Stochastic Exploration of Real Varieties

David J. Kahle
Associate Professor

Joint with Jon Hauenstein

icerm

BAYLOR UNIVERSITY 1845
Department of Statistical Science

Overview

1. Motivation

2. Variety distributions

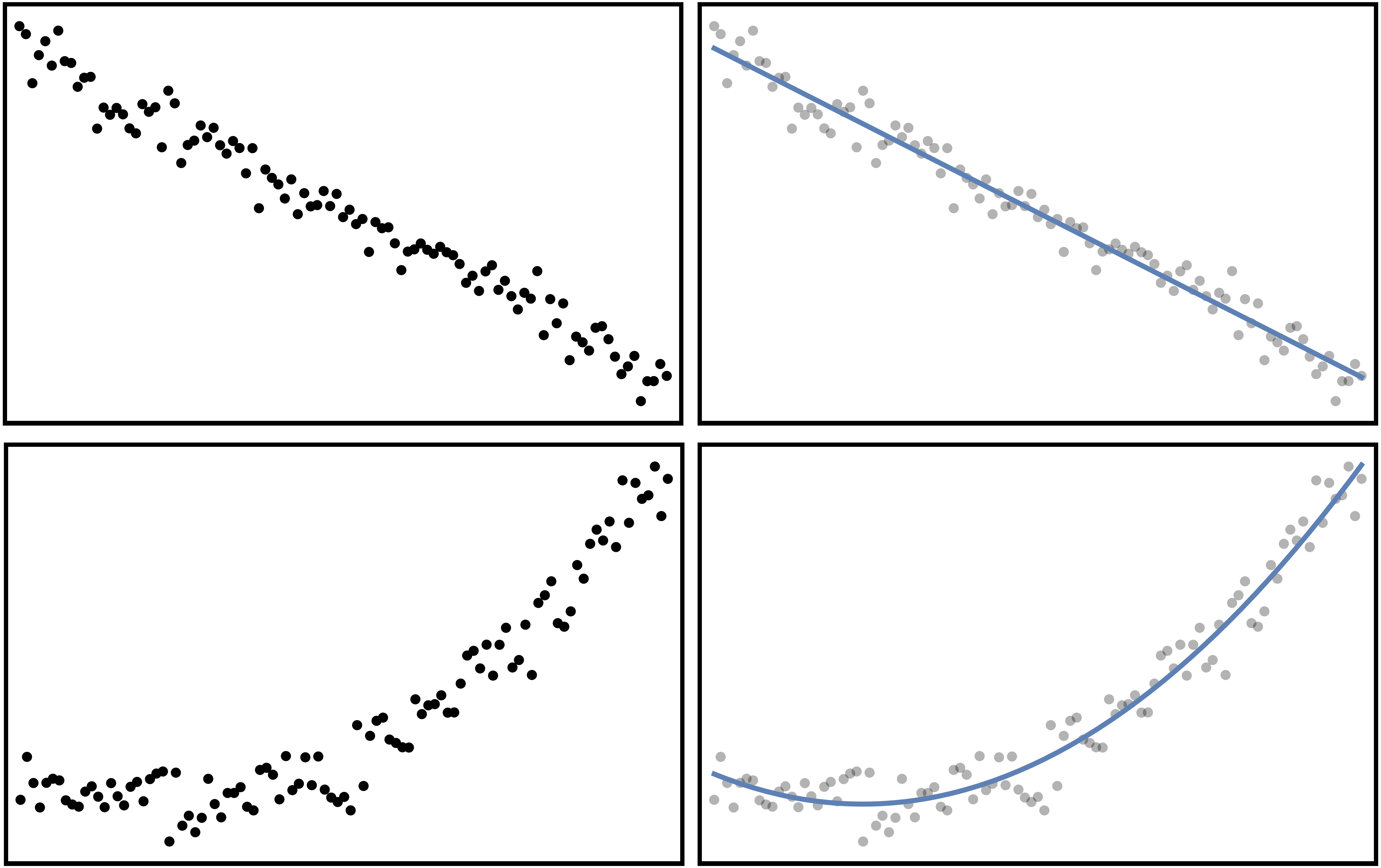
3. Sampling and implementation

4. Examples

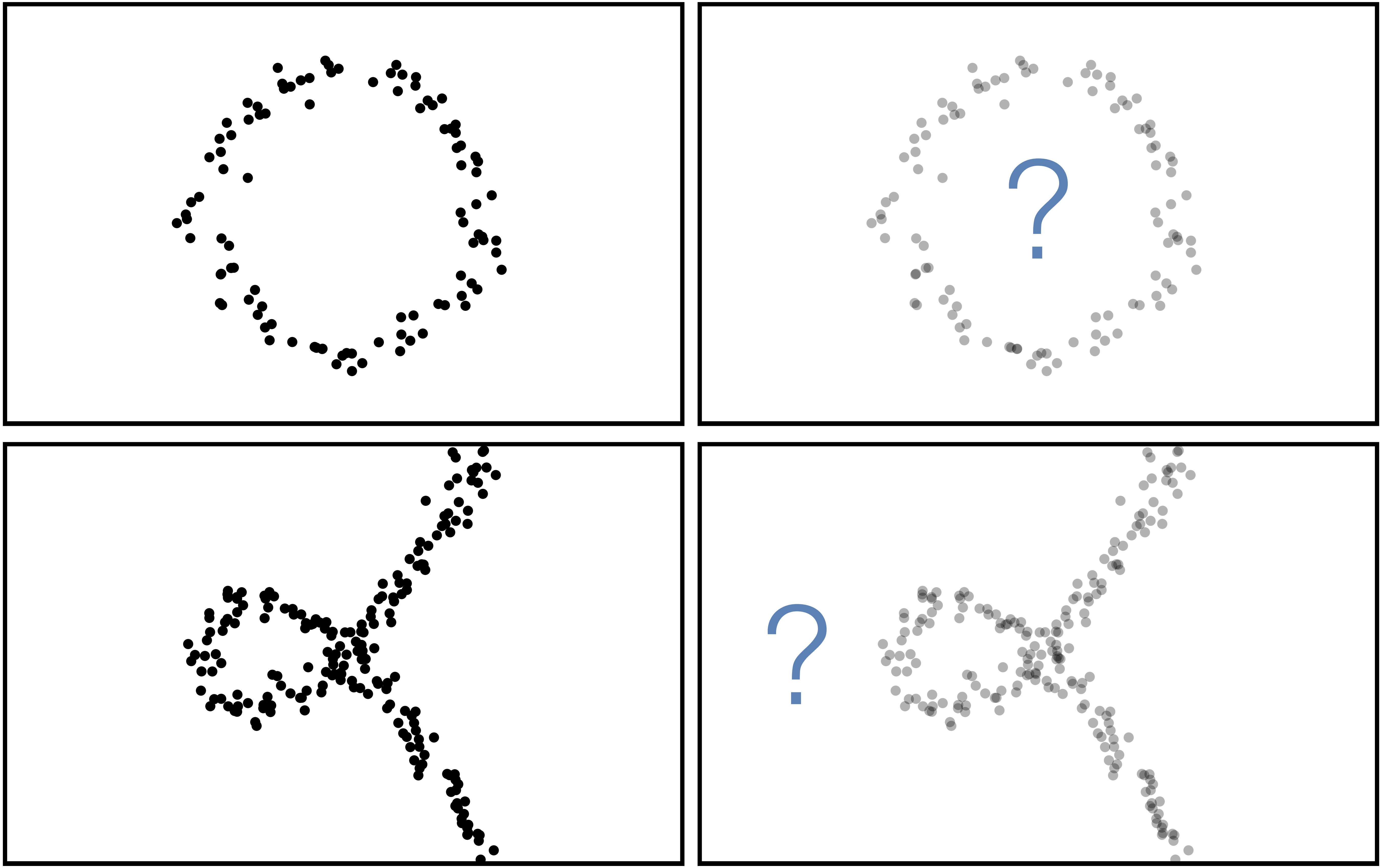
5. Concluding thoughts

Motivation

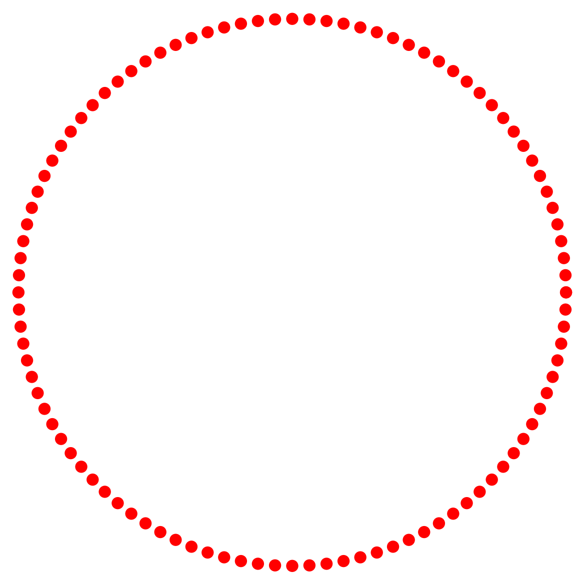
Motivation

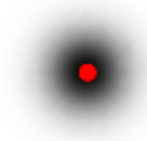


Motivation

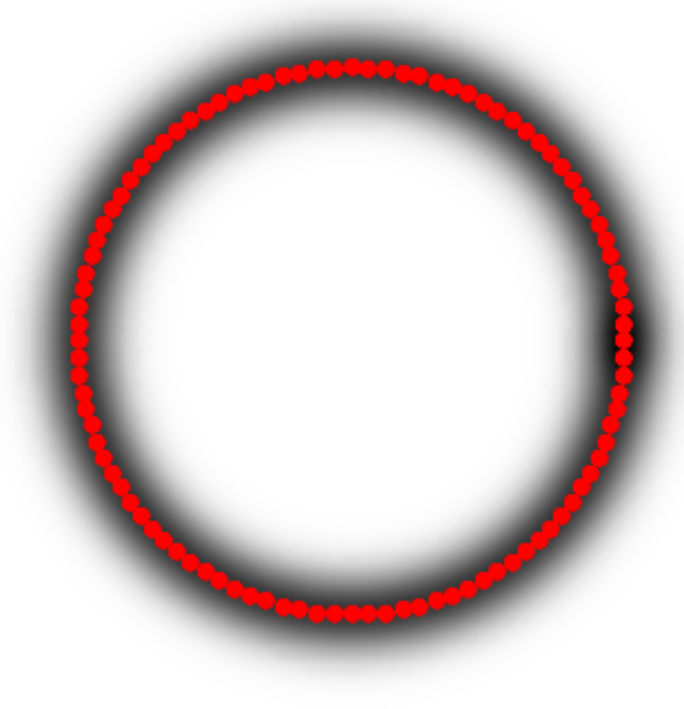


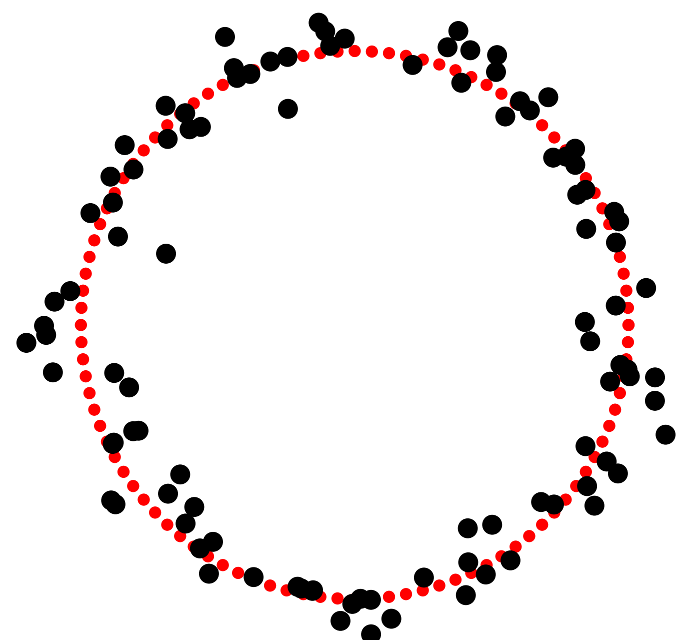
Motivation





Add bivariate normal noise





Careful: not uniform on circle!

Motivation

Problems for pattern recognition:

Very limiting – only can generate points from parametric varieties

No stochastic structure – distribution of estimators? etc.

General problem – how to sample near varieties?

Applications: algebraic pattern recognition (datasets/stochastic framework), TDA, solving nonlinear systems, optimization

Strategy for stochastically exploring real varieties

Create a distribution with mass near the variety of interest

Sample from the distribution

Magnetize the sampled points onto the variety with endgames

Variety distributions

Recasting the normal distribution

The normal density is

Partition function, normalizing constant dependent on parameters

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

μ is the mean; the center of the bell curve

σ is the standard deviation; governs dispersion about μ

Empirical rule –

68% of distribution within ±σ of μ

95% of distribution within ±2σ of μ

99.7% of distribution within ±3σ of μ

Recasting the normal distribution

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 $\boldsymbol{X}$ has the variety normal distribution if

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

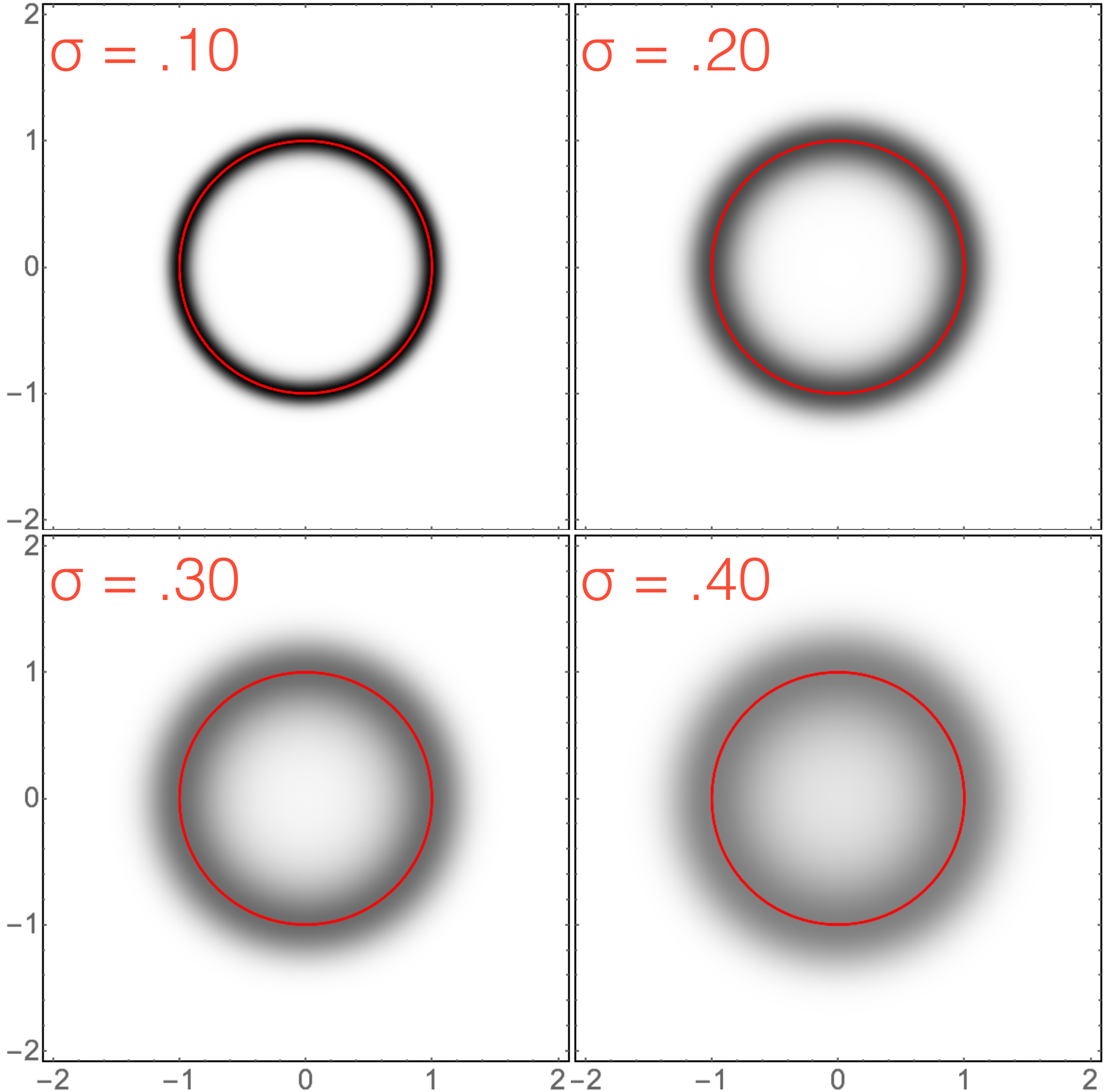
with $g(\boldsymbol{x}|\boldsymbol{\beta}) \in \mathbb{R}[\boldsymbol{x}]$

g is "given" in the sense that the vector $\boldsymbol{\beta}$ is known and the polynomial form is specified

