Regularization

Adding the Identity

- Add $Ic = 0$ to drive components related to small/zero singular values to zero
  - Motivated by minimal norm solution
- Combine with the original system $\begin{pmatrix} A \\ I \end{pmatrix} c = \begin{pmatrix} b \\ 0 \end{pmatrix}$ so that $\begin{pmatrix} A \\ I \end{pmatrix}$ has full column rank
  - Can be solved with Householder, etc.
- Normal equations: $\begin{pmatrix} A^T & I \end{pmatrix} \begin{pmatrix} A \\ I \end{pmatrix} c = \begin{pmatrix} A^T & I \end{pmatrix} \begin{pmatrix} b \\ 0 \end{pmatrix}$ or $(A^T A + I)c = A^T b$
- Use $A = U\Sigma V^T$ to get $(V\Sigma^T \Sigma V^T + I)c = V\Sigma^T \hat{b}$ or $(\Sigma^T \Sigma + I)\hat{c} = \Sigma^T \hat{b}$
- Use $\Sigma = \begin{pmatrix} \hat{\Sigma} & 0 \\ 0 & 0 \end{pmatrix}$ to get $\left(\begin{pmatrix} \hat{\Sigma}^T & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} \hat{\Sigma} & 0 \\ 0 & 0 \end{pmatrix} + I \right) \begin{pmatrix} \hat{c}_r \\ \hat{c}_z \end{pmatrix} = \begin{pmatrix} \hat{\Sigma}^T & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} \hat{b}_r \\ \hat{b}_z \end{pmatrix}$
- Then $\left(\begin{pmatrix} \hat{\Sigma}^2 & 0 \\ 0 & 0 \end{pmatrix} + I \right) \begin{pmatrix} \hat{c}_r \\ \hat{c}_z \end{pmatrix} = \begin{pmatrix} \hat{\Sigma} \hat{b}_r \\ 0 \end{pmatrix}$, which gives $\hat{c}_z = 0$ as desired

Perturbation

- However, $\left(\begin{pmatrix} \hat{\Sigma}^2 & 0 \\ 0 & 0 \end{pmatrix} + I\right)\begin{pmatrix} \hat{c}_r \\ \hat{c}_z \end{pmatrix} = \begin{pmatrix} \hat{\Sigma}\hat{b}_r \\ 0 \end{pmatrix}$ perturbs the equations for the $\hat{c}_r$ terms as well
- Instead of the usual $\hat{c}_k = \frac{1}{\sigma_k}\hat{b}_k$, the new solution is $\hat{c}_k = \frac{\sigma_k}{\sigma_k^2+1}\hat{b}_k$
  - This perturbs these $\hat{c}_k$ away from their correct (unique or least squares) solution
  - Adding $Ic = 0$ interferes with $Ac = b$ for the $\hat{c}_k$ with $\sigma_k \neq 0$
- For larger $\sigma_k \gg 1$, $\frac{\sigma_k}{\sigma_k^2+1} \approx \frac{1}{\sigma_k}$ and the perturbation of the (unique or least squares) solution is negligible
- For $\sigma_k \approx 1$, the perturbation is quite large
- For $\sigma_k \ll 1$, $\frac{\sigma_k}{\sigma_k^2+1} \approx 0$ drives the associated $\hat{c}_k$ towards zero

Regularization

- Adding $\epsilon I c = 0$ (with $\epsilon > 0$) instead of $Ic = 0$, that is $\begin{pmatrix} A \\ \epsilon I \end{pmatrix} c = \begin{pmatrix} b \\ 0 \end{pmatrix}$
- Normal equations: $\begin{pmatrix} A^T & \epsilon I \end{pmatrix} \begin{pmatrix} A \\ \epsilon I \end{pmatrix} c = \begin{pmatrix} A^T & \epsilon I \end{pmatrix} \begin{pmatrix} b \\ 0 \end{pmatrix}$ or $(A^T A + \epsilon^2 I)c = A^T b$
- This results in a modified $\left(\begin{pmatrix} \hat{\Sigma}^2 & 0 \\ 0 & 0 \end{pmatrix} + \epsilon^2 I \right) \begin{pmatrix} \hat{c}_r \\ \hat{c}_z \end{pmatrix} = \begin{pmatrix} \hat{\Sigma} \hat{b}_r \\ 0 \end{pmatrix}$
- Instead of the usual $\hat{c}_k = \frac{1}{\sigma_k} \hat{b}_k$, the new solution is $\hat{c}_k = \frac{\sigma_k}{\sigma_k^2 + \epsilon^2} \hat{b}_k$
- This has limited effect on $\sigma_k \gg \epsilon$
- This helps to stabilize/regularize the solution for $\sigma_k \approx \epsilon$ and $\sigma_k < \epsilon$
  - driving the associated $\hat{c}_k$ towards zero

A Nonzero Initial Guess

- Can view setting $Ic = 0$ as guessing a solution of $c = 0$
- Instead, suppose one had an initial guess of $c = c^*$
- Add $Ic = c^*$ to the equations to get: $\begin{pmatrix} A \\ I \end{pmatrix} c = \begin{pmatrix} b \\ c^* \end{pmatrix}$
- Normal equations: $(A^T A + I)c = A^T b + c^*$
- This leads to $(\Sigma^T \Sigma + I)\hat{c} = \Sigma^T \hat{b} + V^T c^* = \Sigma^T \hat{b} + \hat{c}^*$
- Then, $\hat{c}_k = \frac{\sigma_k}{\sigma_k^2+1} \hat{b}_k + \frac{1}{\sigma_k^2+1} \hat{c}_k^*$ tends towards $\hat{b}_k$ for larger σ_k (as desired) but tends towards $\hat{c}_k^*$ for smaller σ_k (with $\hat{c}_k = \hat{c}_k^*$ for any $\sigma_k = 0$)
- Adding $\epsilon Ic = \epsilon c^*$ gives $\hat{c}_k = \frac{\sigma_k}{\sigma_k^2+\epsilon^2} \hat{b}_k + \frac{\epsilon^2}{\sigma_k^2+\epsilon^2} \hat{c}_k^*$

