

Exploring Varieties

A Practical Guide

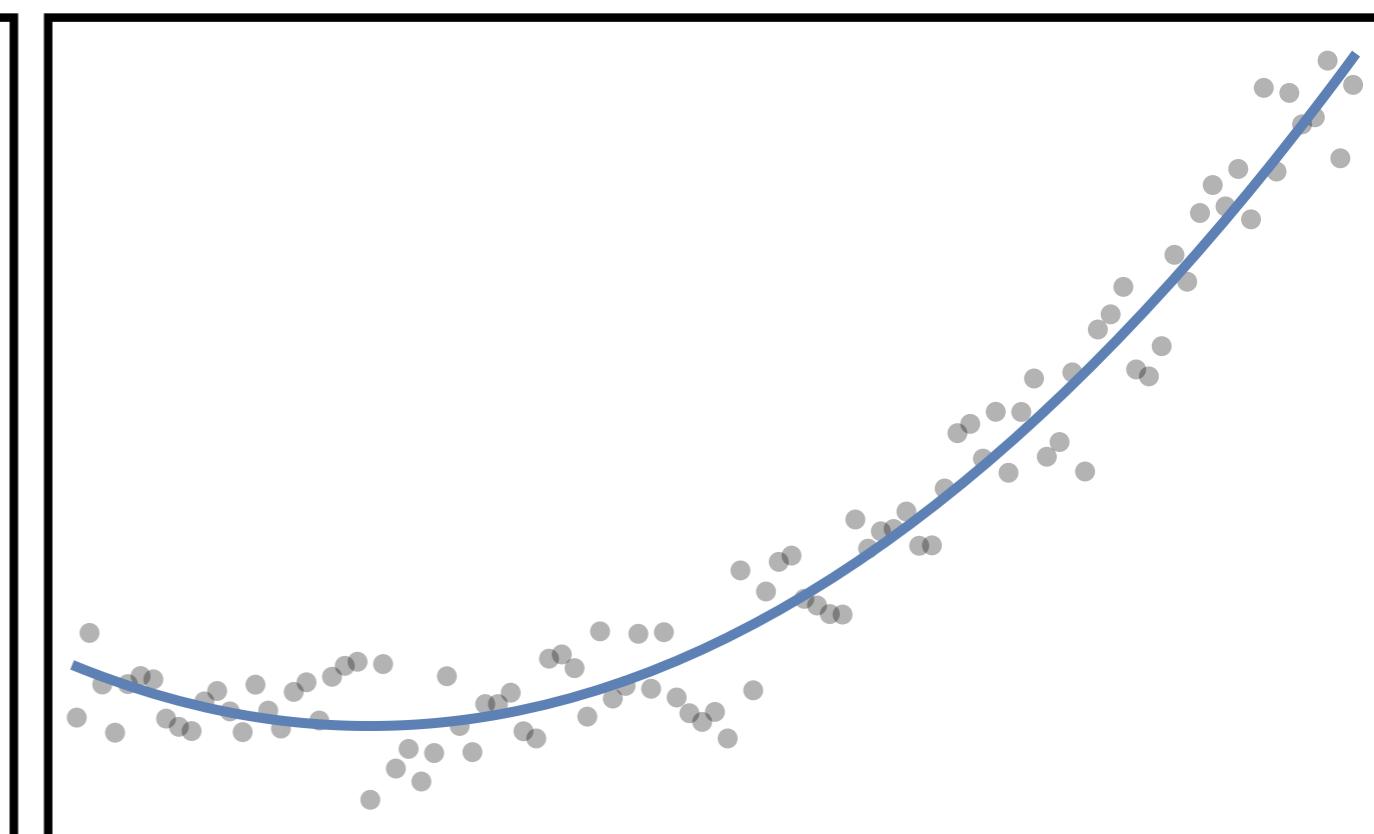
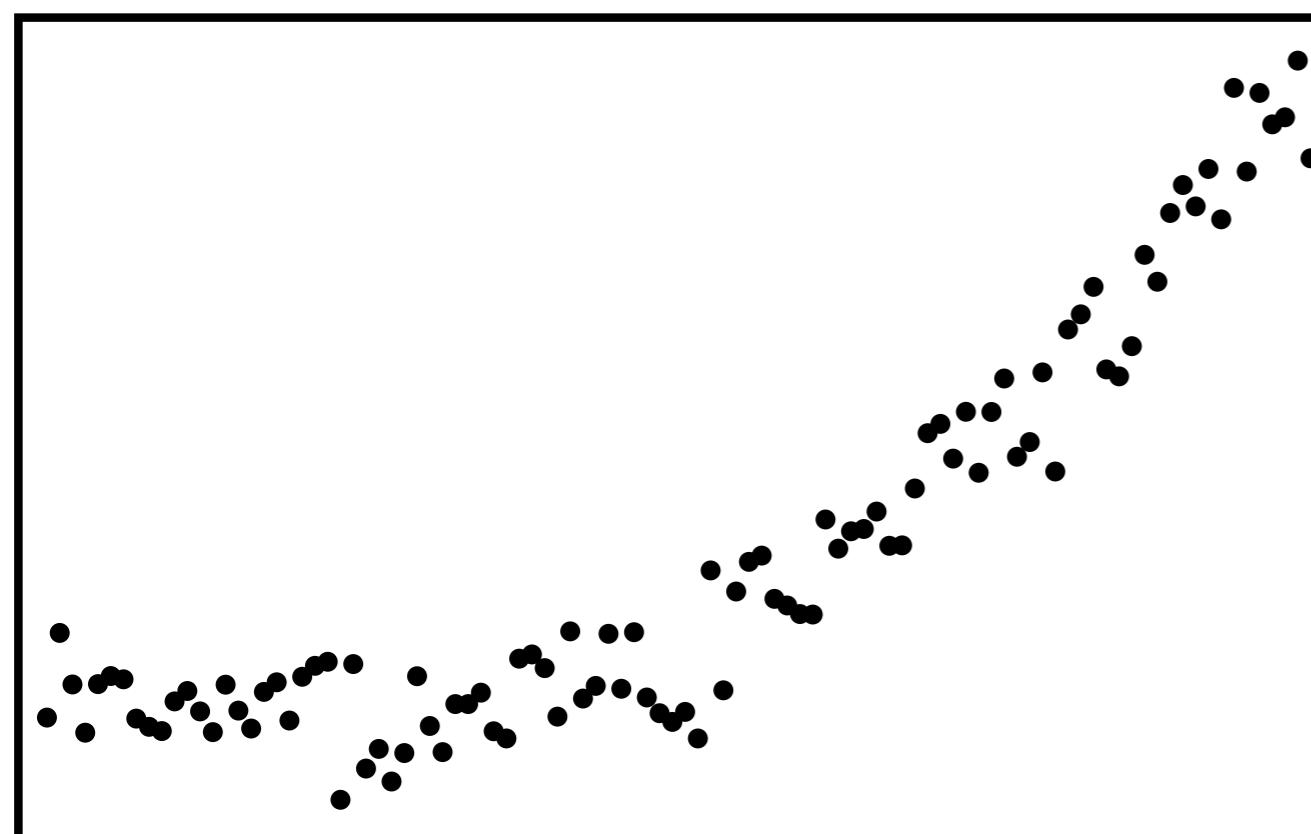
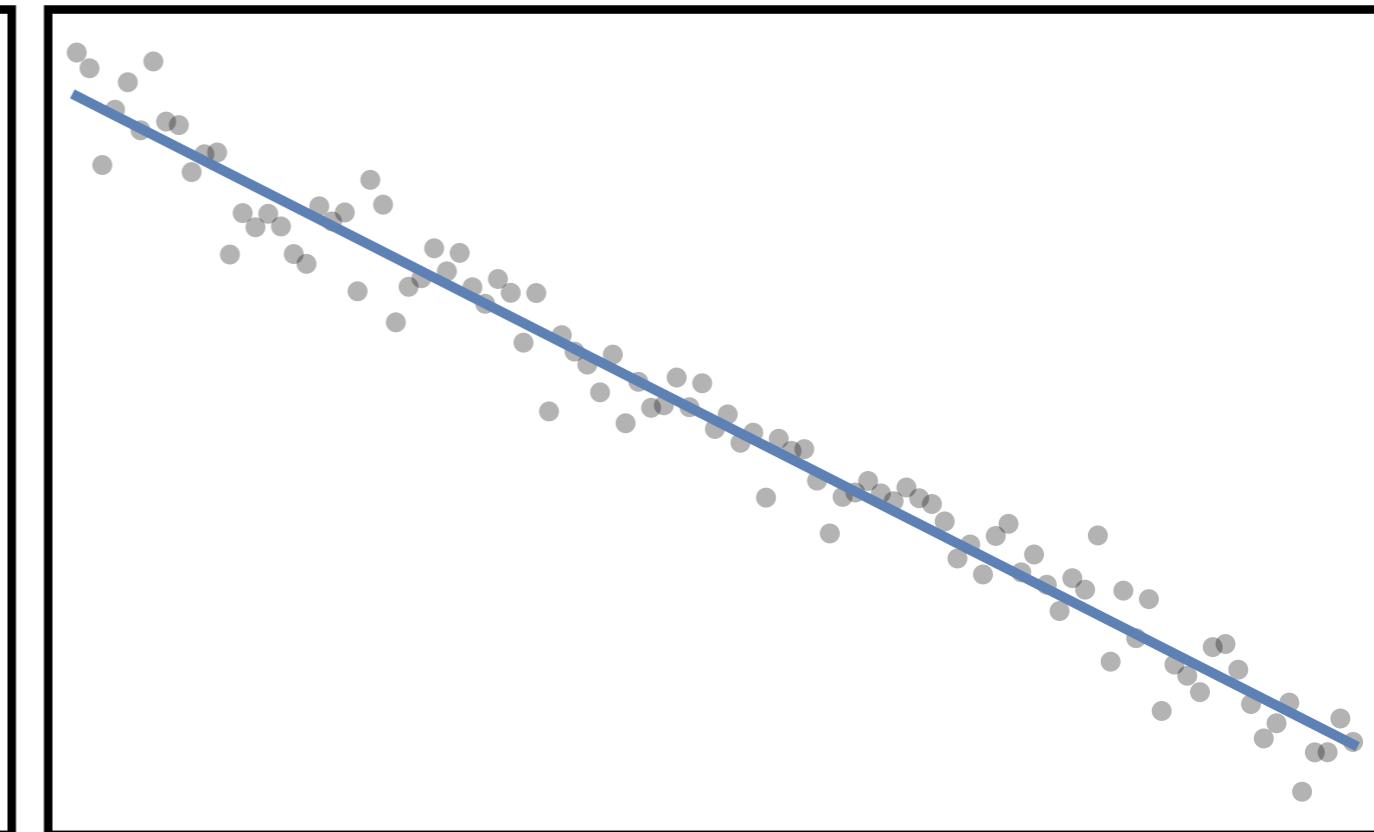
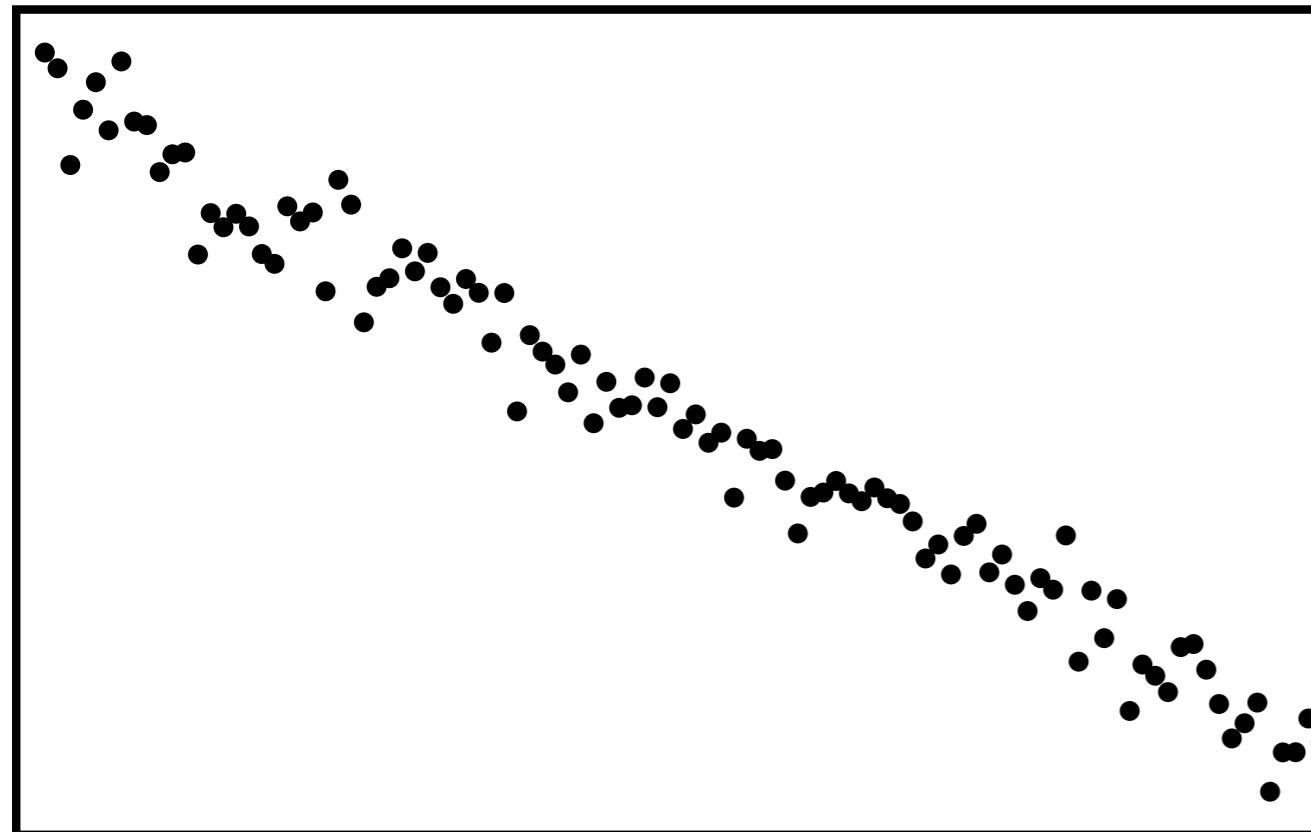
David Kahle

Joint work with Jon Hauenstein and Jerry Ma

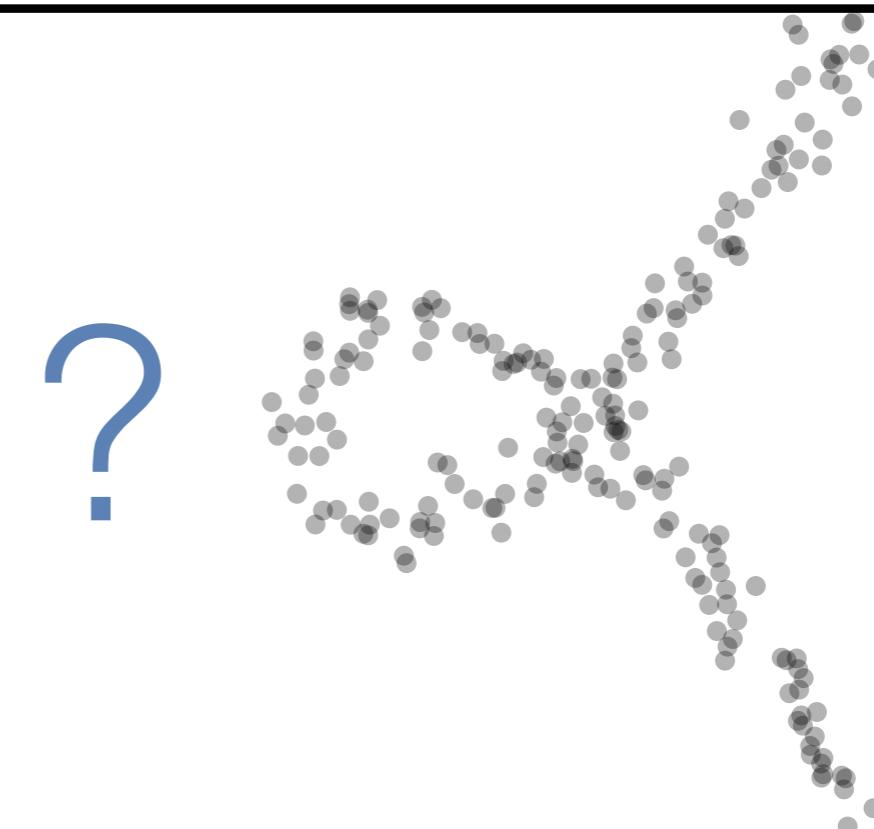
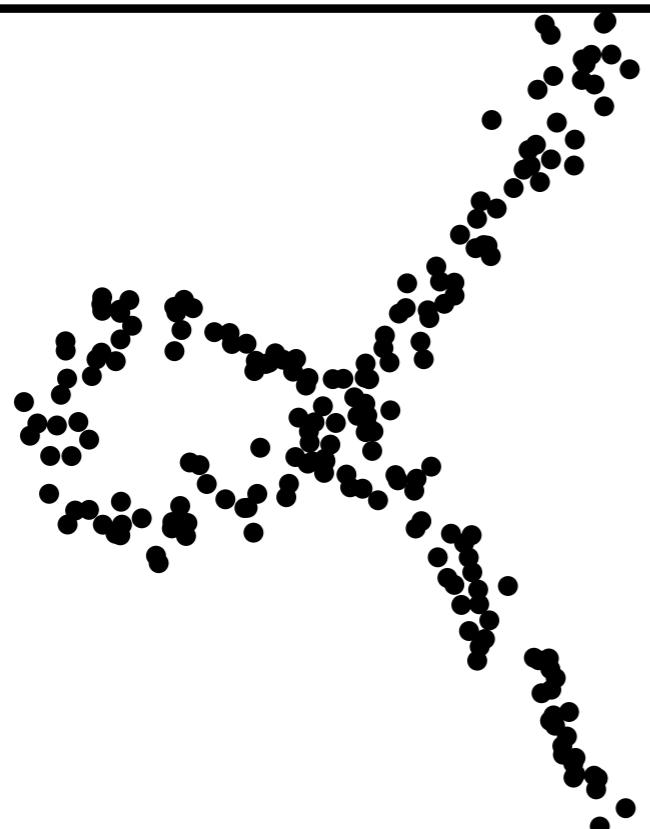
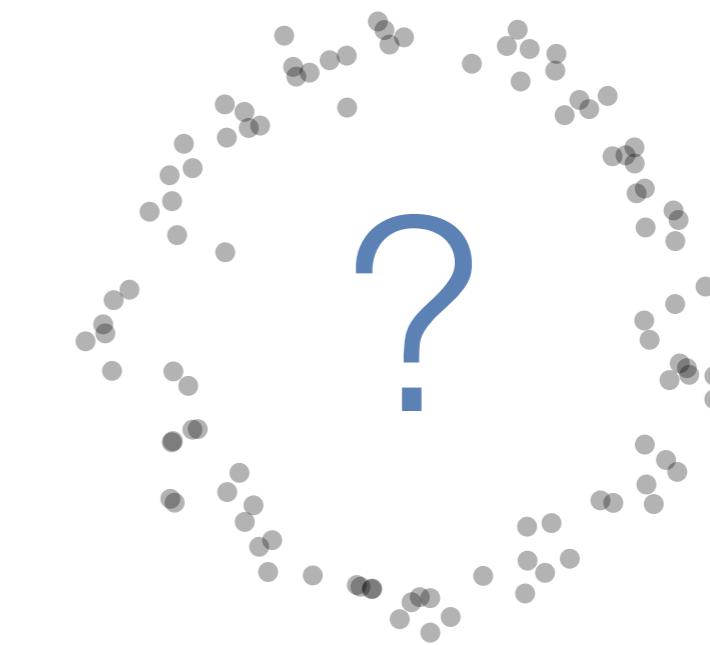
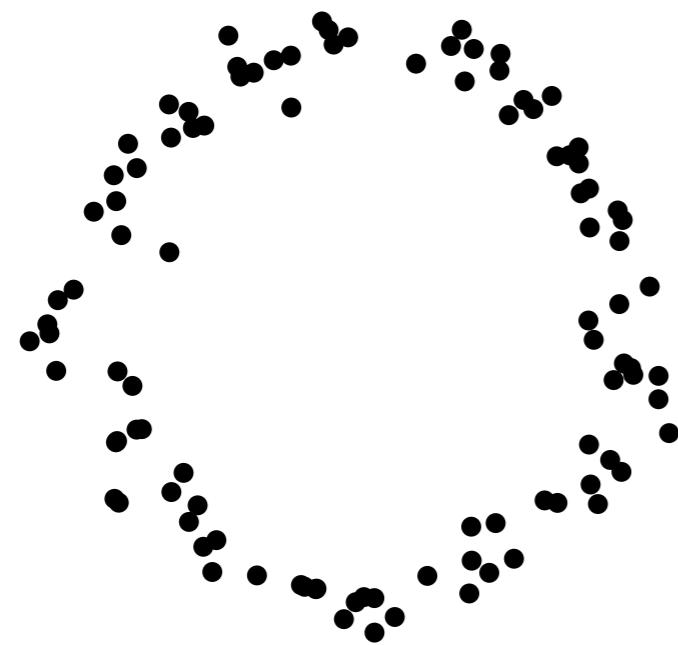


