

Medical Image Analysis

Instructor: Moo K. Chung

mchung@stat.wisc.edu

Lecture 09.

Gaussian Kernel Smoothing

March 01, 2007

Gaussian Kernel Smoothing

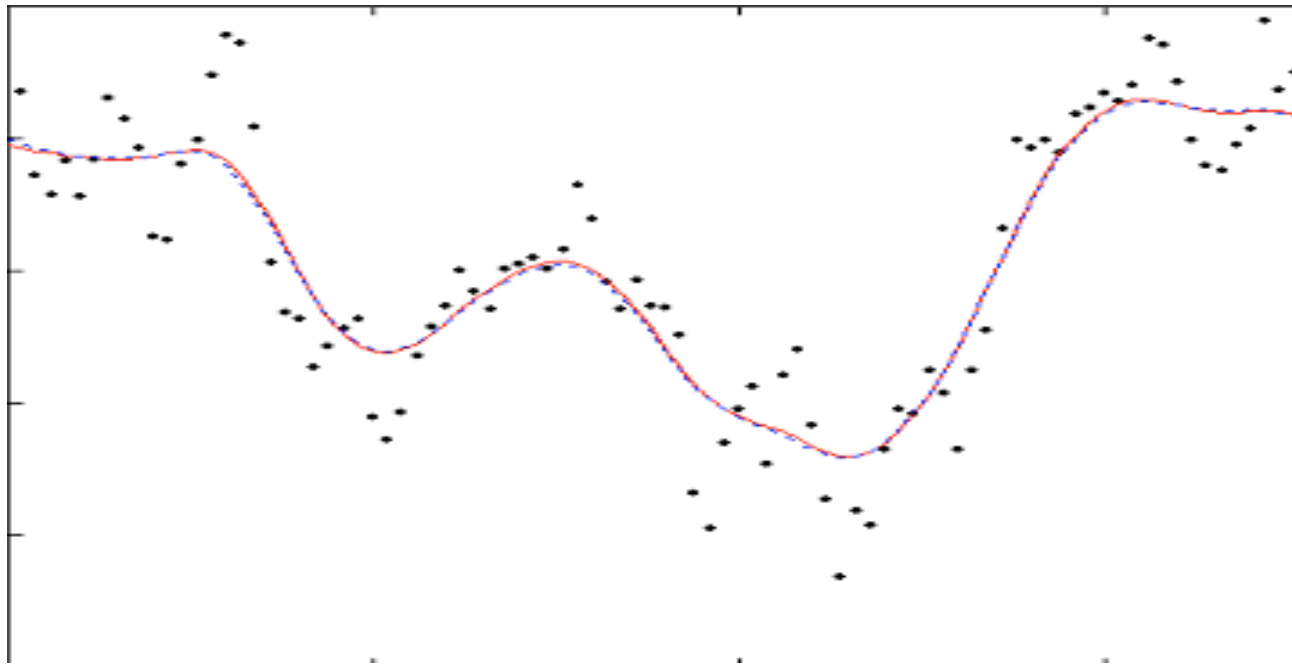
We will study basic properties of Gaussian kernel smoothing and numerical implementation issues.

Kernel Smoothing, Convolution, Linear Filter

$$Y(t) = \int K(t, s)X(s) ds.$$

output kernel input

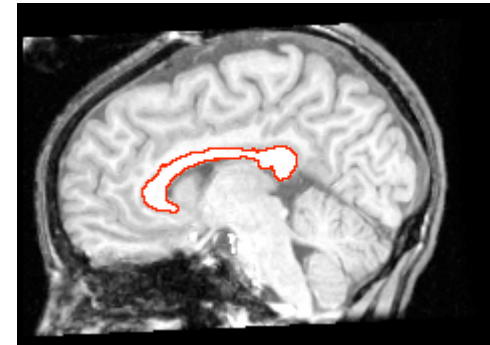
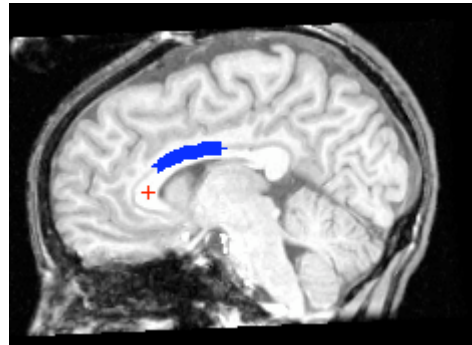
$$Y(t) = K * X(t) = \int K(t - s)X(s) ds.$$



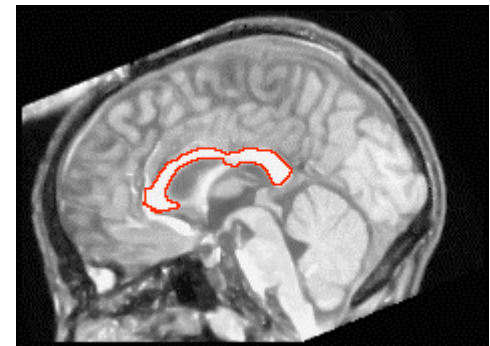
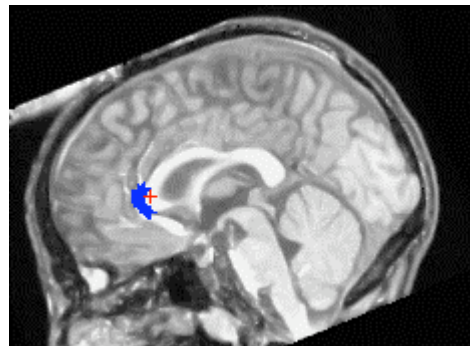
2D example



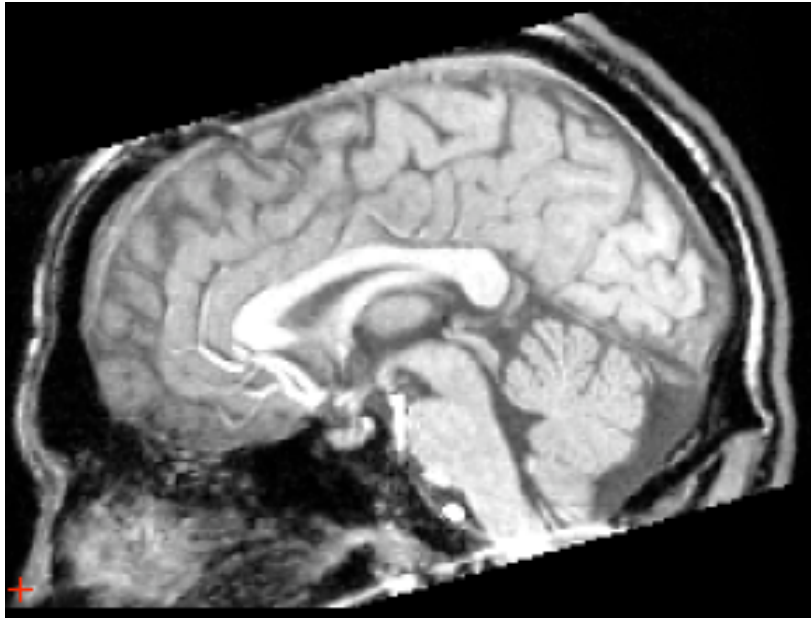
Motivation for image smoothing: Improve performance of PDE based segmentation - level set



