**PROBABILISTIC ROBOTICS:
NONPARAMETRIC FILTERS**

Pierre-Paul TACHER

1

We restate the motion model considered:

$$\begin{bmatrix} x_{t+1} \\ \dot{x}_{t+1} \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}}_{A} \begin{bmatrix} x_t \\ \dot{x}_t \end{bmatrix} + \underbrace{\begin{bmatrix} \frac{1}{2}\ddot{x}_t \\ \ddot{x}_t \end{bmatrix}}_{\epsilon_t}$$

$$\begin{bmatrix} x_{t+1} \\ \dot{x}_{t+1} \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_t \\ \dot{x}_t \end{bmatrix} + \underbrace{\begin{bmatrix} \frac{1}{2} \\ 1 \end{bmatrix}}_{G} \ddot{x}_t$$

$$G \times G^T = \begin{bmatrix} \frac{1}{4} & \frac{1}{2} \\ \frac{1}{2} & 1 \end{bmatrix}$$

As a linear function of gaussian $\ddot{x}_t$, the random variable ϵ_t is known to be multivariate Gaussian $\mathcal{N}(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \underbrace{\sigma^2 GG^T}_{R})$. The conditional law of random variable $\begin{bmatrix} x_{t+1} \\ \dot{x}_{t+1} \end{bmatrix}$ given $(\begin{bmatrix} x_t \\ \dot{x}_t \end{bmatrix})$ is then known to be also Gaussian $\mathcal{N}(A \begin{bmatrix} x_t \\ \dot{x}_t \end{bmatrix}, R)$.

We define the discretization grid in figure 1. Matrix indexes are related to cell center coordinates by

$$\begin{aligned} i &= \frac{y - y_{\min}}{c} + 1 \\ &= \frac{k_y c - k_{y_{\min}} c}{c} + 1 \\ &= k_y - k_{y_{\min}} + 1 \\ j &= \frac{x - x_{\min}}{c} + 1 \\ &= k_x - k_{x_{\min}} + 1 \end{aligned}$$

where $(k_x, k_y) \in \mathbb{Z}^2$.To compute the motion update, we take advantage of the fact the linear map

$$\begin{aligned} f : \mathbb{R}^2 &\to \mathbb{R}^2 \\ \begin{bmatrix} x \\ y \end{bmatrix} &\mapsto A \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \end{aligned}$$

leaves the grid stable:

$$\begin{aligned} \forall (k_x, k_y) \in \mathbb{Z}^2, \quad \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} &= \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} k_x c \\ k_y c \end{bmatrix} \\ &= \begin{bmatrix} \underbrace{(k_x + k_y)}_{\in \mathbb{Z}} c \\ k_y c \end{bmatrix} \end{aligned}$$