Motivation

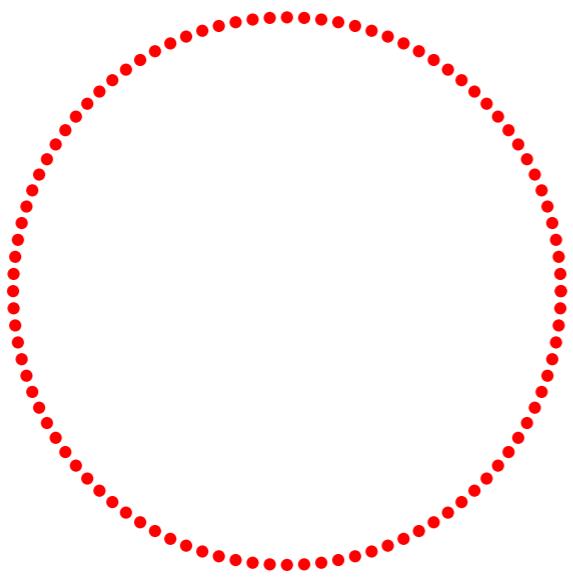
Motivation



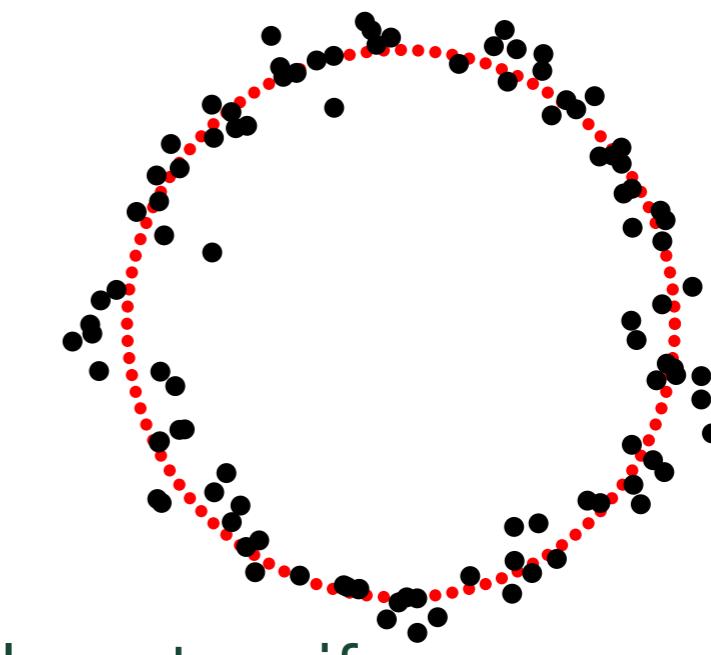
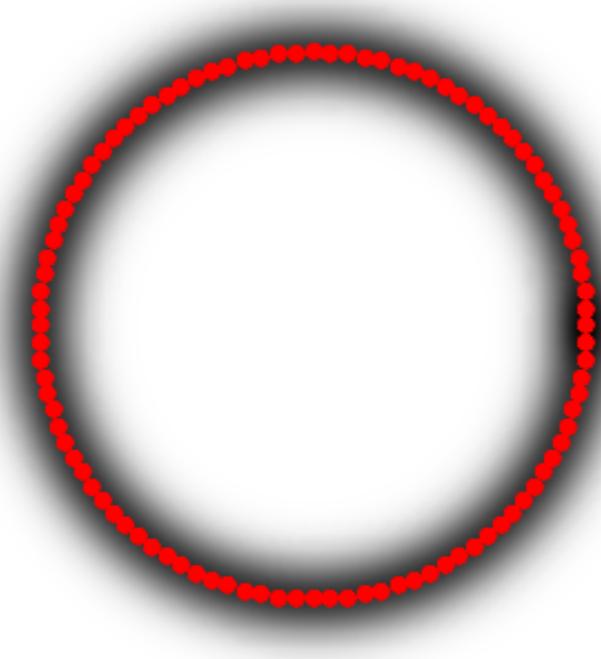
Motivation



Motivation



Add
bivariate
normal
noise



Careful: not uniform on circle!

Problem

Given polynomial(s), generate points near corresponding variety

Applications

Algebraic pattern recognition, TDA, solving nonlinear systems, optimization, gridding, Monte Carlo on semi algebraic sets, ...

Strategy

- Create a distribution with mass near the variety of interest
- Sample from the distribution
- (Optional) Magnetize the sampled points onto the variety

1. Motivation
2. Variety distributions
3. Sampling and implementation
4. Demo and examples
5. Other thoughts

Variety distributions

The normal density is

Partition function, normalizing constant
dependent on parameters

$$p(x|\mu, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} \exp \left\{ -\frac{(x - \mu)^2}{2\sigma^2} \right\}$$

μ is the mean; the center of the bell curve

σ is the standard deviation; governs dispersion about μ

The normal density is

$$p(x|\mu, \sigma) \propto \exp\left\{-\frac{(x - \mu)^2}{2\sigma^2}\right\}$$

Probability mass concentrates near root of polynomial

$$g(x) = g(x|\mu) = x - \mu \in \mathbb{R}[x]$$

Same is true for arbitrary polynomials

$\exp\{-g^2\}$ is largest on the variety, where it has value 1

Decays exponentially as you move away from variety

A random vector \mathbf{X} has the variety normal distribution if

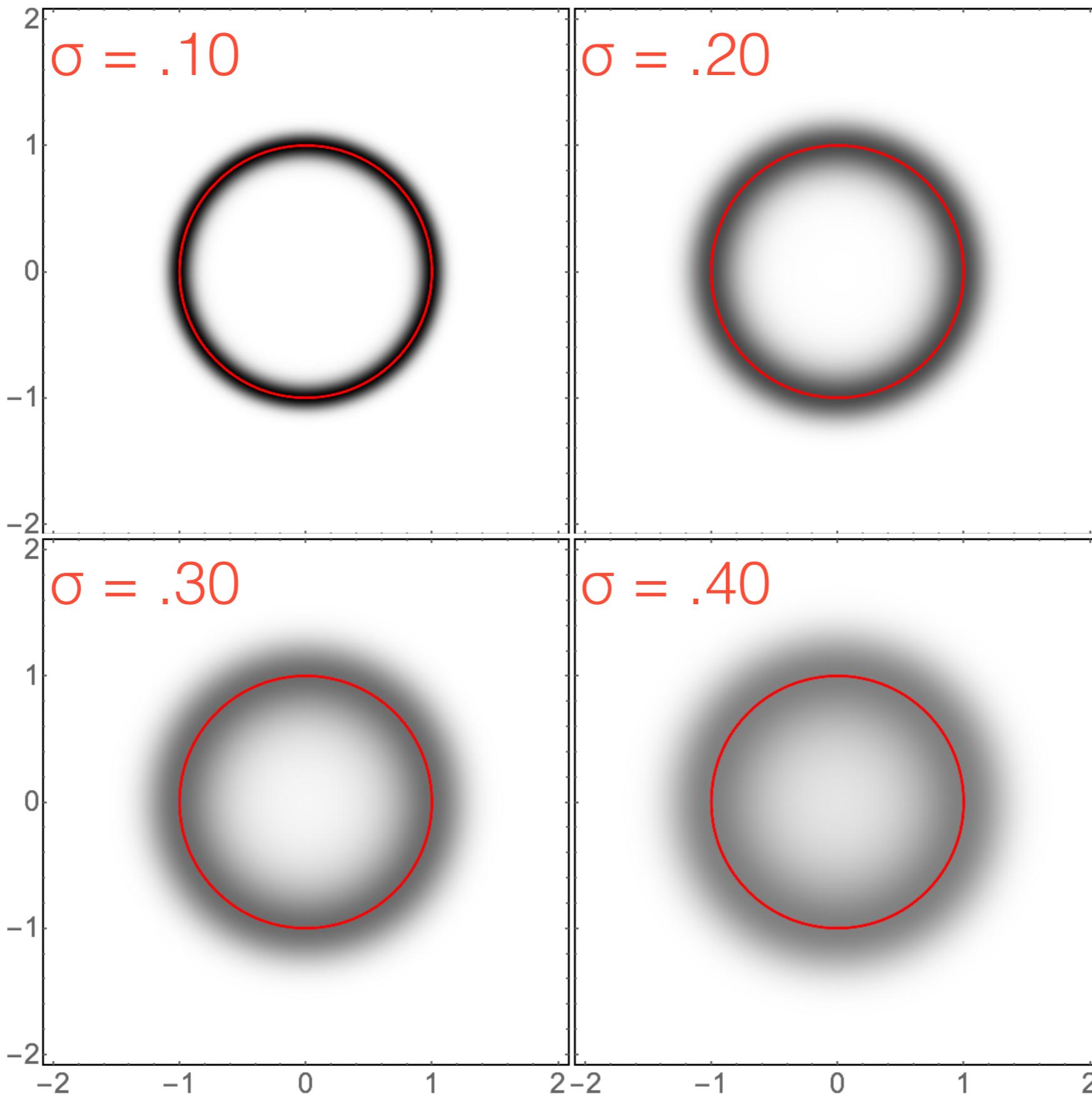
$$p(\mathbf{x}|g, \sigma) \propto \exp \left\{ -\frac{g(\mathbf{x}|\boldsymbol{\beta})^2}{2\sigma^2} \right\}$$

with $g(\mathbf{x}|\boldsymbol{\beta}) \in \mathbb{R}[\mathbf{x}]$

g is “given” : $\boldsymbol{\beta}$ is known and the polynomial form is specified

Example. $\mathbf{X} = (X \ Y)' \sim \mathcal{N}_2(x^2 + y^2 - 1, \sigma)$

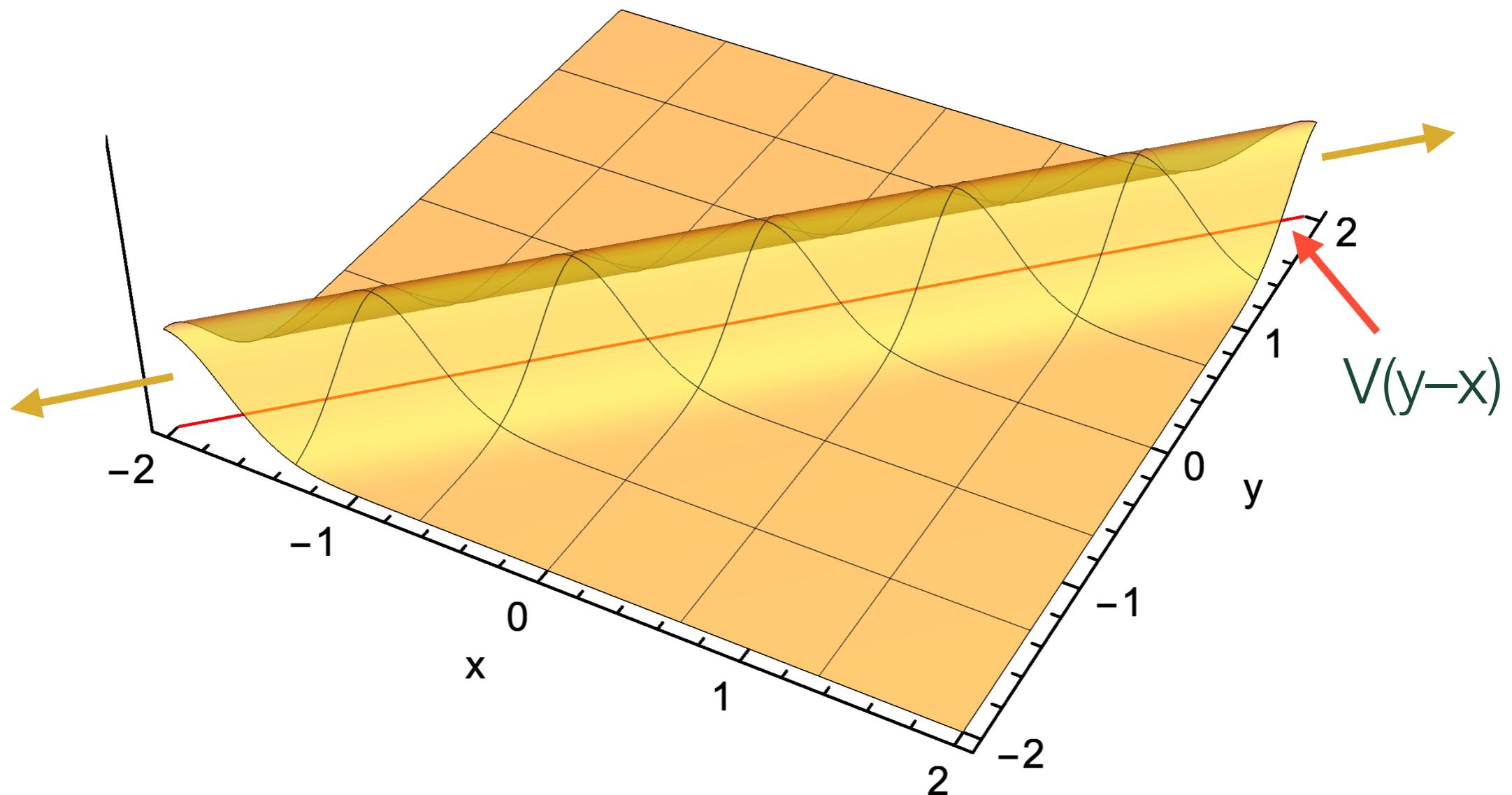
$$p(x, y|g, \sigma) \propto \exp \left\{ -\frac{(x^2 + y^2 - 1)^2}{2\sigma^2} \right\}$$



