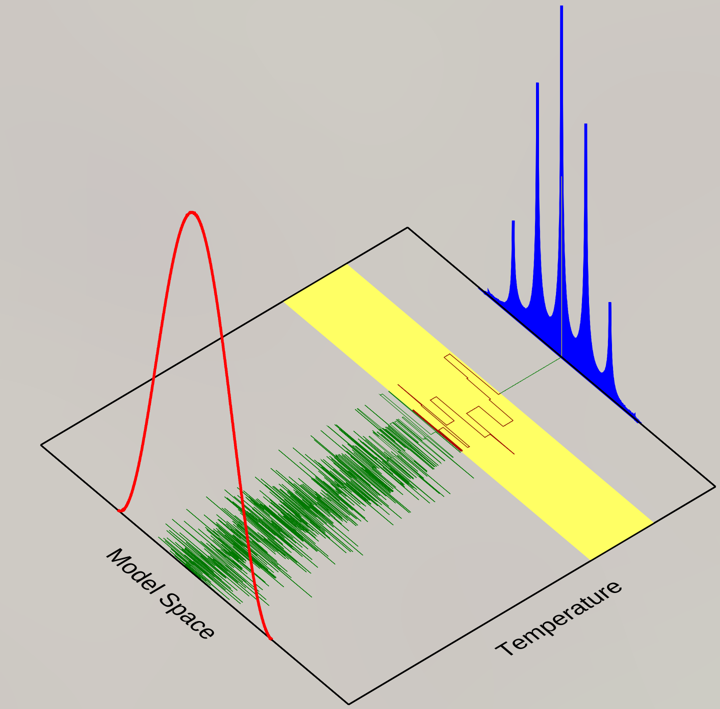


Wybierz To

Wygeneruj C z p(C)

Zmniejsz T

Zakoncz gdy T = Tk



SA - pseudo-code

```
✦ Create Boltzman-like distribution $p(\mathbf{m}, T) = \exp(-S(\mathbf{m})/T)$
✦ Set $T = T_o$
    ★ generate test sample $\mathbf{m}^\beta$:    $\mathbf{m}^\beta = \mathbf{m}^\alpha + \delta\mathbf{m}$
    ★ evaluate it $\mathbf{m}^\beta$:    $p_\beta = p(\mathbf{m}^\beta, T_k)$
    ★ create a new one $\mathbf{m}^{\alpha+1}$
        ➡ accept $\mathbf{m}^\beta$ with probability $p = min\,(1, p_\beta/p_\alpha)$
              $\mathbf{m}^{\alpha+1} = \mathbf{m}^\beta$
        ➡ if $\mathbf{m}^\beta$ rejected duplicate
              $\mathbf{m}^{\alpha+1} = \mathbf{m}^\alpha$
✦ Keep the “best” $\mathbf{m}^\alpha$
✦ Decrease $T_k = \frac{T_0}{\ln(k)}$
✦ continue sampling untill $T_k > T_{final}$
```

Simulated annealing - elements

Two most important elements

1. generating sample from $p(\mathbf{m})$
2. schedule of T change

Efficient sampling - the Metropolis algorithm (generalized by Hastings)
Optimum cooling schedule (Kirkpatrick)

$$T_k = \frac{T_o}{\ln(k)}$$

Gobal Optimization Algorithm

Hamiltonian MCMC

Original Metropolis algorithm is actually performing purely random walk over $\mathcal{M}$

$$\mathbf{m}^{\beta} = \mathbf{m}^{\alpha} + \delta\mathbf{m} \qquad \delta\mathbf{m} = \mathbf{m}_o \, U_{rand}(-1, 1)$$

(filtered next by Metropolis selection KerneL)

For this reason the algorithm requires an extremely long runs and subsequent samples are strongly correlated

Hamiltonian MCMC

Hastings has extended the Metropolis algorithm by allowing generating test samples from auxiliary (*a priori*) distribution what has improved (sometimes not) convergence of the original algorithm.

Many other attemps have also been undertaken to improve the situation (convergence of the chain) with varying results. Usually proposed extensions work better for some classes of problems but fails in others

NO FREE LUNCH THEOREM

Hamiltonian MCMC

Significant improvements of this situation requires additional information on sampled $\sigma(\mathbf{m})$. Particular (often met) situation occurs when

$$\sigma(\mathbf{m}) \;=\; k\, f(\mathbf{m})\, L(\mathbf{m}, \mathbf{d}^o)$$

and one can calculate gradient of $L(\mathbf{m}, \mathbf{d}^o)$ with respect to $\mathbf{m}$

$$\nabla_i L \;=\; \frac{\partial L(\mathbf{m}, \mathbf{d}^o)}{\partial m_i}$$

One can use this information for a more efficient generation of test samples

$$\mathbf{m}^\beta = \mathbf{m}^\alpha + \delta\mathbf{m}$$

Hamiltonian MCMC - idea

The most popular method of including $\nabla_i L$ information into sampling algorithm is so called

Hamiltonian Monte Carlo

It's basic idea is avoiding generation of test sample $\mathbf{m}^\beta = \mathbf{m}^\alpha + \delta\mathbf{m}$ more or less randomly but rather to take it as a result of a deterministic evolution from a given chain state $\mathbf{m}^\alpha$.

