Lecture 13: Gradient Descent Optimization

Chapter 5.3

Charles A. Bouman

Purdue University

Wednesday, September 23, 2026

Outline

1. **Gradient descent for MAP restoration**
   The update, and its convolutional form.
2. **Why the step size is hard to choose**
   Eigenvalues, the condition number, and spatial frequency.
3. **Line search**
   Choosing the step size automatically at each iteration.

Reading: Chapter 5.3, *Foundations of Computational Imaging*.

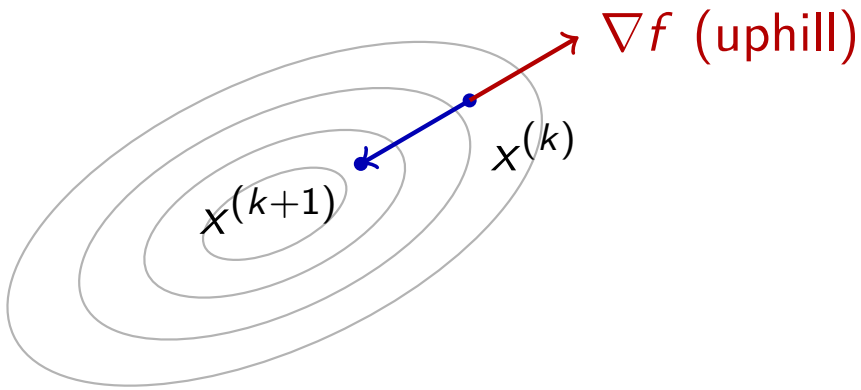
Today's Big Ideas

- No matrix inverse: each step applies A, A^t, and the prior.
- Step size is hard: a large condition number leaves no good choice.
- For deblurring, high frequencies are slowest, so edges come last.
- Line search picks each step by minimizing the cost.

Gradient descent is the first of four optimization methods in Chapter 5. The next lecture covers iterative coordinate descent and conjugate gradient.

Gradient descent

Start at an initial value $x^{(0)}$ and move in the direction opposite the gradient. The gradient points in the direction of the function's most rapid increase.



The general form of **gradient descent** iterations is

$$x^{(k+1)} = x^{(k)} - \beta \nabla f\big(x^{(k)}\big) \ ,$$

where $x^{(k)}$ is the current image estimate, $x^{(k+1)}$ is the updated image, and β is a positive **step size** that sets how far one moves along the negative gradient direction.

The gradient descent update

For the image restoration problem, the gradient of $f(x)$ is $\nabla f(x) = -\frac{1}{\sigma^2} A^t(y - Ax) + Bx$, so the update becomes

$$x^{(k+1)} = x^{(k)} + \alpha \left[\underbrace{A^t(y - Ax^{(k)})}_{\text{correct measurement mismatch}} - \underbrace{\sigma^2 B x^{(k)}}_{\text{enforce image prior}} \right],$$

