

Smoothing and Interpolation



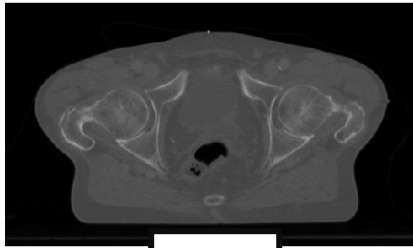
Aalto-yliopisto
Aalto-universitetet
Aalto University

Medical Image Analysis

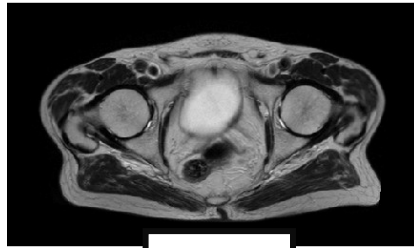
Koen Van Leemput

Why care about smoothness?

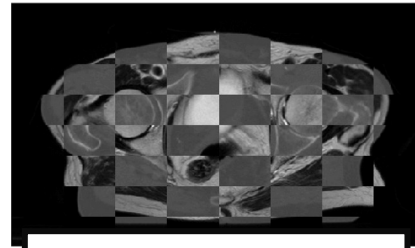
Example: image registration



CT



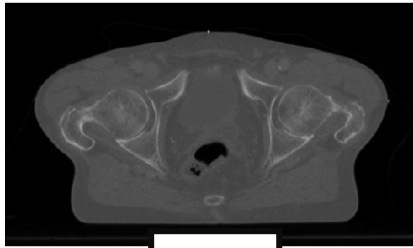
MRI



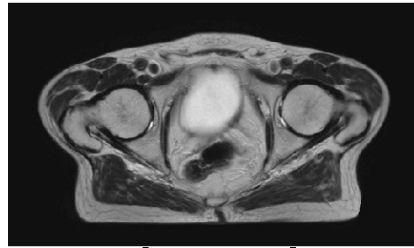
mosaic CT/MRI

Why care about smoothness?

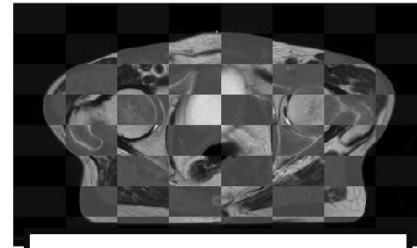
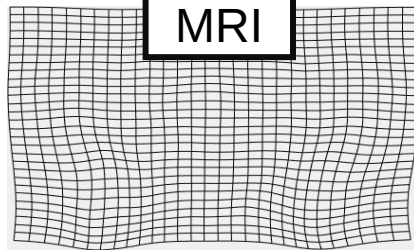
Example: image registration



CT



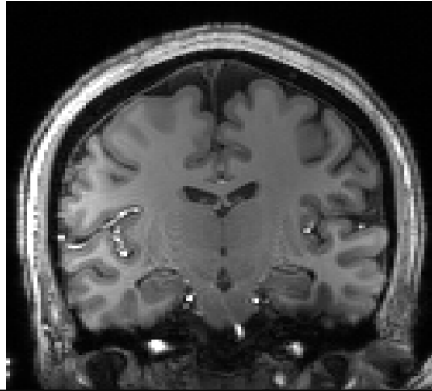
MRI



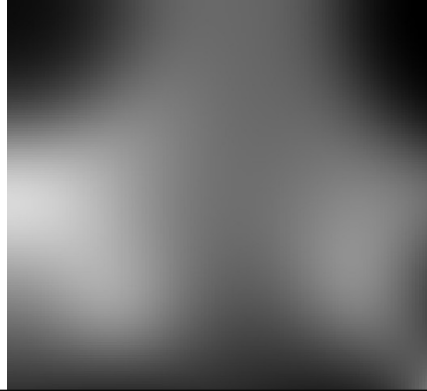
mosaic CT/MRI

Why care about smoothness?

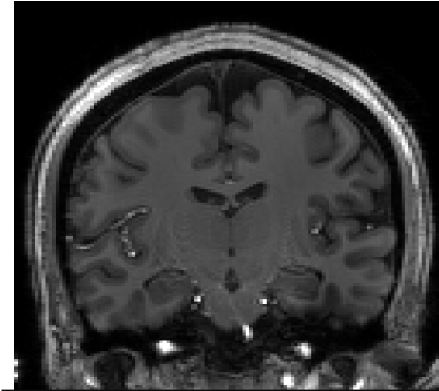
Example: image segmentation



MR scan (7 Tesla)



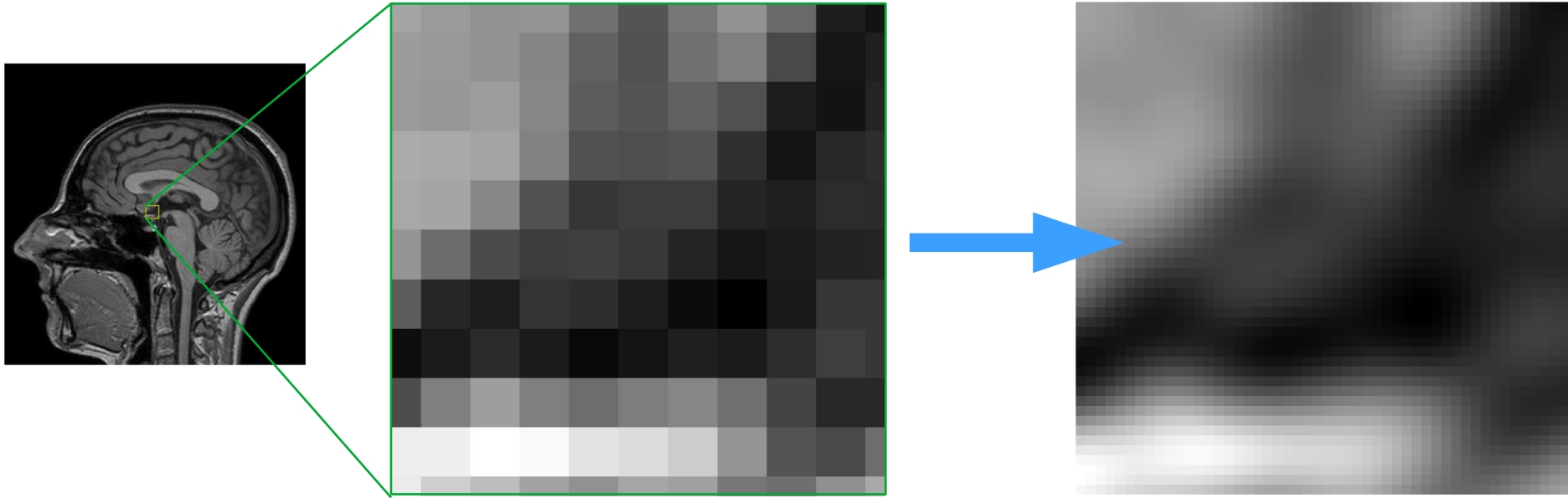
estimated "bias field"



bias field corrected

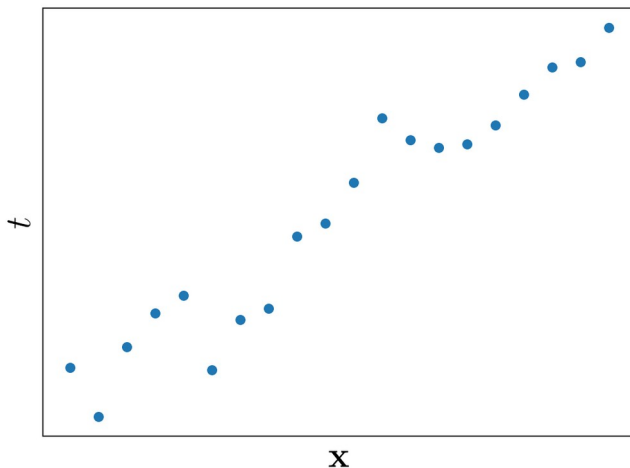
Why care about smoothness?