A Nonzero Initial Guess

- Rewrite this as $\hat{c}_k = \left(\frac{\sigma_k^2}{\sigma_k^2+\epsilon^2}\right)\frac{\hat{b}_k}{\sigma_k} + \left(\frac{\epsilon^2}{\sigma_k^2+\epsilon^2}\right)\hat{c}_k^*$
  - Note the convex weights: $\left(\frac{\sigma_k^2}{\sigma_k^2+\epsilon^2}\right)+\left(\frac{\epsilon^2}{\sigma_k^2+\epsilon^2}\right) = 1$
- This is a convex combination of the (unique or least squares) solution $\frac{\hat{b}_k}{\sigma_k}$ and the initial guess $\hat{c}_k^*$
  - Also valid for an initial guess of $\hat{c}_k^* = 0$
- Large σ_k ($\sigma_k \gg \epsilon$) tend toward the usual solution: $\hat{c}_k = \frac{\hat{b}_k}{\sigma_k}$
- Small σ_k ($\sigma_k \ll \epsilon$) tend toward the initial guess: $\hat{c}_k = \hat{c}_k^*$

An Iterative Approach

- First, solve with $\epsilon I c = 0$ to get $\hat{c}_k = \left(\frac{\sigma_k^2}{\sigma_k^2+\epsilon^2}\right)\frac{\hat{b}_k}{\sigma_k}$
- Then, use this solution as an initial guess and solve again to get:

$$\hat{c}_k = \left(\frac{\sigma_k^2}{\sigma_k^2+\epsilon^2}\right)\frac{\hat{b}_k}{\sigma_k} + \left(\frac{\epsilon^2}{\sigma_k^2+\epsilon^2}\right)\left(\frac{\sigma_k^2}{\sigma_k^2+\epsilon^2}\right)\frac{\hat{b}_k}{\sigma_k} = \left(1+\left(\frac{\epsilon^2}{\sigma_k^2+\epsilon^2}\right)\right)\left(\frac{\sigma_k^2}{\sigma_k^2+\epsilon^2}\right)\frac{\hat{b}_k}{\sigma_k}$$

- Then, use this solution as an initial guess and solve again to get:

$$\hat{c}_k = \left(\frac{\sigma_k^2}{\sigma_k^2+\epsilon^2}\right)\frac{\hat{b}_k}{\sigma_k} + \left(\frac{\epsilon^2}{\sigma_k^2+\epsilon^2}\right)\left(1+\left(\frac{\epsilon^2}{\sigma_k^2+\epsilon^2}\right)\right)\left(\frac{\sigma_k^2}{\sigma_k^2+\epsilon^2}\right)\frac{\hat{b}_k}{\sigma_k}$$

$$= \left(1+\left(\frac{\epsilon^2}{\sigma_k^2+\epsilon^2}\right)+\left(\frac{\epsilon^2}{\sigma_k^2+\epsilon^2}\right)^2\right)\left(\frac{\sigma_k^2}{\sigma_k^2+\epsilon^2}\right)\frac{\hat{b}_k}{\sigma_k}$$

Convergence

- Continuing leads to $\hat{c}_k = \left(1 + \left(\frac{\epsilon^2}{\sigma_k^2+\epsilon^2}\right) + \left(\frac{\epsilon^2}{\sigma_k^2+\epsilon^2}\right)^2 + \left(\frac{\epsilon^2}{\sigma_k^2+\epsilon^2}\right)^3 + \cdots\right)\left(\frac{\sigma_k^2}{\sigma_k^2+\epsilon^2}\right)\frac{\hat{b}_k}{\sigma_k}$
- The geometric series in parenthesis has $r = \frac{\epsilon^2}{\sigma_k^2+\epsilon^2}$
- It converges to $\frac{1}{1-r} = \frac{\sigma_k^2+\epsilon^2}{\sigma_k^2}$ giving $\hat{c}_k = \frac{\hat{b}_k}{\sigma_k}$ in the limit (as desired)
- When $\sigma_k = 0$, the convex weights are 0 and 1, so $\hat{c}_k = 0$ identically at every step
  - This is the desired minimum norm solution for these σ_k

Convergence Rate

- After q iterations, the geometric series sums to $\frac{1-r^q}{1-r} = \frac{\sigma_k^2+\epsilon^2}{\sigma_k^2}\left(1 - \left(\frac{\epsilon^2}{\sigma_k^2+\epsilon^2}\right)^q\right)$
- This gives $\hat{c}_k = \left(1 - \left(\frac{\epsilon^2}{\sigma_k^2+\epsilon^2}\right)^q\right)\frac{\hat{b}_k}{\sigma_k}$ implying monotonic convergence to $\hat{c}_k = \frac{\hat{b}_k}{\sigma_k}$
  - since $r = \left(\frac{\epsilon^2}{\sigma_k^2+\epsilon^2}\right) < 1$ implies $r^q \to 0$ monotonically as $q \to \infty$
- The convergence is quick for large σ_k (as desired)
- Smaller σ_k have $\frac{\epsilon^2}{\sigma_k^2+\epsilon^2}$ closer to 1, so their $\hat{c}_k$ increase more slowly from zero towards $\frac{\hat{b}_k}{\sigma_k}$ (smaller σ_k are thus regularized)