where α is a scaled step size. Rearranging terms yields

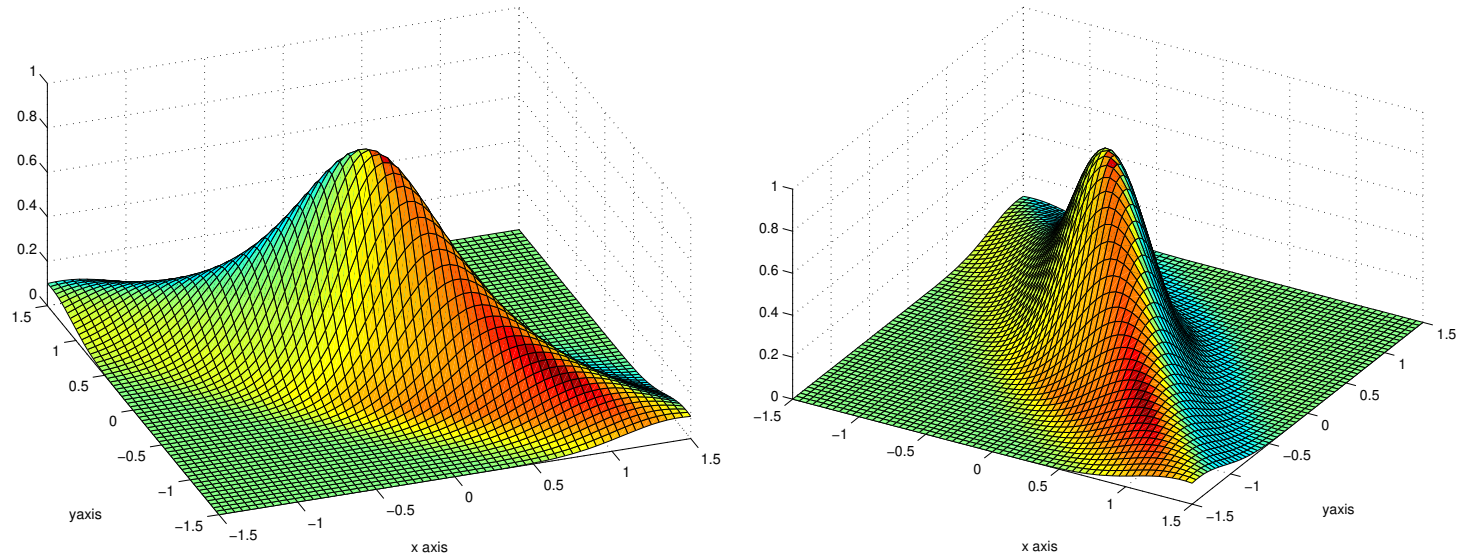
$$\begin{aligned} x^{(k+1)} &= \left(I - \alpha \left[A^t A + \sigma^2 B\right]\right) x^{(k)} + \alpha A^t y \\ &= \left(I - \alpha H\right) x^{(k)} + \alpha A^t y \ , \end{aligned} \tag{1}$$

where $H = A^t A + \sigma^2 B$.

The difficulty: choosing the step size

Gradient descent has a key disadvantage: selecting the step-size parameter. Consider minimizing

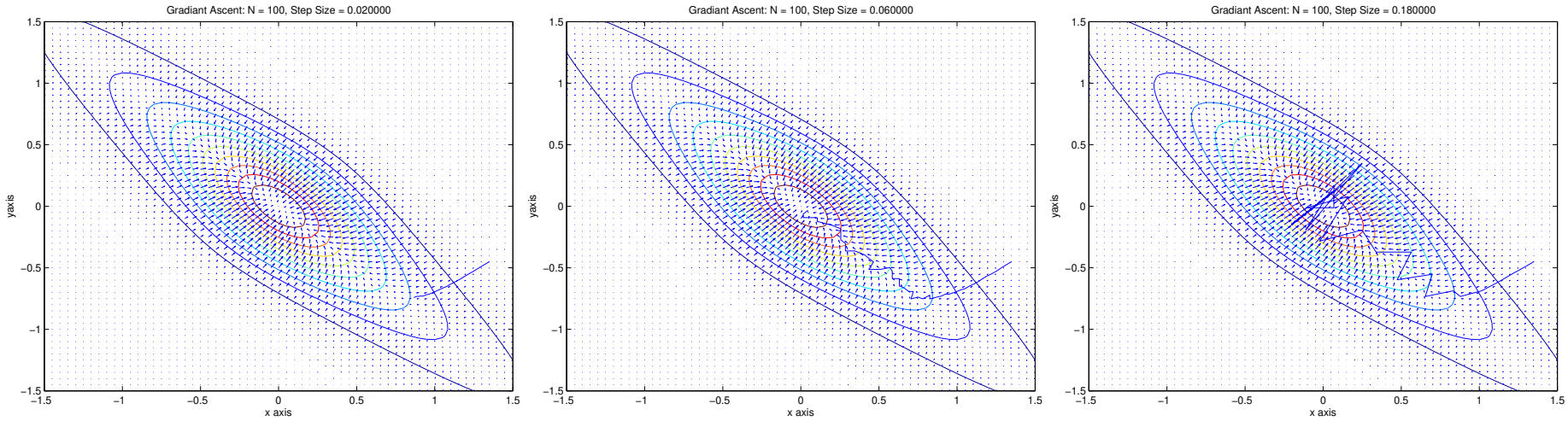
$$c(x, y) = -\frac{1}{(x-y)^2+1} \exp\left(-4(x+y)^2\right) \ .$$



- The function has a **sharp peak** and is **flat** far from it.
- Maximizing $-c(x, y)$ is equivalent to minimizing $c(x, y)$.

