EE16B
Designing Information Devices and Systems II

Lecture 11A
Moore Penrose Inverse
Sampling and Interpolation
(Polynomial Interpolation)

Intro

- Last time:
  - Examples of PCA
  - Labeled vs non-labeled clustering
- Today
  - Finish the PCA/SVD Module
    Moore Penrose Pseudo-Inverse
  - New module: Sampling and interpolation
  - Polynomial interpolation

Labeled VS non labeled Classification

Word1

Word2

Word3

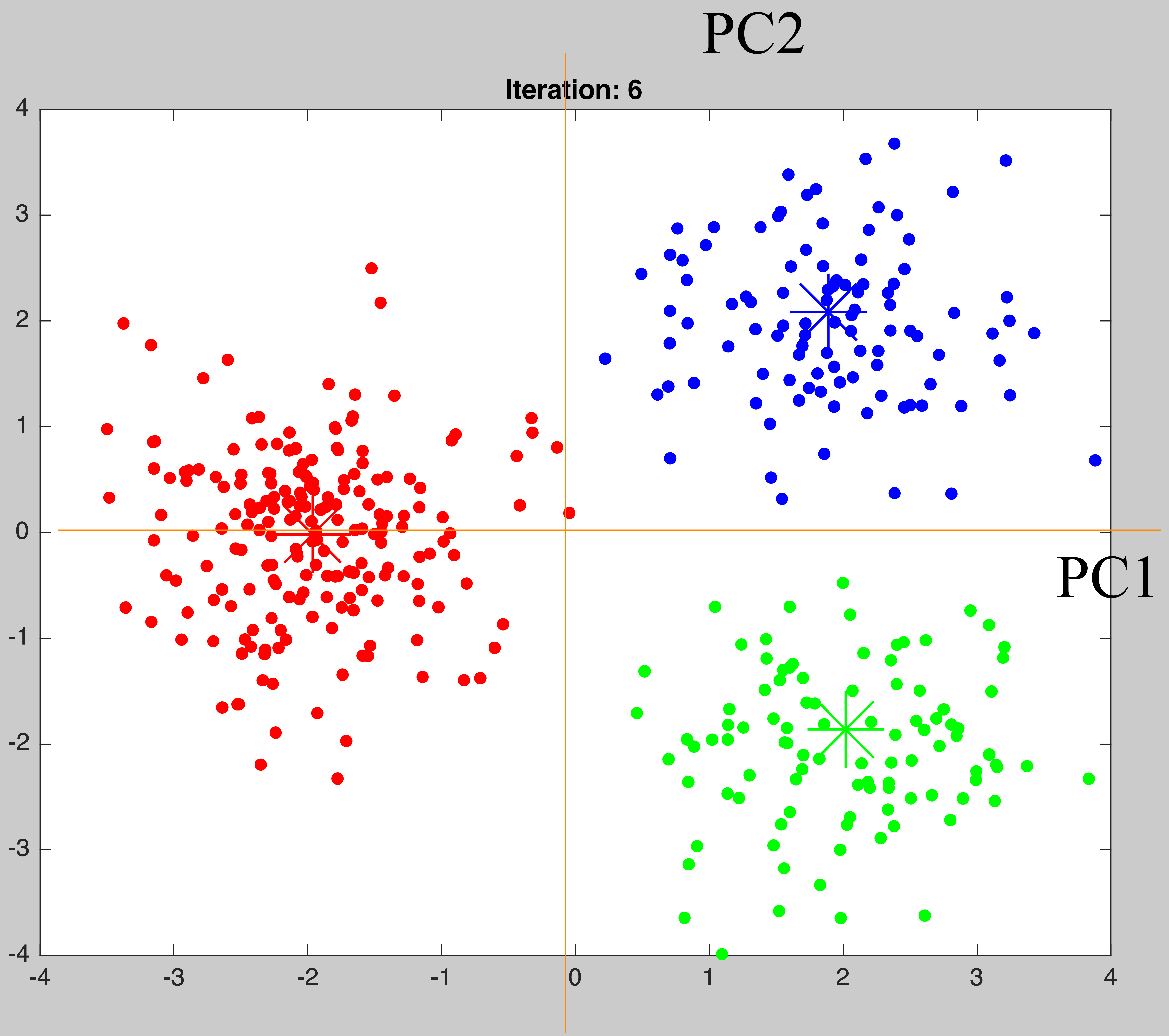
Word4

Word5

Word6

Word7

Word8



Labeled VS non labeled Classification

Word1

Word2

Word3

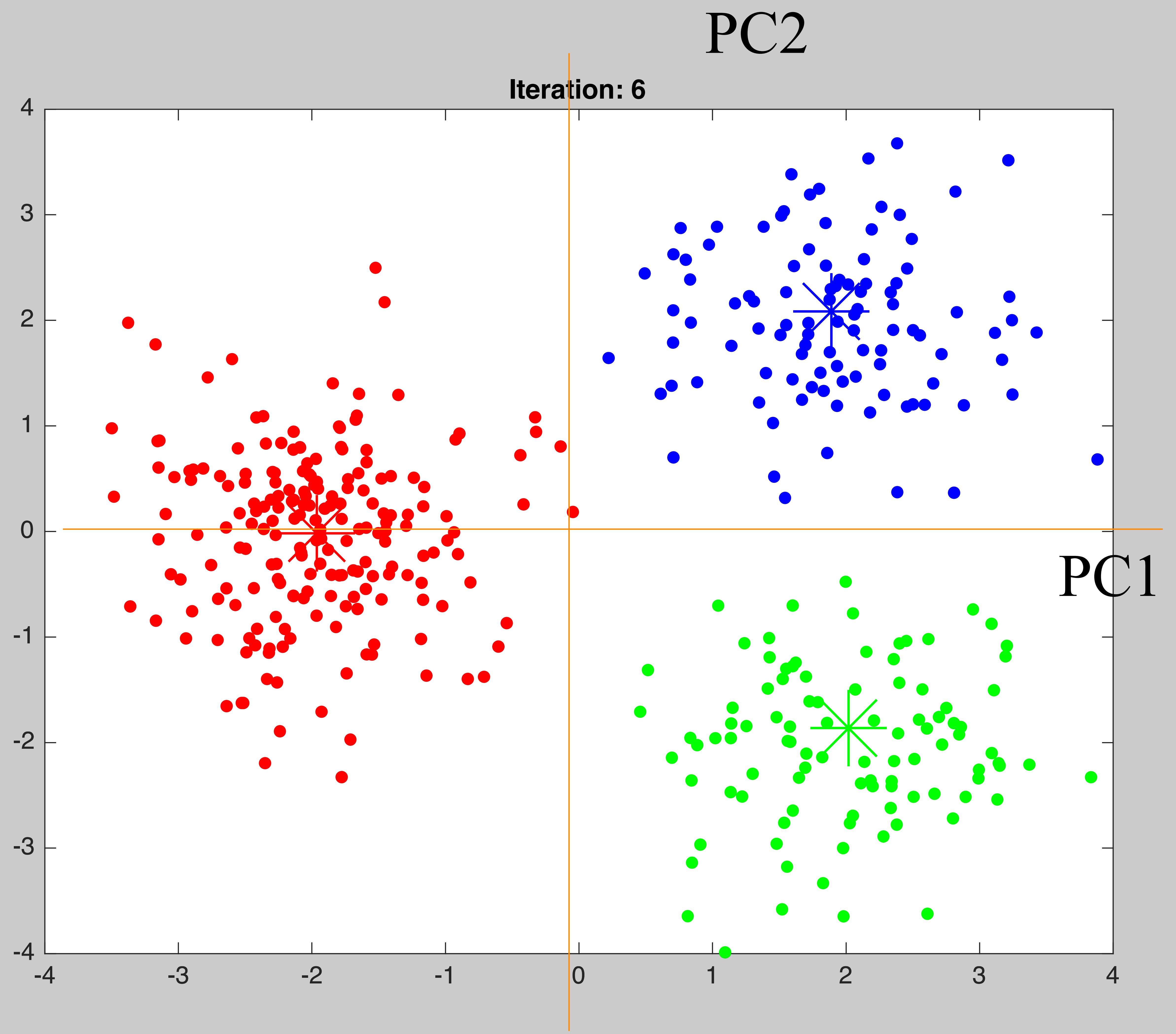
Word4

Word5

Word6

Word7

Word8



Labeled VS non labeled Classification

"Banana"

"Banana"

"Banana"

"Mango"

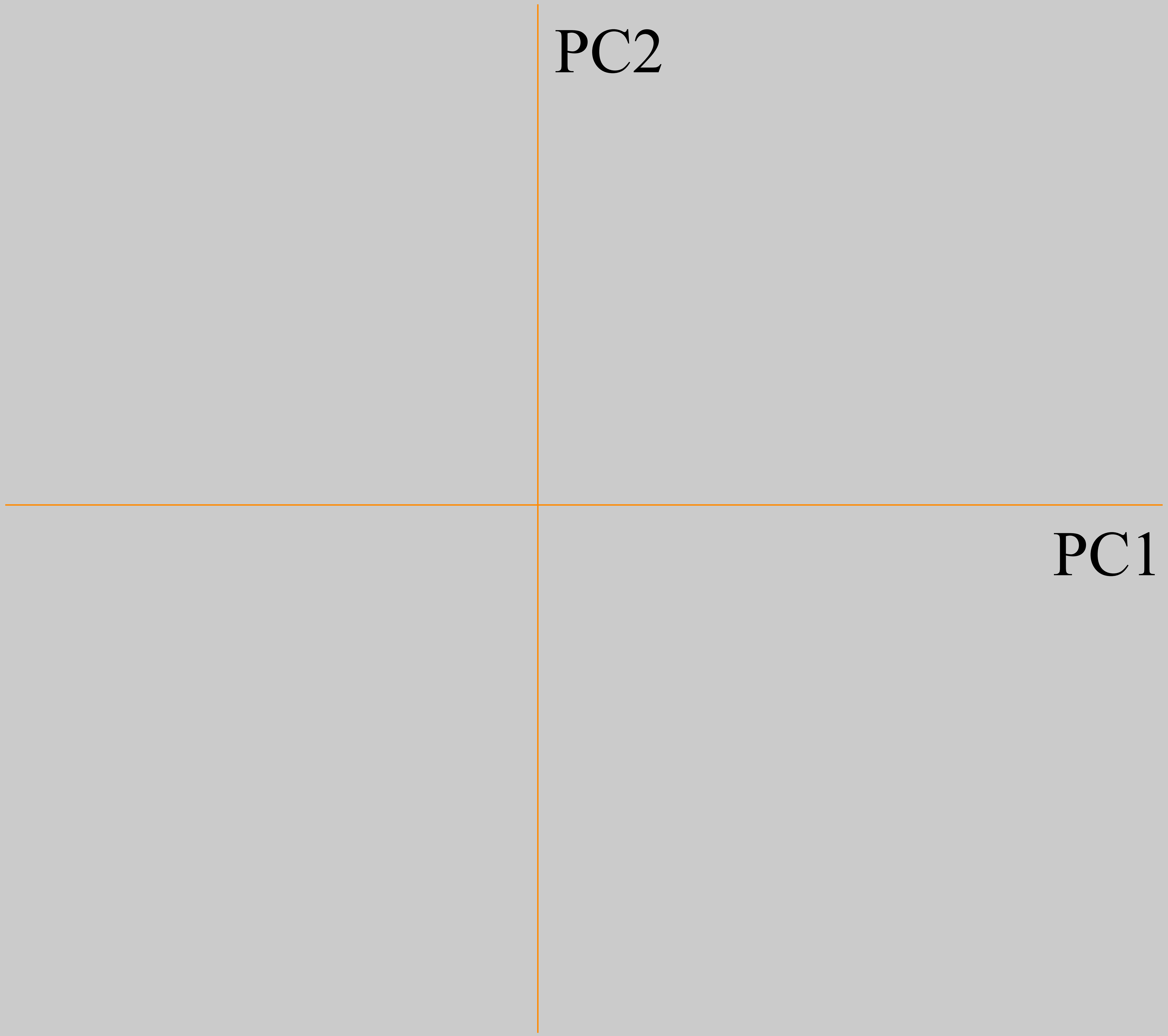
"Mango"