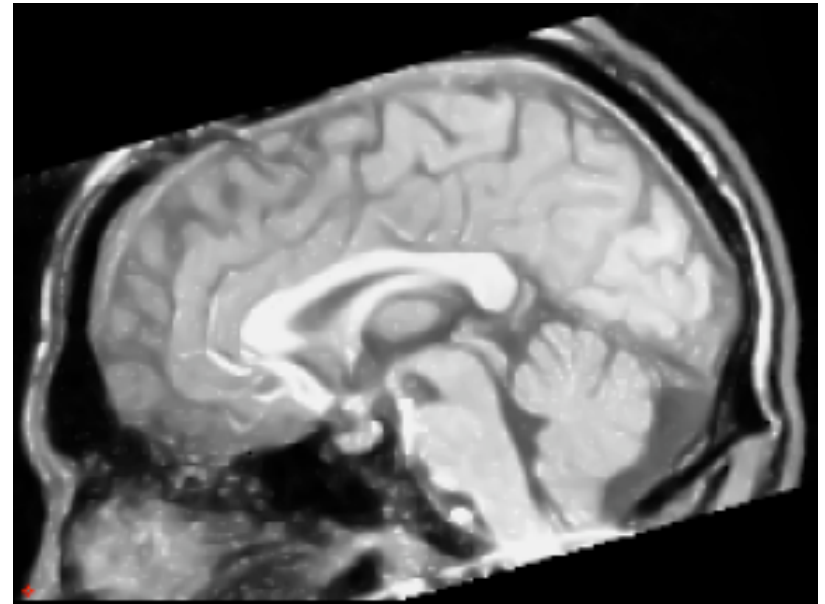
No image filtering
= More manual correction ☹️



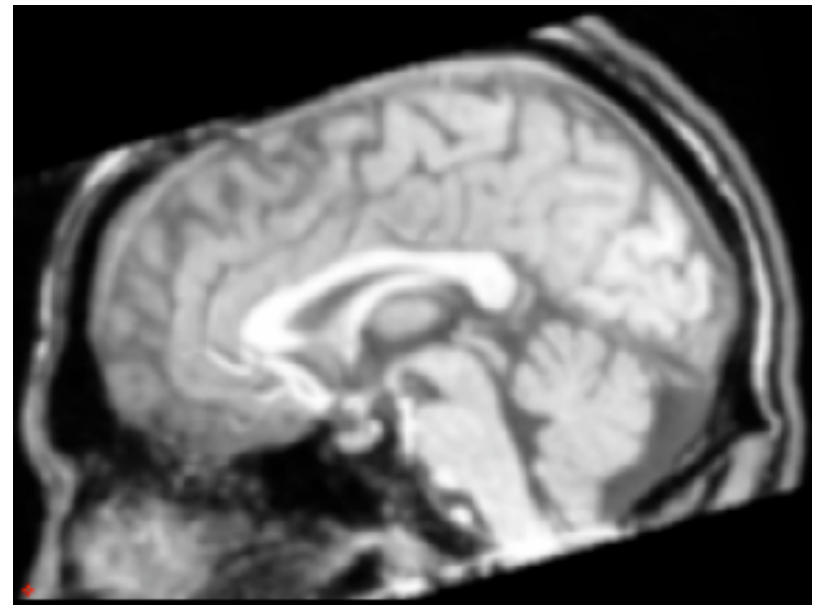
Malladi & Sethian's Min/Max Flow
done by Thomas Hoffmann.
This is basically a PDE smoother.



Original



Min/Max Flow

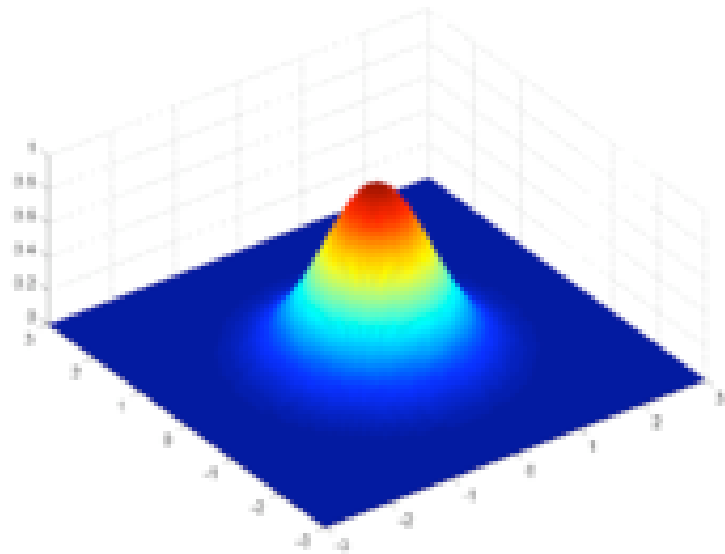
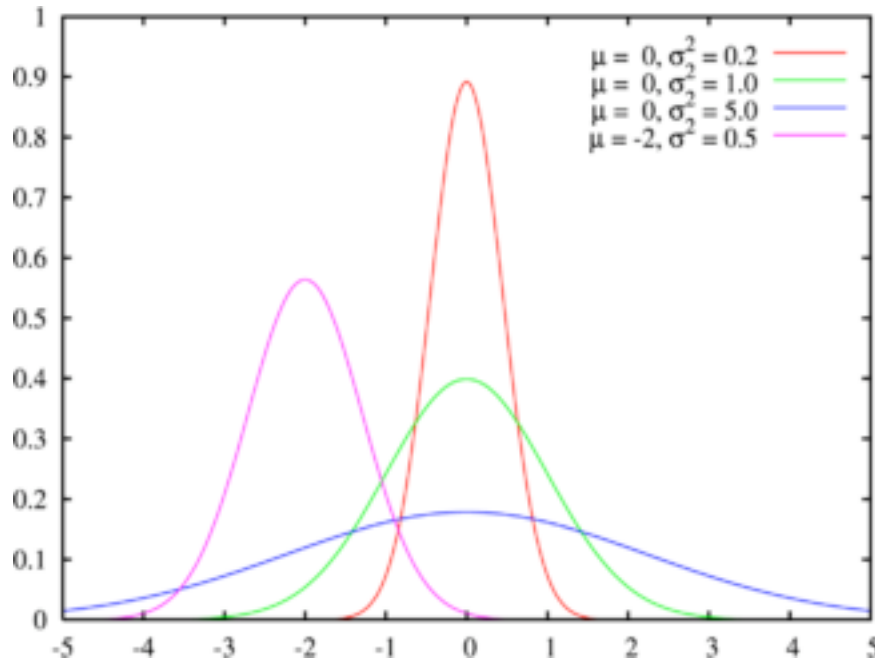


