Lecture 4

Linear Systems III

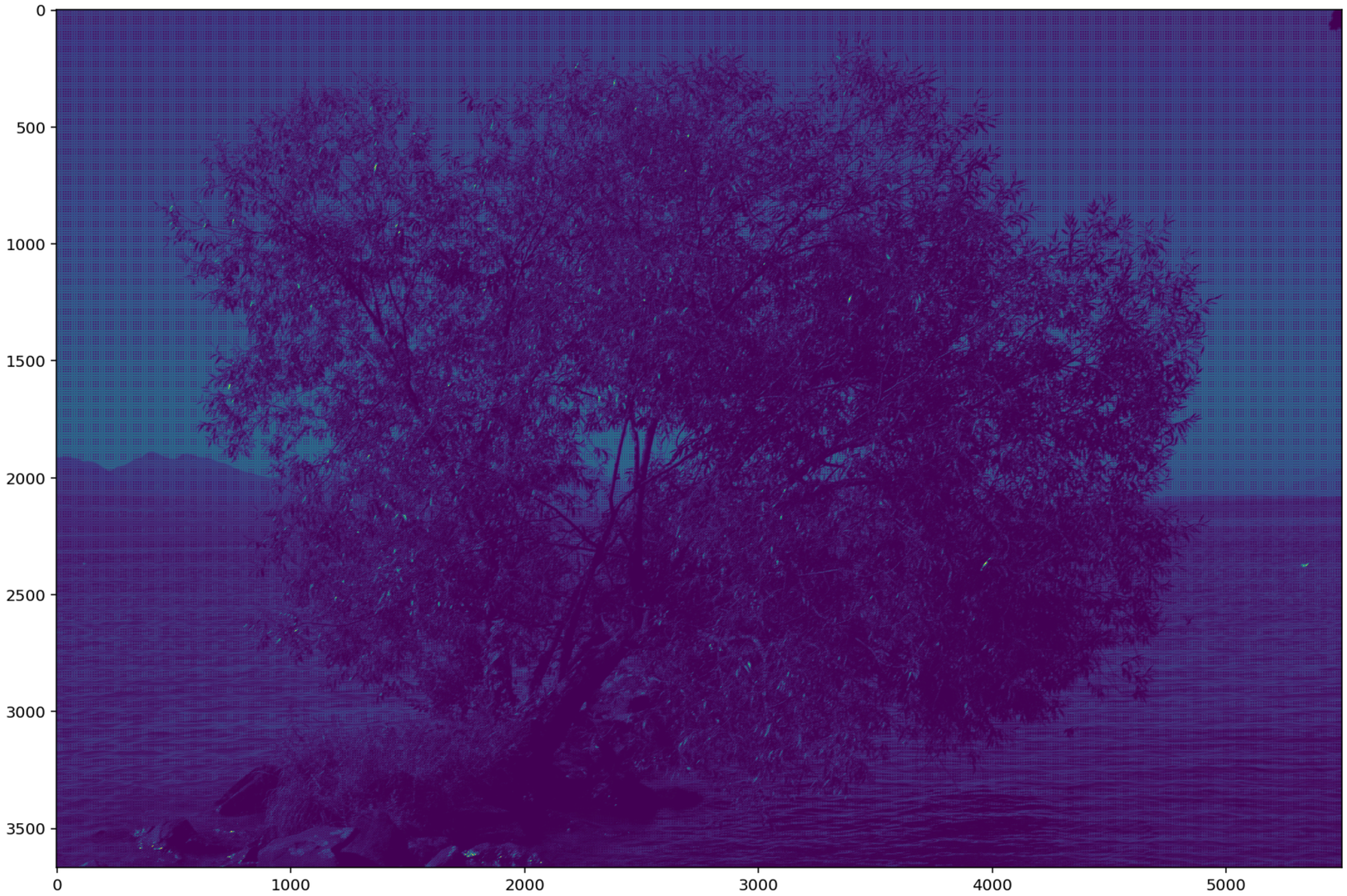
Least squares problems, orthogonality, polynomial regression

CS328 - Numerical Methods for Visual Computing and Machine Learning

Prof. Wenzel Jakob

Homework 2 out today

```
In [37]: plt.imshow(tree);
```

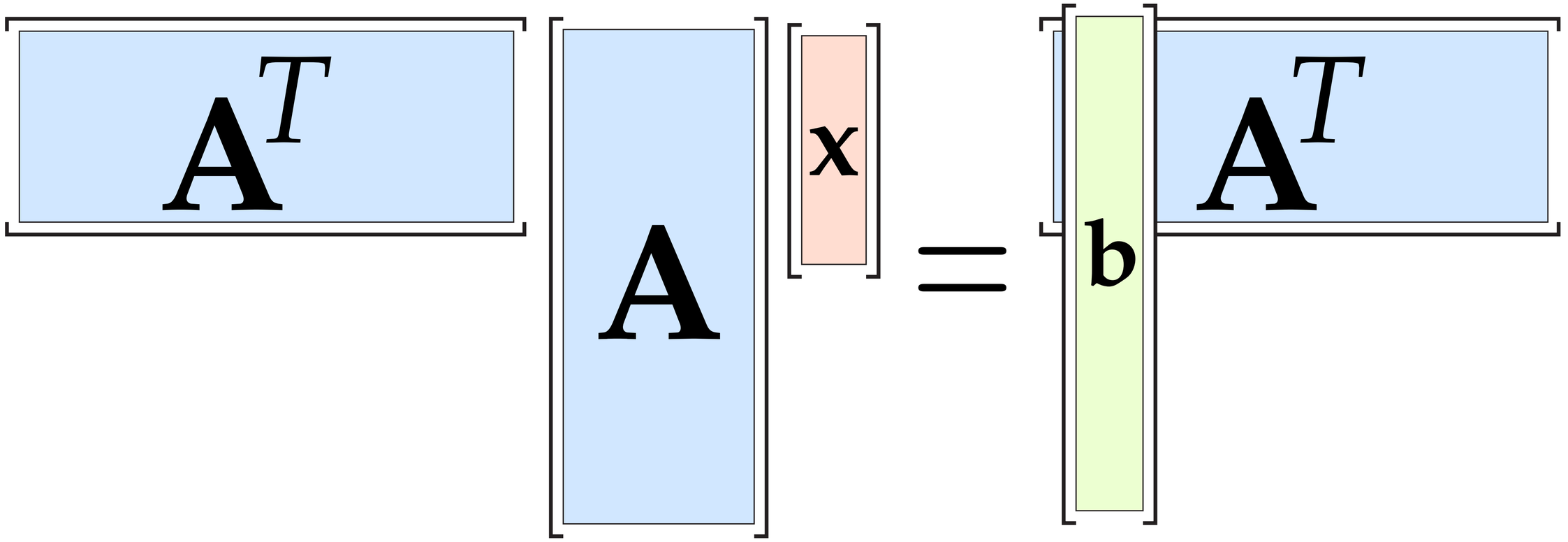


Common theme today:

Orthogonality

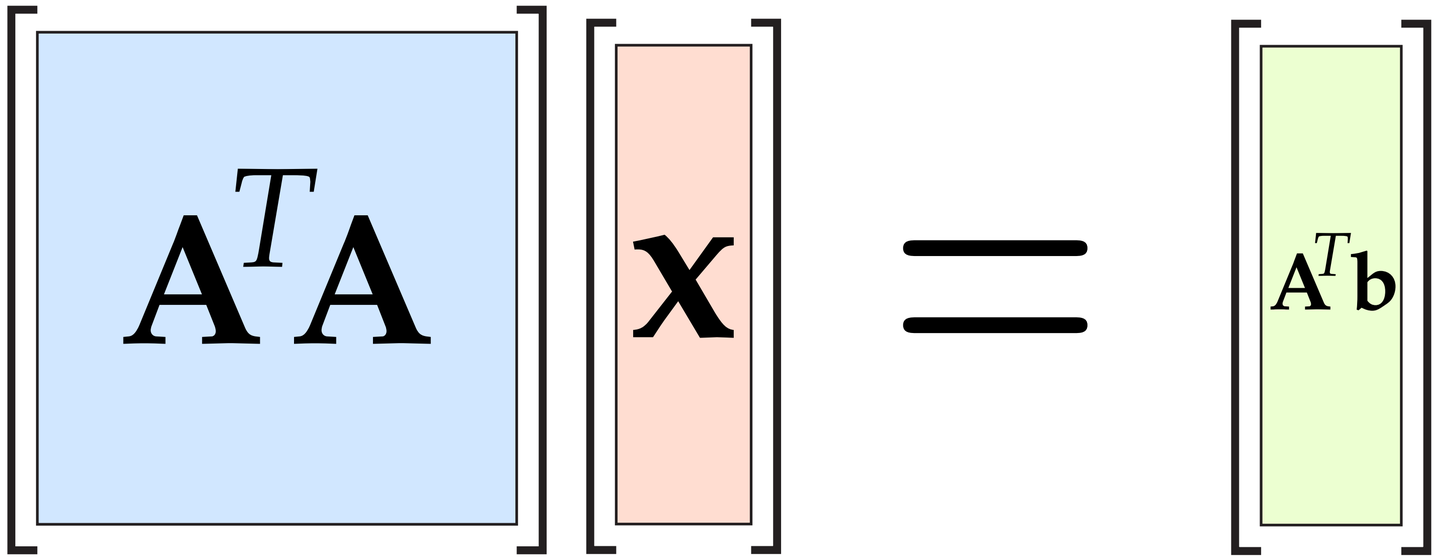
Linear least squares via the normal eqns.

The “best” solution (in the L_2 sense) is given by:



Linear least squares via the normal eqns.

