Improved predictive sampling using quasi-Monte Carlo with application to neutron-cross-section evaluation

Ken Hanson

T-16, Nuclear Physics; Theoretical Division
Los Alamos National Laboratory

Los Alamos
NATIONAL LABORATORY

This presentation available at
http://www.lanl.gov/home/kmh/

April 14, 2005 Center for Devices and Radiological Health, FDA

LA-UR-05-3096

Collaborators and counselors

Toshihiko Kawano, Patrick Talou, Mark Chadwick

Stephanie Frankle, Bob Little

Mike Fitzpatrick, David Haynor, Dave Brown, Brandon Gallas

Chris Orum, Bill Press

Dave Higdon, Brian Williams, Rick Picard, David Mease, Mike McKay

Max Gunzburger, Vicente Romero, Steve Wojtkiewicz

Tony Warnock, Paul Pederson

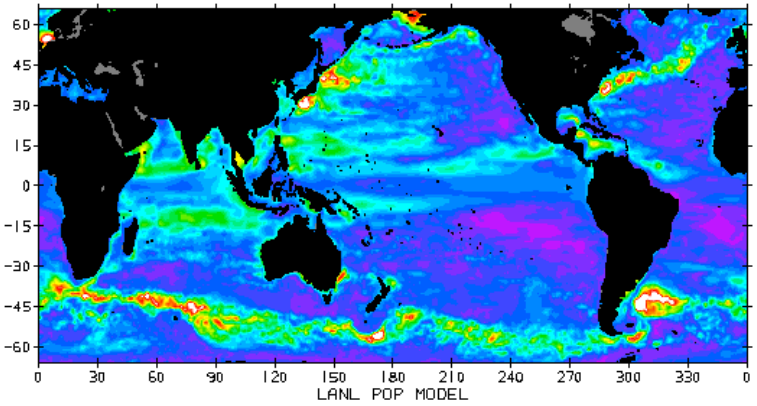
Overview

- Quasi-Monte Carlo (QMC) – purpose
- Digital halftoning – purpose and constraints
- New approaches to generating sample sets with uniform spacing
  - halftoning algorithm provides good point sets for QMC
  - leads to Repulsive Particle Model (RPM) and Centroidal Voronoi Tessellation (CVT)
- Point sets for sampling distributions in high dimensions
  - predictive sampling – estimate prediction mean and uncertainty
- Neutron cross-section evaluation
  - combine directly measured neutron cross sections for fission of ^{239}Pu with a high-accuracy criticality measurement
  - Bayesian update
  - objective – to characterize prediction uncertainty using 30 samples for 30 input parameters

Context – big physics simulation codes

- Computer simulation codes
  - many input parameters, many output variables
  - very expensive to run; days to weeks on super computers
- Important to assess uncertainties in predictions - thus need to
  - compare codes to experimental data; make inferences
  - use advanced methods to estimate sensitivity of simulation outputs to inputs
    - Latin square (hypercube), stratified sampling, quasi-Monte Carlo, CVT
- Examples of complex simulations
  - nuclear-reactor design
  - ocean and atmosphere modeling
  - aircraft design; space shuttle design
  - casting of metals

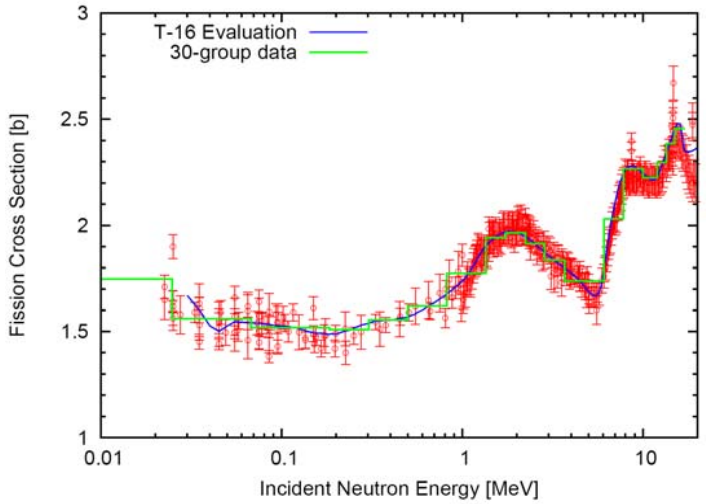
calculations can take 3 months!



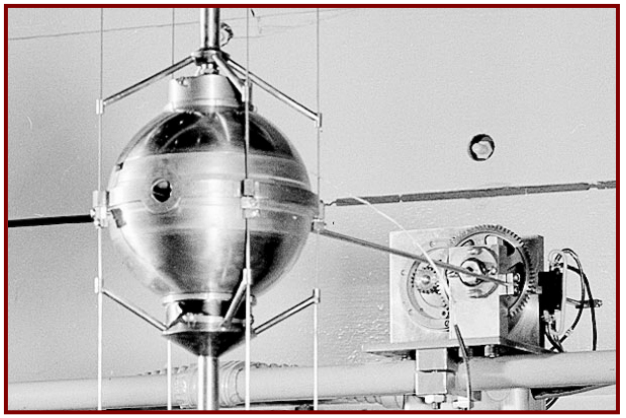
Our application

- Focus on neutron cross sections
- Aim to improve our knowledge of the cross sections by incorporating a high-precision criticality measurement; integral constraint
- Criticality experiment simulated with a discrete-ordinate code, based on 30 energy bins
- Ultimate goal is to use the improved cross sections to predict other similar physical situations
- Need to characterize prediction uncertainties; 30-D parameter space

Neutron fission cross section



Criticality experiment



Monte Carlo integration techniques

- Generic purpose of Monte Carlo
  - estimate integral of a function over a specified region R in m dimensions, based on evaluations at n sample points

$$\int_R f(\boldsymbol{x})\, d\boldsymbol{x} = \frac{V_R}{n} \sum_{\boldsymbol{i}=1}^{\boldsymbol{n}} f(\boldsymbol{x_i})$$

- Constraints
  - integrand not available in analytic form, but calculable
  - function evaluations may be expensive, so minimize number
- Algorithmic approaches – want best accuracy with fixed number of function evaluations n
  - simple quadrature (Riemann) – good for few dimensions; rms err ~ n^{-1}
  - Monte Carlo – useful for many dimensions; rms err ~ $n^{-1/2}$
  - quasi-Monte Carlo – reduce number of evaluations; rms err ~ n^{-1}

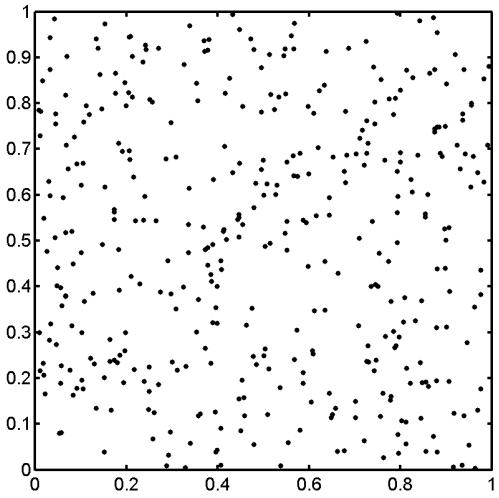
Quasi-Monte Carlo

- Purpose
  - estimate integral of a function over a specified domain in *d* dimensions
  - obtain better rate of convergence of integral estimation than occurs in classic Monte Carlo
- Constraints
  - integrand function not available analytically, but calculable
  - function known (or assumed) to be reasonably well behaved, e.g. smooth
- Standard QMC approaches use low-discrepancy sequences; product space (Halton, Sobel, Faure, Hammersley, …)
  - most studies usually involve many samples in a few dimensions
- Propose here new ways of generating sample point sets
  - our focus is on a few samples in high dimensions

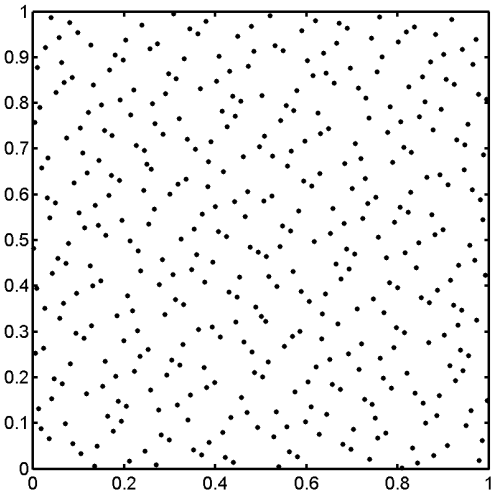
Point set examples

- Scatter plots of different kinds of point sets (400 points)
- Halton sequence reduces clustering that occurs in random seqs.
- If quasi-MC sequences have better integration properties than random, is halftone pattern even better?

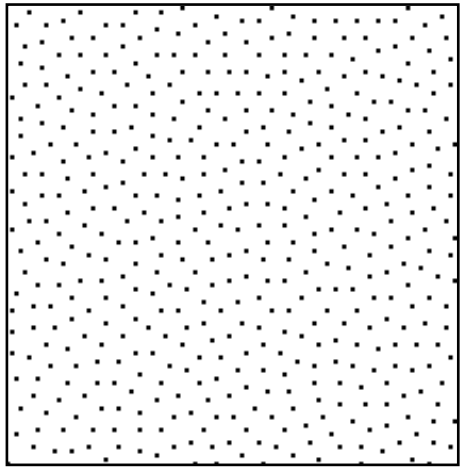
Random
(independent)



Quasi-Random
(Halton sequence)



Halftone
(DBS sky)