1. Non-compact varieties

If the variety is unbounded, then it obviously can't be normalized

Example: $g(x, y) = y - x$



1. Non-compact varieties

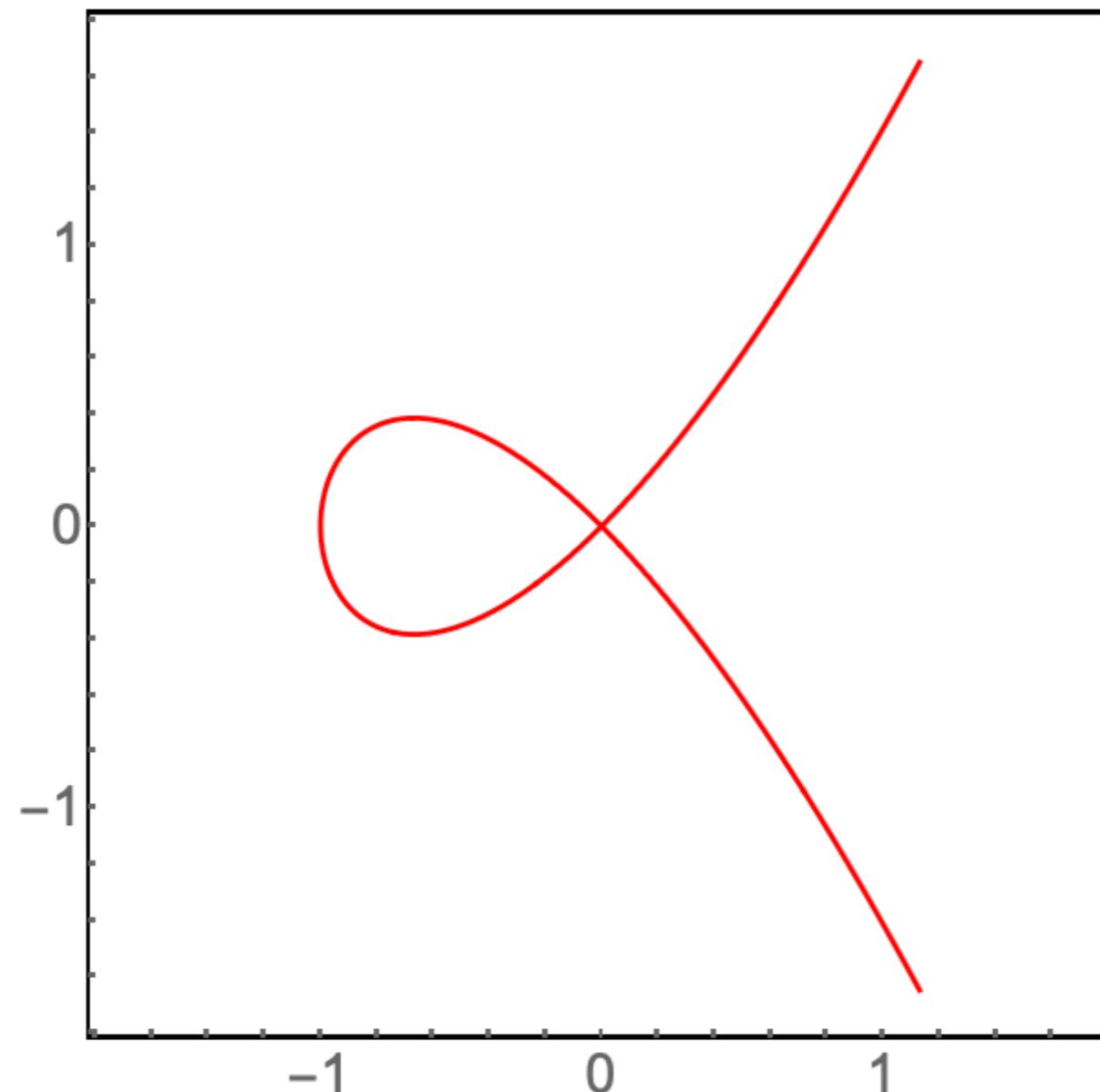
Solution: Truncate or taper

2. σ does not gauge variability globally

2. σ does not gauge variability globally

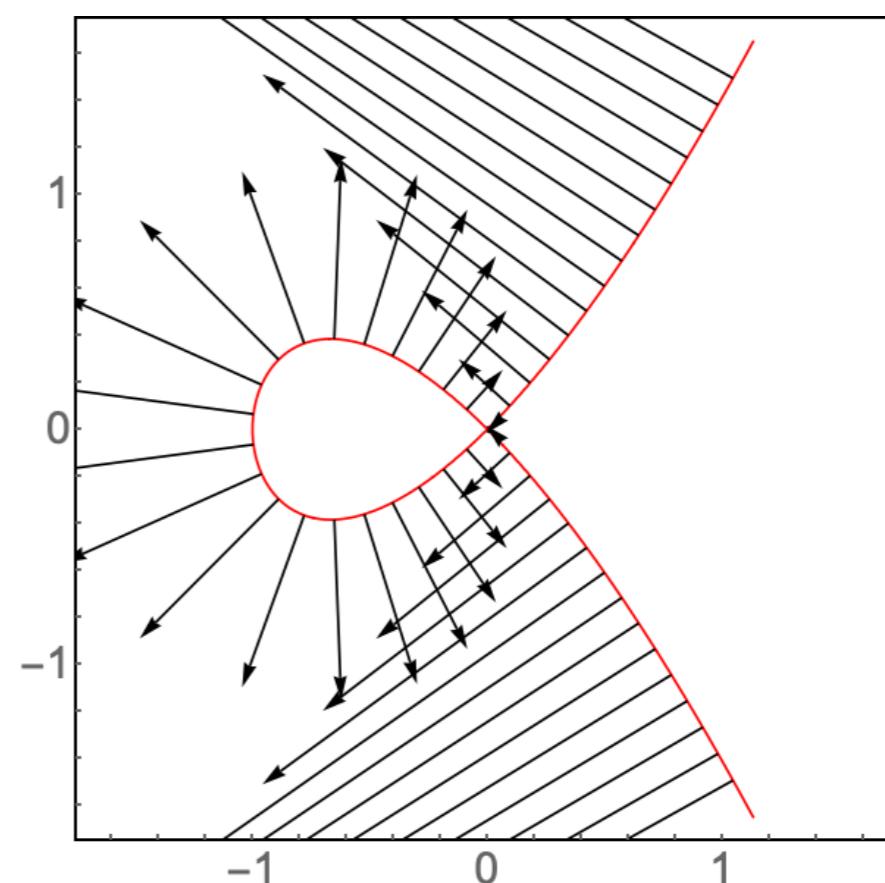
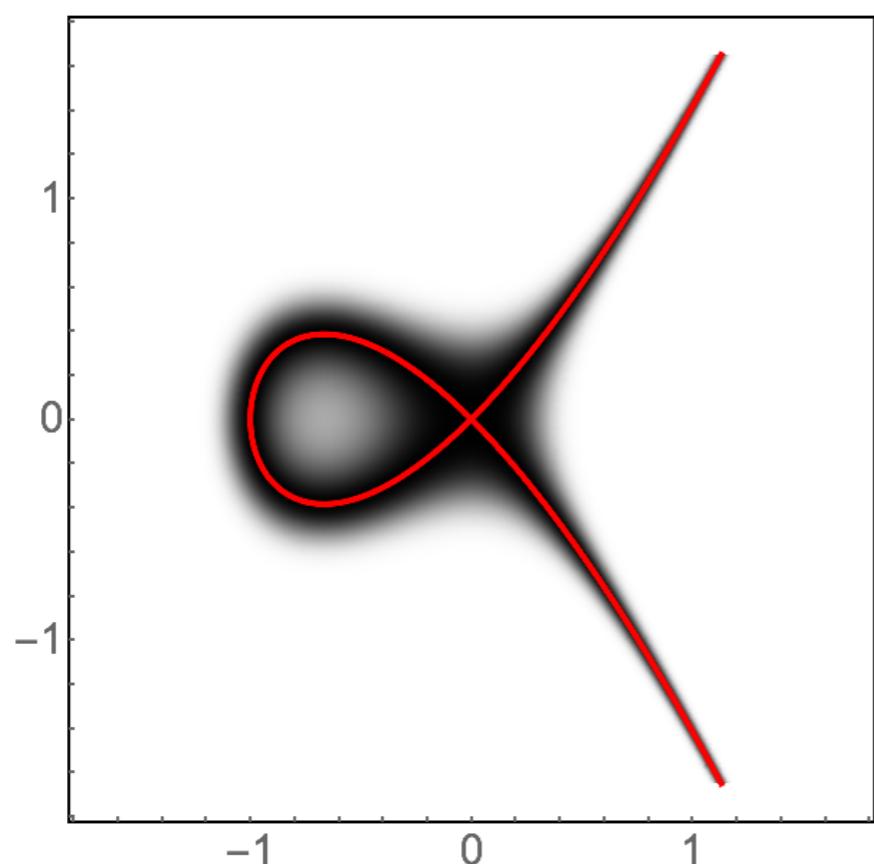
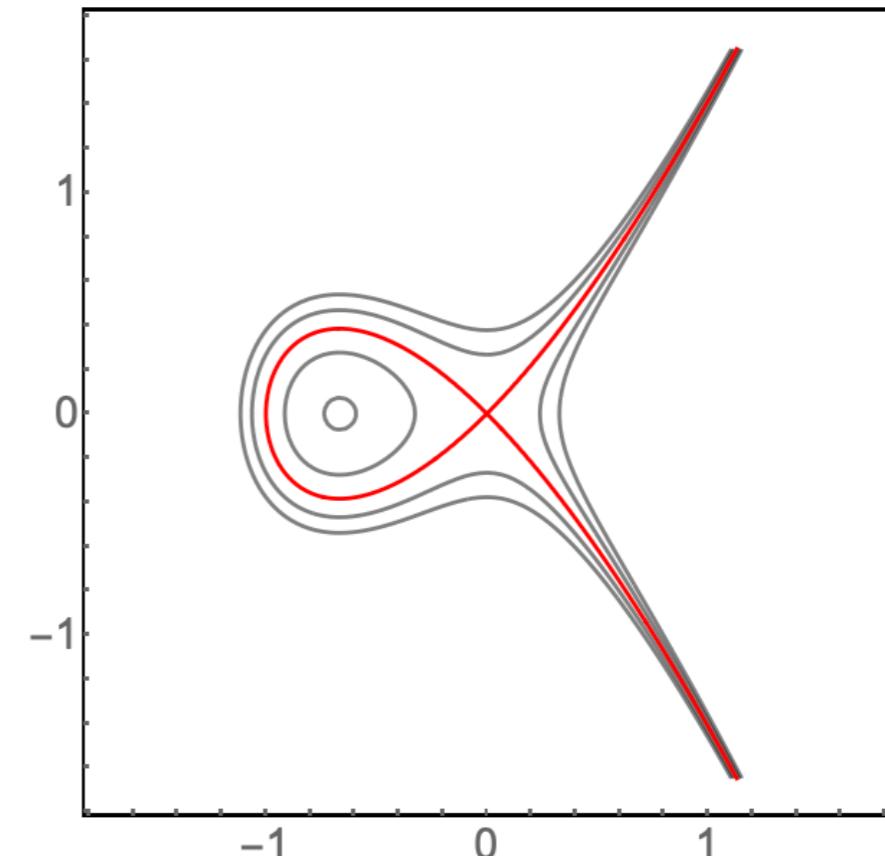
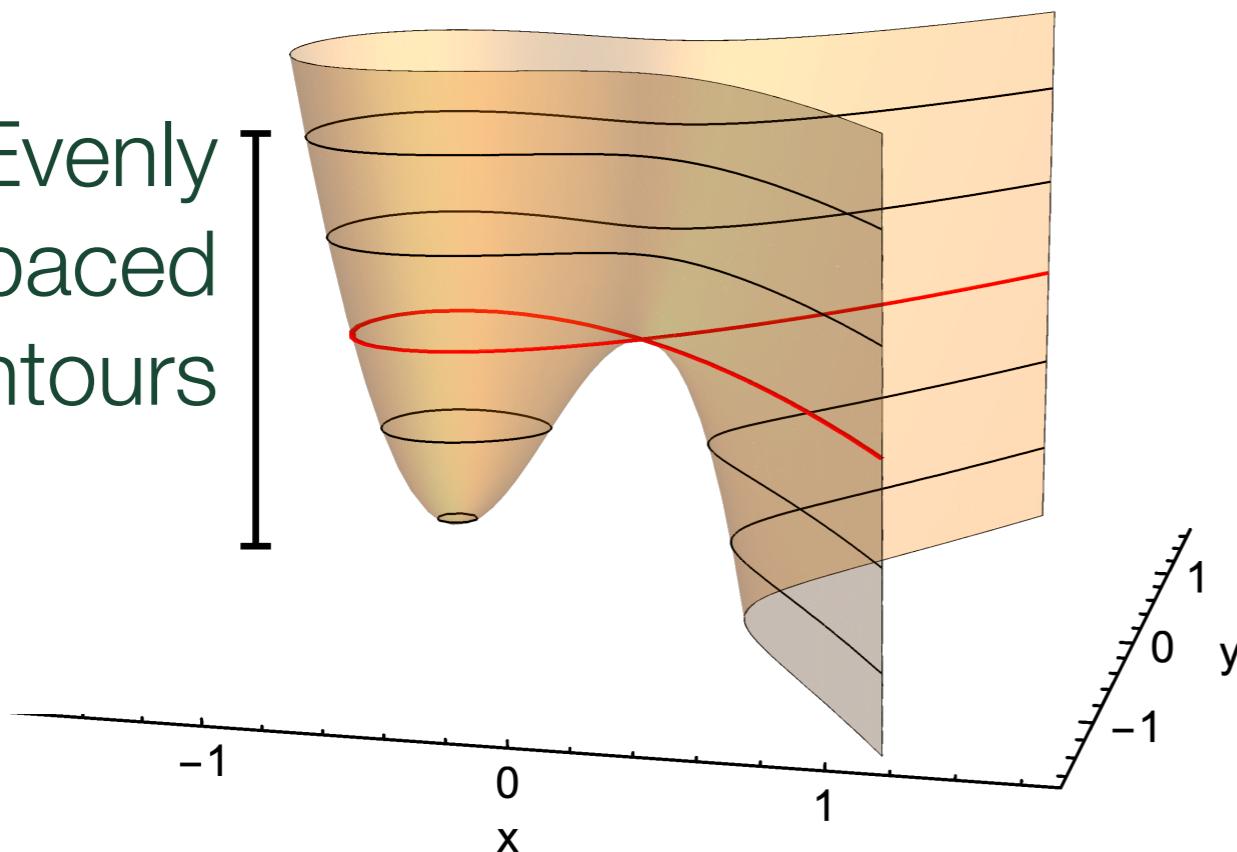
Probability mass does not decay evenly across variety

Example: Alpha curve, $V(y^2 - (x^3 + x^2))$



Variety normal distribution provisional

Evenly
spaced
contours

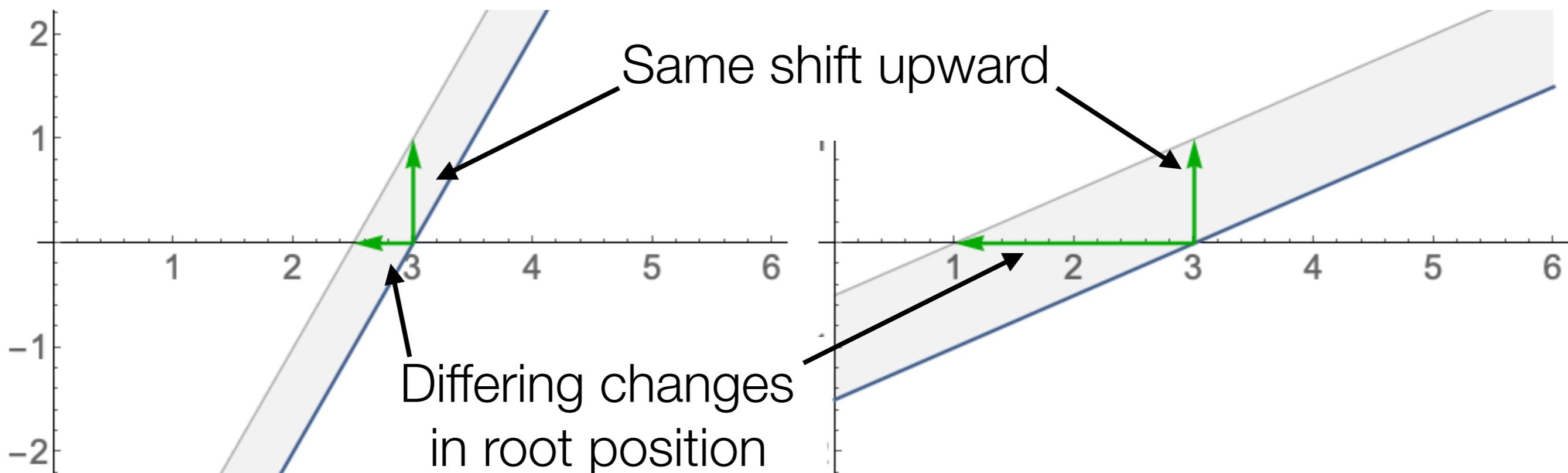


2. σ does not gauge variability globally

Probability mass does not decay evenly across variety

Example: Alpha curve, $V(y^2 - (x^3 + x^2))$

Cause: root mobility is inversely related to gradient, which varies

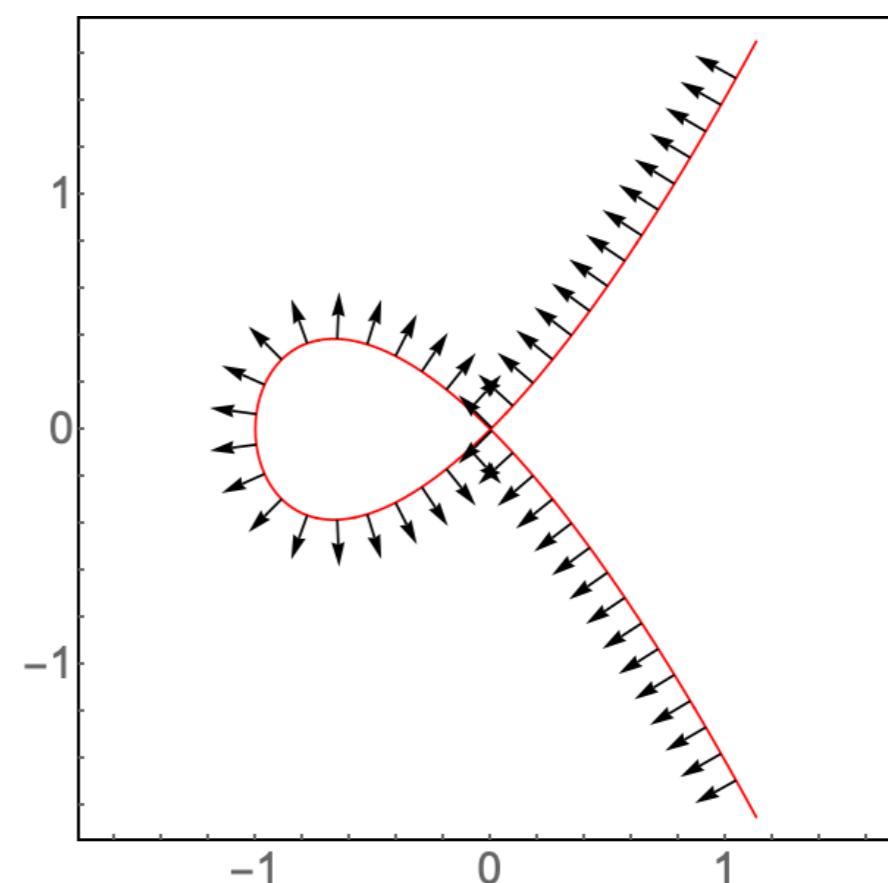
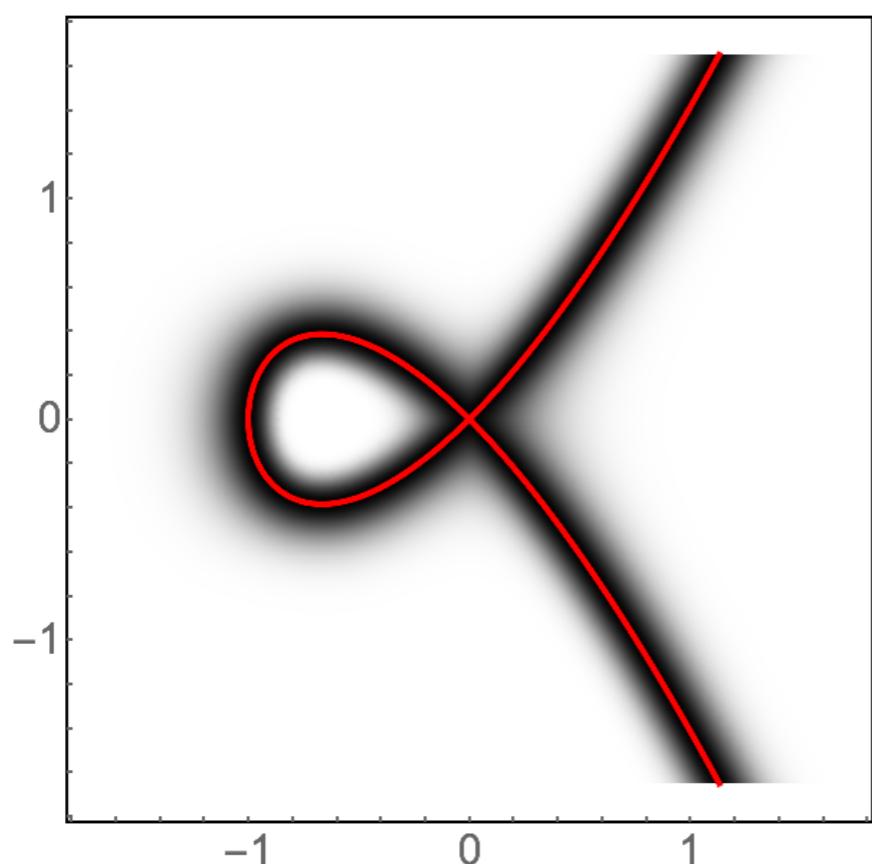
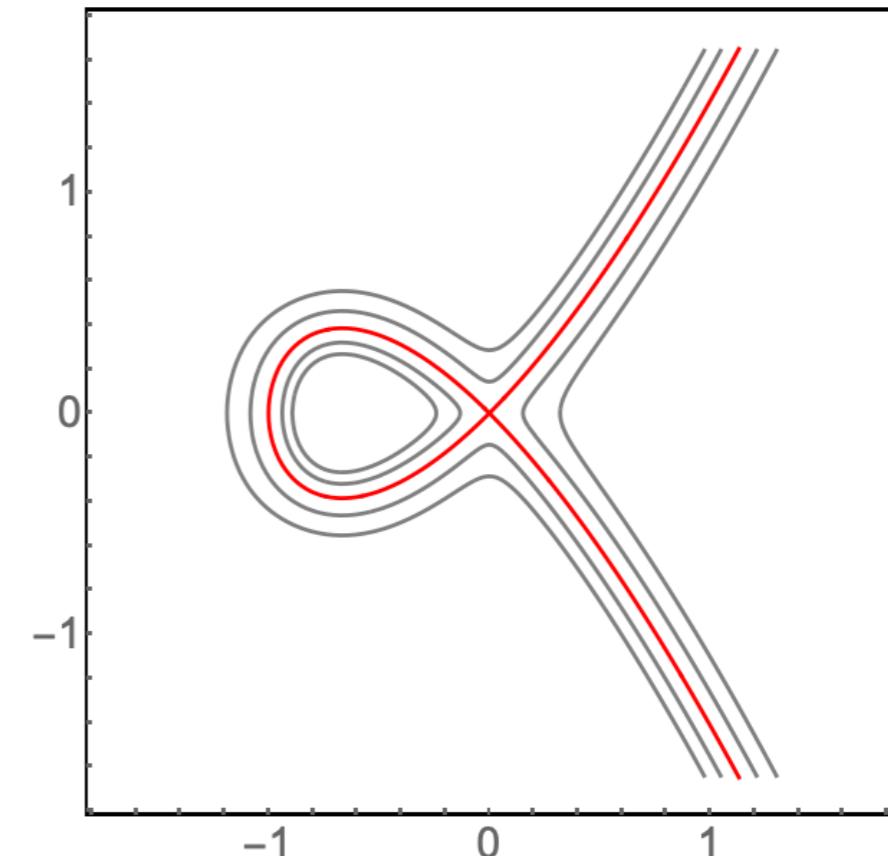
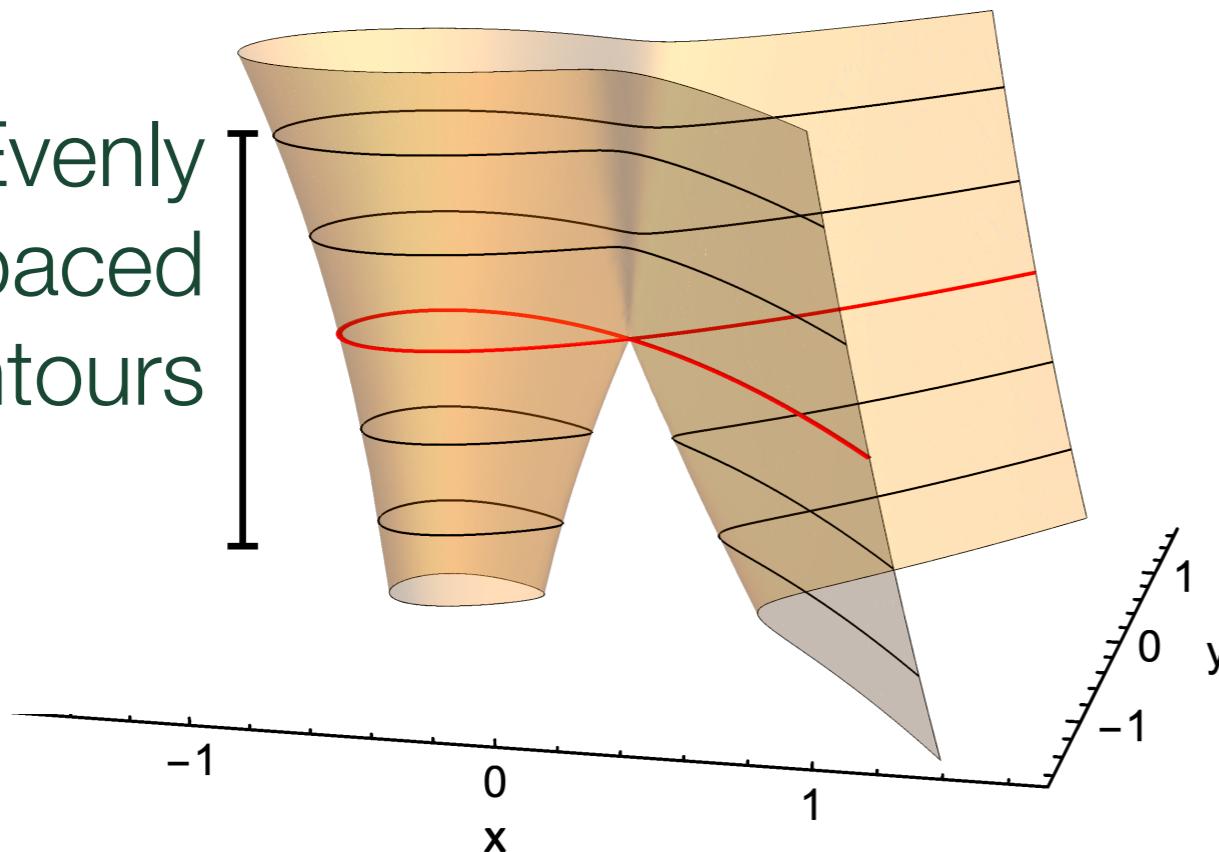


Solution: normalize g by the size of its gradient

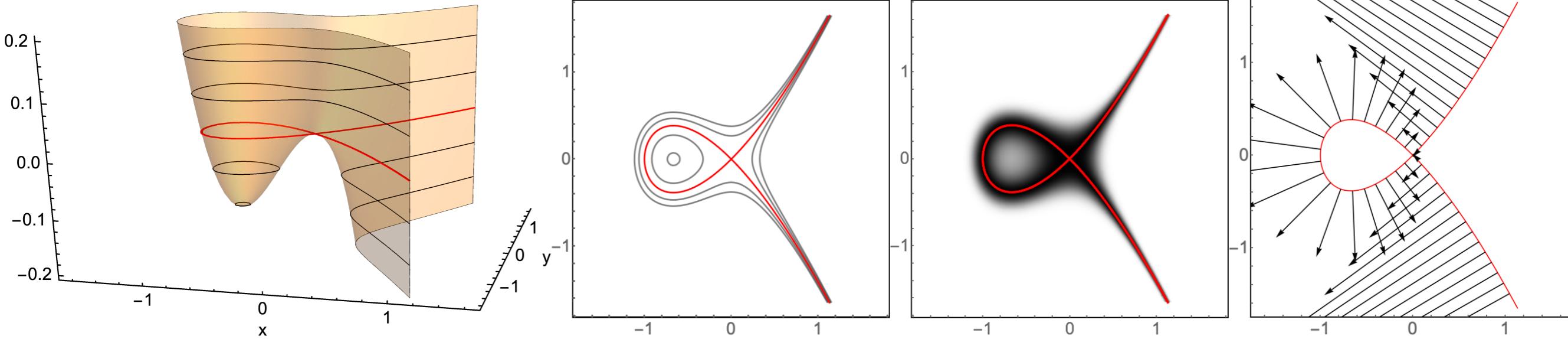
(That doesn't change the zero locus.)