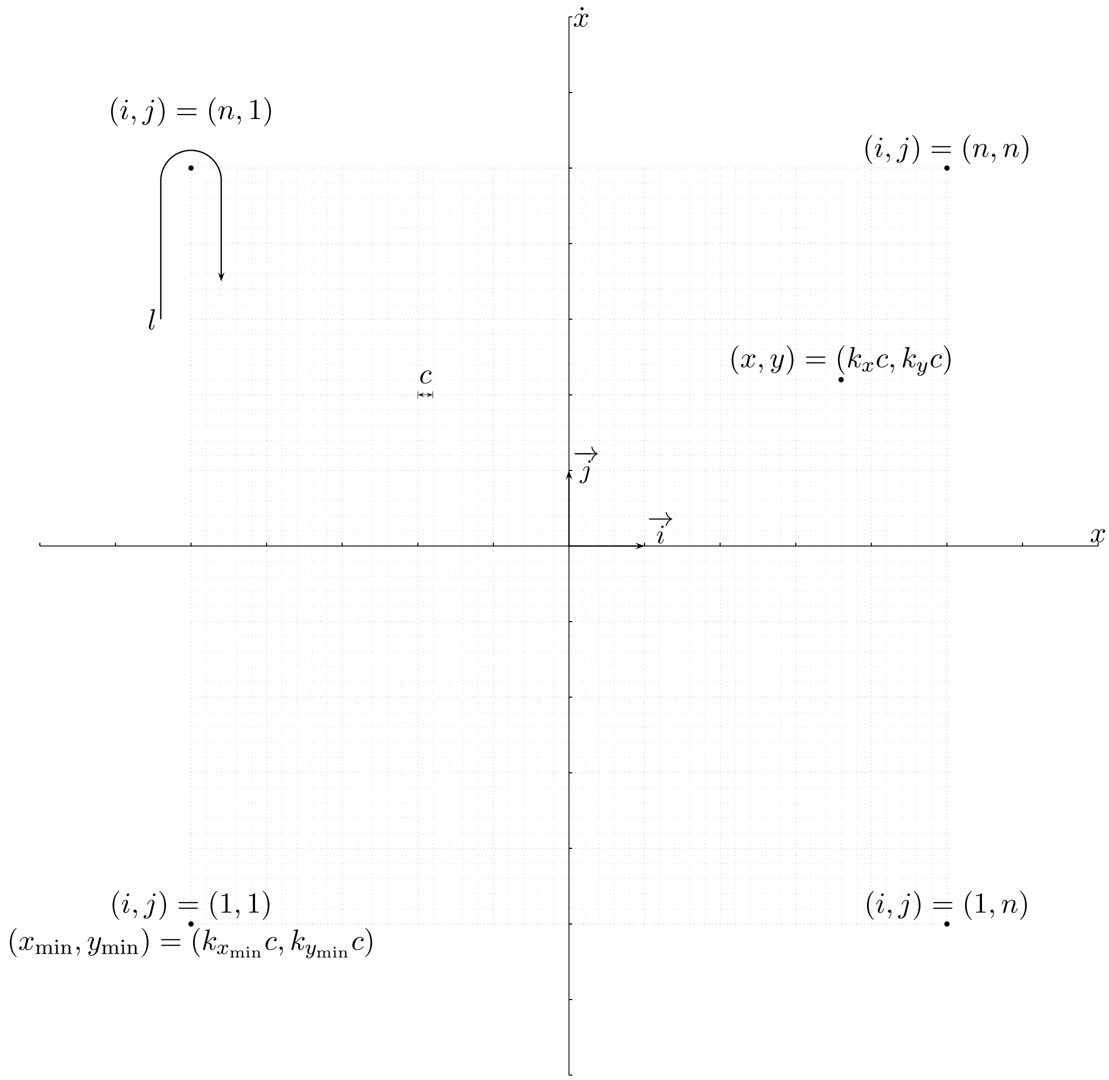


Figure 1. Definition of the grid

This shows the linear map A induces a map on matrix indexes:

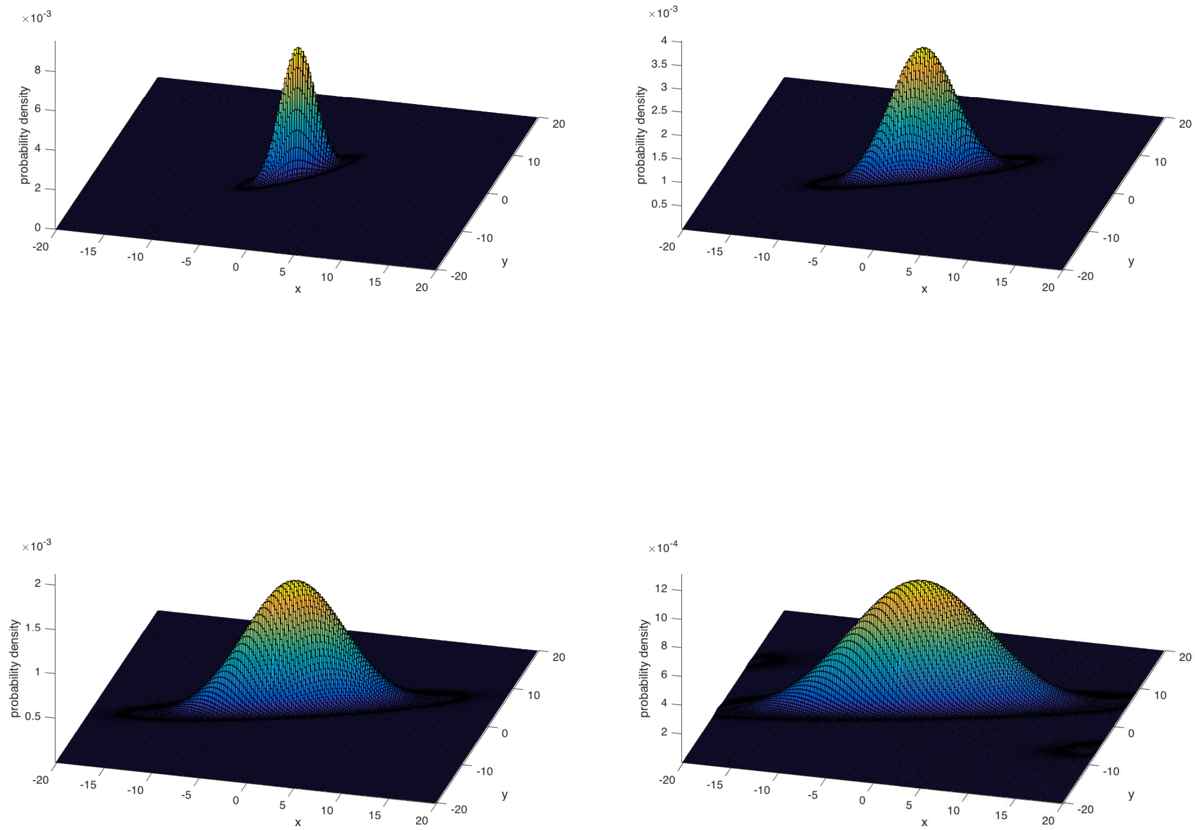
$$\begin{aligned} [\![1,n]\!]^2 &\to [\![1,n]\!]^2 \\ \begin{bmatrix} i \\ j \end{bmatrix} &\mapsto \begin{bmatrix} i' \\ j' \end{bmatrix} \end{aligned}$$

where

$$\begin{aligned} i' &= (k_x + k_y) + 1 - k_{y_{\min}} \\ &= i + j - 1 + k_{x_{\min}} \\ j' &= j \end{aligned}$$

provided $i + j - 1 + k_{x_{\min}} \in [\![1,n]\!]$. The Matlab style 1-based linear index is defined by

$$\begin{aligned} & l = n(j-1) + i \\ \Leftrightarrow\ & i = (l-1) \pmod n + 1 \\ & j = \left\lfloor \frac{l-1}{n} \right\rfloor + 1 \end{aligned}$$



Then we consider the map on linear indexes defined by

$$\begin{aligned} l' &= n(j'-1)+i' \\ &= n(j-1)+i+j-1+k_{x_{\min}} \\ l' &= l+j-1+k_{x_{\min}} \end{aligned}$$

provided $i+j-1+k_{x_{\min}} \in [\![1,n]\!]$. The Matlab code for this exercise is in `histogram_filter_1.m`, `plotdistrib3.m`. In figures 2 and **??**, we draw the histograms corresponding to motion update $t=2,\ldots,5$. The last histogram of each figure represents the probability distribution after the measurement update at $t=5$.