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

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

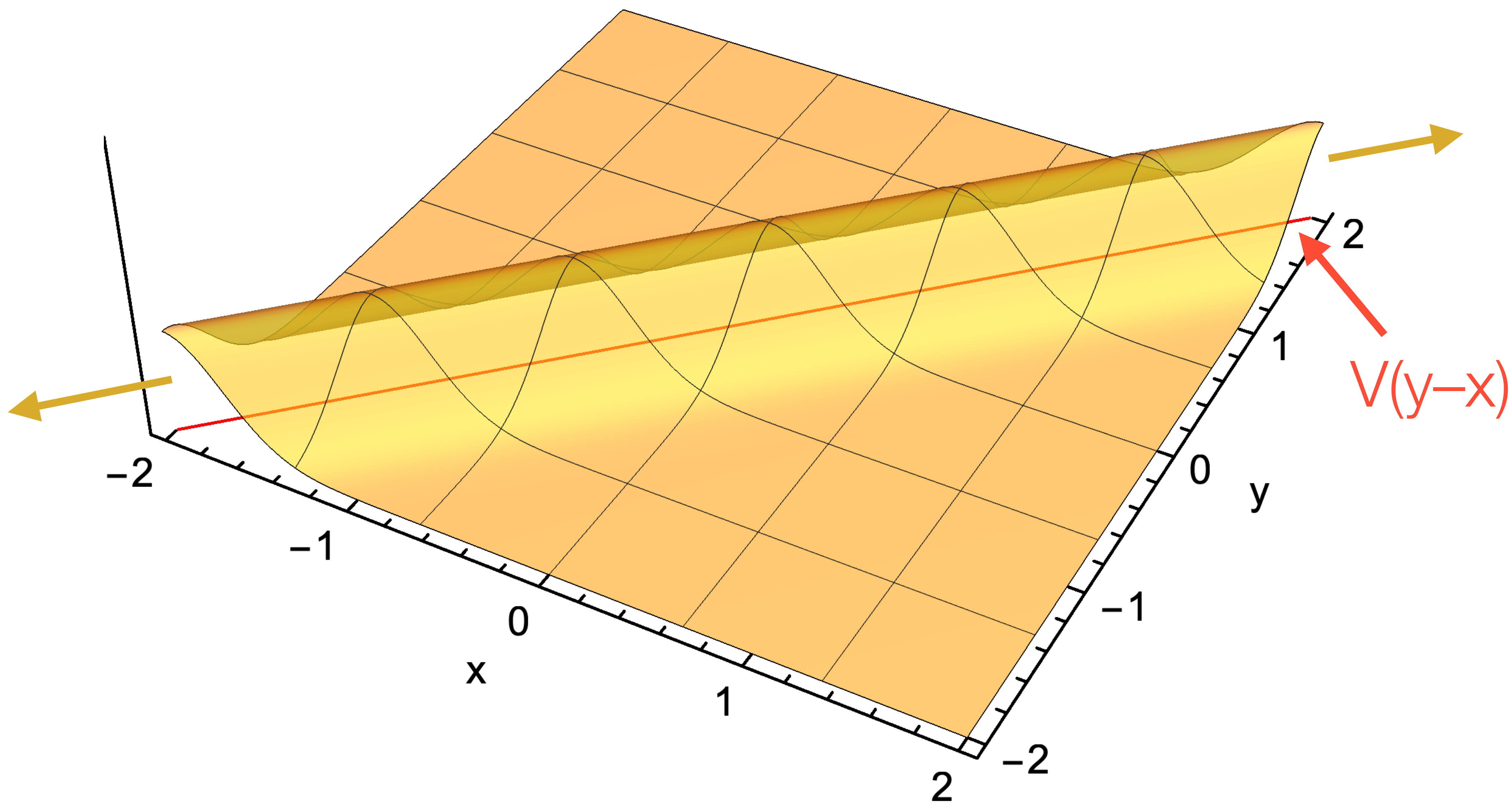
Variety normal distribution – provisional



1. Non-compact varieties

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

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



Variety normal* distribution – problems

1. Non-compact varieties

Solution: Truncate or taper

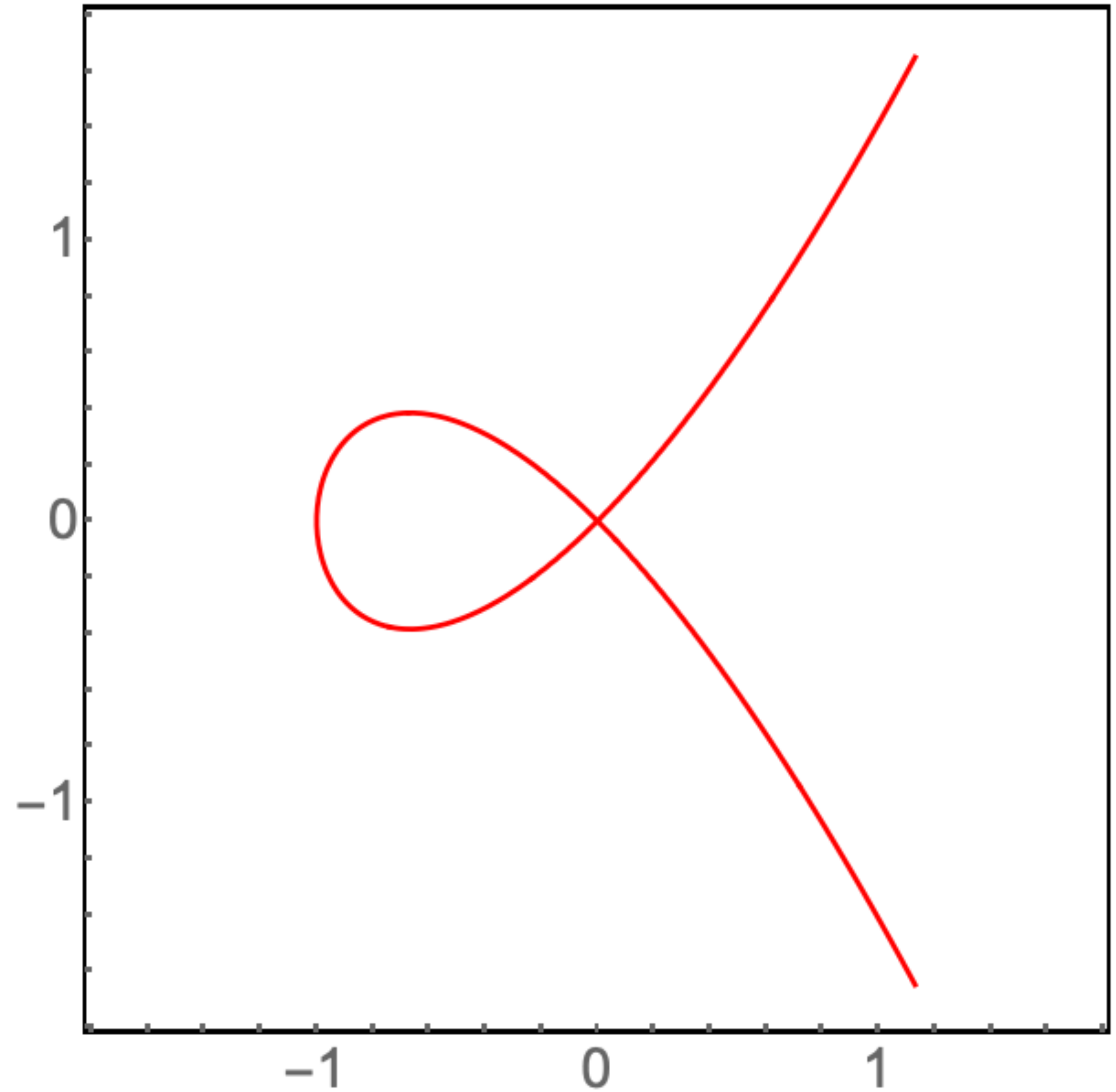
2. σ does not gauge variability globally

Variety normal* distribution – problems

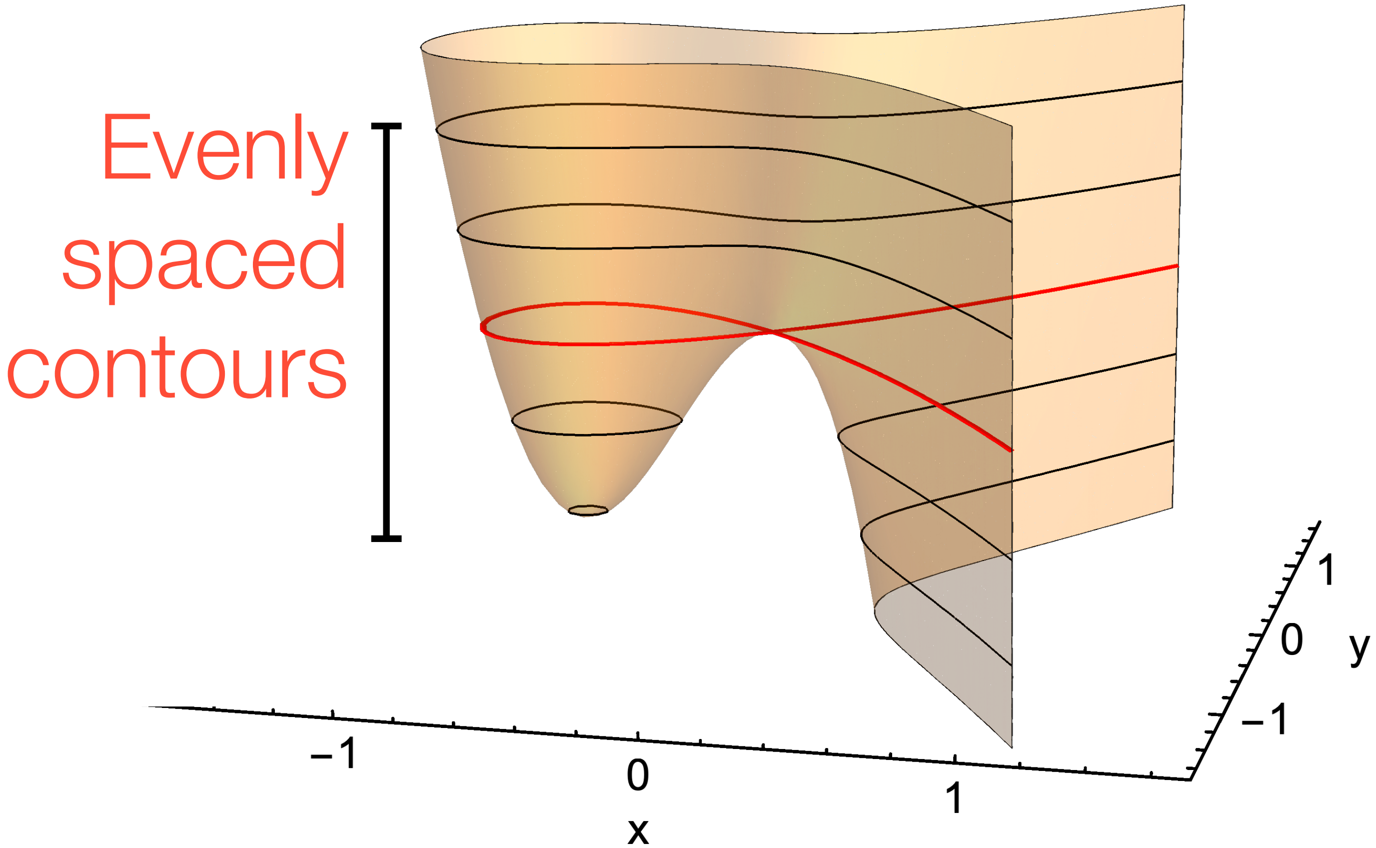
2. σ does not gauge variability globally

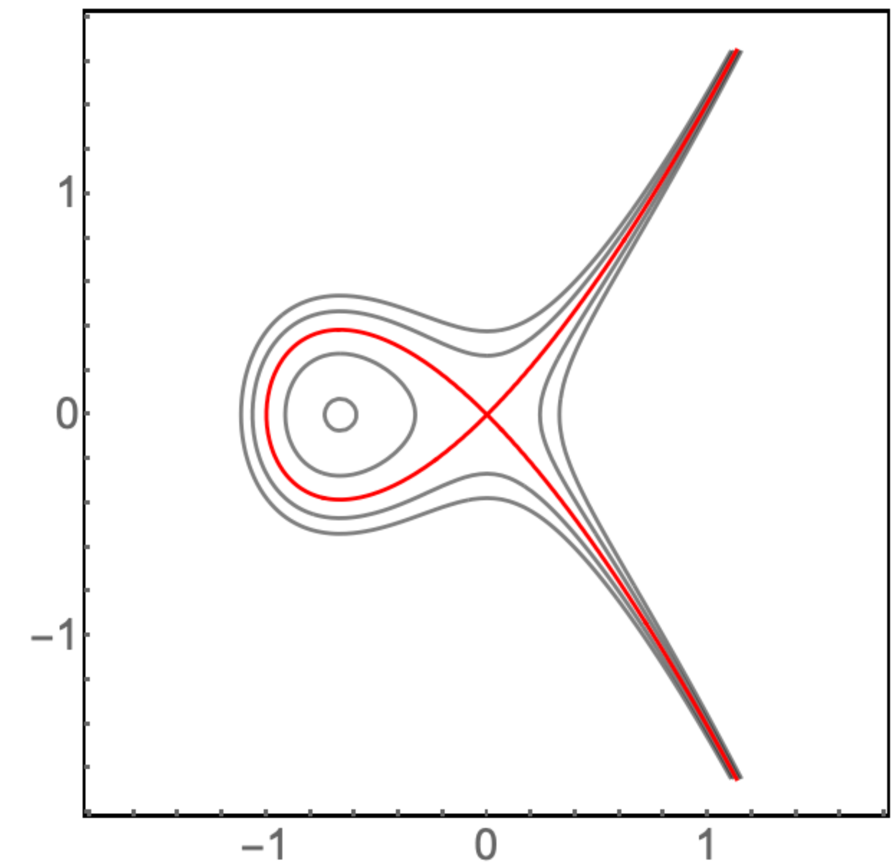
Probability mass does not decay evenly across variety

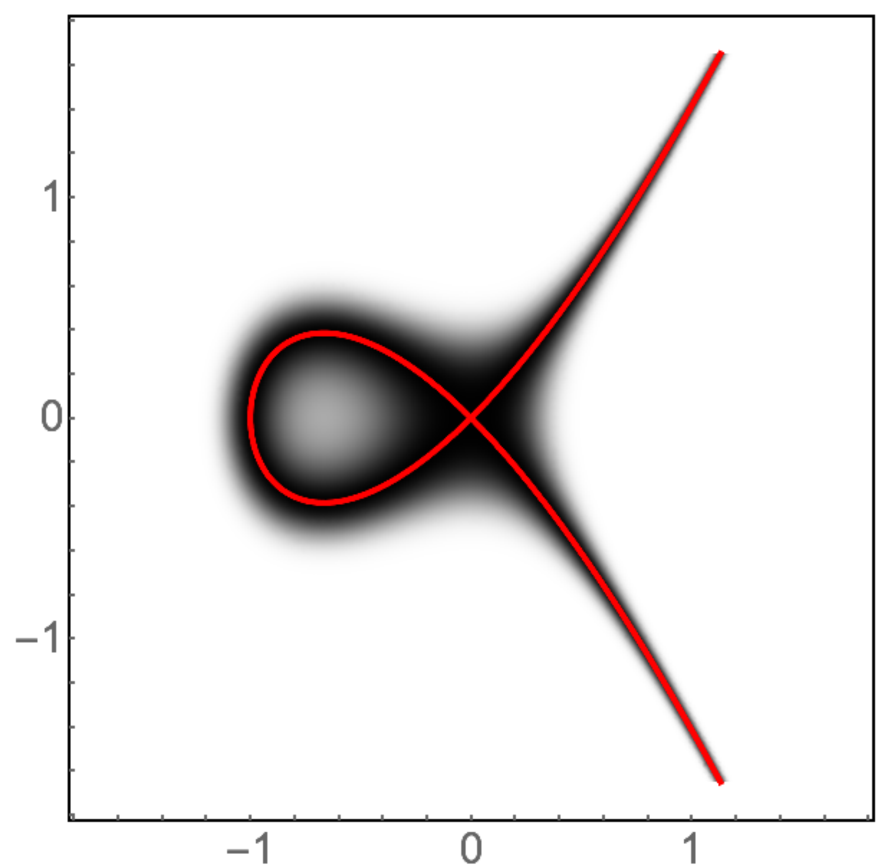
Example: Alpha curve, $V(y^2 - (x^3 + x^2))$