Comparison with PCA

- After q iterations, PCA incorporates the q largest σ_k components into the solution
- PCA does not include any contribution (at all) for the other components
  - Smaller σ_k components are Heaviside thresholded to be identically zero

- After q iterations, this iterative approach does not include the full contribution of the q largest σ_k components
  - It includes $1 - r_k^q$ times those components, but $1 - r_k^q \approx 1$ when σ_k is large
- This iterative approach includes contributions from all components
  - The contribution from smaller σ_k components is smaller, since their $1 - r_k^q$ is not as close to 1 when σ_k is small
  - This iterative approach has a significantly smoother fall-off as σ_k decreases

Aside

- This iterative method and the analysis via a geometric series (slides 7-10) were derived in preparation for the Winter 2019 offering of this course
  - Hyde, D., Bao, M., and Fedkiw, R., "On Obtaining Sparse Semantic Solutions for Inverse Problems, Control, and Neural Network Training", J. Comp. Phys. 443, 110498 (2021).

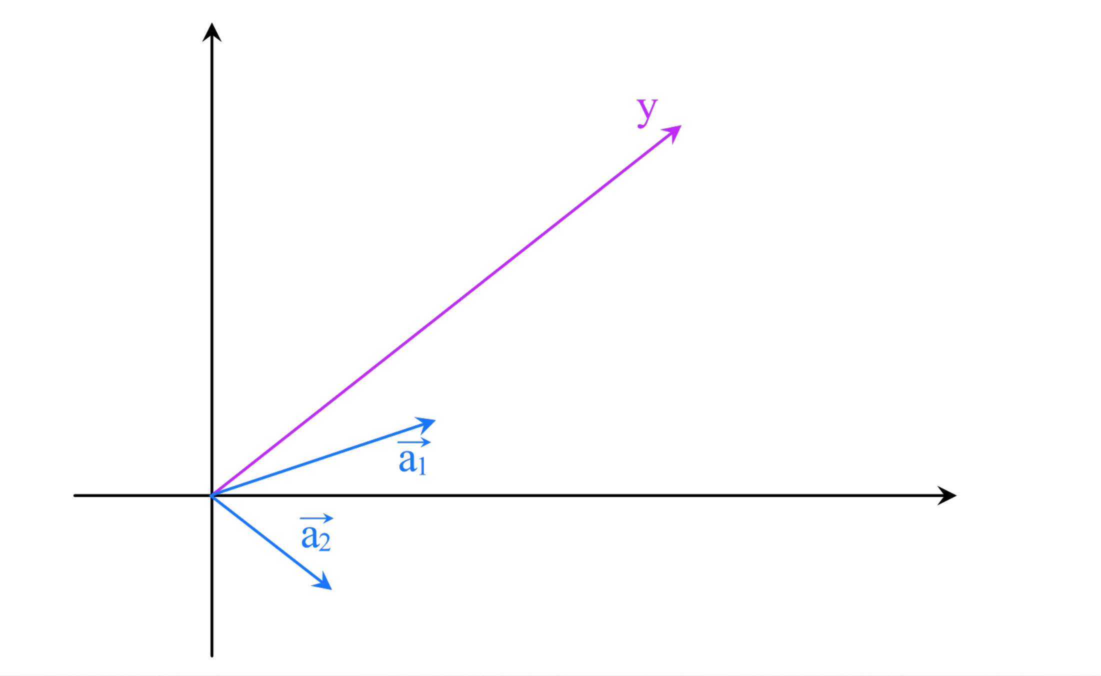
- The non-iterative version of the method is a version of Levenberg-Marquardt