Gaussian

Shape of kernel

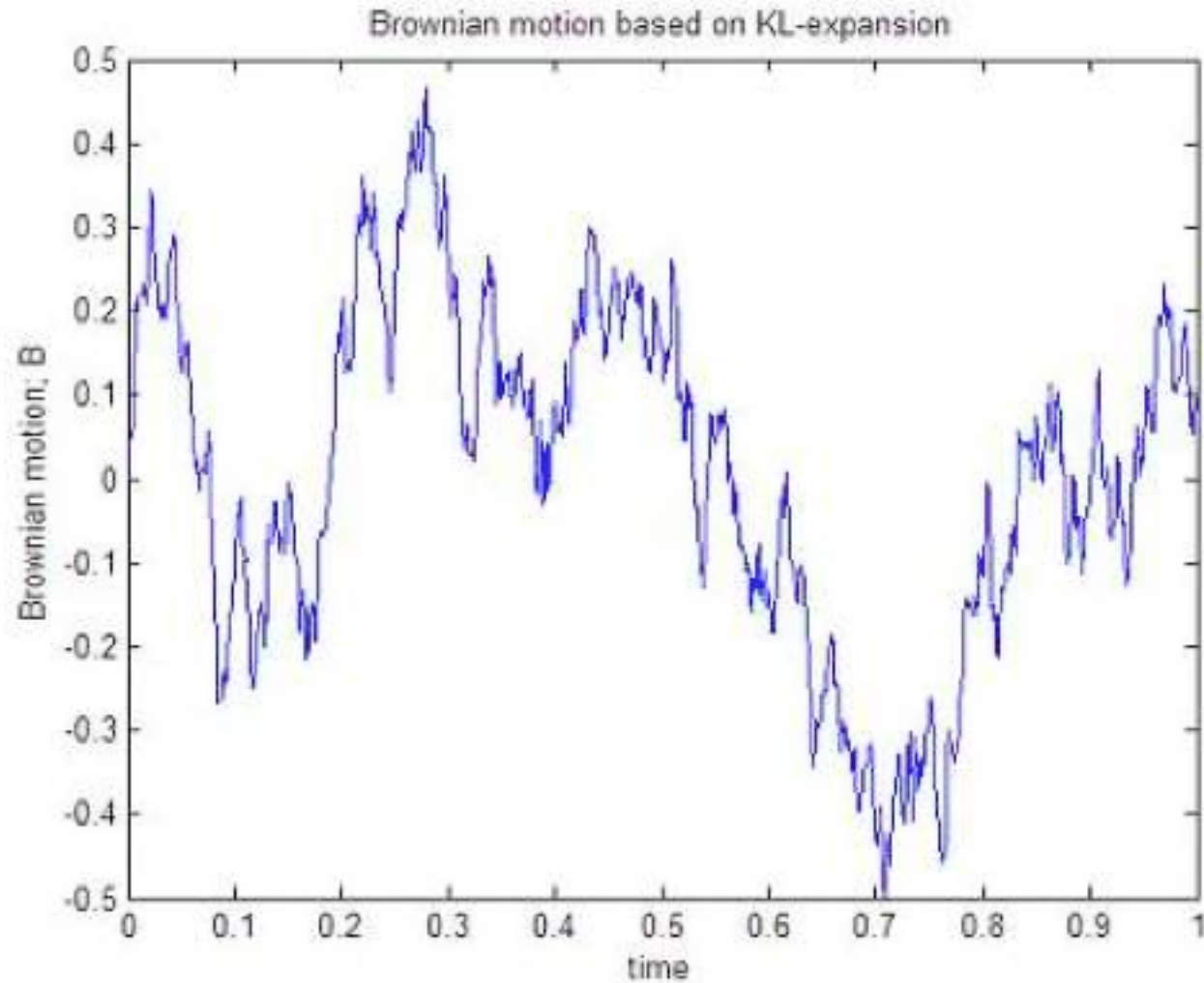
Unimodal, Symmetric (Isotropic), normalized

1D and 2D Gaussian kernel



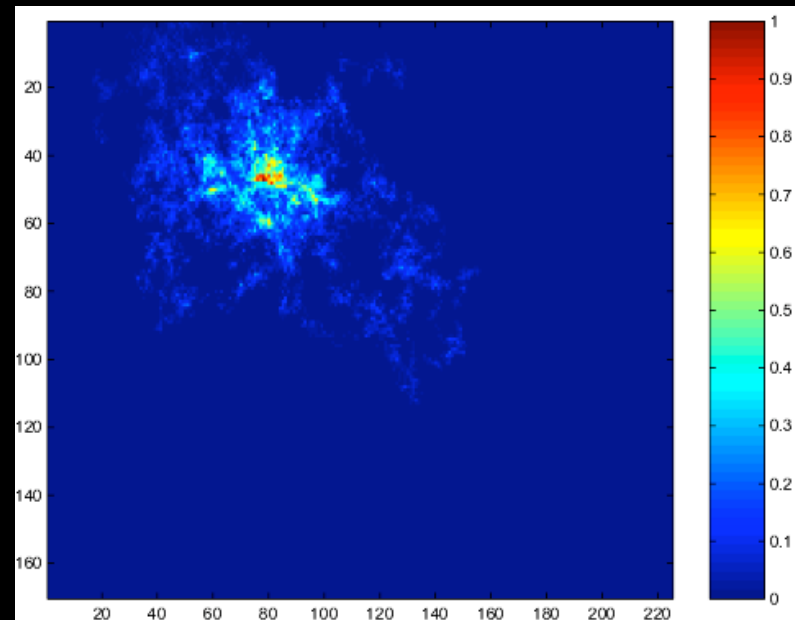
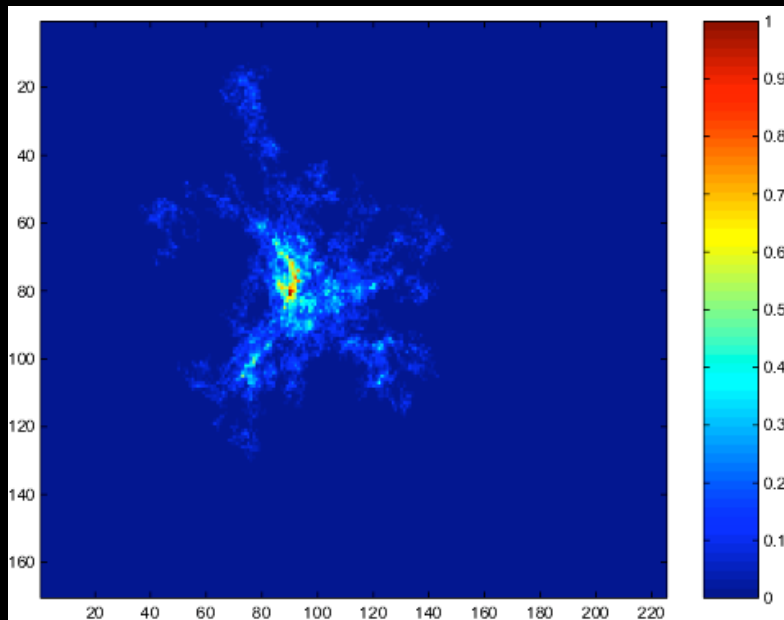
Quiz: The cross section of 2D Gaussian kernel ?

