Gibbs Sampler: Diagnostics

The basic version of full scan Gibbs sampler is based on updating component by component *in a given coordinate basis.*

Consequence: the performance *depends on the selected coordinate basis.*

It is sometimes possible to improve the performance by redefining the coordinate basis.

Demonstration of the coordinate effect

Probability density

$$\pi(x) = \exp\left(-\frac{1}{2\sigma^2}\|A\mathbf{x} - \mathbf{y}\|^2\right) + \text{ bounds, } 0 \le x_1, x_2 \le 10,$$

where

$$A = \begin{bmatrix} 1 & \\ & 0.01 \end{bmatrix} \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix},$$

and

$$\sigma = 0.01.$$

Sample with Gibbs sampler, using different values of θ.

```
% Full scan Gibbs sampler. The program
% produces a sample of size nsample from
% the probability distribution
%
% Pi(x) ~ exp(-0.5*||A*x - b||^2), lb< x < ub
%
% Input: A -      array of size (m,n)
%        b -      column vector of length m
%        lb, ub - column vectors of length n
%        x_init - column vector of length n
%        nsample - number of samples
% Output: X - array of size (n,nsample)
```

```
% Defining the matrix and the rhs

 th = pi/3;

 lratio = 0.01;
 A = diag([1,lratio])*[cos(th) -sin(th); sin(th) cos(th)];
 xdata = [1;1];
 b = A*xdata;
 std = 0.01;
 A = (1/std)*A;
 b = (1/std)*b;

% Upper and lower bounds and initial point

 lb = [0;0];
 ub = [10;10];
 x_init = [5;5];
```

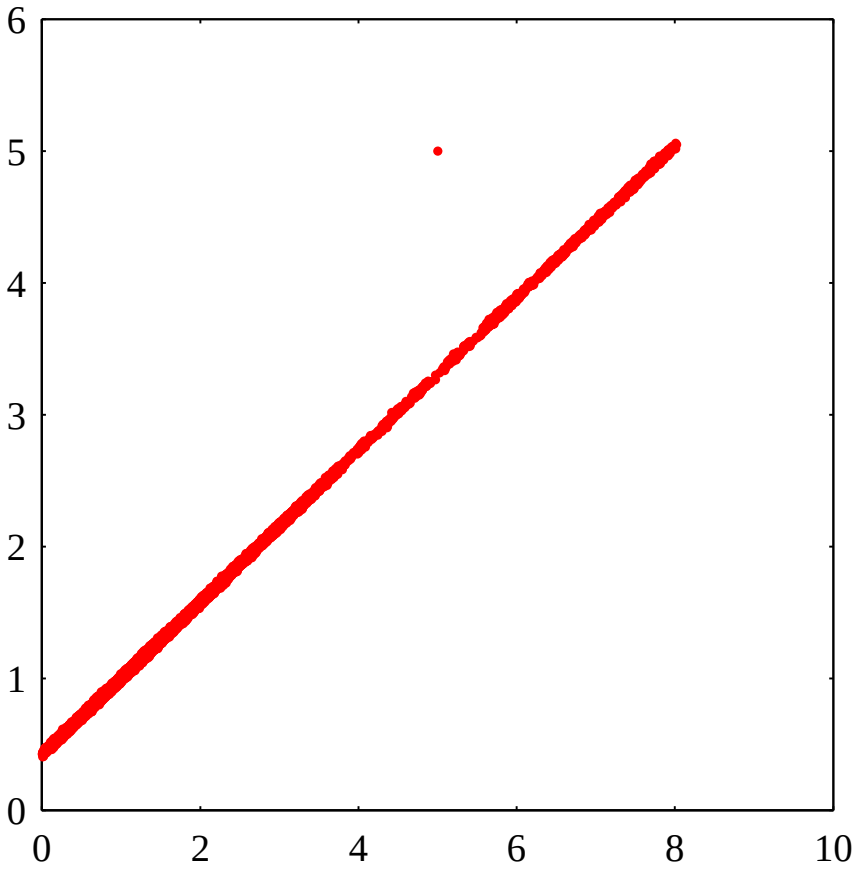
```
 nsample = 20000;

% Checking the inputs

Iaux = find(lb>=ub); if length(Iaux) > 0
   display('Lower bound not less than upper bound; check input!')
   return
end

Iaux = find(x_init>=ub); if length(Iaux) > 0
   display('Initial point not below upper bound; check input!')
end

Iaux = find(x_init<=lb); if length(Iaux) > 0
   display('Initial point not above lower bound; check input!')
end
```