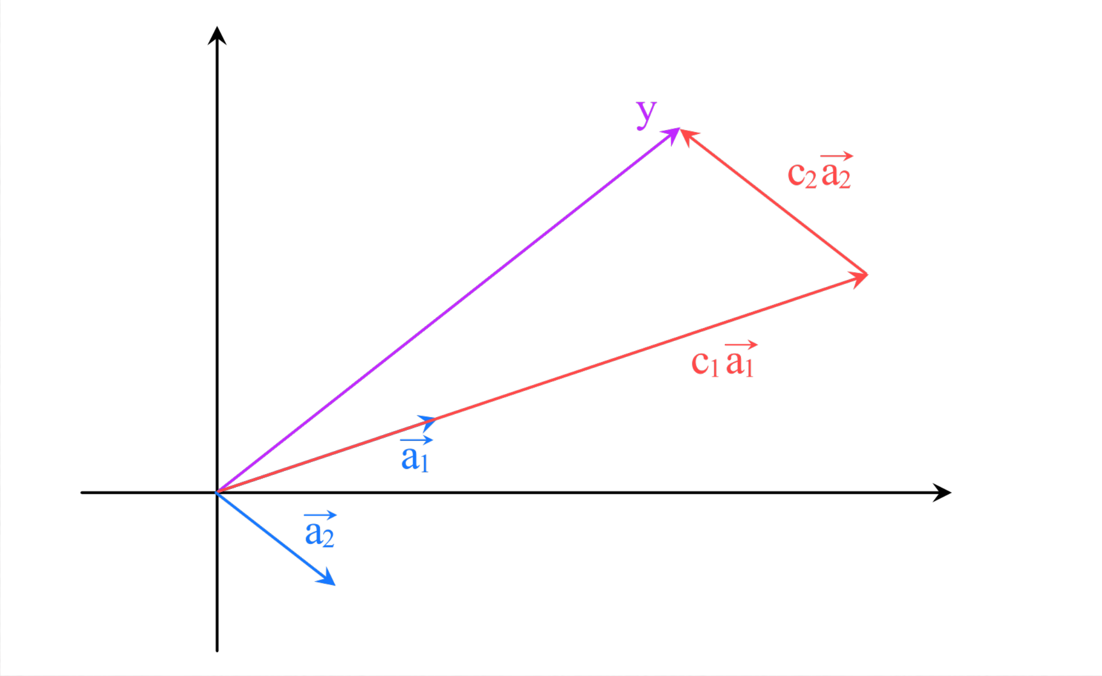
Adding a Diagonal Matrix

- Adding $Dc = 0$ to obtain: $\begin{pmatrix} A \\ D \end{pmatrix} c = \begin{pmatrix} b \\ 0 \end{pmatrix}$ drives some variables more strongly towards zero than others
- The normal equations are $(A^T A + D^2)c = A^T b$
- Equivalently $(V\Sigma^T \Sigma V^T + D^2)c = V\Sigma^T \hat{b}$ or $(\Sigma^T \Sigma + V^T D^2 V)\hat{c} = \Sigma^T \hat{b}$
- These normal equations can also be derived starting from $\begin{pmatrix} \Sigma \\ DV \end{pmatrix} \hat{c} = \begin{pmatrix} \hat{b} \\ 0 \end{pmatrix}$
  - Unfortunately, D shears the vectors in V creating issues
- This motivates first column scaling $\begin{pmatrix} AD^{-1} \\ I \end{pmatrix} Dc = \begin{pmatrix} b \\ 0 \end{pmatrix}$ to obtain an $\begin{pmatrix} \tilde{A} \\ I \end{pmatrix} \tilde{c} = \begin{pmatrix} b \\ 0 \end{pmatrix}$ that can be treated in the original way (by adding $I\tilde{c} = 0$)

Recall: Matrix Columns as Vectors (unit 1)

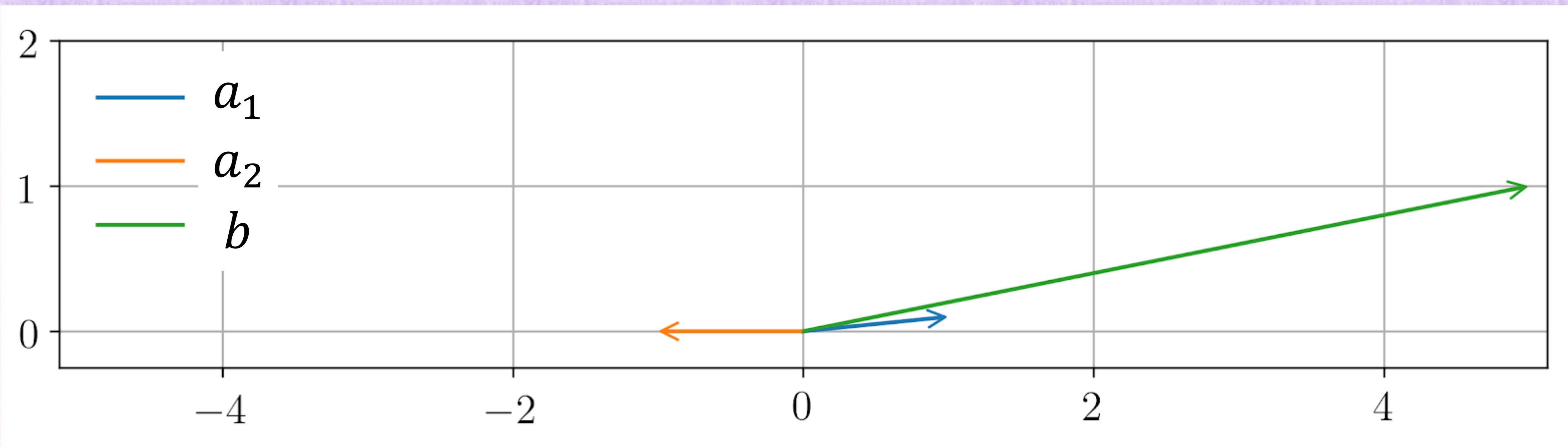
- Let the k-th column of A be vector a_k, so $Ac = y$ is equivalent to $\sum_k c_k a_k = y$
- Find a linear combination of the columns of A that gives the right hand side vector y