Variety normal distribution provisional

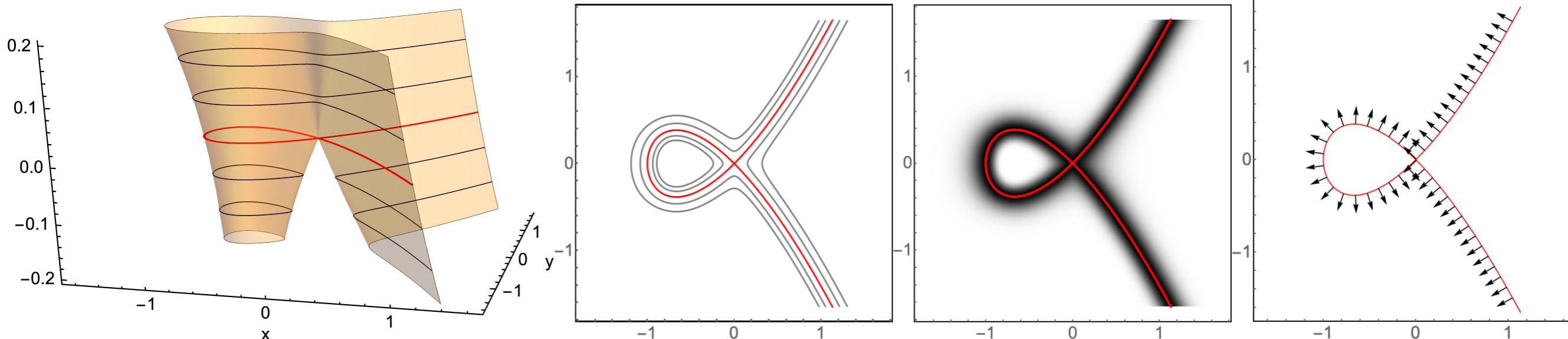
Evenly
spaced
contours



$$p(\mathbf{x}|g, \sigma) \propto \exp \left\{ -\frac{g(\mathbf{x}|\boldsymbol{\beta})^2}{2\sigma^2} \right\}$$



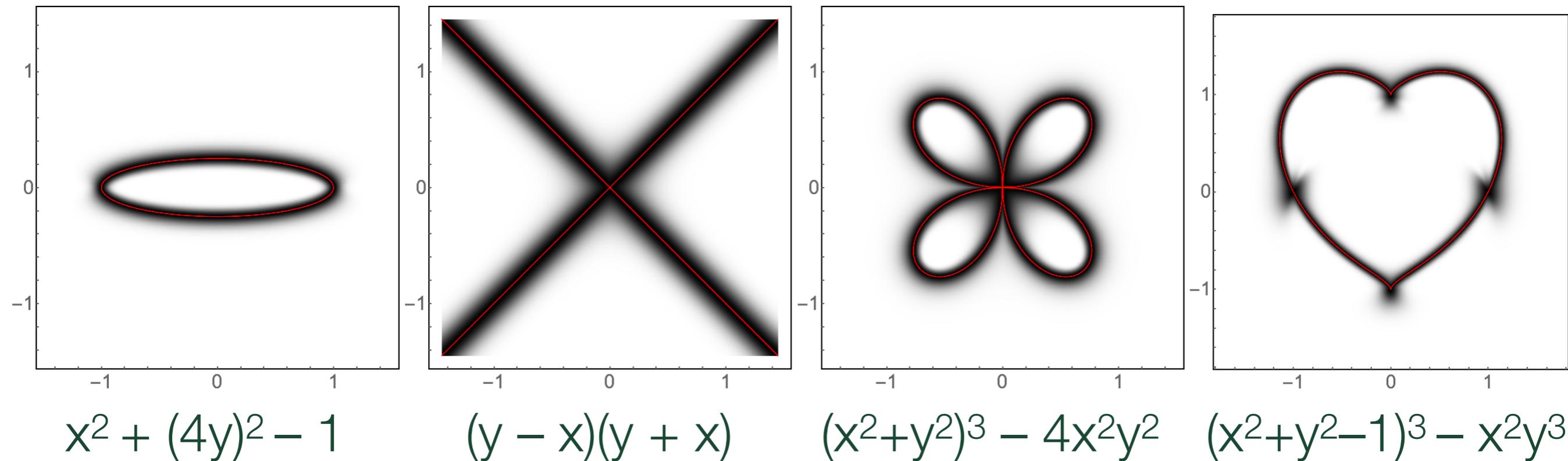
$$p(\mathbf{x}|g, \sigma) \propto \exp \left\{ -\frac{1}{2\sigma^2} \left(\frac{g(\mathbf{x}|\boldsymbol{\beta})}{\|\nabla_{\mathbf{x}}g(\mathbf{x}|\boldsymbol{\beta})\|_2} \right)^2 \right\} = \exp \left\{ -\frac{\bar{g}(\mathbf{x}|\boldsymbol{\beta})^2}{2\sigma^2} \right\}$$



A random vector \mathbf{X} has the variety normal distribution if

$$p(\mathbf{x}|g, \sigma) \propto \exp \left\{ -\frac{1}{2\sigma^2} \left(\frac{g(\mathbf{x}|\boldsymbol{\beta})}{\|\nabla_{\mathbf{x}}g(\mathbf{x}|\boldsymbol{\beta})\|_2} \right)^2 \right\} = \exp \left\{ -\frac{\bar{g}(\mathbf{x}|\boldsymbol{\beta})^2}{2\sigma^2} \right\}$$

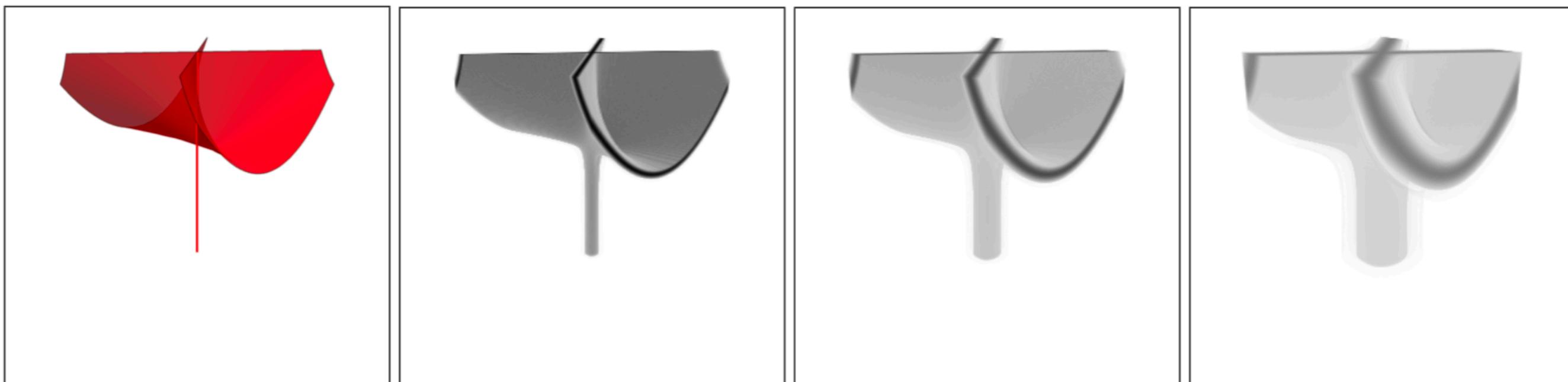
with $g(\mathbf{x}|\boldsymbol{\beta}) \in \mathbb{R}[\mathbf{x}]$



A random vector \mathbf{X} has the variety normal distribution if

$$p(\mathbf{x}|g, \sigma) \propto \exp \left\{ -\frac{1}{2\sigma^2} \left(\frac{g(\mathbf{x}|\boldsymbol{\beta})}{\|\nabla_{\mathbf{x}}g(\mathbf{x}|\boldsymbol{\beta})\|_2} \right)^2 \right\} = \exp \left\{ -\frac{\bar{g}(\mathbf{x}|\boldsymbol{\beta})^2}{2\sigma^2} \right\}$$

with $g(\mathbf{x}|\boldsymbol{\beta}) \in \mathbb{R}[\mathbf{x}]$



Whitney umbrella $V(x^2 - y^2 z)$ for differing σ

Systems of polynomials g_1, \dots, g_m are supported by the multivariety normal distribution

The multivariate normal distribution has density

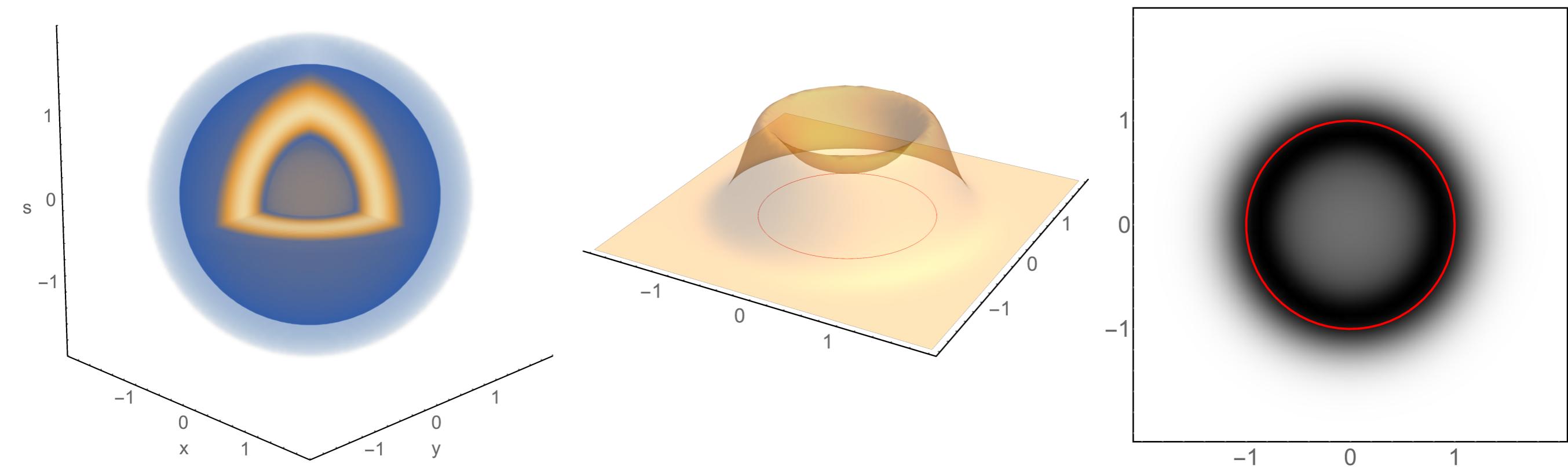
$$p(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) \propto \exp \left\{ -\frac{1}{2} (\mathbf{x} - \boldsymbol{\mu})' \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu}) \right\}$$

The multivariety normal distribution has density

$$\begin{aligned} p(\mathbf{x}|g, \boldsymbol{\Sigma}) &\propto \exp \left\{ -\frac{1}{2} \bar{\mathbf{g}}(\mathbf{x}|\mathbf{B})' \boldsymbol{\Sigma}^{-1} \bar{\mathbf{g}}(\mathbf{x}|\mathbf{B}) \right\} \\ &= \exp \left\{ -\frac{1}{2} (\mathbf{J}_x^+ g(\mathbf{x}|\mathbf{B}))' \boldsymbol{\Sigma}^{-1} (\mathbf{J}_x^+ g(\mathbf{x}|\mathbf{B})) \right\} \end{aligned}$$

Distributions about semi-algebraic regions can be created through slack variables and marginalization

$$\begin{aligned}x^2 + y^2 \leq 1 &\iff 0 \leq 1 - (x^2 + y^2) \\&\iff 0 = 1 - (x^2 + y^2) - s^2\end{aligned}$$



Sampling and implementation

Markov chain Monte Carlo (MCMC) is a class of algorithms for sampling probability distributions

Stationary distribution is the target distribution

Target distribution does not need to be normalized

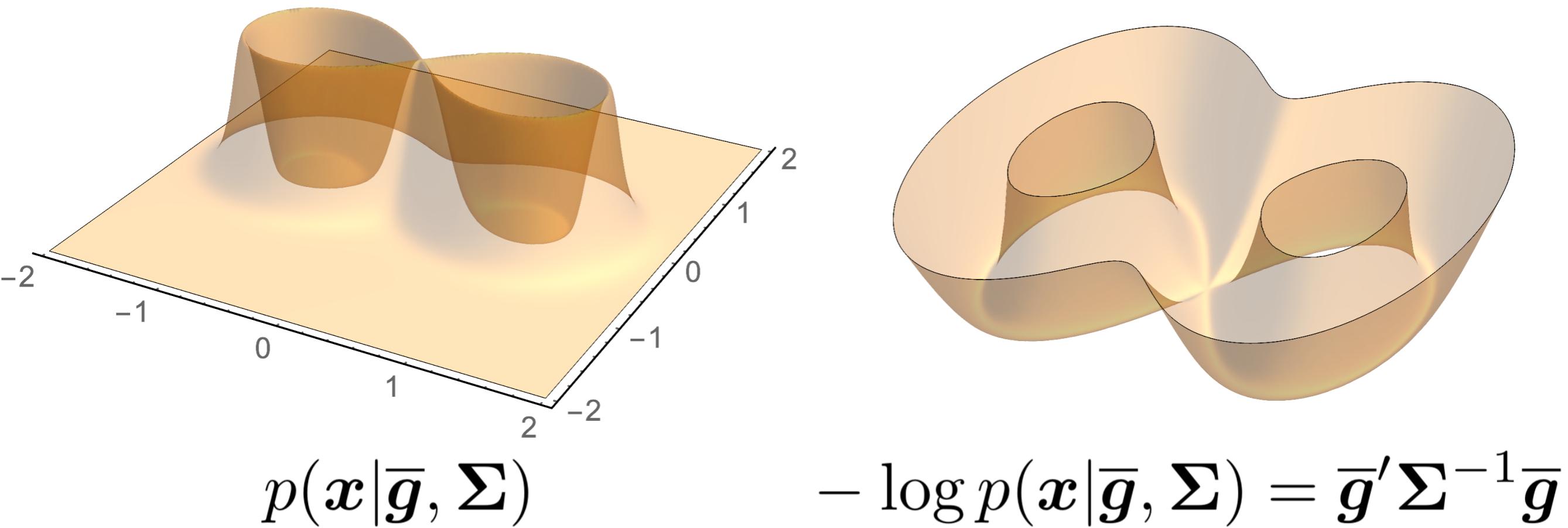
Foundational in Bayesian statistics \Rightarrow good software (BUGS, Stan)

Iterate two basic steps (MCMC used here)

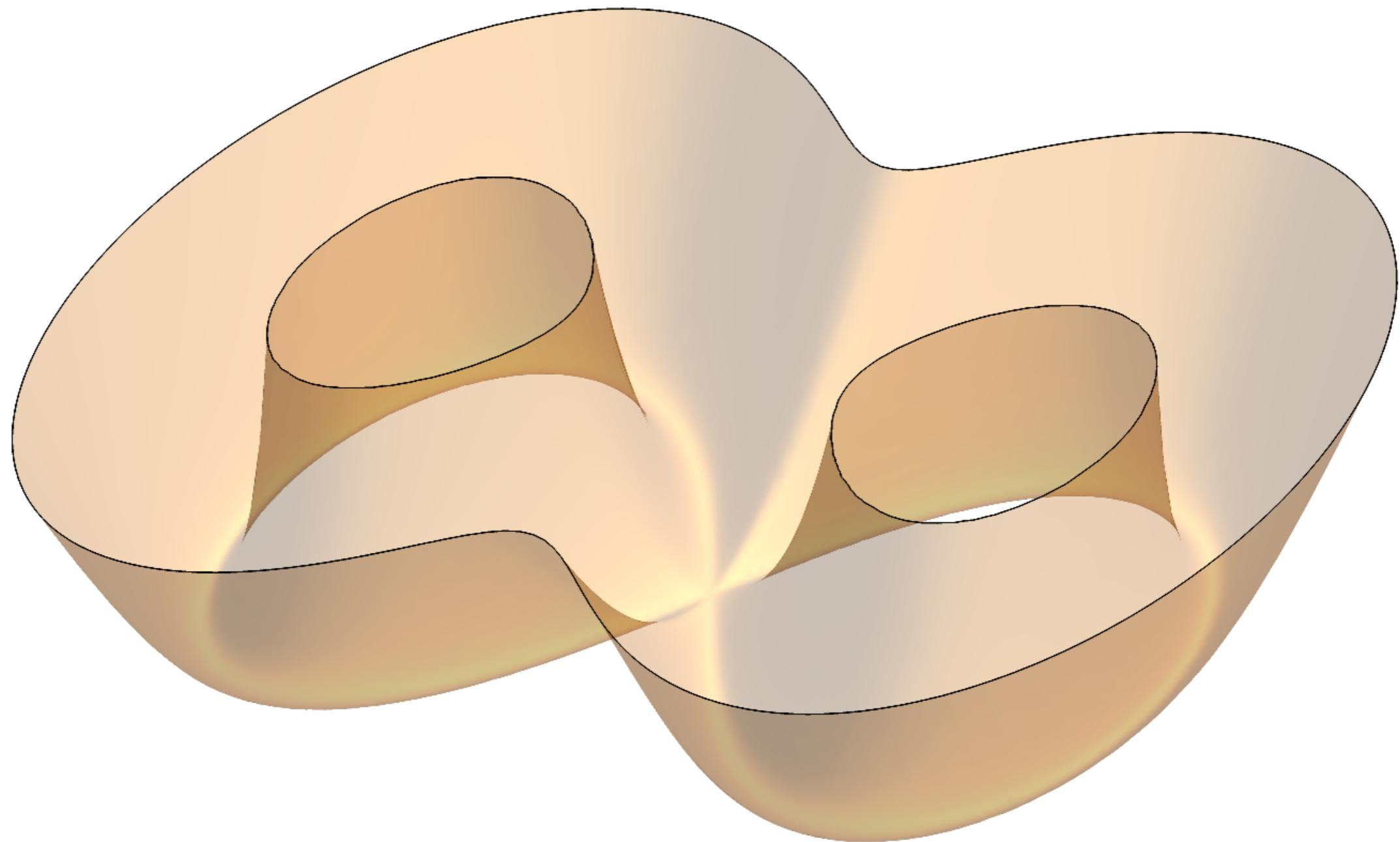
1. Generate an observation that might come from target (proposal)
2. Accept/reject probabilistically according to Metropolis-Hastings

Best case: Starting anywhere, chain converges to draws from target distribution

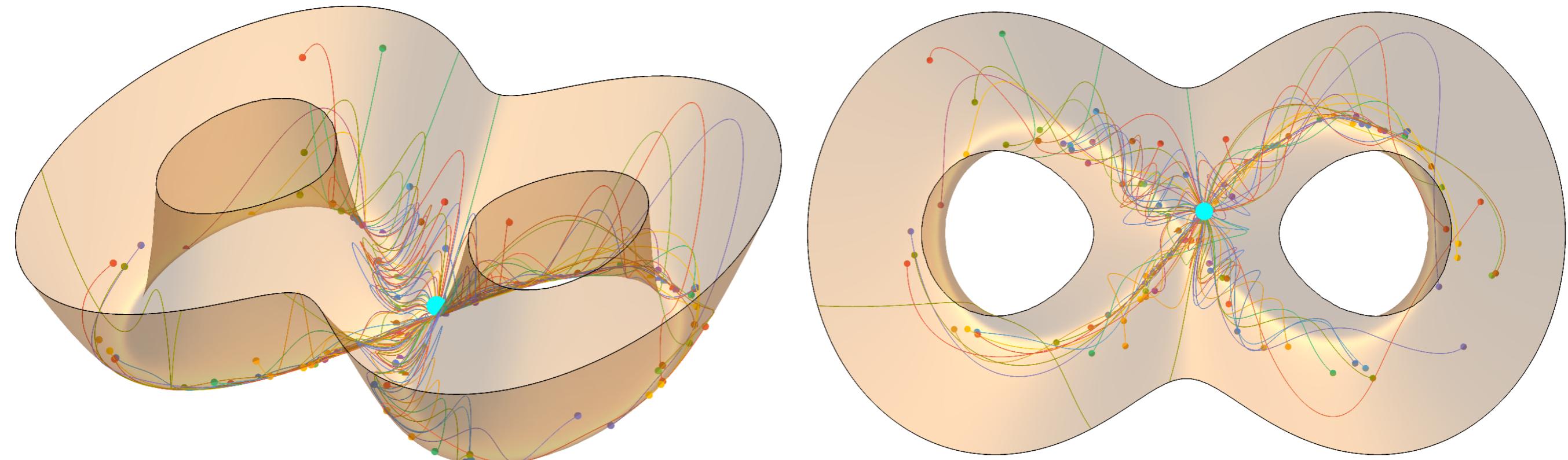
From current, propose step from physics simulation
Marble rolling on (\bar{g}^2/σ^2) 's surface, frictionless, given initial flick