Example: image interpolation



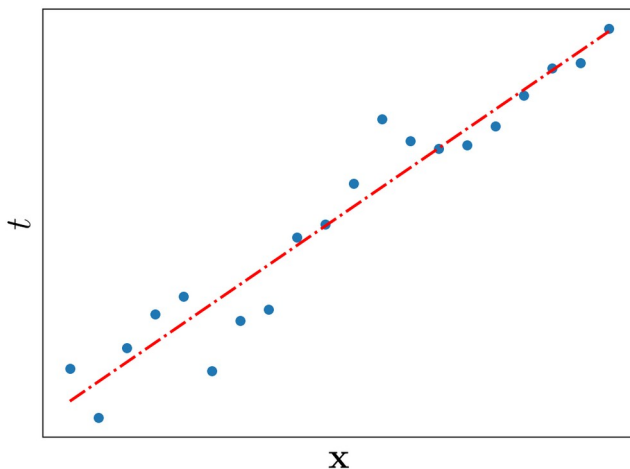
Linear regression

- ✓ Let $\mathbf{x} = (x_1, \dots, x_D)^T$ denote a spatial position in a D -dimensional space
- ✓ Given N measurements $\{t_n\}_{n=1}^N$ at locations $\{\mathbf{x}_n\}_{n=1}^N$, what is t at a new location \mathbf{x} ?



Linear regression

- ✓ Let $\mathbf{x} = (x_1, \dots, x_D)^T$ denote a spatial position in a D -dimensional space
- ✓ Given N measurements $\{t_n\}_{n=1}^N$ at locations $\{\mathbf{x}_n\}_{n=1}^N$, what is t at a new location \mathbf{x} ?

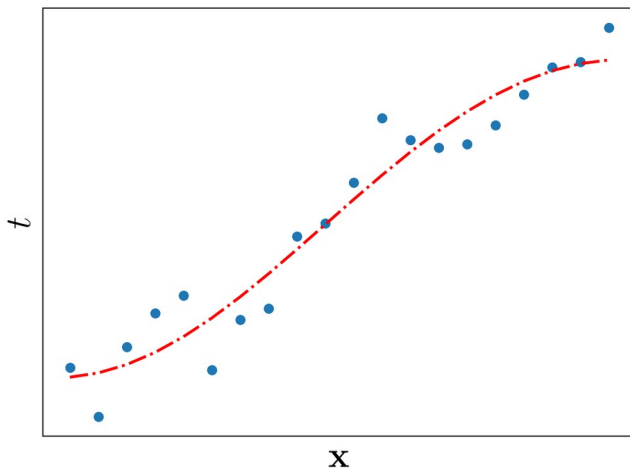


$$y(\mathbf{x}, \mathbf{w}) = w_0 + w_1 x_1 + \dots + w_D x_D$$

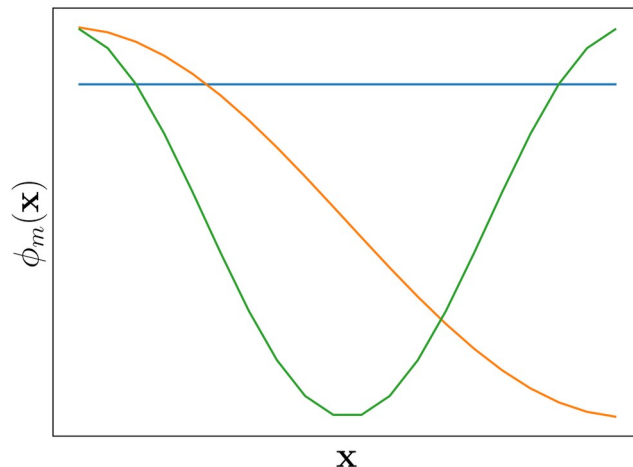
tunable weights

Linear regression

- ✓ Let $\mathbf{x} = (x_1, \dots, x_D)^T$ denote a spatial position in a D -dimensional space
- ✓ Given N measurements $\{t_n\}_{n=1}^N$ at locations $\{\mathbf{x}_n\}_{n=1}^N$, what is t at a new location \mathbf{x} ?



nonlinear basis functions



$$y(\mathbf{x}, \mathbf{w}) = \sum_{m=0}^{M-1} w_m \phi_m(\mathbf{x})$$

tunable weights

Linear regression

✓ What are “suitable” values for the weights $\mathbf{w} = (w_0, \dots, w_{M-1})^T$?

✓ Minimize the energy $E(\mathbf{w}) = \sum_{n=1}^N \left(t_n - \sum_{m=0}^{M-1} w_m \phi_m(\mathbf{x}_n) \right)^2$

Linear regression

✓ What are “suitable” values for the weights $\mathbf{w} = (w_0, \dots, w_{M-1})^T$?

✓ Minimize the energy $E(\mathbf{w}) = \sum_{n=1}^N \left(t_n - \sum_{m=0}^{M-1} w_m \phi_m(\mathbf{x}_n) \right)^2$

Task: find w that minimizes $E(w) = (5 - 4w)^2 + (3 - 2w)^2$

Linear regression

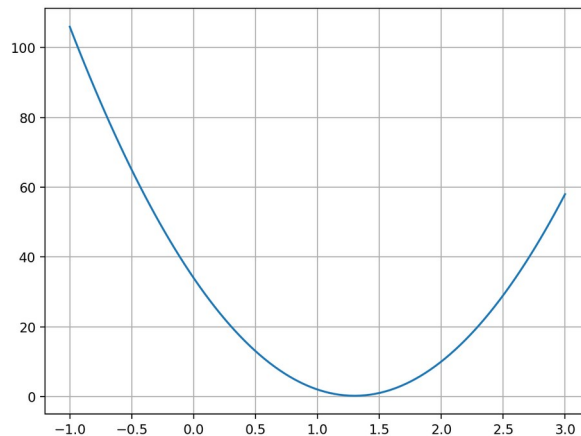
✓ What are “suitable” values for the weights $\mathbf{w} = (w_0, \dots, w_{M-1})^T$?

✓ Minimize the energy $E(\mathbf{w}) = \sum_{n=1}^N \left(t_n - \sum_{m=0}^{M-1} w_m \phi_m(\mathbf{x}_n) \right)^2$

Task: find w that minimizes $E(w) = (5 - 4w)^2 + (3 - 2w)^2$

$$\frac{dE(w)}{dw} = -8(5 - 4w) - 4(3 - 2w) = -52 + 40w$$

$$\frac{dE(w)}{dw} = 0 \quad \Rightarrow \quad w = 1.3$$



Linear regression

✓ What are “suitable” values for the weights $\mathbf{w} = (w_0, \dots, w_{M-1})^T$?

✓ Minimize the energy $E(\mathbf{w}) = \sum_{n=1}^N \left(t_n - \sum_{m=0}^{M-1} w_m \phi_m(\mathbf{x}_n) \right)^2$

Linear regression

✓ What are “suitable” values for the weights $\mathbf{w} = (w_0, \dots, w_{M-1})^T$?

✓ Minimize the energy $E(\mathbf{w}) = \sum_{n=1}^N \left(t_n - \sum_{m=0}^{M-1} w_m \phi_m(\mathbf{x}_n) \right)^2$

$$\frac{\partial E(\mathbf{w})}{\partial w_m} = -2 \sum_{n=1}^N \left(t_n - \sum_{m=0}^{M-1} w_m \phi_m(\mathbf{x}_n) \right) \phi_m(\mathbf{x}_n)$$

Linear regression

✓ What are “suitable” values for the weights $\mathbf{w} = (w_0, \dots, w_{M-1})^T$?