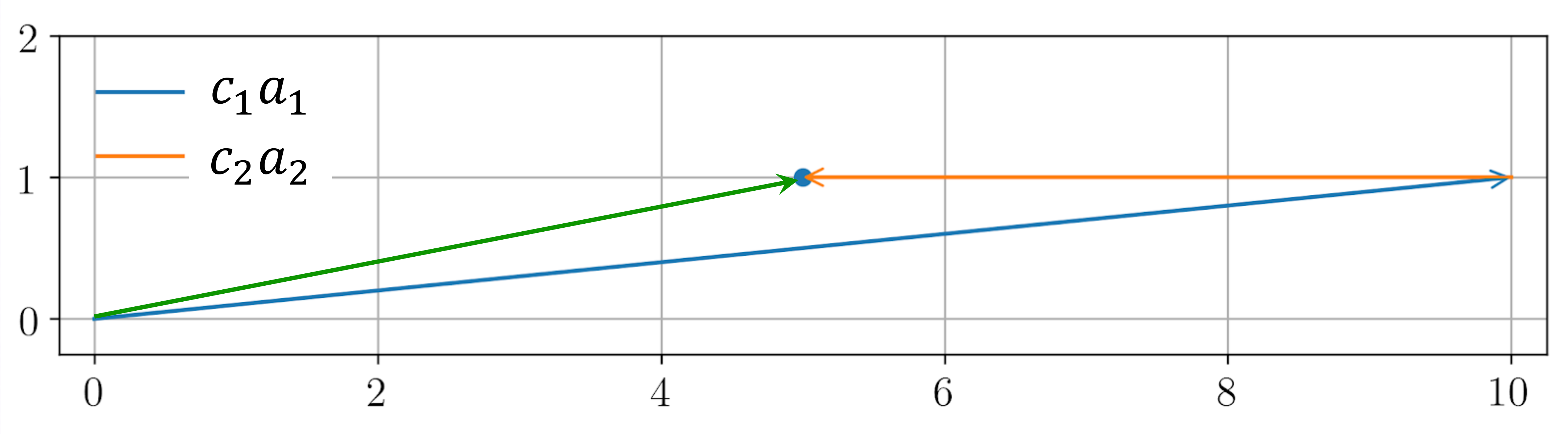
An Example

- Determine c_1 and c_2 such that $c_1 a_1 + c_2 a_2 = b$ or $Ac = b$



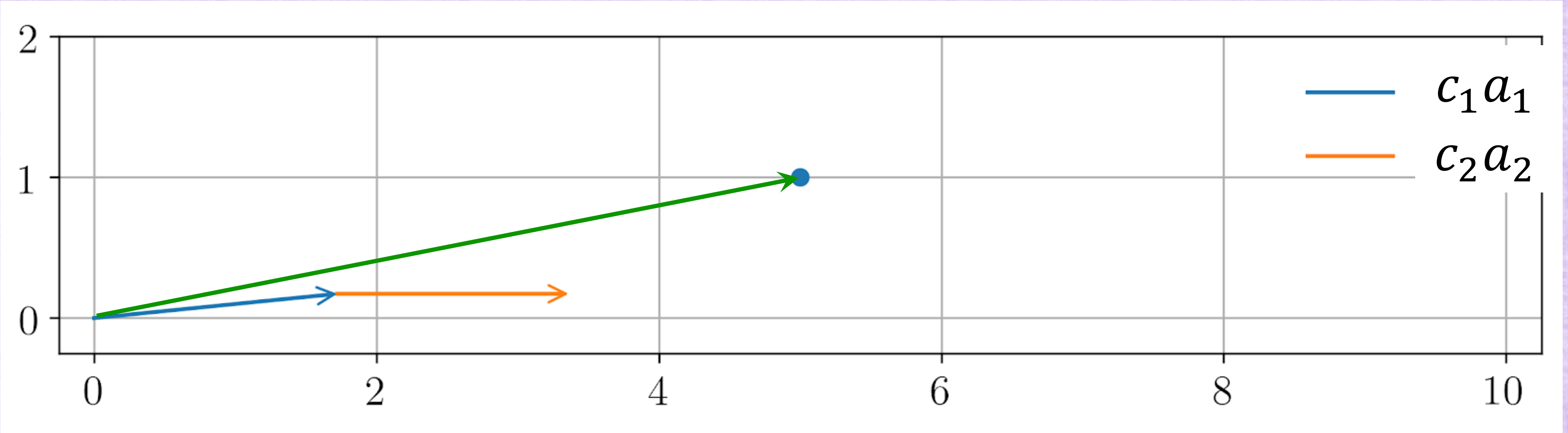
Overshooting

- Since a_1 and a_2 are not parallel, there is a unique solution
- However, this solution overshoots b by quite a bit, and then backtracks