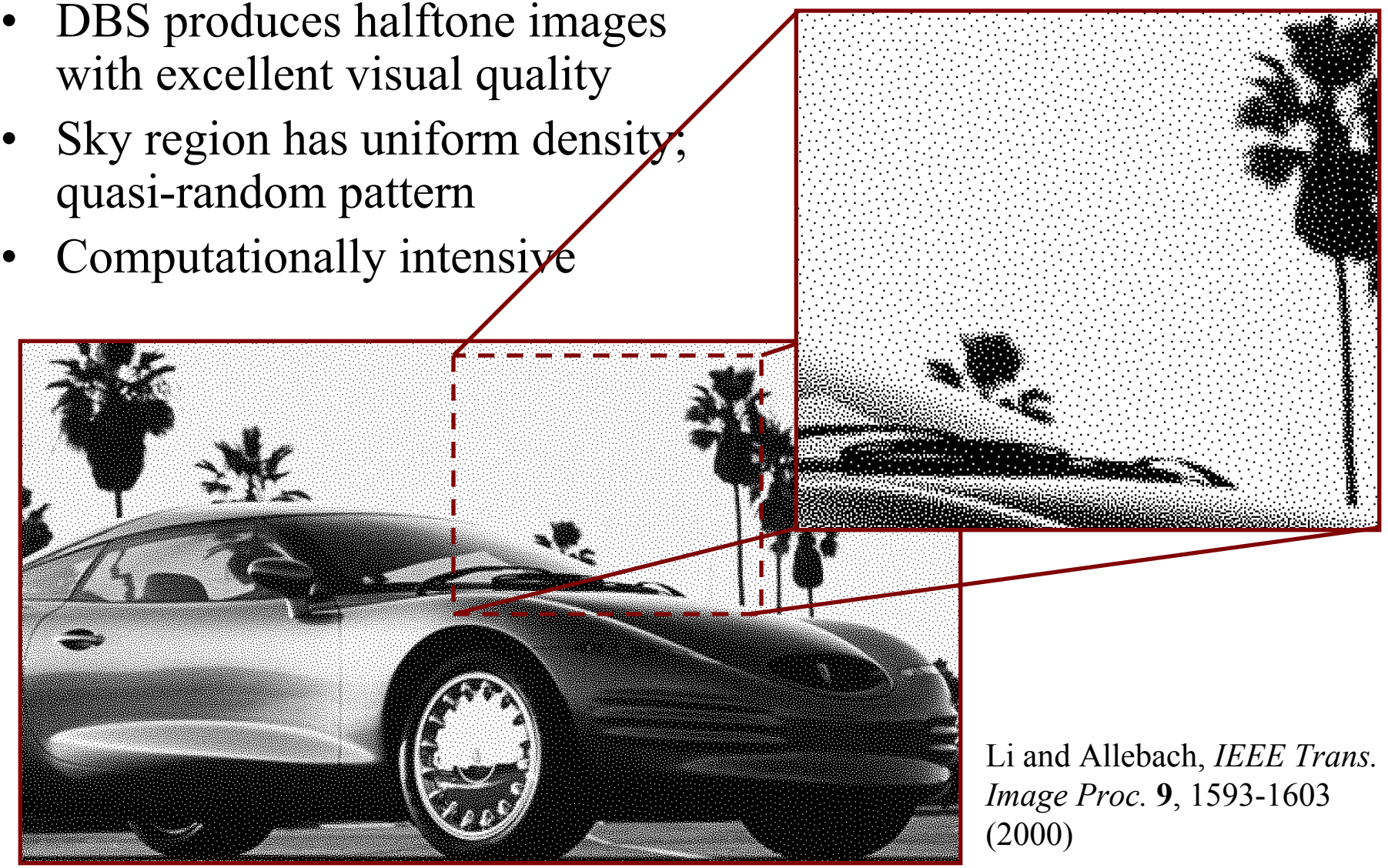


Digital halftoning techniques

- Purpose
  - render a gray-scale image by placing black dots on white background
  - make halftone rendering **look** like original gray-scale image
  - related to characteristics of human observer
  - important for laser printers
- Constraints
  - resolution – size and closeness of dots, number of dots
  - speed of rendering
- Various algorithmic approaches
  - error diffusion, look-up tables, blue-noise, …
  - concentrate here on Direct Binary Search (Allebach et al.)

Direct Binary Search example

- DBS produces halftone images with excellent visual quality
- Sky region has uniform density; quasi-random pattern
- Computationally intensive



Li and Allebach, *IEEE Trans. Image Proc.* **9**, 1593-1603 (2000)

Direct Binary Search (DBS) algorithm

- Consider digital halftone image to be composed of black or white pixels
- Cost function is based on perception of two images

$$\varphi = \left|\mathbf{h} * (\mathbf{d} - \mathbf{g})\right|^2$$

  - where **d** is the dot image, **g** is the gray-scale image to be rendered, * represents convolution, and **h** is the image of the blur function of the human eye, for example, $(w^2 + r^2)^{-3/2}$
- To minimize φ
  - start with a collection of dots with average local density ~ **g**
  - iterate sequentially through all image pixels;
  - for each pixel, swap value with neighborhood pixels, or toggle its value to reduce φ
- Edge effects must be dealt with
  - in above, dot image surrounded by field of uniform density

Minimum Visual Discrepancy (MVD) algorithm

Inspired by Direct Binary Search halftoning algorithm:

- Start with an initial set of points
- Goal is to create uniformly distributed set of points
- Cost function is variance in blurred point image

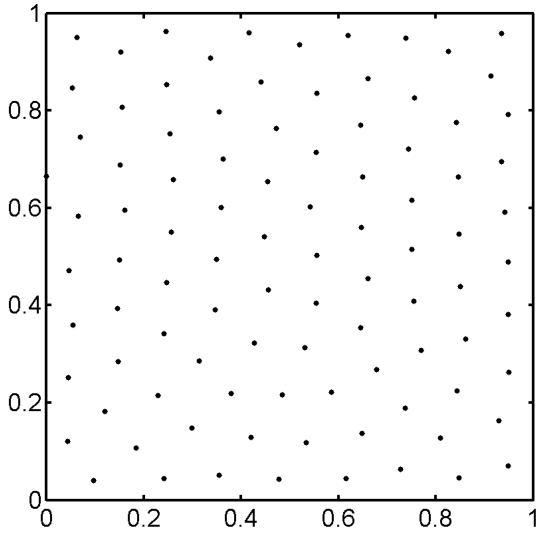
$$\psi = \mathrm{var}(\mathbf{h} * \mathbf{d})$$

  - where **d** is the point (dot) image, **h** is the blur function of the human eye, and * represents convolution
- Minimize ψ by
  - starting with some point set (random, stratified, Halton,…)
  - visiting each point in random order;
  - moving each point in 8 directions, and accept move that lowers ψ the most

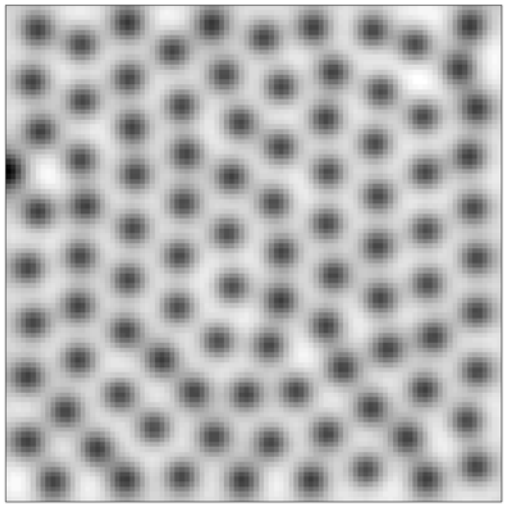
Minimum Visual Discrepancy (MVD) algorithm

- MVD result; start with 95 points from Halton sequence
- MVD objective is to minimize variance in blurred image
- Effect is to force points to be evenly distributed, or as far apart from each other as possible
- Might expect global minimum is a regular pattern

MVD, 95



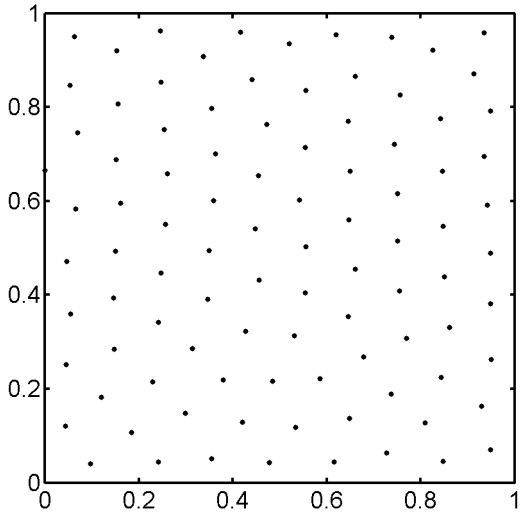
Blurred image



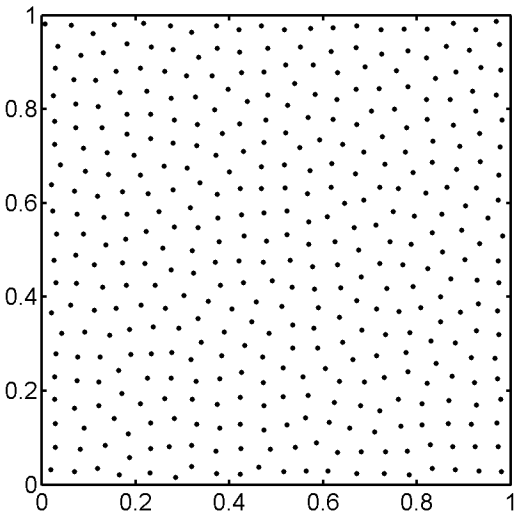
MVD results

- In each optimization, final pattern depends on initial point set
  - algorithm seeks local minimum, not global (as does DBS)
- Patterns somewhat resemble regular hexagonal array
  - similar to lattice structure in crystals or glass
  - however, lack long-range (coarse scale) order
  - best to start with point set with good long-range uniformity

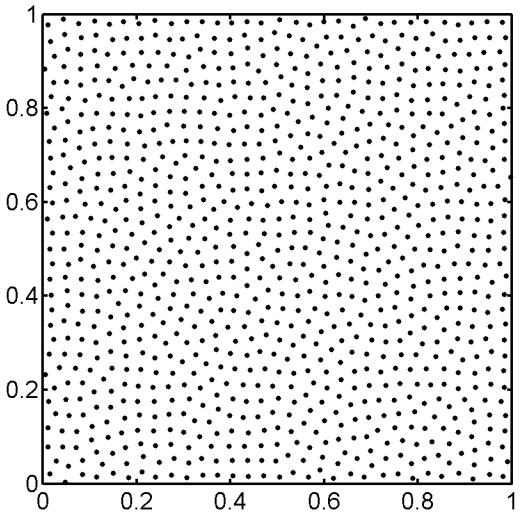
MVD, 95



MVD, 400



MVD, 1000



Repulsive particle model

- Model points as set of interacting (repulsive) particles
- Cost function is the potential

$$\psi = \sum_{i,\, j \geq i+1} V(\boldsymbol{x}_i, \boldsymbol{x}_j) + \sum_i U(\boldsymbol{x}_i)$$