1D Brownian motion



Brownian motion simulation ---> Gaussian kernel

$$\text{Probability} = \frac{\text{\# random walk hitting a target voxel}}{\text{\# total random walk}}$$



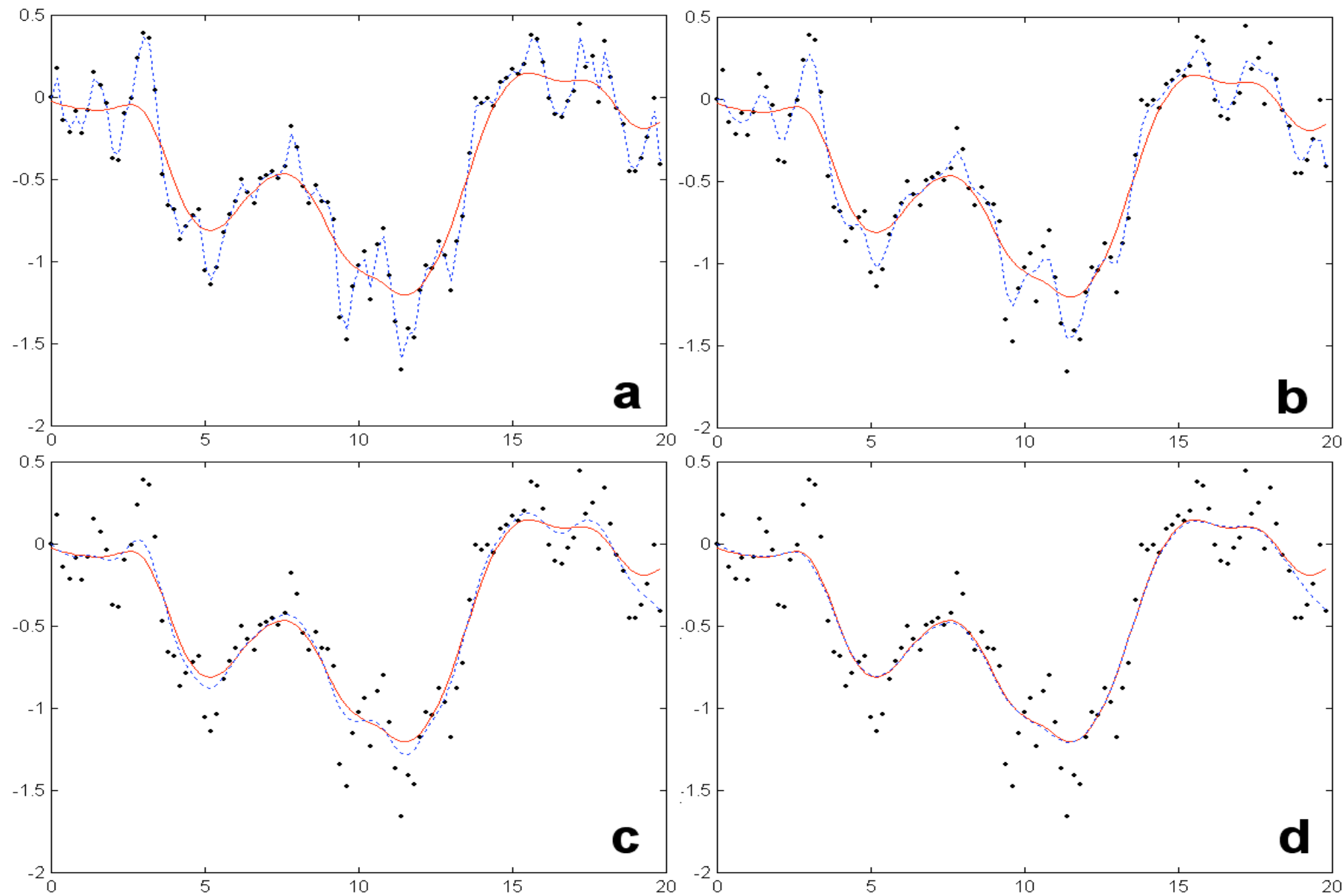
Constructing n -dimensional Gaussian Kernel

$$K(t) = \frac{1}{\sqrt{2\pi}} e^{t^2/2},$$

$$K_\sigma(t) = \frac{1}{\sigma} K\left(\frac{t}{\sigma}\right)$$

$$t = (t_1, \dots, t_n)' \in \mathbb{R}^n$$

$$K(t) = K_\sigma(t_1)K_\sigma(t_2) \cdots K_\sigma(t_n)$$



Red= Gaussian kernel smoothing

Blue = Diffusion smoothing after 5, 25 and 50 iterations

MATLAB code

```
K=inline('exp(-(x.^2+y.^2)/2/sig^2)');
```

```
>>K
```

Inline function:

```
K(sig,x,y) = exp(-(x.^2+y.^2)/2/sig^2)
```

```
[dx,dy]=meshgrid([-2:2]);
```

```
weight=K(0.5,dx,dy)/sum(sum(K(0.5,dx,dy)));
```

```
>>weight
```

```
0.0000 0.0000 0.0002 0.0000 0.0000  
0.0000 0.0113 0.0837 0.0113 0.0000  
0.0002 0.0837 0.6187 0.0837 0.0002  
0.0000 0.0113 0.0837 0.0113 0.0000  
0.0000 0.0000 0.0002 0.0000 0.0000
```

```
weight=K(1,dx,dy)/sum(sum(K(1,dx,dy)));
```