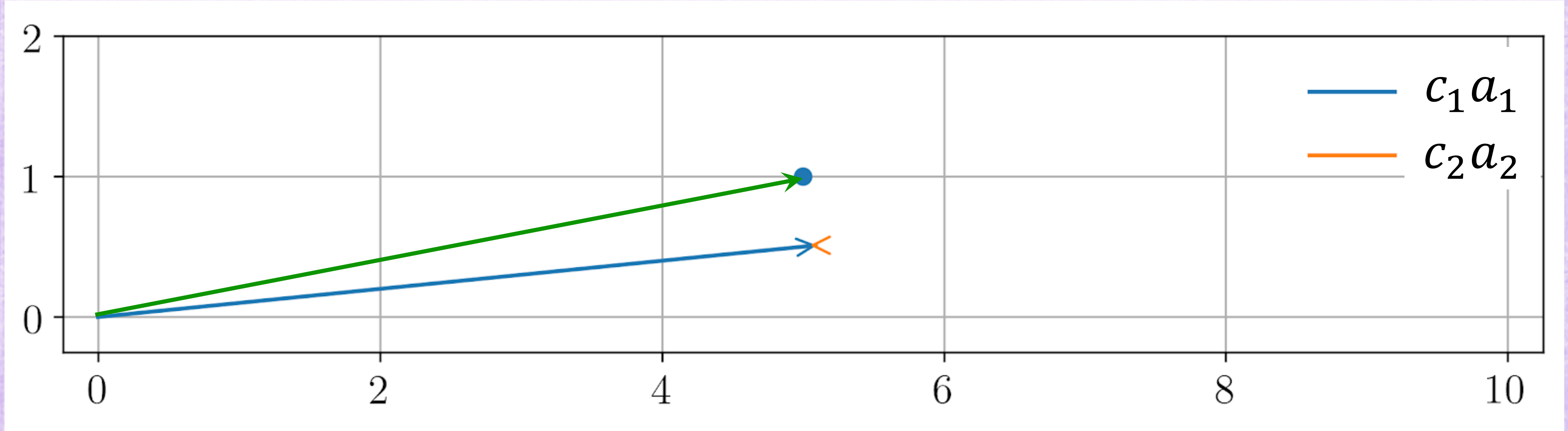
Regularization/Damping

- Adding regularization of $Ic = 0$ damps both components of the solution



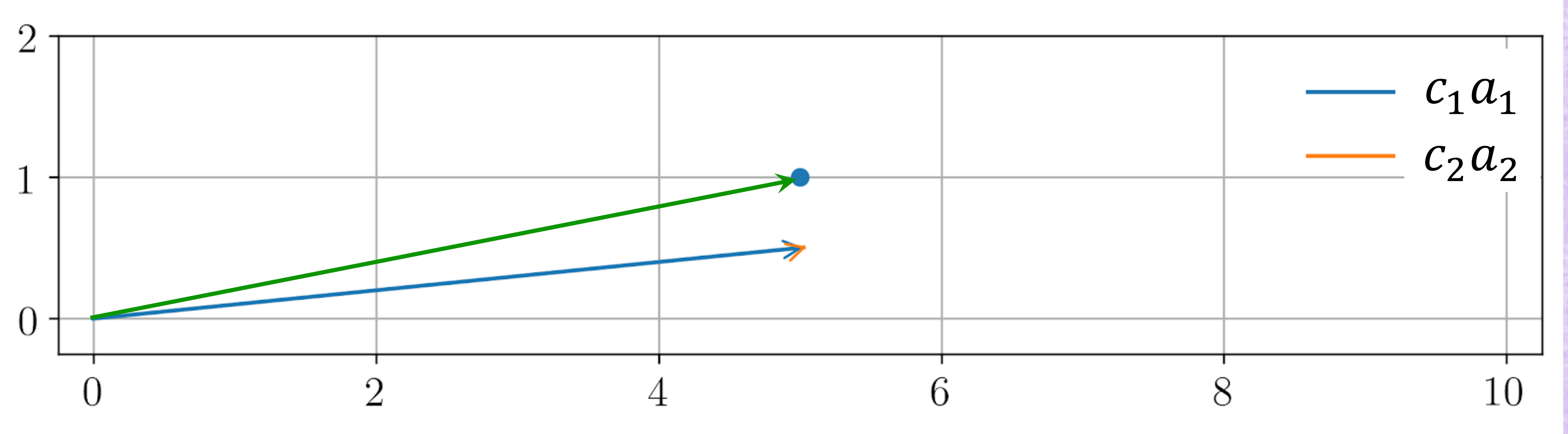
Smarter Regularization

- Adding regularization of $\begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} c = 0$ only damps c_2 and allows $c_1 a_1$ to estimate b unhindered



Coordinate Descent

- Coordinate Descent looks at one vector at a time
- After making good progress with a_1, there is little advantage to using a_2



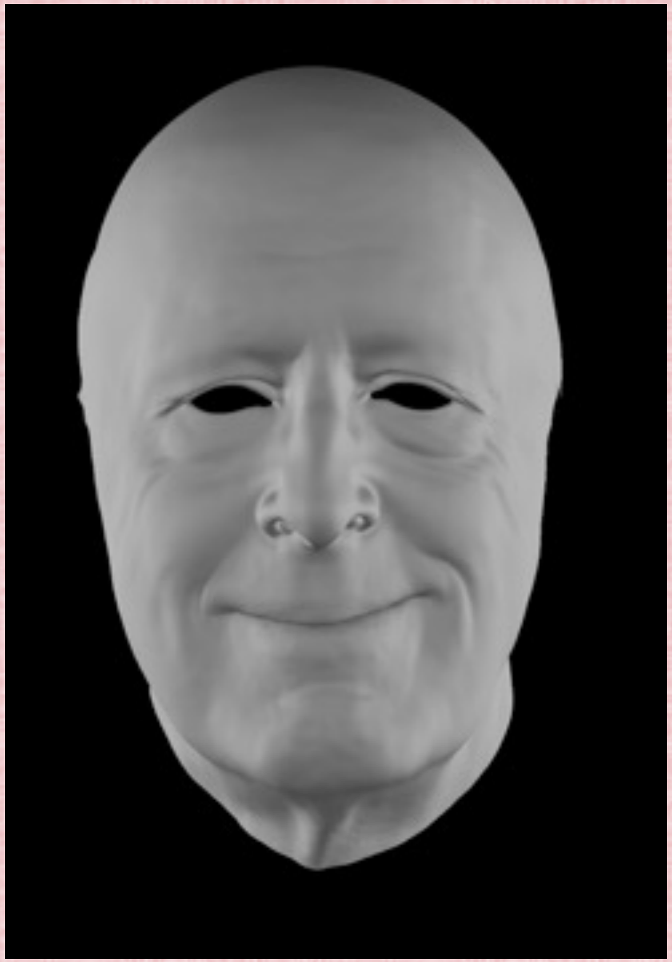
Geometric Approaches

- Thinking geometrically avoids issues with the rank of A
- Other concerns may be more important:
  - Use as few columns as possible - Setting many c_k to zero gives a sparser solution (which is easier to glean semantic information from)
  - Correlation - Columns more parallel to b may be more relevant than those that are more perpendicular
  - Gains - Columns that have a large dot product with b's direction make more progress towards b with smaller c_k values (more minimal solution norm)