2

In accordance with discussion in previous chapter, we assume in this exercise $\alpha = \theta - \lfloor \frac{\theta}{2\pi} \rfloor \times 2\pi$ is an uniform r.v. $\alpha \hookrightarrow \mathcal{U}([0,2\pi])$, and $\begin{bmatrix} x \\ y \end{bmatrix}$ is gaussian $\begin{bmatrix} x \\ y \end{bmatrix} \hookrightarrow \mathcal{N}(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} \sigma_1^2 & 0 \\ 0 & \sigma_1^2 \end{bmatrix})$, where $\sigma_1^2 = 0.01$. We can then define the transition model by a discretized circular uniform distribution (cf. figure 4). The code for this exercise is in `histogram_filter_2.m`, `detect_circle.m` and `simmov.m`. In figures 5 and **??**, we draw the histograms after motion step and after measurement $z_1 = 0.65$.

3

To do.

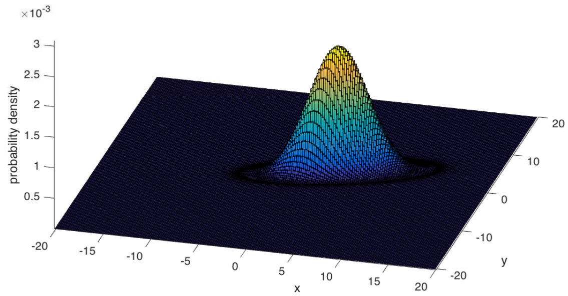
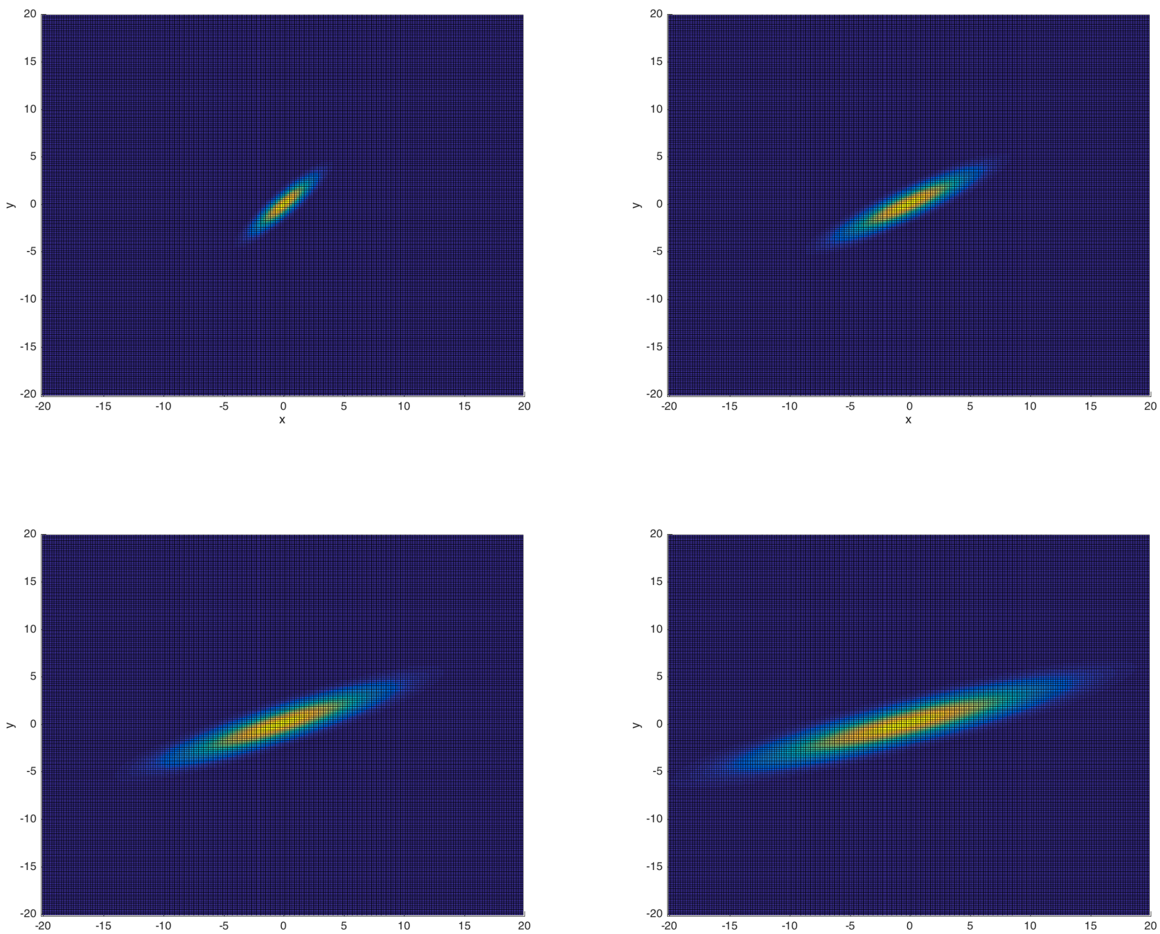


FIGURE 2. Histograms for the probability distributions for $t \in [\![2, 5]\!]$, perspective views.