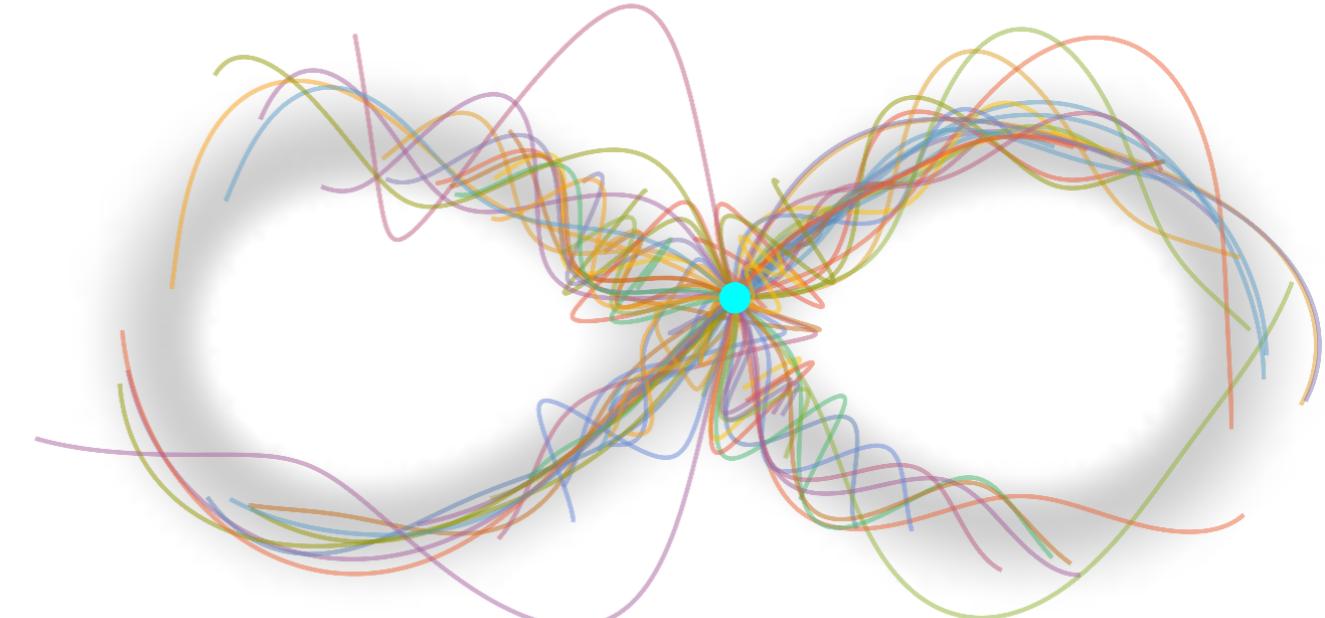
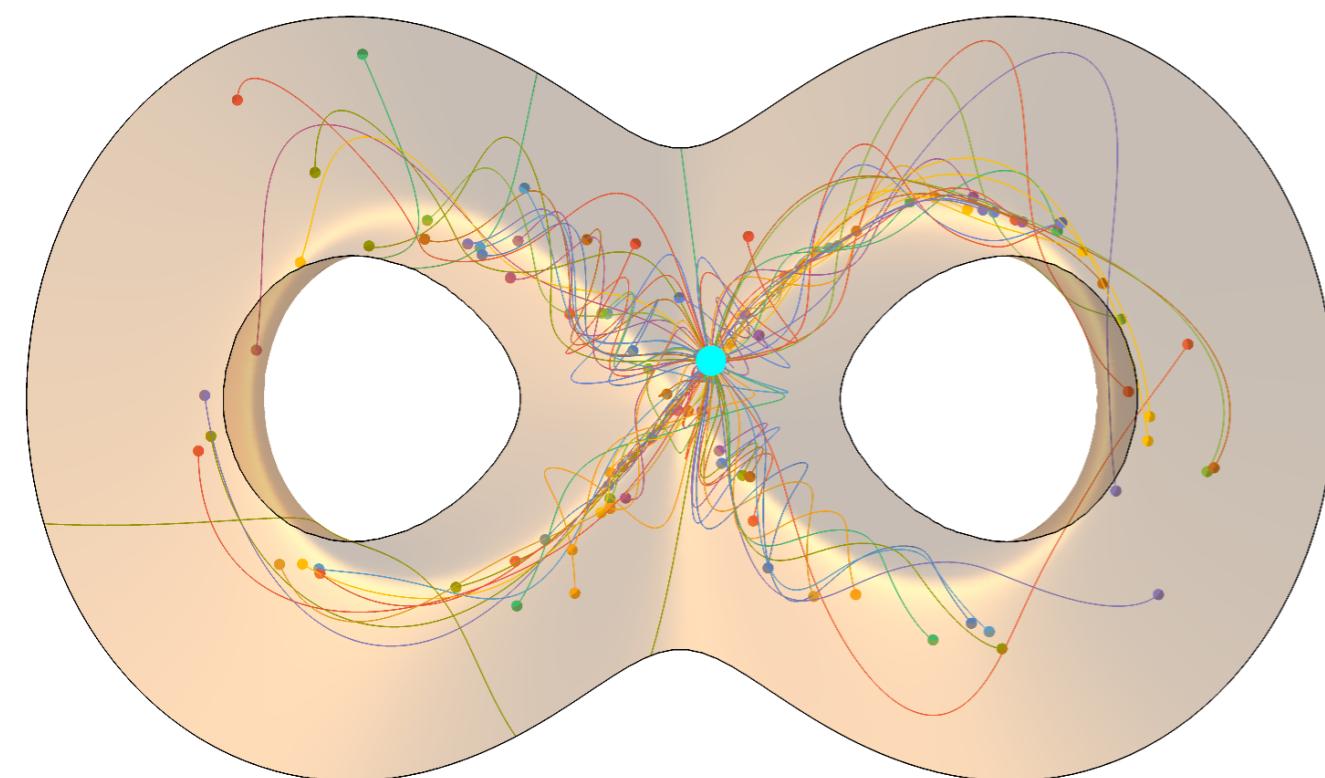
Impart random momentum, track position numerically, stop



100 proposed transitions from a single state

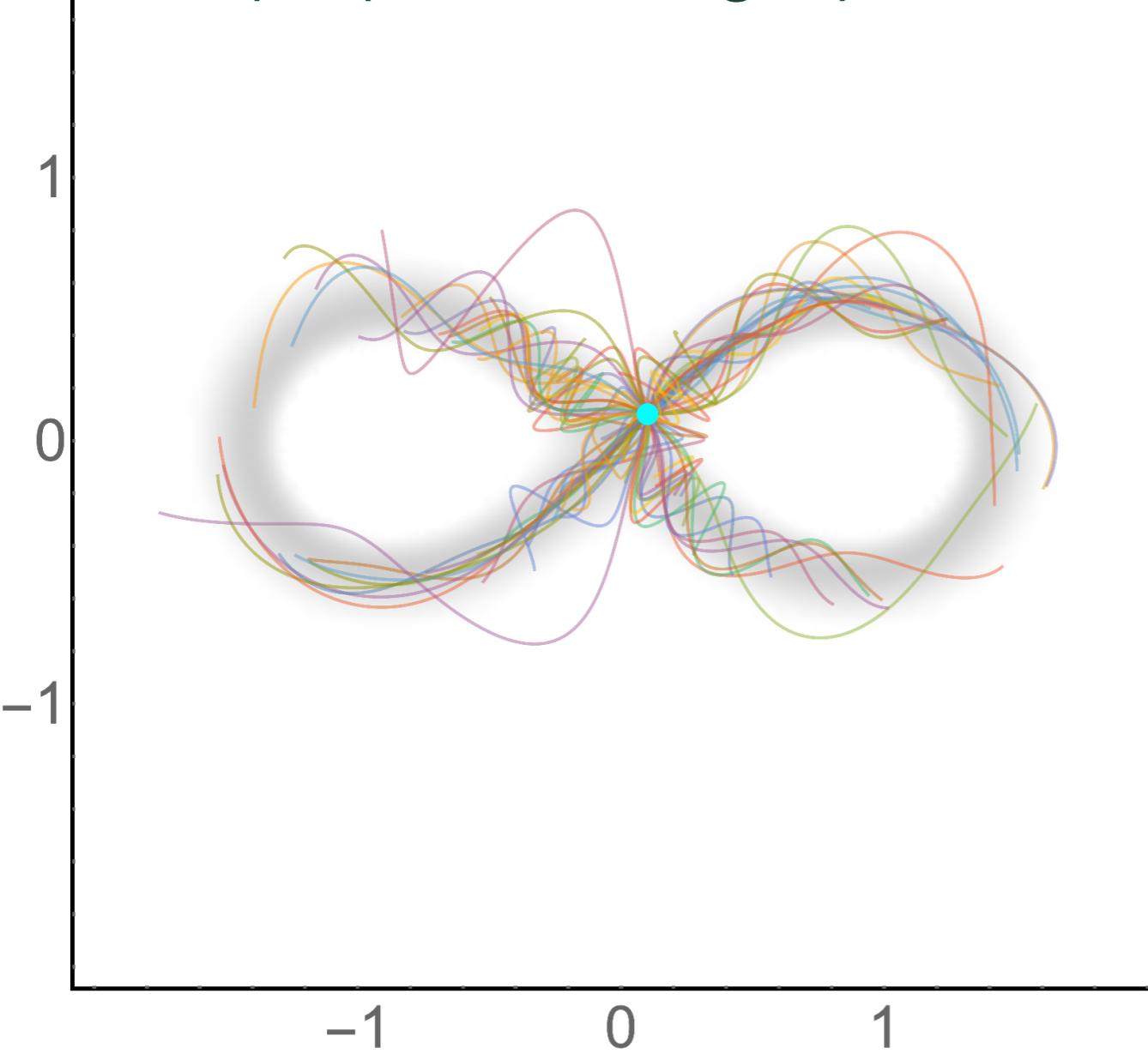


100 proposed transitions from a single state

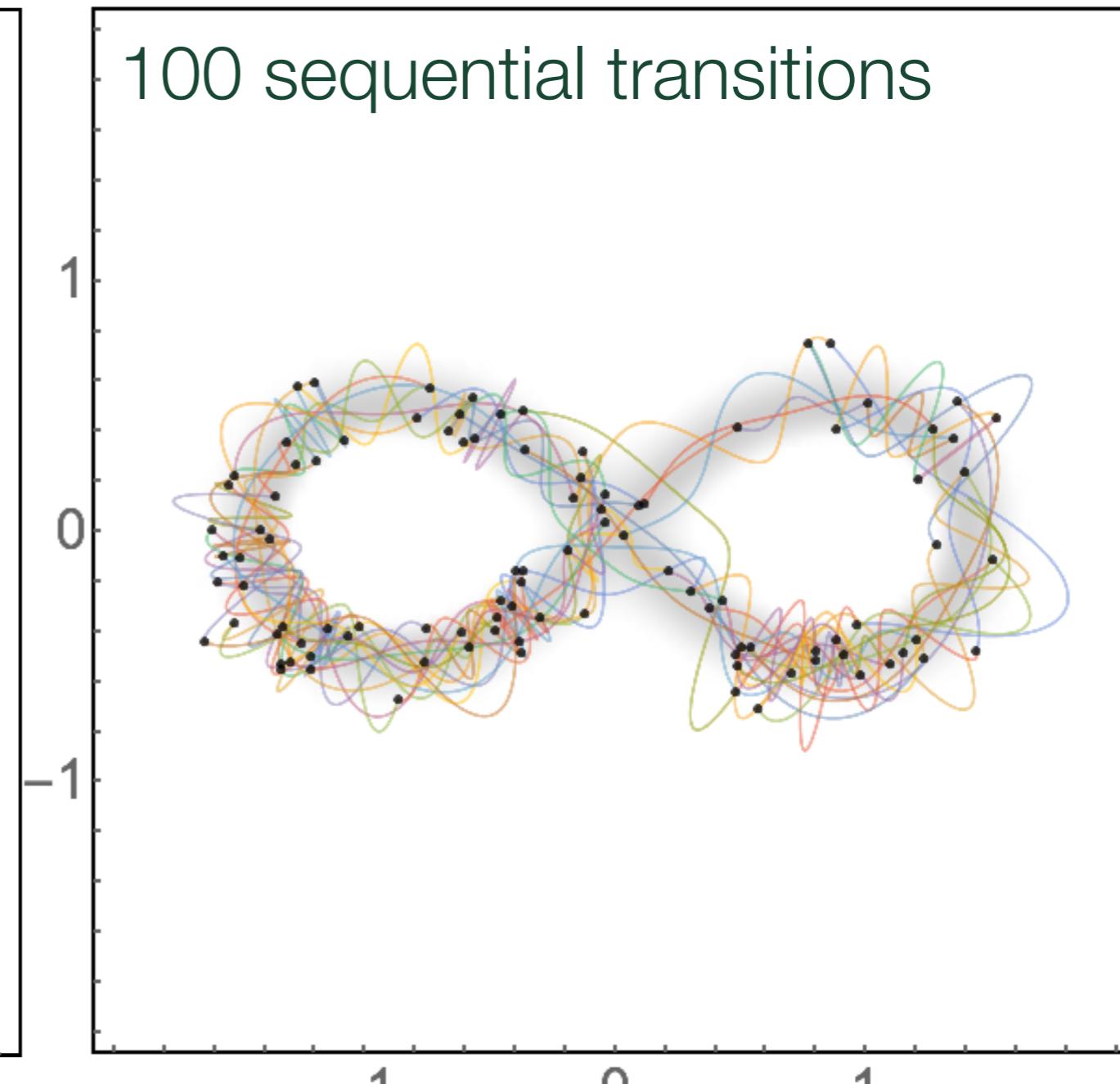


(Path colors inconsistent)

100 proposals, single position



100 sequential transitions



$$(x^2 + y^2)^2 - 2(x^2 - y^2)$$

1

2

3

4

5

6

7

8

9

10

11

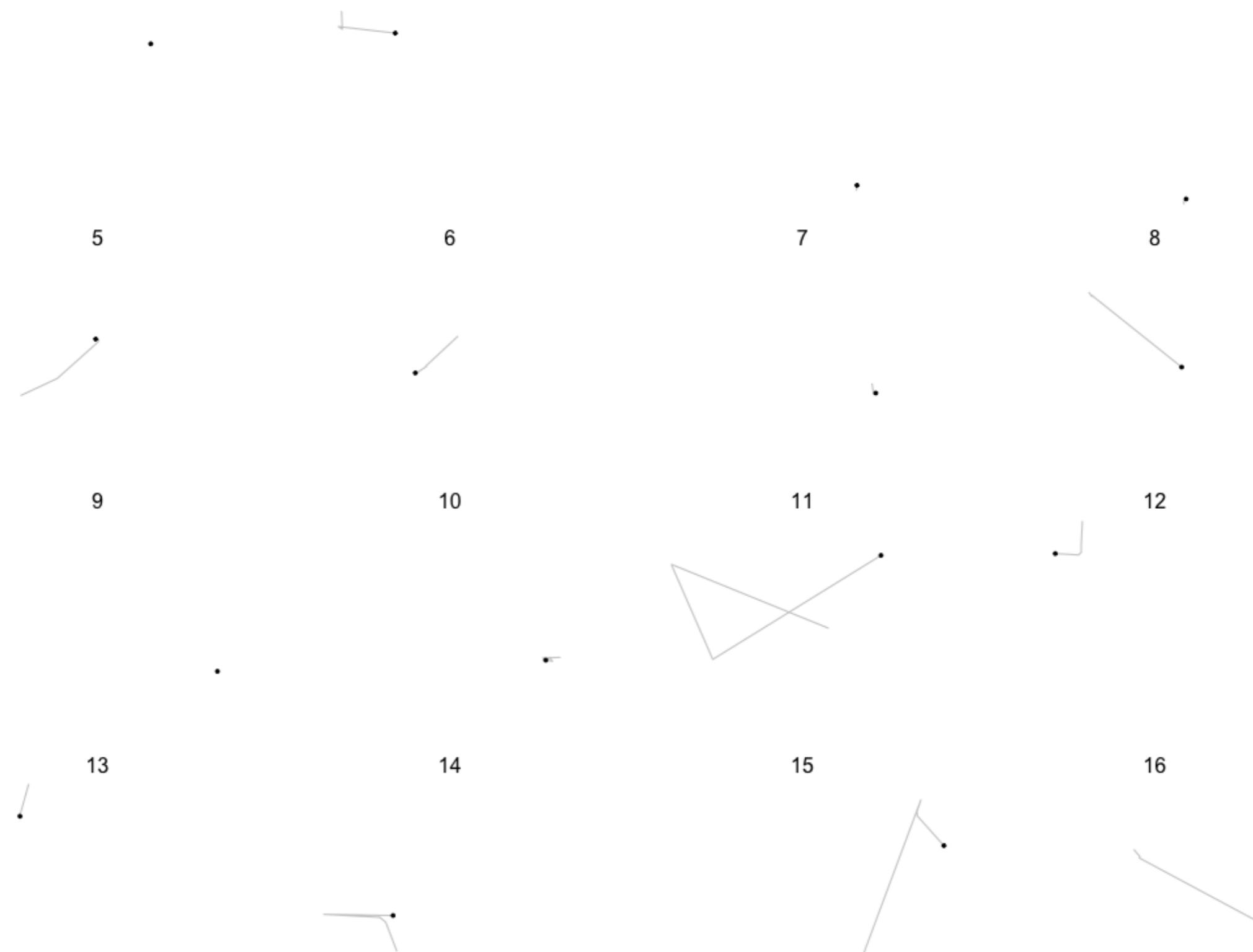
12

13

14

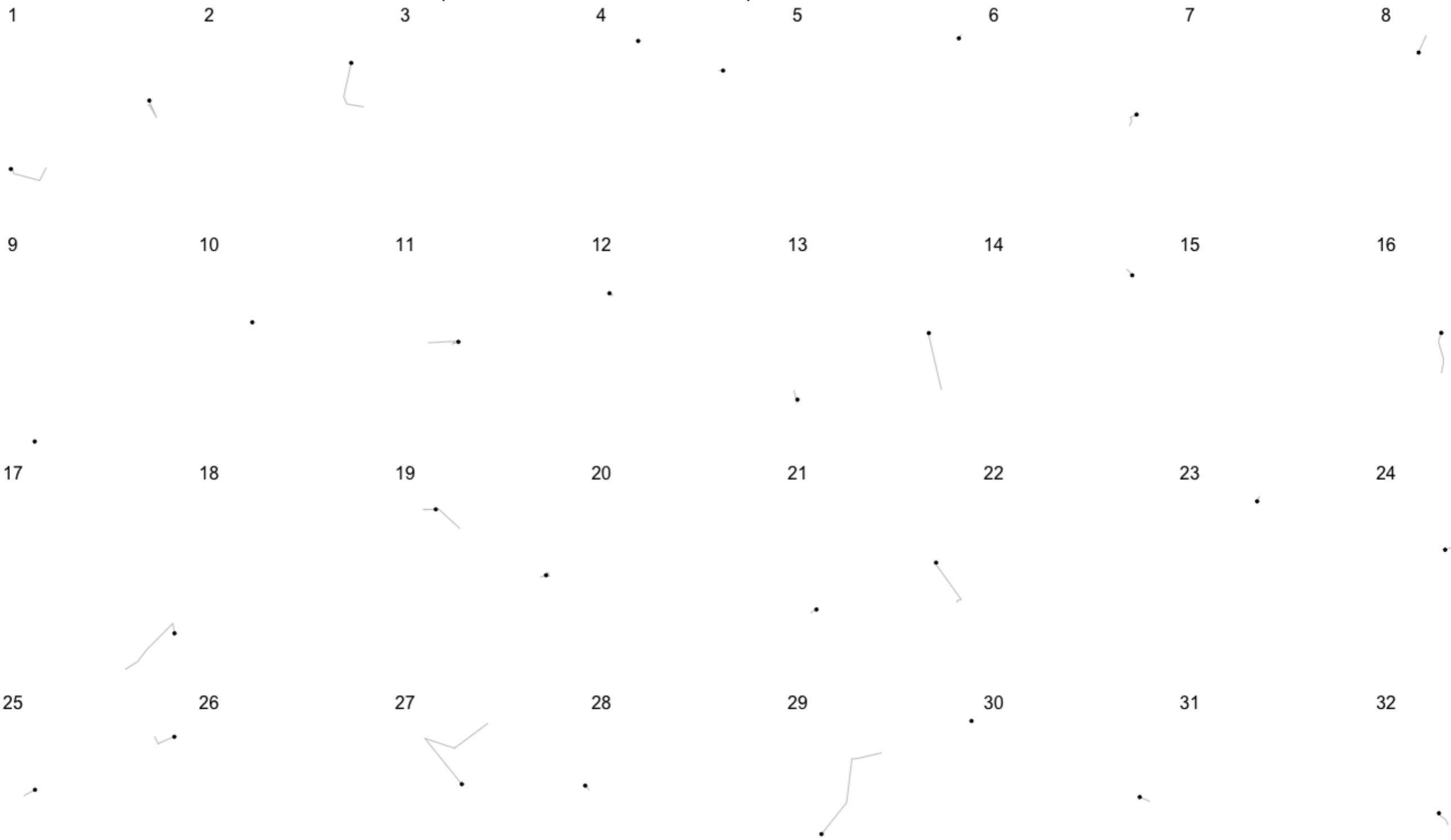
15

16



Hamiltonian Monte Carlo (HMC)

$$(x^2 + y^2 - 1)^3 - x^2 y^3$$



HMC is implemented in Stan, a probabilistic programming language and Bayesian engine

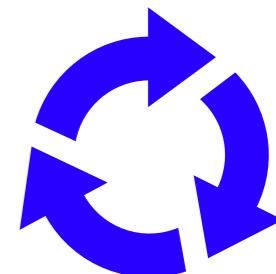
Stan specification

```
data {  
    real<lower=0> si;  
}  
  
parameters {  
    real x;  
    real y;  
}  
  
transformed parameters {  
    real g = x^2+y^2-1;  
    real ndg = sqrt(4*x^2+4*y^2);  
}  
  
model {  
    target += normal_lpdf(0.00 | g, ndg*si);  
}
```

C++

```
// Code generated by Stan version 2.17.0  
  
#include <stan/model/model_header.hpp>  
  
namespace model867873611022_stan_code_namespace {  
  
    using std::istream;  
    using std::string;  
    using std::stringstream;  
    using std::vector;  
    using stan::io::dump;  
    using stan::math::lgamma;  
    using stan::model::prob_grad;  
    using namespace stan::math;  
  
    typedef Eigen::Matrix<double, Eigen::Dynamic, 1> vec;  
    typedef Eigen::Matrix<double, 1, Eigen::Dynamic> row;  
    typedef Eigen::Matrix<double, Eigen::Dynamic, Eigen::Dynamic> m
```

Sample



Interfaces : R, Julia, Python, CLI, ...