"Mango"

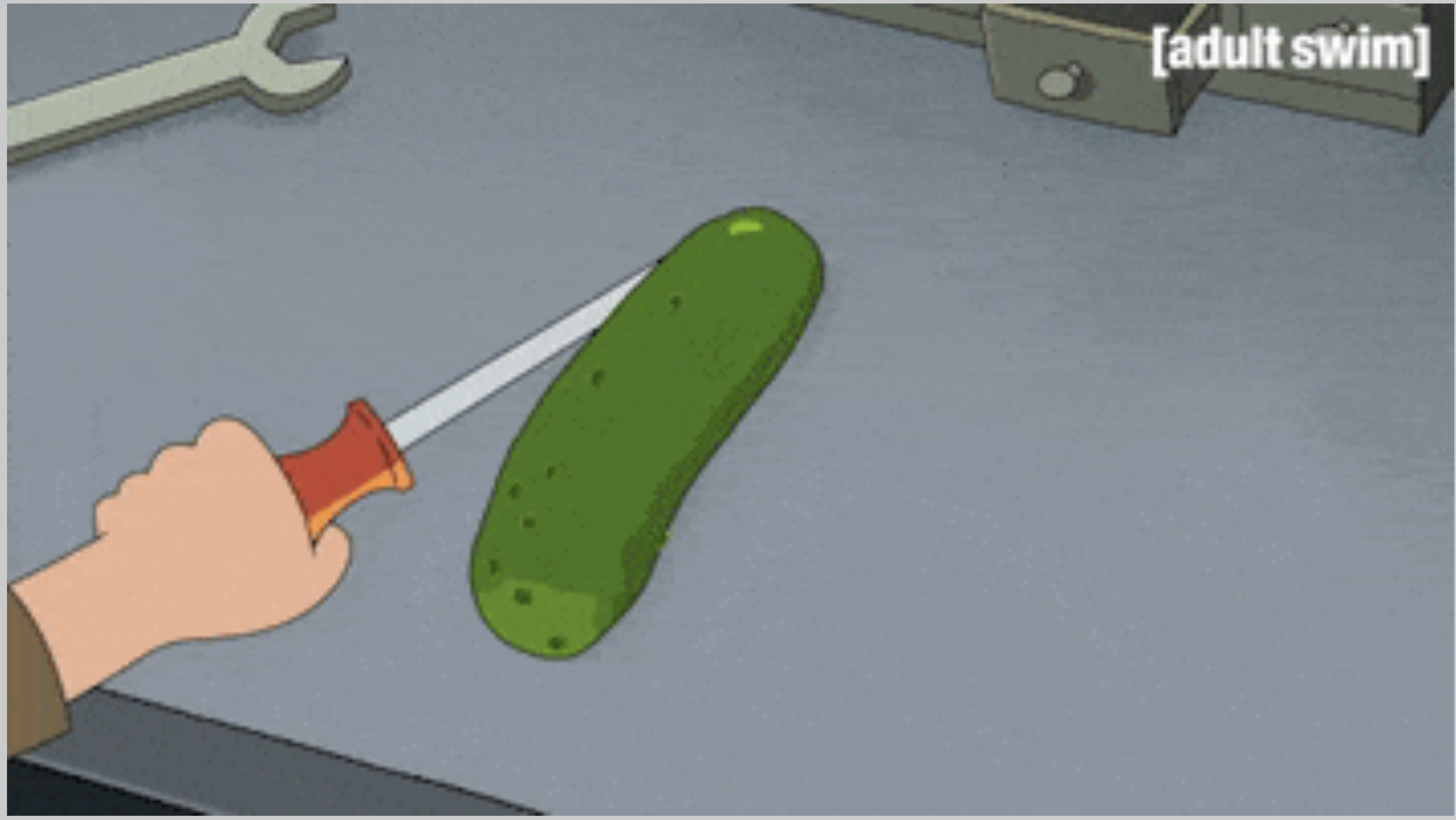
"Chop"

"Chop"

"Chop"



Matrix Inverse, and Pseudo-Inverse



- Square, full rank (N) A:

$$Ax = y$$

$$x = A^{-1}y$$

$$A^{-1}A = AA^{-1} = I_{N \times N}$$

- Tall, full rank (N) A (Least squares):

$$Ax = y$$

$$x = (A^TA)^{-1}A^Ty$$

$$(A^TA)^{-1}A^T\ A = I_{N \times N}$$

Also, left inverse

Pseudo Inverse and the SVD

$$Ax = y$$

$$x = (A^TA)^{-1}A^Ty$$

- SVD:

$$A = U_1 \; S \; V_1^T$$

$$(A^TA) = V_1SU_1 \;\; U_1SV_1^T = V_1S^2V_1^T$$

$$(A^TA)^{-1} = \; V_1S^{-2}V_1^T$$

$$(A^TA)^{-1}A^T = V_1S^{-2}V_1^T \;\; V_1^TSU_1^T = V_1S^{-2}SU_1^T =$$

$$= V_1S^{-1}U_1^T = A^\dagger$$

Moore-Penrose Pseudo-inverse

Uder-determined Linear Systems

- Fat, full rank (N) A:

$$Ax = y$$ Infinite solutions!

- Claim:

$$AA^{\dagger} = I_{N \times N}$$ Also, right inverse

- SVD:

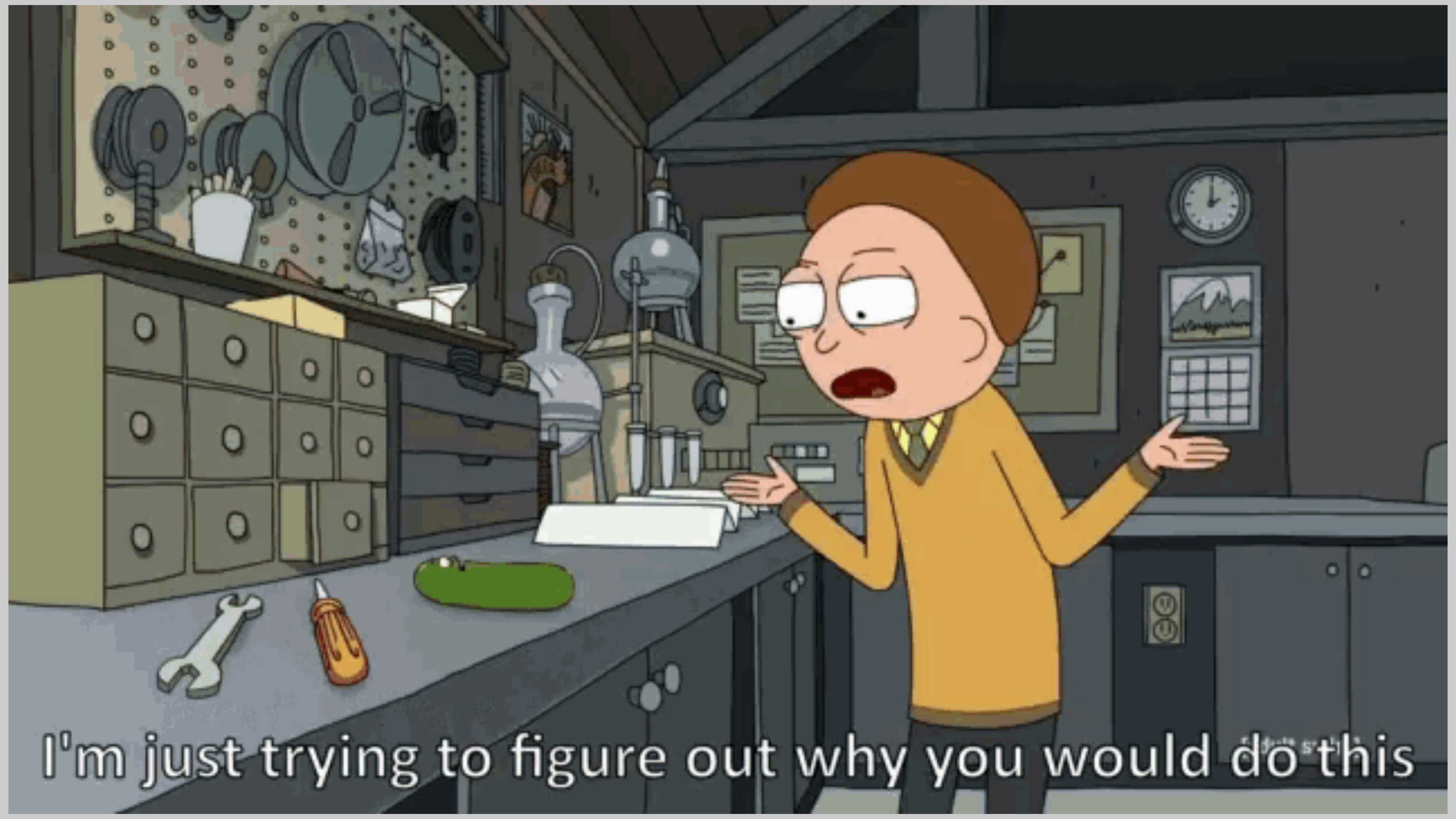
$$A = U_1 \; S \; V_1^T$$

$$AA^{\dagger} = U_1 S V_1^T \; V_1 S^{-1} U_1^T = U_1 S S^{-1} U_1^T = U_1 U_1^T = I_{N \times N}$$

Uder-determined Linear Systems

- Fat, full rank (N) A:

$$Ax = y$$ Infinite solutions!



