# Lecture 19

## Iterative Methods

David Semeraro

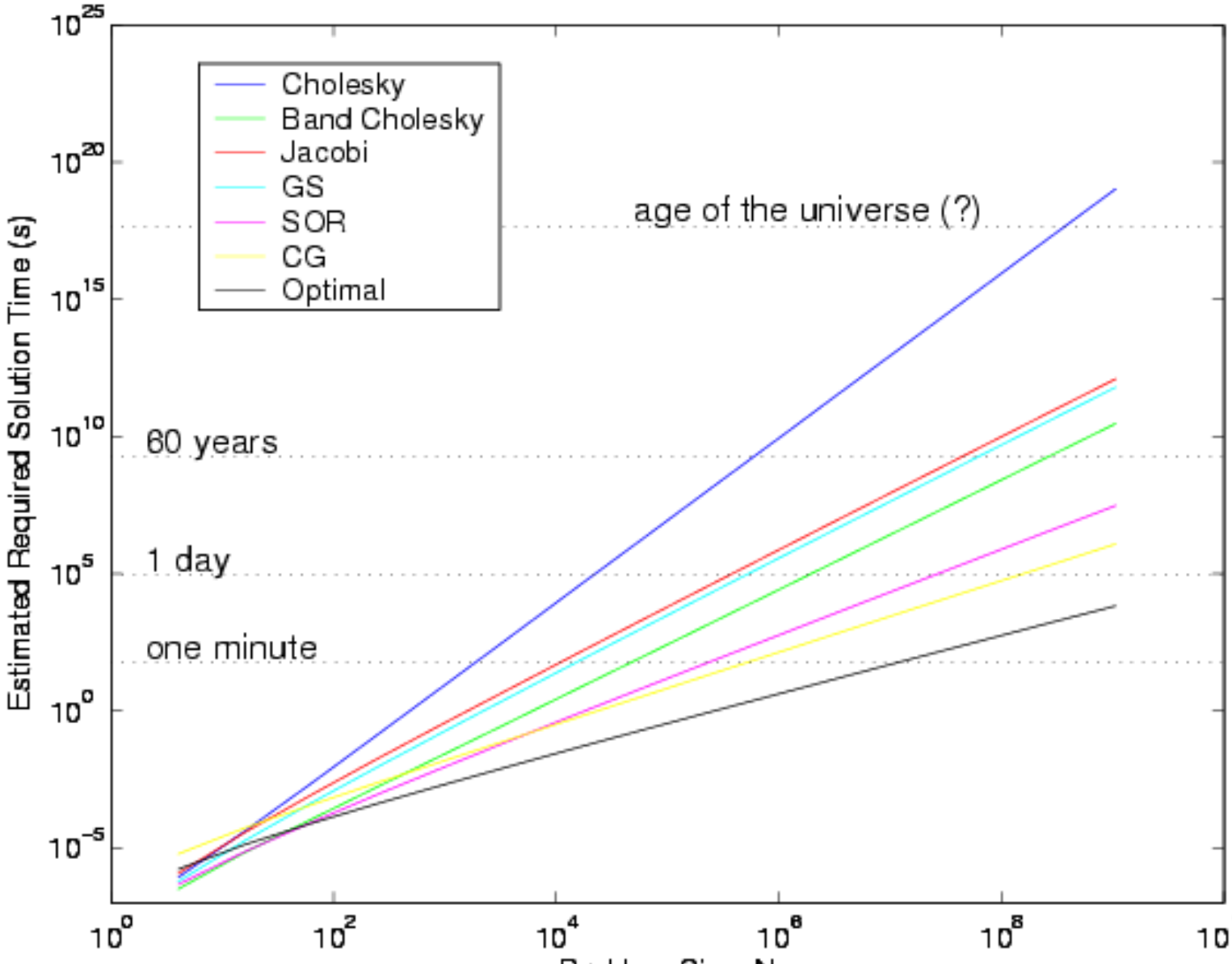
University of Illinois at Urbana-Champaign

April 6, 2010

# Summary: Complexity of Linear Solves

- $Ax = b$
- diagonal system: $\mathcal{O}(n)$
- upper or lower triangular system: $\mathcal{O}(n^2)$
- full system with GE: $\mathcal{O}(n^3)$
- scaled partial pivoting adds $\mathcal{O}(n^2)$
- full system with LU: $\mathcal{O}(n^3)$
- LU back solve: $\mathcal{O}(n^2)$
- $m$ different right-hand sides: $\mathcal{O}(mn^3)$ or $\mathcal{O}(n^3 + mn^2)$
- tridiagonal system: $\mathcal{O}(n)$
- $m$-band system: $\mathcal{O}(m^2 n)$

# Summary: Complexity



# Approximate solutions

So far, we are seeking "exact" solutions $x^*$ to

$$Ax = b$$

What if we only need an approximations $\hat{x}$ to $x^*$?