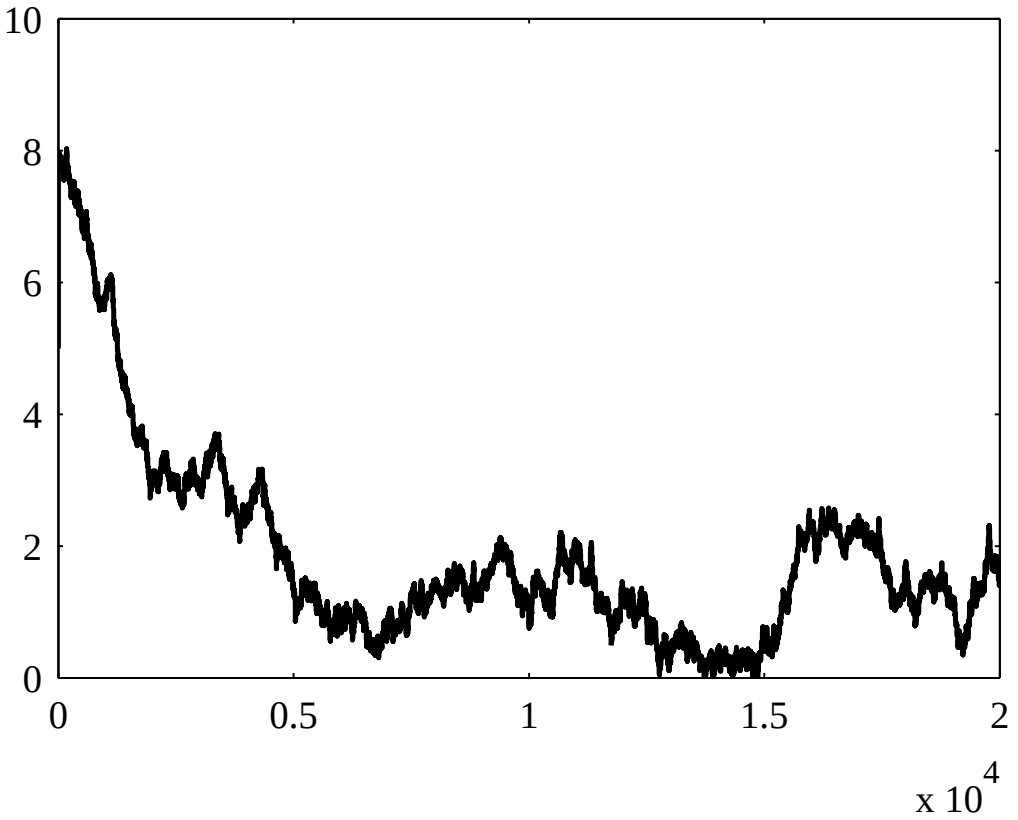
```
% Gibbs sampler: Full scan, no SVD, no model reduction

 ny = length(x_init);

 bin = 100;       % Number of division points in random draws
 cutoff = 1e-3;   % Cutoff level of the pdf
 tiny = 1e-8;     % Minimum length of reasonable interval

 X = zeros(ny,nsample);
 x = x_init;
 X(:,1) = x;
```

```
for n = 2:nsample

   % Updating component by component the vector y

   for j = 1:ny
       a = A(:,j);
       Ij = setdiff([1:ny],j);
       Aj = A(:,Ij);
       yj = x(Ij);
       dj = sqrt(a'*a);
       y0 = a'*(b-Aj*yj)/dj;
       ymin = dj*lb(j);
       ymax = dj*ub(j);

       if ymax - ymin < tiny
      % the effective interval is reduced
          x(j) = ymin/dj;
```

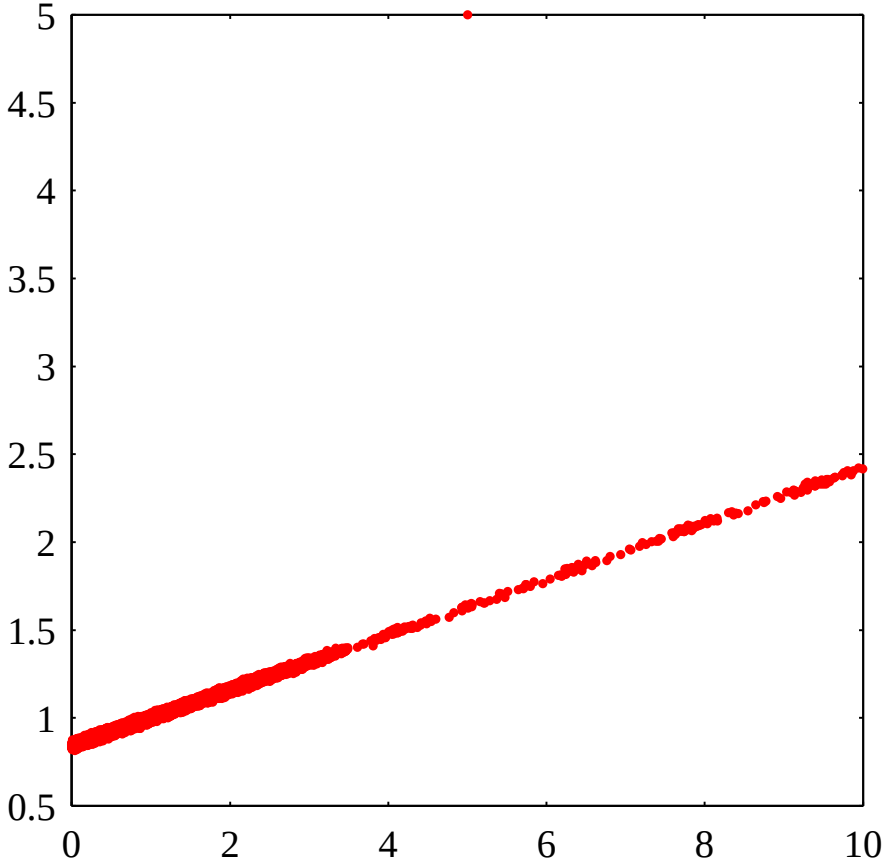
```
else
   if y0 > ymax
      f0 = (ymax-y0)^2;
      ystar = y0 - sqrt(f0+2*log(1/cutoff));
      ymin = max(ymin,ystar);
   elseif y0<ymin
      f0 = (ymin-y0)^2;
      ystar = y0 + sqrt(f0+2*log(1/cutoff));
      ymax = min(ymax,ystar);
   else
      f0 = 0;
      ymin = max(ymin,y0 - sqrt(2*log(1/cutoff)));
      ymax = min(ymax,y0 + sqrt(2*log(1/cutoff)));
   end

   t = linspace(ymin,ymax,bin);
   p = exp(-0.5*((t-y0).^2 - f0));
   phi = cumsum(p);
```