The step-size dilemma

Trajectories of gradient descent for three different step sizes:



(a) $\alpha = 0.02$ (b) $\alpha = 0.06$ (c) $\alpha = 0.18$

- In (a) the step size is too **small**, and convergence is **slow**.
- In (c) the step size is too **large**, and convergence is **unstable**.

Unfortunately, for high-dimensional optimization problems such as MAP image restoration, we will see that **there is often no good choice of α.**

Convergence analysis

Assume the iteration converges to the limit $x^{(\infty)}$, and define the error

$$\epsilon^{(k)} = x^{(k)} - x^{(\infty)} \ .$$

Taking the limit of the recursion (1) gives

$$x^{(\infty)} = (I - \alpha H)\, x^{(\infty)} + \alpha A^t y \ ,$$

where $H = A^t A + \sigma^2 B$.

Subtracting this equation from (1) results in

$$\epsilon^{(k+1)} = (I - \alpha H)\, \epsilon^{(k)} \ .$$

- So the error obeys a **homogeneous** linear recursion.

Convergence of eigen-modes of H

Since $H = A^t A + \sigma^2 B$ is symmetric, we have that

$$\begin{aligned}\tilde{\epsilon}^{(k+1)} &= E^t \left(I - \alpha H\right) \epsilon^{(k)} \\ &= \left(I - \alpha E^t H E\right) E^t \epsilon^{(k)} \\ &= \left(I - \alpha \Lambda\right) \tilde{\epsilon}^{(k)} \ ,\end{aligned}$$

where $H = E \Lambda E^t$ is the eigen decomposition of H with $E^t E = I$, $\Lambda = \text{diag}\{h_1, \ldots, h_N\}$, and $\tilde{\epsilon}^{(k)} = E^t \epsilon^{(k)}$ is the transformed error.

Since Λ is diagonal, we have that:

$$\begin{aligned}\tilde{\epsilon}^{(k)} &= \left(I - \alpha \Lambda\right)^k \tilde{\epsilon}^{(0)} \\ \tilde{\epsilon}_i^{(k)} &= \left(1 - \alpha h_i\right)^k \tilde{\epsilon}_i^{(0)} \ .\end{aligned}$$

Stability and speed of gradient descent

The per-mode decay $\tilde{\epsilon}_i^{(k)} = (1 - \alpha h_i)^k \, \tilde{\epsilon}_i^{(0)}$ places two requirements on the step size α.

- Convergence is **stable** only when $|1 - \alpha h_i| < 1$ for every mode i.
- Convergence is **fast** only when $1 - \alpha h_i \approx 0$.

If $\alpha = 1/h_{max}$ and $\kappa(H) = \frac{h_{max}}{h_{min}}$ is the **condition number**, then the error decays as

$$\left(1 - \frac{1}{h_{max}} h_{min}\right)^k = \left(1 - \frac{1}{\kappa(H)}\right)^k .$$

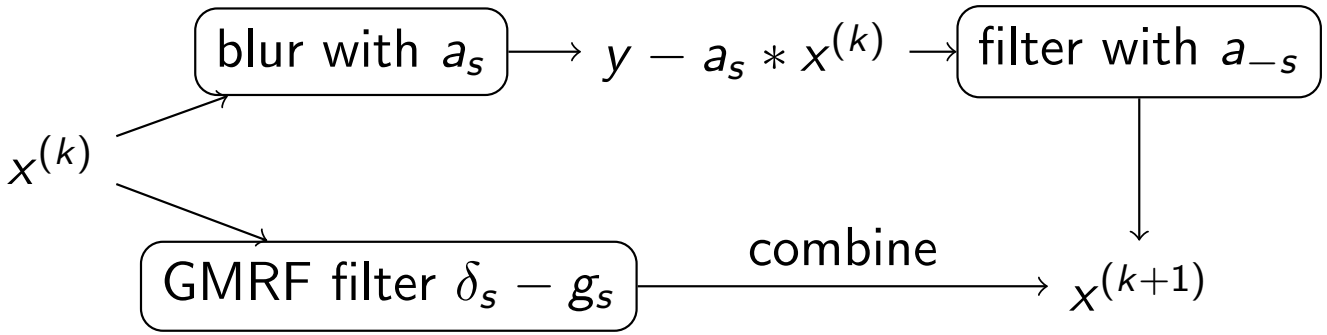
- Larger **condition number** $\Rightarrow$ slower convergence.
- Condition numbers for ill posed problems can easily $> 10^4$.