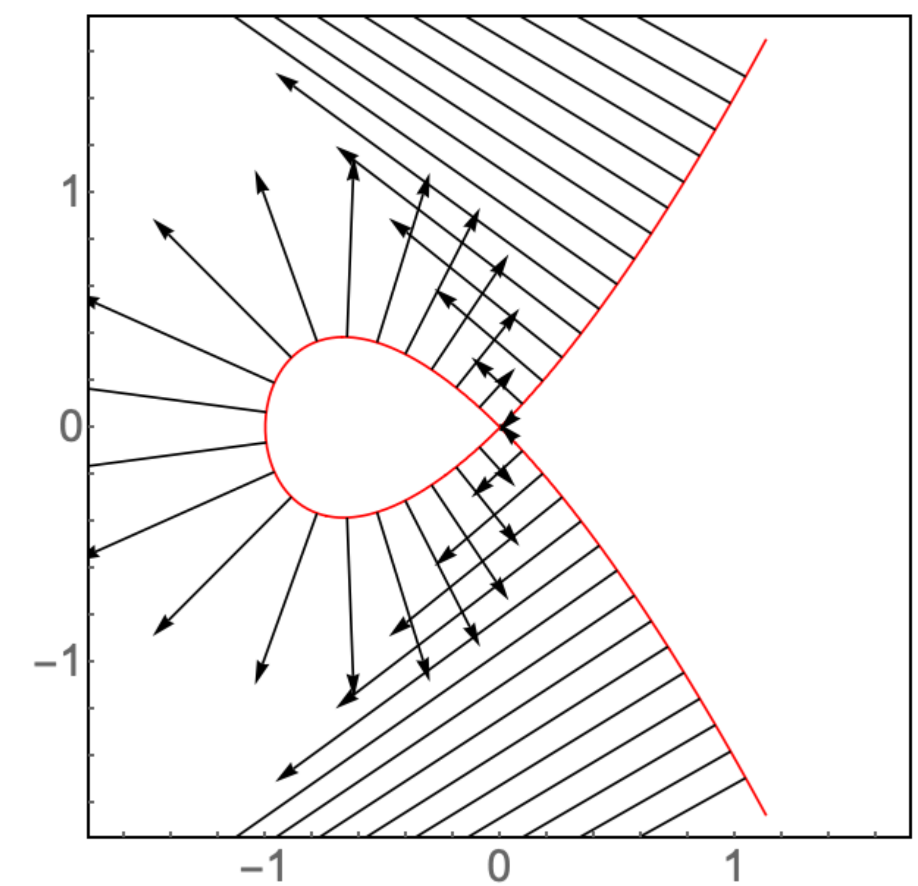


Variety normal* distribution – problems









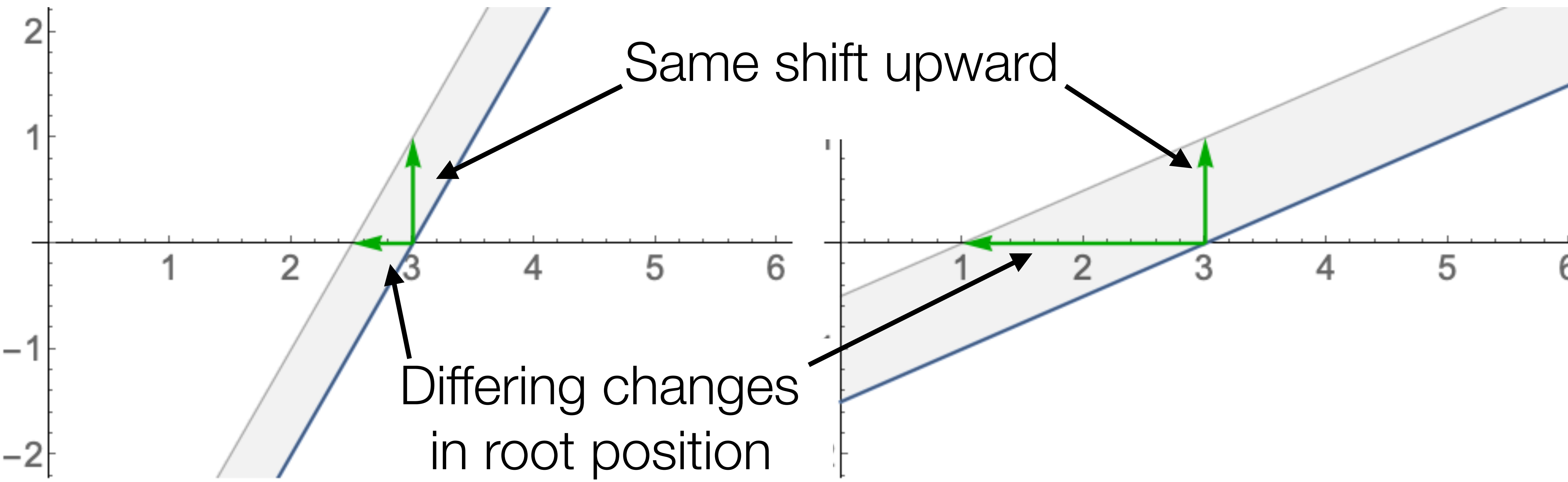
Variety normal* distribution – problems

2. σ does not gauge variability globally

Probability mass does not decay evenly across variety

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

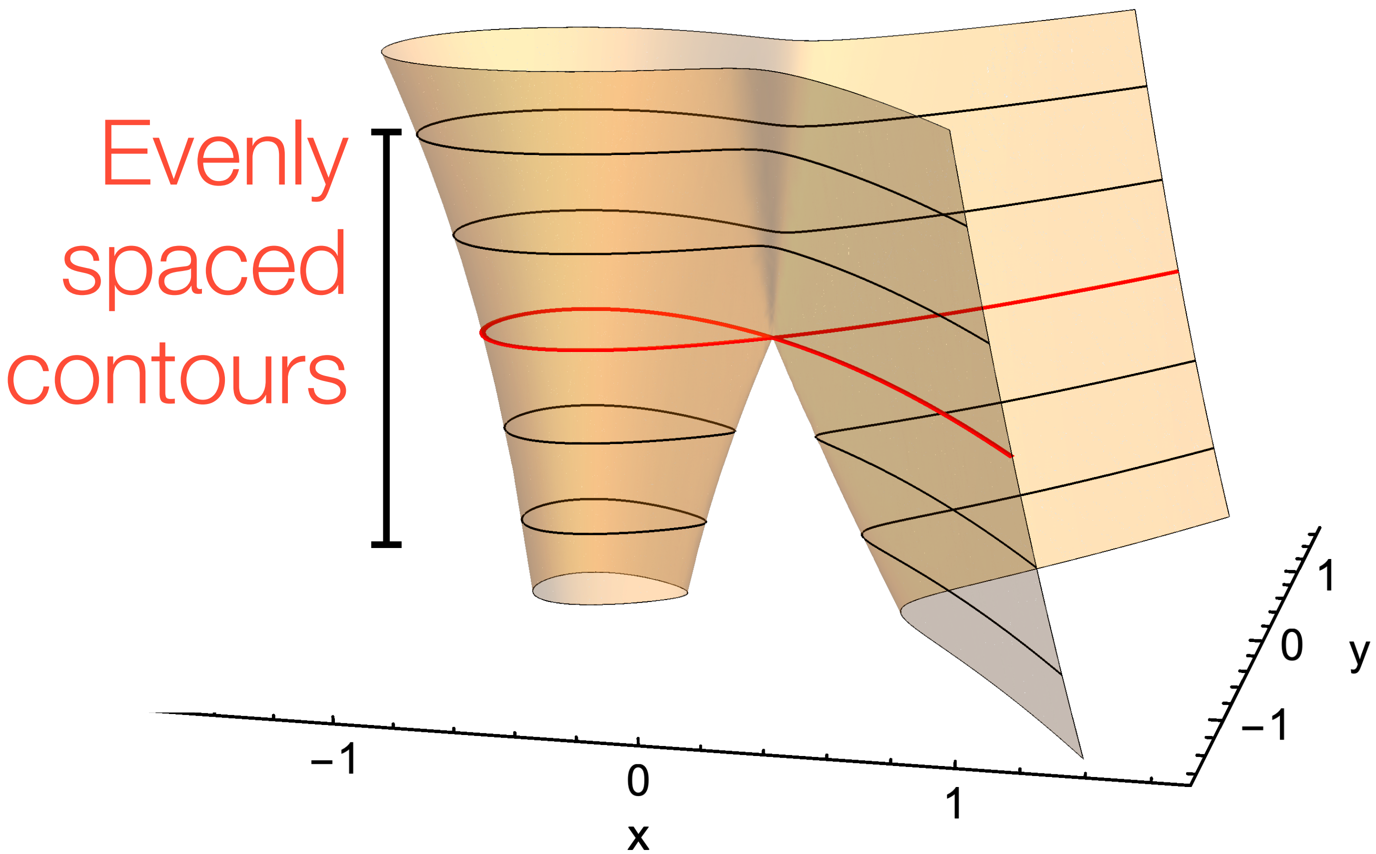
Cause: differing gradient sizes ⇒ differing change in variety

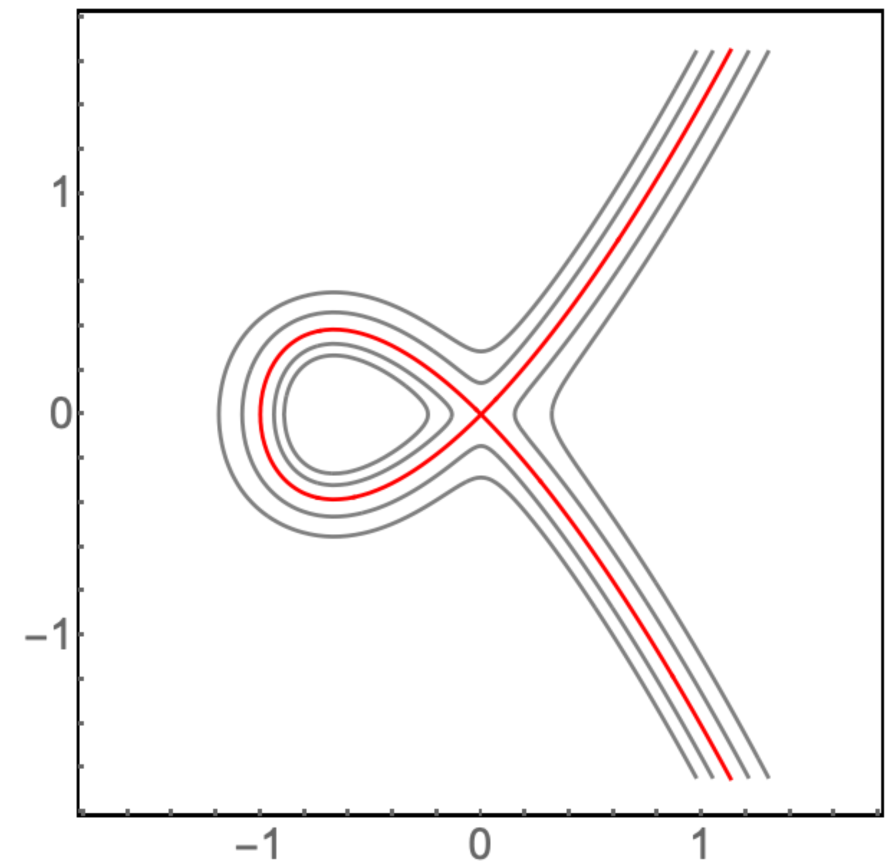


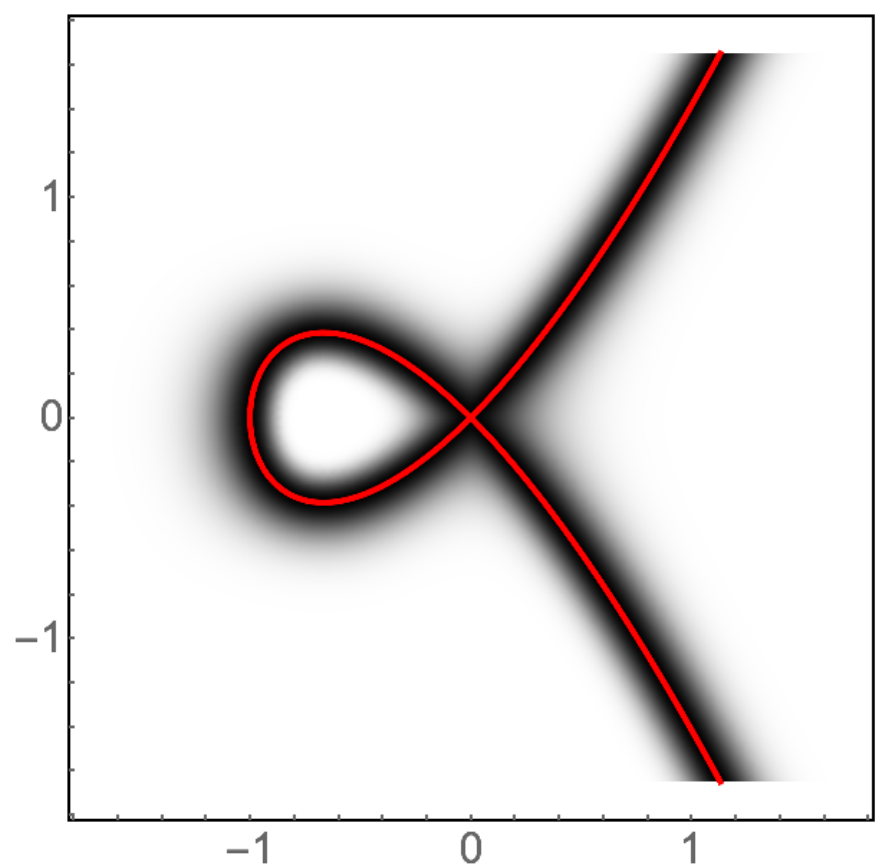
Solution: normalize g by the size of its gradient

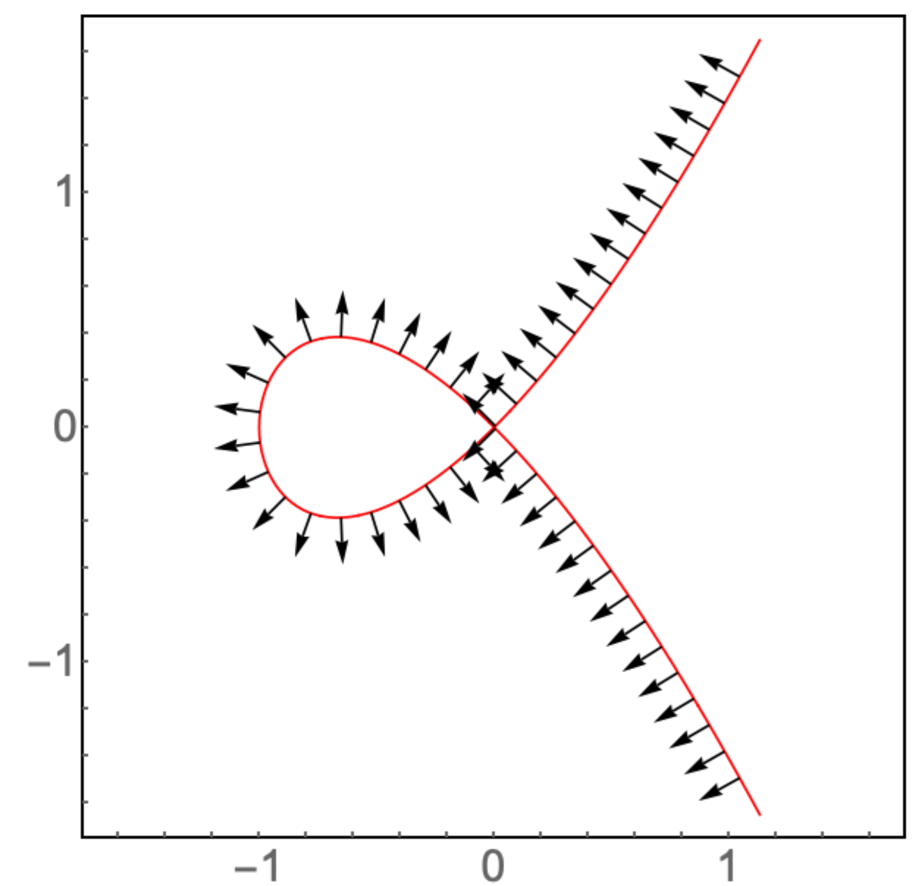
(That doesn't change the zero locus.)