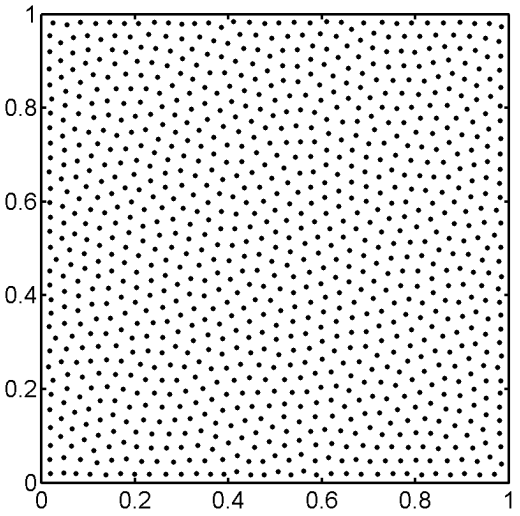
  - where V is a particle-particle interaction potential and U is a particle-boundary potential
  - particles are repelled by each other and from boundary
- Minimize ψ by moving particles by small steps
- This model is **analytically equivalent** to Minimum Visual Discrepancy (V and U directly related to blur function **h**)
- Suitable for generating point sets in high dimensions

Repulsive particle model

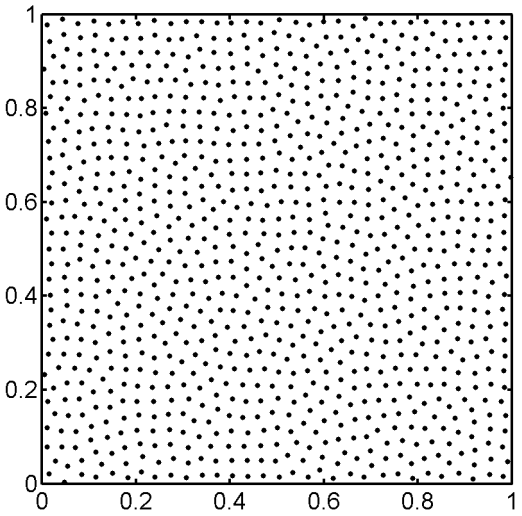
- Equivalent to Minimum Visual Discrepancy algorithm
- Example of repulsive-particle results
  - resulting point pattern is visually indistinguishable from MVD pattern

1000 points

Repulsive Particle Model



MVD

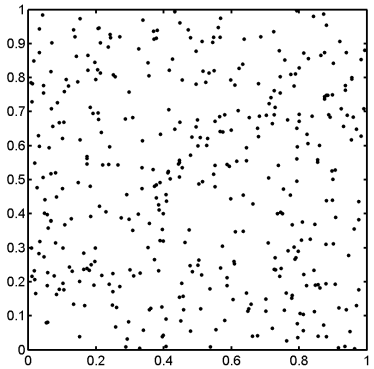


Point set examples

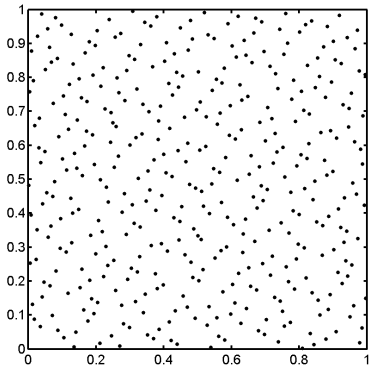
- Compare various kinds of point sets (400 points)
  - varying degrees of randomness and uniformity
- As the points become more uniformly distributed, the more accurate are the values of estimated integrals
- Example:

RMS relative accuracies of integral of $\text{func2} = \prod_{i} \exp\left(-2\left|x_i - x_i^0\right|\right); \quad 0 < x_i^0 < 1$

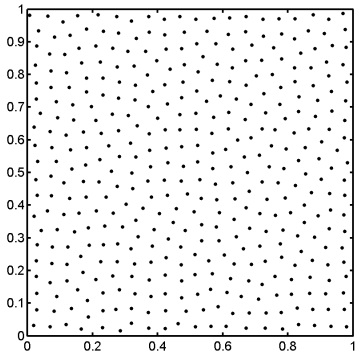
Random, 2.5%



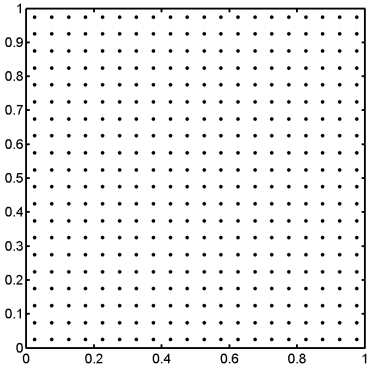
Halton, 0.5%



MVD, 0.14%

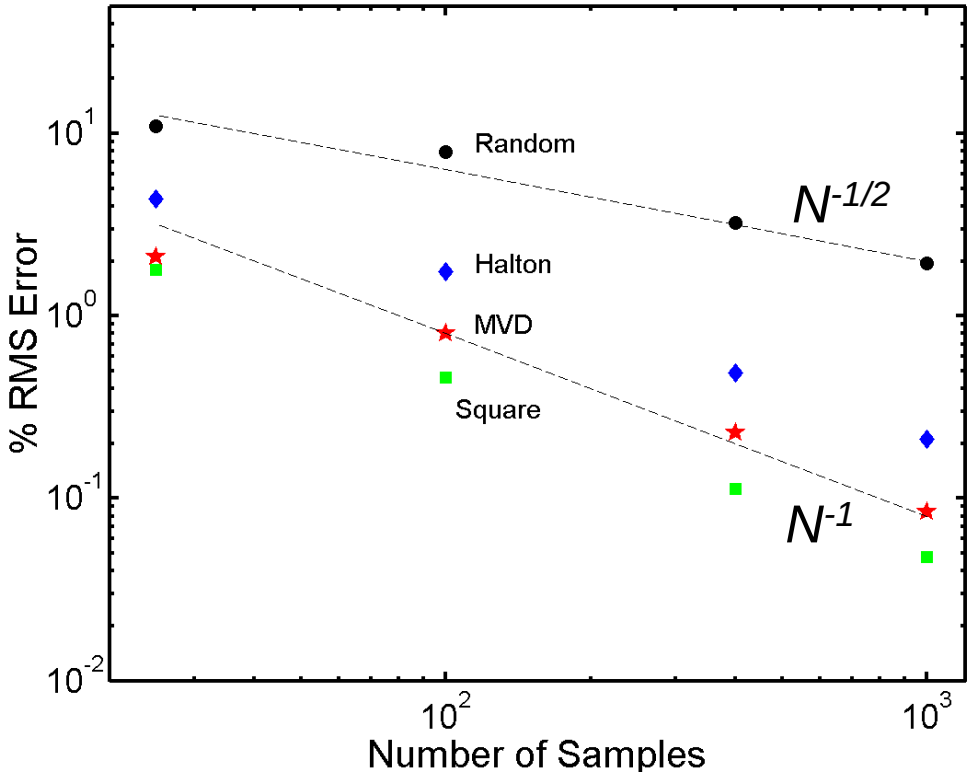


Grid, 0.09%



More Uniform, Higher Accuracy

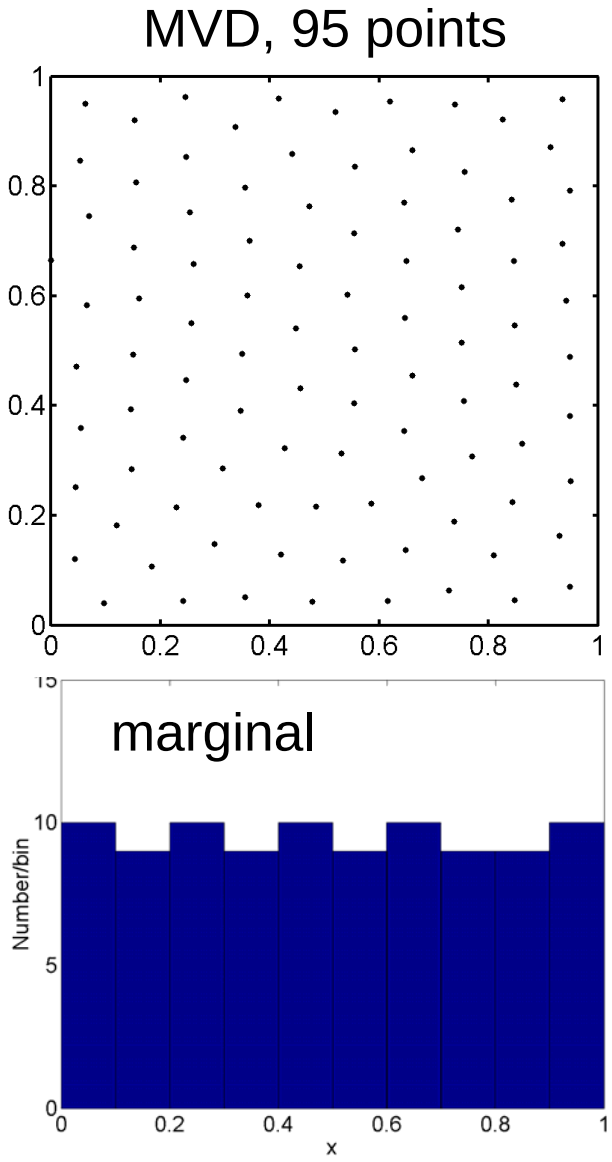
Integration test problem



- RMS error for integral of $\text{func2} = \prod_i \exp\left(-2\left|x_i - x_i^0\right|\right); \quad 0 < x_i^0 < 1$
  - integrate over x_i^0 by using MC, drawing x_i^0 from uniform distribution
  - from worst to best: random, Halton, MVD, square grid
  - lines show $N^{-1/2}$ (expected for MC) and N^{-1} (expected for QMC)