When A is an LSI filter

When A is an LSI filter with impulse response a_s, the product Ax is the convolution $a_s * x_s$, and $A^t y$ is the convolution with the space-reversed response, $a_{-s} * y_s$. The update becomes

$$x_s^{(k+1)} = x_s^{(k)} + \alpha \left[a_{-s} * (y_s - a_s * x_s^{(k)}) - \lambda(\delta_s - g_s) * x_s^{(k)} \right] , \qquad (2)$$

where g_s is the GMRF predictor and $\lambda = \sigma^2/\sigma_x^2$ is the **noise-to-signal ratio**. The prior term $\sigma^2 B$ becomes the convolution $\lambda(\delta_s - g_s)*$, because $B = (I - G)/\sigma_x^2$.



The computation is not excessive as long as the impulse response is spatially limited.

Frequency analysis of gradient descent

Taking the DSFT of the LSI update (2) results in

$$x^{(k+1)}(\omega) = \left(1 - \alpha \left[|a(\omega)|^2 + \lambda(1 - g(\omega))\right]\right) x^{(k)}(\omega) + \alpha a^*(\omega) y(\omega) \ .$$

Defining $\epsilon^{(k)}(\omega)$ to be the DSFT of $\epsilon_s^{(k)} = x_s^{(k)} - x_s^{(\infty)}$, we can derive a recursion for the error in the computation of the MAP estimate:

$$\epsilon^{(k+1)}(\omega) = \left(1 - \alpha \left[|a(\omega)|^2 + \lambda(1 - g(\omega))\right]\right) \epsilon^{(k)}(\omega) \ .$$

The values of $h(\omega)$ are the eigenvalues of H

Using the definition

$$h(\omega) \stackrel{\triangle}{=} |a(\omega)|^2 + \lambda(1 - g(\omega)) \ ,$$

we have that

$$\epsilon^{(k)}(\omega) = (1 - \alpha h(\omega))^k \, \epsilon^{(0)}(\omega) \ .$$

- So the values $h(\omega)$ are the **eigenvalues of** H.
- Furthermore, we know that $h(\omega) > 0$ because $1 - g(\omega) > 0$.

Define $h_{max} = \max_\omega h(\omega)$ and $h_{min} = \min_\omega h(\omega)$. Selecting $\alpha = 1/h_{max}$ for fast convergence of the largest eigenvalue then results in

$$\epsilon^{(k)}(\omega) = \left(1 - \frac{h(\omega)}{h_{max}}\right)^k \epsilon^{(0)}(\omega) \ .$$

Example 5.1: a 1-D deconvolution problem

Consider a 1-D signal deconvolution problem of the form

$$Y_n = \sum_{k=-1}^{1} a_k X_{n-k} + W_n \ ,$$

where $a_n = \delta_n + \frac{1}{2}\left(\delta_{n-1} + \delta_{n+1}\right)$ is a smoothing filter and W_n is additive noise modeled as i.i.d. Gaussian random variables with $N(0, \sigma^2)$ distribution.

We use a prior model for X of a GMRF with noncausal prediction variance σ_x^2 and noncausal prediction filter