✓ Minimize the energy $E(\mathbf{w}) = \sum_{n=1}^N \left(t_n - \sum_{m=0}^{M-1} w_m \phi_m(\mathbf{x}_n) \right)^2$

$$\nabla E(\mathbf{w}) = \begin{pmatrix} \frac{\partial E(\mathbf{w})}{\partial w_0} \\ \vdots \\ \frac{\partial E(\mathbf{w})}{\partial w_{M-1}} \end{pmatrix} = -2\Phi^T (\mathbf{t} - \Phi\mathbf{w}), \quad \Phi = \begin{pmatrix} \phi_0(\mathbf{x}_1) & \phi_1(\mathbf{x}_1) & \dots & \phi_{M-1}(\mathbf{x}_1) \\ \phi_0(\mathbf{x}_2) & \phi_1(\mathbf{x}_2) & \dots & \phi_{M-1}(\mathbf{x}_2) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_0(\mathbf{x}_N) & \phi_1(\mathbf{x}_N) & \dots & \phi_{M-1}(\mathbf{x}_N) \end{pmatrix}$$

$$\mathbf{t} = (t_1, \dots, t_N)^T$$

$$\mathbf{w} = \left(\Phi^T \Phi \right)^{-1} \Phi^T \mathbf{t}$$

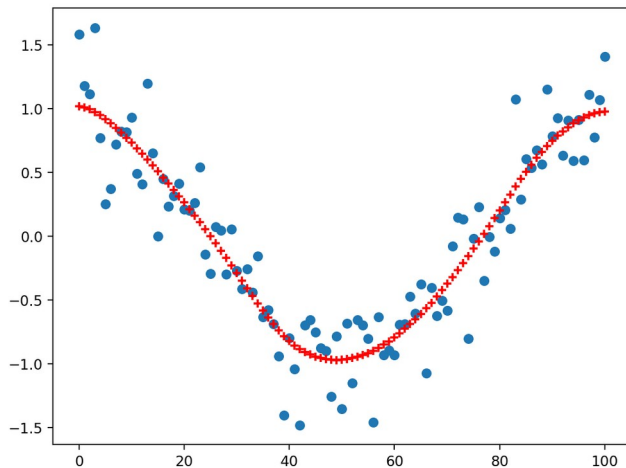
Smoothing

Let's concentrate on one-dimensional (1D) “images”:

- ✓ Functions of the form $y(x, \mathbf{w}) = \sum_{m=0}^{M-1} w_m \phi_m(x)$, where the location x is a scalar
- ✓ Measurement points are defined on a regular grid: $x_1=0, x_2=1, \dots, x_N=N-1$

“Denoising”:

- ✓ The measurements $t_n, n = 1, \dots, N$ are noisy observations
- ✓ Recover the underlying signal $\hat{t}_n = y(x_n, \mathbf{w})$ at the locations x_n



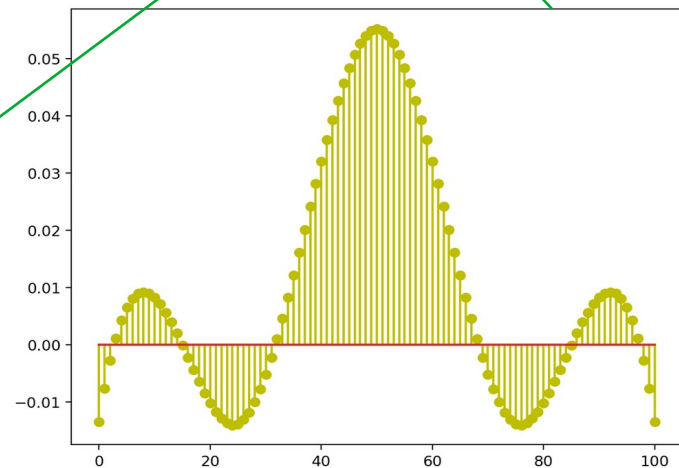
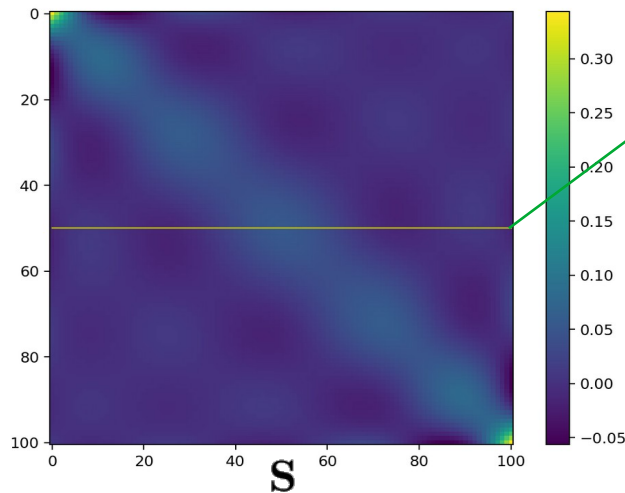
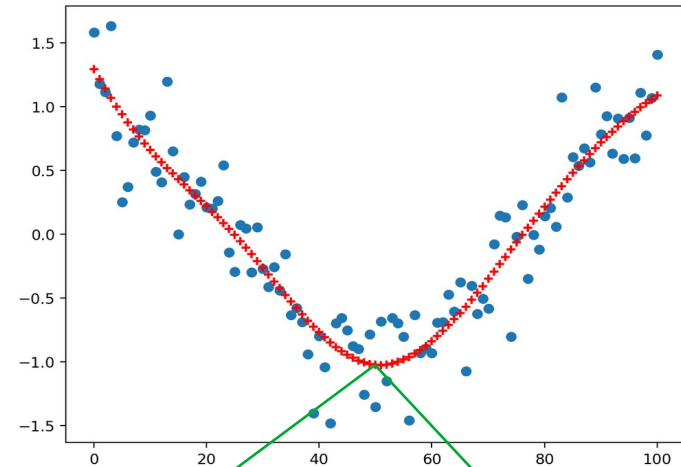
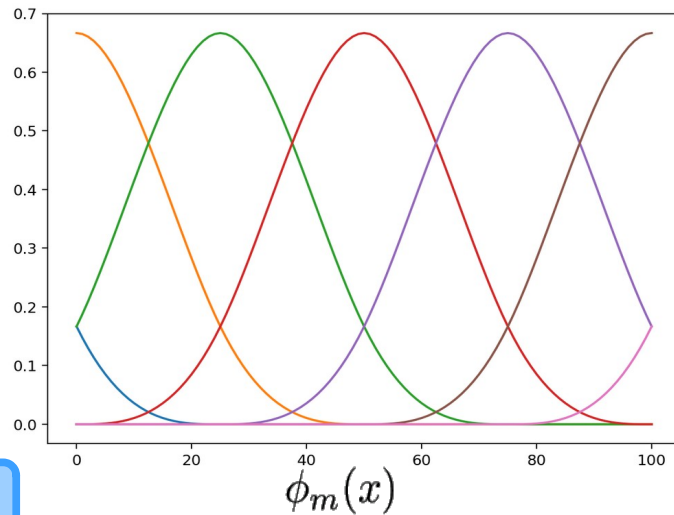
Smoothing

- ✓ We aim to recover $\hat{\mathbf{t}} = (\hat{t}_1, \dots, \hat{t}_N)^T$ from $\mathbf{t} = (t_1, \dots, t_N)^T$
- ✓ Since $\hat{\mathbf{t}} = \Phi \mathbf{w}$ and $\mathbf{w} = (\Phi^T \Phi)^{-1} \Phi^T \mathbf{t}$:

$$\hat{\mathbf{t}} = \mathbf{S} \mathbf{t} \text{ with } \mathbf{S} = \Phi (\Phi^T \Phi)^{-1} \Phi^T$$