Variety normal* distribution – problems



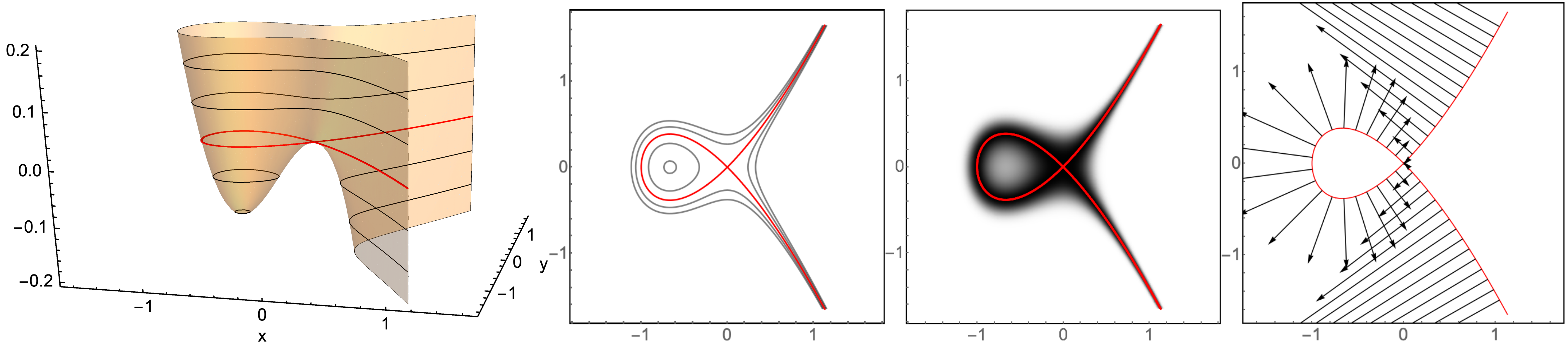




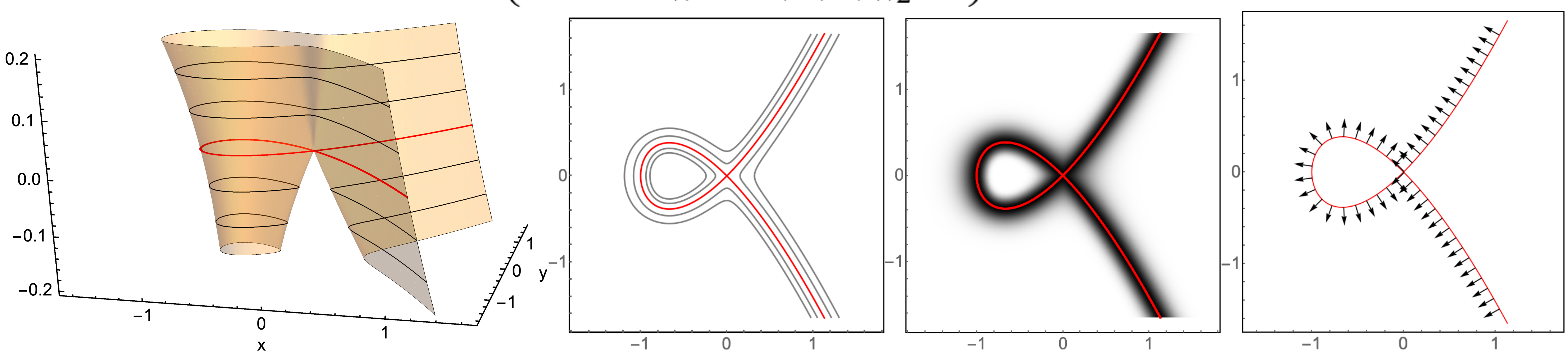


Variety normal* distribution – problems

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



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



Variety normal* distribution – problems

1. Non-compact varieties

Solution: Truncate or taper

2. σ does not gauge variability globally

Solution: Normalize by gradient

3. Awkward parameter space B

Non-trivial choices of β's can make the variety empty or full

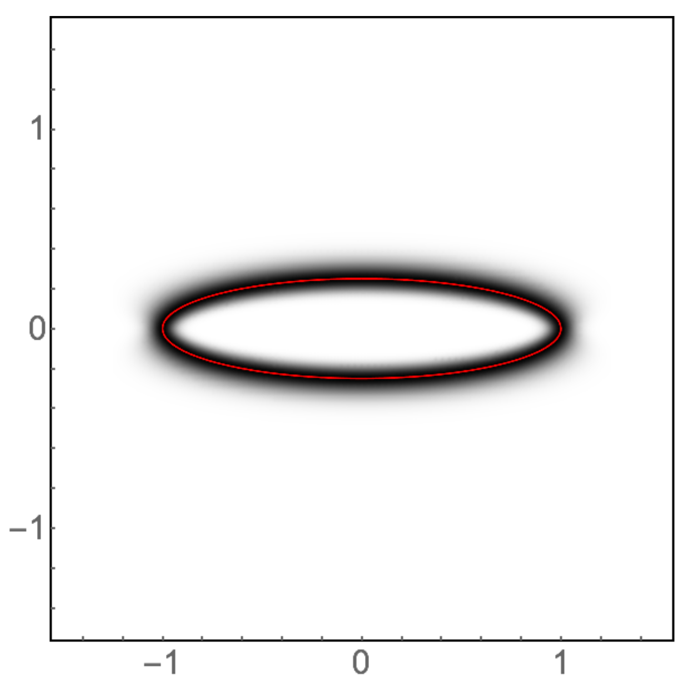
B is not explicit: parameters don't range over a convenient open subset of R^b

Variety normal (VN) distribution

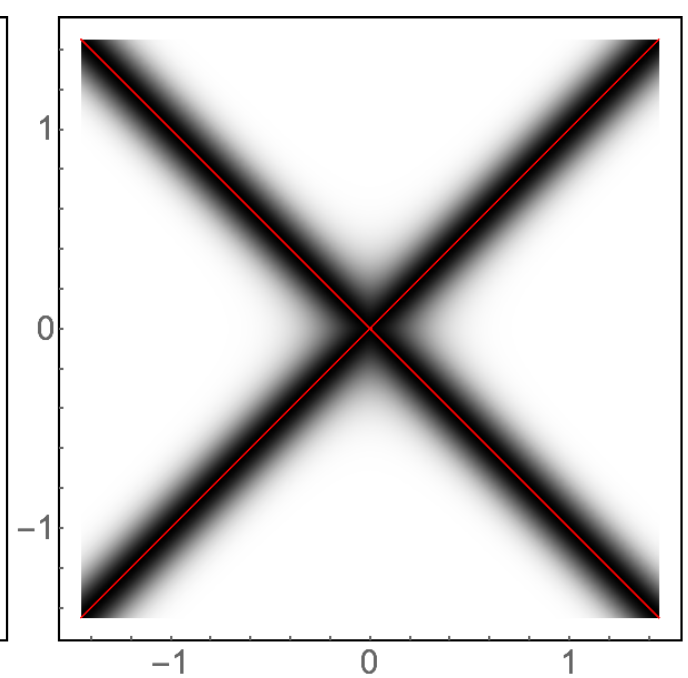
A random vector $\boldsymbol{X}$ has the variety normal distribution if

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

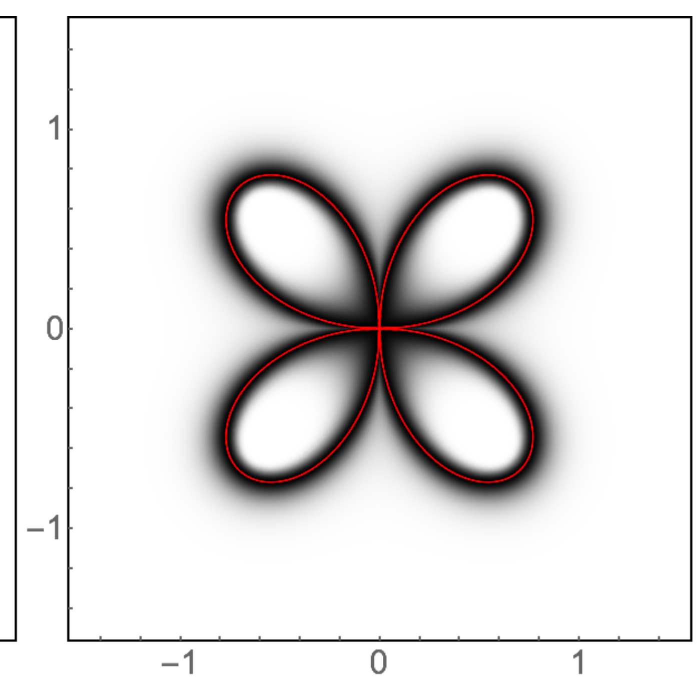
with $g(\boldsymbol{x}|\boldsymbol{\beta}) \in \mathbb{R}[\boldsymbol{x}]$



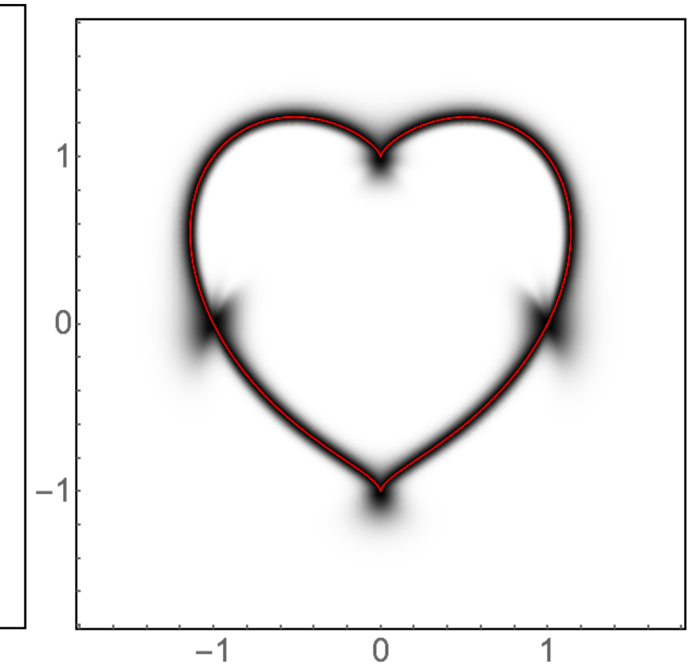
$x^2 + (4y)^2 - 1$



$(y - x)(y + x)$



$(x^2+y^2)^3 - 4x^2y^2$



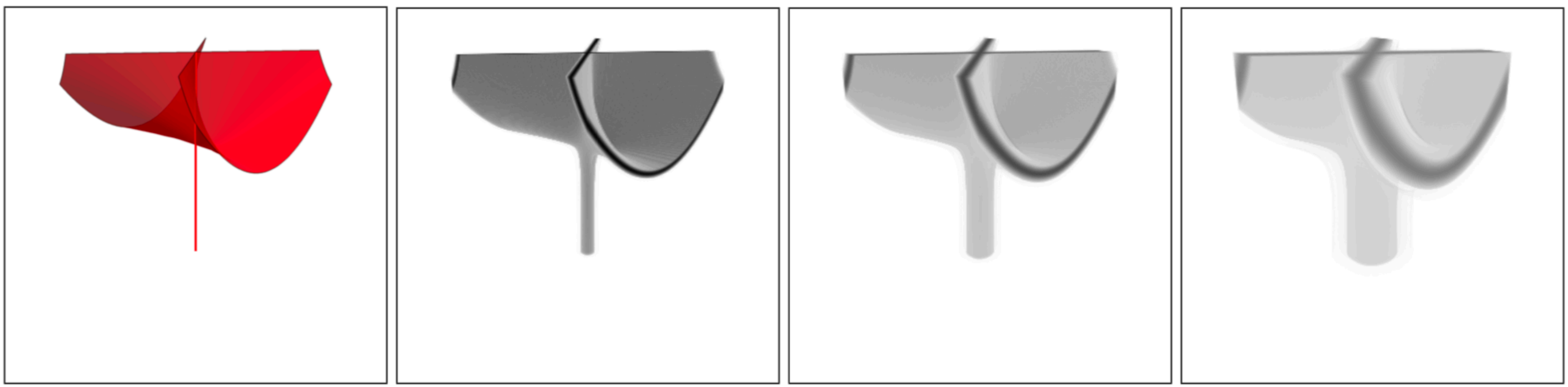
$(x^2+y^2-1)^3 - x^2y^3$

Variety normal (VN) distribution

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

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

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



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

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

The multivariate normal distribution has density

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

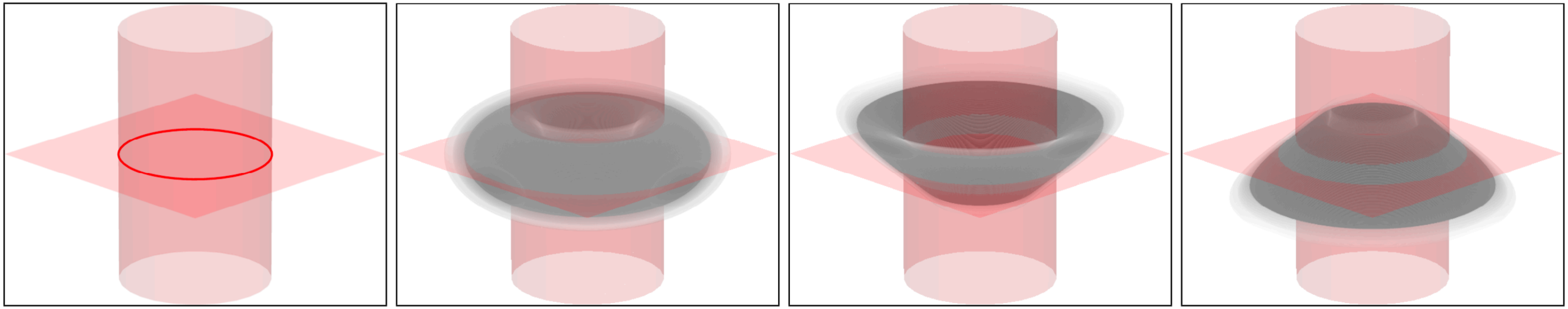
The multivariety normal distribution has density

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

The multivariety normal distribution has density

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

Example. $V(x^2 + y^2 - 1, z)$



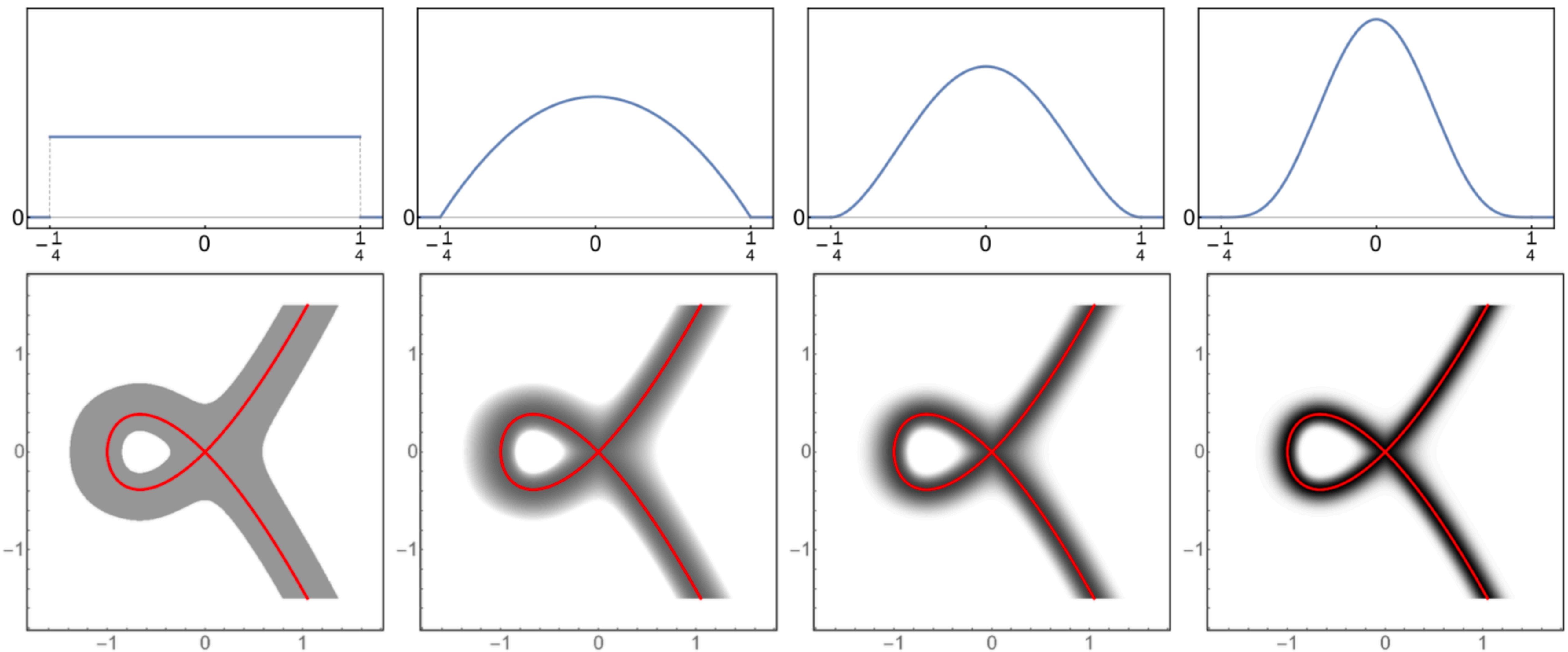
corr = 0 corr = .9 corr = –.9

+ correlation: mass aligns with same signed cells

– correlation: mass aligns with opposite signed cells

The kernel of any PDF can be used to induce variety distributions via location-scale transformations