We would like some $\hat{x}$ so that $\|\hat{x} - x^*\| \leqslant \epsilon$, where $\epsilon$ is some tolerance.

# The Residual

We can't actually evaluate

$$e = x^* - \hat{x}$$

But...

For $x = x^*$

$$b - Ax \equiv 0$$

For $x = \hat{x}$

$$b - Ax \not\equiv 0$$

We call $\hat{r} = b - A\hat{x}$ the *residual*. It is way to measure the error. In fact

$$\begin{aligned} \hat{r} &= b - A\hat{x} \\ &= Ax^* - A\hat{x} \\ &= A\hat{e} \end{aligned}$$

# How big is the residual?

For a given approximation, $\hat{x}$ to $x$, how "big" is the residual $\hat{r} = b - A\hat{x}$?

- $\|r\|$ gives a magnitude
- $\|r\|_1 = \sum_{j=1}^{n} |r_i|$
- $\|r\|_2 = \left( \sum_{j=1}^{n} r_i^2 \right)^{1/2}$
- $\|r\|_\infty = \max_{1 \leqslant j \leqslant n} |r_i|$

# Approximating $x$...

Suppose we made a wild guess to the solution $x$ of $Ax = b$:

$$x^{(0)} \approx x$$

How do I improve $x^{(0)}$?

Ideally:

$$x^{(1)} = x^{(0)} + e^{(0)}$$

but to obtain $e^{(0)}$, we must know $x$. Not a viable method.

Ideally (another way):

$$\begin{aligned} x^{(1)} &= x^{(0)} + e^{(0)} \\ &= x^{(0)} + (x^* - x^{(0)}) \\ &= x^{(0)} + (A^{-1}b - x^{(0)}) \\ &= x^{(0)} + A^{-1}(b - Ax^{(0)}) \\ &= x^{(0)} + A^{-1}r^{(0)} \end{aligned}$$

# An iteration

Again, the method

$$x^{(1)} = x^{(0)} + A^{-1}r^{(0)}$$

is nonsense since $A^{-1}$ is needed.

What if we approximate $A^{-1}$? Suppose $Q^{-1} \approx A^{-1}$ and is cheap to construct, then

$$x^{(1)} = x^{(0)} + Q^{-1}r^{(0)}$$

is a good step.

continuing...

$$x^{(k)} = x^{(k-1)} + Q^{-1}r^{(k-1)}$$

# What kind of $Q^{-1}$ do I need?

Rewrite:

$$x^{(k)} = x^{(k-1)} + Q^{-1}(b - Ax^{(k-1)})$$

This becomes

$$\begin{aligned} Qx^{(k)} &= Qx^{(k-1)} + (b - Ax^{(k-1)}) \\ &= (Q - A)x^{(k-1)} + b \end{aligned}$$

This is the form in the text (page 322 NMC6).

Or

$$x^{(k)} = Q^{-1}(Q - A)x^{(k-1)} + Q^{-1}b$$

# Two Popular Choices

## Example

Jacobi iteration approximates $A$ with $Q = diag(A)$.

```
1  x = x^(0)
2
3  Q = D
4
5  for k = 1 to k_max
6     r = b − Ax
7     if ||r = b − Ax|| ⩽ tol, stop
8
9     x = x + Q^(−1) r
10 end
```

# Two Popular Choices

## Example

Gauss-Seidel iteration approximates $A$ with $Q = lowertri(A)$.

```
1  x = x^(0)
2
3  Q = D - L
4
5  for k = 1 to k_max
6     r = b - Ax
7     if ||r = b - Ax|| ⩽ tol, stop
8
9     x = x + Q^(-1) r
10 end
```