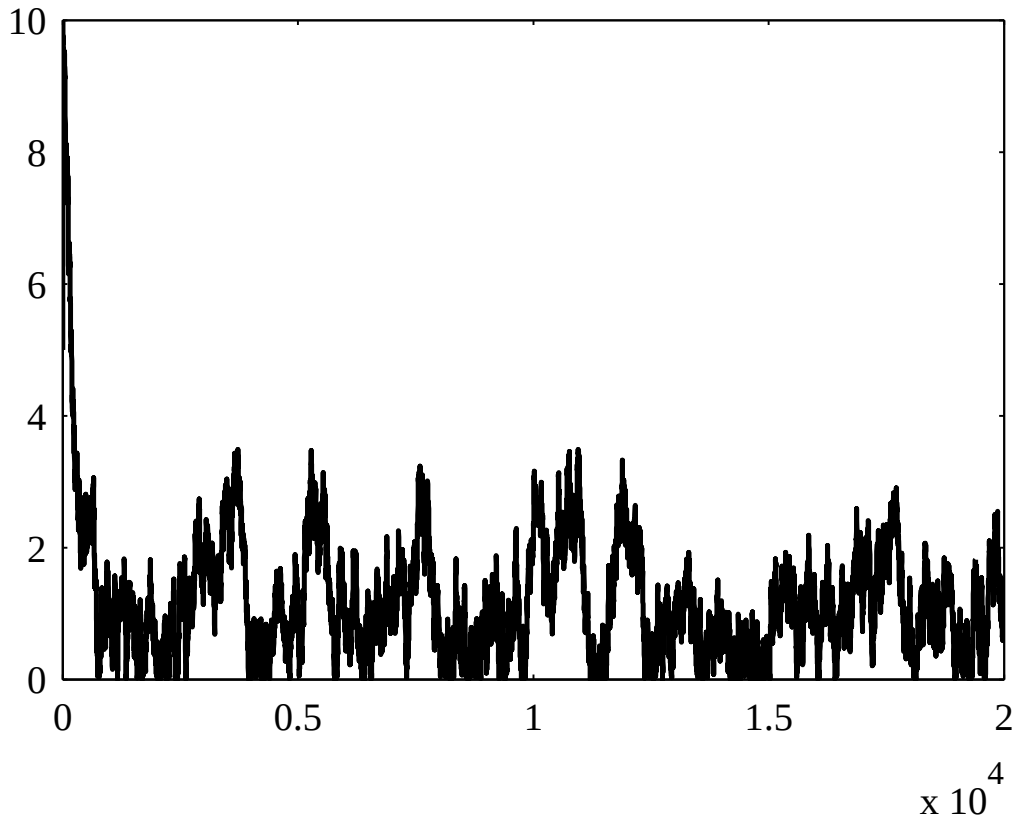
```
            xi = phi(bin)*rand;
            jj = min(find(phi>xi));
            x(j) = t(jj)/dj;
        end
    end

    X(:,n) = x;
end
```