Example. Beta distributions scaled and shifted by 1/2



Sampling and implementation

Markov chain Monte Carlo

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 ⇒ good software (BUGS, Stan)

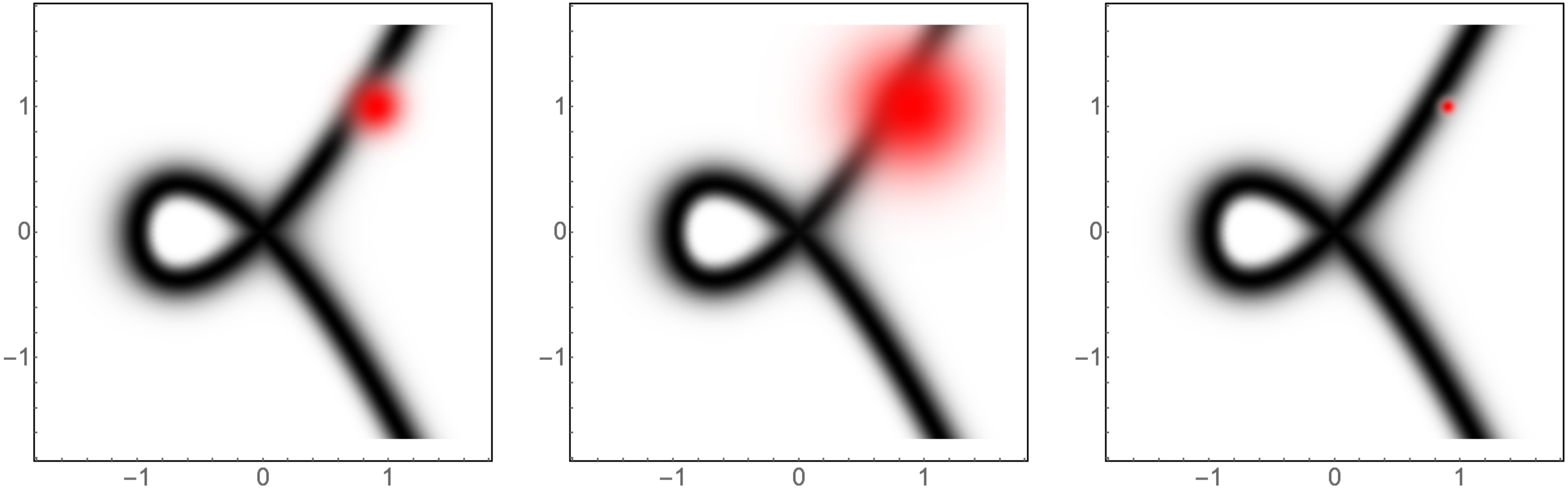
Iterate two basic steps (MCMC used here)

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

Random walk Metropolis

From current location, propose multivariate normal step



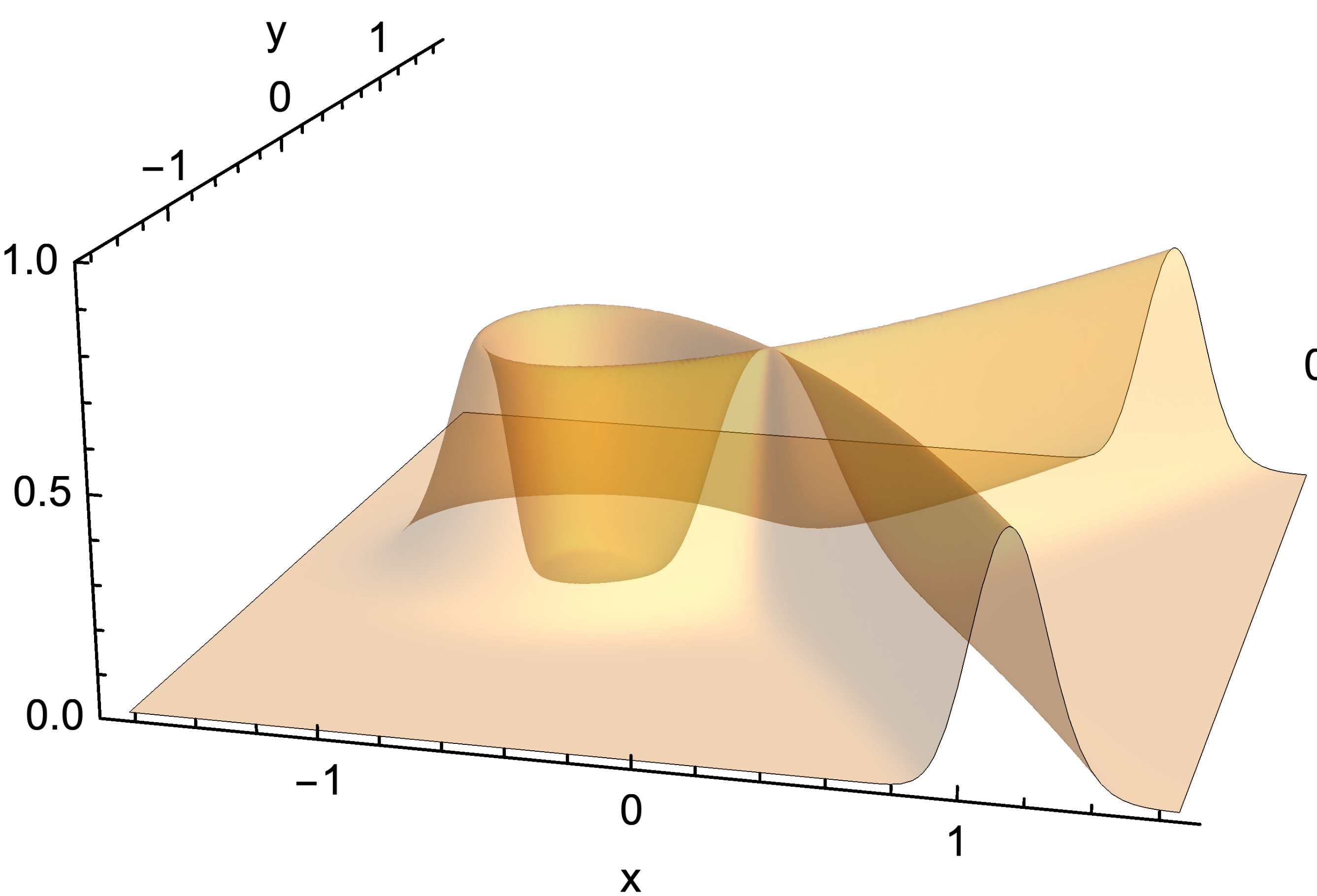
If variability is too large, unacceptably low acceptance rate

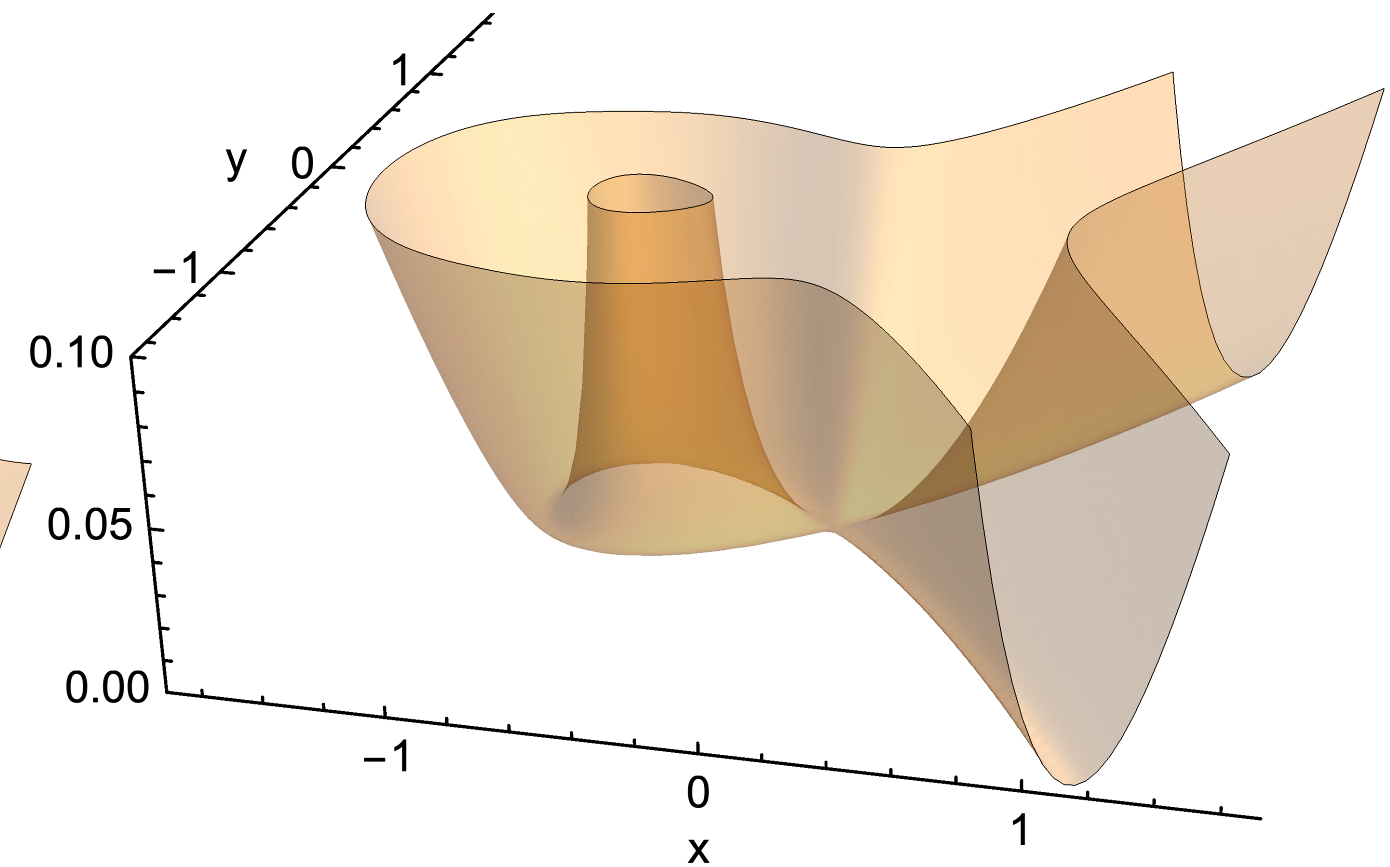
If variability is too small, unacceptably slow exploration

Both problems get worse in high dimensions

From current, propose step from physics simulation

Marble rolling on ($\bar{g}^2/\sigma^2$)'s surface, frictionless, given initial flick





$$p(\boldsymbol{x}|\overline{\boldsymbol{g}}, \boldsymbol{\Sigma}) \qquad -\log p(\boldsymbol{x}|\overline{\boldsymbol{g}}, \boldsymbol{\Sigma}) = \overline{\boldsymbol{g}}'\boldsymbol{\Sigma}^{-1}\overline{\boldsymbol{g}}$$

Impart random momentum, track position numerically, stop

Introduce auxiliary momenta variables, track level curve of Hamiltonian numerically, project back down

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

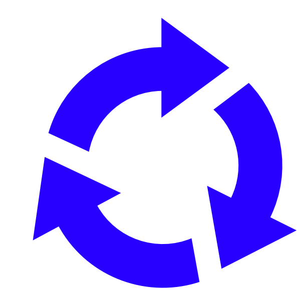
Stan specification

```
data {
  real y_obs;
  real<lower=0> si;
}

parameters {
  real x;
  real y;
}

transformed parameters {
  real g = (x^2 + (4*y)^2 - 1);
  real ndg = sqrt((2*x)^2 + (2*(4*y)*4)^2);
  real gbar = g / ndg;
}

model {
  y_obs ~ normal(gbar, 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:
```

Sample



Interfaces : R, Julia, Python, CLI, …

Many chains can be run in parallel

MVN distribution as a posterior distribution

The MVN distribution can be represented as the posterior distribution of a non-identifiable model

Bayes' theorem is

$$\underbrace{p(\boldsymbol{\theta}|\boldsymbol{y})}_{\text{Posterior}} = \frac{p(\boldsymbol{y}|\boldsymbol{\theta})p(\boldsymbol{\theta})}{\int p(\boldsymbol{y}|\boldsymbol{\theta})p(\boldsymbol{\theta})d\boldsymbol{\theta}} \propto \underbrace{p(\boldsymbol{y}|\boldsymbol{\theta})}_{\text{Likelihood}}\underbrace{p(\boldsymbol{\theta})}_{\text{Prior}}$$

Data Parameter

MVN is the posterior of the model $\boldsymbol{Y} \sim \mathcal{N}_m\left(\overline{\boldsymbol{g}}, \boldsymbol{\Sigma}\right)$ with an improper flat prior on $\boldsymbol{x}$ and $\boldsymbol{y} = \boldsymbol{0}$ is observed

Roles of data and parameter swap

Bayes – Greek varies, Latin fixed/known

Here – Greek fixed/known, Latin varies

Bayes' theorem is

$$\underbrace{p(\boldsymbol{\theta}|\boldsymbol{y})}_{\text{Posterior}} = \frac{p(\boldsymbol{y}|\boldsymbol{\theta})p(\boldsymbol{\theta})}{\int p(\boldsymbol{y}|\boldsymbol{\theta})p(\boldsymbol{\theta})d\boldsymbol{\theta}} \propto \overbrace{p(\boldsymbol{y}|\boldsymbol{\theta})}^{\text{Likelihood}}\overbrace{p(\boldsymbol{\theta})}^{\text{Prior}}$$

Data Parameter

$$p(\boldsymbol{x}|\boldsymbol{g}, \boldsymbol{\Sigma}) \propto \overbrace{\exp\left\{-\frac{1}{2}(\mathbf{0} - \overline{\boldsymbol{g}}(\boldsymbol{x}|\boldsymbol{\beta}))'\boldsymbol{\Sigma}^{-1}(\mathbf{0} - \overline{\boldsymbol{g}}(\boldsymbol{x}|\boldsymbol{\beta}))\right\}}^{\text{Likelihood}} \underbrace{\times 1}_{\text{Prior}}$$

Parameter

Data Given

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

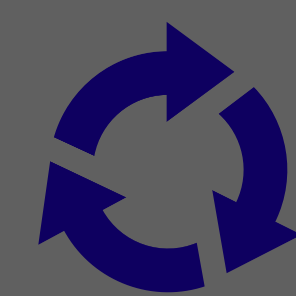
Stan specification

```
data {
  real y_obs;
  real<lower=0> si;
}

parameters {
  real x;
  real y;
}

transformed parameters {
  real g = (x^2 + (4*y)^2 - 1);
  real ndg = sqrt((2*x)^2 + (2*(4*y)*4)^2);
  real gbar = g / ndg;
}

model {
  y_obs ~ normal(gbar, 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:
```