- Fact:

$$AA^{\dagger} = I_{N\times N}$$

- A Solution:

$$\hat{x} = A^{\dagger}y$$

$$A\hat{x} = AA^{\dagger}y = Iy = y$$

- A minimum-norm solution!

$$||\hat{x}|| < ||\vec{x}|| \qquad \forall A\vec{x} = y$$

Minimum - Norm

- A minimum-norm solution! $\hat{x} = A^{\dagger} y$

$$||\hat{x}|| < ||\vec{x}|| \qquad \forall A\vec{x} = y$$

- Proof outline:

$$\vec{x} = \hat{x} + \tilde{x}$$

  - Show that $A\tilde{x} = 0$
  - Show that $\hat{x}^T \tilde{x} = 0$
  - Show that $||\vec{x}||^2 = ||\hat{x} + \tilde{x}||^2 > ||\hat{x}||^2$

Back to Open Loop Control

$$\vec{x}(t) - A^t\vec{x}(0) = \underbrace{\begin{bmatrix} A^{t-1}B & A^{t-2}B & \cdots & AB & B \end{bmatrix}}_{R_t} \begin{bmatrix} u(0) \\ u(1) \\ \vdots \\ u(t-1) \end{bmatrix}$$

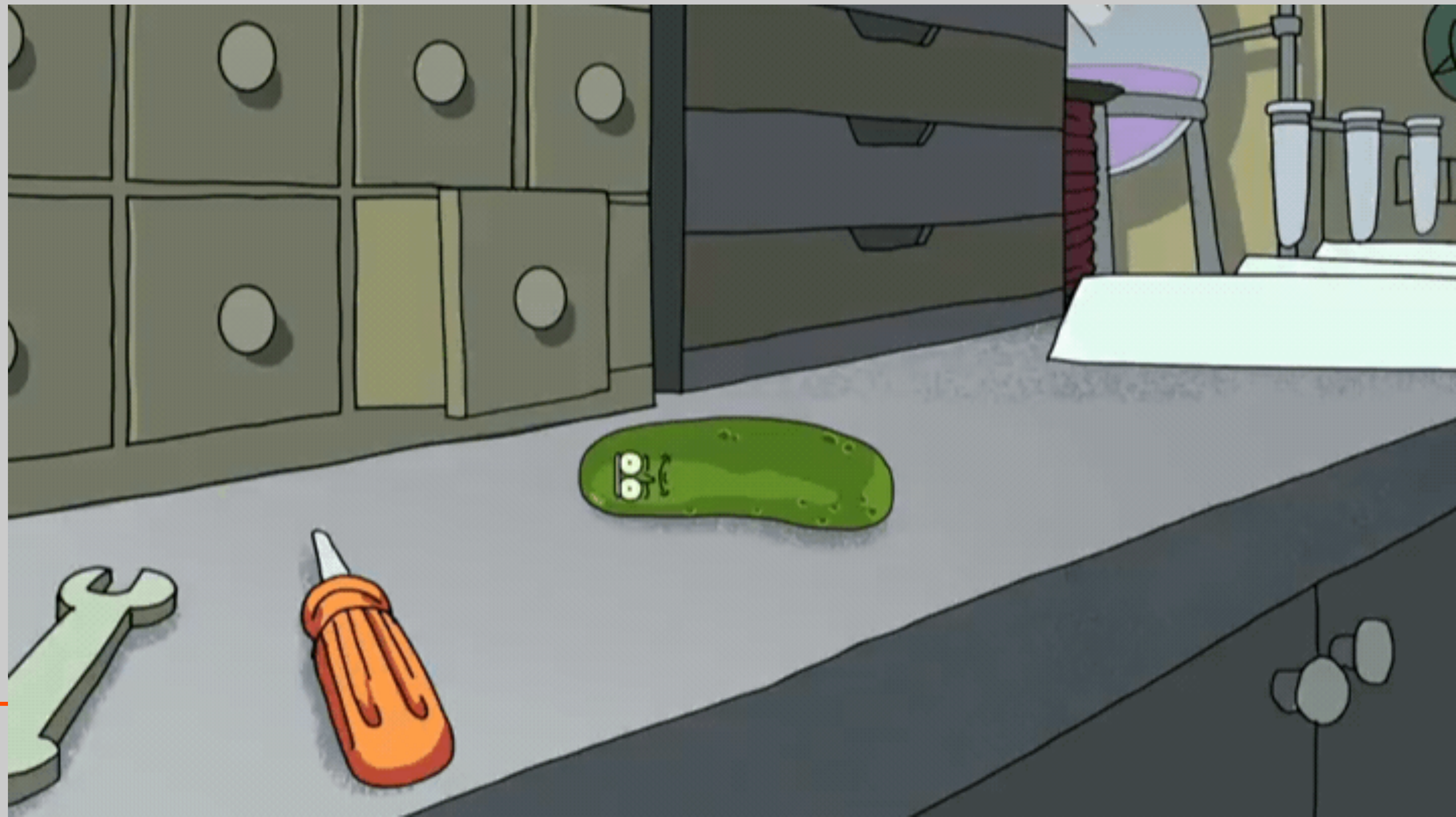
System is controllable. What if t >> N?
Whats a good sequence of u(t)?

Summary:

- Moore penrose Pseudo Inverse

$$A^{\dagger} = V_1 S^{-1} U_1^T$$

- For tall matrices is the least squares solution!
- For fat matrices is the least-norm solution!
- Computed via the SVD!

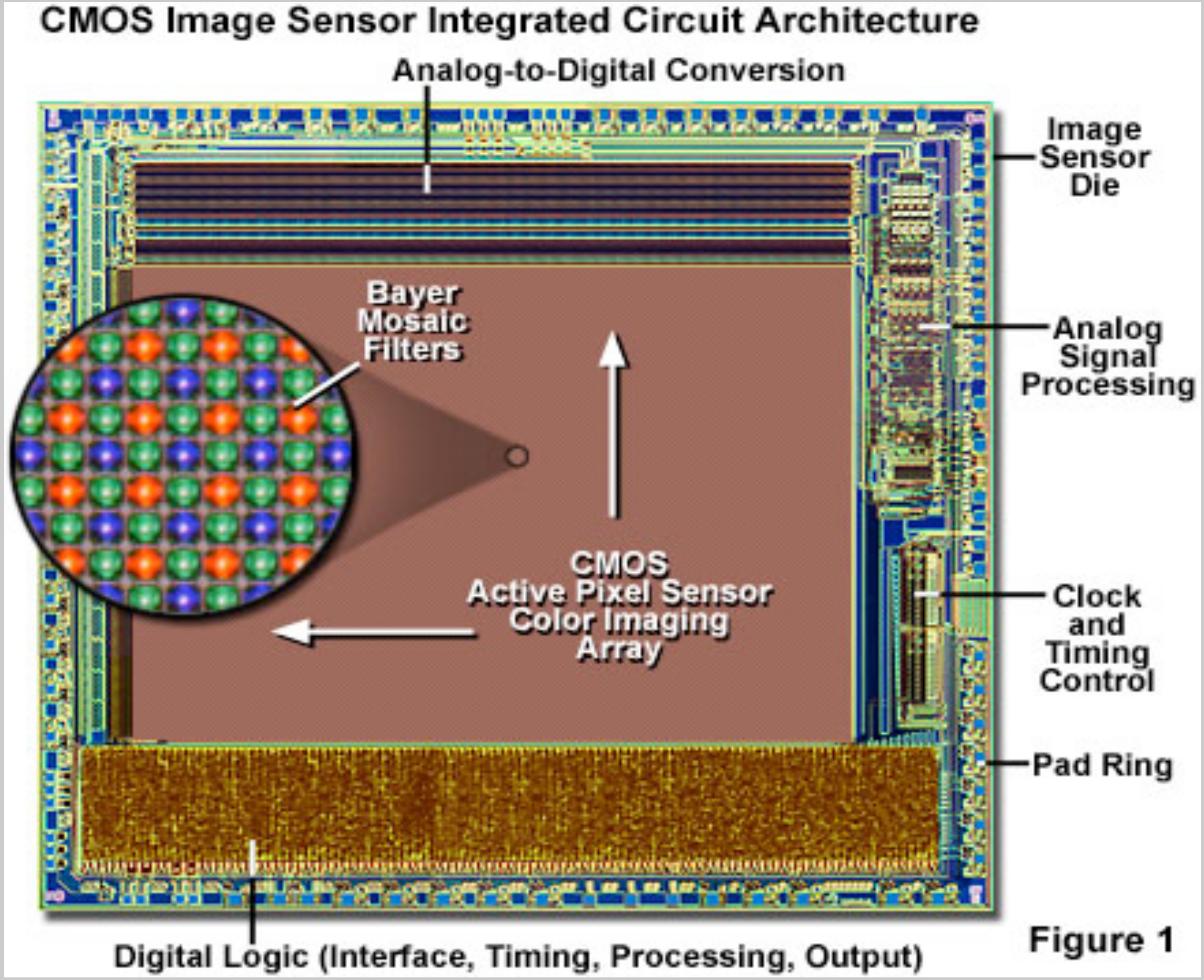


Sampling and Interpolation

- (Digital) Signal Processing - Only going to touch the surface
  – EE120, EE123, EE145A, EE121, EE225A, EE225B, CS 194-026, CS280

The Real World	Acquisition System (Physics/Physical)	Analog to Digital Conversion	Digital "magic": Processing, manupulation, interpretation Storage	Reconstruction & Digital to Analog Conversion
RF Audio Light Energy …	Conversion of X to electric Analog Filtering Amplification Modulation Microphone CCD/CMOS Accelerometers Antennas Radio-receivers …	Discrete values Discrete representation	Digital filtering Decoding Coding Compression Learning Rendering …	Audio Light RF Magnetic Display Communications …