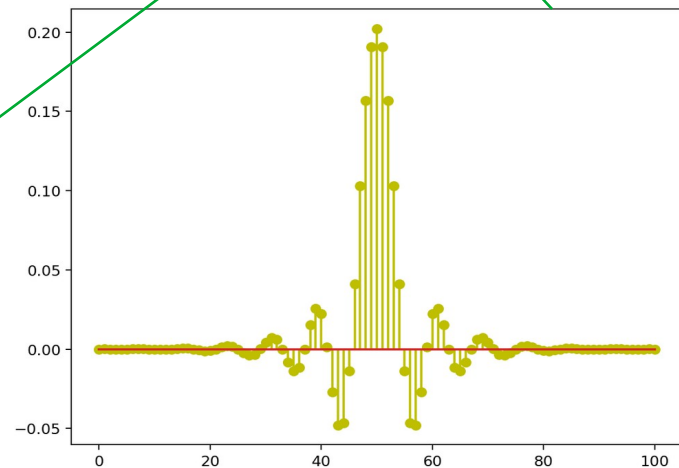
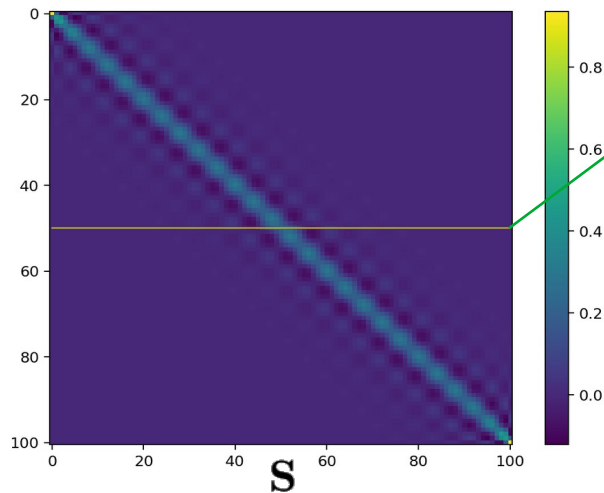
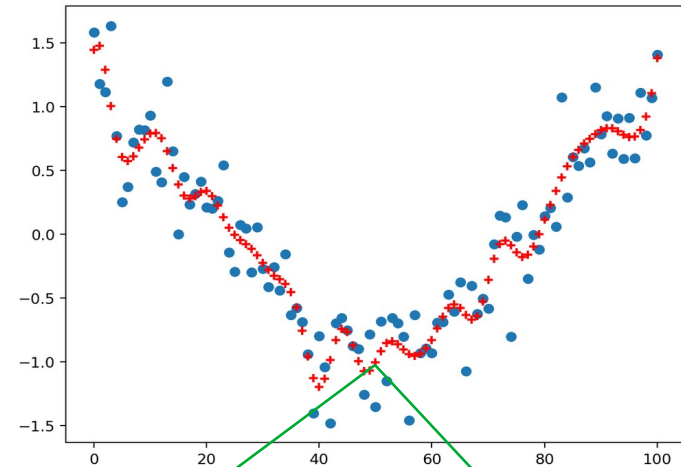
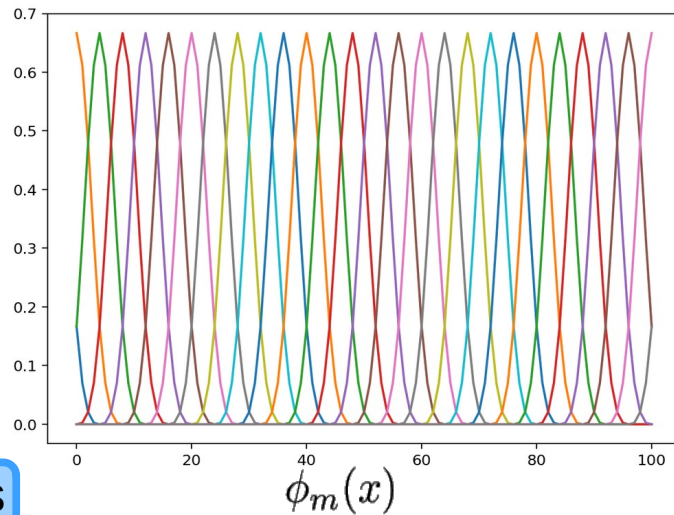
Example

$M=7$ basis functions



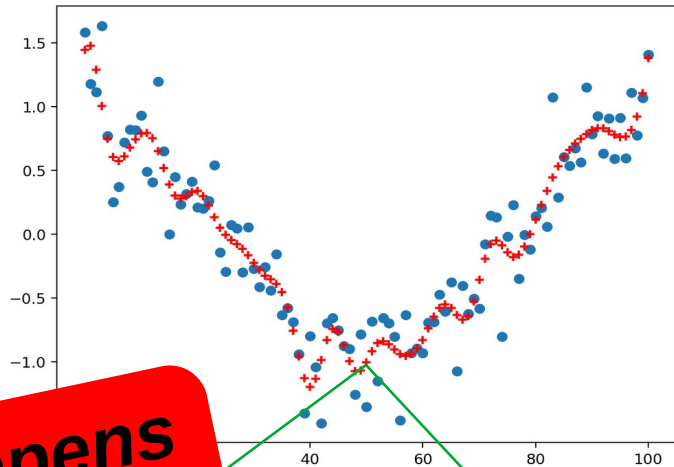
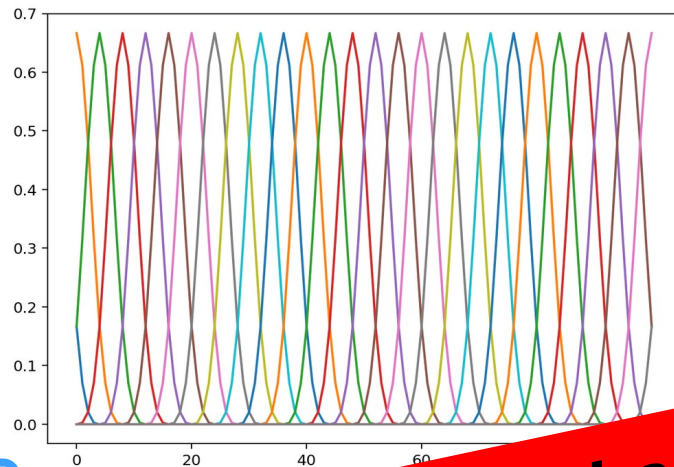
Example

$M=28$ basis functions

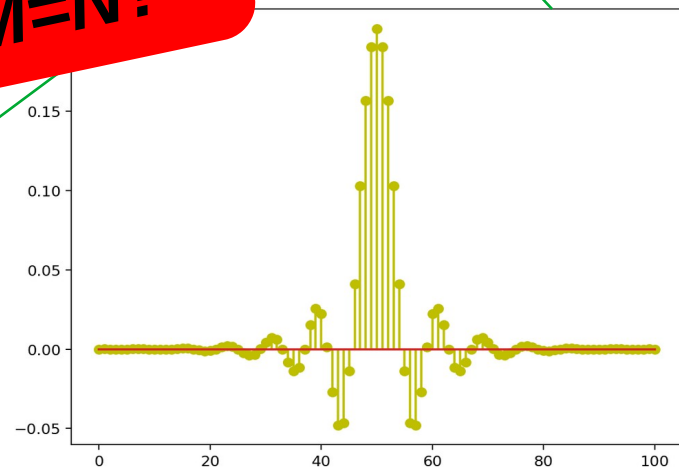
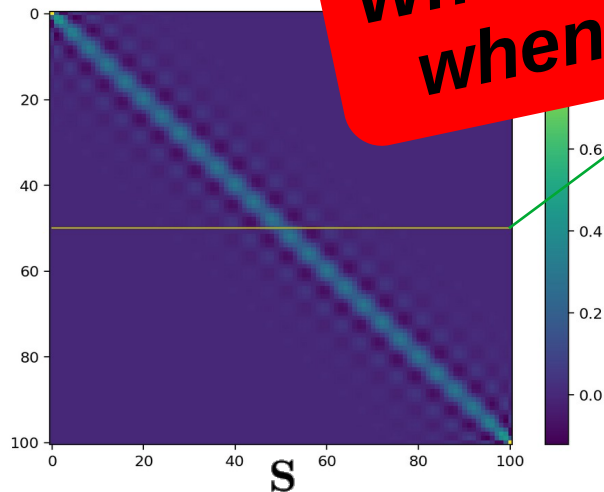


Example

$M=28$ basis functions



What happens
when $M=N$?