4

In this exercise and the following, I use the Epanechnikov kernel density estimator to construct a smooth density from the M particles produced by the algorithm. The definition is

$$\forall x \in \mathbb{R}^d, \quad f(x) = \frac{1}{Mh^d} \sum_{i=1}^{M} K(\frac{x - X_i}{h})$$

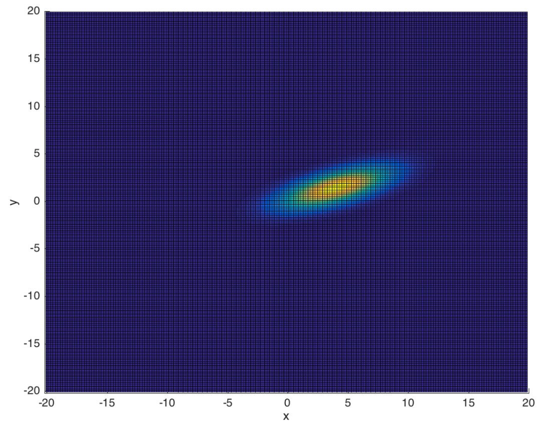


FIGURE 3. Histograms for the probability distributions for $t \in [\![2, 5]\!]$, top views.

where

$$K(x) = \begin{cases} \frac{1}{2} c_d^{-1}(d+2)(1 - x^T x) & \text{if } x^T x < 1 \\ 0 & \text{otherwise} \end{cases}$$

where c_d is the volume of the unit sphere in $\mathbb{R}^d$. The results of running particle algorithm are plotted in figure 7, the density estimation is represented in figure 8.

4.1. The results of running particle algorithm for the model of the exercise are plotted in figure 9, the density estimation is represented in figure 10.