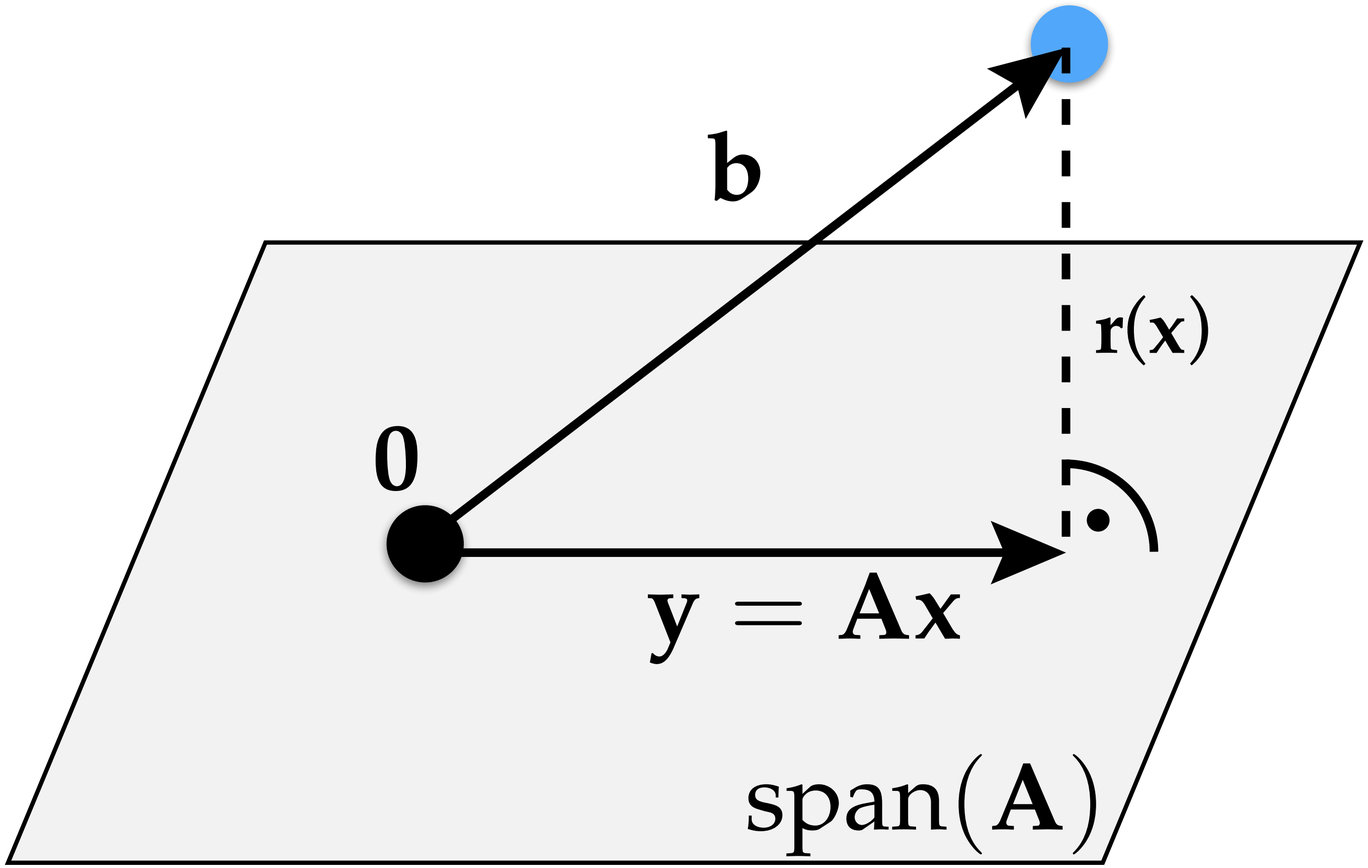
.. which is equivalent to a smaller square linear system



The normal equations yield a symmetric *quadratic* linear system,
whose size is given by the number of columns of **A**.

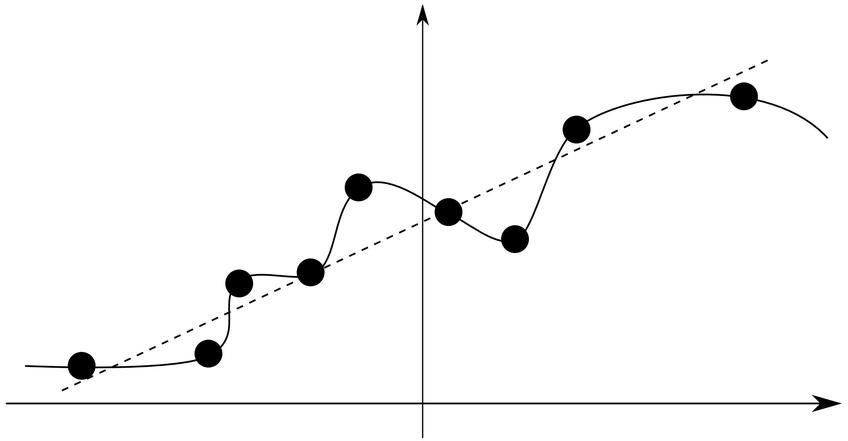
Normal equations

$$\mathbf{A}^T\mathbf{A}\mathbf{x} = \mathbf{A}^T\mathbf{b}$$

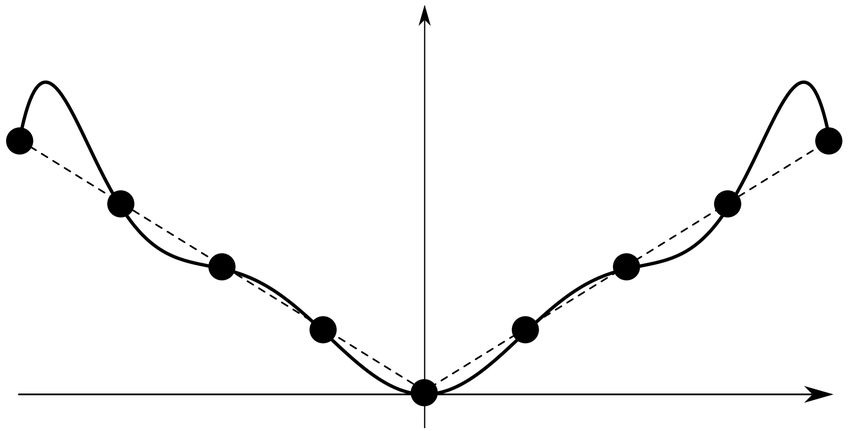


Regression and its limitations

Overfitting:



Wrong model:



[Justin Solomon]

Demo time

Use of the normal equations in practice

Okay, so what's the catch?

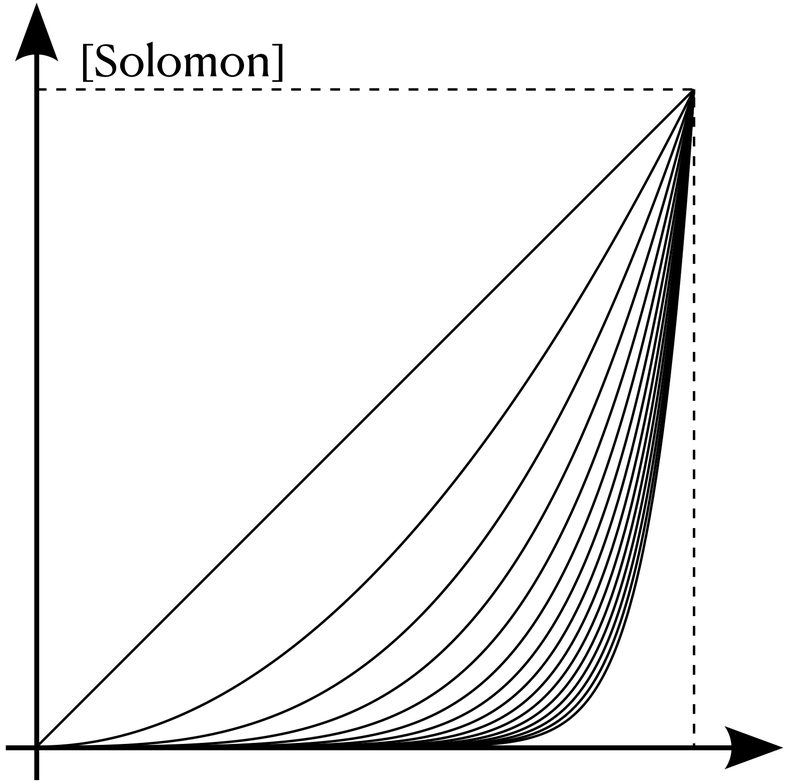
$$\text{cond}(\mathbf{A}^T\mathbf{A}) \approx \text{cond}(\mathbf{A})^2$$

The condition number of a problem expressed using the normal equations is often *much worse* that the condition number of the original problem.

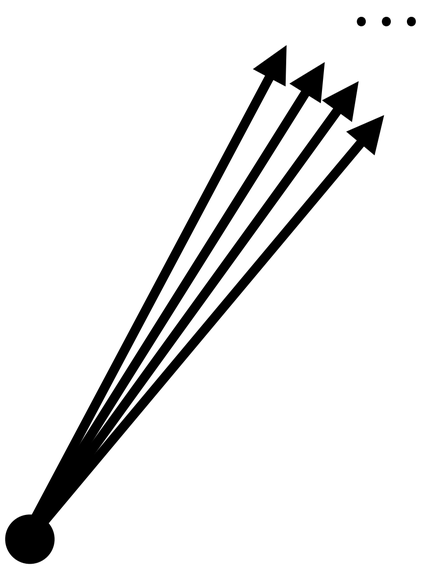
One of our goals for today: can we solve the same problem, but without ever building $\mathbf{A}^T\mathbf{A}$?

The issue with the Vandermonde matrix

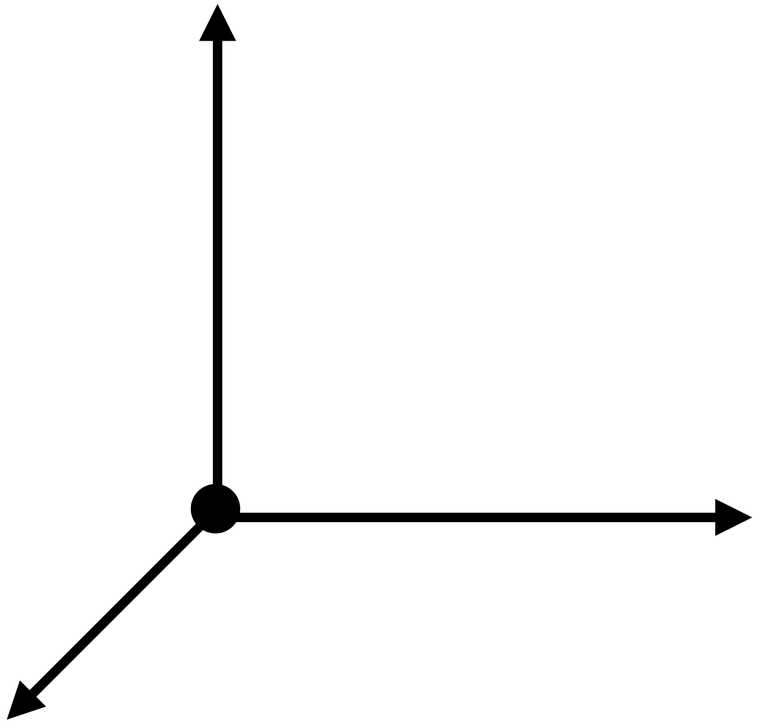
Geometric interpretation



Monomials of different powers



Almost parallel columns.