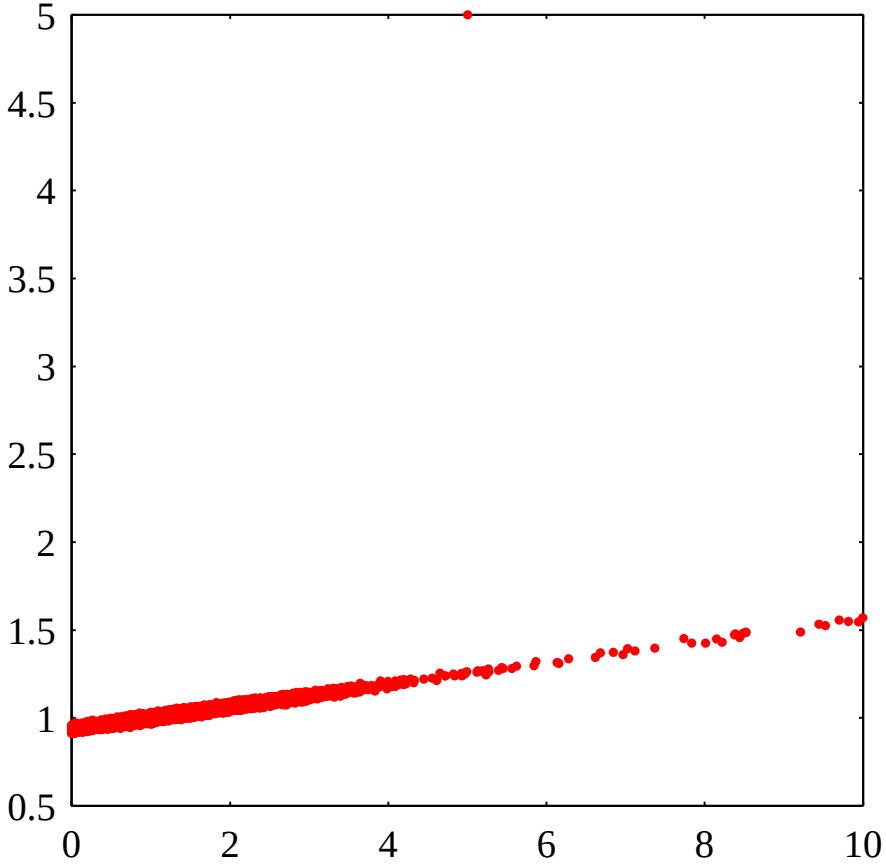


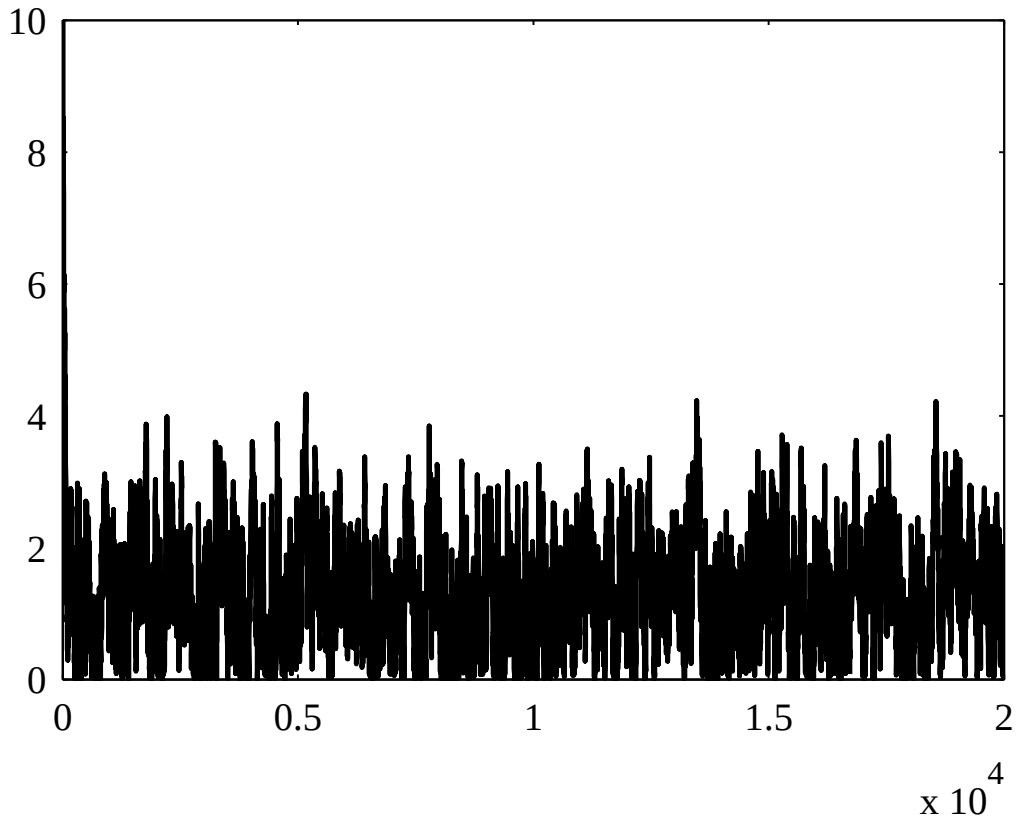
$$\theta = \frac{\pi}{3}.$$



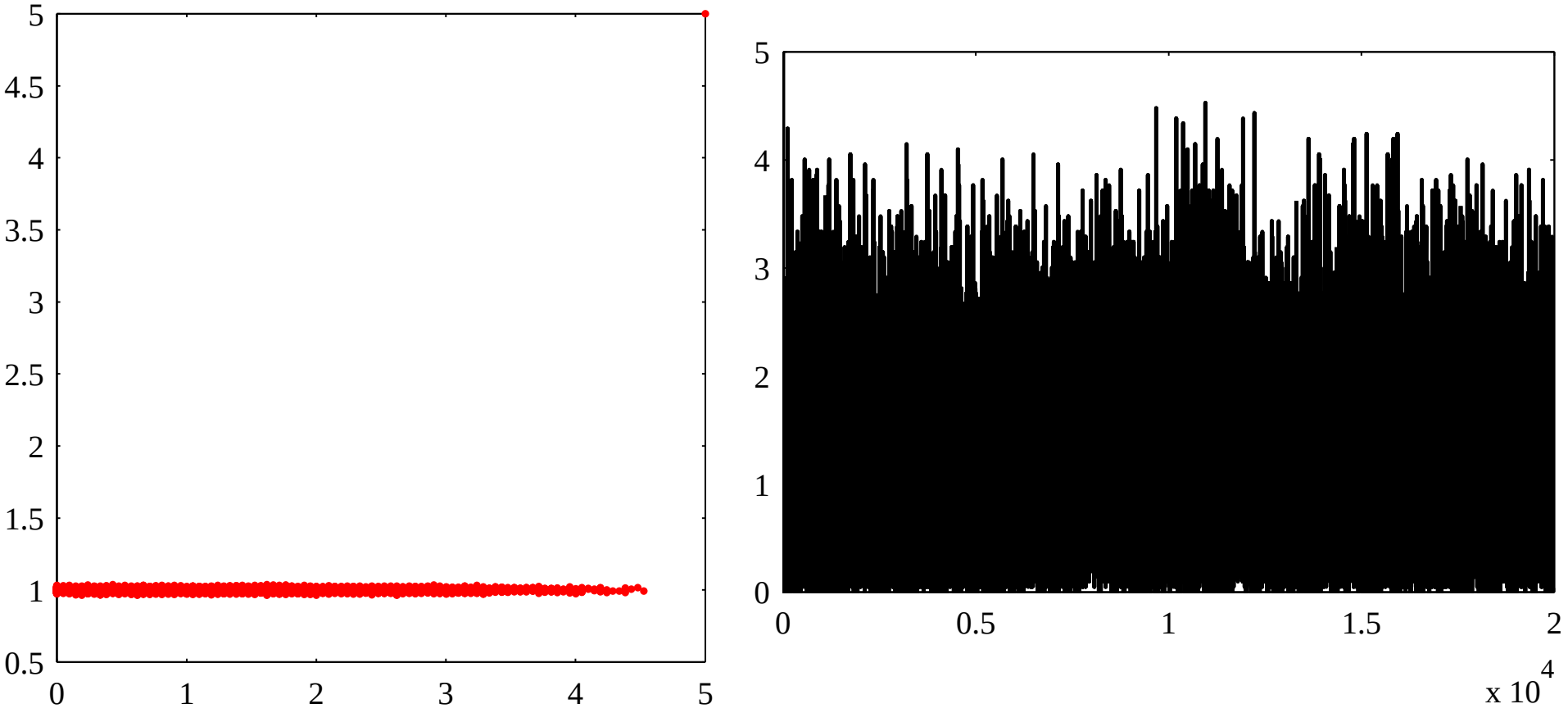


$$\theta = 0.9\frac{\pi}{2}.$$





$$\theta = 0.95\frac{\pi}{2}.$$



$$\theta = \frac{\pi}{2} \text{ (perfect sampling).}$$

CONCLUSION: If possible, try to find a coordinate system that minimizes the correlation between the components.

NOTE: In coordinate transformations, probability densities transform *as densities, not as scalar functions!*

(In the language of differential geometry, probability densities are n–forms, not 0–forms.)