Sample

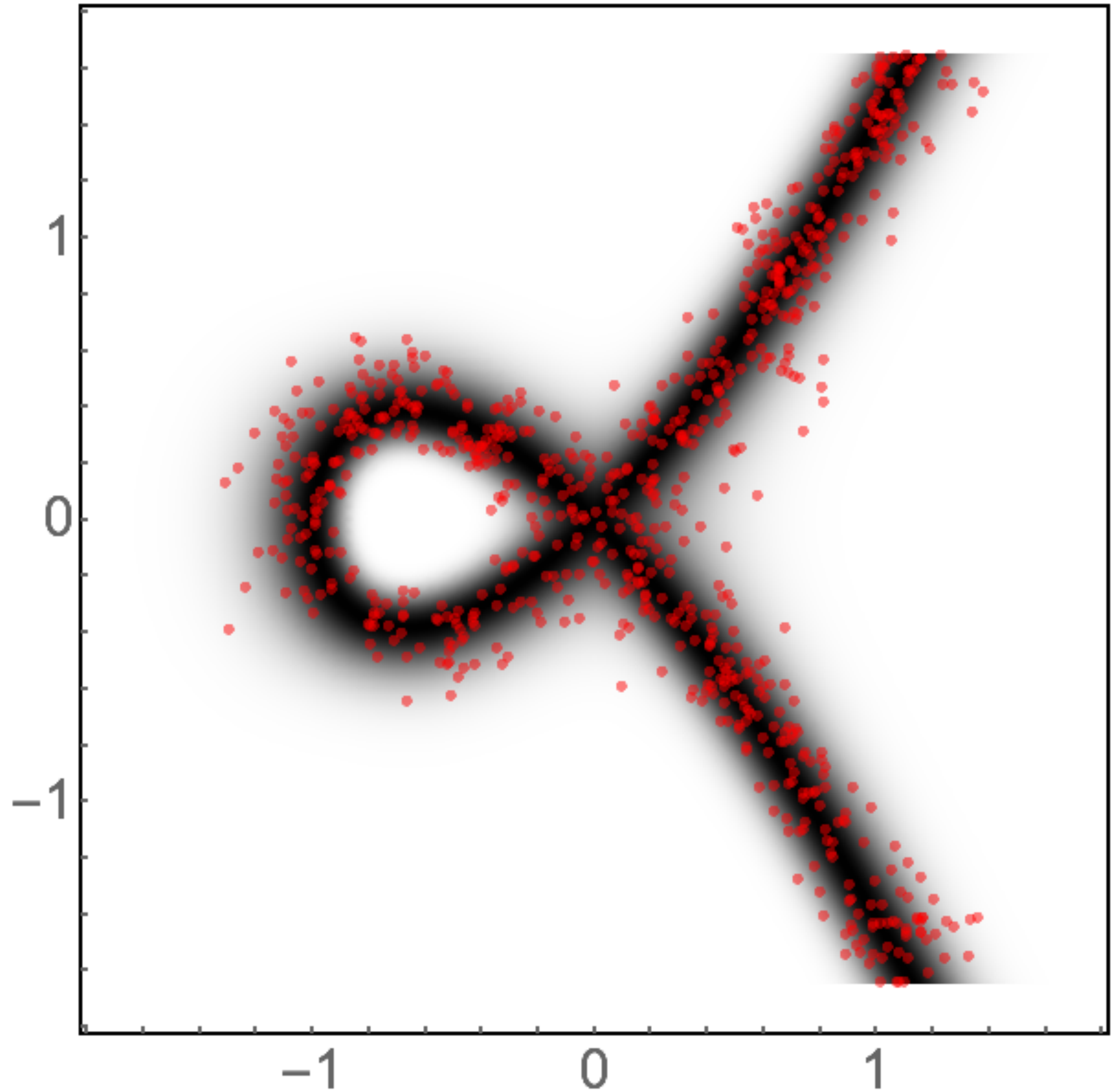


Interfaces : R, Julia, Python, CLI, …

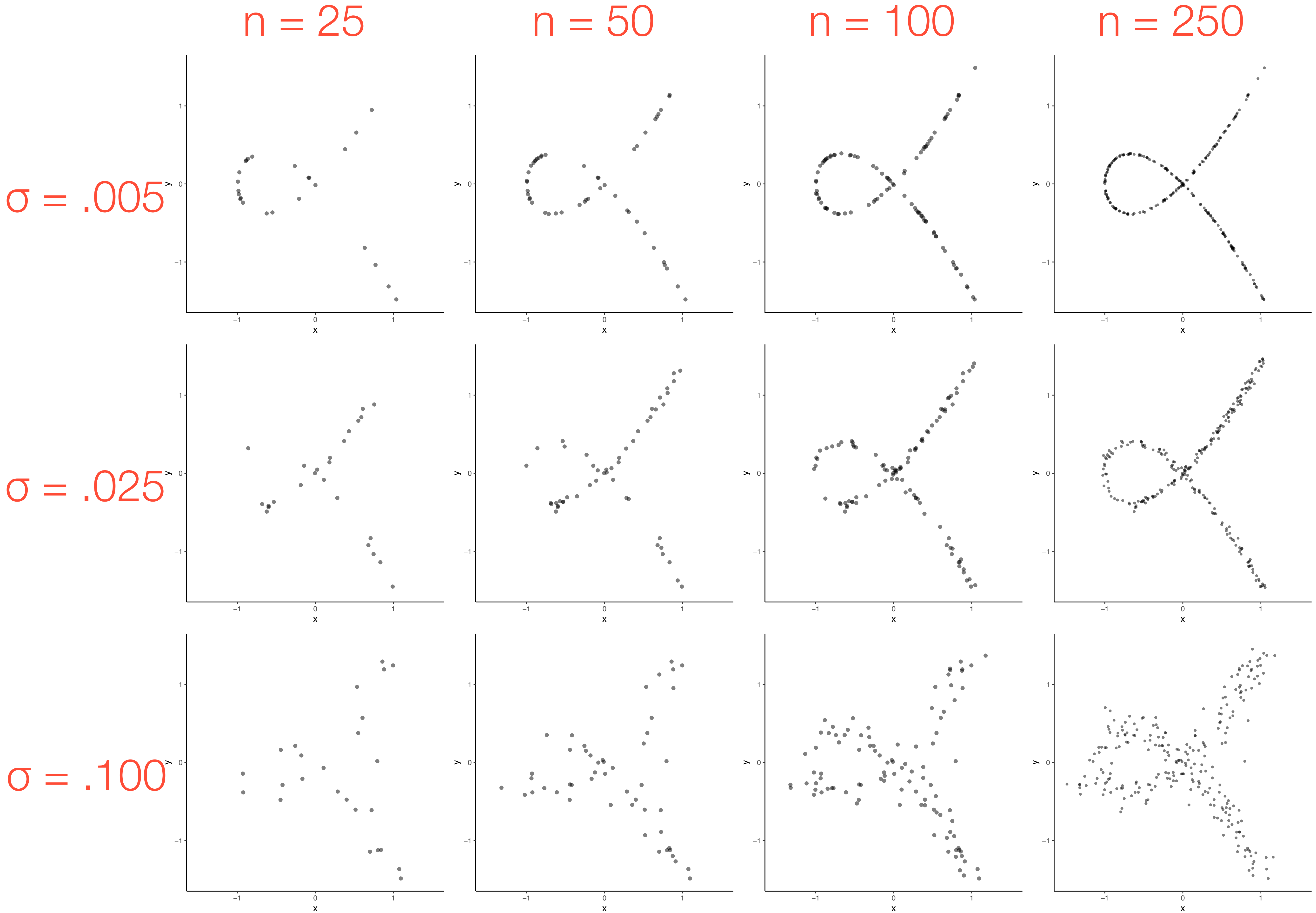
Many chains can be run in parallel

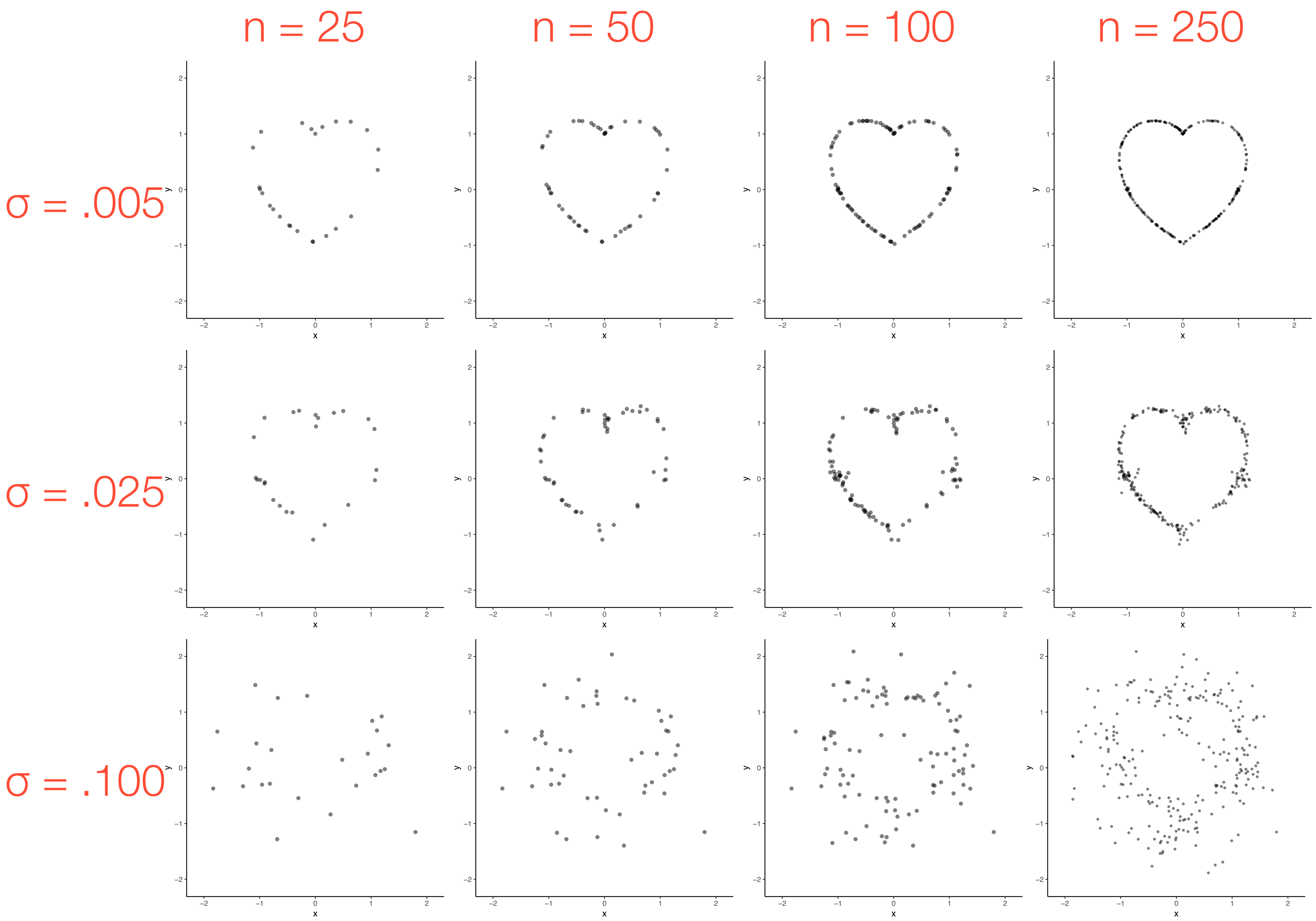
Examples

VN(alpha curve, σ = .10); 100 x eight chains = 800 abs

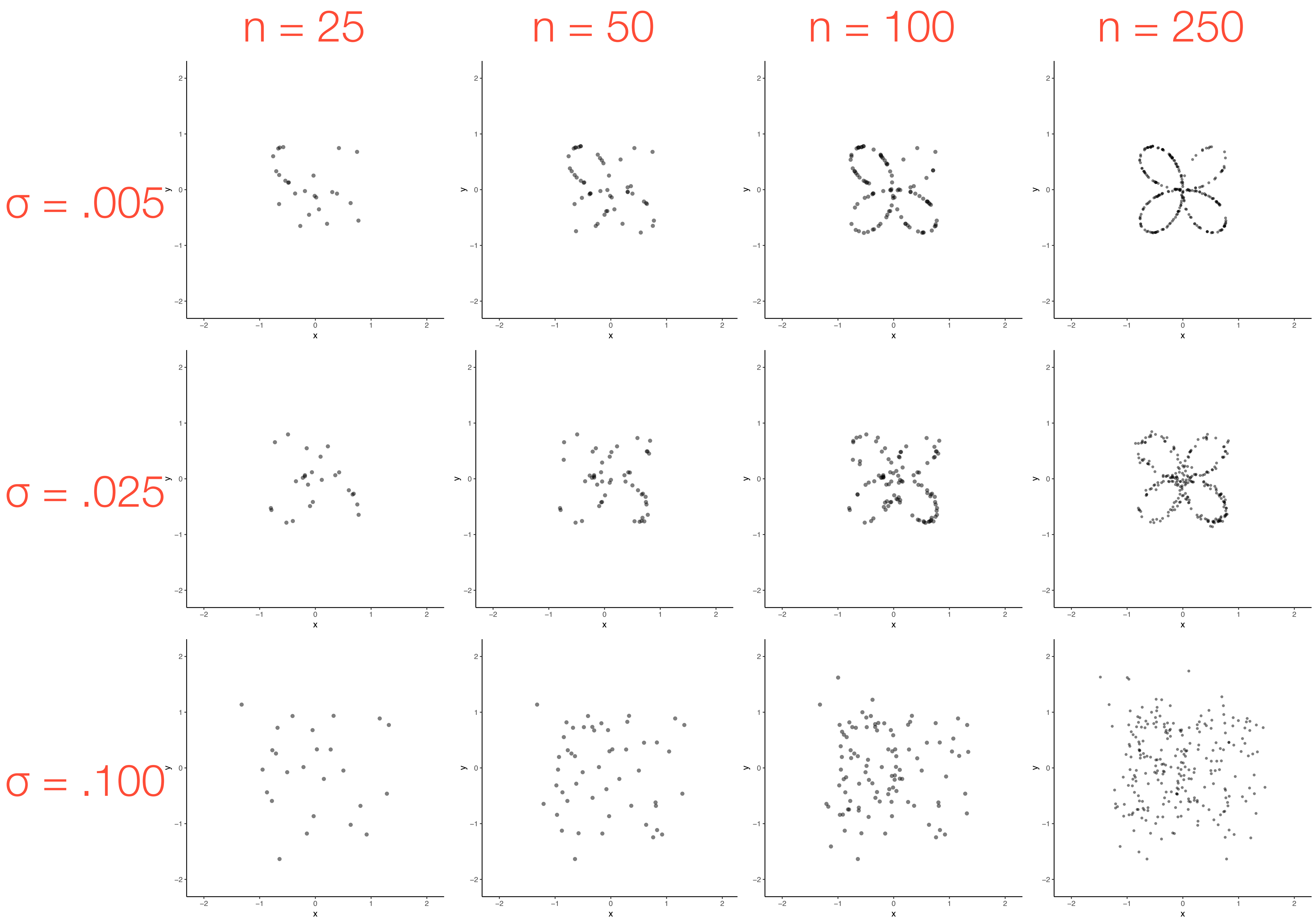


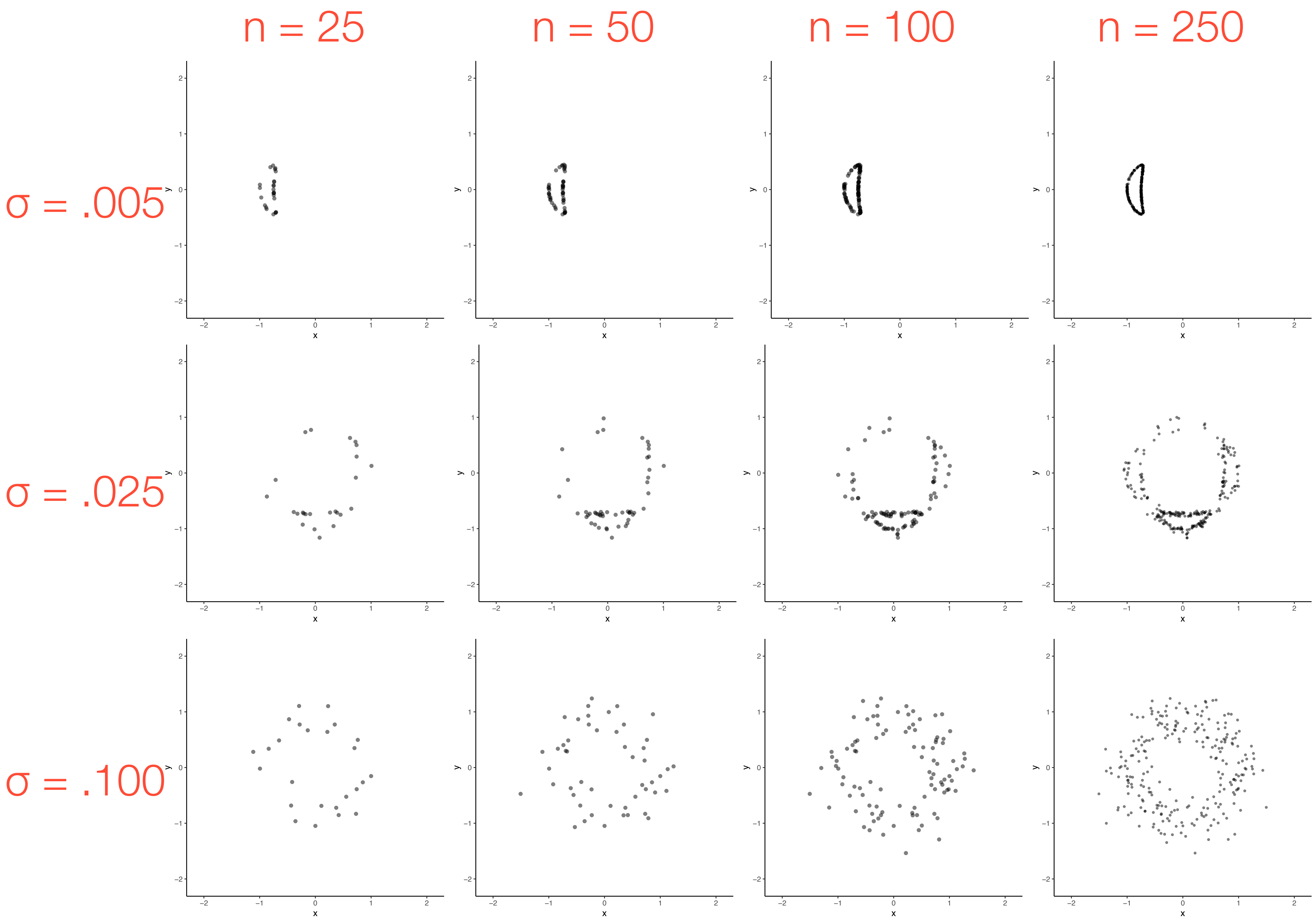
Examples


n = 25
n = 50
n = 100
n = 250
σ = .005
σ = .025
σ = .100



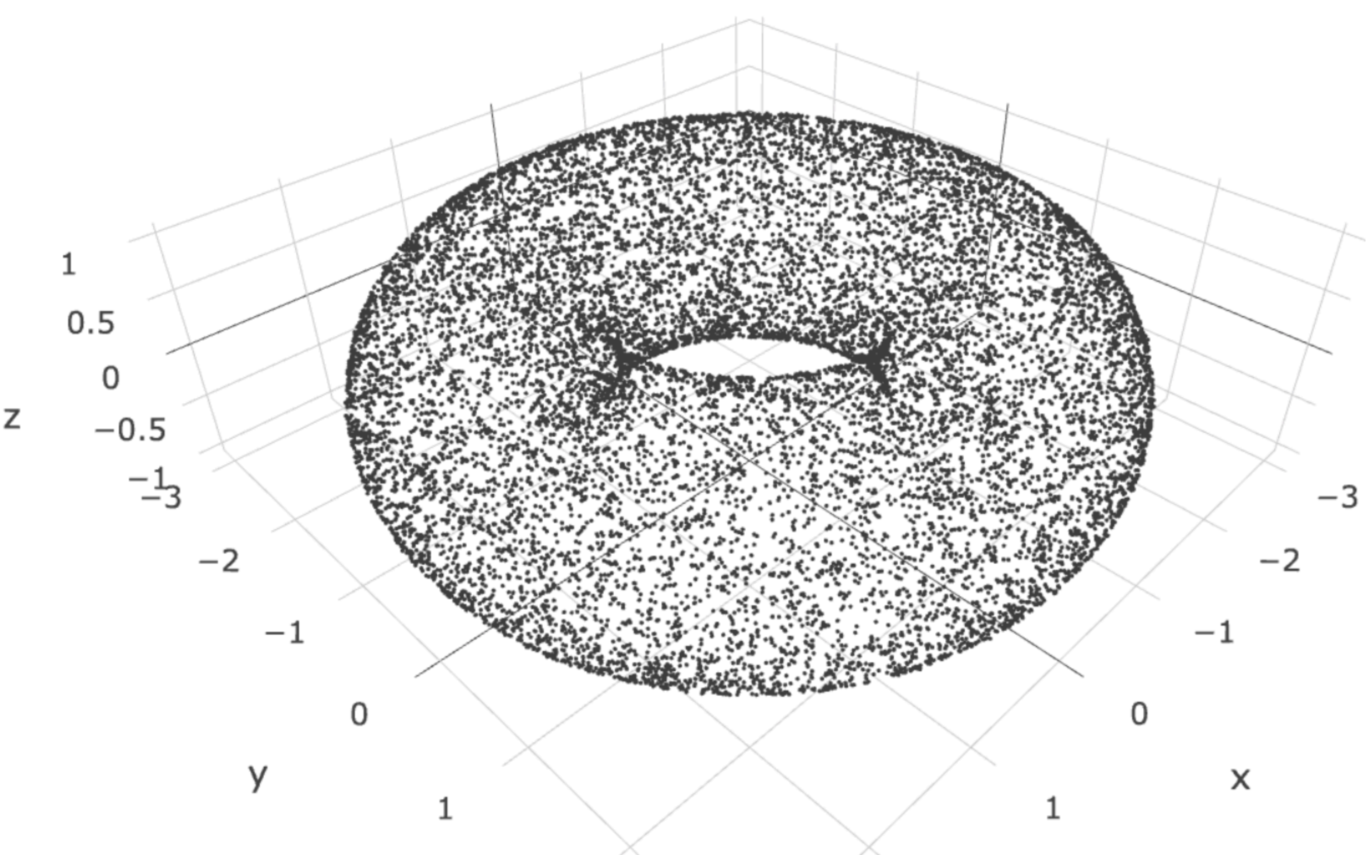
n = 25
n = 50
n = 100
n = 250
σ = .005
σ = .025
σ = .100
x
y