Example Digital Imaging Camera

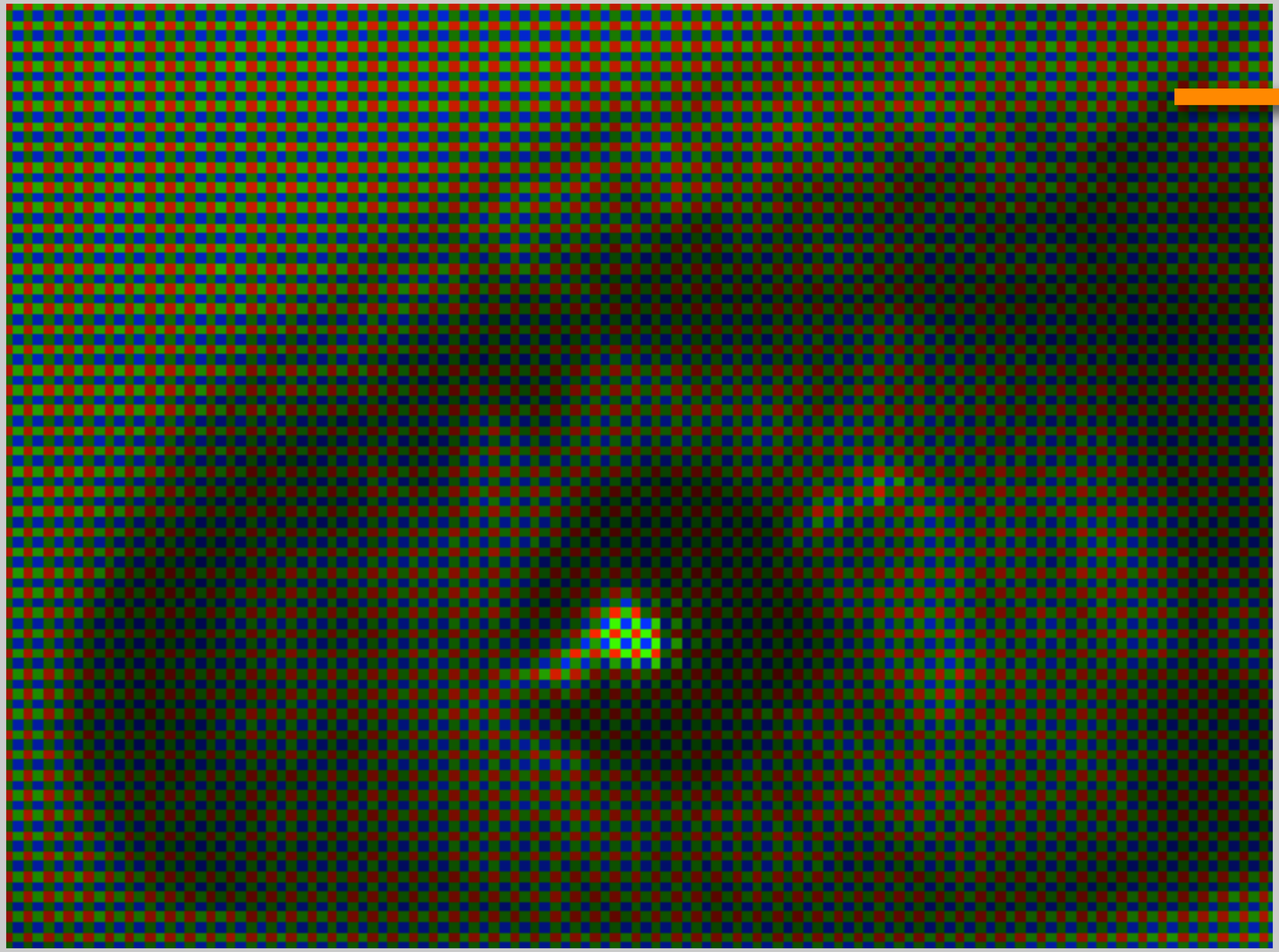


Focus/exposure Control → preprocessing → white-balancing → demosaic → Color transform → Post-processing → Compression

http://micro.magnet.fsu.edu/primer/digitalimaging/cmosimagesensors.html

Example: Digital Camera

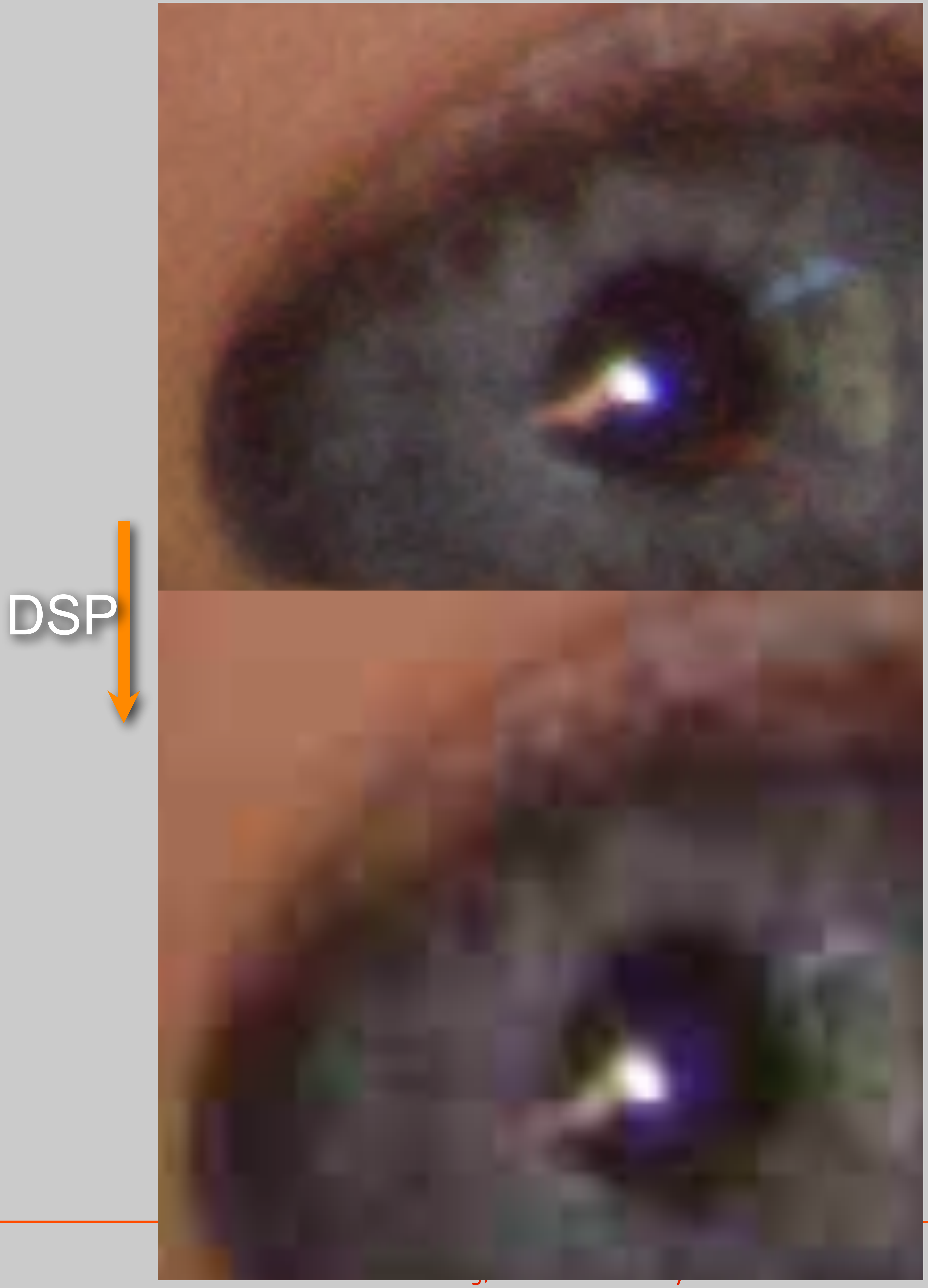
DSP



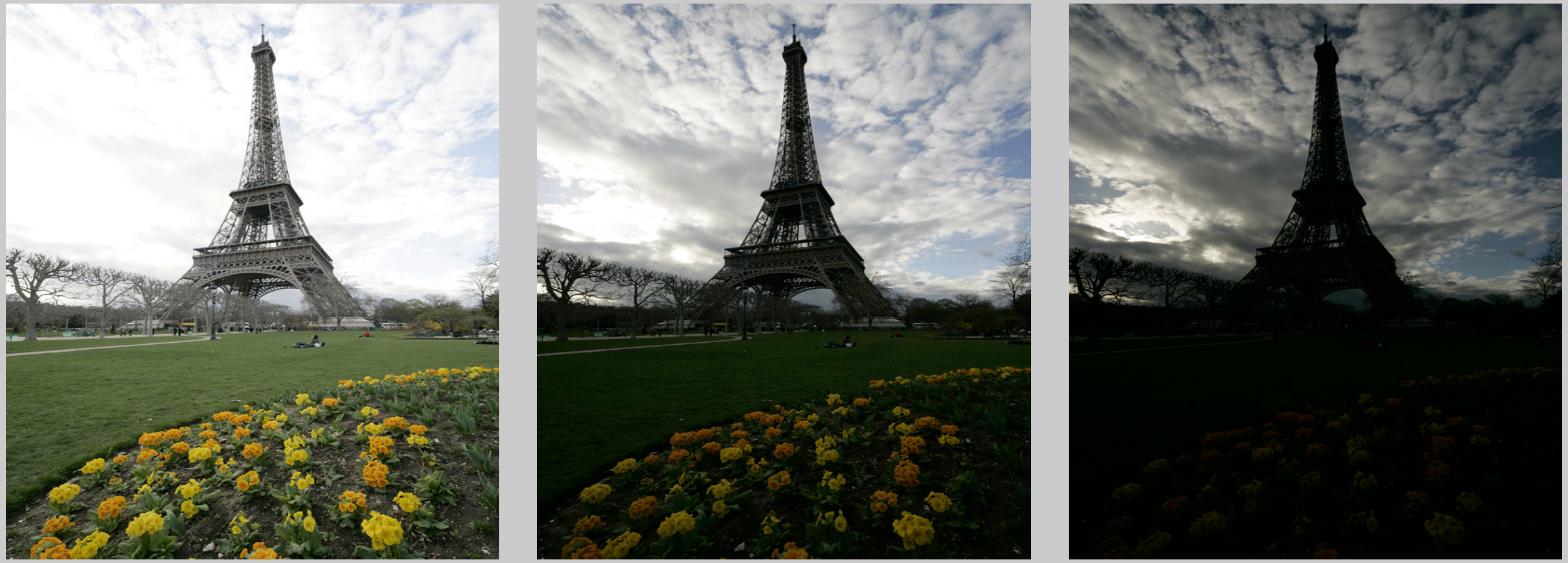
Example: Digital Camera

- Compression of 40x without perceptual loss of quality.

- Example of slight overcompression: difference enables x60 compression!