Marginals for MVD points

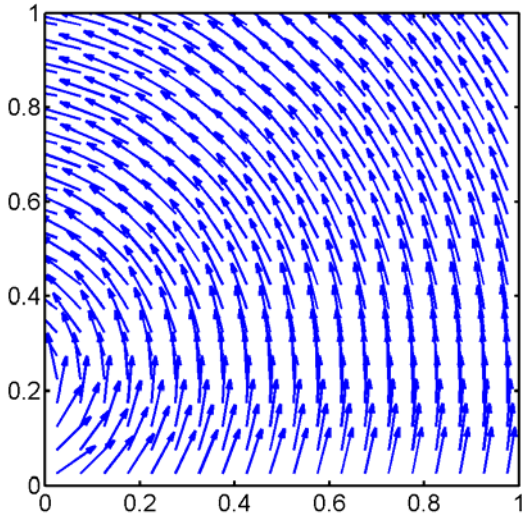
- Sometimes desirable for projections of high dimensional point sets to sample each parameter uniformly
- Latin hypercube sampling designed to achieve this property (for specified number of points)
- Plot shows histogram of 95 MVD samples along x-axis, i.e., marginalized over y direction
- MVD points have relatively uniform marginal distributions



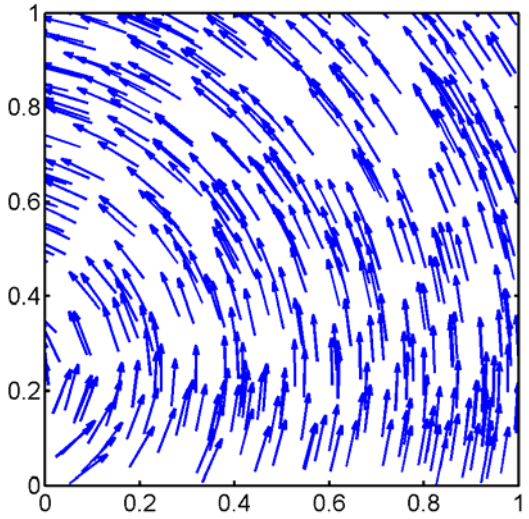
Another use of MVD: visualization of flow field

- Fluid flow often visualized as field of vectors
- Location of vector bases may be chosen as
  - square grid (typical) - regular pattern produces visual artifacts
  - random points - fewer artifacts, but nonuniform placement
  - quasi-random - fewest artifacts and uniform placement

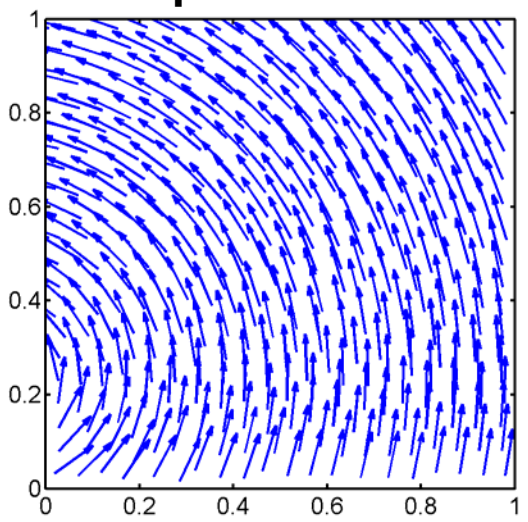
Square grid



Random points



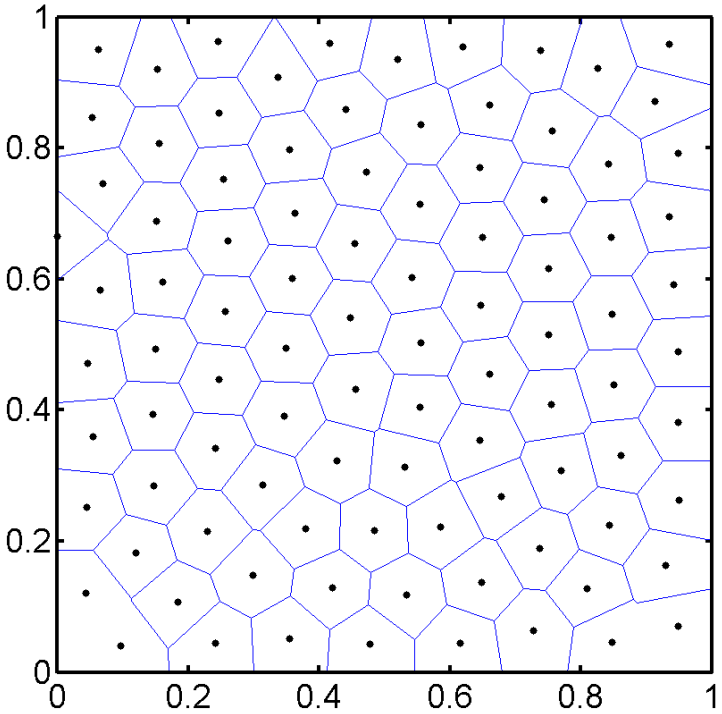
Quasi-random (MVD) point set



Voronoi analysis

- Voronoi diagram (2D)
  - partitions domain into polygons
  - points in *i*th polygon are closest to *i*th generating point, $\boldsymbol{x}_i$
  - boundaries shown are obtained by geometrical construction
- MC technique for Voronoi analysis
  - randomly throw large number of points $\boldsymbol{z}_k$ into region
  - compute distance of each $\boldsymbol{z}_k$ to all generating points $\{\boldsymbol{x}_i\}$
  - $\boldsymbol{z}_k$ belongs to Voronoi region of closest $\boldsymbol{x}_j$
  - can compute A_i, radial moments, identify neighbors, …
- Readily extensible to high dimensions

Voronoi analysis: MVD, 95



Voronoi analysis can improve classic MC

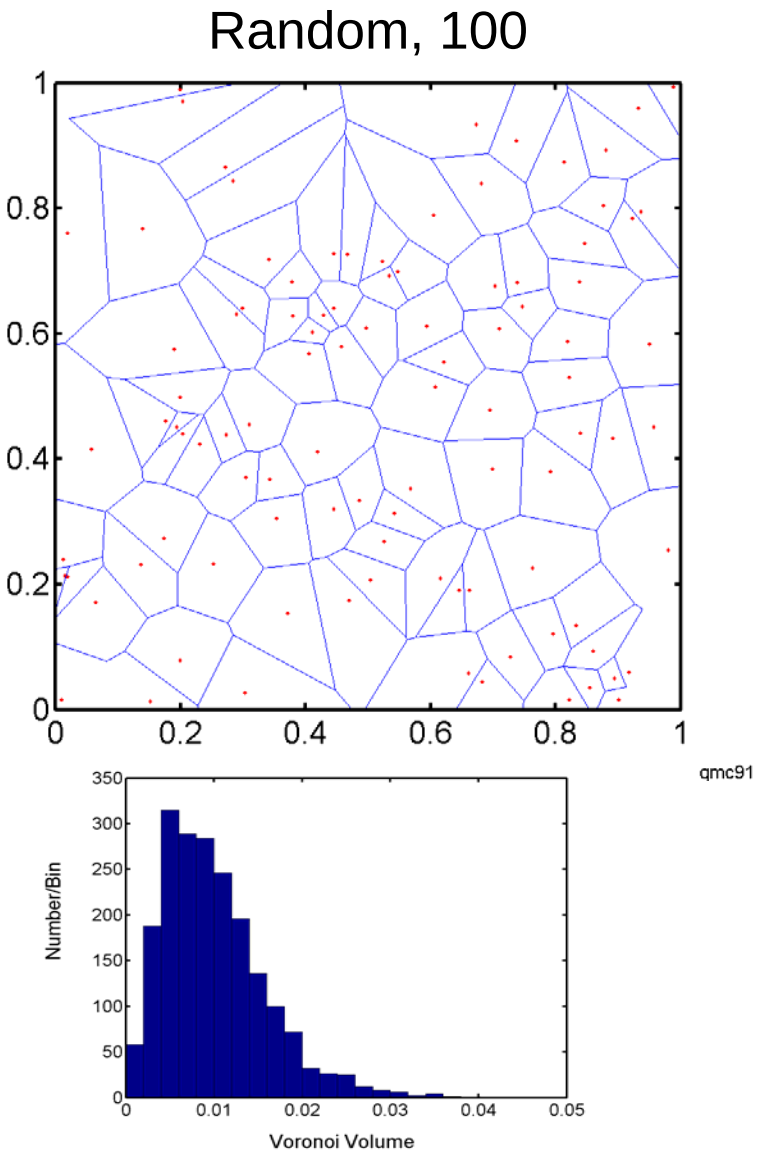
- Standard MC formula

$$\int_R f(\boldsymbol{x})\,d\boldsymbol{x} = \frac{V_R}{n}\sum_{i=1}^{n} f(\boldsymbol{x}_i)$$

- Instead, use weighted average

$$\int_R f(\boldsymbol{x})\,d\boldsymbol{x} = \sum_{i=1}^{n} f(\boldsymbol{x}_i)V_i$$

  - where V_i is the volume of Voronoi region for ith point; Riemann integr.
- Accuracy of integral estimate dramatically improved in 2D:
  - factor of 6.3 for $N = 100$ (func2)
  - factor of > 20 for $N = 1000$ (func2)
- Suitable for adaptive sampling
- Less useful in high dimensions (?)