$$g_n = \frac{1}{2}\left(\delta_{n-1} + \delta_{n+1}\right) \ .$$

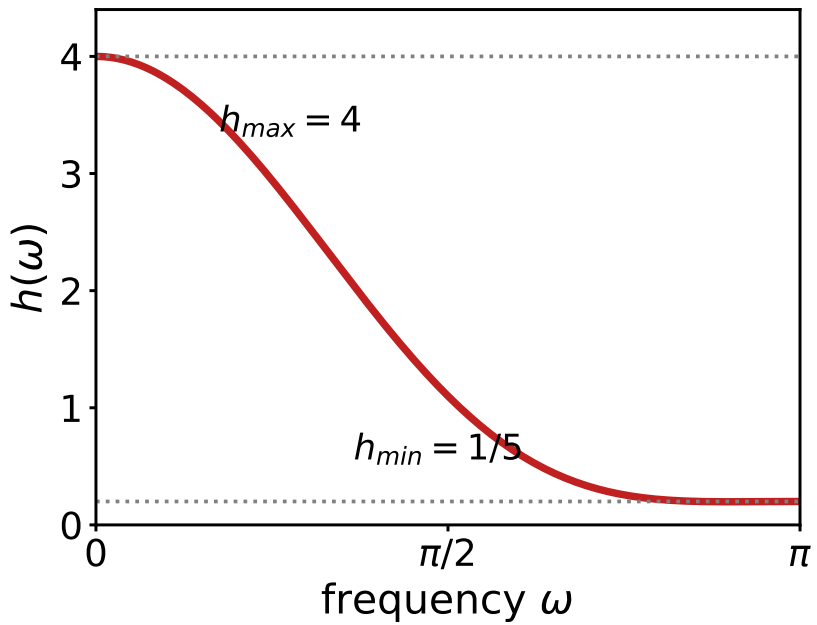
Taking the DTFT of a_n and g_n results in the expressions

$$\begin{aligned} a(\omega) &= 1 + \cos(\omega) \ , \\ 1 - g(\omega) &= 1 - \cos(\omega) \ . \end{aligned}$$

Example 5.1: computing $h(\omega)$

If we assume that $\sigma^2 = 1$ and $\sigma_x^2 = 10$, then $\lambda = \sigma^2/\sigma_x^2 = 1/10$, and we can calculate the following explicit expression for $h(\omega)$:

$$h(\omega) = |1 + \cos(\omega)|^2 + \frac{1}{10}(1 - \cos(\omega)) \ .$$

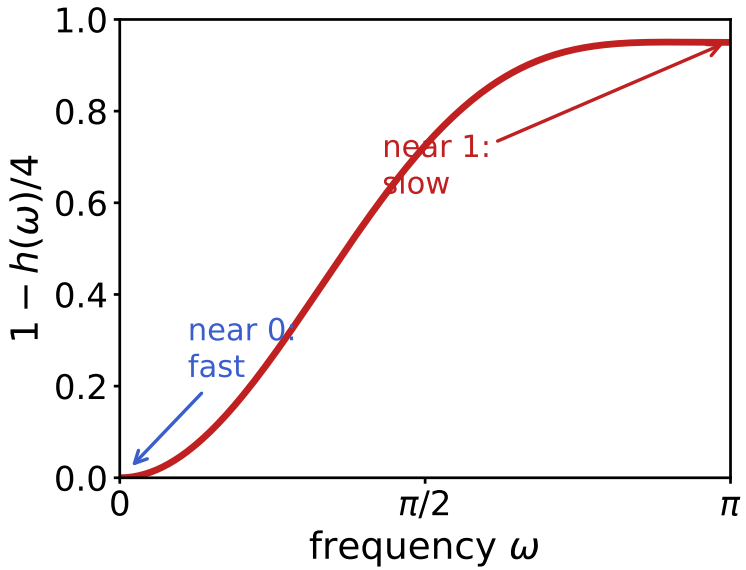


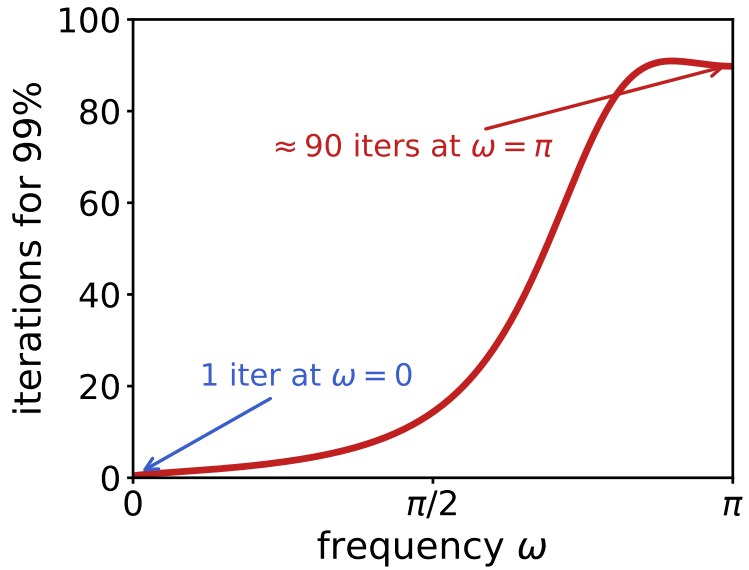
The eigenvalues span $h_{max} = 4$ down to $h_{min} = 1/5$, so $\kappa(H) = 20$.