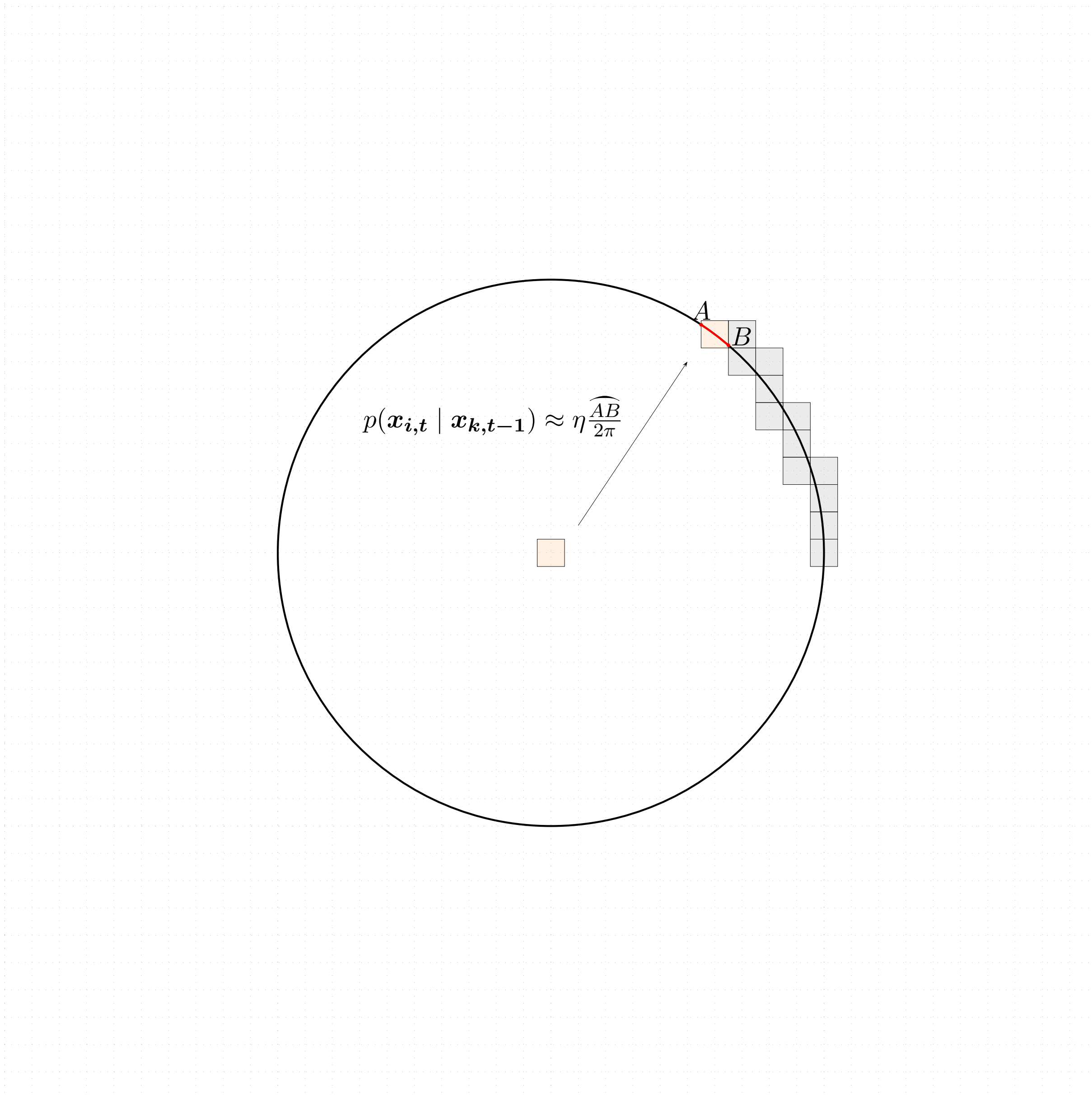


FIGURE 4. Transition model

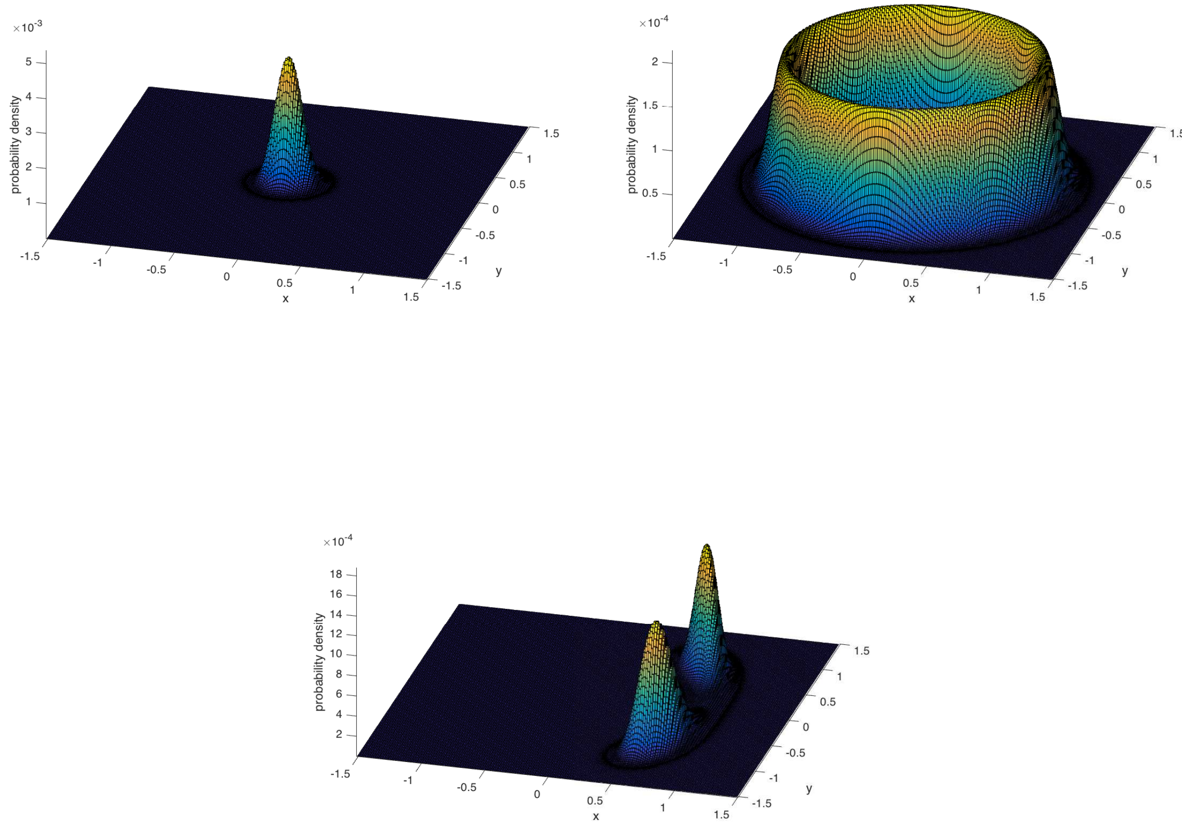
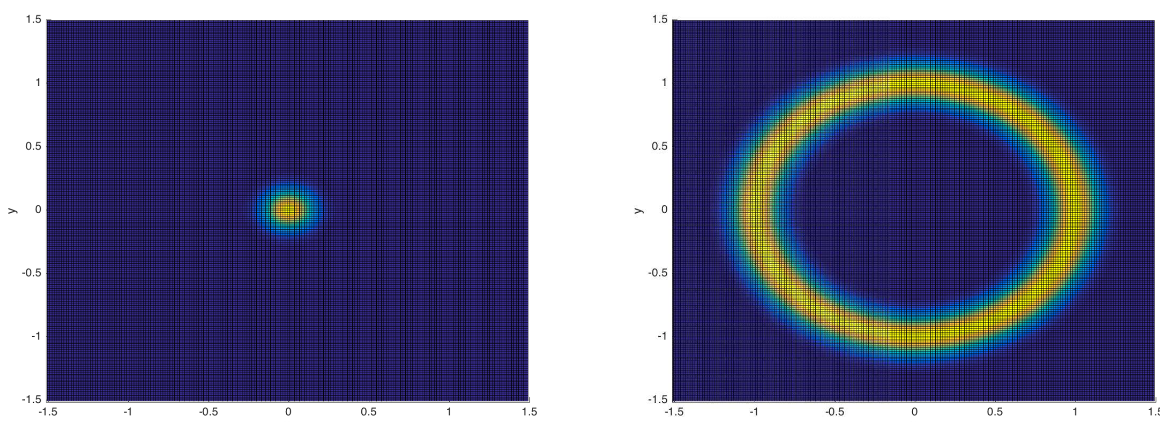


FIGURE 5. Histograms for the probability distributions, perspective views.