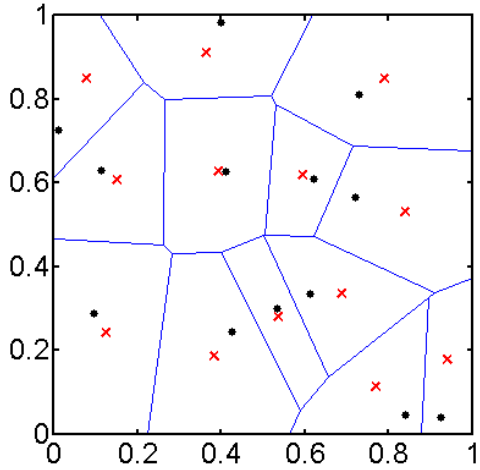
Centroidal Voronoi Tessellation

- Plot shows 13 random points (·) and the centroids of their Voronoi regions (×)
- A point set is called a Centroidal Voronoi Tessellation (CVT) when the generating points $\mathbf{z}^j$ coincide with the centroids their Voronoi regions; a CVT minimizes

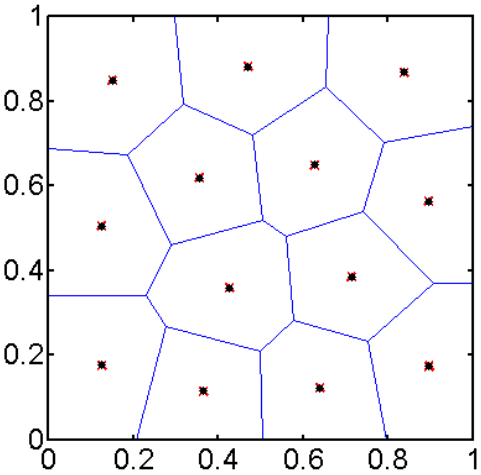
$$\sum_j \int_{V_j} \left|\mathbf{z}^j - \mathbf{x}\right|^2 d\mathbf{x}$$

- Algorithm (McQueen)
  - start with arbitrary set of generating points
  - perform Voronoi analysis using MC algorithm
  - move each generating point to its Voronoi centroid
  - iterate lasts two steps until convergence
- Final CVT points uniformly distributed

Start with random points



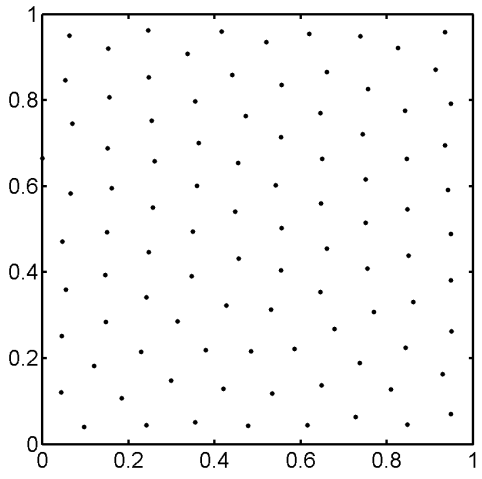
Final CVT point set



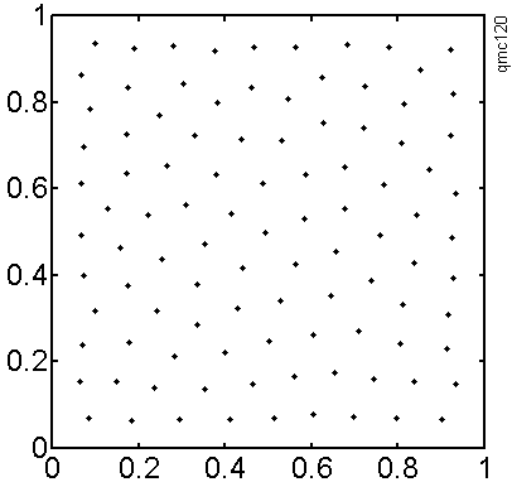
Comparison of methods

- Preceding three algorithms provide uniformly-space points, have essentially equivalent patterns, and are useful for QMC
  - Minimum Visual Discrepancy (MVD)
  - Repulsive Particle Model (RPM)
  - Centroidal Voronoi Tessellation (CVT)
- For high dimensions: both CVT and RPM may be useful, RPM likely most efficient

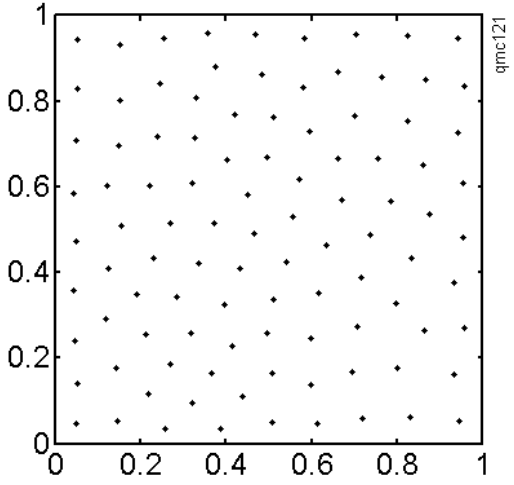
MVD, 95



RPM, 100

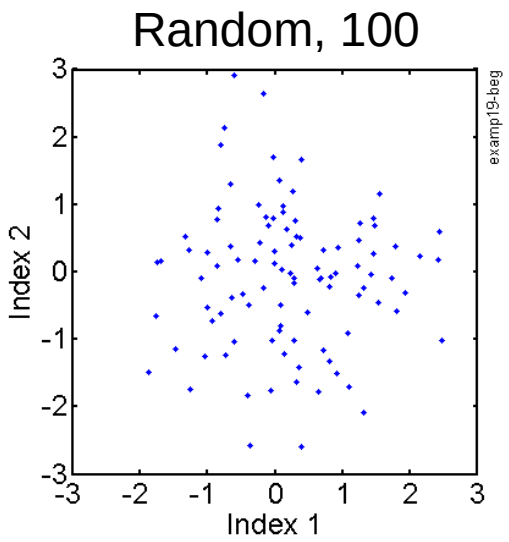


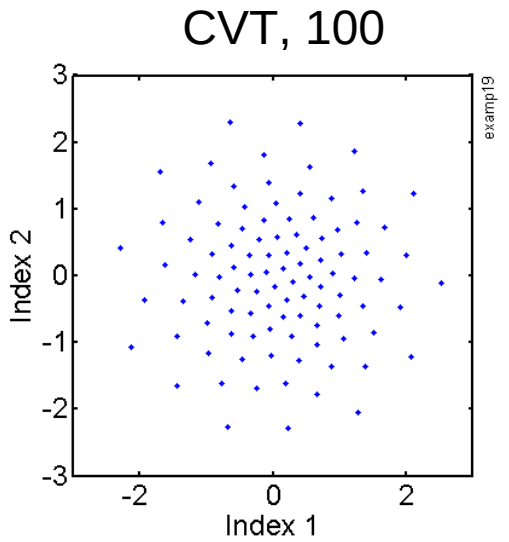
CVT, 100



CVT for multi-variate normal distribution

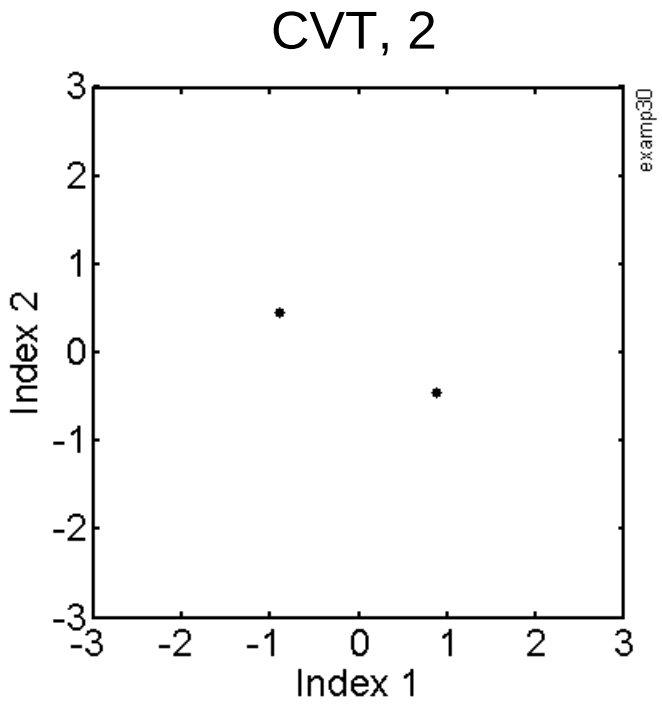
- CVT algorithm works for an arbitrary density function, e.g., a normal distribution
- In above MC algorithm for Voronoi analysis, simply draw random numbers from desired distribution
- Plots show starting random point set and final CVT set
- Radii of points are rescaled to achieve desired average variance along axes





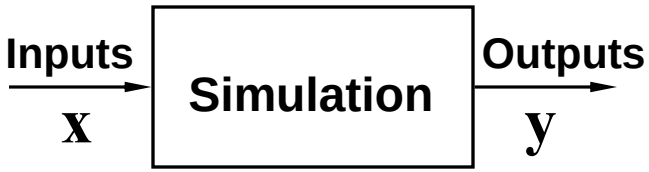
CVT: 2 points in 2 dimensions

- Bi-variate normal distribution is rotationally symmetric
- Symmetry of situation means that the CVT points must be symmetric about origin; at the same radius
- This pattern is unique,
  up to a random rotation
- Both x_1 and x_2 values sampled (with near certainty), but there is a subspace, orthogonal to x_1-x_2 line, whose dependence is not sampled
- Generalizing, the d-D space is under-sampled when $n < d + 1$



Recall context

- Our interest is in characterizing the uncertainty in a simulation output, based on what we know about the uncertainties in the inputs

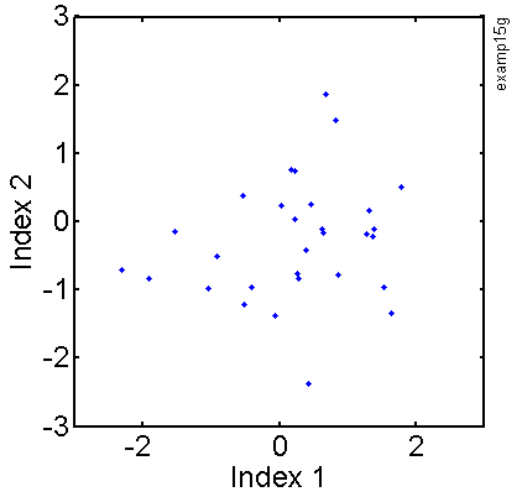


- But, the high cost of running the simulation limits how many samples can be drawn from a parameter distribution to obtain a predictive distribution
- We are often in a situation where the number of points is comparable to number of parameters $(n \approx d)$
- Our goal is to draw a modest number of points from a high-dimensional normal distribution
- Let's explore some of the characteristics of the problem by starting with the example of 2 sample points in 2D