Computational Photography



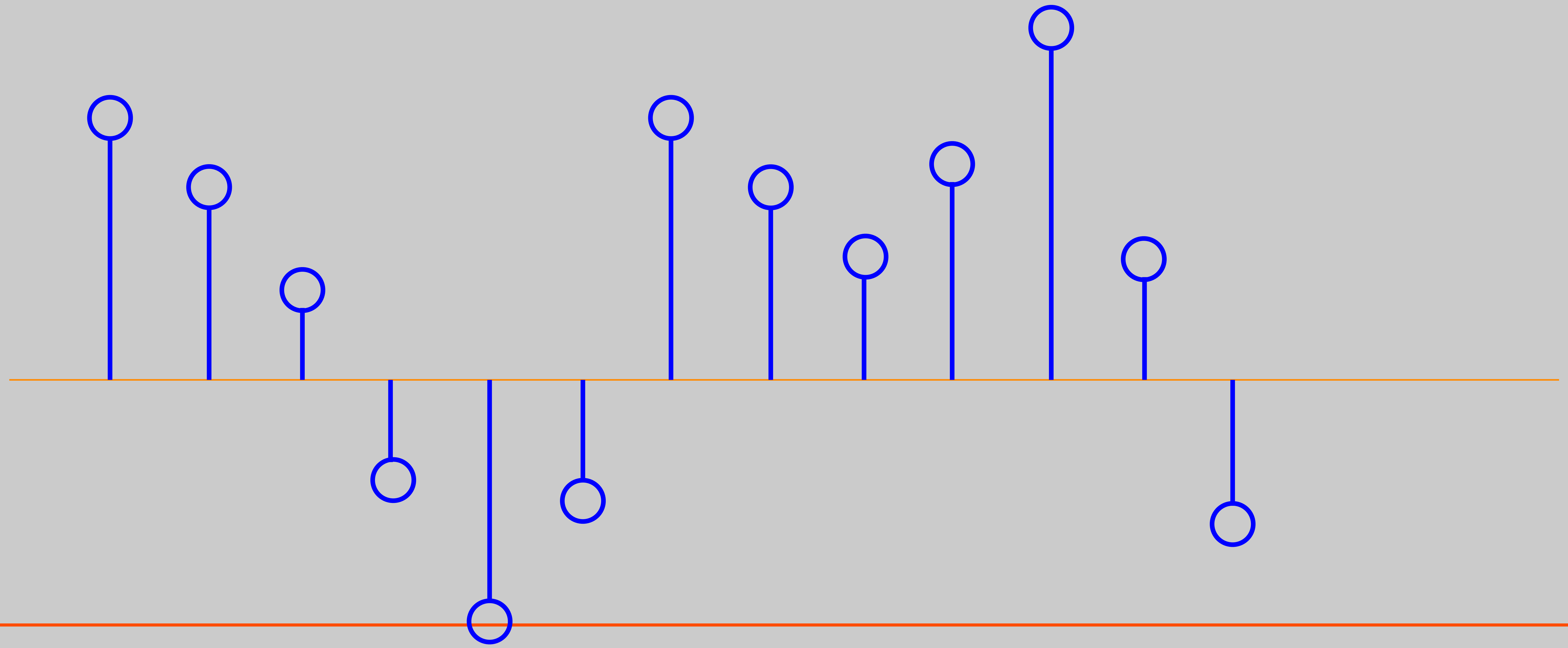
Implemented in all smart phones (HDR)

*www.hdrsoft.com

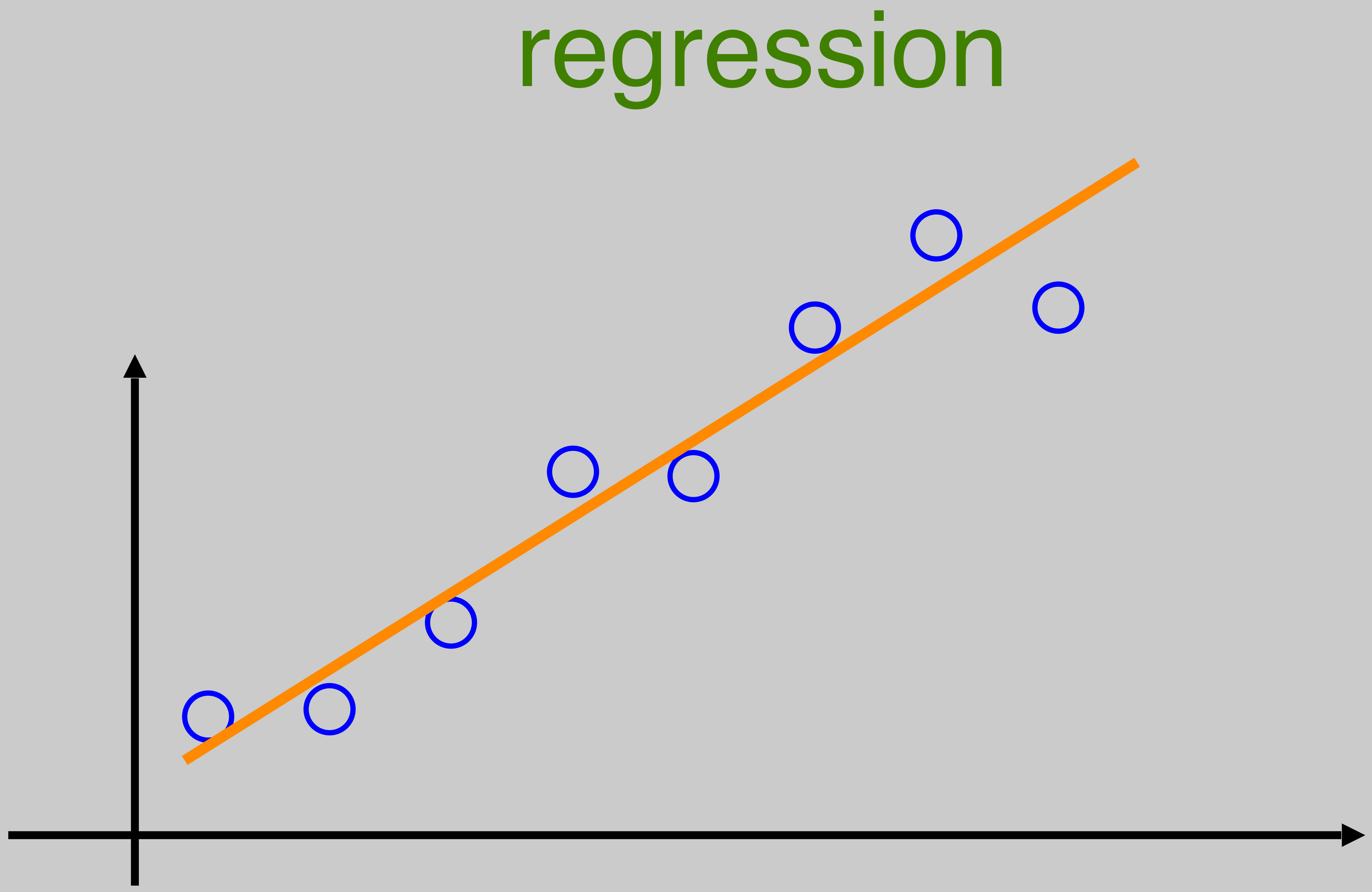
Interpolation

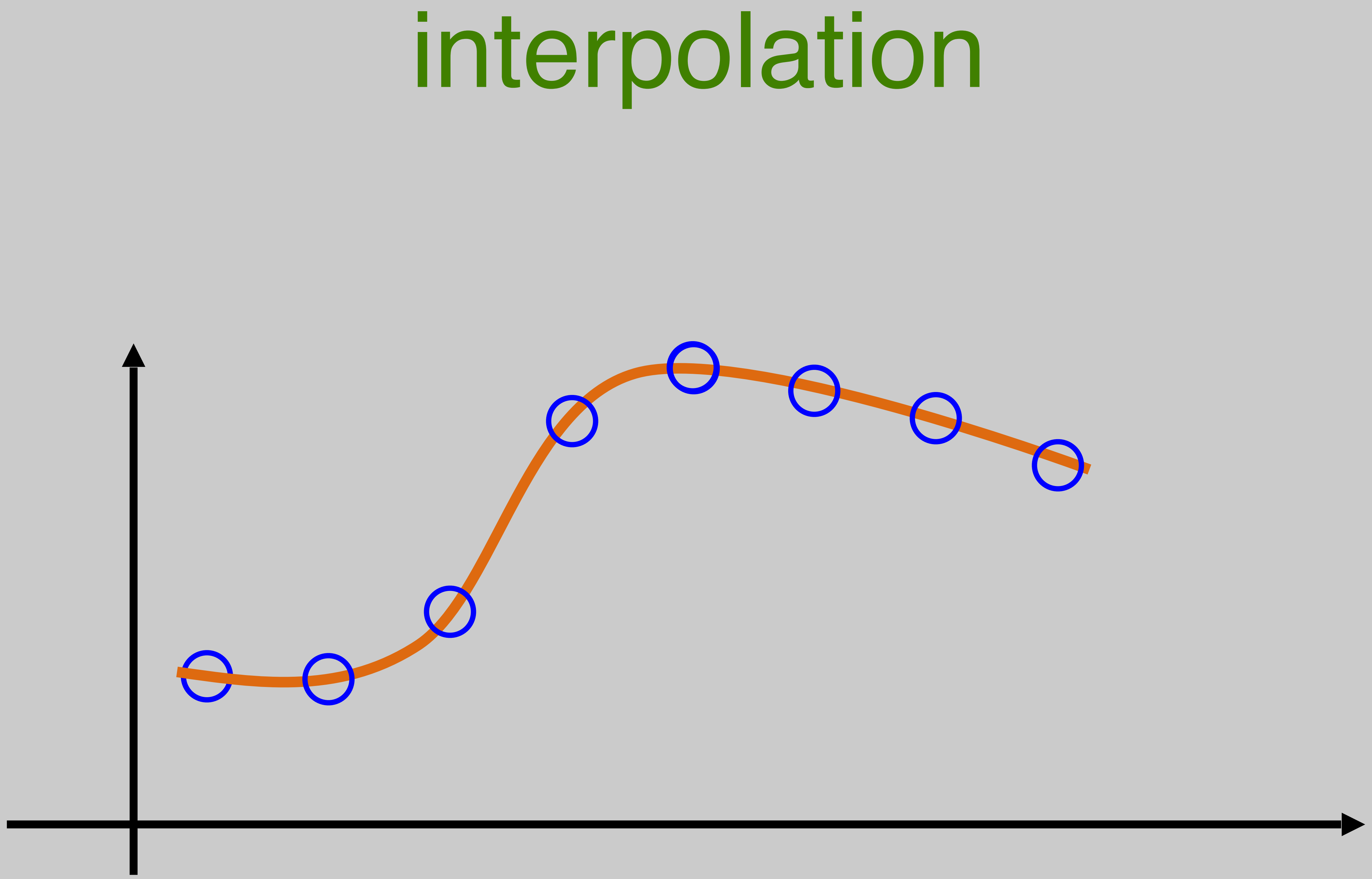
- Given data points (x_i, y_i) i=1,2,… n find a continuous function that exactly matches the points.