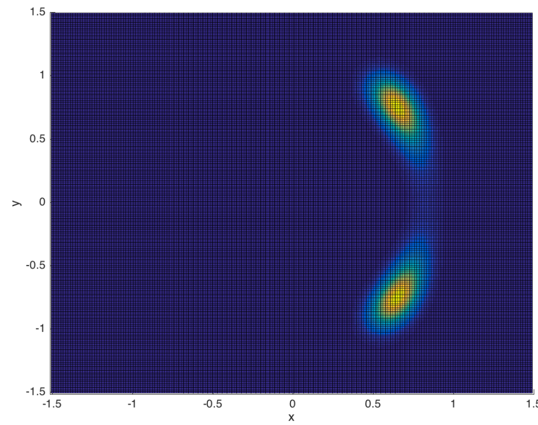
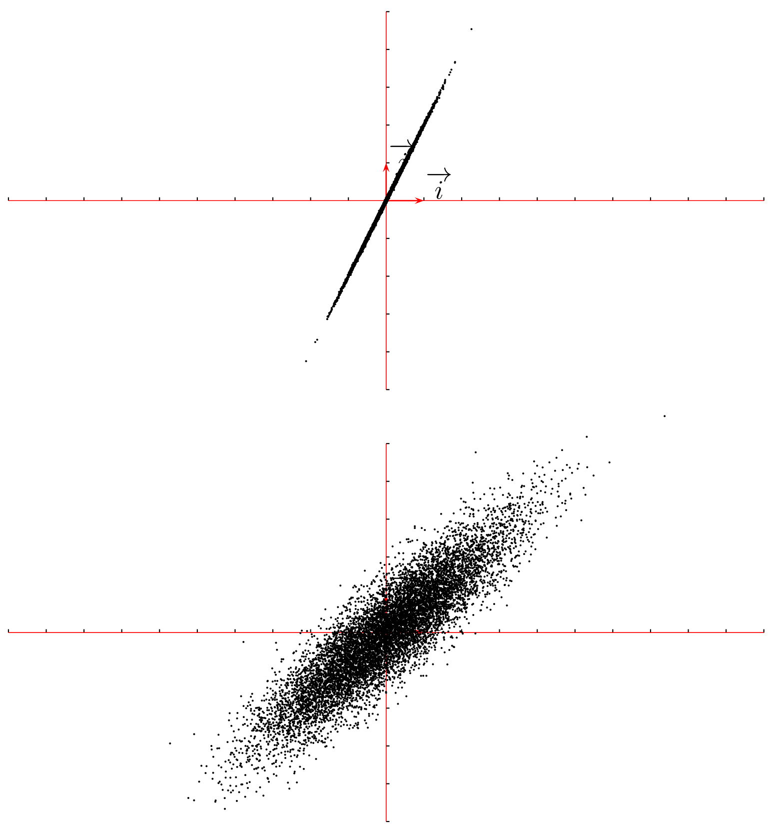
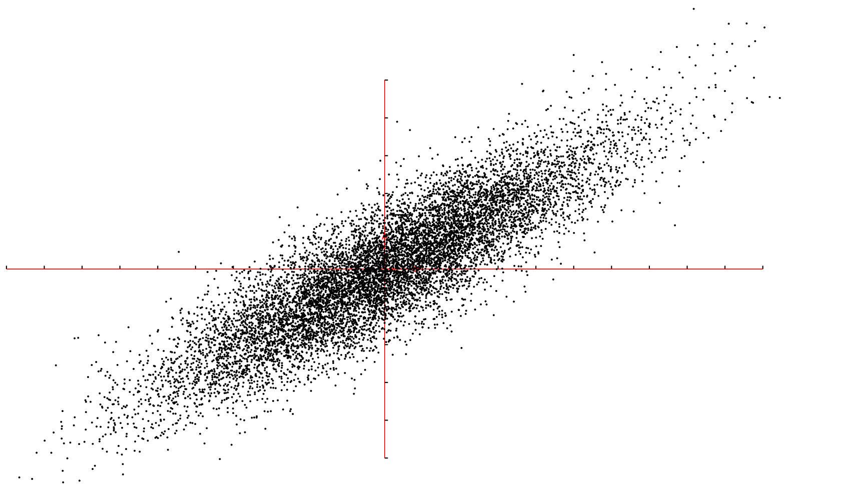
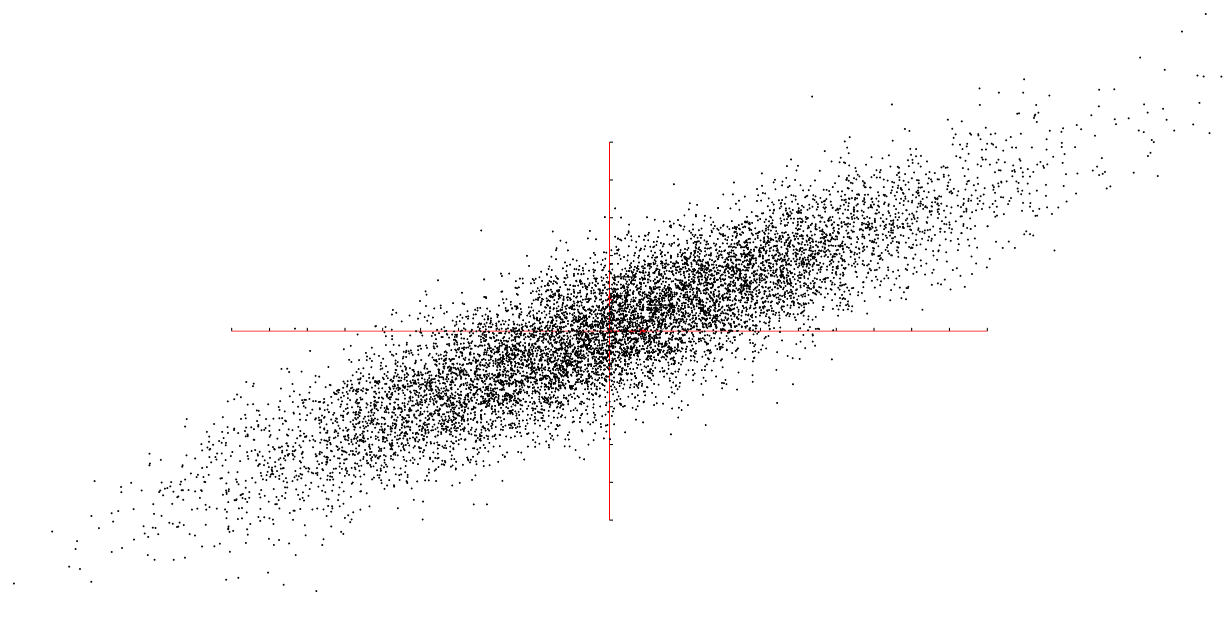


FIGURE 6. Histograms for the probability distributions, top views.