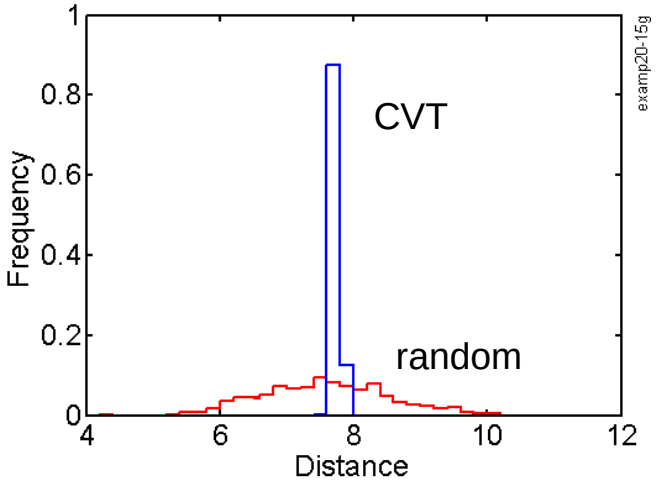
CVT: 30 points in 30 dimensions

- 30 dimensional normal distribution
- Projected onto 2D plane, CVT result doesn't look much different than random sample set
- However, CVT points are uniformly separated in *d*-D, while random points are not

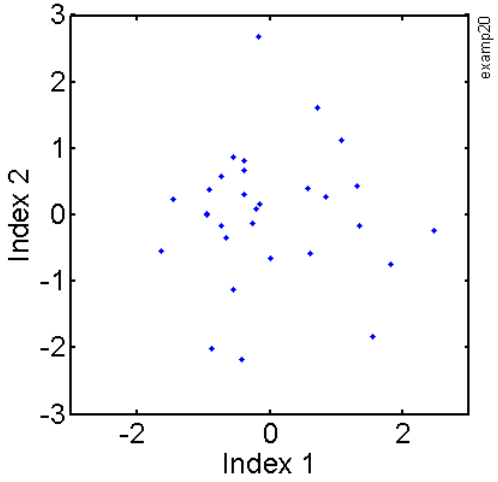
Random, 30



Point separation histogram



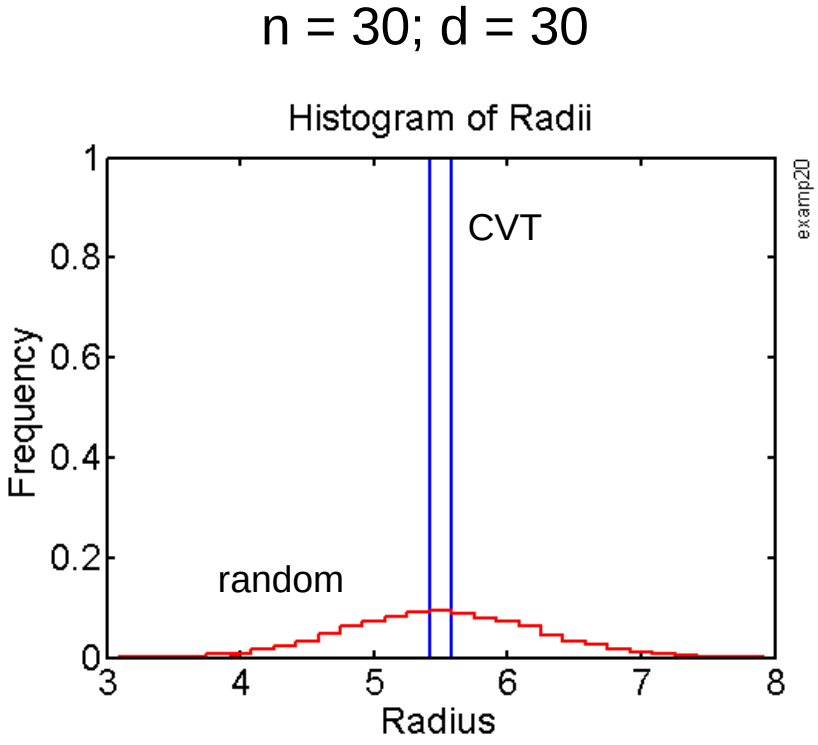
CVT, 30



All points are nearest neighbors!

CVT radial distribution: 30 points in 30D

- As with 2 points in 2D, all 30 CVT points in 30D are at the same radius
  - lie on the surface of a hypersphere
- As seen in last slide, the inter-point distances for CVT are essentially identical
  - regular point pattern (unique?)
- Rotation is only degree of freedom between different realizations of CVT
- One can generate new CVT patterns by randomly rotating an existing one

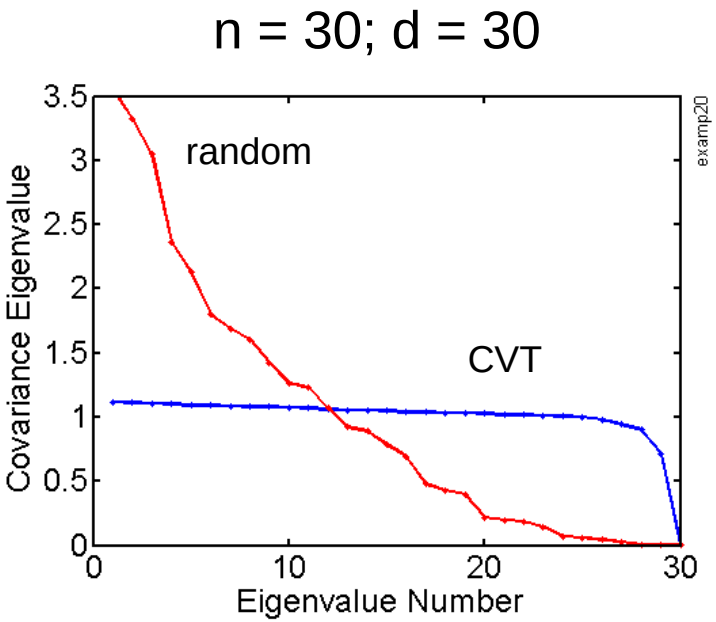


Covariance analysis of point set

- Let $\mathbf{x}^j$ be vector for jth point; point set is represented by matrix

$$\mathbf{X} = (\mathbf{x}^1; \mathbf{x}^2; \mathbf{x}^3; \cdots \mathbf{x}^n)^T$$

- Covariance of point set along the axes is $\mathbf{X}\mathbf{X}^T$
- Eigenanalysis of $\mathbf{X}\mathbf{X}^T$ yields the covariance spectrum
  - the ith eigenvalue is the variance of the points projected onto the ith eigenvector
- Conclude that spectrum for CVT point set is much more uniform than for random set, which is quite variable (the Wishart distribution)
- Last eigenvalue is zero; rank = 29



Linear response model

- Assume outputs of a simulation are linearly related to perturbations in the inputs, $\delta \boldsymbol{y} = \mathbf{S}_{\mathbf{y}}^{\mathrm{T}} \delta \mathbf{x}$
  where $\mathbf{S}_{\mathbf{y}}$ is sensitivity matrix $\partial \mathbf{y} / \partial \mathbf{x}$

- The covariance in the output **y** is

$$\mathbf{C}_{\mathbf{y}} = \mathbf{S}_{\mathbf{y}}^{\mathrm{T}} \mathbf{C}_{\mathbf{x}} \mathbf{S}_{\mathbf{y}}$$

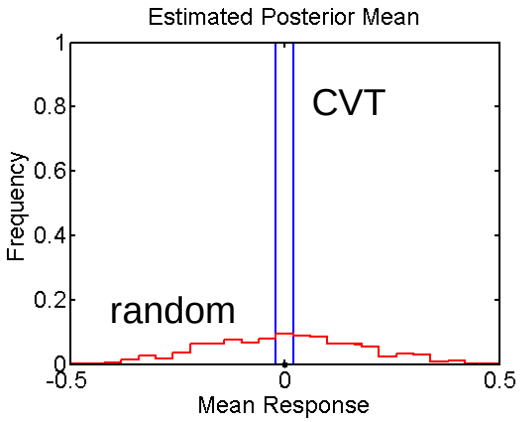
  - when output *y* is a scalar,
    the covariance $\mathbf{C}_{\mathbf{y}}$ is a scalar (variance),
    and $\mathbf{S}_{\mathbf{y}}$ is a vector

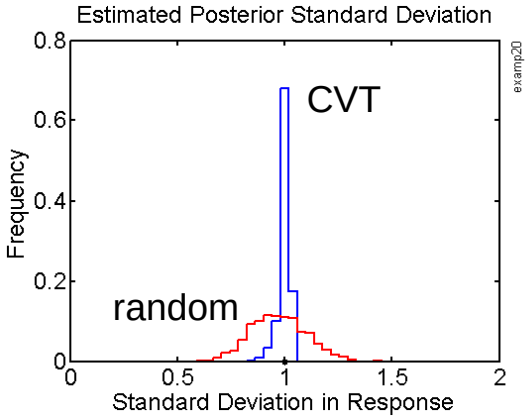
- If linear model is sufficient and one knows the sensitivity matrix, then predictive distribution is easily characterized
- However, for large simulations, the sensitivity matrix is often unknown

Test single point set using random sensitivities

- Assume linear model, $\delta \boldsymbol{y} = \mathbf{S}_{\mathbf{y}}^{\mathrm{T}} \delta \mathbf{x}$ where $\mathbf{S}_{\mathbf{y}}$ is sensitivity matrix $\partial \mathrm{y}/\partial \mathbf{x}$
- Test predictive response of a single sample set for an ensemble of random sensitivity vectors $\mathbf{S}_{\mathbf{y}}$:
  - $\mathbf{S}_{\mathbf{y}} = N_d(0,1)$; mean($\mathbf{S}_{\mathbf{y}}$) = 0 , var($\mathbf{S}_{\mathbf{y}}$) = 1
  - assume input **x** distribution is uncorrelated, unit-variance *d*–dimen. normal distribution, $N_d(0,1)$; $\mathbf{C}_{\mathbf{x}} = \mathbf{1}$
  - then expect: mean(**y**) = 0 , var(**y**) = 1
- Plots show CVT (blue) predicts mean and standard deviation of predictive distribution more accurately than random Monte Carlo (red)