$$y = f(x) \qquad \Rightarrow f(x_i) = y_i, \qquad i = 1,\ 2,\ 3\ ,\cdots,\ n$$



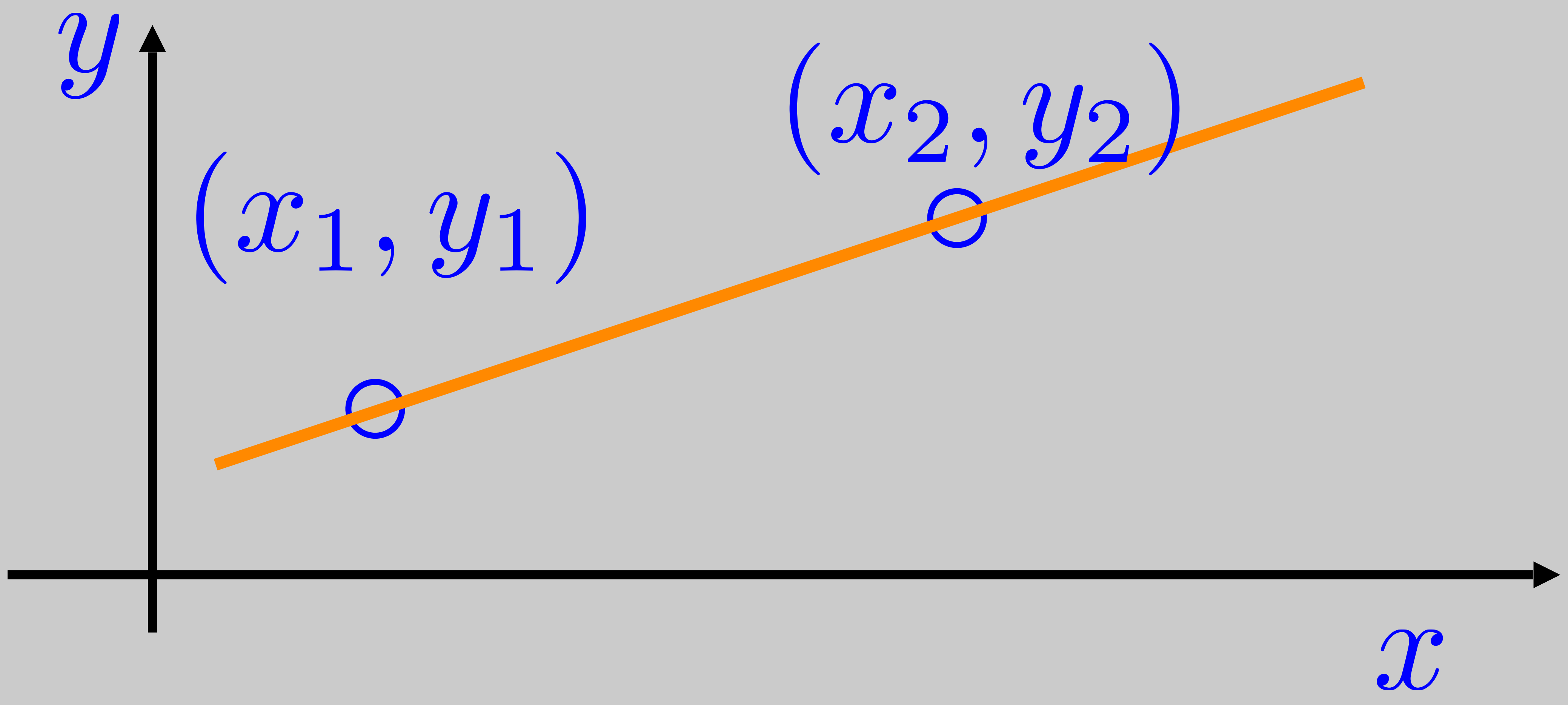
Regression Vs Interpolation





Polynomial Interpolation

- Assumption:
  - Data are samples of a polynomial function (smooth)
  - Lowest order polynomial that exactly fits points

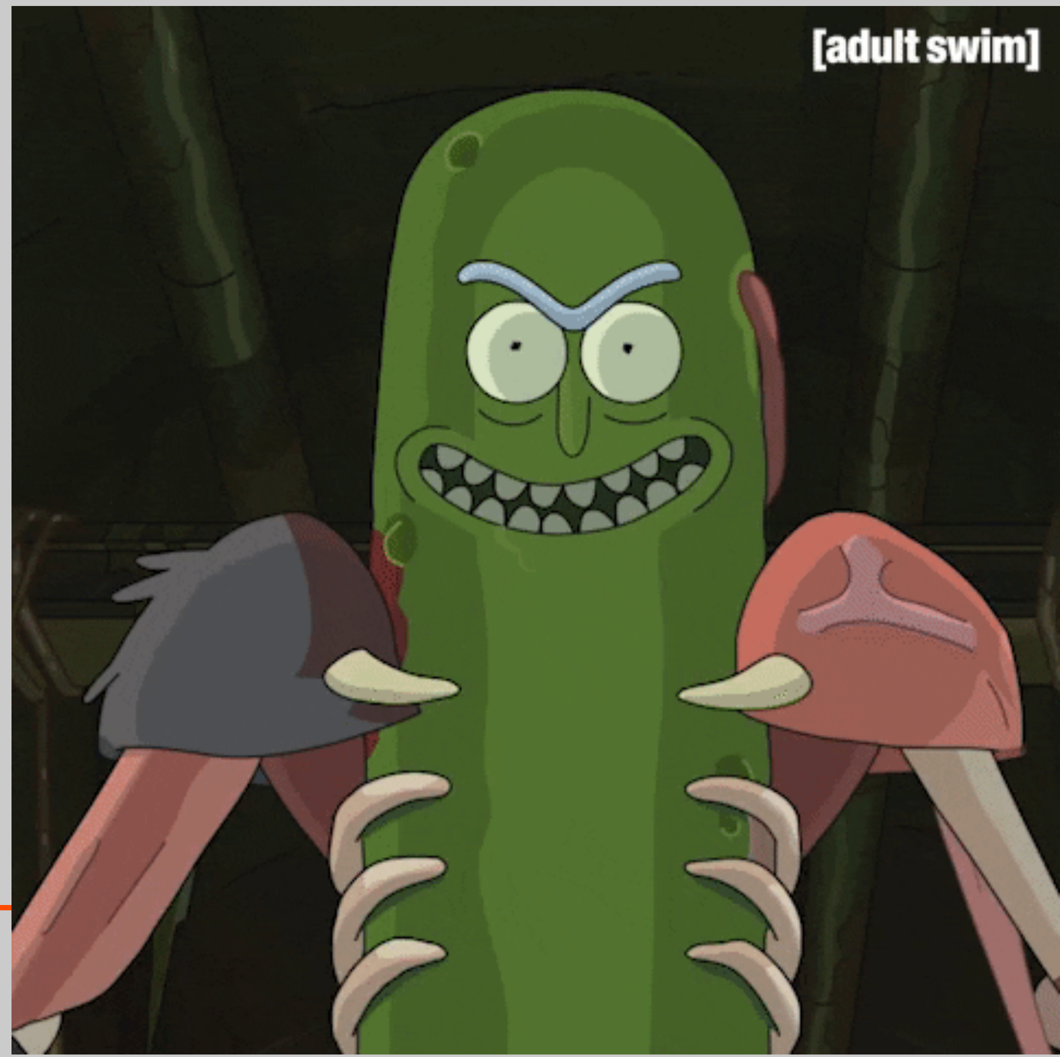


$$y = a_0 + a_1 x$$

$$a_0 + a_1 x_1 = y_1$$

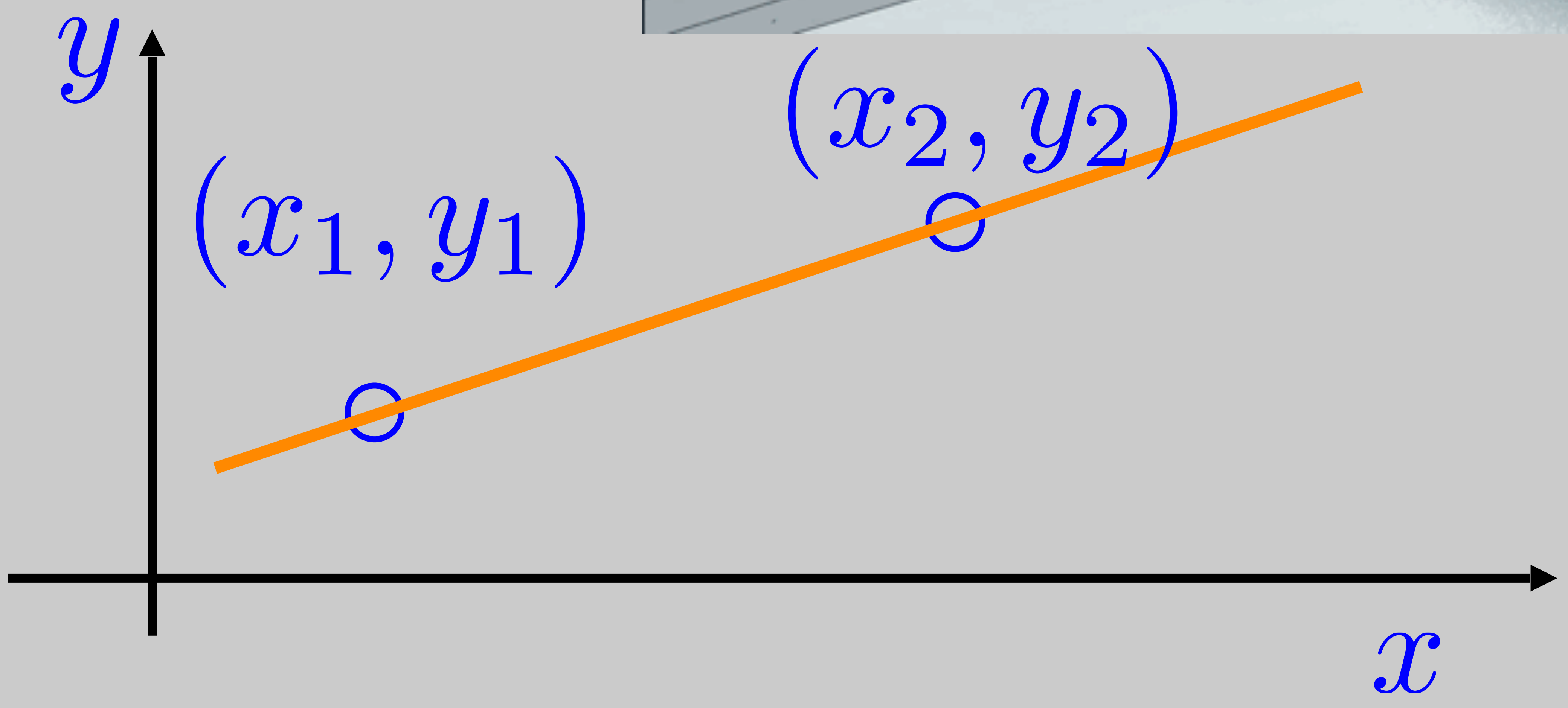
$$a_0 + a_1 x_2 = y_2$$

$$\rightarrow \begin{bmatrix} \quad & \quad \\ \quad & \quad \end{bmatrix} \begin{bmatrix} \quad \\ \quad \end{bmatrix} = \begin{bmatrix} \quad \\ \quad \end{bmatrix}$$



Polynomial Interpolation

- Assumption:
  - Data are samples of a polynomial function (smooth)
  - Lowest order polynomial that exactly fits points



$$y = a_0 + a_1 x$$

$$\begin{aligned} a_0 + a_1 x_1 &= y_1 \\ a_0 + a_1 x_2 &= y_2 \end{aligned} \rightarrow \begin{bmatrix} 1 & x_1 \\ 1 & x_2 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}$$