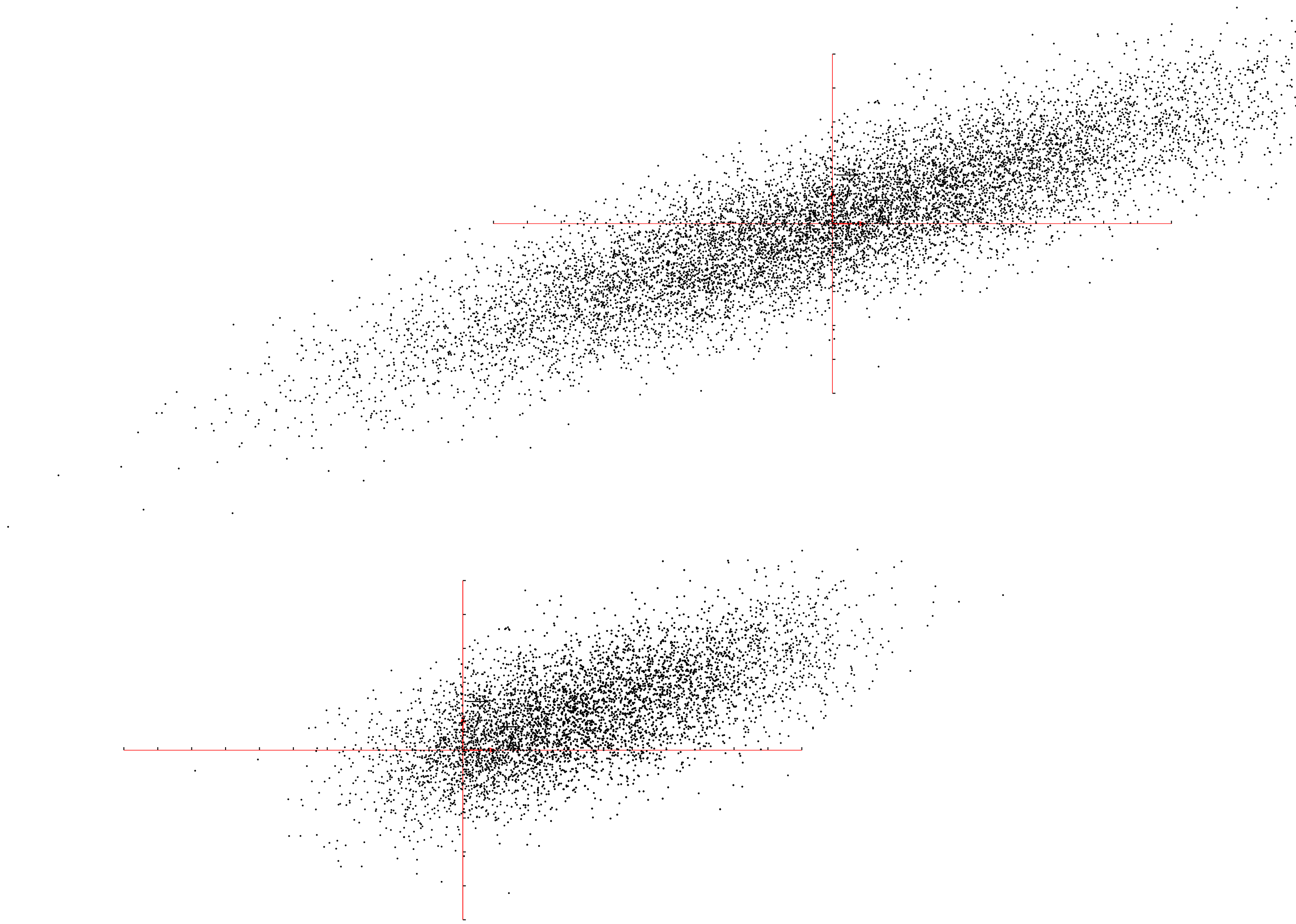


Figure 7. Particles algorithm running for $t \in 1, \ldots, 5$

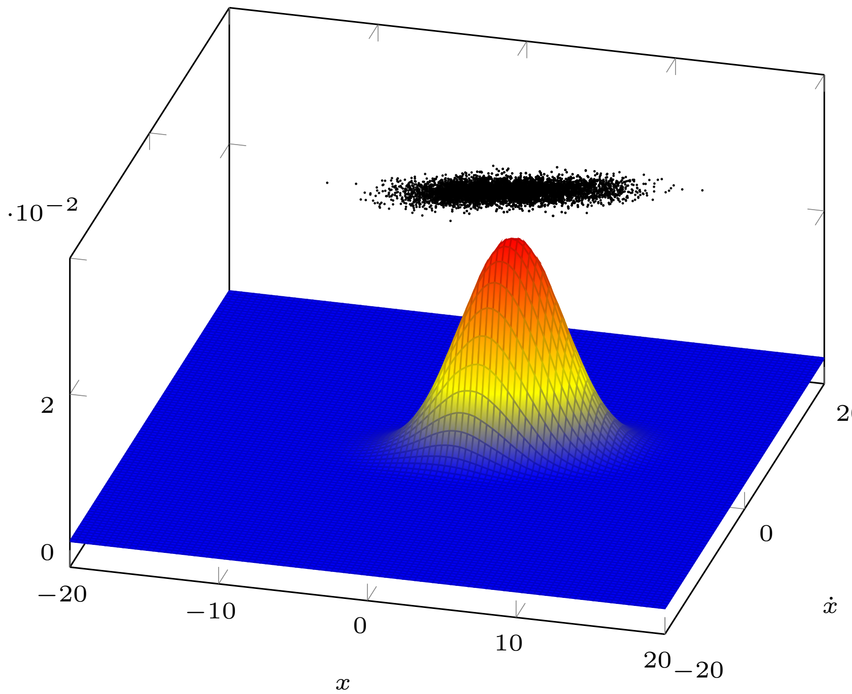
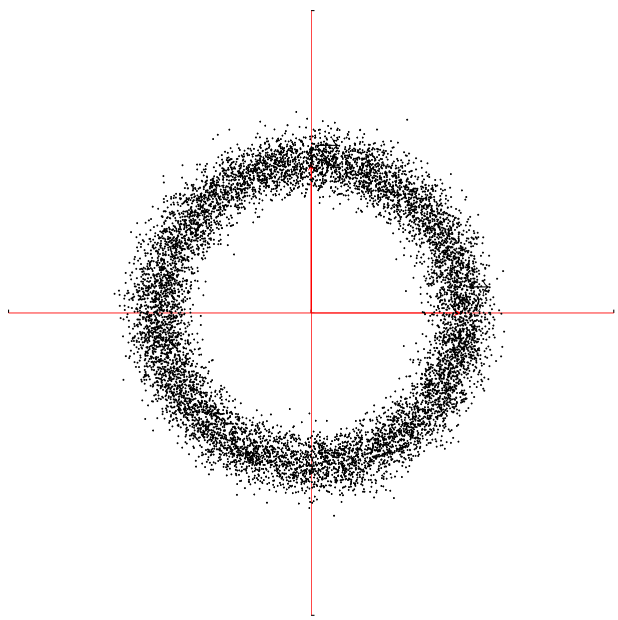