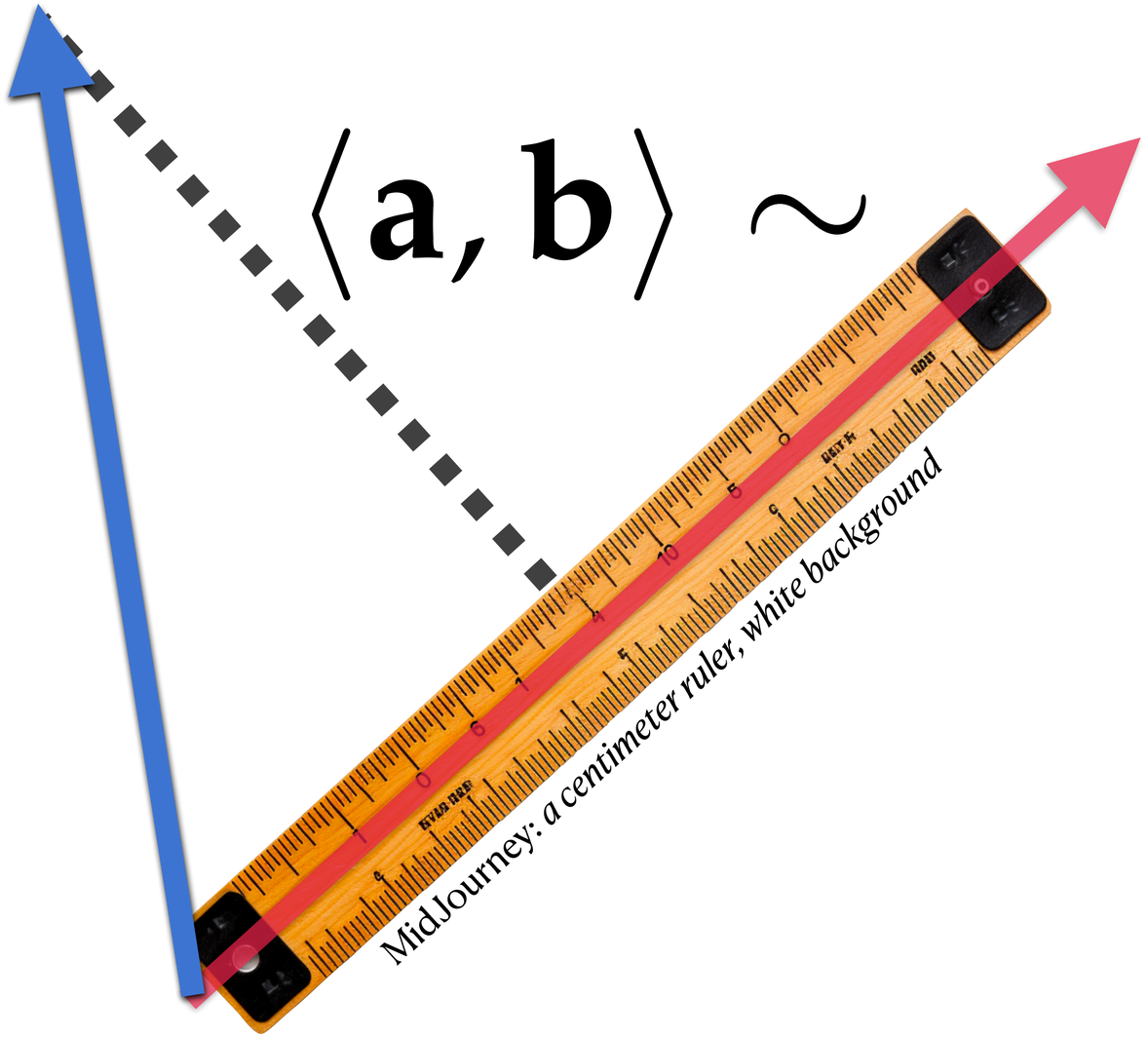


Idea: somehow make problem more "orthogonal"

Measuring orthogonality

The inner product measures a projection

$$\begin{aligned}\langle \mathbf{a}, \mathbf{b} \rangle &= \mathbf{a} \cdot \mathbf{b} \\ &= \mathbf{a}^T \mathbf{b} \\ &= \|\mathbf{a}\| \|\mathbf{b}\| \cos\theta\end{aligned}$$

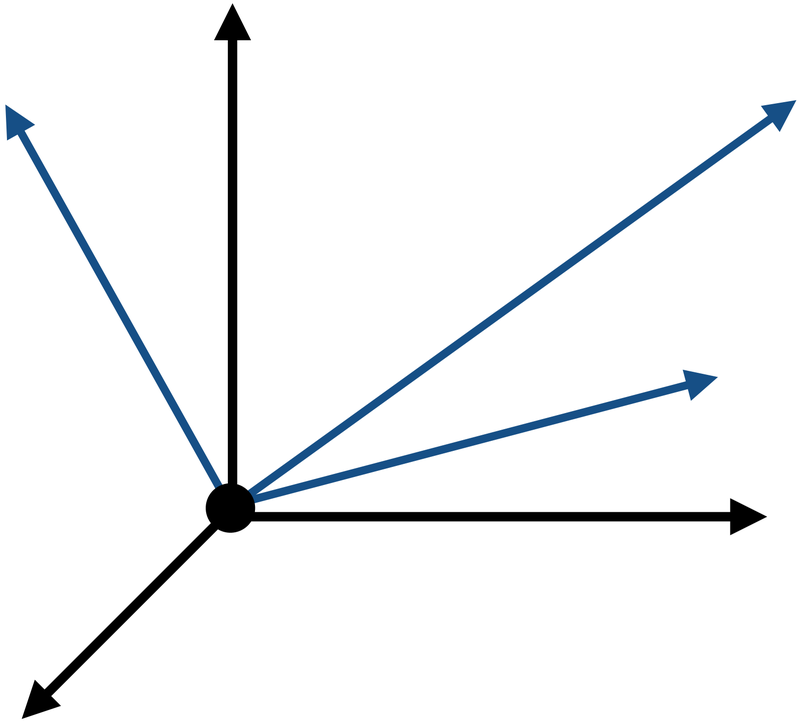


If $\mathbf{a}$ has unit length, $\mathbf{a}^T \mathbf{a} = 1$.

If $\mathbf{a}$ and $\mathbf{b}$ are perpendicular, $\mathbf{a}^T \mathbf{b} = 0$.

Gram-Schmidt orthogonalization

$$\mathbf{x} = \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix} \qquad \mathbf{y} = \begin{pmatrix} 4 \\ 5 \\ 6 \end{pmatrix}$$



$$\langle \mathbf{x}, \mathbf{y} \rangle = 1 \cdot 4 + 2 \cdot 5 + 3 \cdot 6 = 32$$

$$\boxed{\hat{\mathbf{y}} := \mathbf{y} - \frac{\langle \mathbf{x}, \mathbf{y} \rangle}{\langle \mathbf{x}, \mathbf{x} \rangle} \mathbf{x}}$$

Then $\langle \mathbf{x}, \hat{\mathbf{y}} \rangle = 0.$

Inner products are more general

$$\langle \mathbf{x}, \mathbf{y} \rangle := \sum_{i=1}^{n} x_i y_i$$

$$\langle f, g \rangle := \int_a^b f(x) g(x) \mathrm{d}x$$

Gram-schmidt applied to polynomials

Let's assume that everything is defined on the interval [-1, 1].

Input (monomials)

$$f(x) = 1$$

$$g(x) = x$$

$$h(x) = x^2$$

$$\langle f, g \rangle = \int_{-1}^{1} 1 \cdot x \, \mathrm{d}x = \left[\frac{1}{2} x^2 \right]_{-1}^{1} = 0$$

$$\langle g, h \rangle = \int_{-1}^{1} x \cdot x^2 \, \mathrm{d}x = \left[\frac{1}{4} x^4 \right]_{-1}^{1} = 0.$$

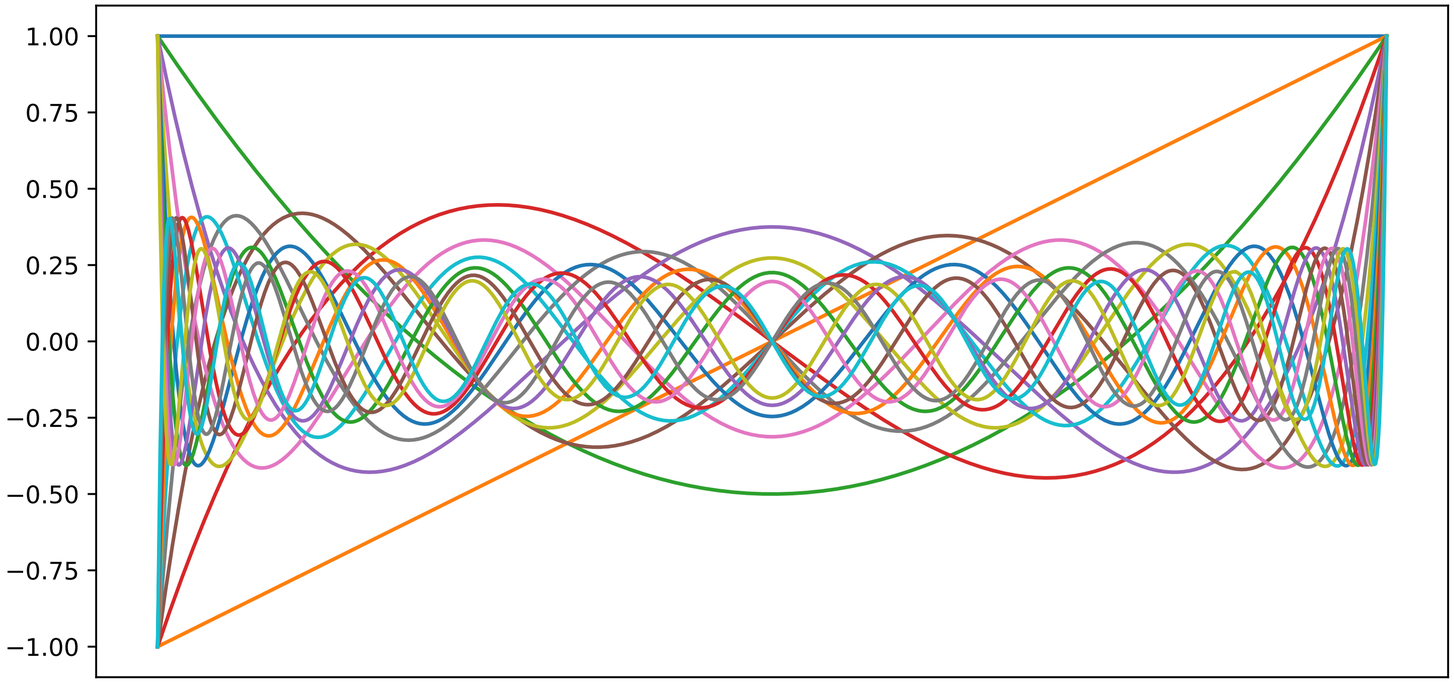
$$\langle f, h \rangle = \int_{-1}^{1} 1 \cdot x^2 \, \mathrm{d}x = \left[\frac{1}{3} x^3 \right]_{-1}^{1} = \frac{2}{3}$$

$$\hat{h}(x) = h(x) - \frac{\langle f, h \rangle}{\langle f, f \rangle} f(x) = x^2 - \frac{1}{3}$$

Demo time

Same problem, but once more with *Legendre Polynomials*

Legendre Polynomials: summary



- *Legrendre polynomials* span the same domain as monomials
  - but they are orthogonal to each other!
- Problems with poor condition number disappear completely!
- Use them when you need to fit a smooth function to some data.