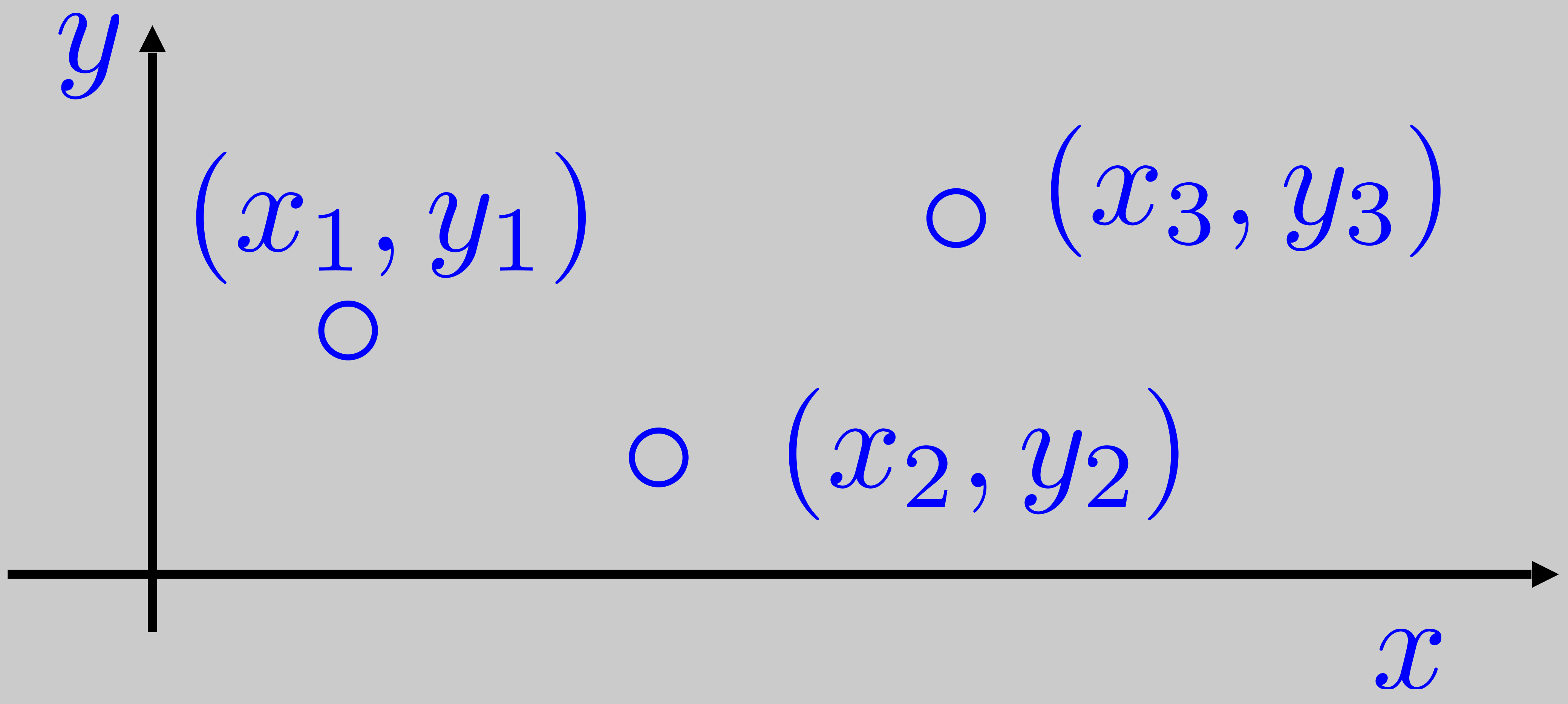
Invertible if $x_1 \neq x_2$

Polynomial Interpolation

$$(x_1, y_1)\ (x_2, y_2)\ (x_3, y_3)$$

$$y = a_0 + a_1 x + a_2 x^2 \rightarrow \begin{bmatrix} 1 & x_1 & x_1^2 \\ 1 & x_2 & x_2^2 \\ 1 & x_3 & x_3^2 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix}$$

$$x_1 \neq x_j$$



Polynomial Interpolation

- Given n distinct points, then there's exist a unique (n-1) order polynomial passing through them

$$y = a_0 + a_1 x + a_2 x^2 + \cdots a_{n-1} x^{n-1}$$

$$\rightarrow \begin{bmatrix} & & & \\ & & & \\ & & & \\ & & & \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_{n-1} \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$$

Polynomial Interpolation

- Given n distinct points, then there's exist a unique (n-1) order polynomial passing through them

$$y = a_0 + a_1 x + a_2 x^2 + \cdots a_{n-1} x^{n-1}$$

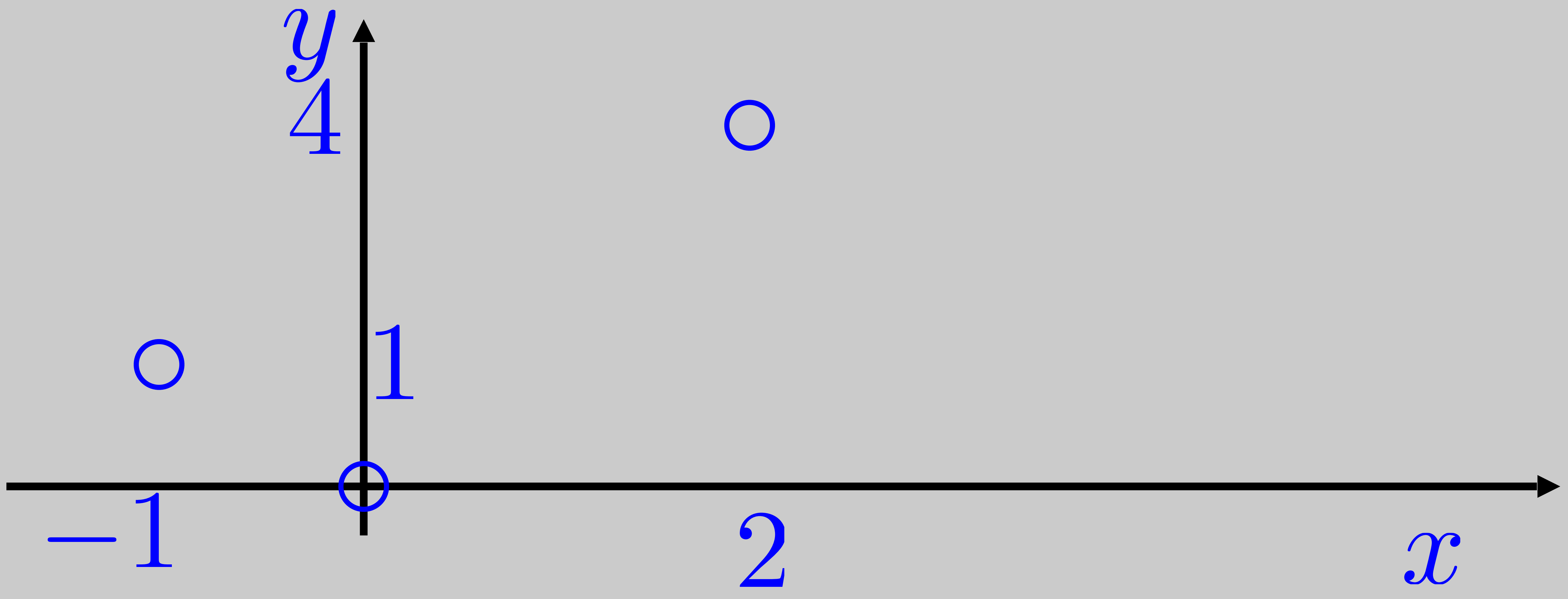
$$\rightarrow \begin{bmatrix} 1 & x_1 & x_1^2 & \cdots & x_1^{n-1} \\ 1 & x_2 & x_2^2 & \cdots & x_2^{n-1} \\ & \vdots & & & \\ 1 & x_n & x_n^2 & \cdots & x_n^{n-1} \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_{n-1} \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$$

"Vandermonde" Matrix $\det(v) = \Pi_{1 \le i < j \le n}(x_j - x_i)$

Quiz

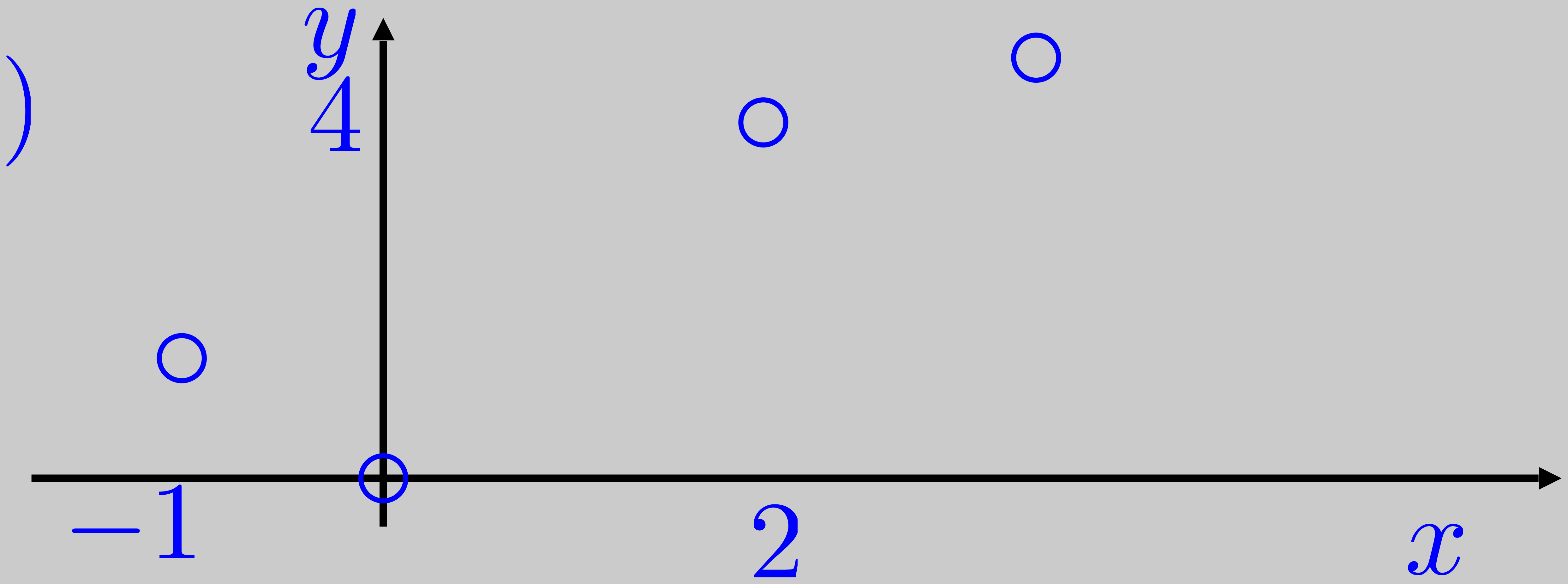
- What's the polynomial that passes through these points:

$$(-1,1),(0,0),(2,4)$$



Polynomial Regression

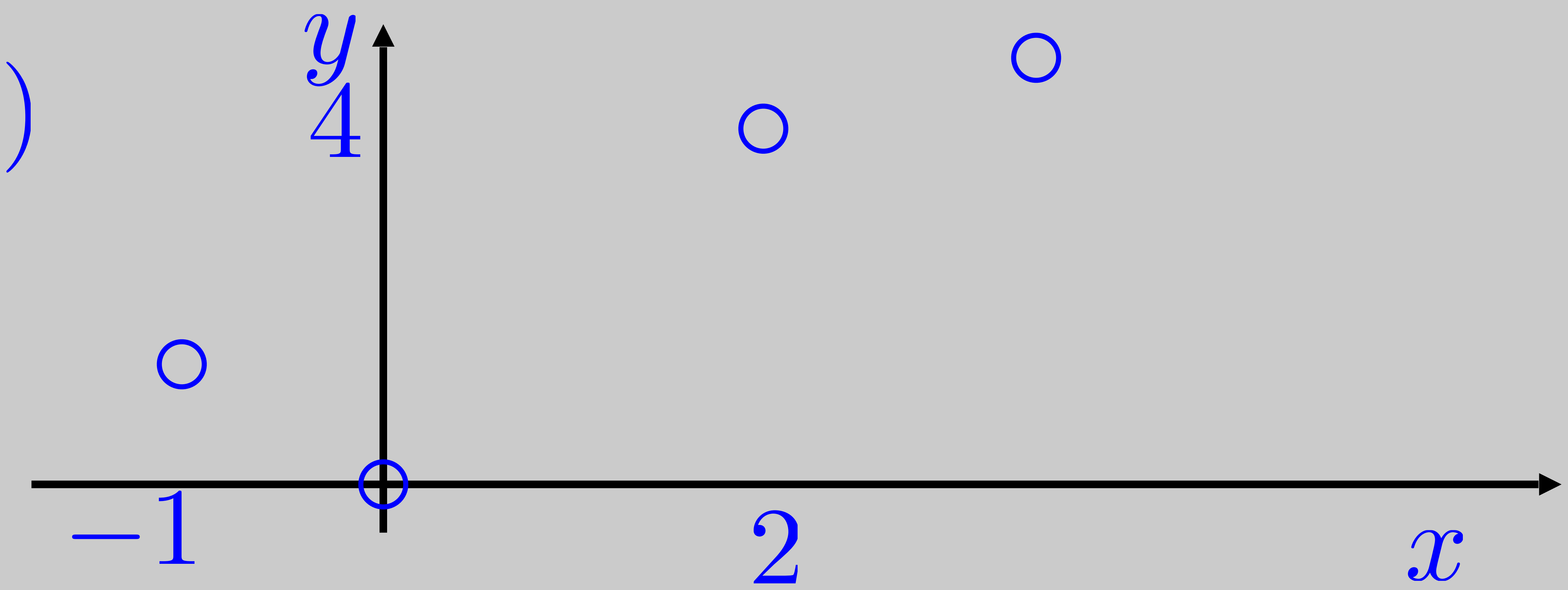
$$(-1, 1), (0, 0), (2, 4), (3, 5)$$



- What is the “best” quadratic polynomial that passes through the points?

Polynomial Regression

$$(-1,1),(0,0),(2,4),(3,5)$$



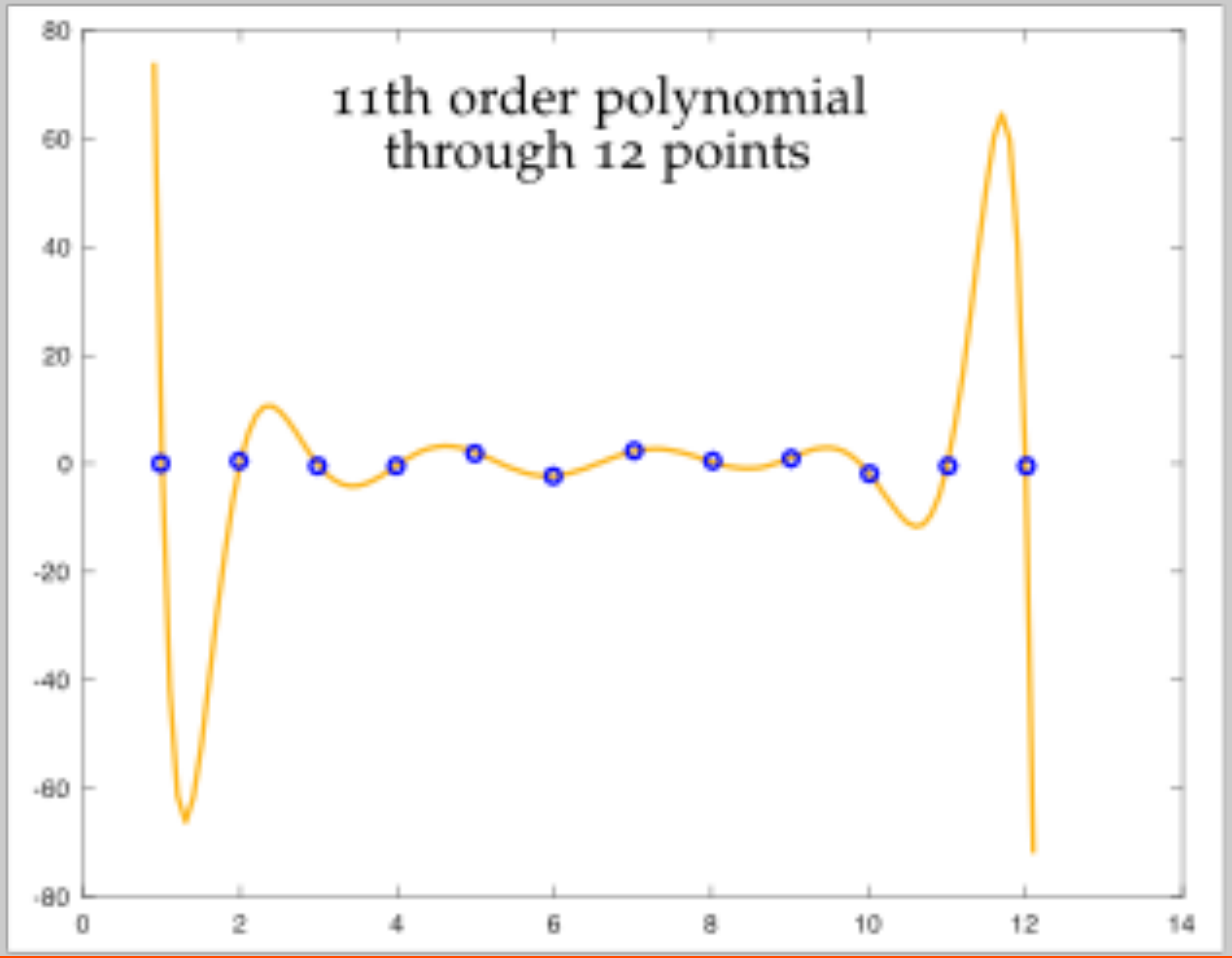
What is the “best” quadratic polynomial that passes through the points?

$$y = a_0 + a_1 x + a_2^2 + \cdots a_{n-1} x^{n-1}$$

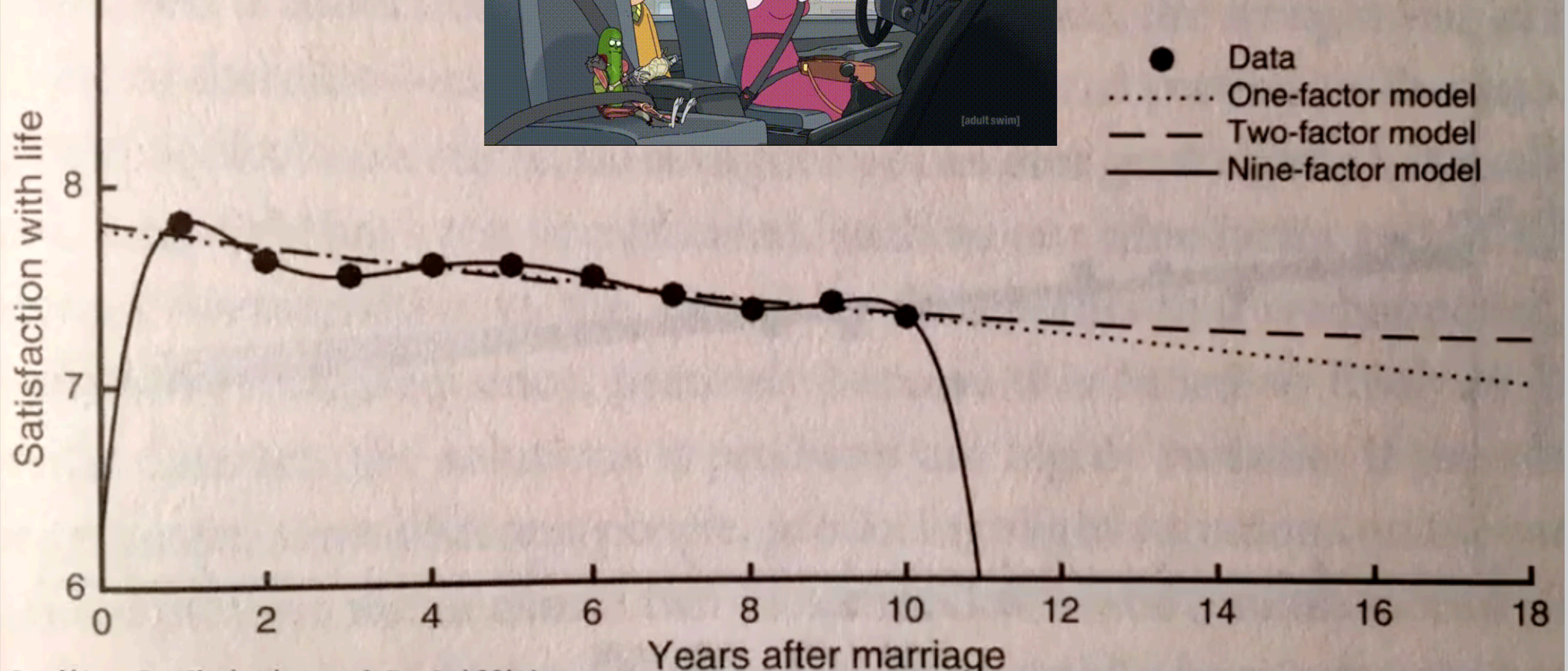
$$\begin{bmatrix} 1 & x_1 & x_1^2 \\ 1 & x_2 & x_2^2 \\ 1 & x_3 & x_3^2 \\ 1 & x_4 & x_4^2 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix}$$

Issue with Polynomial Interpolation

- Tend to be oscilatory in high order interp/regression
- Not numerically stable! x^{n-1}



Example



Credit: B. Christian and T. Griffiths,