Coordinate transformations

Assume that we have new coordinates,

$$\begin{aligned} y_1 &= \varphi_1(x_1, x_2, \ldots, x_n), \\ y_2 &= \varphi_2(x_1, x_2, \ldots, x_n), \\ &\vdots \\ y_n &= \varphi_n(x_1, x_2, \ldots, x_n), \end{aligned}$$

or briefly,

$$\mathbf{y} = \Phi(\mathbf{x}),$$

where Φ is a diffeomorphism, i.e., invertible, and both Φ and Φ^{-1} are differentiable.

Now

$$\begin{aligned} \mathrm{P}\{\mathbf{y} \in B\} &= \mathrm{P}\{\Phi(\mathbf{x}) \in B\} \\ &= \mathrm{P}\{\mathbf{x} \in \Phi^{-1}(B)\} \\ &= \int_{\Phi^{-1}(B)} \pi(\mathbf{x}) d\mathbf{x}. \end{aligned}$$

Changing variables in the integral:

$$d\mathbf{x} = \left|\det(D(\Phi^{-1})(\mathbf{y}))\right| d\mathbf{y} = \frac{1}{|\det(D\Phi(\mathbf{x}))|} d\mathbf{y},$$

so

$$\mathrm{P}\{\mathbf{y} \in B\} = \int_B \pi(\Phi^{-1}(\mathbf{y})) \left(\frac{1}{|\det(D\Phi(\mathbf{x}))|} \right) \Big|_{x=\Phi^{-1}(\mathbf{y})} d\mathbf{y}.$$

CONCLUSION: With respect to the new coordinates,

$$\pi(\mathbf{y}) = \pi(\Phi^{-1}(\mathbf{y}))\left(\frac{1}{|\det(D\Phi(\mathbf{x}))|}\right)\Bigg|_{x=\Phi^{-1}(\mathbf{y})}$$

(Notice the dangerous notation: π has two different meanings.)

Shorthand notation: if $\pi(\mathbf{x}) = p(\mathbf{x})$, then

$$p(\mathbf{x})d\mathbf{x} = \underbrace{p(\Phi^{-1}(\mathbf{y}))\left|\det\left(\frac{d\mathbf{x}}{d\mathbf{y}}\right)\right|}_{=\pi(\mathbf{y})} d\mathbf{y}.$$

Linear transformations

Assume that

$$\mathbf{y} = R\mathbf{x}, \quad R \in \mathbb{R}^{n\times n} \text{ invertible.}$$

Then,

$$d\mathbf{y} = |\det(R)| d\mathbf{x},$$

and consequently, if $\pi(\mathbf{x}) = p(\mathbf{x})$, we have

$$\pi(\mathbf{y}) = p(R^{-1}\mathbf{y}) \frac{1}{|\det(R)|}.$$

How to find good coordinates?

Consider a Gaussian density of the form

$$\pi(\mathbf{x}) \propto \exp\left(-\frac{1}{2}\|A\mathbf{x}-\mathbf{b}\|^2\right),$$

with a possible bound constraint

$$C\mathbf{x} \geq \mathbf{r}.$$

We seek to solve a system of basis vectors in which the components of $\mathbf{x}$ are as uncorrelated as possible.

Singular value decomposition

Let $A \in \mathbb{R}^{m \times n}$. SVD of A is

$$A = UDV^{\mathrm{T}},$$

where

$$U \in \mathbb{R}^{m \times m} \text{ and } V \in \mathbb{R}^{n \times n}$$

are orthogonal, i.e.,

$$U^{\mathrm{T}}U = UU^{\mathrm{T}} = I, \quad V^{\mathrm{T}}V = VV^{\mathrm{T}} = I,$$

and

$$D \in \mathbb{R}^{m \times n}$$

is diagonal.

The diagonal elements of D are non-negative, and ordered in decreasing order,

$$d_1 \geq d_2 \geq \cdots \geq d_r > d_{r+1} = \cdots d_{\min(m,n)} = 0.$$

Example: underdetermined system, $m < n$,

$$D = \begin{bmatrix} d_1 & & & & & & \\ & \ddots & & & & & \\ & & d_r & & & & \\ & & & 0 & & & \\ & & & & \ddots & & \\ & & & & & 0 \quad \cdots \quad 0 \end{bmatrix}$$

The columns of U and V form a *singular system* of the matrix A: let

$$U = \left[\begin{array}{cccc} \mathbf{u}_1 & \mathbf{u}_2 & \cdots & \mathbf{u}_m \end{array} \right], \quad V = \left[\begin{array}{cccc} \mathbf{v}_1 & \mathbf{v}_2 & \cdots & \mathbf{v}_n \end{array} \right].$$

Assume that $m \leq n$. We have

$$A\mathbf{v}_j = \begin{cases} d_j \mathbf{u}_j, & 1 \leq j \leq m, \\ 0, & m+1 \leq j \leq n. \end{cases}$$

If, on the other hand, $n \leq m$, we have simply

$$A\mathbf{v}_j = d_j \mathbf{u}_j, \quad 1 \leq j \leq n,$$

and

$$\mathbf{u}_j \perp A\mathbf{x}, \quad n+1 \leq j \leq m$$

for all $\mathbf{x} \in \mathbb{R}^n$.

Exponent of the Gaussian:

$$\|A\mathbf{x} - \mathbf{b}\|^2 = \|UDV^{\mathrm{T}}\mathbf{x} - \mathbf{b}\|^2 = \|U(DV^{\mathrm{T}}\mathbf{x} - U^{\mathrm{T}}\mathbf{b})\|^2.$$

Orthogonal matrix does not change the norm:

$$\|U\mathbf{v}\|^2 = (U\mathbf{v})^{\mathrm{T}}U\mathbf{v} = \mathbf{v}^{\mathrm{T}}\underbrace{U^{\mathrm{T}}U}_{=I}\mathbf{v} = \|\mathbf{v}\|^2,$$

so

$$\|A\mathbf{x} - \mathbf{b}\|^2 = \|DV^{\mathrm{T}}\mathbf{x} - U^{\mathrm{T}}\mathbf{b}\|^2.$$