Demo time

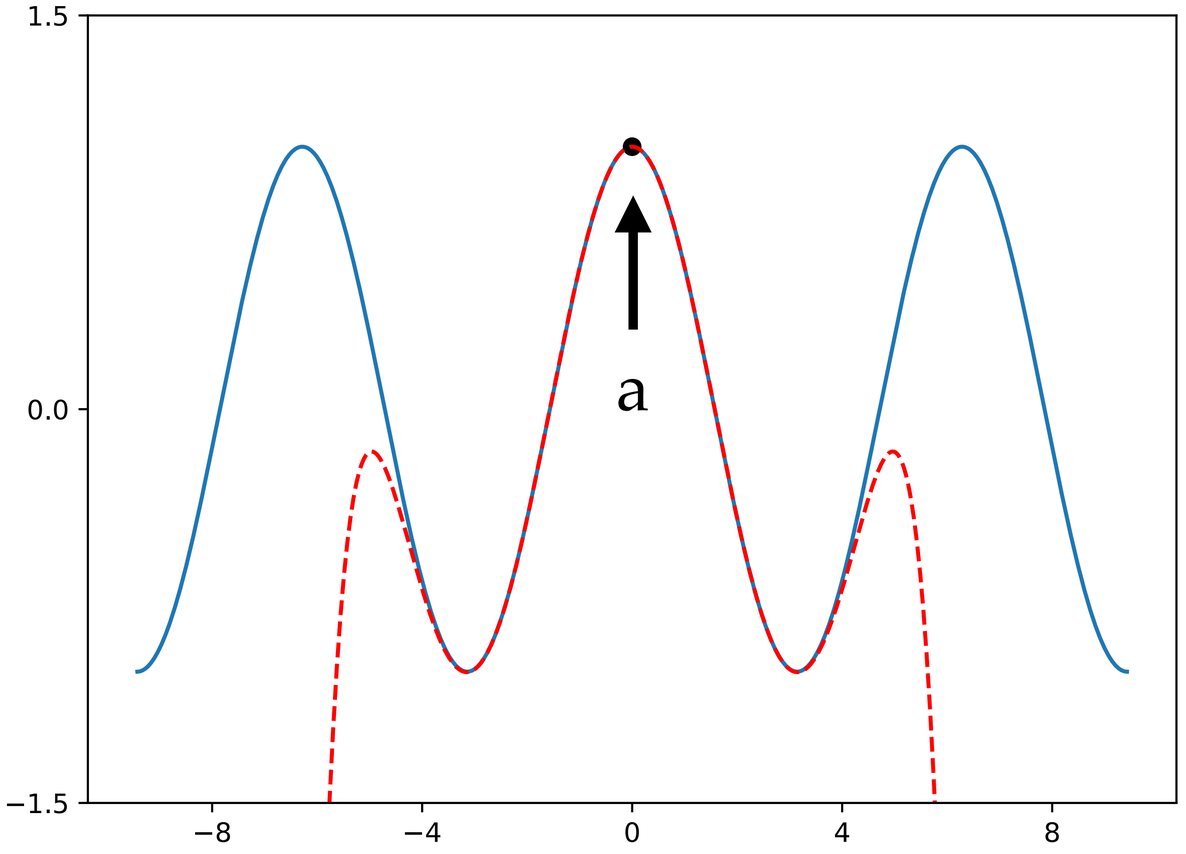
Runge function

Taylor's theorem

Approximation of a function around a point.

What about the error away from this point?

$$f(x) \approx f(a) + \dots$$



The residual of the approximation is equal to the $k+1^{\text{st}}$ derivative *somewhere* between the expansion point and where the function is evaluated.

Taylor's theorem for polynomials

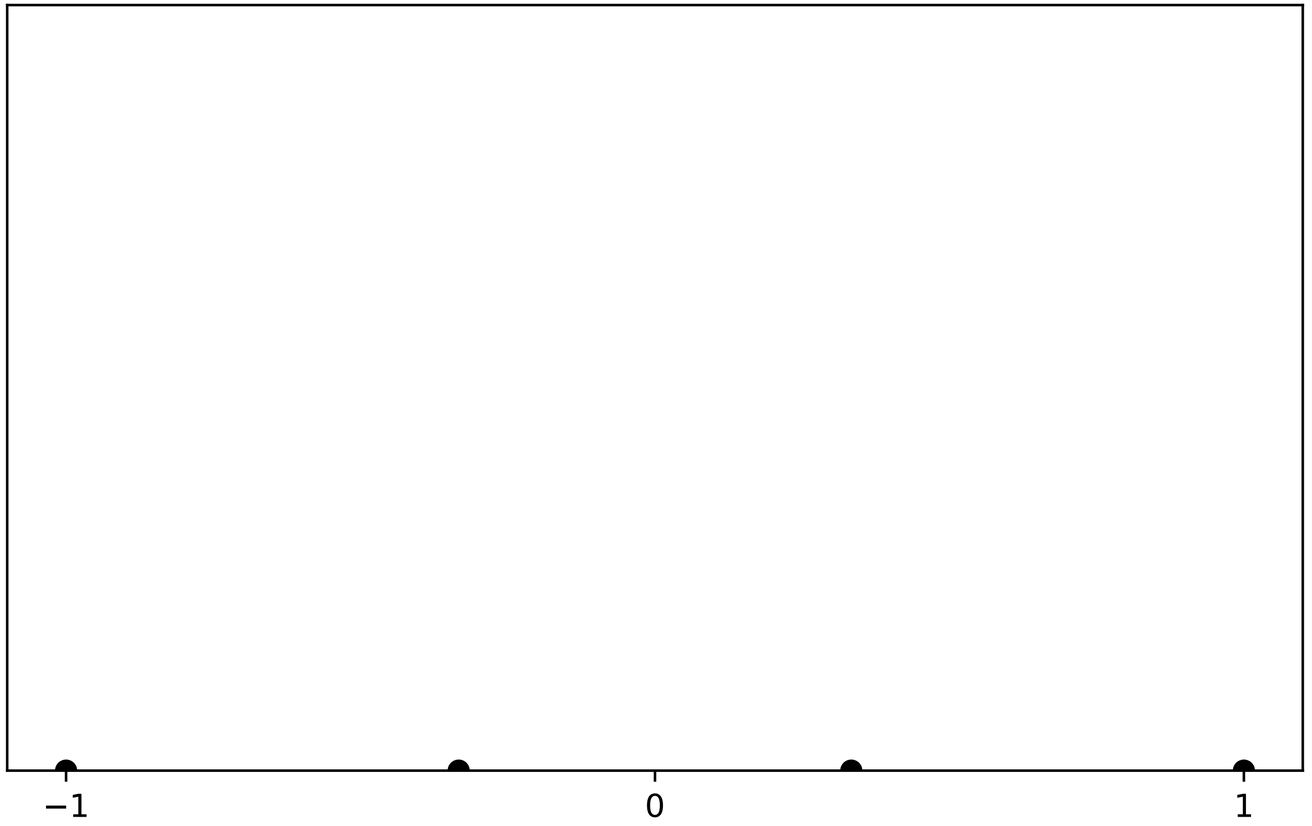
Approximation of a function around a sequence of points.

What about the error?

Residual (x_i are the points used for the fit)

$$R_{k-1}(x) = \frac{f^k(y)}{k!} \prod_{i=1}^{k} |x - x_i|$$

Idea: try to make this small



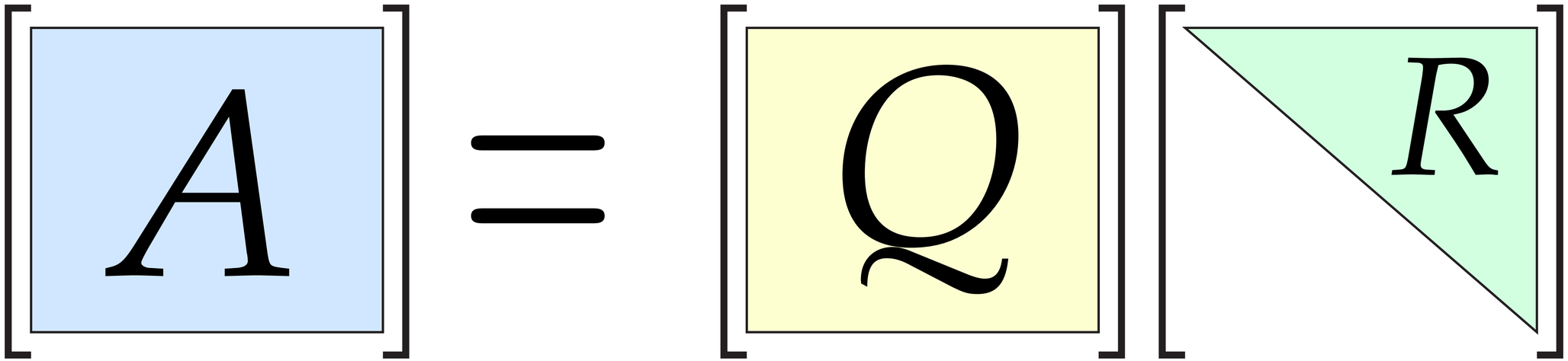
Optimal set of points that achieve this:

$$x_i = \cos\left(\frac{2k-1}{2n}\pi\right)$$

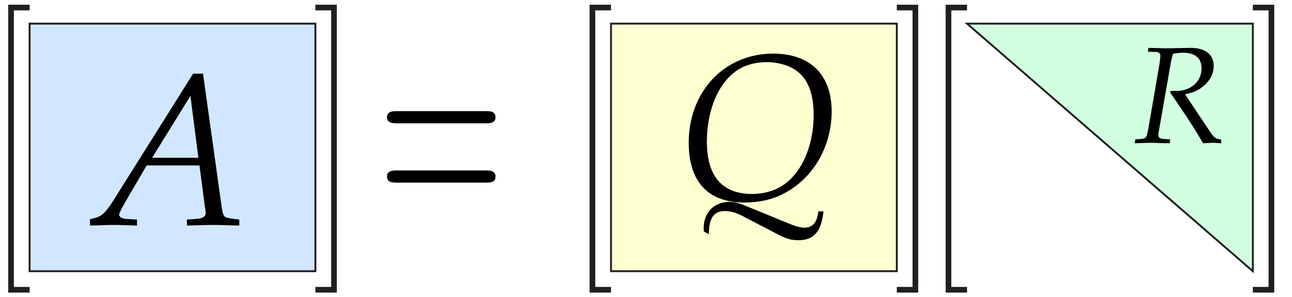
Demo time

Runge function, part II
(this time, using *Chebyshev* points)