Example 5.1: convergence rate versus frequency

Setting the step size to $\alpha = 1/h_{max} = 1/4$, the convergence at frequency ω is given by

$$\epsilon^{(k)}(\omega) = \left(1 - \frac{h(\omega)}{4}\right)^k \epsilon^{(0)}(\omega) \ .$$

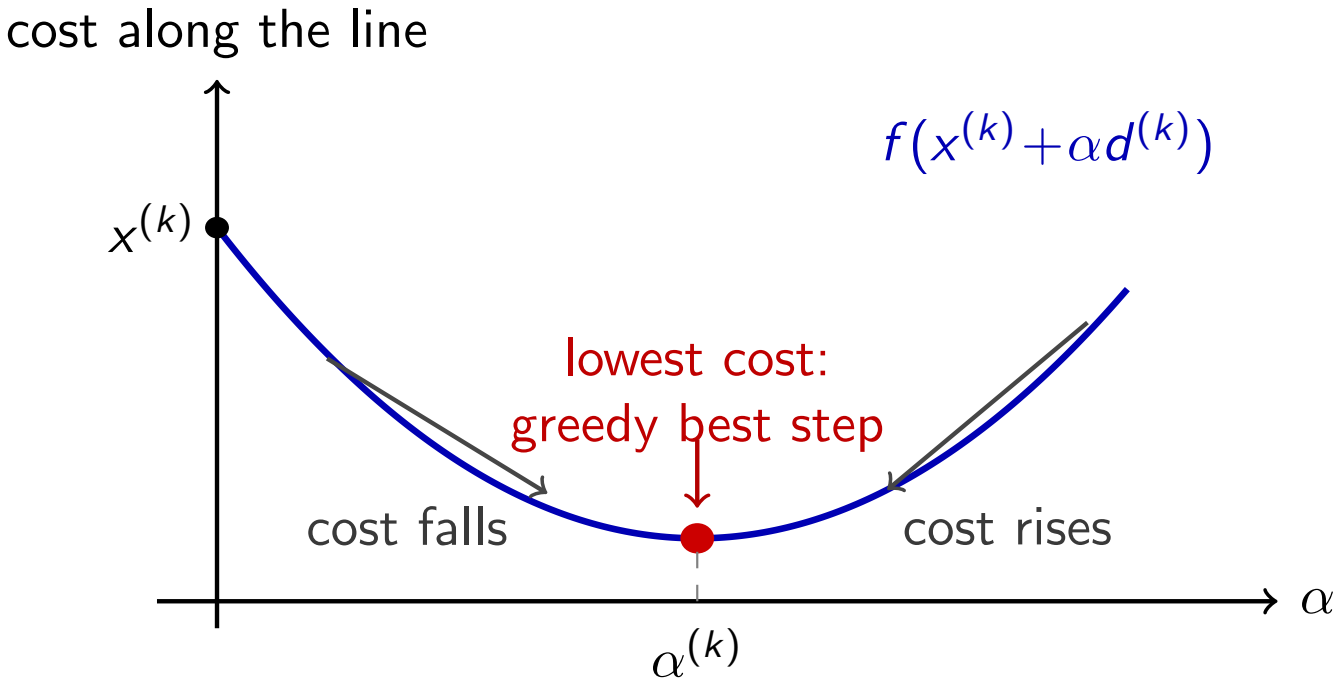




High frequencies converge slowest: one iteration at $\omega = 0$ but about 90 at $\omega = \pi$.

Line search

Line search picks the best step-size for each iteration. It moves along the downhill direction $d^{(k)} = -\sigma^2 \nabla f(x^{(k)})$ and stops where the cost along the line is lowest.



$$\alpha^{(k)} = \arg\min_{\alpha \geq 0} f\left(x^{(k)} + \alpha d^{(k)}\right) , \qquad x^{(k+1)} = x^{(k)} + \alpha^{(k)} d^{(k)} .$$

For a quadratic cost, the best step has closed form

Consider the quadratic MAP cost function:

$$f(x) = \frac{1}{2\sigma^2}\|y - Ax\|^2 + \frac{1}{2}x^t Bx \ .$$

The cost along the line, $g(\alpha) = f(x^{(k)} + \alpha d^{(k)})$, is a parabola, so its minimum solves $g'(\alpha) = 0$, which gives

$$\alpha^{(k)} = \frac{(y - Ax^{(k)})^t A\, d^{(k)} - \sigma^2 \left[x^{(k)}\right]^t B\, d^{(k)}}{\|d^{(k)}\|_H^2} \ ,$$

where $\|z\|_H^2 = z^t Hz$ and $H = A^t A + \sigma^2 B$.