Many chains can be run in parallel

1. Install R

<https://cran.rstudio.com>

2. Install RStudio

<https://rstudio.com/products/rstudio/download/>

3. Install rstan

<https://github.com/stan-dev/rstan/wiki/RStan-Getting-Started>

4. Install packages

```
source("https://tinyurl.com/y3hghhx")
```

5. Explore with `explore.R!`

Demo and examples

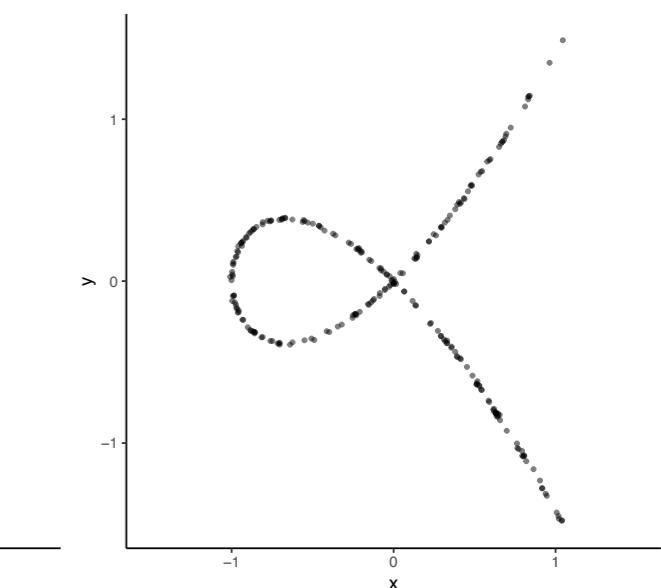
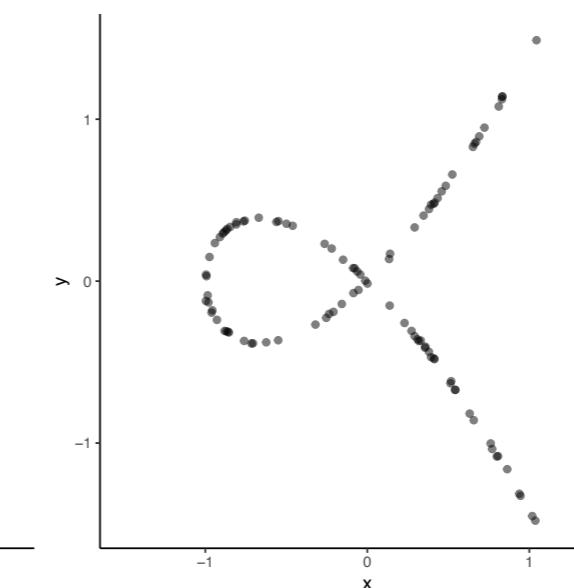
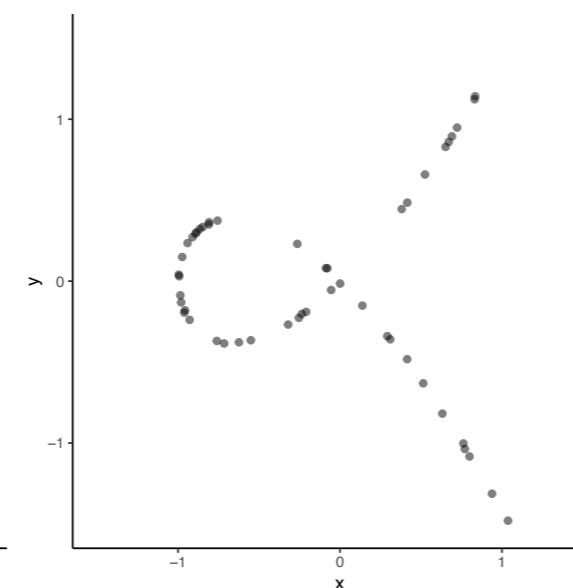
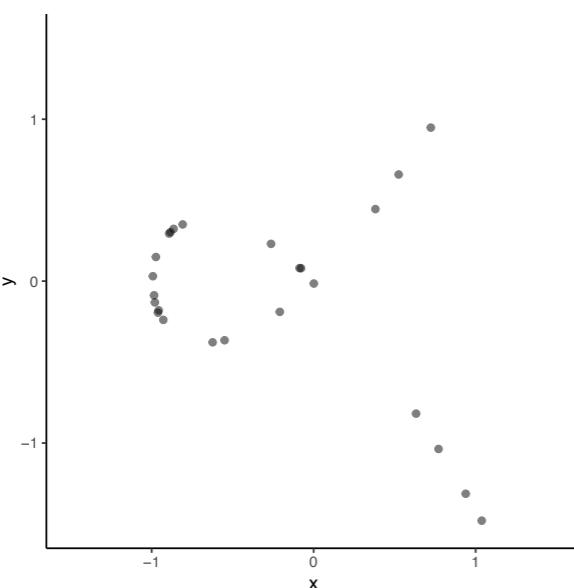
$n = 25$

$n = 50$

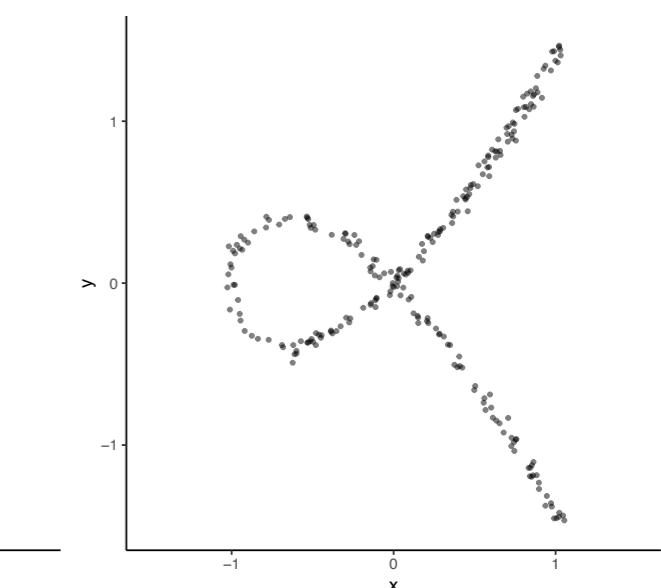
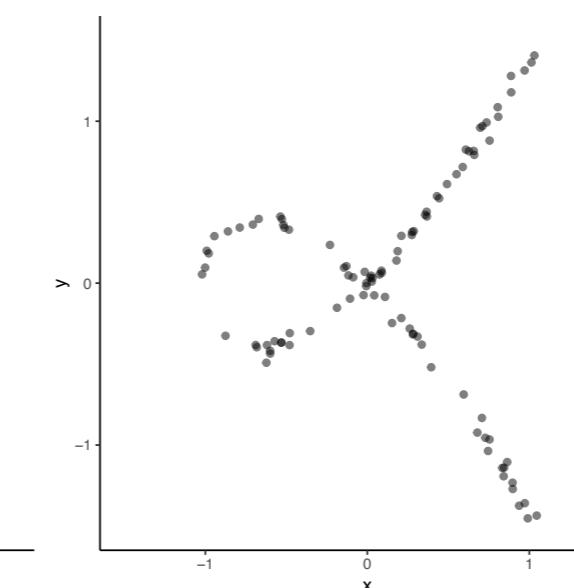
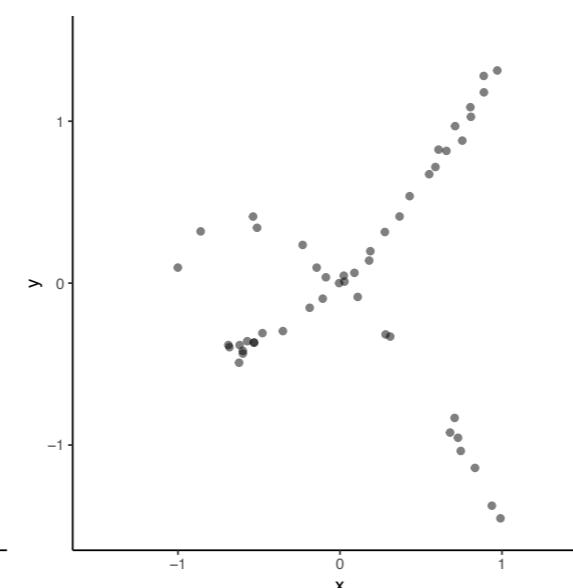
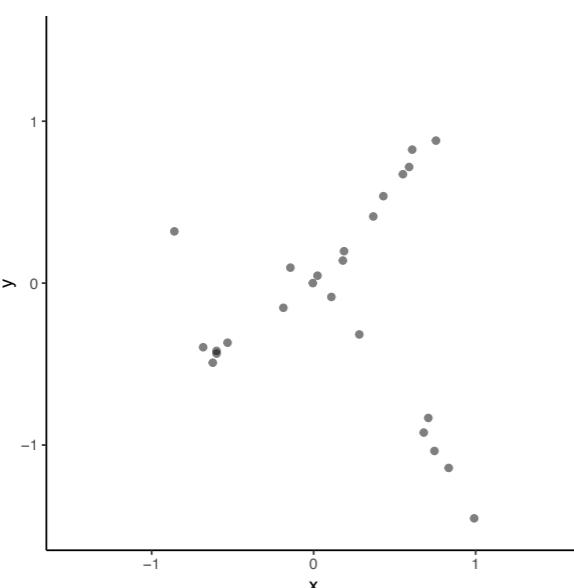
$n = 100$

$n = 250$

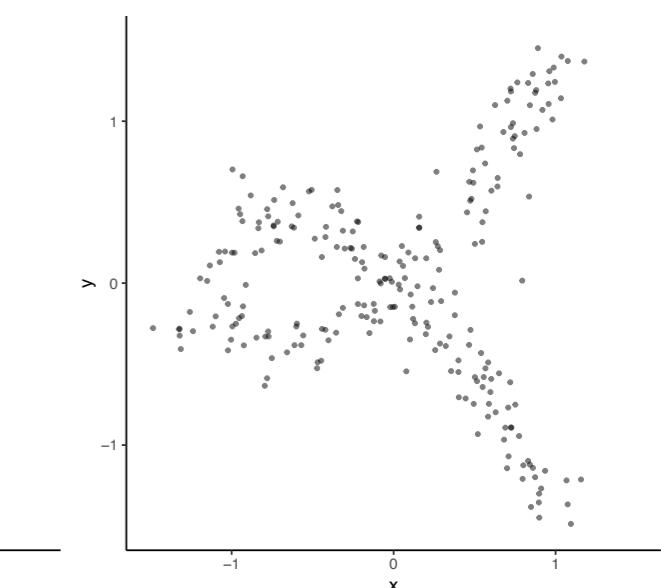
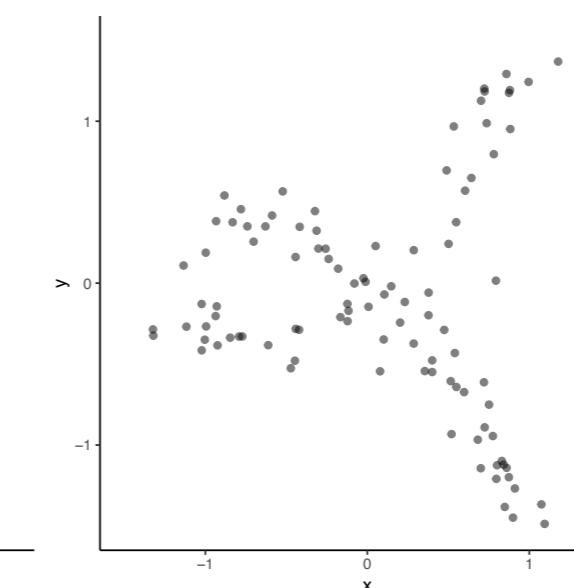
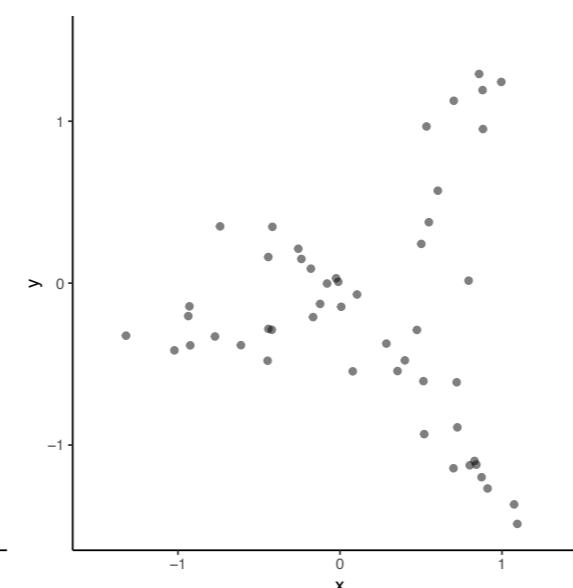
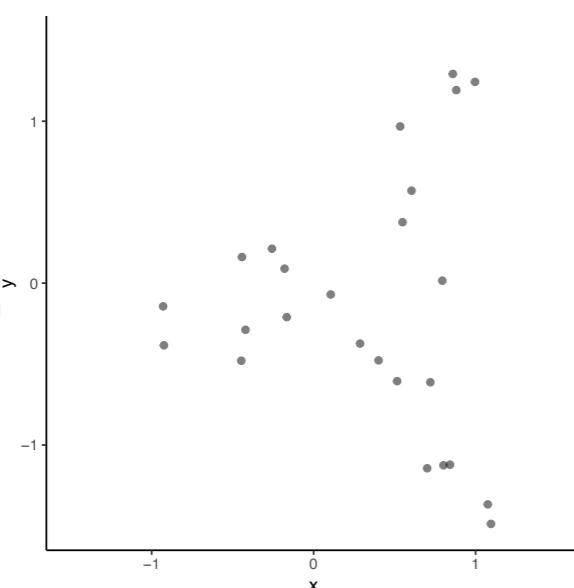
$\sigma = .005$



$\sigma = .025$



$\sigma = .100$



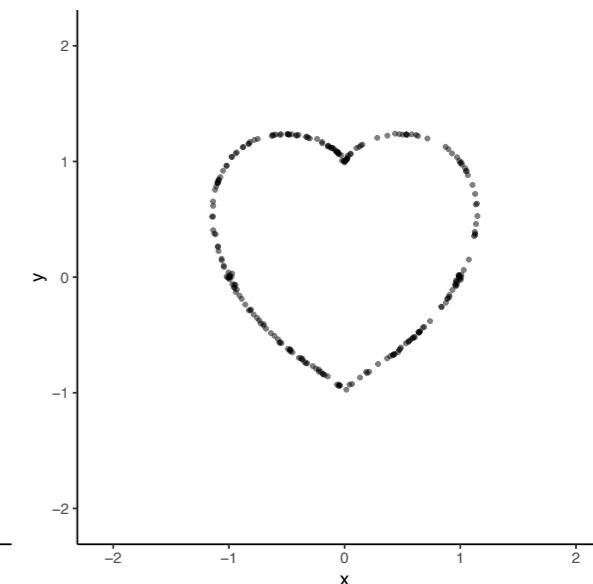
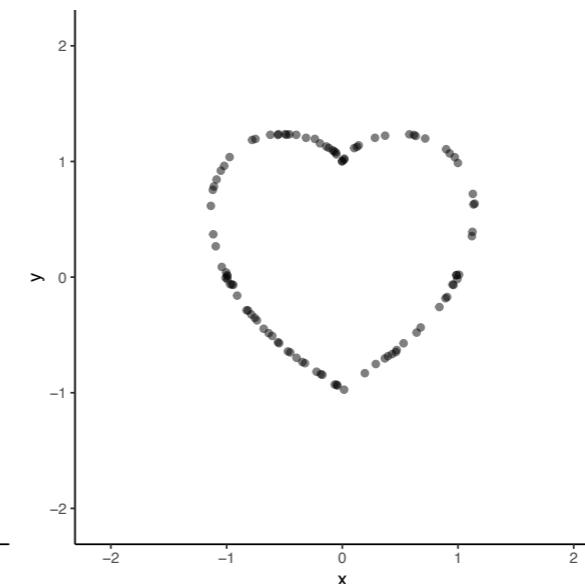
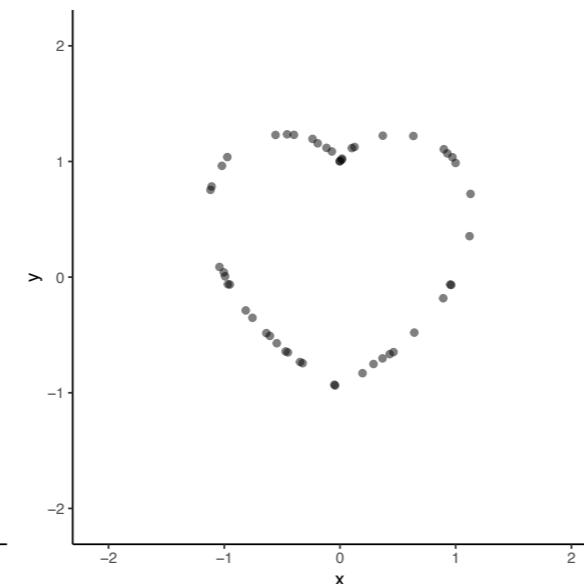
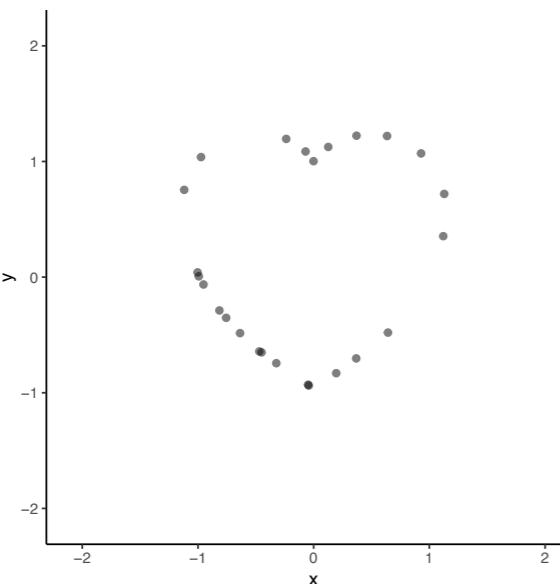
$n = 25$