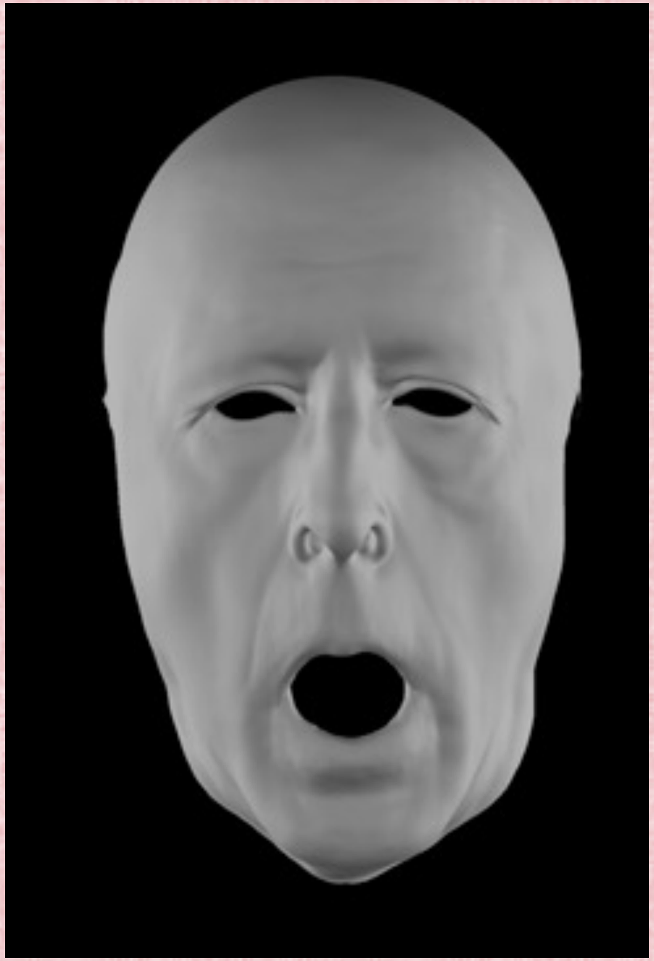
Correlation vs. Gains

- Consider $a_k \cdot b = \|a_k\|_2 \|b\|_2 \cos\Theta$ where Θ measures how parallel a_k and b are
- Correlation preference uses the columns a_k with a larger $\cos\Theta$, i.e. columns that point more closely in the same direction as b
- When the c_k represent actions, the goal of minimizing action (gains) leads to a preference for smaller c_k
  - similar in spirit to $Ic = 0$ or minimum norm solutions
- Then, columns that make more progress in the direction of b are preferable
- Progress in the direction of b is measured via $a_k \cdot \frac{b}{\|b\|_2}$ or $\|a_k\|_2 \cos\Theta$

Facial Animation



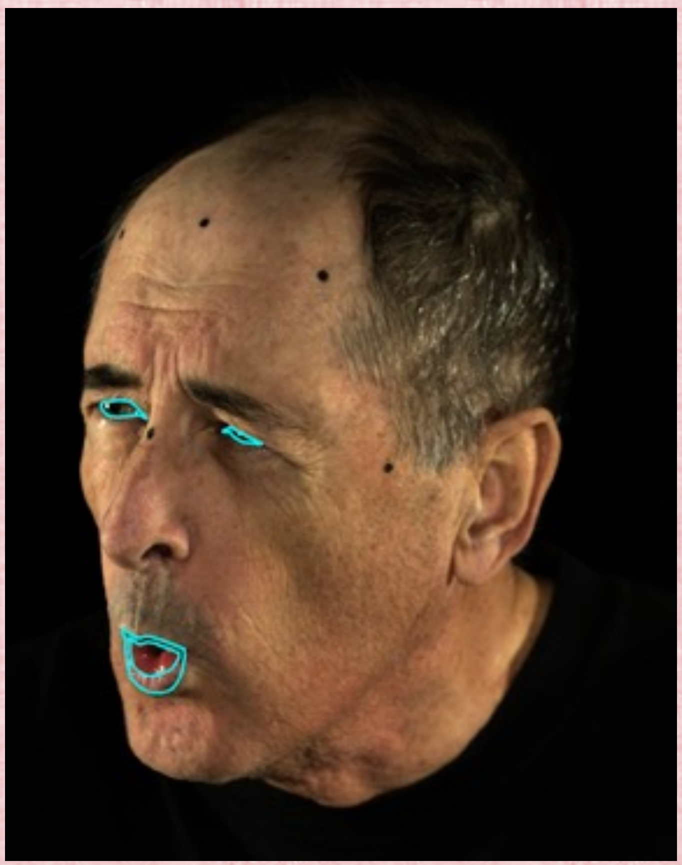
$\varphi(\theta_1)$



$\varphi(\theta_2)$

- Create a procedural skinning model of a face, where (input) animation parameters θ lead to a 3D position (output) for every vertex of the face mesh $\varphi(\theta)$
- E.g. in blend shape systems, each component of θ corresponds to a different expression (or sub-expression), and setting multiple components to be nonzero mixes expressions