The QR decomposition



The QR factorization

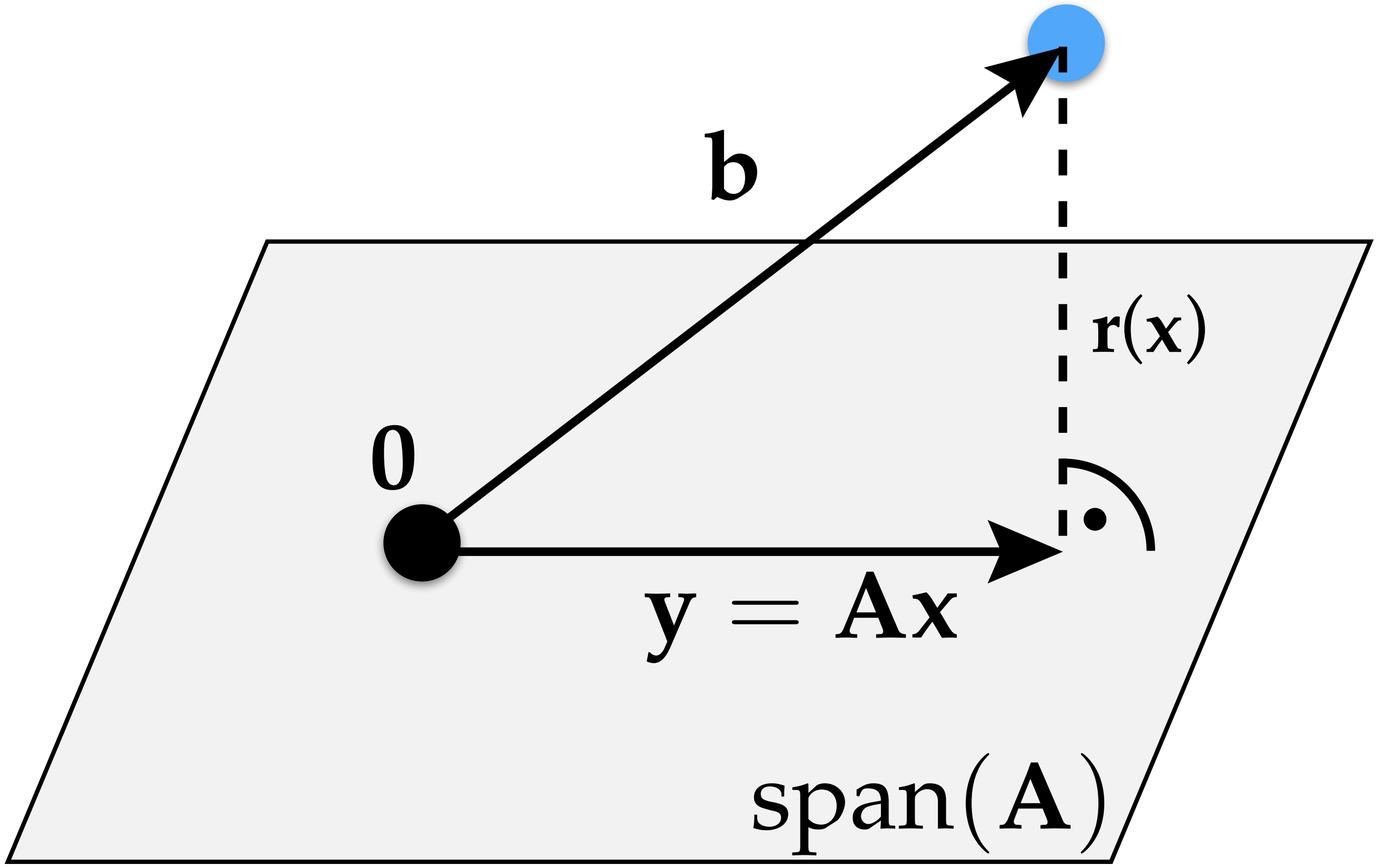


- *Different flavors exist*: Gram-Schmidt, Givens Rotations, **Housholder Transformations**
- Follows previous approach of using (pre-)computation to transform a "difficult" linear system into one that has a special structure, making it "easy".
- Compared to LU, one triangular factor was replaced by an **orthogonal matrix**.

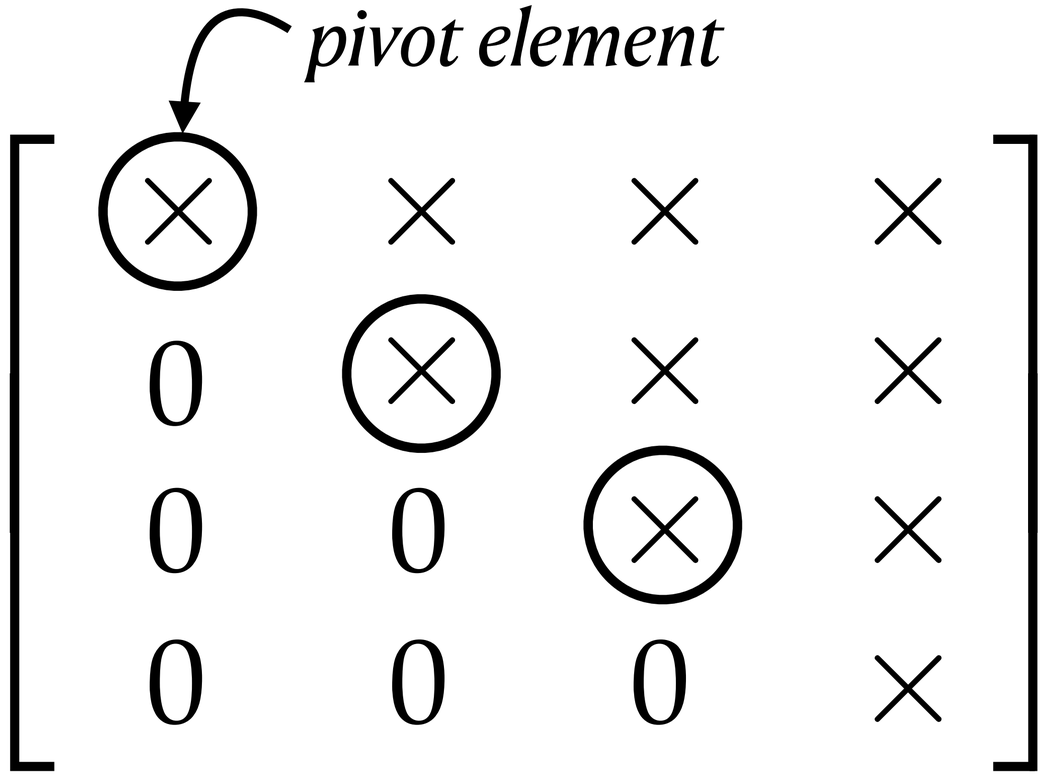
$$\mathbf{Q} = \begin{pmatrix} | & & | \\ \mathbf{q}_1 & \cdots & \mathbf{q}_n \\ | & & | \end{pmatrix} \qquad \mathbf{Q}^T\mathbf{Q} = \begin{pmatrix} \mathbf{q}_1^T\mathbf{q}_1 & \cdots & \mathbf{q}_1^T\mathbf{q}_n \\ \vdots & & \vdots \\ \mathbf{q}_n^T\mathbf{q}_1 & \cdots & \mathbf{q}_n^T\mathbf{q}_n \end{pmatrix} = \mathbf{I}.$$

Orthogonal transformations: preserve distances, angles. They do ***not*** amplify error!

Recap: the geometry of least squares problems



Motivation from the LU decomposition

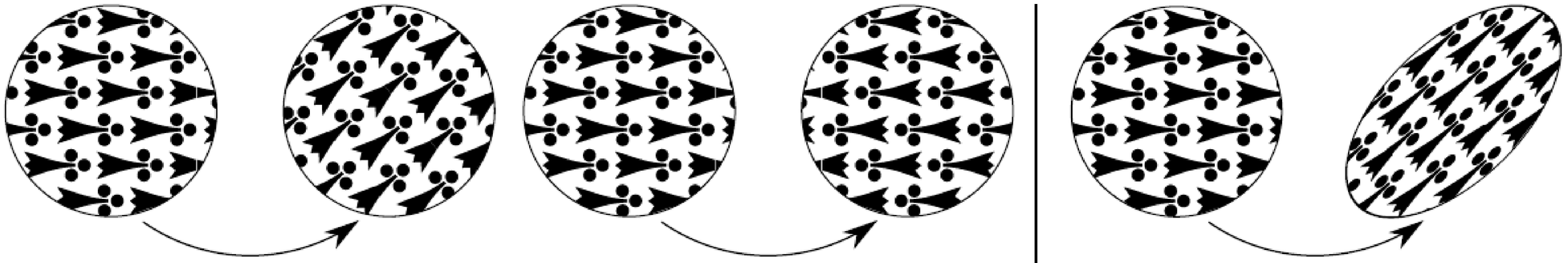


$\mathbf{A}$

Isometries

A transformation that preserves angles & distances is called an *isometry*.

[Justin Solomon]



rotate *mirror* *shear*

Isometric **Not isometric**

The problem: least squares problems seek the solution in $\mathrm{span}(\mathbf{A})$ that is *closest* to $\mathbf{b}$. If we non-isometrically transform the problem, then that actually changes the answer..

Recap: matrix transformations in 2D

Scaling:

$$\begin{bmatrix} \mathbf{v}'_x \\ \mathbf{v}'_y \end{bmatrix} = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix} \begin{bmatrix} \mathbf{v}_x \\ \mathbf{v}_y \end{bmatrix}$$

Shear:

$$\begin{bmatrix} \mathbf{v}'_x \\ \mathbf{v}'_y \end{bmatrix} = \begin{bmatrix} 1 & c \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{v}_x \\ \mathbf{v}_y \end{bmatrix}$$

Rotation:

$$\begin{bmatrix} \mathbf{v}'_x \\ \mathbf{v}'_y \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} \mathbf{v}_x \\ \mathbf{v}_y \end{bmatrix}$$

Mirror:

$$\begin{bmatrix} \mathbf{v}'_x \\ \mathbf{v}'_y \end{bmatrix} = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{v}_x \\ \mathbf{v}_y \end{bmatrix}$$

Not an Isometry!