$n = 50$

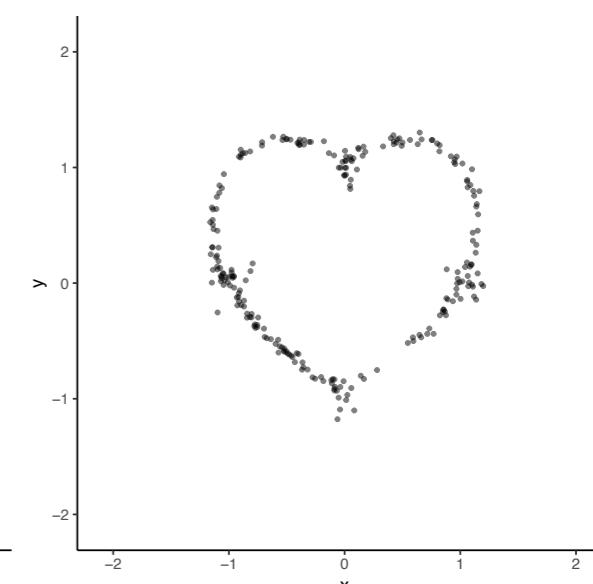
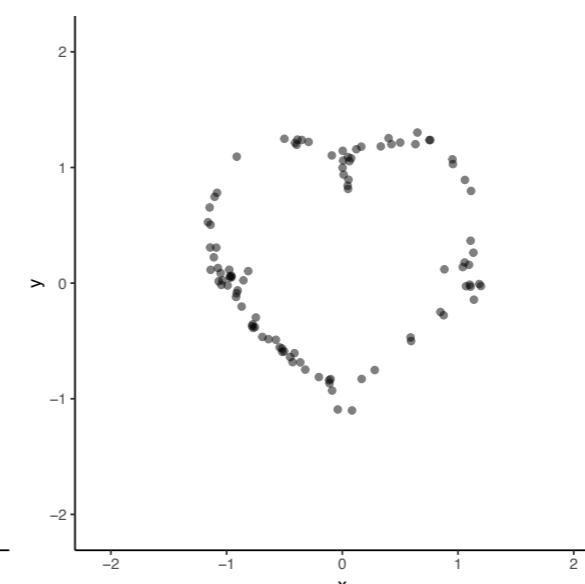
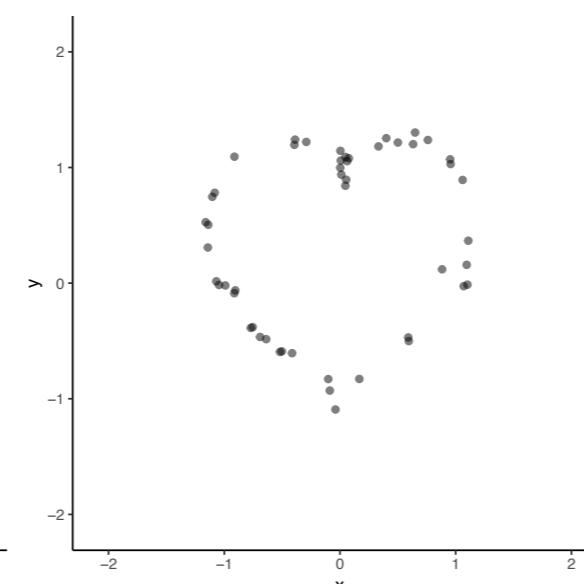
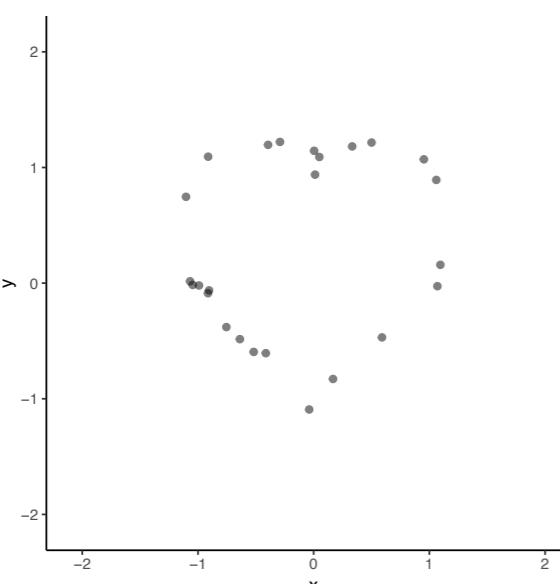
$n = 100$

$n = 250$

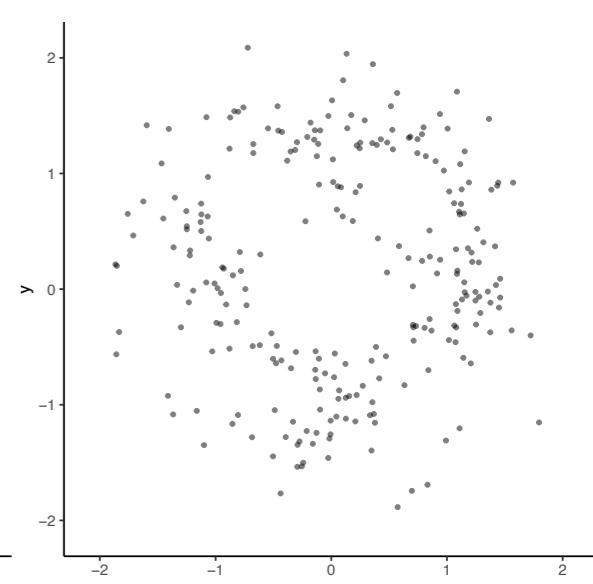
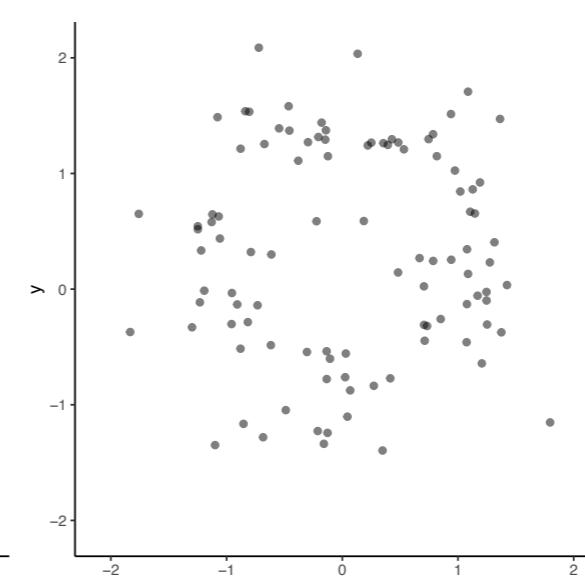
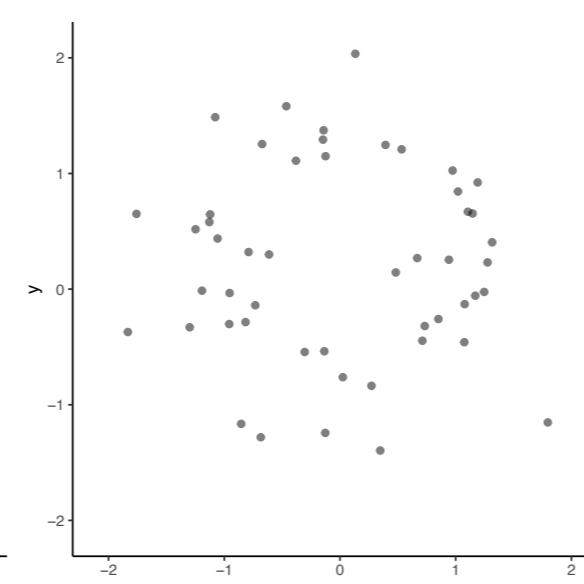
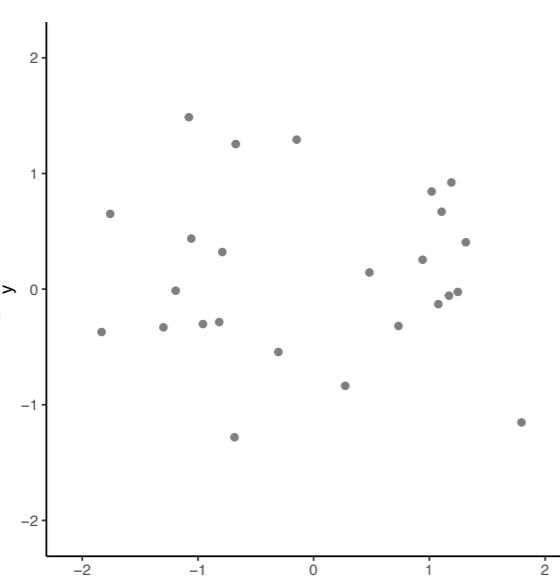
$\sigma = .005$



$\sigma = .025$

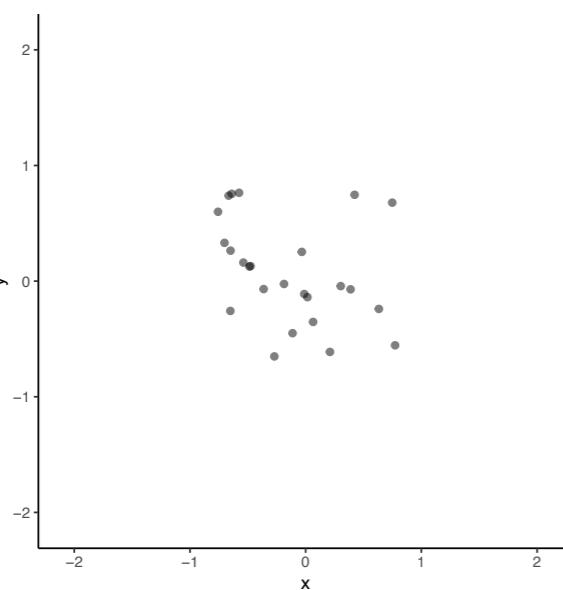


$\sigma = .100$

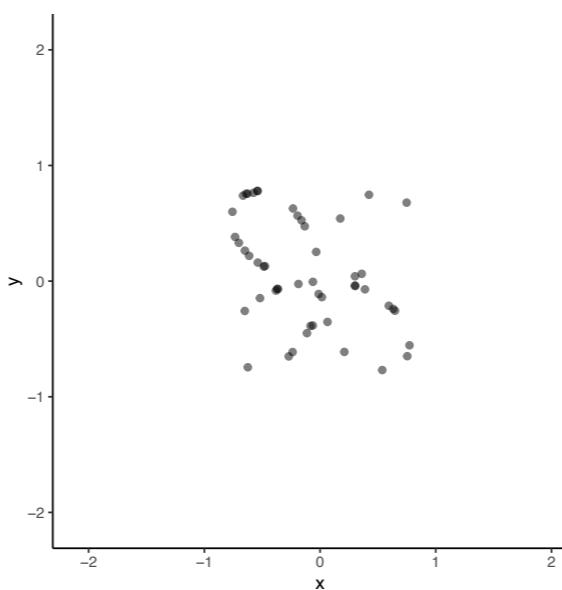


$n = 25$

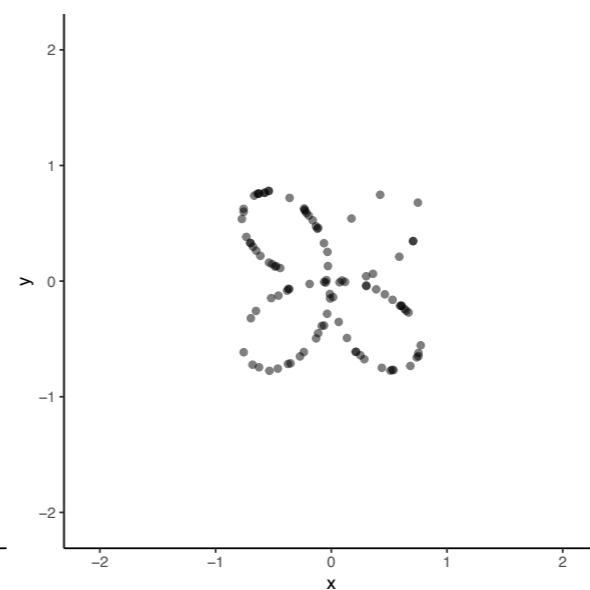
$\sigma = .005$



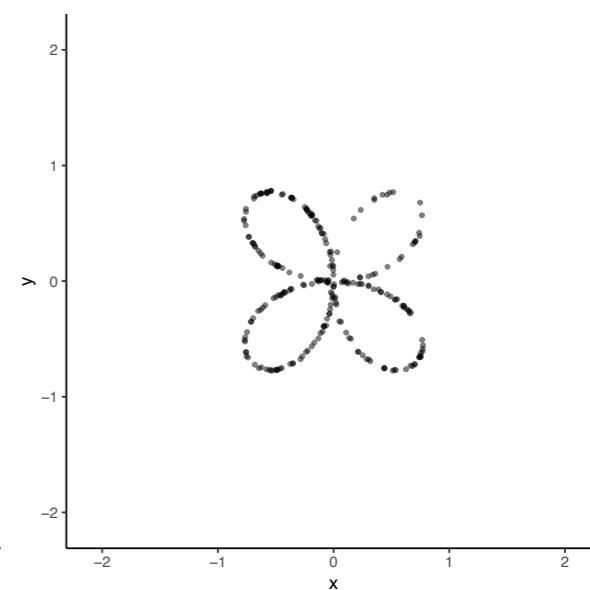
$n = 50$



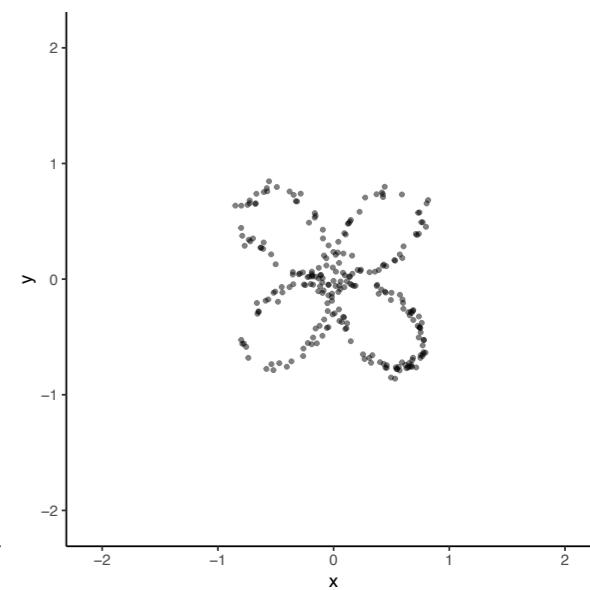
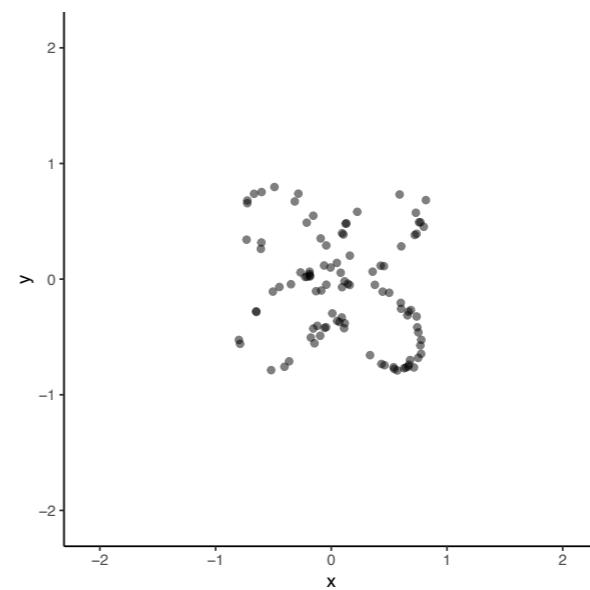
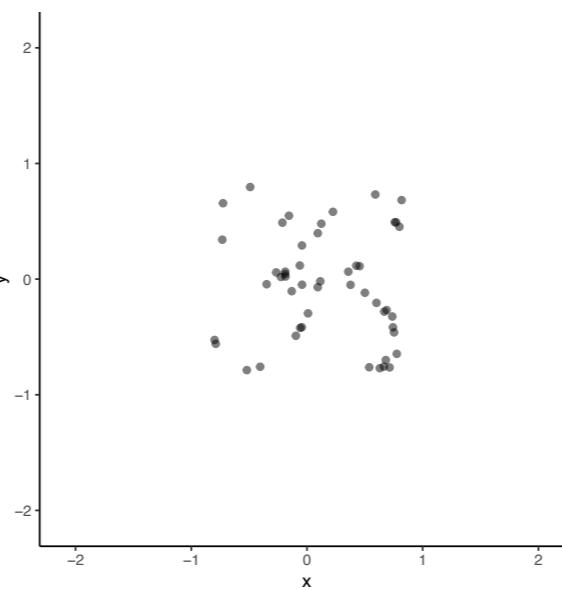
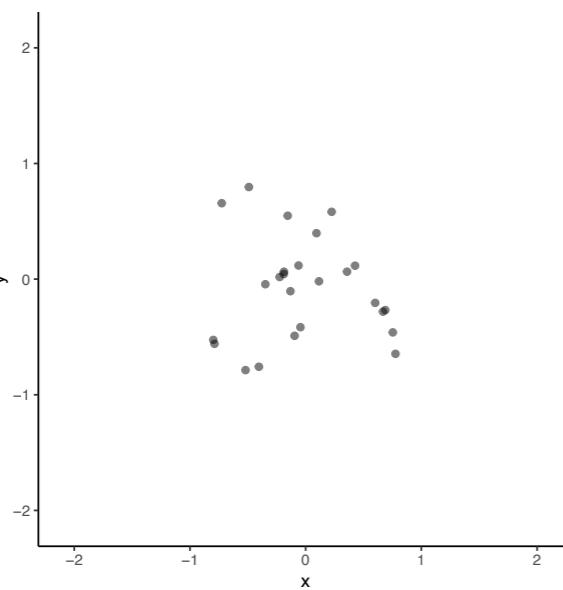
$n = 100$



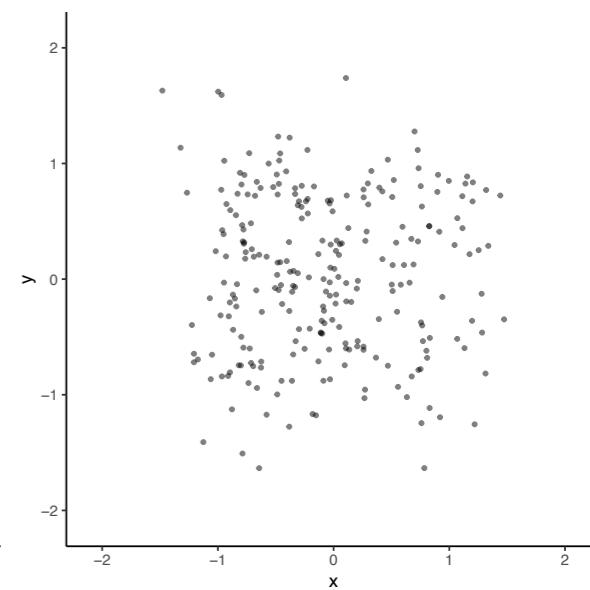
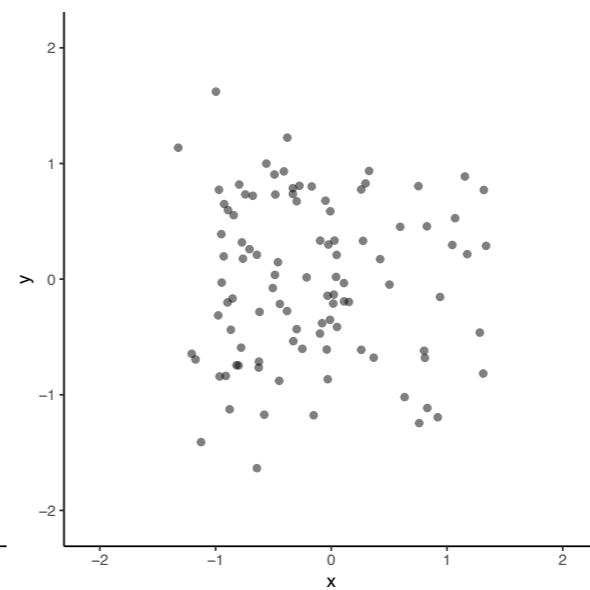
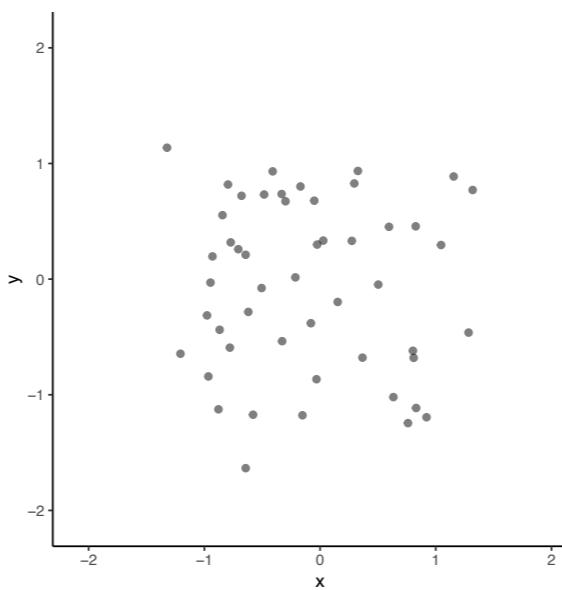
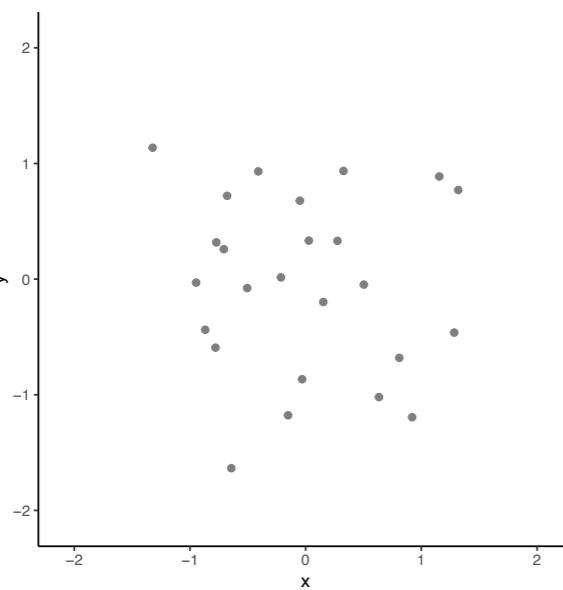
$n = 250$