Change of variable: define

$$\mathbf{y} = V^{\mathrm{T}}\mathbf{x}.$$

Determinant of V: since

$$\begin{aligned} 1 &= \det(I) = \det(V^{\mathrm{T}}V) \\ &= \det(V^{\mathrm{T}})\det(V) = \det(V)^2, \end{aligned}$$

so

$$|\det(V)| = 1.$$

The probability density, in the new coordinates $\mathbf{y}$, is therefore

$$\begin{aligned}
\pi(\mathbf{y}) &\propto \exp\left(-\frac{1}{2}\|D\mathbf{y} - U^{\mathrm{T}}\mathbf{b}\|^2\right) \\
&= \exp\left(-\frac{1}{2}\left(\sum_{j=1}^{r}(d_j y_j - b_j)^2 + \sum_{j=r+1}^{m} b_j^2\right)\right) \\
&\propto \exp\left(-\frac{1}{2}\sum_{j=1}^{r}(d_j y_j - b_j)^2\right).
\end{aligned}$$

Hence, we observe that *if no bound constraints were present*, the components y_j are independent,

$$\pi(\mathbf{y}) = \pi(y_1)\cdots\pi(y_r).$$

Bound constraints

The bound constraints break down the independency:

$$C\mathbf{x} \geq \mathbf{r}$$

in terms of $\mathbf{y}$ reads

$$\underbrace{CV}_{=W}\mathbf{y} \geq \mathbf{r}.$$

Observe: The components $j > r$ are restricted only by the bound constraints.

```
% Full scan Gibbs sampler, adaptation using SVD. The program
% produces a sample of size nsample from
% the probability distribution
%
% Pi(x) ~ exp(-0.5*||A*x - b||^2), lb< x < ub
%
% Input: A -      array of size (m,n)
%        b -      column vector of length m
%        lb, ub - column vectors of length n
%        x_init - column vector of length n
%        nsample - number of samples
% Output: X - array of size (n,nsample)
```

```
% Defining the matrix and the rhs

 th = pi/3;
 lratio = 0.01;
 A = diag([1,lratio])*[cos(th) -sin(th);sin(th) cos(th)];
 xdata = [1;1];
 b = A*xdata;
 std = 0.01;
 A = (1/std)*A;
 b = (1/std)*b;
```

```
% Upper and lower bounds and initial point

 lb = [0;0];
 ub = [10;10];
 x_init = [5;5];

% Writing the boun constraints in the standard form C*x>=r

 C = [eye(2);-eye(2)];
 r = [lb;-ub];
```

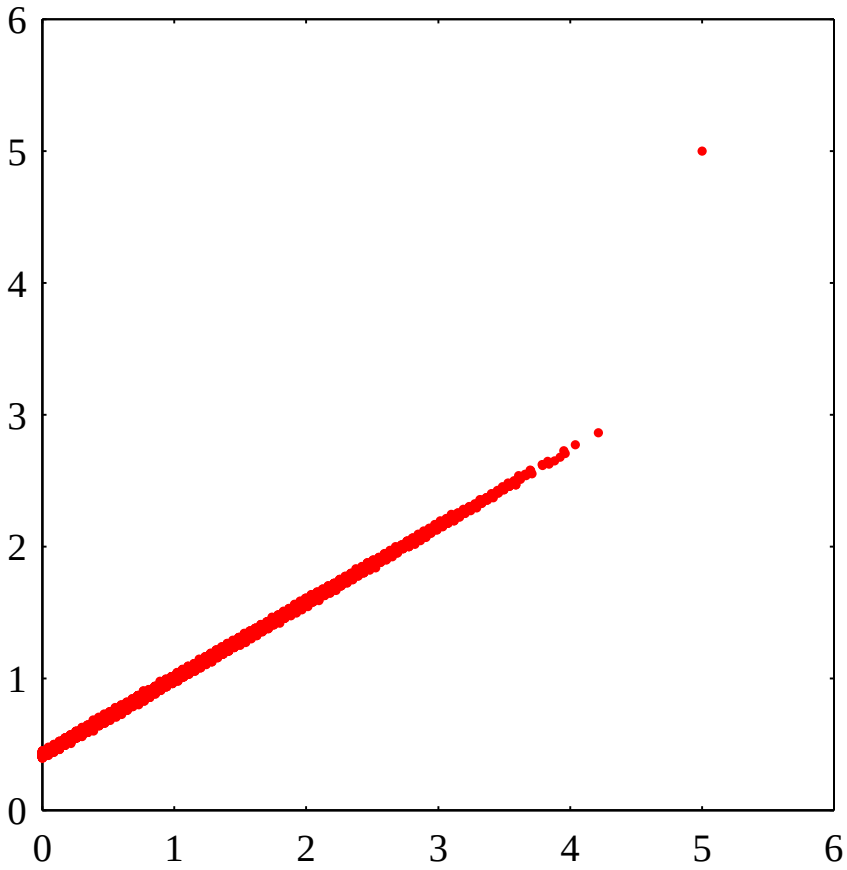
```
% Checking the inputs
 Iaux = find(lb>=ub);
 if length(Iaux) > 0
    display('Lower bound not less than upper bound; check input!')
    return
 end

 Iaux = find(x_init>ub);
 if length(Iaux) > 0
    display('Initial point not below upper bound; check input!')
 end

 Iaux = find(x_init<lb);
 if length(Iaux) > 0
    display('Initial point not above lower bound; check input!')
 end
```