Discuss with your neighbor

Consider the case $\mathbf{t} = (2, 1, 3)^T$ and $\Phi = (1, 1, 1)^T$

Task 1: what is $\mathbf{w} = \left(\Phi^T \Phi\right)^{-1} \Phi^T \mathbf{t}$?

Task 2: what are $\hat{\mathbf{t}} = \mathbf{S} \mathbf{t}$ and $\mathbf{S} = \Phi \left(\Phi^T \Phi\right)^{-1} \Phi^T$?

Can you explain (e.g., draw) what's happening?

Task 3: same as task 2 but when $\Phi = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 2 & 4 \\ 1 & 3 & 9 \end{pmatrix}$

Discuss with your neighbor

Consider the case $\mathbf{t} = (2, 1, 3)^T$ and $\Phi = (1, 1, 1)^T$

Task 1: what is $\mathbf{w} = \left(\Phi^T \Phi\right)^{-1} \Phi^T \mathbf{t}$?

Task 2: what are $\hat{\mathbf{t}} = \mathbf{S} \mathbf{t}$ and $\mathbf{S} = \Phi \left(\Phi^T \Phi\right)^{-1} \Phi^T$?

Can you explain (e.g., draw) what's happening?

Task 3: same as task 2 but when $\Phi = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 2 & 4 \\ 1 & 3 & 9 \end{pmatrix}$

Discuss with your neighbor

Consider the case $\mathbf{t} = (2, 1, 3)^T$ and $\Phi = (1, 1, 1)^T$

Task 1: what is $\mathbf{w} = (\Phi^T \Phi)^{-1} \Phi^T \mathbf{t}$?

$$\mathbf{w} = 3^{-1}6 = 2$$

Task 2: what are $\hat{\mathbf{t}} = \mathbf{S}\mathbf{t}$ and $\mathbf{S} = \Phi (\Phi^T \Phi)^{-1} \Phi^T$?

Can you explain (e.g., draw) what's happening?

Task 3: same as task 2 but when $\Phi = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 2 & 4 \\ 1 & 3 & 9 \end{pmatrix}$

Discuss with your neighbor

Consider the case $\mathbf{t} = (2, 1, 3)^T$ and $\Phi = (1, 1, 1)^T$

Task 1: what is $\mathbf{w} = \left(\Phi^T \Phi\right)^{-1} \Phi^T \mathbf{t}$?

Task 2: what are $\hat{\mathbf{t}} = \mathbf{S} \mathbf{t}$ and $\mathbf{S} = \Phi \left(\Phi^T \Phi\right)^{-1} \Phi^T$?

Can you explain (e.g., draw) what's happening?

Task 3: same as task 2 but when $\Phi = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 2 & 4 \\ 1 & 3 & 9 \end{pmatrix}$

Discuss with your neighbor

Consider the case $\mathbf{t} = (2, 1, 3)^T$ and $\Phi = (1, 1, 1)^T$

Task 1: what is $\mathbf{w} = (\Phi^T \Phi)^{-1} \Phi^T \mathbf{t}$?

Task 2: what are $\hat{\mathbf{t}} = \mathbf{S} \mathbf{t}$ and $\mathbf{S} = \Phi (\Phi^T \Phi)^{-1} \Phi^T$?

Can you explain (e.g., draw) what's happening?

Task 3: same as task 2 but when $\Phi = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 2 & 4 \\ 1 & 3 & 9 \end{pmatrix}$

$$\hat{\mathbf{t}} = \begin{pmatrix} 2 \\ 2 \\ 2 \end{pmatrix}$$

$$\begin{aligned} \mathbf{S} &= \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \cdot 3^{-1} \cdot (1 \ 1 \ 1) \\ &= \begin{pmatrix} 1/3 & 1/3 & 1/3 \\ 1/3 & 1/3 & 1/3 \\ 1/3 & 1/3 & 1/3 \end{pmatrix} \end{aligned}$$

Discuss with your neighbor

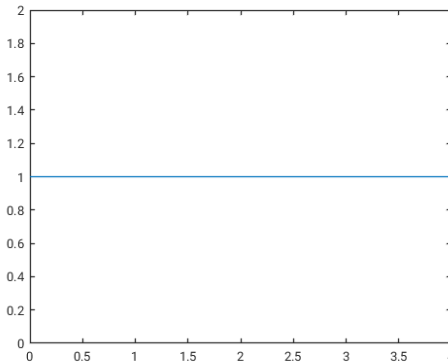
Consider the case $\mathbf{t} = (2, 1, 3)^T$ and $\Phi = (1, 1, 1)^T$

Task 1: what is $\mathbf{w} = (\Phi^T \Phi)^{-1} \Phi^T \mathbf{t}$?

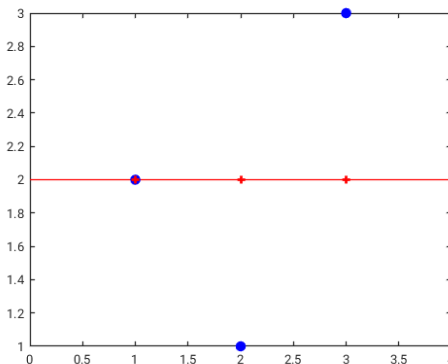
Task 2: what are $\hat{\mathbf{t}} = \mathbf{S} \mathbf{t}$ and $\mathbf{S} = \Phi (\Phi^T \Phi)^{-1} \Phi^T$?

Can you explain (e.g., draw) what's happening?

Task 3: same as task 2 but when $\Phi = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 2 & 4 \\ 1 & 3 & 9 \end{pmatrix}$



$\phi_m(x)$



Discuss with your neighbor

Consider the case $\mathbf{t} = (2, 1, 3)^T$ and $\Phi = (1, 1, 1)^T$

Task 1: what is $\mathbf{w} = \left(\Phi^T \Phi\right)^{-1} \Phi^T \mathbf{t}$?

Task 2: what are $\hat{\mathbf{t}} = \mathbf{S} \mathbf{t}$ and $\mathbf{S} = \Phi \left(\Phi^T \Phi\right)^{-1} \Phi^T$?

Can you explain (e.g., draw) what's happening?

Task 3: same as task 2 but when $\Phi = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 2 & 4 \\ 1 & 3 & 9 \end{pmatrix}$

Discuss with your neighbor

Consider the case $\mathbf{t} = (2, 1, 3)^T$ and $\Phi = (1, 1, 1)^T$

Task 1: what is $\mathbf{w} = (\Phi^T \Phi)^{-1} \Phi^T \mathbf{t}$?

Task 2: what are $\hat{\mathbf{t}} = \mathbf{S} \mathbf{t}$ and $\mathbf{S} = \Phi (\Phi^T \Phi)^{-1} \Phi^T$?

Can you explain (e.g., draw) what's happening?

Task 3: same as task 2 but when $\Phi = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 2 & 4 \\ 1 & 3 & 9 \end{pmatrix}$

$$(\Phi^T \Phi)^{-1} = \Phi^{-1} (\Phi^T)^{-1}$$

$$\mathbf{S} = \overbrace{\Phi \Phi^{-1}}^{\mathbf{I}} \underbrace{(\Phi^T)^{-1} \Phi^T}_{\mathbf{I}}$$

$$= \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$\hat{\mathbf{t}} = \mathbf{t} = \begin{pmatrix} 2 \\ 1 \\ 3 \end{pmatrix}$$

Discuss with your neighbor

Consider the case $\mathbf{t} = (2, 1, 3)^T$ and $\Phi = (1, 1, 1)^T$

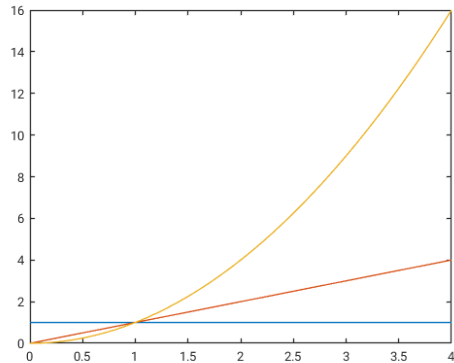
Task 1: what is $\mathbf{w} = (\Phi^T \Phi)^{-1} \Phi^T \mathbf{t}$?