QR decomposition

$$\begin{bmatrix} \times & \times & \times & \times \\ 0 & \times & \times & \times \\ 0 & 0 & \times & \times \\ 0 & 0 & 0 & \times \end{bmatrix}$$

A

"**R**" as in *right triangular*. Means the same thing as upper-triangular.

The challenge

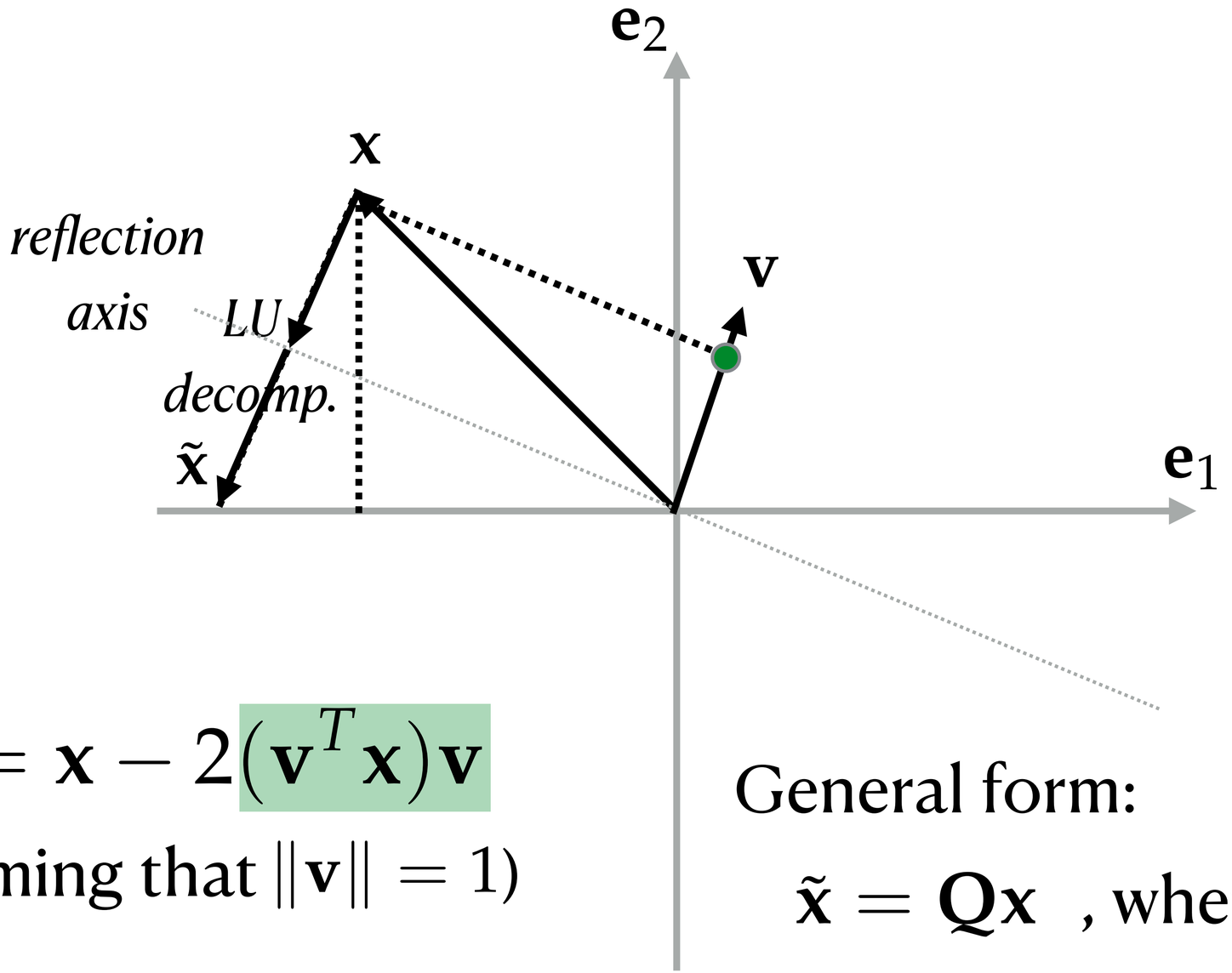
$$\mathbf{Q}\mathbf{x} = \mathbf{Q}\begin{pmatrix}\mathbf{x}_1\\ \mathbf{x}_2\\ \vdots\\ \mathbf{x}_n\end{pmatrix} = \begin{pmatrix}\tilde{\mathbf{x}}_1\\ 0\\ \vdots\\ 0\end{pmatrix}$$

Goal: zero out all elements of $\mathbf{x}$ except for the first.

Challenge: need to do this only using rotations or reflections..

Let's look at a 2D example!

Householder reflections



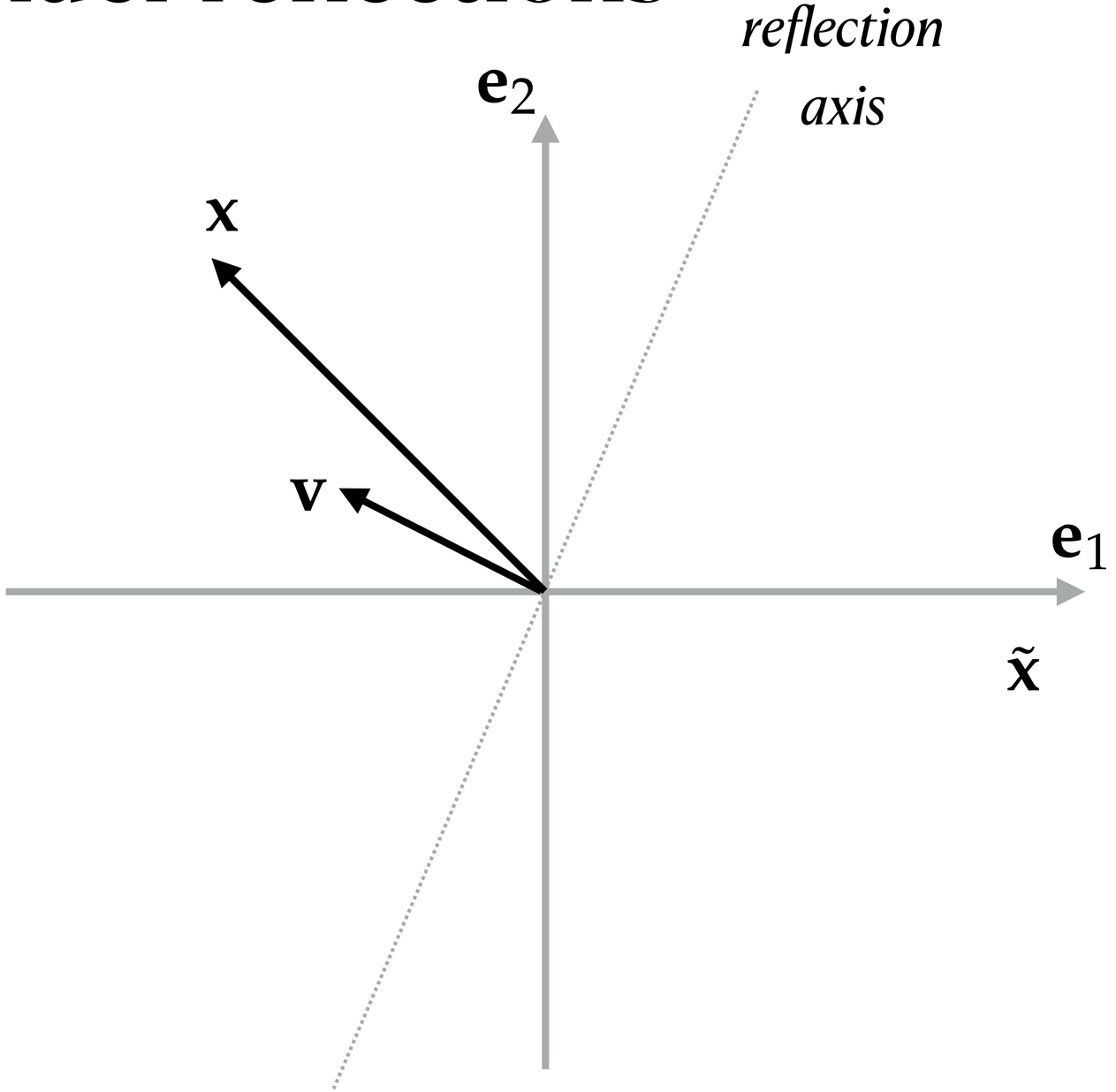
$$\tilde{\mathbf{x}} = \mathbf{x} - 2(\mathbf{v}^T\mathbf{x})\mathbf{v}$$

(assuming that $\|\mathbf{v}\| = 1$)