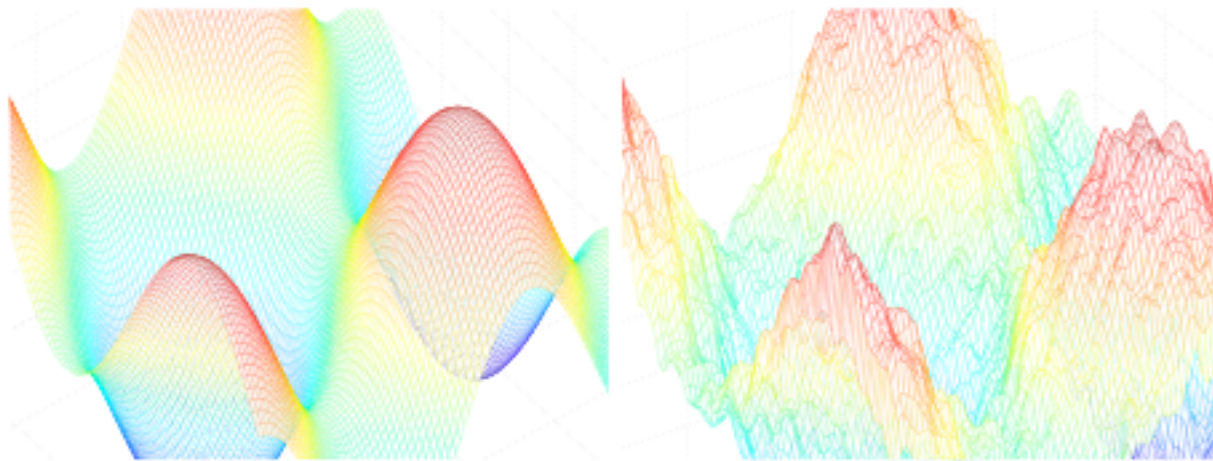
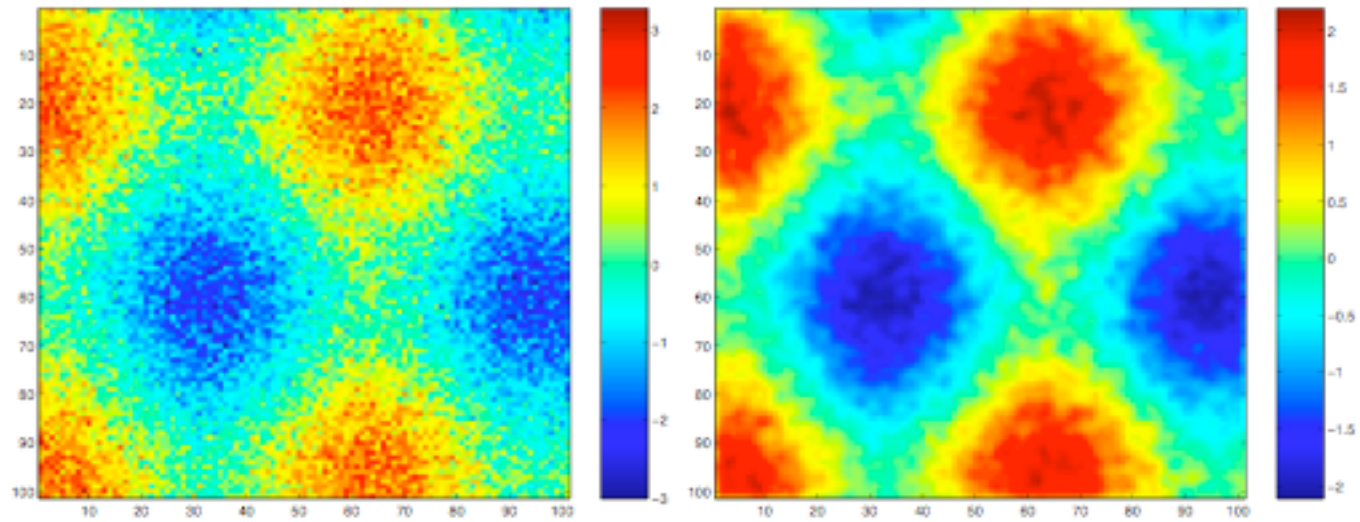
```
>>weight =
```

```
0.0030 0.0133 0.0219 0.0133 0.0030  
0.0133 0.0596 0.0983 0.0596 0.0133  
0.0219 0.0983 0.1621 0.0983 0.0219  
0.0133 0.0596 0.0983 0.0596 0.0133  
0.0030 0.0133 0.0219 0.0133 0.0030
```

```
Y=conv2(X,weight,'same');
```

Quiz: Why there is no $\sqrt{2} \cdot \sigma$ term in the computation?

2D simulation results



Better Algorithm

2D approach = 1D approach x 2

$$\begin{aligned}\int K_\sigma(t - s)X(s) \, ds &= \int K_\sigma(t_1 - s_1)K_\sigma(t_2 - s_2) \cdots K_\sigma(t_n - s_n)X(s) \, ds \\ &= \int K_\sigma(t_1 - s_1) \cdots K_\sigma(t_{n-1} - s_{n-1}) \, ds_1 \cdots ds_{n-1} \int K_\sigma(t_n - s_n)X(s) \, ds_n\end{aligned}$$