- This holds for **any** direction $d^{(k)}$.
- Conjugate gradient changes the direction; this general form is needed.

For the gradient direction, the step is a single ratio

For the gradient direction,

$$d^{(k)} = -\sigma^2 \nabla f(x^{(k)}) = A^t \left(y - Ax^{(k)} \right) - \sigma^2 B x^{(k)} \ ,$$

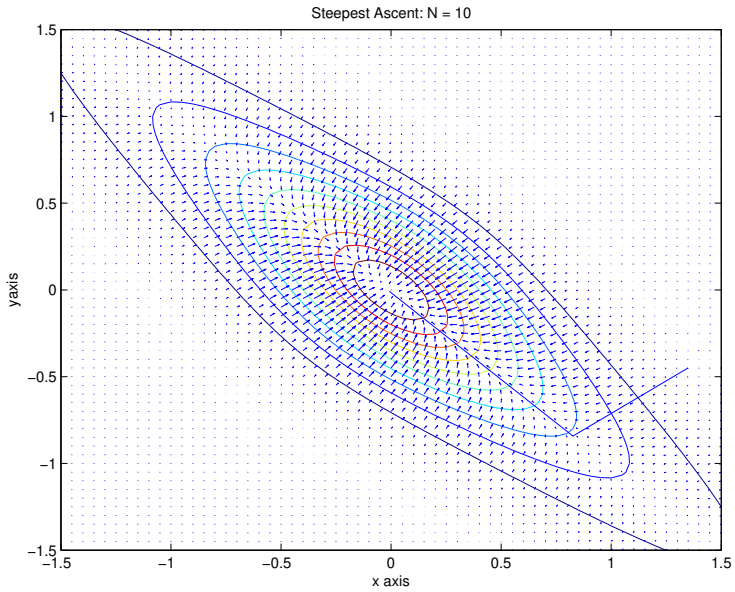
the numerator above becomes $\|d^{(k)}\|^2$, so the step size reduces to

$$\alpha^{(k)} = \frac{\|d^{(k)}\|^2}{\|d^{(k)}\|_H^2} \ .$$

- This is **steepest descent**: gradient direction, exact line search.
- The ratio requires this specific scaling of the gradient.

Line search is nearly free, but the valley persists

- $\|d^{(k)}\|_H^2$ costs one extra H product: **nearly free**.
- No step to tune; the cost never increases: $f(x^{(k+1)}) \leq f(x^{(k)})$.



Line search takes the best step each time, yet still zig-zags across the narrow valley. The condition-number problem persists, and the next lecture addresses it.

Summary

- Gradient descent minimizes $f(x) = \frac{1}{2\sigma^2}\|y - Ax\|^2 + \frac{1}{2}x^t Bx$ without inverting $H = A^t A + \sigma^2 B$. Each step is $x^{(k+1)} = x^{(k)} + \alpha[A^t(y - Ax^{(k)}) - \sigma^2 Bx^{(k)}]$, a few convolutions for an LSI blur.
- The eigenvalues h_i of H set the rate, $\tilde{\epsilon}_i^{(k)} = (1 - \alpha h_i)^k \tilde{\epsilon}_i^{(0)}$. With $\alpha = 1/h_{max}$ the slowest mode decays as $(1 - 1/\kappa(H))^k$, so a large condition number is slow.
- For an LSI blur $h(\omega) = |a(\omega)|^2 + \lambda(1 - g(\omega))$ is low pass, so the high spatial frequencies, the edges, converge last.
- Line search removes the step parameter by minimizing along the direction. For the quadratic cost the step is $\alpha^{(k)} = \|d^{(k)}\|^2 / \|d^{(k)}\|_H^2$.

Reading: Chapter 5.3. Next lecture: iterative coordinate descent and conjugate gradient.