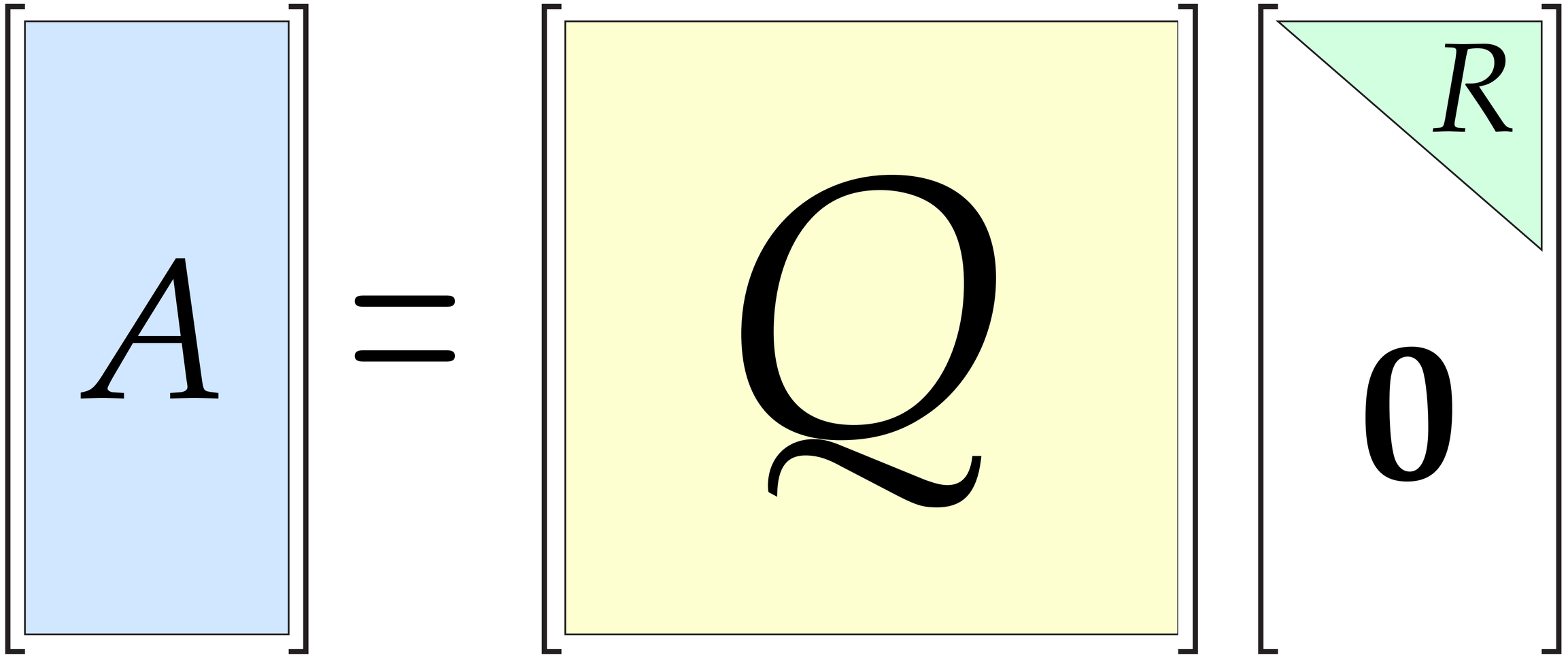
General form:

$$\tilde{\mathbf{x}} = \mathbf{Q}\mathbf{x} \quad \text{, where} \quad \mathbf{Q} = \mathbf{I} - 2\frac{\mathbf{v}\mathbf{v}^T}{\mathbf{v}^T\mathbf{v}}$$

Householder reflections

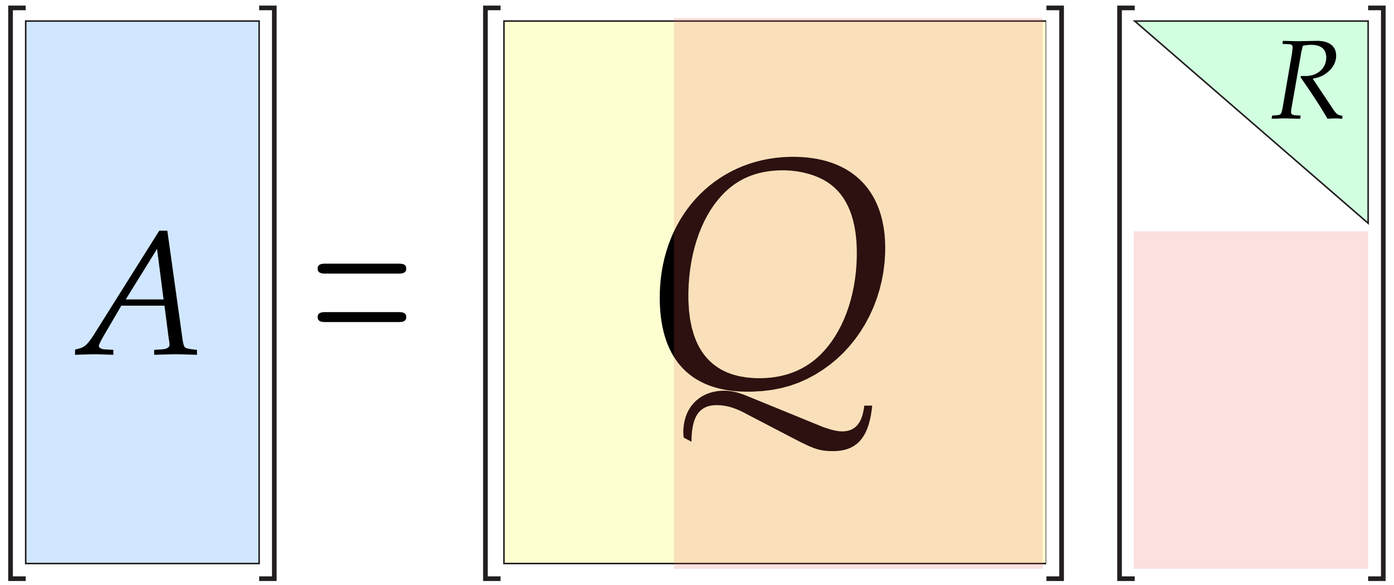


The end result

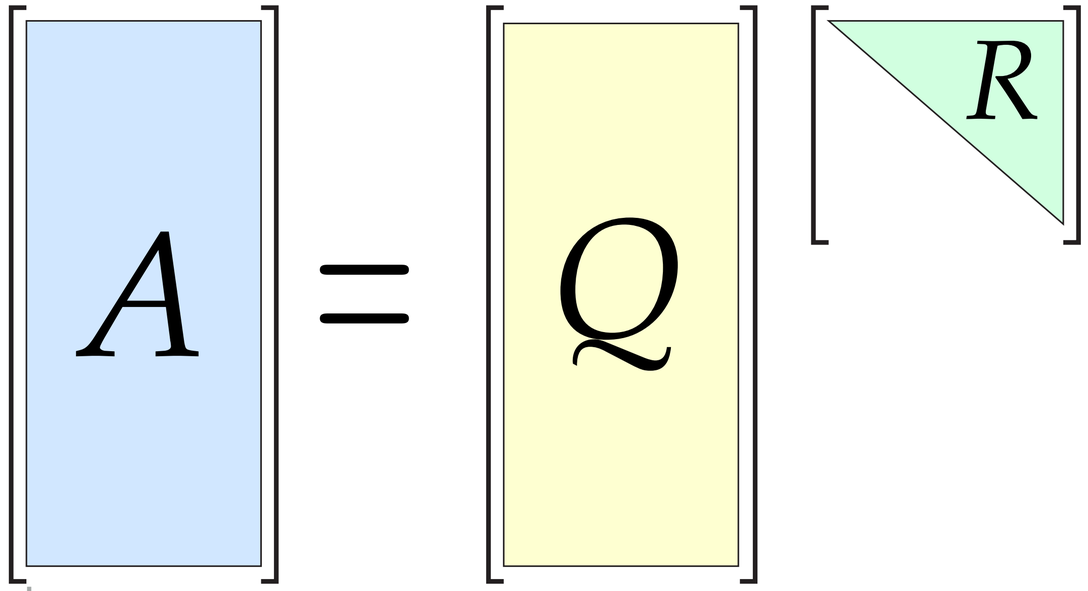


"Economy size" QR factorization

Normal QR:



Economy size:



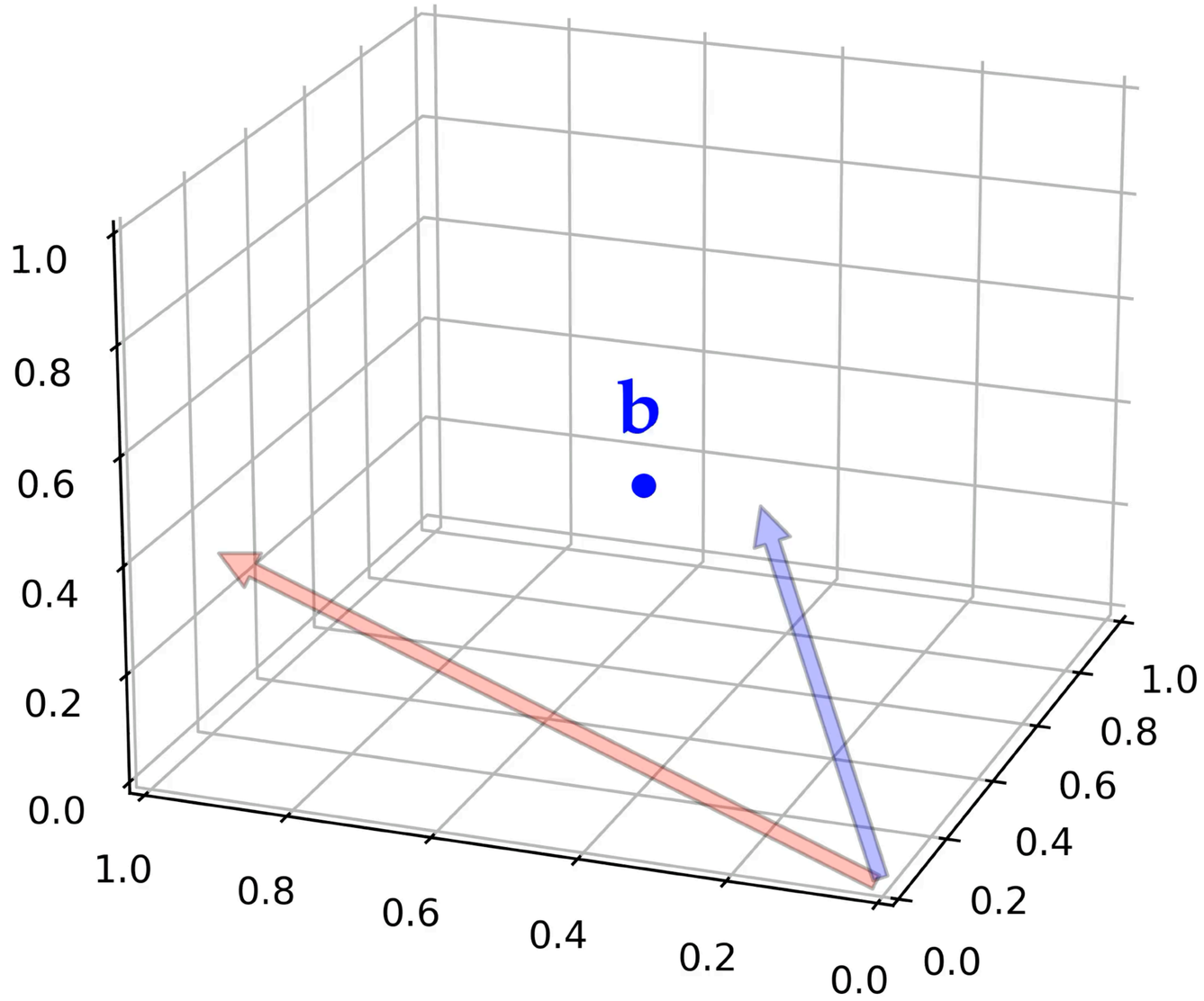
Using the QR factorization

$$\mathbf{x} = \left[\mathbf{R}\right]^{-1}\left(\left[\mathbf{Q}^T\right]\mathbf{b}\right)$$

- Be mindful to **not** compute an inverse of **R.** Instead, solve the upper-triangular system $\mathbf{R}\,\mathbf{x} = \mathbf{Q}^T\mathbf{b}$ via back-substitution!
- More expensive to compute than LU factorization (~2X). Can solve *tall* linear system, giving the least squares solution *without having to use the normal equations*.
- QR Factorization is numerically extremely well-behaved. Works even for singular (or zero-valued!) matrices.