Task 2: what are $\hat{\mathbf{t}} = \mathbf{S} \mathbf{t}$ and $\mathbf{S} = \Phi (\Phi^T \Phi)^{-1} \Phi^T$?

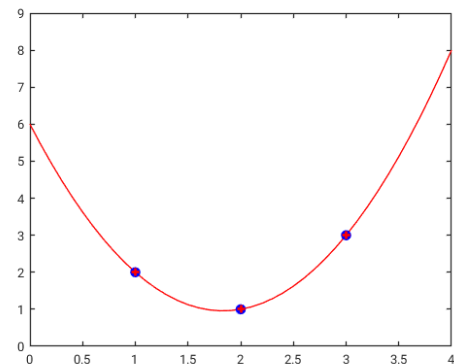
Can you explain (e.g., draw) what's happening?

Task 3: same as task 2 but when $\Phi = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 2 & 4 \\ 1 & 3 & 9 \end{pmatrix}$

A? *When $M=N$,
no smoothing is applied!*



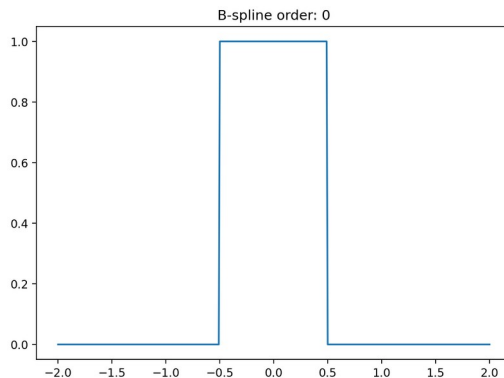
$\phi_m(x)$



Interpolation

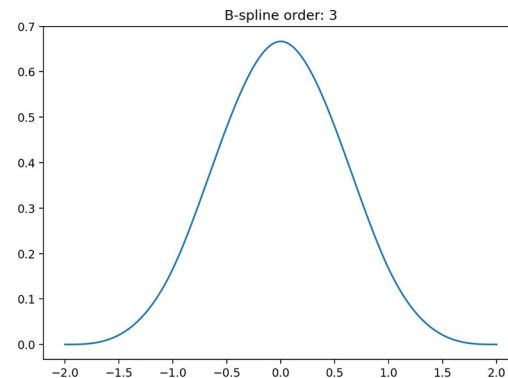
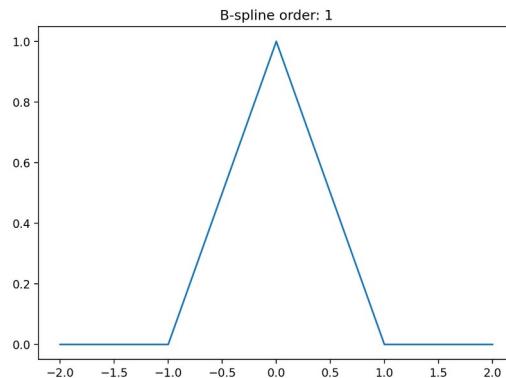
Meet the B-spline family:

$$\checkmark \quad \beta^0(x) = \begin{cases} 1, & -\frac{1}{2} < x < \frac{1}{2} \\ \frac{1}{2}, & |x| = \frac{1}{2} \\ 0, & \text{otherwise,} \end{cases}$$



$$\checkmark \quad \beta^p(x) = \underbrace{(\beta^0 * \beta^0 * \dots * \beta^0)}_{(p+1) \text{ times}}(x) \quad \text{where} \quad (f * g)(x) = \int_{\tau=-\infty}^{\infty} f(\tau)g(x - \tau)d\tau$$

“order”

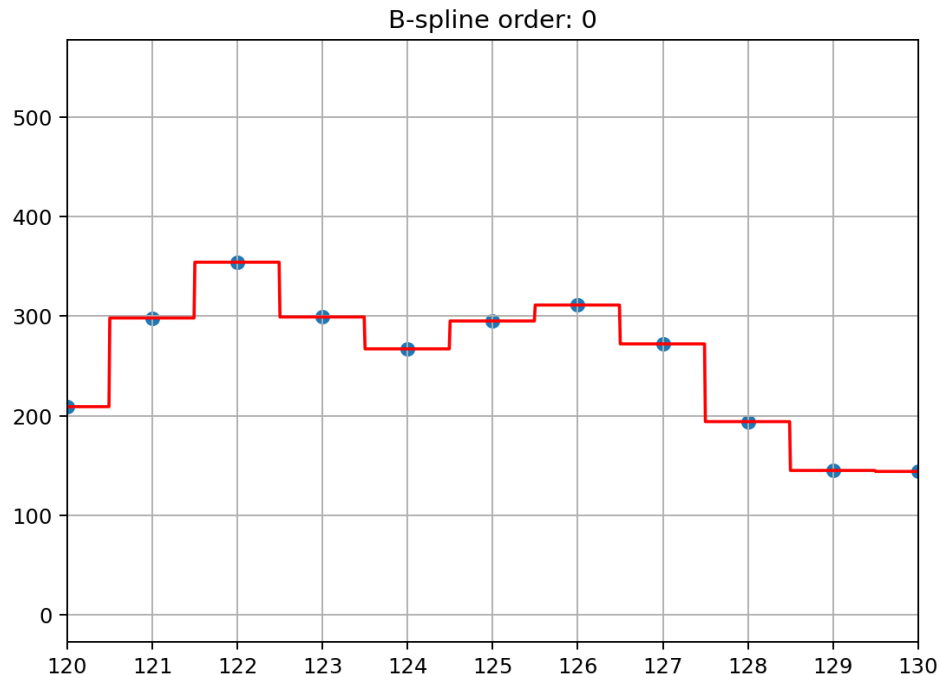


Interpolation

Use $M = N$ basis functions: $\phi_m(x) = \beta^p(x - m)$, $m = 0, \dots, N - 1$

✓ $y(x, \mathbf{w})$ will pass exactly through t_n at the integer locations x_n

Order 0: “nearest neighbor” interpolation
(almost)