# Why $D$ and $D-L$?

Look again at the iteration

$$x^{(k)} = x^{(k-1)} + Q^{-1}r^{(k-1)}$$

Looking at the error:

$$x - x^{(k)} = x - x^{(k-1)} - Q^{-1}r^{(k-1)}$$

Gives

$$e^{(k)} = e^{(k-1)} - Q^{-1}Ae^{(k-1)}$$

or

$$e^{(k)} = (I - Q^{-1}A)e^{(k-1)}$$

or

$$e^{(k)} = (I - Q^{-1}A)^k e^{(0)}$$

# Why $D$ and $D - L$?

We want

$$e^{(k)} = (I - Q^{-1}A)^k e^{(0)}$$

to converge.

When does $a_k = c^k$ converge? .....when $|c| < 1$

Likewise, our iteration converges

$$\begin{aligned} \|e^{(k)}\| &= \|(I - Q^{-1}A)^k e^{(0)}\| \\ &\leqslant \|I - Q^{-1}A\|^k \|e^{(0)}\| \end{aligned}$$

when $\|I - Q^{-1}A\| < 1$.

# Matrix Norms

What is $\|I - Q^{-1}A\|$ ?

- $\|A\|_1 = \max_{1 \leqslant j \leqslant n} \sum_{i=1}^{n} |a_{ij}|$
- $\|A\|_2 = \sqrt{\rho(A^T A)}$
- $\rho(A) = \max_{1 \leqslant j \leqslant n} |\lambda_i|$
- $\|A\|_2 = \rho(A)$ for symmetric $A$
- $\|A\|_\infty = \max_{1 \leqslant i \leqslant n} \sum_{j=1}^{n} |a_{ij}|$

## Spectral Radius Theorem

In order that the sequence generated by $Qx^k = (Q - A)x^{k-1} + b$ to converge, no matter what the starting point $x^0$ is selected, it is necessary and sufficient that all eigenvalues of $I - Q^{-1}A$ lie in the open unit disc, $|Z| < 1$, in the complex plane.

For any $nxn$ matrix $A$ having eigenvalues $\lambda_i$ the spectral radius of $A$, is given by $\rho(A) = max_{1 \leqslant i \leqslant n} |\lambda_i|$.
The Sprctral Radius Theorem says in order to converge the spectral radius of the iteration matrix must be less than 1. Or the absolute value of the largest eigenvalue of the iteration matrix must be less than 1.

# Again, why do Jacobi and Gauss-Seidel work?

## Jacobi, Gauss-Seidel (sufficient) Convergence Theorem

If $A$ is diagonally dominant, then the Jacobi and Gauss-Seidel methods converge for any initial guess $x^{(0)}$.

## Definition: Diagonal Dominance

A matrix is *diagonally dominant* if

$$|a_{ii}| > \sum_{j=1, j\neq i}^{n} |a_{ij}|$$

for all $i$.

# Smart Jacobi Algorithm

The algorithm above uses the matrix representation:

$$x^{(k)} = D^{-1}(L+U)x^{(k-1)} + D^{-1}b$$

The diagonal is decoupled from the $L+U$, so we have an update in the form of

$$x_i^{(k)} = -\sum_{j=1, j\neq i}^{n} \left(\frac{a_{ij}}{a_{ii}}\right) x_j^{(k-1)} + \frac{b_i}{a_{ii}}$$

So each sweep (from $k-1$ to $k$) uses $\mathcal{O}(n)$ operations per vector element. If, for each row $i$, $a_{ij} = 0$ for all but $m$ values of $j$, each sweep uses $\mathcal{O}(mn)$ operations.

# Smart Gauss-Seidel Algorithm

The algorithm above uses the matrix representation:

$$x^{(k)} = (D - L)^{-1} U x^{(k-1)} + (D - L)^{-1} b$$

Component-wise:

$$x_i^{(k)} = -\sum_{j=1, j<i}^{n} \left( \frac{a_{ij}}{a_{ii}} \right) x_j^{(k)} - \sum_{j=1, j>i}^{n} \left( \frac{a_{ij}}{a_{ii}} \right) x_j^{(k-1)} + \frac{b_i}{a_{ii}}$$

So again each sweep (from $k-1$ to $k$) uses $\mathcal{O}(n)$ operations per vector element.
If, for each row $i$, $a_{ij} = 0$ for all but $m$ values of $j$, each sweep uses $\mathcal{O}(mn)$ operations.