Geometric interpretation

Rotate, project, then unwarp!



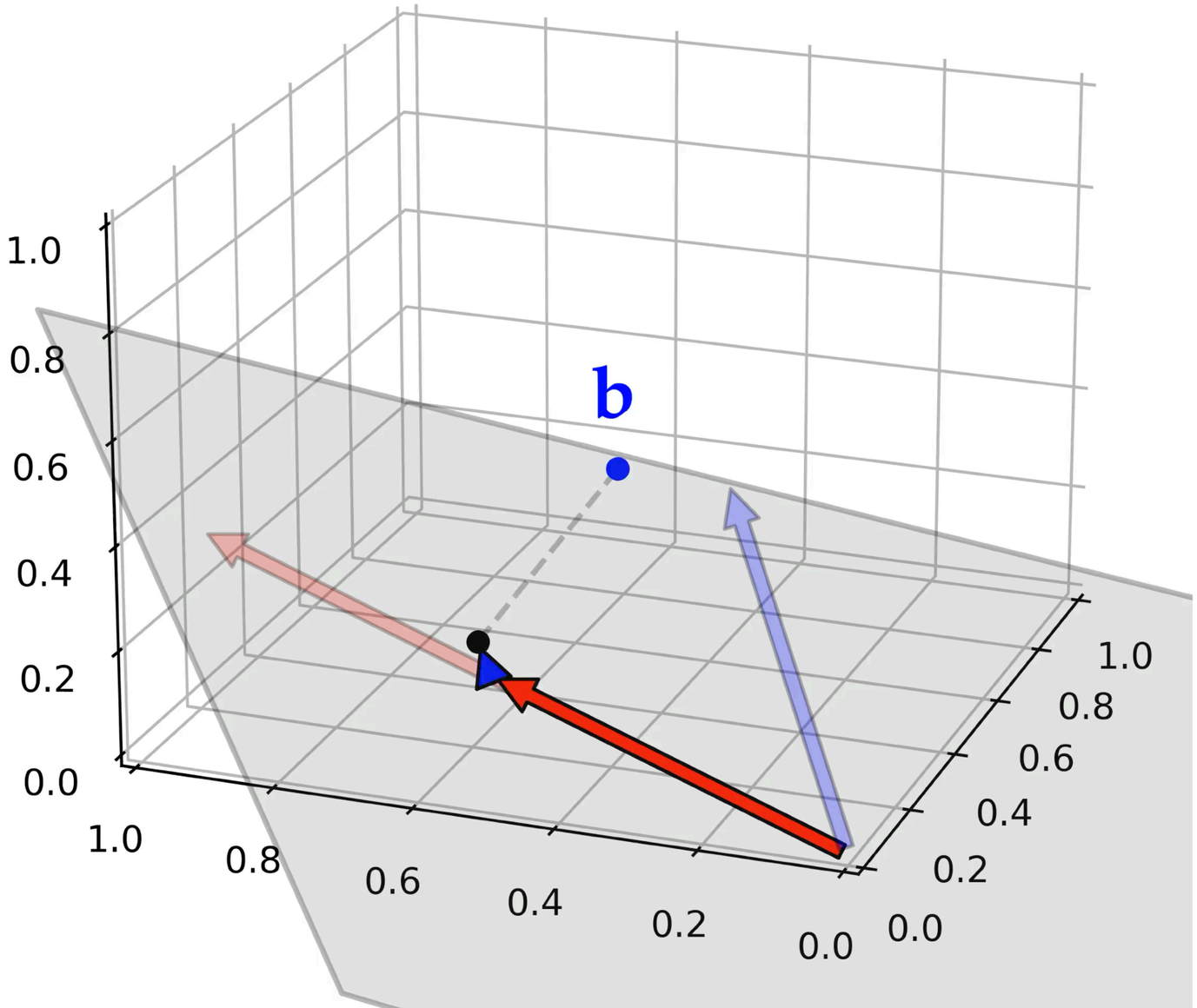
$$\mathbf{A} = \begin{pmatrix} | & | \\ \mathbf{a}^{(1)} & \mathbf{a}^{(2)} \\ | & | \end{pmatrix}$$

$$\mathbf{Q}, \mathbf{R} = \mathrm{qr}(\mathbf{A})$$

$$\mathbf{b}' = \mathbf{Q}^T \mathbf{b}$$

Geometric interpretation

Rotate, project, then unwarp!



$$\mathbf{Q}, \mathbf{R} = \mathrm{qr}(\mathbf{A})$$

$$\mathbf{b}' = \mathbf{Q}^T \mathbf{b}$$

Simpler *square* 2D problem!

$$\mathbf{x} = \mathbf{R}^{-1} \mathbf{b}'$$

Regularization

Regularization

- **Regularization** refers to methods that *improve the condition number* of a problem
  - There are many different ways to do so
  - No free lunch: better conditioning will come at the cost of decreased accuracy, bias, etc.
- **Tikhonov regularization**
  - assuming that a symptom of an unstable solution is that the solution $\mathbf{x}$'s magnitude becomes too large, we will penalize solutions with large L_2 norms

$$\mathbf{x}_{\text{solution}} = \text{argmin} \ \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2$$

- The regularization factor λ controls the degree to which we prefer smaller solutions

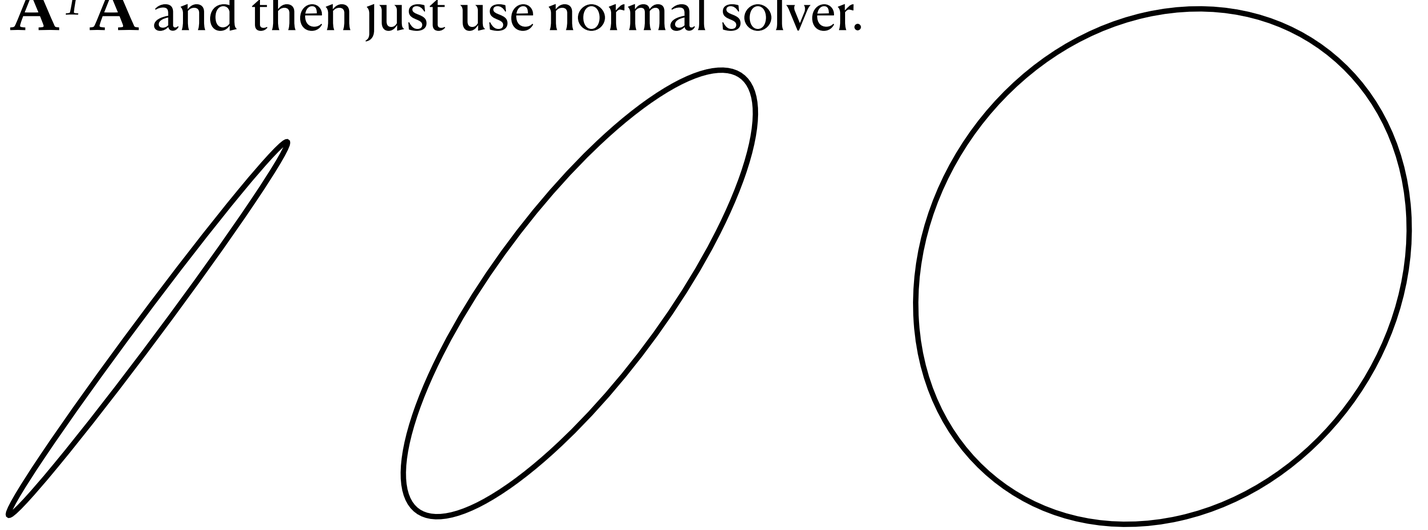
Tikhonov regularization

A simple way to reduce the condition number

$$\begin{aligned} & \text{argmin} \left[\mathbf{x}^T \mathbf{A}^T \mathbf{A} \mathbf{x} - 2\mathbf{b}^T \mathbf{A} \mathbf{x} + \|\mathbf{b}\|_2^2 + \lambda \|\mathbf{x}\|_2^2 \right] \\ = \ & \text{argmin} \left[\mathbf{x}^T \mathbf{A}^T \mathbf{A} \mathbf{x} - 2\mathbf{b}^T \mathbf{A} \mathbf{x} + \|\mathbf{b}\|_2^2 + \lambda \mathbf{x}^T \mathbf{I}^T \mathbf{I} \mathbf{x} \right] \end{aligned}$$

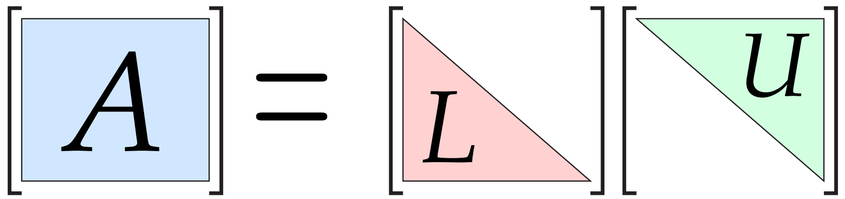
Trick: merge identity into $\mathbf{A}^T\mathbf{A}$ and then just use normal solver.

Geometric interpretation: the addition of an identity matrix causes the matrix to scale input vectors more uniformly in different directions.



small λ medium λ big λ

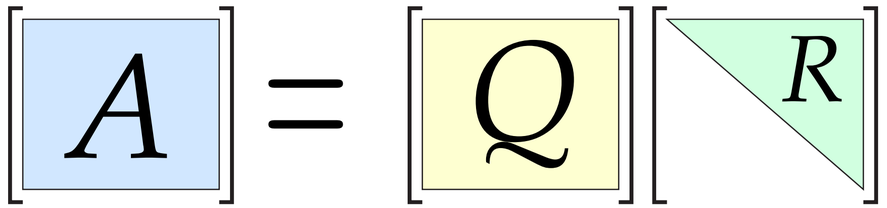
Tikhonov regularization — technical details



LU Factorization:

- simply add $\lambda \mathbf{I}$ to to $\mathbf{A}^T \mathbf{A}$ (via normal eqns)

... and solve!



QR Factorization

- Factorize the extended matrix

$$\begin{bmatrix} \mathbf{A} \\ \sqrt{\lambda}\mathbf{I} \end{bmatrix}$$

instead of $\mathbf{A}$, append n zeros to the right hand side $\mathbf{b}$.