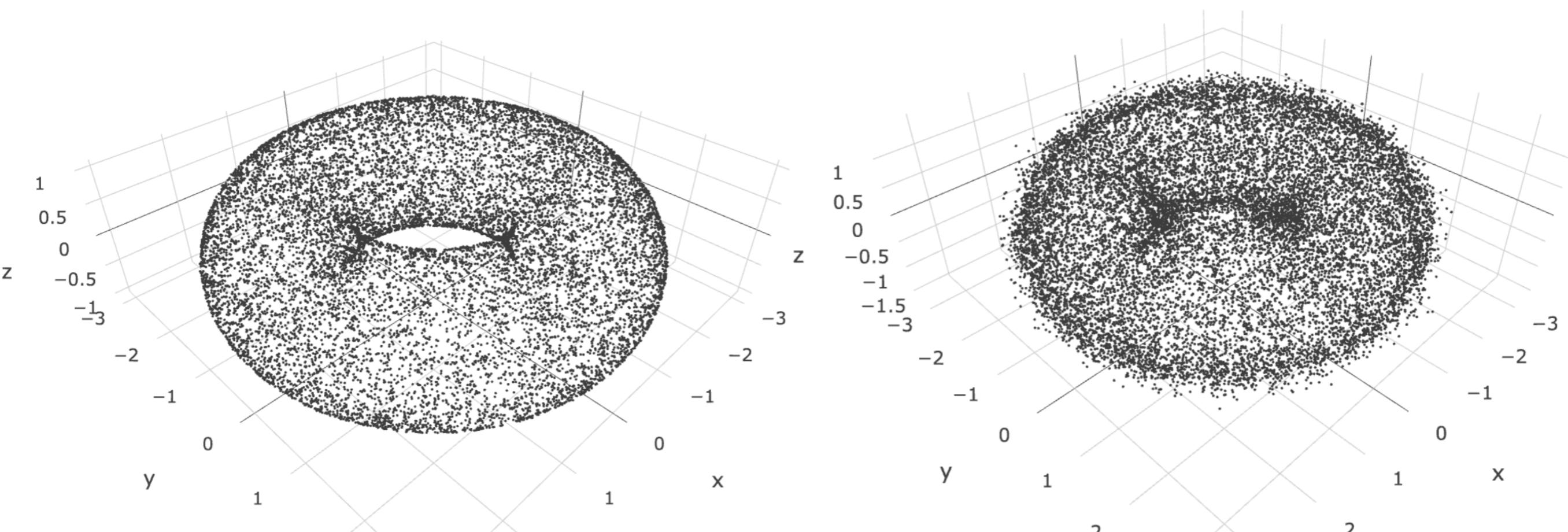
$\sigma = .025$



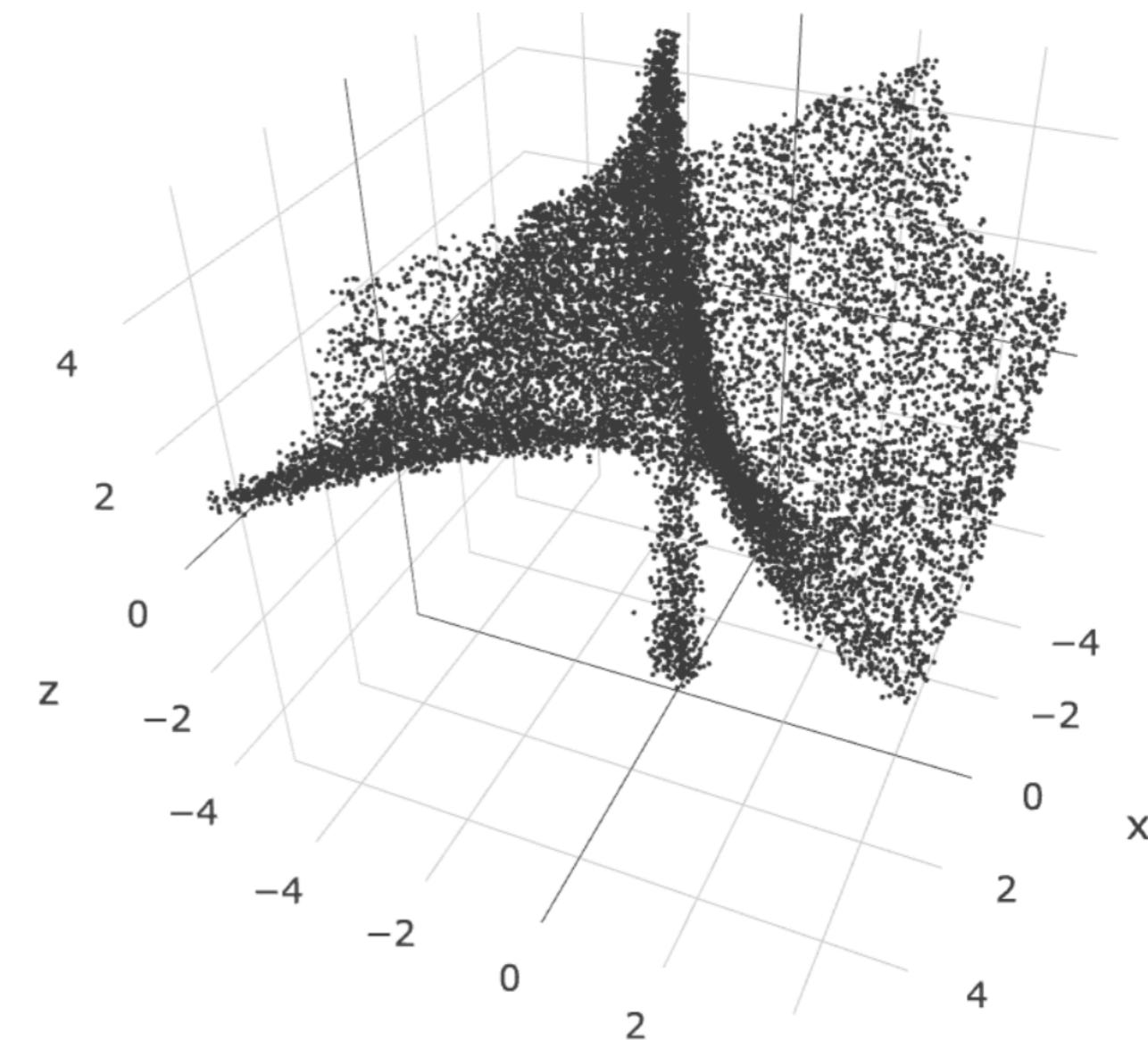
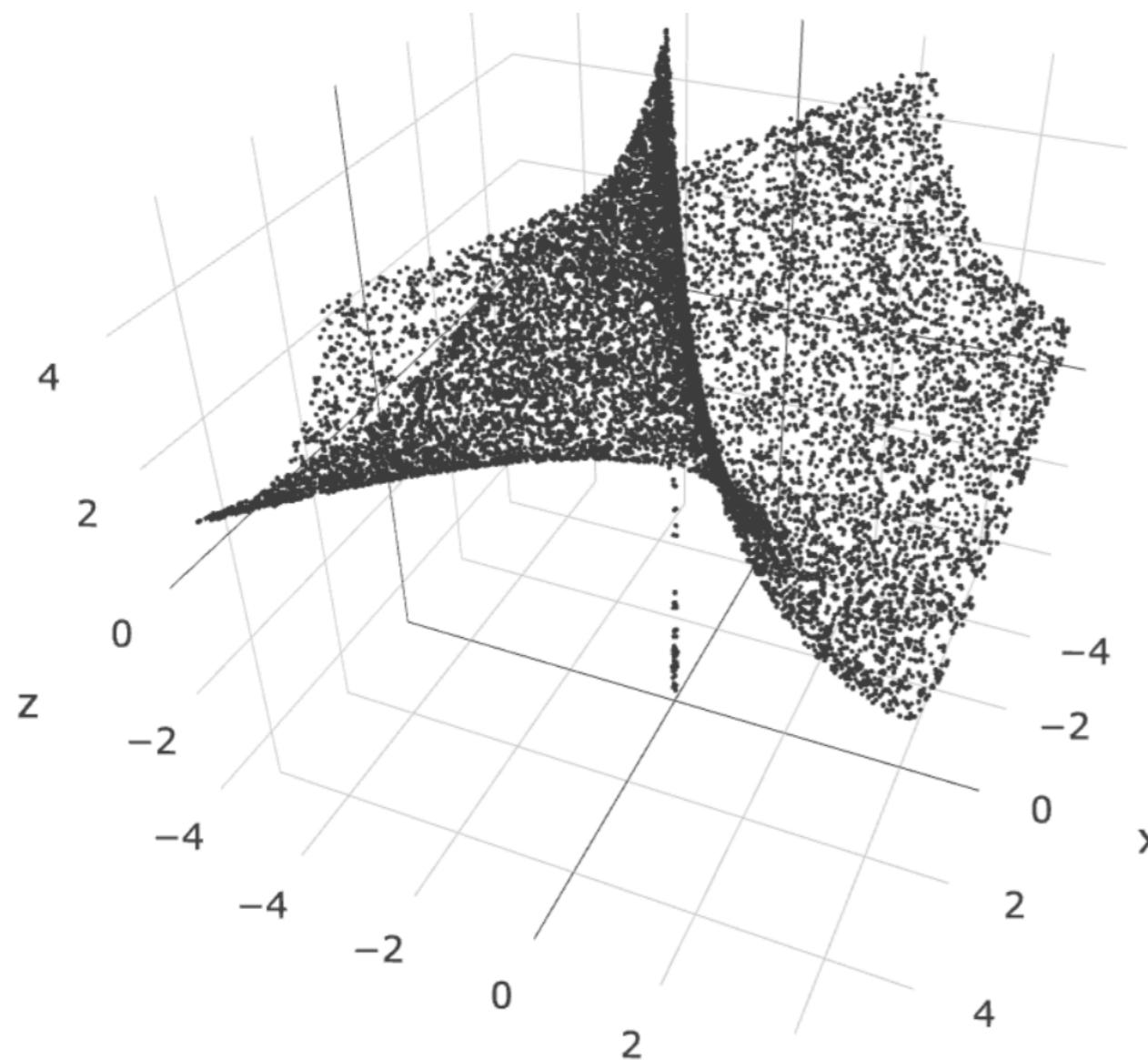
$\sigma = .100$



$\text{VN}(\text{torus}, \sigma = .005/.100); 2000 \text{ points}$

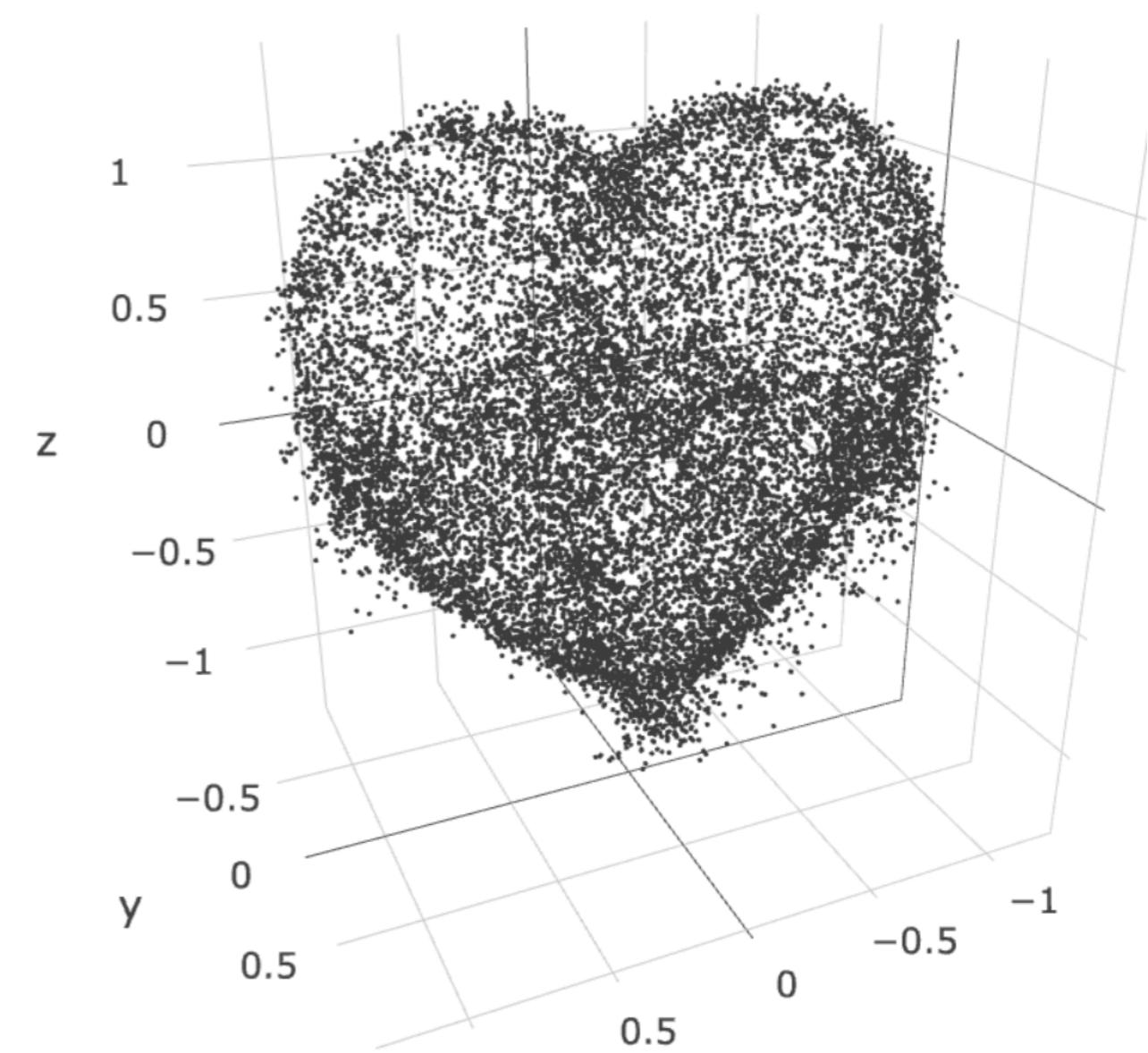
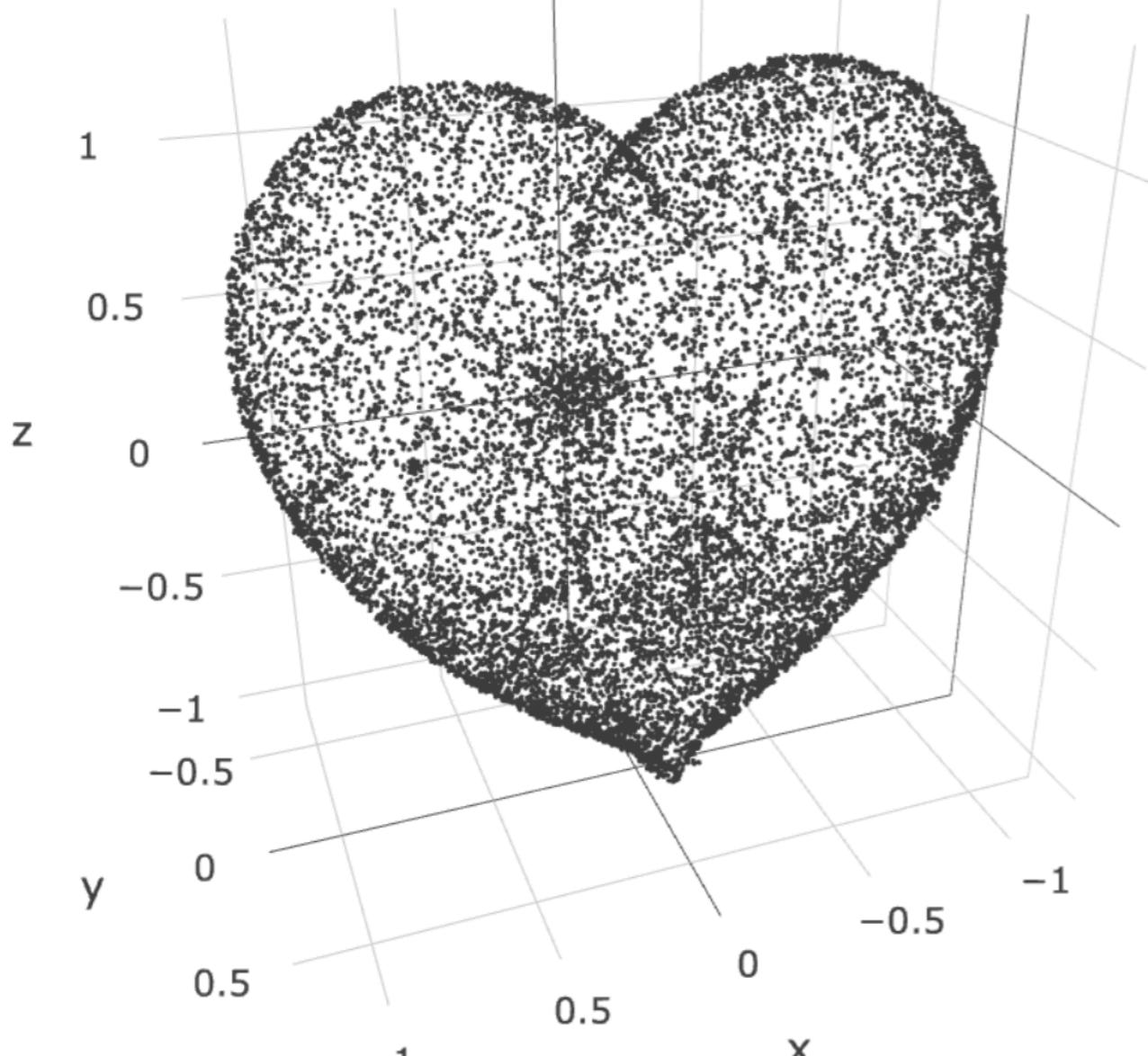


$VN(\text{whitney}, \sigma = .010/.100); 2000 \text{ points}$



Examples

$VN(3d \text{ heart}, \sigma = .005/.025); 2000 \text{ points}$



Other thoughts

For points on variety, we can use endgames from NAG

To move a point \mathbf{x} to a point \mathbf{x}_V on the variety, track

$$\mathbf{H}(\mathbf{x}_V, \boldsymbol{\lambda}; t) = \begin{bmatrix} \mathbf{g}(\mathbf{x}_V) - t\mathbf{g}(\mathbf{x}) \\ \lambda_0(\mathbf{x}_V - \mathbf{x}) + \sum_{i=1}^k \lambda_i \nabla g_i(\mathbf{x}_V) \end{bmatrix} = \mathbf{0}$$

where $\boldsymbol{\lambda} \in \mathbb{P}^m$, from $t = 1$ and $(\mathbf{x}, [1, 0, \dots, 0])$ to $t = 0$

If $\mathbf{J}_{\mathbf{x}}(\mathbf{x})$ is full-rank and \mathbf{x} close enough to V, the path is smooth and ends; terminal point $\mathbf{x}^* \approx \mathbf{x}_V$ can be computed efficiently

Endgames mitigating curvature can also be used

Empirically the strategy seems to work well

Disconnected components are best found by initializing multiple chains with dispersed initial values

Singularities manifest as over-dispersed regions

σ cannot be set too large

Great references:

Betancourt, M. "A Conceptual Introduction to Hamiltonian Monte Carlo." arXiv. (2018)

Neal, R. "MCMC Using Hamiltonian Dynamics" in Handbook of Markov Chain Monte Carlo. Eds. S. Brooks, A. Gelman, G. Jones, X. Meng. (2011)

Thank you!!

www.kahle.io

This material is based upon work supported by the National Science Foundation under Grant Nos. [1622449](#) and [1622369](#).