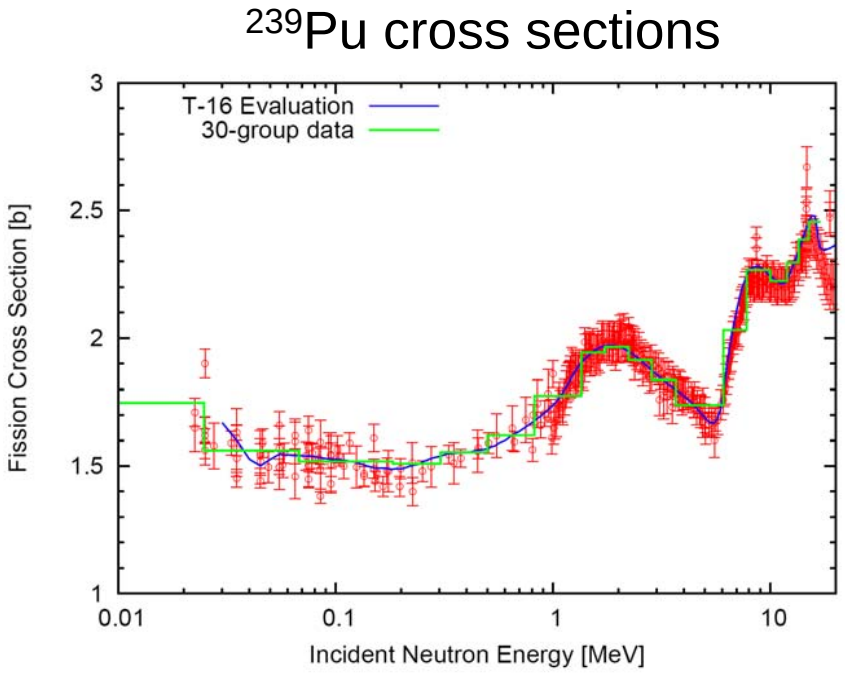
n = 30; d = 30





Neutron cross sections

- Plot shows
  - measured fission cross sections for neutrons on ^{239}Pu; red data points
  - inferred cross sections; blue line
  - weighted average in 30 energy bins (groups); green histogram
- PARITSN code simulates neutron transport based on multigroup, discrete-ordinates method
  - uses 30 energy bins
  - calculates criticality for specified configuration of fissile-material
  - establish dependence of criticality experiment to cross sections

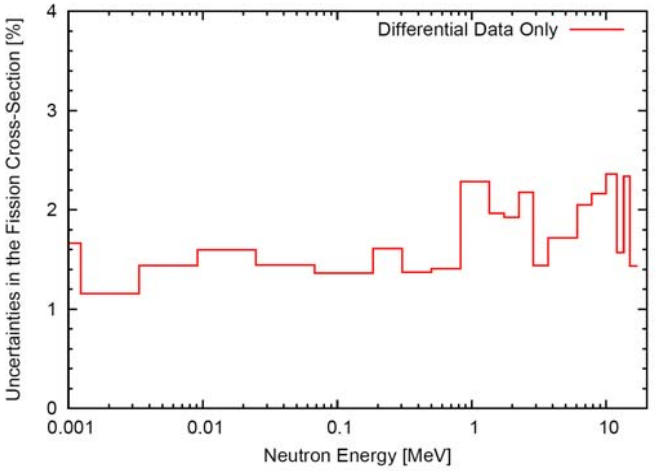


Neutron cross sections - uncertainties

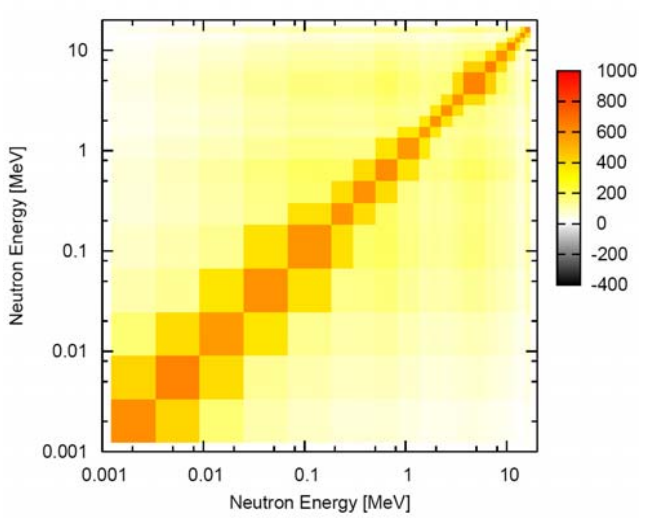
- Analysis of measured cross sections yields a set of evaluated cross sections

- Uncertainties in evaluated cross sections are ~ 1.4-2.4 %
- Covariance matrix important
- Strong positive correlations caused by normalization uncertainties in each experiment

standard error in cross sections



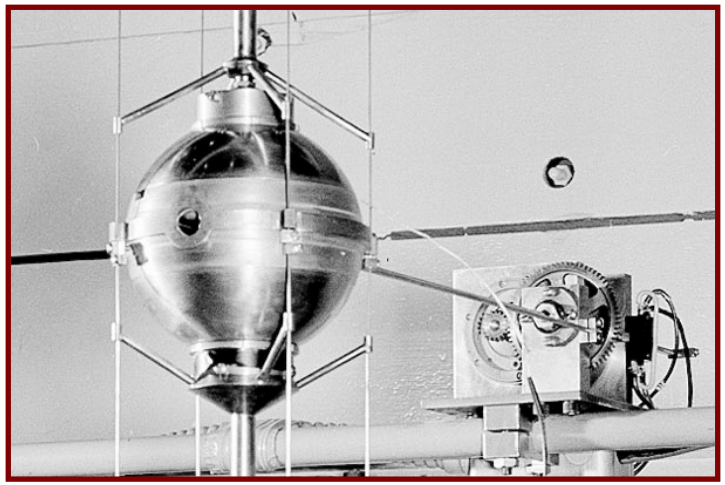
correlation matrix



JEZEBEL – criticality experiment

- JEZEBEL experiment (1950-60)
  - fissile material ^{239}Pu
  - measure neutron multiplication as function of separation of two hemispheres of material
  - summarize criticality with neutron multiplication factor, $k_{eff} = 0.9980 \pm 0.0019$
  - very accurate measurement

JEZEBEL set up

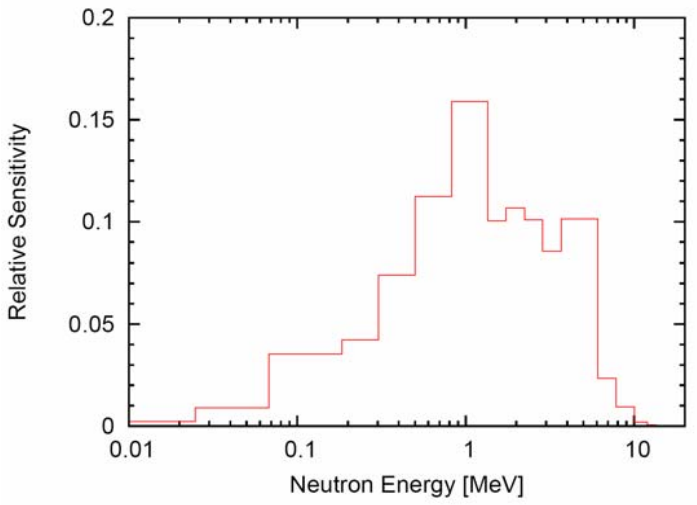


- Our goal – use highly accurate JEZEBEL measurement to improve our knowledge of ^{239}Pu cross sections

JEZEBEL – sensitivity analysis

- PARITSN code relates k_{eff} to neutron cross sections
- Sensitivity of k_{eff} to cross sections found by perturbing cross section in each energy bin by 1% and observing increase in k_{eff}
- Observe that 1% increase in all cross sections results in 1% increase in k_{eff} , as expected

k_{eff} sensitivity to cross sections



Bayesian update

- For data linearly related to the parameters, the Bayesian (aka Kalman) update is

$$\mathbf{C}_1^{-1}\mathbf{x}_1 = \mathbf{C}_0^{-1}\mathbf{x}_0 + \mathbf{S}_\mathbf{y}^{\mathrm{T}}\mathbf{C}_\mathbf{y}^{-1}\mathbf{S}_\mathbf{y}(\mathbf{y}-\mathbf{y}_0)$$

$$\mathbf{C}_1^{-1} = \mathbf{C}_0^{-1} + \mathbf{S}_\mathbf{y}^{\mathrm{T}}\mathbf{C}_\mathbf{y}^{-1}\mathbf{S}_\mathbf{y}$$

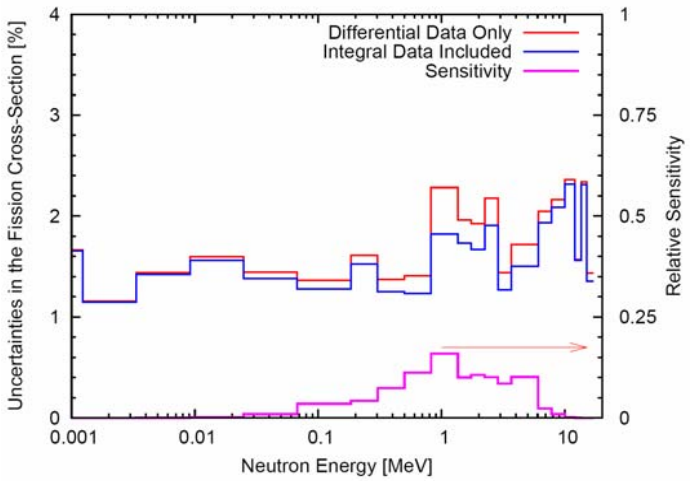
  - $\mathbf{x}_0$ and $\mathbf{x}_1$ are parameter vectors before and after update
  - $\mathbf{C}_0$ and $\mathbf{C}_1$ are their covariance matrices
  - $\mathbf{y}$ and $\mathbf{C}_\mathbf{y}$ are the measured data vector and its covariance
  - $\mathbf{y}_0$ is the value of $\mathbf{y}$ for $\mathbf{x}_0$
  - $\mathbf{S}_\mathbf{y}$ is the matrix of the sensitivity of $\mathbf{y}$ to $\mathbf{x}$; $\partial\mathbf{y}/\partial\mathbf{x}$

- For the JEZEBEL case, $\mathbf{y}$ is a scalar (k_{eff}), $\mathbf{C}_\mathbf{y}$ is a scalar (variance), and $\mathbf{S}_\mathbf{y}$ is a vector