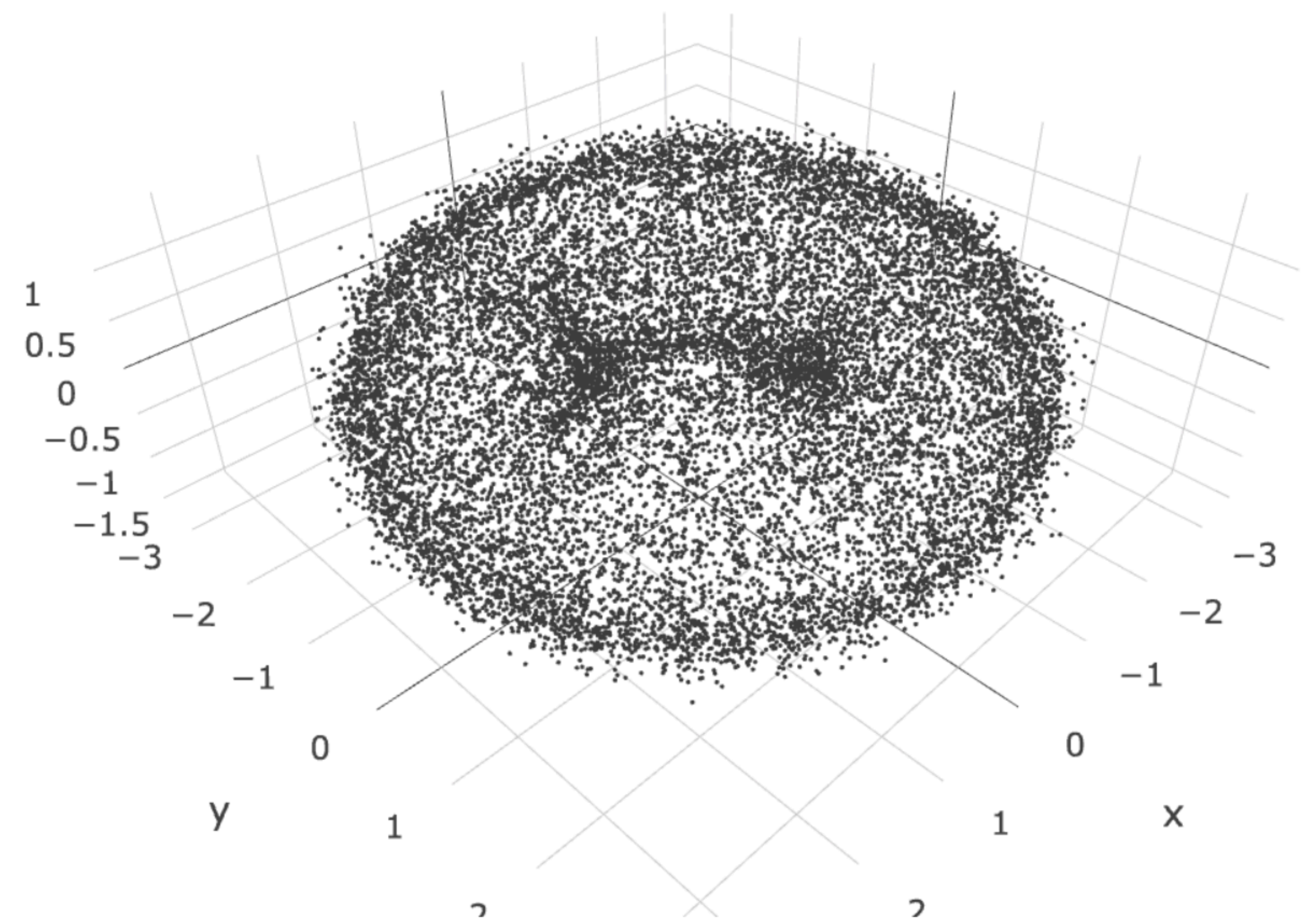

n = 25
n = 50
n = 100
n = 250
σ = .005
σ = .025
σ = .100
x
y


n = 25
n = 50
n = 100
n = 250
σ = .005
σ = .025
σ = .100
x
y

BAYLOR UNIVERSITY

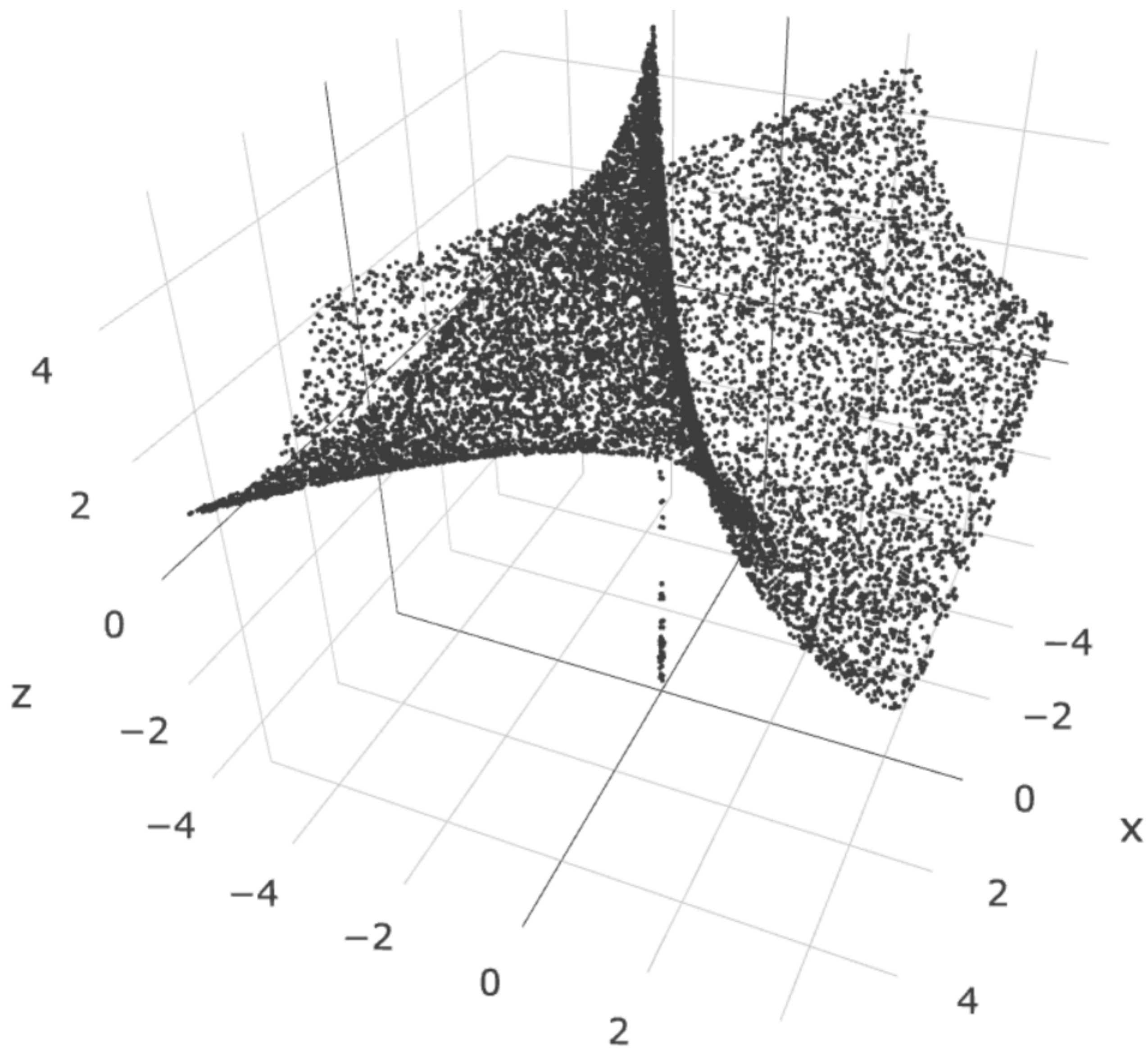
VN(torus, σ = .005/.100); 2000 points

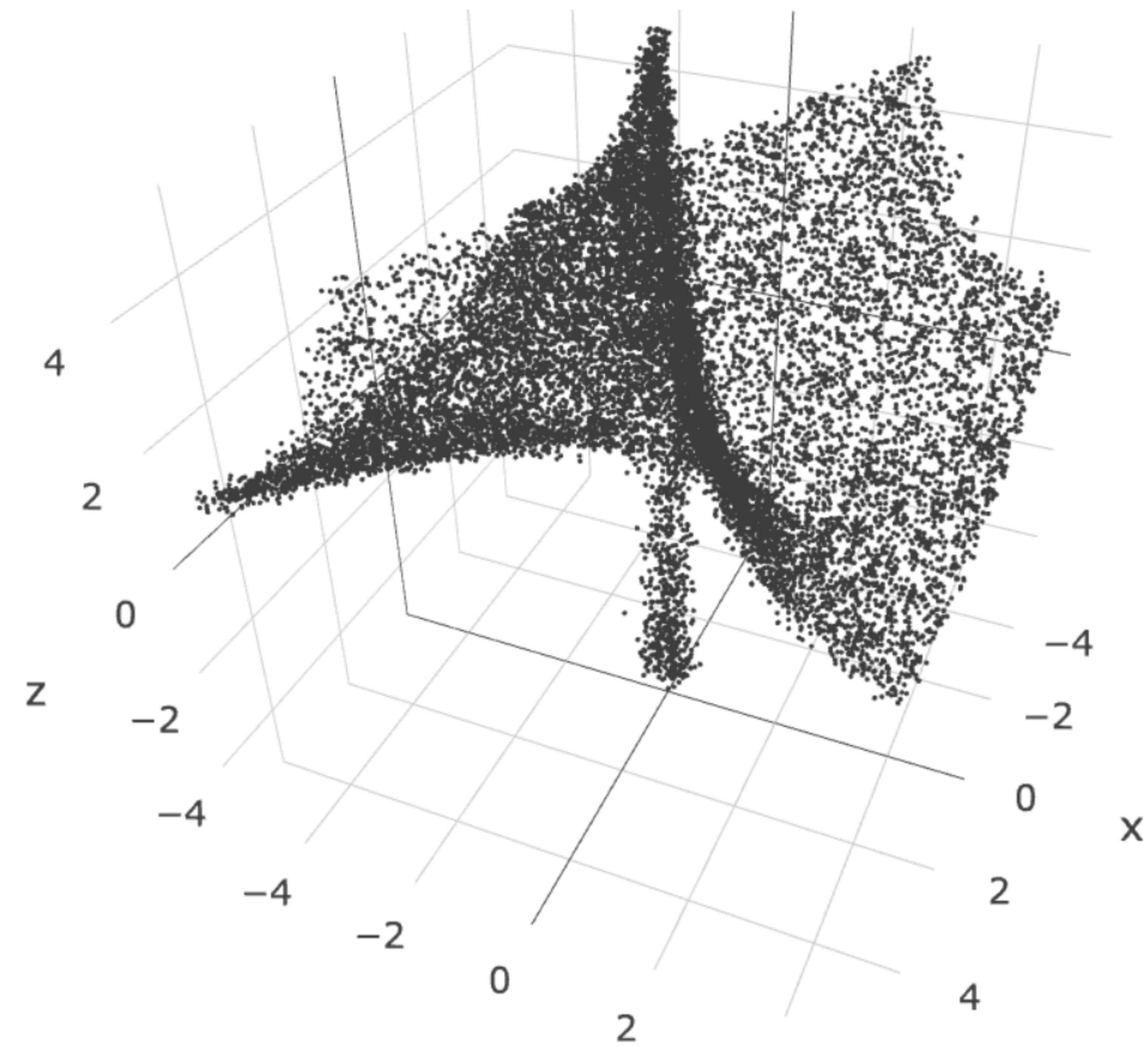




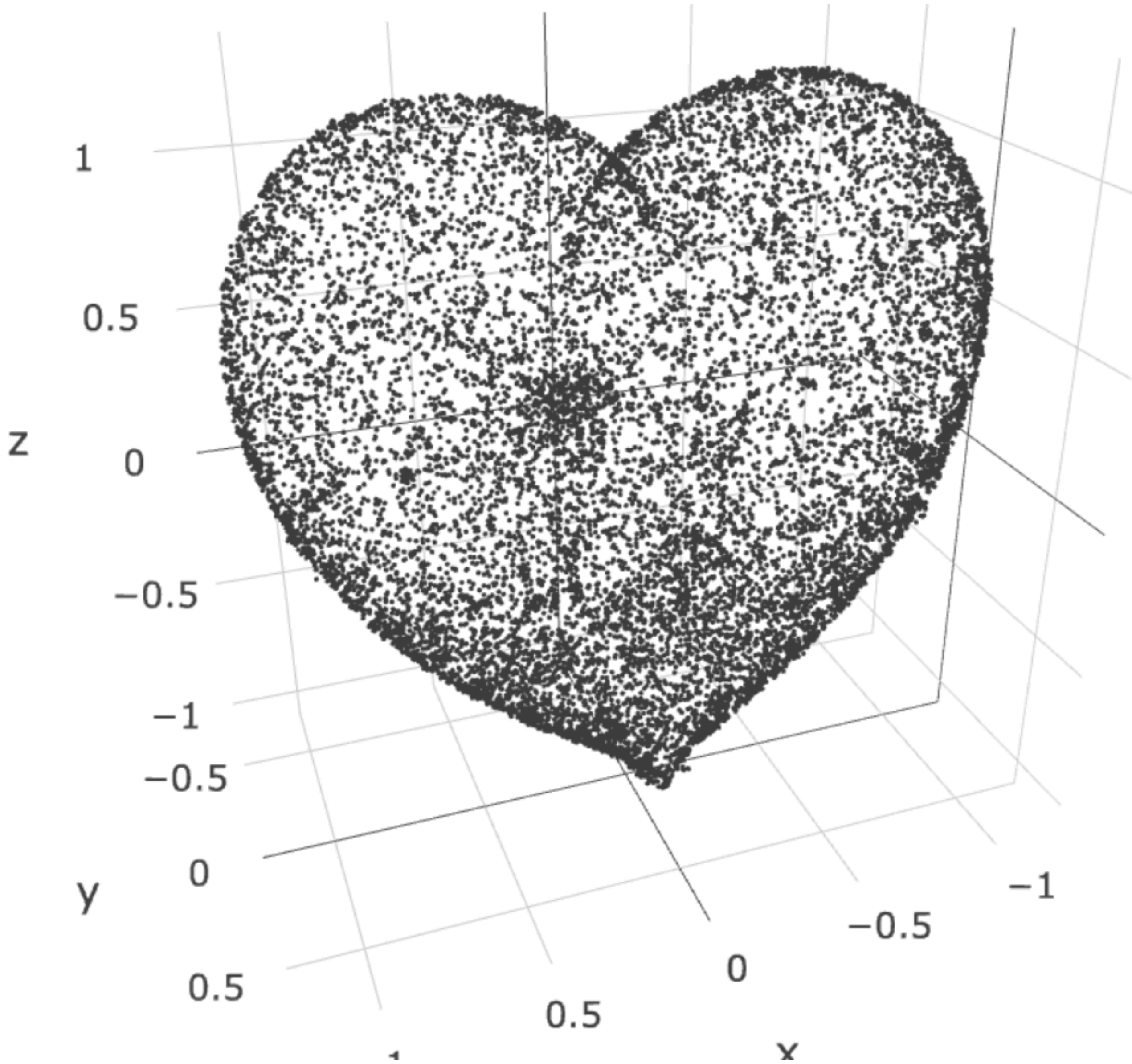
Examples

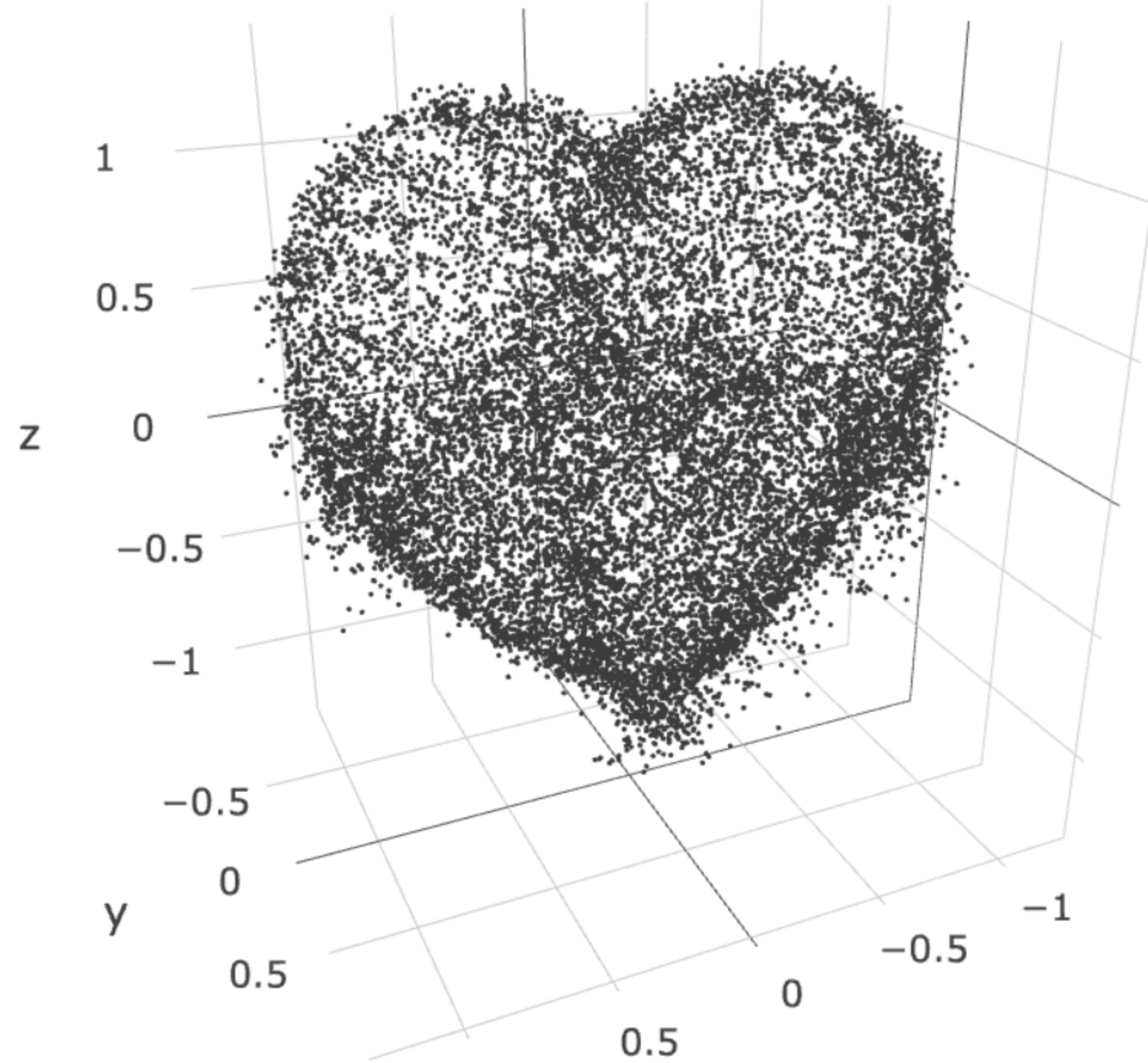
VN(whitney, σ = .010/.100); 2000 points