Facial Tracking

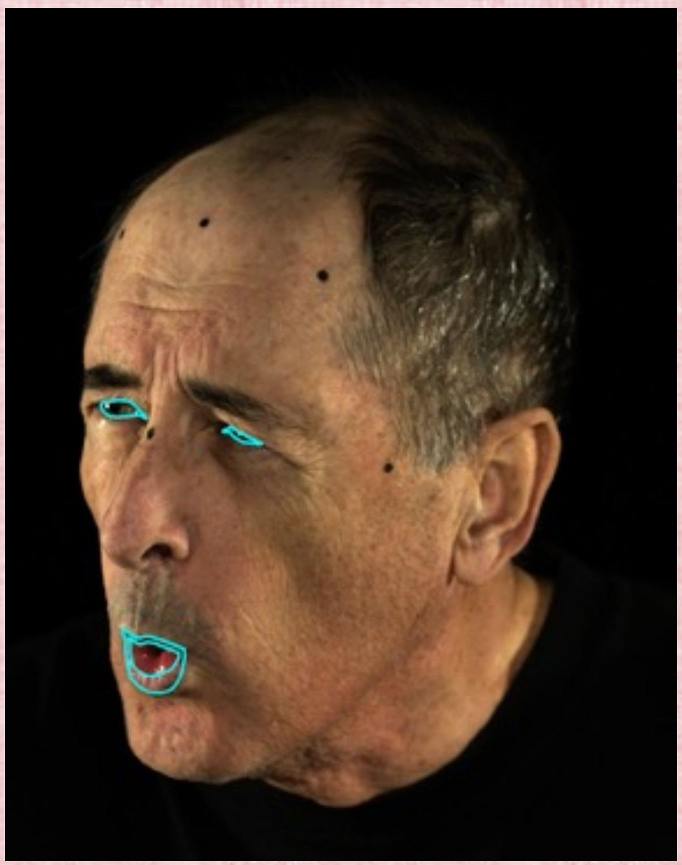


2D RGB Image

3D model

- On the 3D model, embed (red) curves around the eyes/mouth that move with the 3D surface as it deforms
- Draw similar (blue) curves on a 2D RGB image of the actual face
- Goal: projection of the red curves (onto the image plane) should overlap the blue curves (giving an estimate of θ for the 2D RGB image)

Facial Tracking



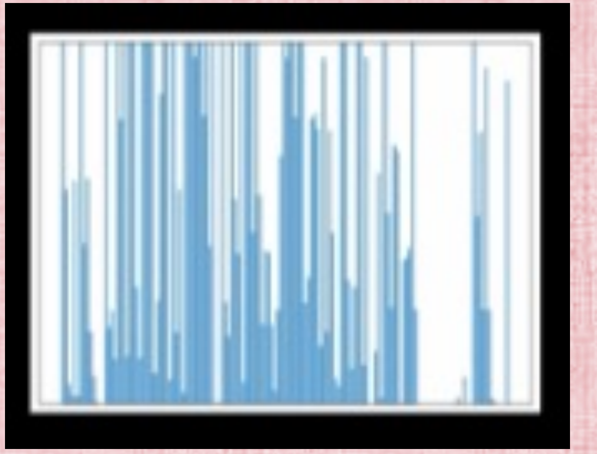
2D RGB Image

3D model

- The blue curves are data C^*
- The projection of the red curves C is a function of the 3D geometry φ, which in turn is a function of the animation parameters θ, i.e. $C\big(\varphi(\theta)\big)$
- Determine θ that minimizes the difference $\|C\big(\varphi(\theta)\big) - C^*\|$ between the curves

Solving for the Animation Parameters

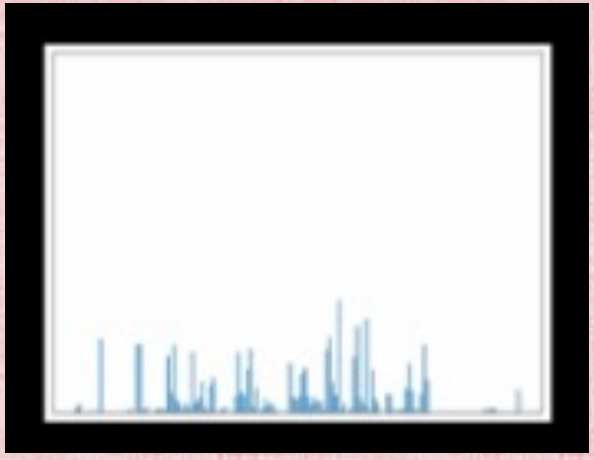
- This nonlinear problem can be solved via optimization
- At every step of optimization, the problem is <u>linearized</u>
- Solving the resulting linear system $Ac = b$ gives a search direction, which is used to make progress towards the solution



- The optimization performs poorly without regularization
- The resulting θ values are wild and arbitrary (as seen in the figure)
- The curves provide too little data for the optimization to work well

L2 Regularization

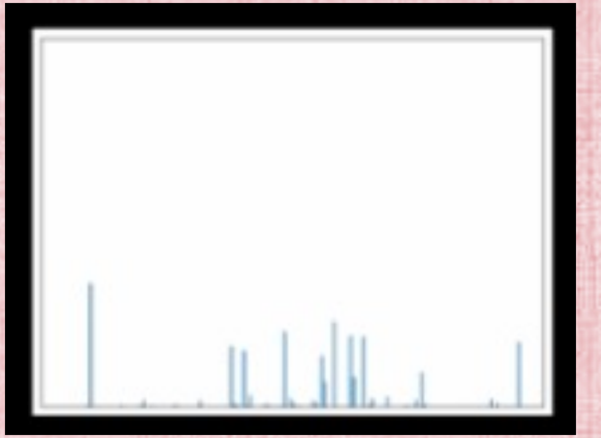
- Adding $Ic = 0$ to the linearized problem at every iteration has the expected result:
  - The regularized problem is much more solvable, and the results are less noisy
- However, θ is overly damped (as seen in the figure)



- Also, a large number of animation parameters θ are nonzero, even for this is relatively simple expression
- This hinders the interpretability (semantics) of θ

"Soft L1" Regularization

- There are many options for regularization
- In particular, "soft L1" typically produces a <u>sparser</u> set of solution parameters than L2 regularization (see figure)
- A sparser solution allows one to better ascertain semantic meaning from the animation parameters θ



- But, θ is still overly damped

A Geometric Approach (Column Space Search)

- The column space search gives a <u>sparse</u> set of solution parameters with significantly <u>less damping</u>
- This allows one to better ascertain semantic meaning from the animation parameters θ