Perform 1D version of kernel smoothing in each coordinate

Gaussian Kernel estimator

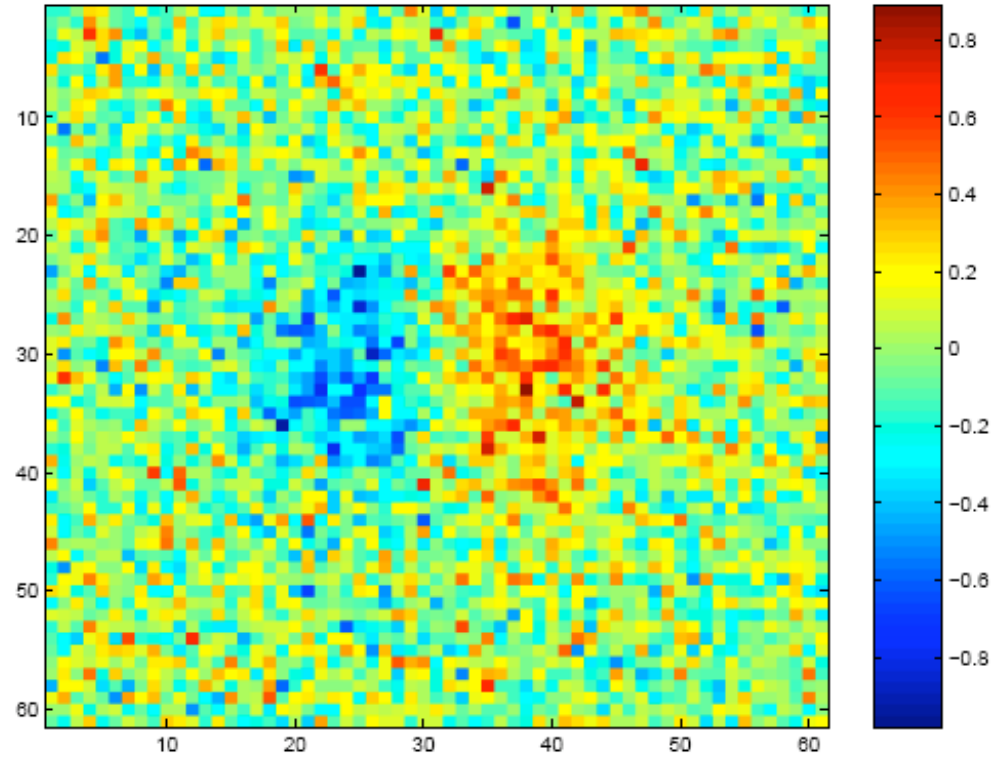
$$X(t) = \mu(t) + \epsilon(t)$$

$$\hat{\mu}(t) = K_{\sigma} * X(t) = \int K_{\sigma}(t - s)X(s) \, ds$$

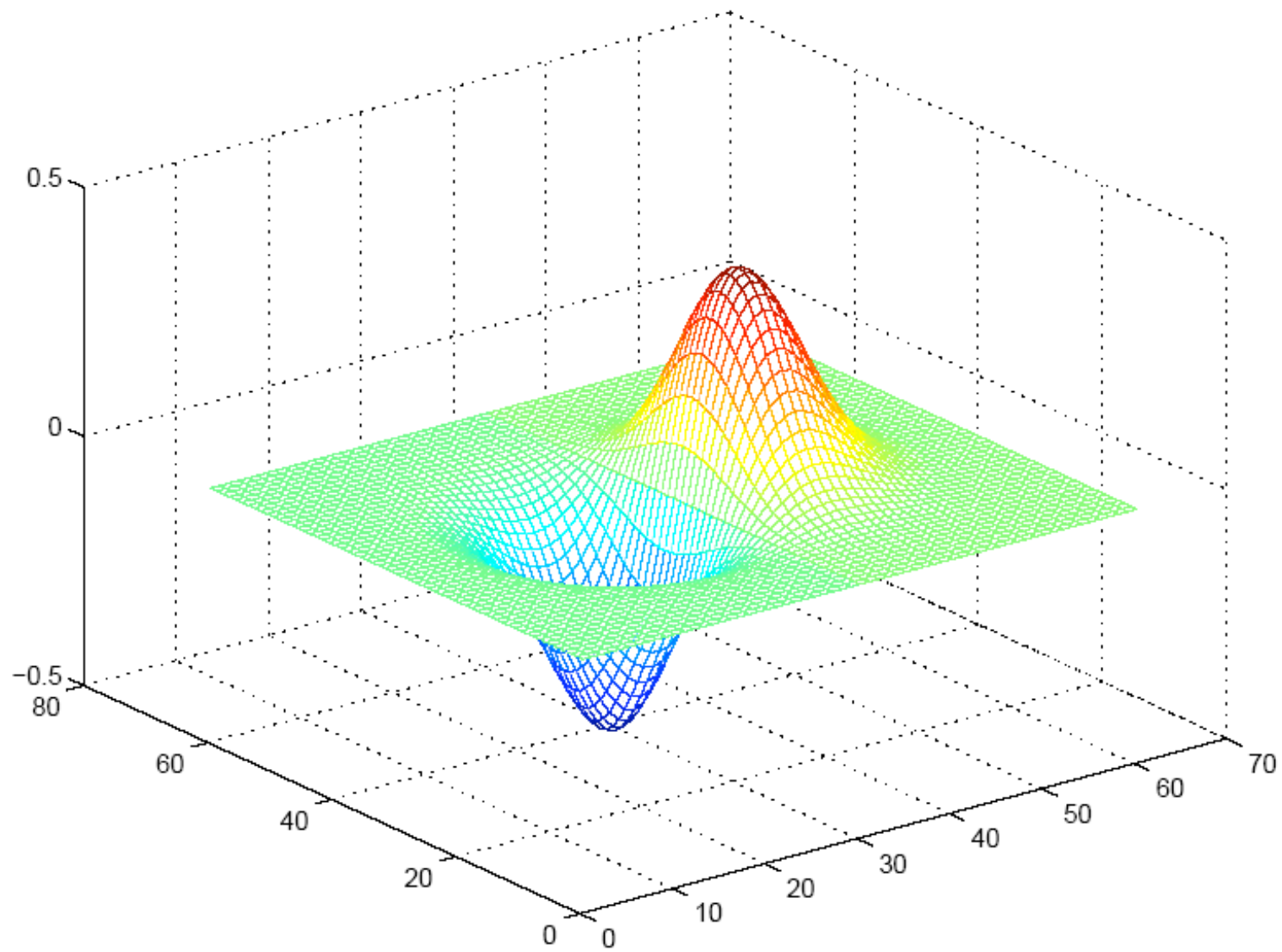
$$\lim_{\sigma \rightarrow 0} \hat{\mu}(t) = \int \delta(t - s)X(s) \, ds = X(t)$$