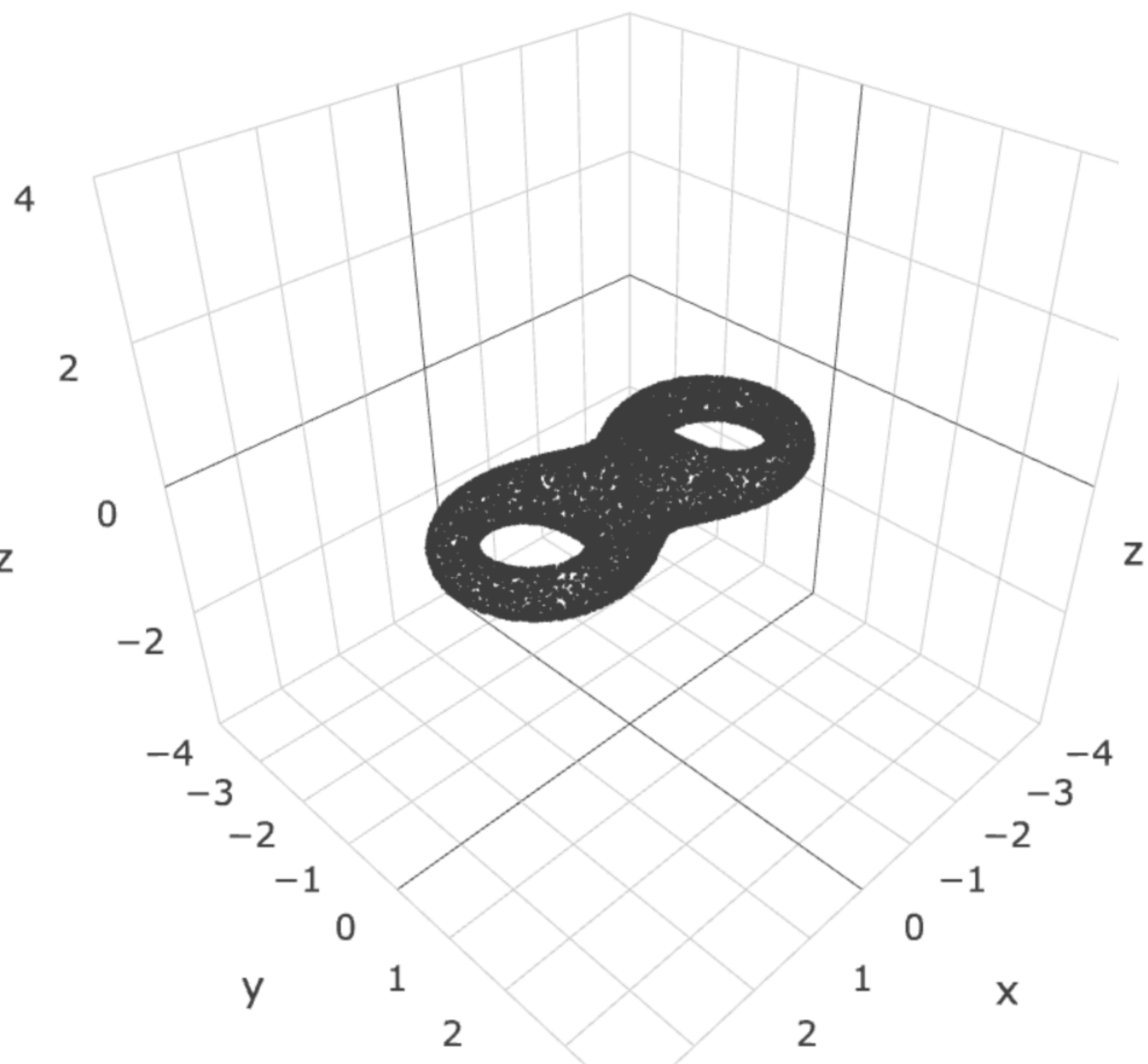
VN(3d heart, σ = .005/.025); 2000 points

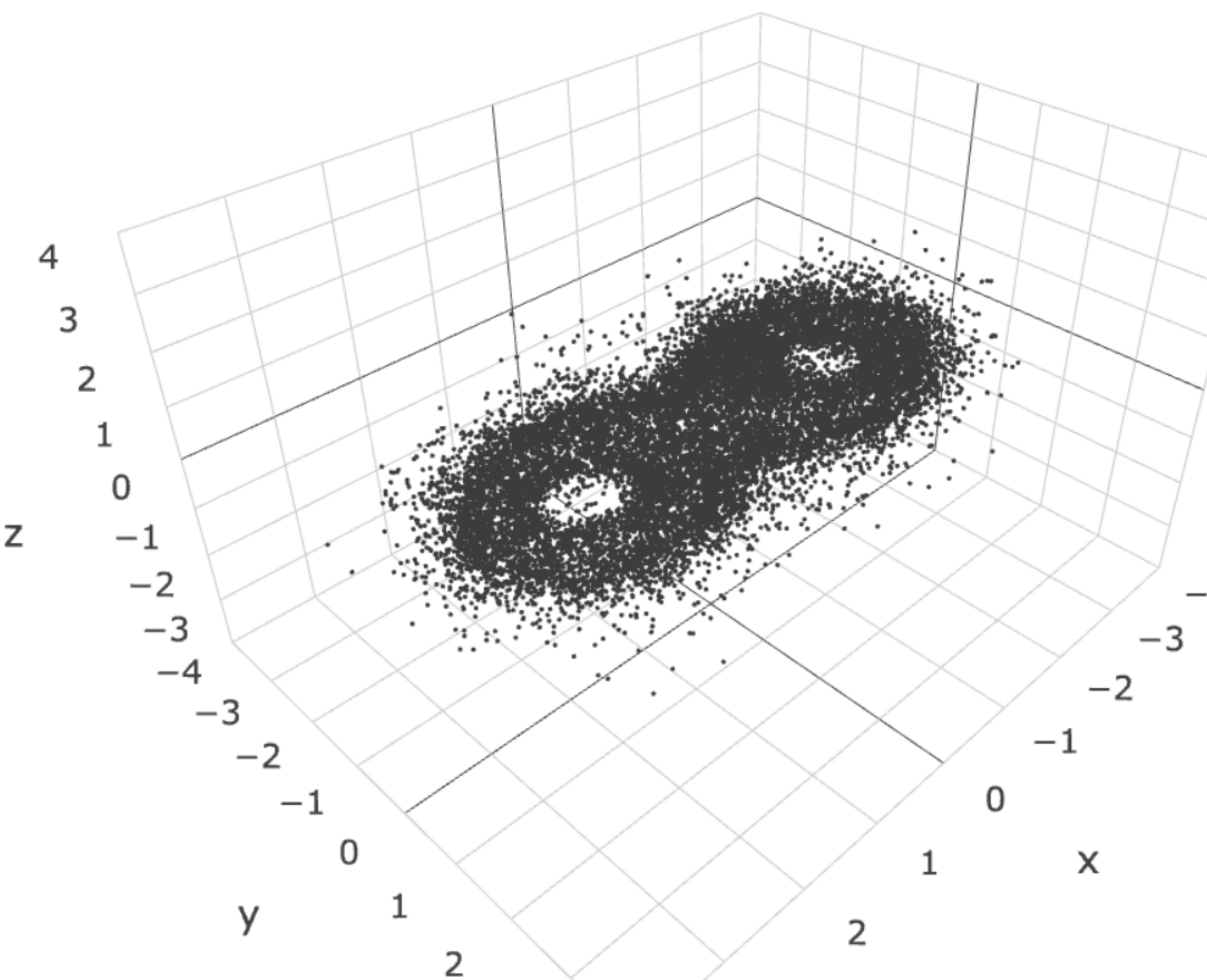




BAYLOR UNIVERSITY

VN(2-torus, σ = .005/.100); 2000 points





Moving to variety

The algorithm works remarkably well even for small σ

For points <u>on</u> the variety, endgames can be used

Basic : Newton, gradient descent, etc.

Harder : Projection with Bertini

Concluding thoughts

Experimentally 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.