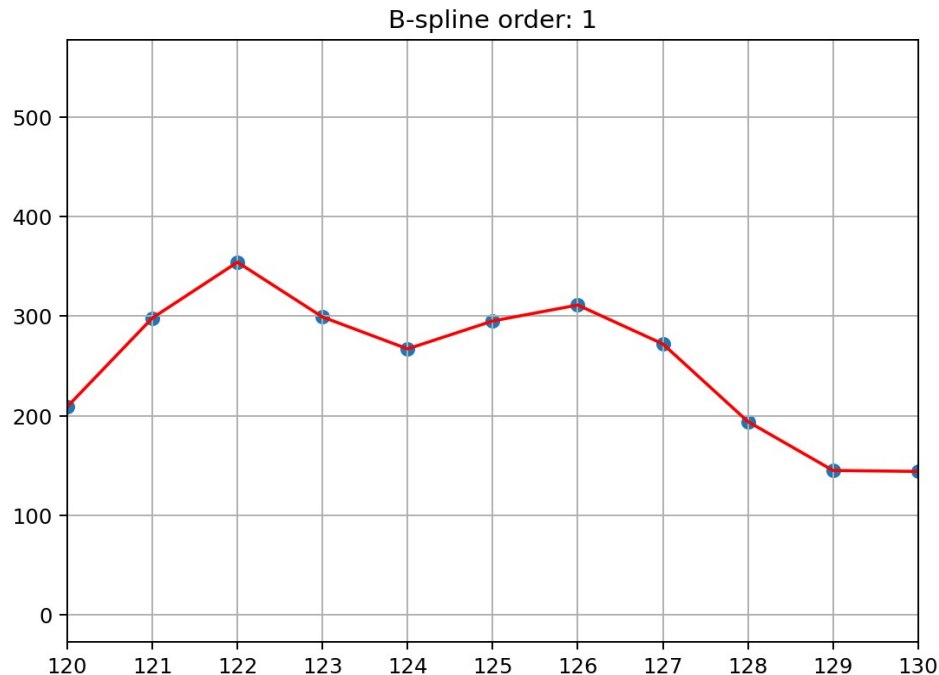


Interpolation

Use $M = N$ basis functions: $\phi_m(x) = \beta^p(x - m)$, $m = 0, \dots, N - 1$

✓ $y(x, \mathbf{w})$ will pass exactly through t_n at the integer locations x_n

Order 1: “linear” interpolation

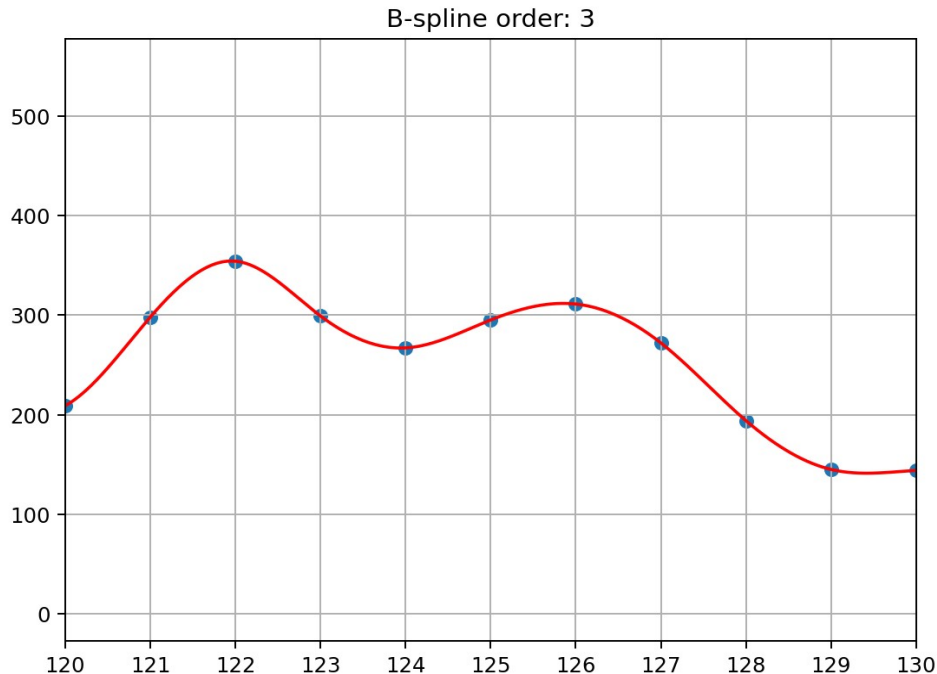


Interpolation

Use $M = N$ basis functions: $\phi_m(x) = \beta^p(x - m)$, $m = 0, \dots, N - 1$

✓ $y(x, \mathbf{w})$ will pass exactly through t_n at the integer locations x_n

Order 3: “cubic” interpolation



Going to higher dimensions

Re-arrange pixels in 2D images of size $N_1 \times N_2$ into 1D signals of length $N = N_1 N_2$

- ✓ Vectorize the image to be smoothed or interpolated

$$\mathbf{T} = \begin{pmatrix} t_{1,1} & t_{1,2} & \cdots & t_{1,N_2} \\ t_{2,1} & t_{2,2} & \cdots & t_{2,N_2} \\ \vdots & \vdots & \ddots & \vdots \\ t_{N_1,1} & t_{N_1,2} & \cdots & t_{N_1,N_2} \end{pmatrix} \quad \longrightarrow \quad \mathbf{t} = \text{vec}(\mathbf{T}) = \begin{pmatrix} t_{1,1} \\ \vdots \\ t_{N_1,1} \\ t_{1,2} \\ \vdots \\ t_{N_1,2} \\ \vdots \\ t_{N_1,N_2} \end{pmatrix}$$

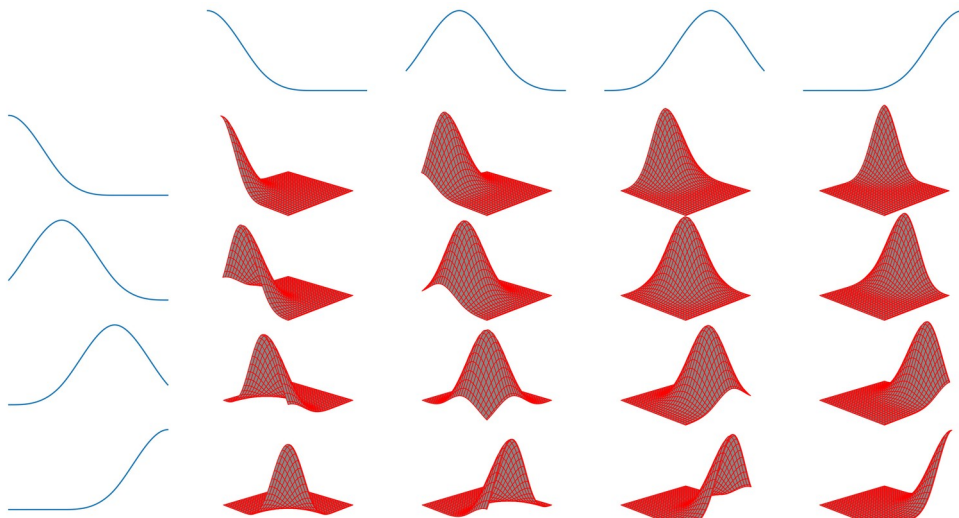
- ✓ Also vectorize each of the 2D basis functions

Solve the resulting 1D problem as before

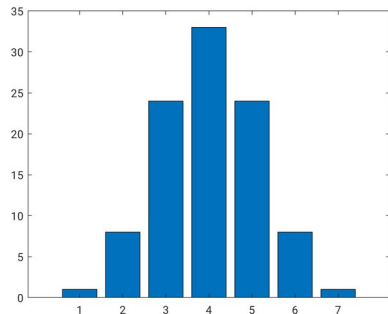
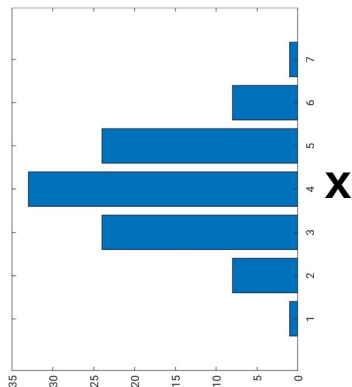
Separable basis functions

Create 2D basis functions from two sets of 1D basis functions

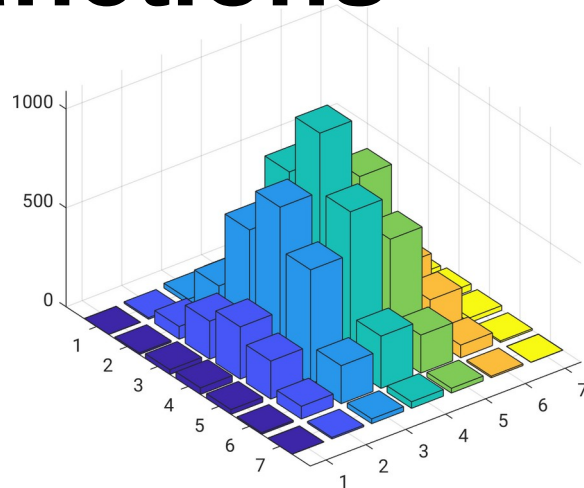
- ✓ Kronecker product: $\Phi = \Phi_2 \otimes \Phi_1$, where $\mathbf{A} \otimes \mathbf{B} = \left(\begin{array}{c|c|c} a_{1,1}\mathbf{B} & a_{2,1}\mathbf{B} & \dots \\ \hline a_{2,1}\mathbf{B} & a_{2,2}\mathbf{B} & \dots \\ \hline \vdots & \vdots & \ddots \end{array} \right)$
- ✓ Column $m = m_1 + m_2M_1$ in Φ contains a vectorized version of $\phi_m(\mathbf{x}) = \phi_{m_1}(x_1)\phi_{m_2}(x_2)$