$$\lim_{\sigma \rightarrow \infty} \hat{\mu}(t) = 0$$

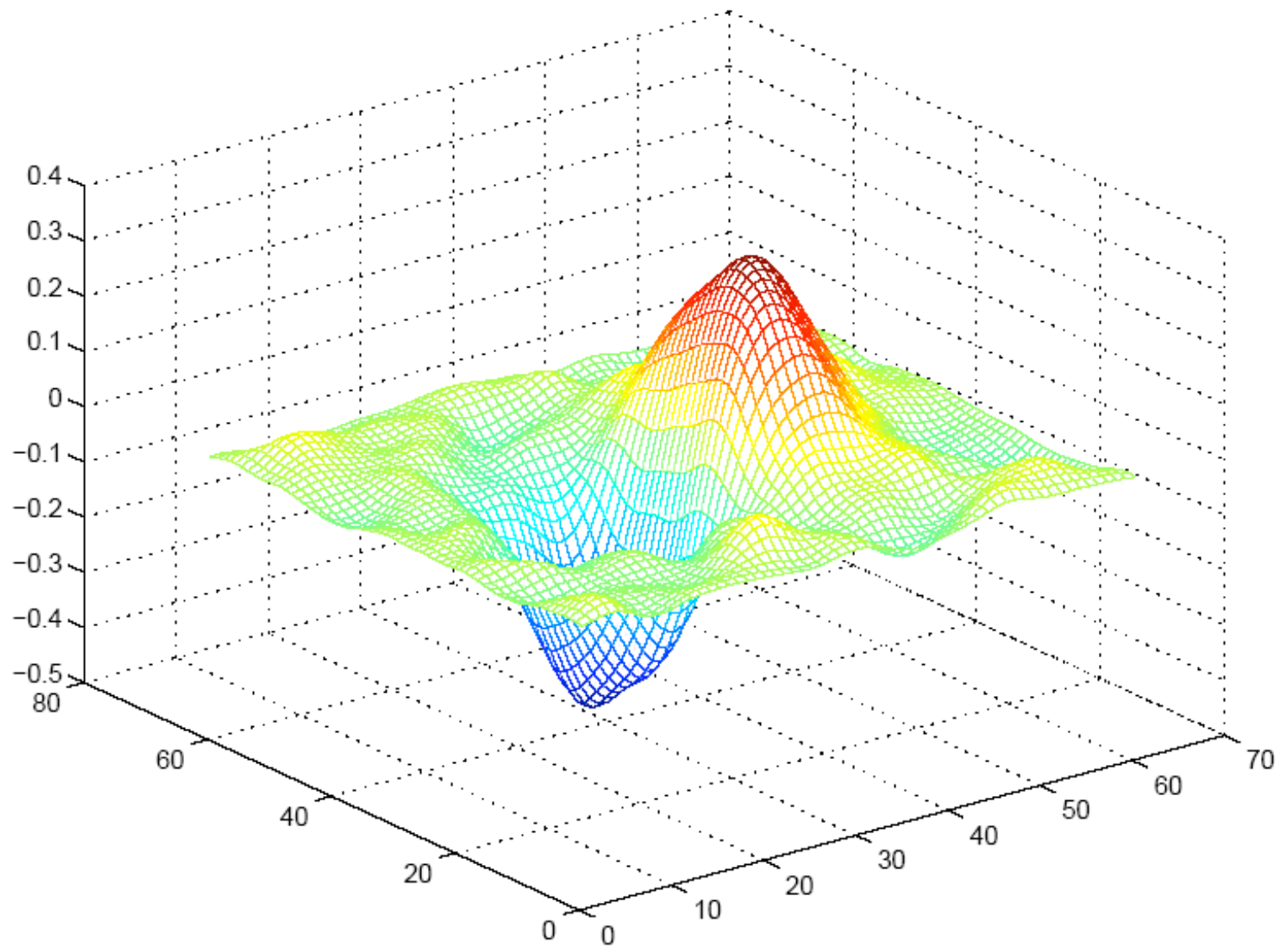
$$\inf \mu(t) \leq \mathbb{E} \hat{\mu}(t) \leq \sup \mu(t)$$