Updated cross sections

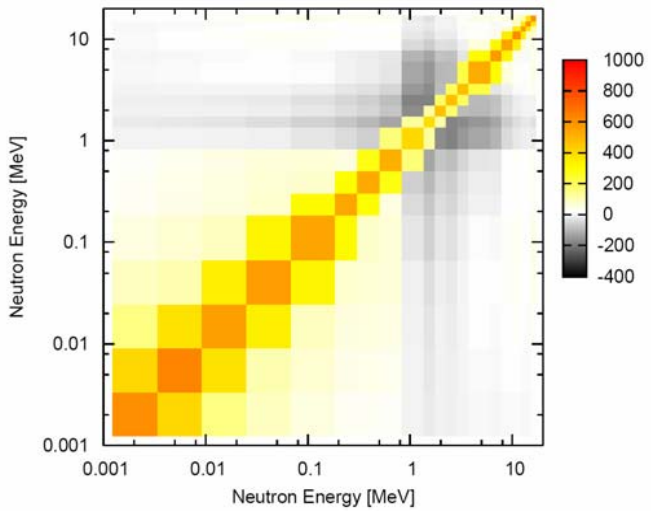
- Plot shows uncertainties in cross sections before and after using JEZEBEL measurement
- Modest reduction in uncertainties; follows energy dependence of sensitivity

standard error in cross sections



- Correlation matrix is significantly altered
- Strong negative correlations introduced by integral constraint of matching JEZEBEL's k_{eff}

correlation matrix

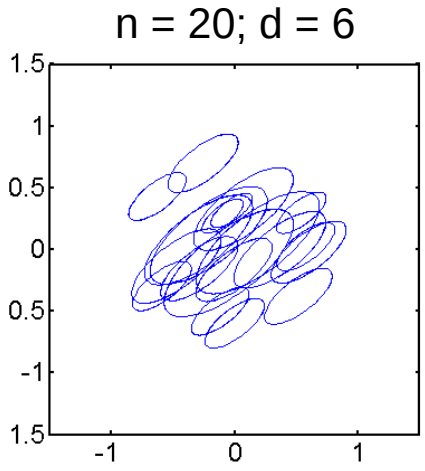


Uncertainty in subsequent simulations

- Intend to use updated cross sections in new calculations, with expectation that integral constraint will reduce uncertainties
- Next, demonstrate use of Monte Carlo sampling for estimating the uncertainty in new predictions; use random MC or qMC?
- Demonstrate usefulness of quasi-MC in form of CVT point sets by "predicting" k_{eff} measured in JEZEBEL
  - for this demo, assume linear model with known sensitivity vector
  - under these assumptions, we can calculate exact answer and compare to MC-style sampling to obtain predictive distribution
- For new physical scenario, we would not have sensitivity vector and we would have to do full simulation calculation, so only a modest number of function evaluations can be done

Rotation matrix in high dimensions

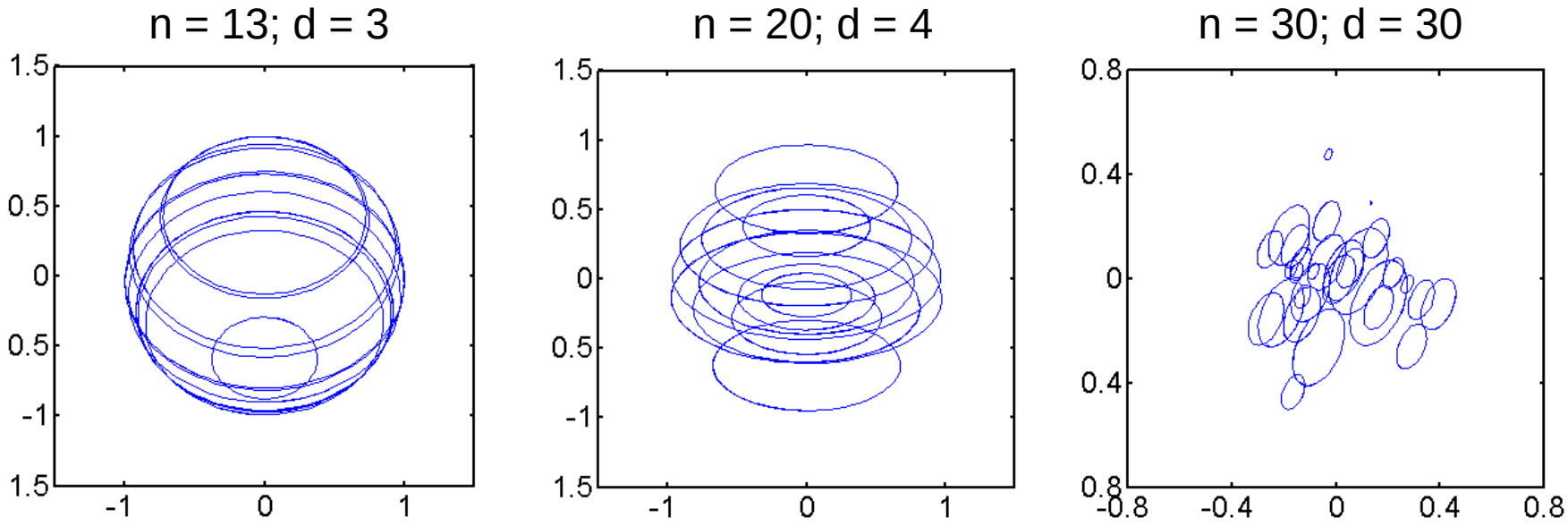
- Given unit vectors **a** and **b**, want rotation matrix that map **a** into **b**
- Algorithm (thanks to Mike Fitzpatrick)
  - for matrix $\mathbf{M}_1 = (\mathbf{a}; \mathbf{b})^{\mathrm{T}}$
  - Singular Value Decomposition (SVD) of:
    $$\mathbf{M}_1 = \mathbf{U}\Sigma\mathbf{V}^{\mathrm{T}}$$
  - the bases of the subspace orthogonal to **a** and **b** are given by singular vectors in **U**, except for first two
  - then, do SVD on outer product matrix $\mathbf{M}_2 = \mathbf{B}\mathbf{A}^{\mathrm{T}}$
    $$\mathbf{A} = (\mathbf{a}; \mathbf{u}^3; \mathbf{u}^4; \cdots \mathbf{u}^{\mathrm{d}})^{\mathrm{T}} \; ; \quad \mathbf{B} = (\mathbf{b}; \mathbf{u}^3; \mathbf{u}^4; \cdots \mathbf{u}^{\mathrm{d}})^{\mathrm{T}}$$
  - rotation matrix is $\mathbf{R} = \mathbf{U}_2 \mathbf{D} \mathbf{V}_2^{\mathrm{T}}$ where **D** is identity matrix, except $[\mathbf{D}]_{d\,d} = \det(\mathbf{U}_2 \mathbf{V}_2^{\mathrm{T}})$



- Random rotations – randomly choose directions of **a** and **b**
  - simple algorithm: randomly draw vector from normal distribution and normalize it to unit length

Examples of rotations in high dimensions

- Random rotations is various dimensions
- All points have unit radius (on surface of unit hypersphere)



Sampling from correlated normal distribution

- Want to draw samples from multi-variate normal distribution with known covariance $\mathbf{C}_\mathbf{x}$
- Important to include correlations among uncertainties, i.e., off-diagonal elements
- Algorithm:
  - perform eigenanalysis of covariance matrix of *d* dimensions

$$\mathbf{C}_\mathbf{x} = \mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^\mathrm{T}$$

    where **U** is orthognal matrix of eigenvectors and
    $\boldsymbol{\Lambda}$ is the diagonal matrix of eigenvalues
  - draw *d* samples from unit variance normal distribution, ξ_i
  - scale this vector by $\lambda_i^{1/2}$
  - transform vector into parameter space using the eigenvector matrix
  - to summarize: $\mathbf{x} = \mathbf{U}\boldsymbol{\Lambda}^{1/2}\boldsymbol{\xi}$

Accuracy of predicted k_{eff} and its uncertainty

- Check accuracy of predicted mean and standard deviation based on 30 samples; CVT vs. random sample sets
  - exact value from known sensitivity and linear model
- Conclude – CVT is more accurate here than random sampling

Results from 1000 sample sets; 'rot' indicates single sample set randomly rotated to achieve each new one

	est. mean k_{eff}		est. std. dev. k_{eff}	
	avg.	rms dev.	avg.	rms dev.
random	0.99788	0.00037	0.00191	0.00028
random-rot	0.99824	0.00010	0.00218	0.00010
CVT-rot	0.99796	0.00001	0.00197	0.00002
exact-linear	0.99796	-	0.00195	-

Summary

- CVT sampling is useful in predictive sampling for obtaining higher accuracy for a limited number of simulations
- CVT and repulsive particle model may be used to generate QMC point sets
  - particularly useful for modest number of points

Future work

- Need a way to estimate accuracy of results, a problem common to all QMC approaches
- Important to learn to cope with under-sampling ($n < d + 1$)
  - a parameter subspace dependence is not sampled so some aspects of uncertainty in input parameters may be missed
  - satisfactory solution requires careful thought, taking into account what is known about the influence of input parameters on the simulation output
- Sequential generation of point sets
  - add additional points, keeping previous points fixed
- Adaptive sampling – improve estimate by importance sampling, i.e., increasing density of points in selected regions
- Employ advanced analysis of outputs produced by input samples
  - weighted Monte Carlo
  - characterize output response as function of inputs

Future work

- Semi-random-quasi-Monte Carlo
  - partially randomize quasi-Monte Carlo
  - strike balance between random and deterministic patterns
  - try to achieve the benefits of each approach

Bibliography

- K. M. Hanson, "Quasi-Monte Carlo: halftoning in high dimensions?," *Proc. SPIE* **5016**, pp. 161-172 (2003)
- P. Li and J. P. Allebach, "Look-up-table based halftoning algorithm," *IEEE Trans. Image Proc.* **9**, pp. 1593-1603 (2000)
- H. Niederreiter, *Random Number Generation and Quasi-Monte Carlo Methods* (SIAM, 1992)
- Q. Du, V. Faber, and M. Gunzburger, "Centroidal Voronoi tessellations: applications and algorithms," *SIAM Review* **41**, 637-676 (1999)
- T. Kawano et al., "Uncertainty quantification for applications of ^{239}Pu fission cross sections using a Monte Carlo technique," to appear in *Nucl. Sci. Eng.* (2005)

This presentation available at http://www.lanl.gov/home/kmh/