FIGURE 8. Kernel density estimation from particles



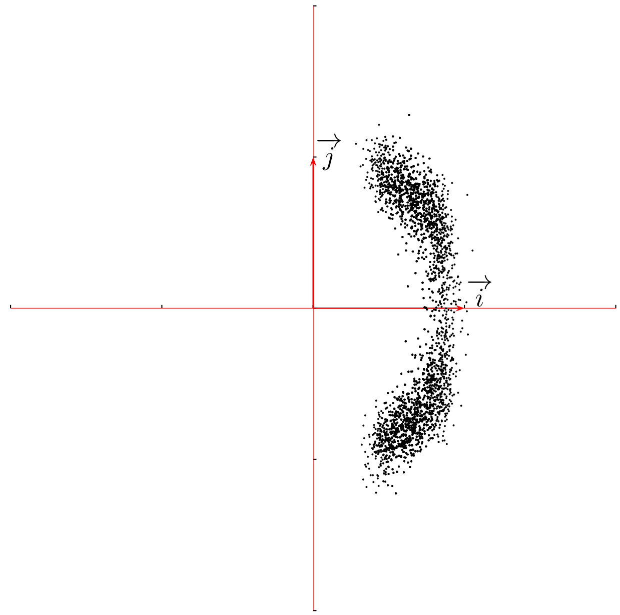


FIGURE 9. Particles algorithm

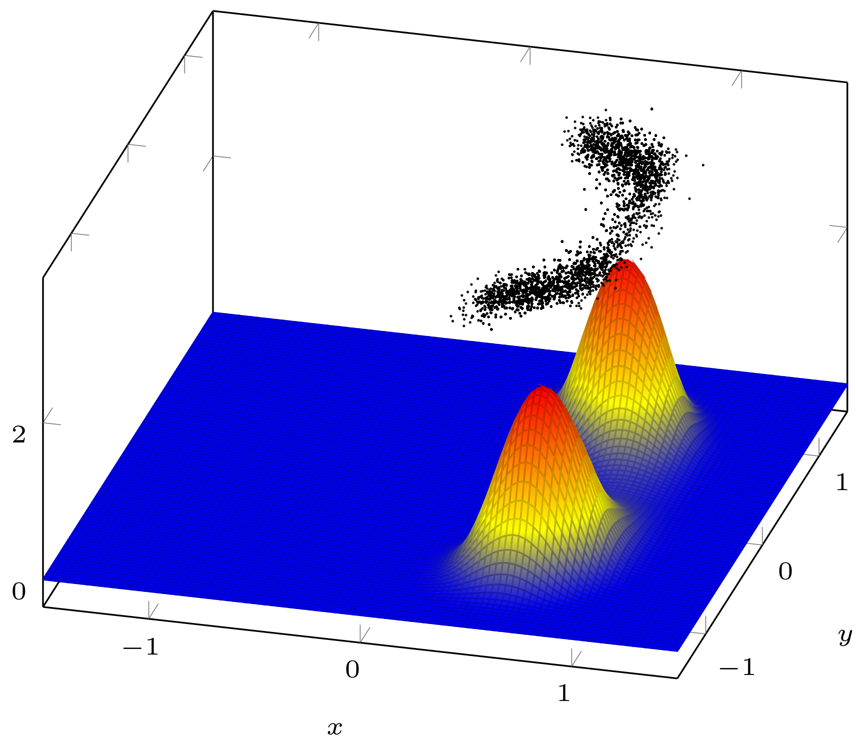


FIGURE 10. Kernel density estimation from particles