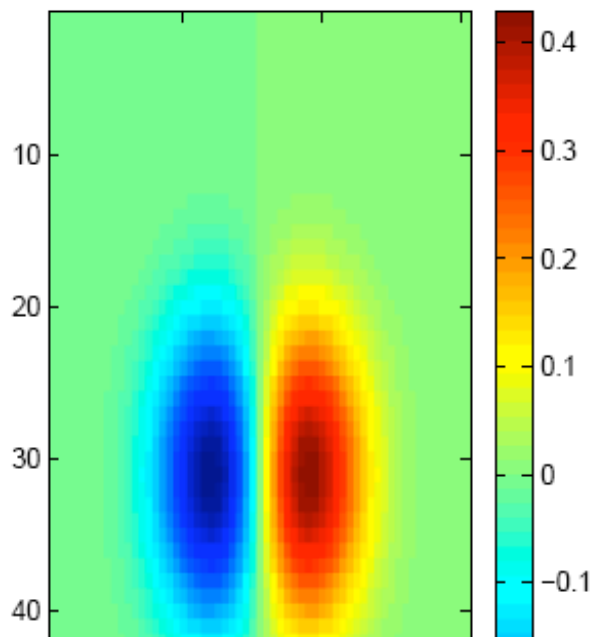
observation = signal + noise



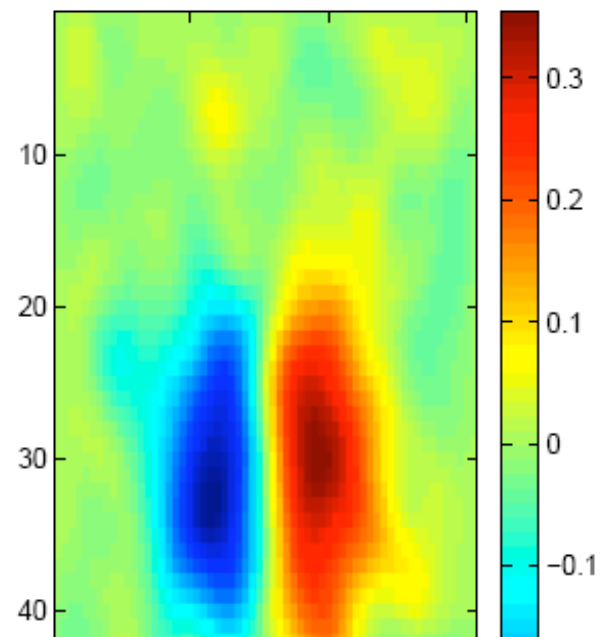
Signal



Prediction

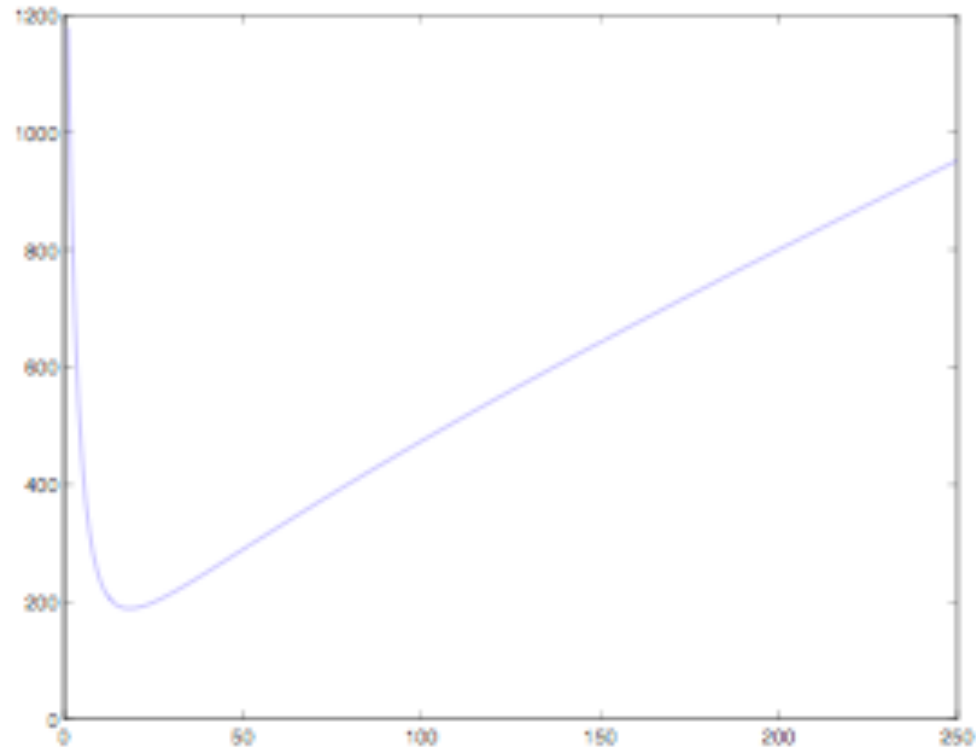


Signal



Prediction

Optimal bandwidth
choose sigma that minimizes the
integrated squared error



Many technique uses some sort of cross-validation

Iterated kernel smoothing

$$K_{m\sigma} = \underbrace{K_{\sigma} * \dots * K_{\sigma}}_{m^2 \text{ times}}$$

```
smoothedX=X;  
for i=1:100  
    smoothedX=conv2(smoothedX, weight,'same');  
end;  
Y=smoothedX;
```

Covariance function of random field

$$R(t, s) = \mathbb{E}[(e(t)e(s))]$$

$$\forall e(t) = R(t, t)$$

White noise = random field with Dirac-delta function as the covariance function.

Gaussian white noise = Gaussian + white noise

Dirac-delta function

This is not really a function in traditional mathematical sense.

How you construct numerically?

Let the bandwidth of isotropic Gaussian kernel goes to zero.

How to simulate Gaussian field

$$X(t) = K_\sigma * e(t)$$

Smooth field $\longleftarrow$ Gaussian white noise

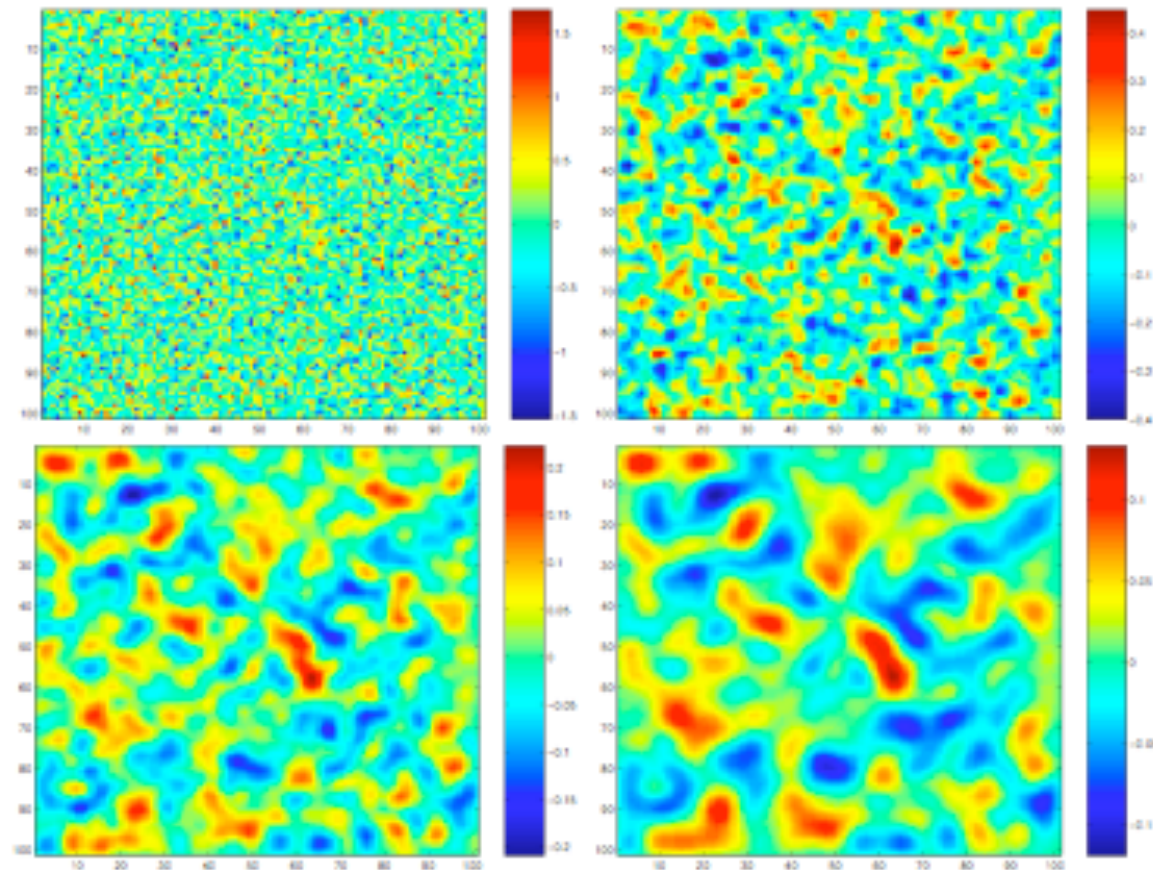
$$R_X(t, s) = \mathbb{E}[K_\sigma * e(t) K_\sigma * e(s)] = \int K_\sigma(t - u) K_\sigma(u - s) du.$$

How?

Numerical Implementation

```
e=normrnd(0,0.4,101,101);  
smooth_e=e;  
for i=1:10  
    smooth_e=conv2(smooth_e,K,'same');  
    figure;imagesc(smooth_e);colorbar;  
end;
```

Simulating Gaussian field



$N(0, 0.4^2)$ Gaussian white noise

Iterative kernel smoothing with $\sigma=0.4$ and 1, 4, 9 iterations

Lecture 10 Topics

Random Field Theory
Multiple Comparison Corrections