Separable basis functions



=



$$\begin{pmatrix} 1 \\ 8 \\ 24 \\ 33 \\ 24 \\ 8 \\ 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 8 & 24 & 33 & 24 & 8 & 1 \end{pmatrix} =$$

$$\begin{pmatrix} 1 \cdot 1 & 1 \cdot 8 & 1 \cdot 24 & 1 \cdot 33 & 1 \cdot 24 & 1 \cdot 8 & 1 \cdot 1 \\ 8 \cdot 1 & 8 \cdot 8 & 8 \cdot 24 & 8 \cdot 33 & 8 \cdot 24 & 8 \cdot 8 & 8 \cdot 1 \\ 24 \cdot 1 & 24 \cdot 8 & 24 \cdot 24 & 24 \cdot 33 & 24 \cdot 24 & 24 \cdot 8 & 24 \cdot 1 \\ 33 \cdot 1 & 33 \cdot 8 & 33 \cdot 24 & 33 \cdot 33 & 33 \cdot 24 & 33 \cdot 8 & 33 \cdot 1 \\ 24 \cdot 1 & 24 \cdot 8 & 24 \cdot 24 & 24 \cdot 33 & 24 \cdot 24 & 24 \cdot 8 & 24 \cdot 1 \\ 8 \cdot 1 & 8 \cdot 8 & 8 \cdot 24 & 8 \cdot 33 & 8 \cdot 24 & 8 \cdot 8 & 8 \cdot 1 \\ 1 \cdot 1 & 1 \cdot 8 & 1 \cdot 24 & 1 \cdot 33 & 1 \cdot 24 & 1 \cdot 8 & 1 \cdot 1 \end{pmatrix}$$

Exploiting separability

- ✓ When $\Phi = \Phi_2 \otimes \Phi_1$, we can compute $\mathbf{c} = \Phi^T \mathbf{t}$ faster as follows:

$$\mathbf{C} = \Phi_1^T \mathbf{T} \Phi_2 \quad \text{where} \quad \text{vec}(\mathbf{C}) = \mathbf{c}$$

- ✓ As a result, we can also compute $\mathbf{w} = (\Phi^T \Phi)^{-1} \Phi^T \mathbf{t}$ as:

$$\mathbf{W} = (\Phi_1^T \Phi_1)^{-1} \Phi_1^T \mathbf{T} \Phi_2 (\Phi_2^T \Phi_2)^{-1} \quad \text{where} \quad \text{vec}(\mathbf{W}) = \mathbf{w}$$

Exploiting separability

- ✓ When $\Phi = \Phi_2 \otimes \Phi_1$, we can compute $\mathbf{c} = \Phi^T \mathbf{t}$ faster as follows:

$$\mathbf{C} = \Phi_1^T \mathbf{T} \Phi_2 \quad \text{where} \quad \text{vec}(\mathbf{C}) = \mathbf{c}$$

- ✓ As a result, we can also compute $\mathbf{w} = (\Phi^T \Phi)^{-1} \Phi^T \mathbf{t}$ as:

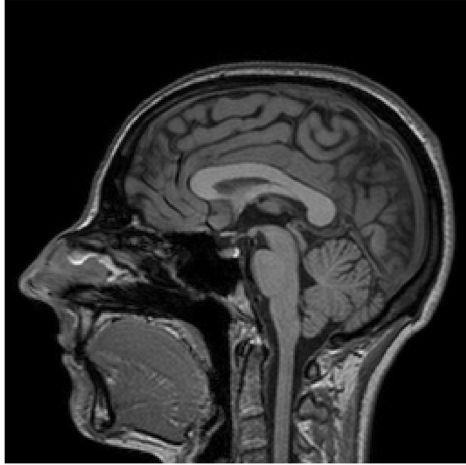
$$\mathbf{W} = (\Phi_1^T \Phi_1)^{-1} \Phi_1^T \mathbf{T} \Phi_2 (\Phi_2^T \Phi_2)^{-1} \quad \text{where} \quad \text{vec}(\mathbf{W}) = \mathbf{w}$$

Example: naively interpolating a 256x256 image ($M = 256 \times 256 = 65,536$ basis functions in 2D!)

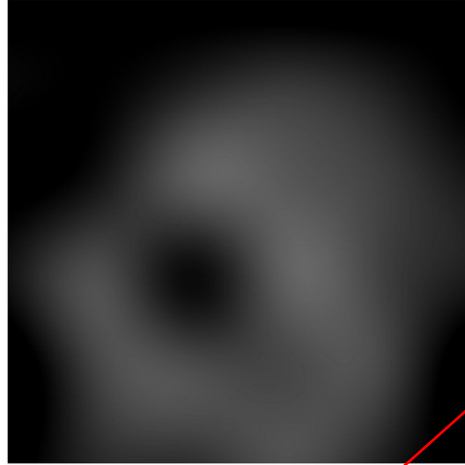
Storing $\Phi^T \Phi$ takes 32 GB, vs. 1MB to store both $\Phi_1^T \Phi_1$ and $\Phi_2^T \Phi_2$

Inverting $\Phi^T \Phi$ is almost **10 million times slower** than inverting both $\Phi_1^T \Phi_1$ and $\Phi_2^T \Phi_2$

Smoothing in 2D



T



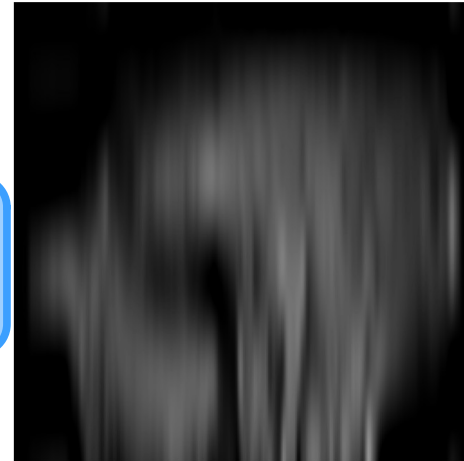
$$\hat{T} = S_1 T S_2^T$$

smoothing in the
column-direction

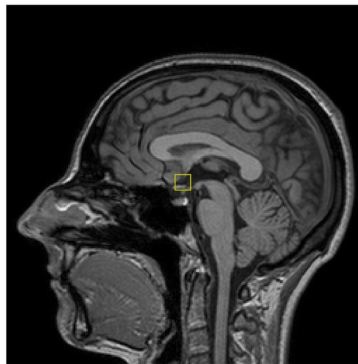


$$S_1 = \Phi_1 \left(\Phi_1^T \Phi_1 \right)^{-1} \Phi_1^T$$

smoothing in the **row-direction**



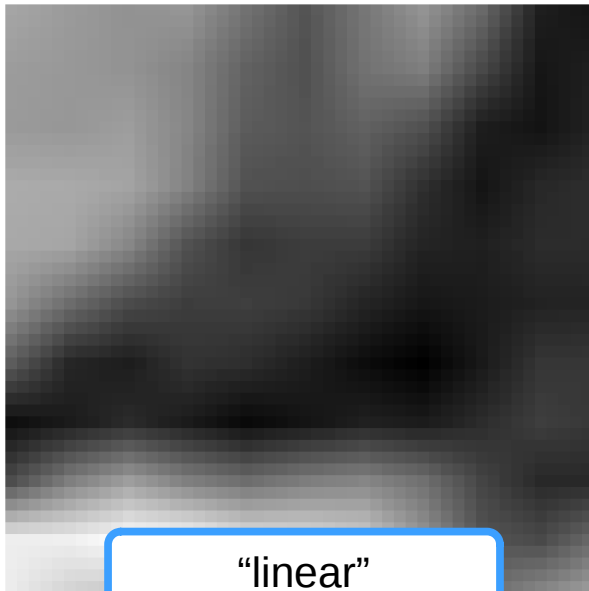
Interpolation in 2D



B-spline order: 0



B-spline order: 1



B-spline order: 3