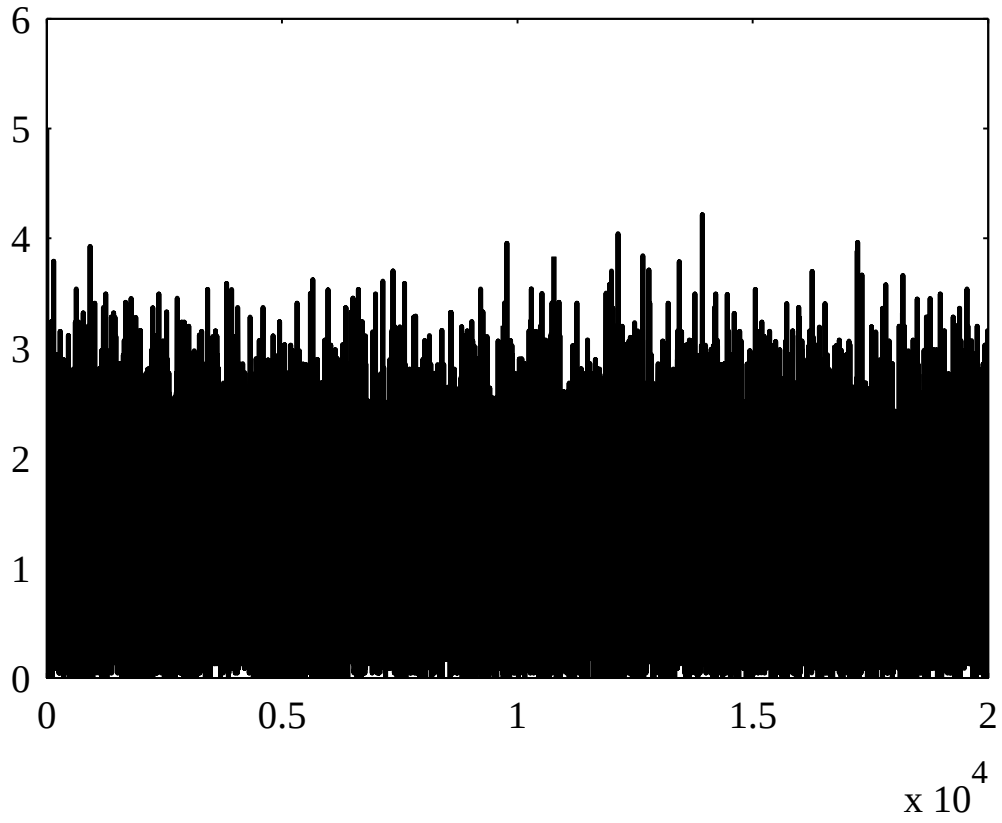
```
% Gibbs sampler: Full scan,SVD adaptation, no model reduction

 [U,D,V] = svd(A);
 W = C*V;
 b = U'*b;
 d = diag(D);
 ny = length(x_init);
 bin = 100;       % Number of division points in random draws
 cutoff = 1e-3;   % Cutoff level of the pdf
 tiny = 1e-8;     % Minimum length of reasonable interval
 tiny2 = 1e-10;   % Lower level for singular values that are
                  % not treated as zeros
 large = 1e6;     % Absolute upper bound
 nsample = 20000;
 Y = zeros(ny,nsample);
 y = V'*x_init;
 Y(:,1) = y;
```

```
for n = 2:nsample

   % Updating component by component the vector y

   for j = 1:ny
       wj = W(:,j);
       Ij = setdiff([1:ny],j);
       rj = r - W(:,Ij)*y(Ij);
       Iplus = find(wj>eps);
       Iminus = find(wj<-eps);
       ymin = max([-large;rj(Iplus)./wj(Iplus)]);
       ymax = min([large;rj(Iminus)./wj(Iminus)]);

       if ymax - ymin < tiny
          % the effective interval is reduced
          y(j) = ymin;
       else
          % draw from the density
```

```
if d(j)>tiny2
   % draw from truncated Gaussian
   tmax = d(j)*ymax;
   tmin = d(j)*ymin;
   if b(j) > tmax
      f0 = (tmax-b(j))^2;
      tstar = b(j) - sqrt(f0+2*log(1/cutoff));
      tmin = max(tmin,tstar);
   elseif b(j)<tmin
      f0 = (tmin-b(j))^2;
      tstar = b(j) + sqrt(f0+2*log(1/cutoff));
      tmax = min(tmax,tstar);
   else
      f0 = 0;
      tmin = max(tmin,b(j) - sqrt(2*log(1/cutoff)));
      tmax = min(tmax,b(j) + sqrt(2*log(1/cutoff)));
   end
```

```
            t = linspace(tmin,tmax,bin);
            p = exp(-0.5*((t-b(j)).^2 - f0));
            phi = cumsum(p);
            xi = phi(bin)*rand;
            jj = min(find(phi>xi));
            y(j) = t(jj)/d(j);
         else
            % Draw from uniform distribution
            y(j) = ymin + (ymax-ymin)*rand;
         end
      end
   end

   Y(:,n) = y;
end

X = V*Y;
```





$$\theta = \frac{\pi}{3}.$$

Correlation length

So far, the performance of a sampler has been assessed only visually.

Rules of thumb: a good sample should

- **not** have a visible trend
- **not** have significant low frequency features
- should cover tightly an interval
- have a sample history that looks like a "fuzzy worm"

Any quantitative measures?

The "stickiness" of a sample can be measured by its *autocorrelation.*

Time series

$$z_1, z_2, \ldots, z_n, \ldots, z_N.$$

Empirical mean is

$$\overline{z} = \frac{1}{N} \sum_{j=1}^{N} z_j.$$

Centered time series,

$$\widehat{z}_j = z_j - \overline{z}, \quad j = 1, \ldots, N.$$

The kth *lagged autocorrelation* is defined as

$$\gamma_k = \frac{1}{\|\widehat{z}\|^2} \sum_{j=1}^{N-k} \widehat{z}_j \widehat{z}_{j-k},$$