This evolution is based on the Hamiltonian dynamics in an extended phase space $(\mathcal{M}, \mathcal{V})$

Hamiltonian dynamics

Let **P** be a particle of a unit mass moving in R^d space due to a potential energy E_p. Let

- ✦ x - position of particle
- ✦ v - its velocity

One can define the Hamiltonian (energy) functional

$$H(x, v) \;=\; E_p \;+\; \frac{1}{2}||v||^2$$

over the phase space $\mathcal{P} = \{(x, v)\}$

Hamiltonian dynamics

Evolution equations:

$$\frac{dx}{dt} = \frac{\partial H}{\partial v}, \qquad \frac{dv}{dt} = -\frac{\partial H}{\partial x}$$

- Evolution conserves “energy”:

$$\frac{dH}{dt} = 0, \qquad H(x(t), v(t)) \; = \; H(x(0), v(0))$$

- Volume in phase space is conserved:

$$\vec{F} = (\frac{dx}{dt}, \frac{dv}{dt}); \quad \nabla \cdot \vec{F} = 0$$

Hamiltonian Monte Carlo

✦ Extend the model parameter space: $\mathcal{M} \rightarrow (\mathcal{M}, \mathcal{P})$

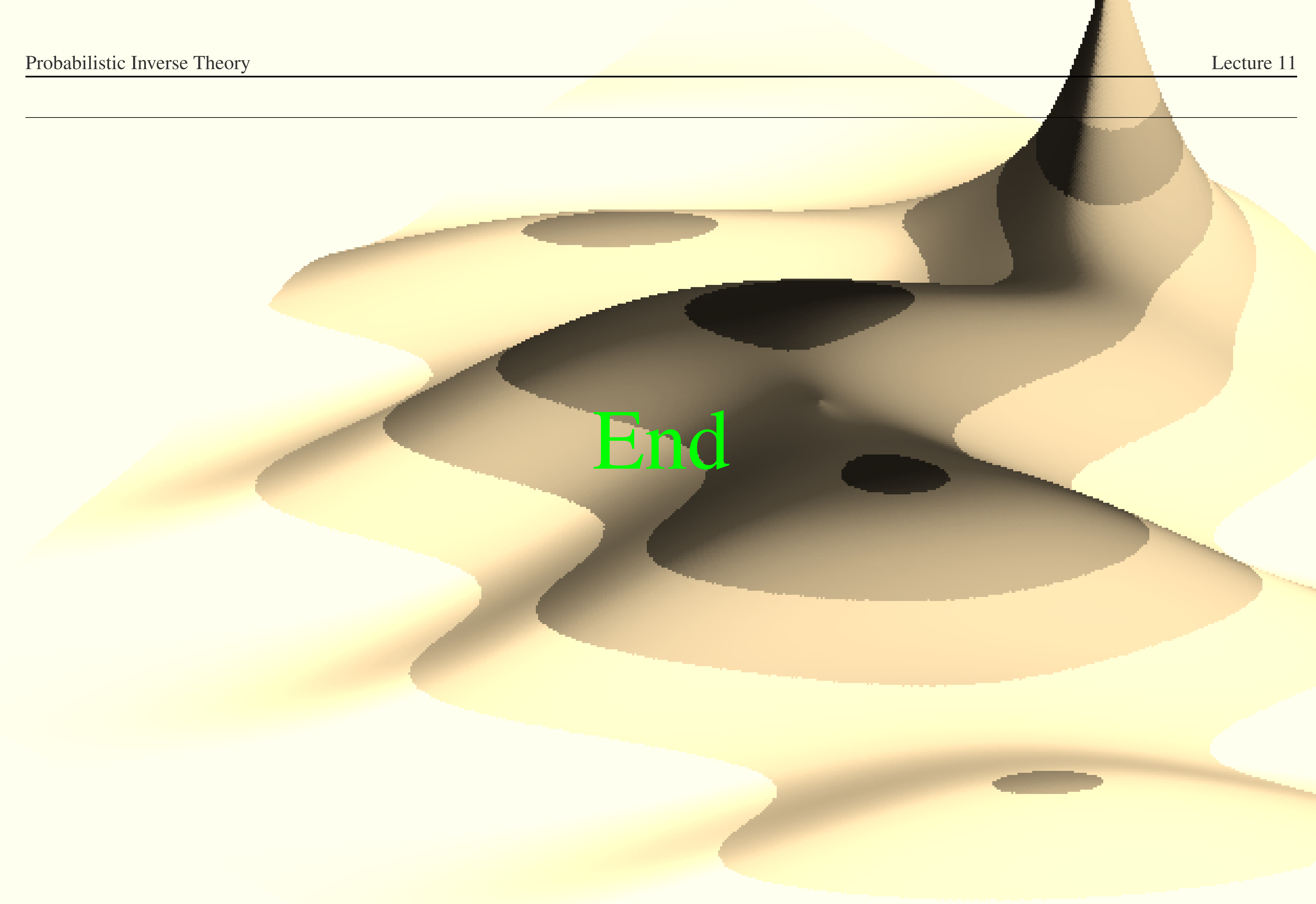
✦ Build the Hamiltonian functional

$$H(\mathbf{m}, \mathbf{p}) \;=\; -\ln(\sigma(\mathbf{m})) + \frac{1}{2}\mathbf{p}^T M^{-1}\mathbf{p}$$

✦ Evolve current state $(\mathbf{m}^\alpha, \mathbf{p}^\alpha)$ over time τ according to Hamiltonian equations to get a proposal MCMC state $\mathbf{m}^\beta = \mathbf{m}^\alpha(\tau)$

✦ accept (reject) with Metropolis acceptance kernel (replacing $S(\mathbf{m})$ by $H(\mathbf{m}, \mathbf{p})$

✦ resample $\mathbf{p}$ from $\mathcal{N}(0, I_M)$



End