where

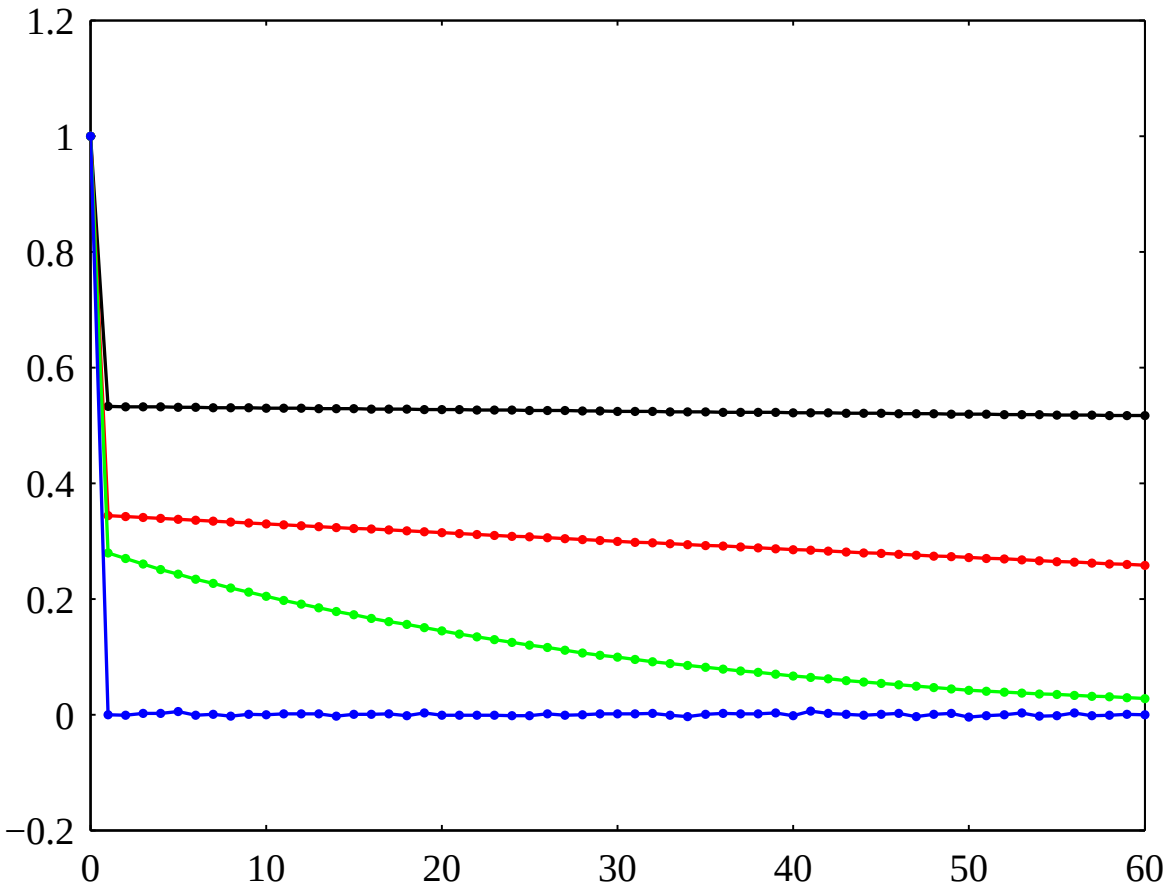
$$\|\widehat{z}\|^2 = \sum_{j=1}^{N} \widehat{z}_j^2.$$

In particular,

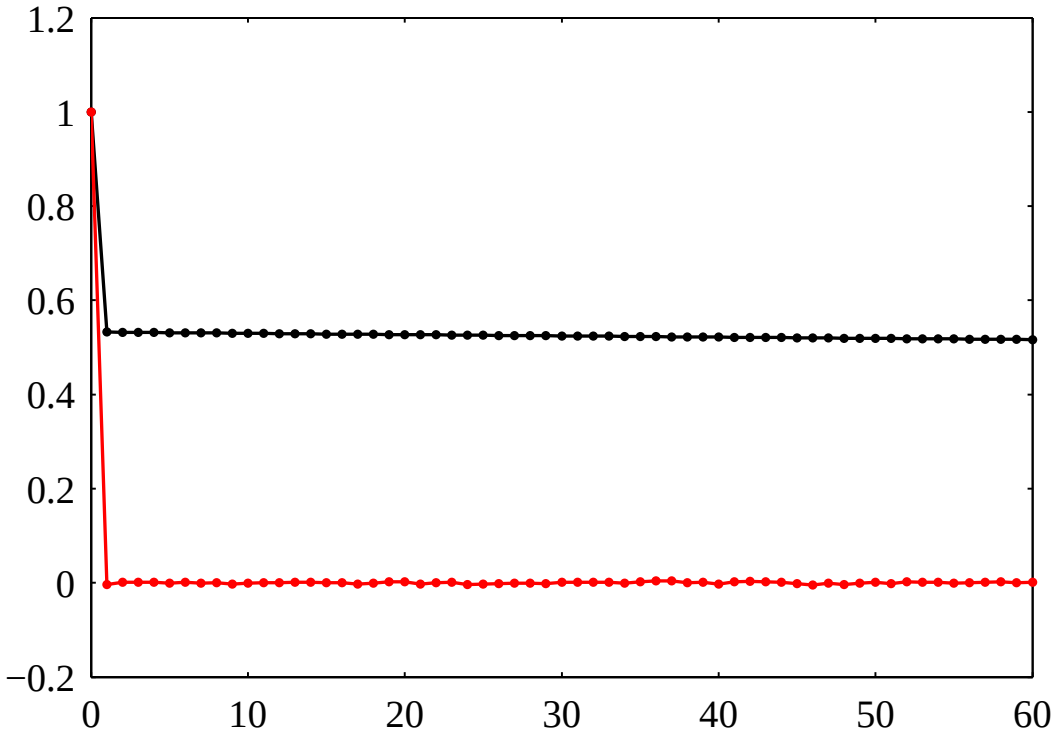
$$\gamma_0 = 1.$$

Program

```
N = length(x);
 xmean = 1/N*sum(x);
 xc = x - xmean;
 normx = norm(x);
 kmax = 61;
 gamma = ones(1,kmax);
 for k = 2:kmax
     x0 = xc(1:N-k+1);
     xk = xc(k:N);
     gamma(k) = (1/normx^2)*x0*xk';
 end
```



Lagged correlations of samples with Gibbs sampler, various θ.



Lagged correlations, $\theta = \pi/3$ with and without SVD