The difference is that in the Jacobi method, updates are saved (and not used) in a new vector. With Gauss-Seidel, an update to an element $x_i^{(k)}$ is used immediately.

# Intuitively...

Both Jacobi and Gauss-Seidel can be viewed as a form of averaging.

## Example

Consider

$$A = \begin{bmatrix} 2 & -1 & & & & & \\ -1 & 2 & -1 & & & & \\ & -1 & 2 & -1 & & & \\ & & & \ddots & & & \\ & & & & -1 & 2 & 1 \\ & & & & & -1 & 2 \end{bmatrix} \qquad b = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \qquad x^{(0)} = rand(n, 1)$$

# Successive overrelaxation (SOR) method

Consider the Gauss-Seidel method. If we construct the next iterate for $x$ in the following way we have SOR method.

$$x^{(k)} = (D - \omega L)^{-1} \left[\omega U + (1 - \omega) D\right] x^{(k-1)} + (D - \omega L)^{-1} \omega b$$

Component-wise:

$$x_i^{(k)} = \omega \left[ - \sum_{j=1, j<i}^{n} \left( \frac{a_{ij}}{a_{ii}} \right) x_j^{(k)} - \sum_{j=1, j>i}^{n} \left( \frac{a_{ij}}{a_{ii}} \right) x_j^{(k-1)} + \frac{b_i}{a_{ii}} \right] + (1 - \omega) x_i^{k-1}$$

# Conjugate Gradients

- Suppose that $A$ is $n \times n$ symmetric and positive definite.
- Since $A$ is positive definite, $x^T Ax > 0$ for all $x \in \mathbb{R}^n$. (Why?)
- Define a quadratic function

$$\phi(x) = \frac{1}{2}x^T Ax - x^T b$$

- It turns out that $-\nabla\phi = b - Ax = r$, or $\phi(x)$ has a minimum for $x$ such that $Ax = b$.
- Optimization methods look in a "search direction" and pick the best step:

$$x_{k+1} = x_k + \alpha s_k$$

  Choose $\alpha$ so that $\phi(x_k + \alpha s_k)$ is minimized in the direction of $s_k$.
- Find $\alpha$ so that $\phi$ is minimized:

$$0 = \frac{d}{d\alpha}\phi(x_{k+1}) = \nabla\phi(x_{k+1})^T \frac{d}{d\alpha}x_{k+1} = -r_{k+1}^T \frac{d}{d\alpha}(x_k + \alpha s_k) = -r_{k+1}^T s_k.$$

# Conjugate Gradients

- Find $\alpha$ so that $\phi$ is minimized:

$$0 = \frac{d}{d\alpha}\phi(x_{k+1}) = \nabla\phi(x_{k+1})^T \frac{d}{d\alpha}x_{k+1} = -r_{k+1}^T \frac{d}{d\alpha}(x_k + \alpha s_k) = -r_{k+1}^T s_k.$$

- We also know

$$r_{k+1} = b - Ax_{k+1} = b - A(x_k + \alpha s_k) = r_k - \alpha A s_k$$

- So, the optimal search parameter is

$$\alpha = -\frac{r_k^T s_k}{s_k^T A s_k}$$

- This is CG: take a step in a search direction

# Conjugate Gradients

- Neat trick: We can compute the $r$ without explicitly forming $b - Ax$:

$$r_{k+1} = b - Ax_{k+1} = b - A(x_k + \alpha s_k) = b - Ax_k - \alpha As_k = r_k - \alpha As_k$$

# Conjugate Gradients

- How should we pick $s_k$?
- Note that $-\nabla\phi = b - Ax = r$, so $r$ is the gradient of $\phi$ (for *any* $x$), and this is a good direction.
- Thus, pick $s_0 = r = b - Ax_0$.
- What is $s_1$? This should be in the direction of $r_1$, but *conjugate* to $s_0$: $s_1^T A s_0 = 0$.
- (Two vectors $u$ and $v$ are $A$-*conjugate* is $u^T A v = 0$)
- So, if we let $s_1 = r_0 + \beta s_0$, we can require

$$0 = s_1^T A s_0 = (r_1^T + \beta s_0^T) A s_0 = r_1^T A s_0 + \beta s_0^T A s_0$$

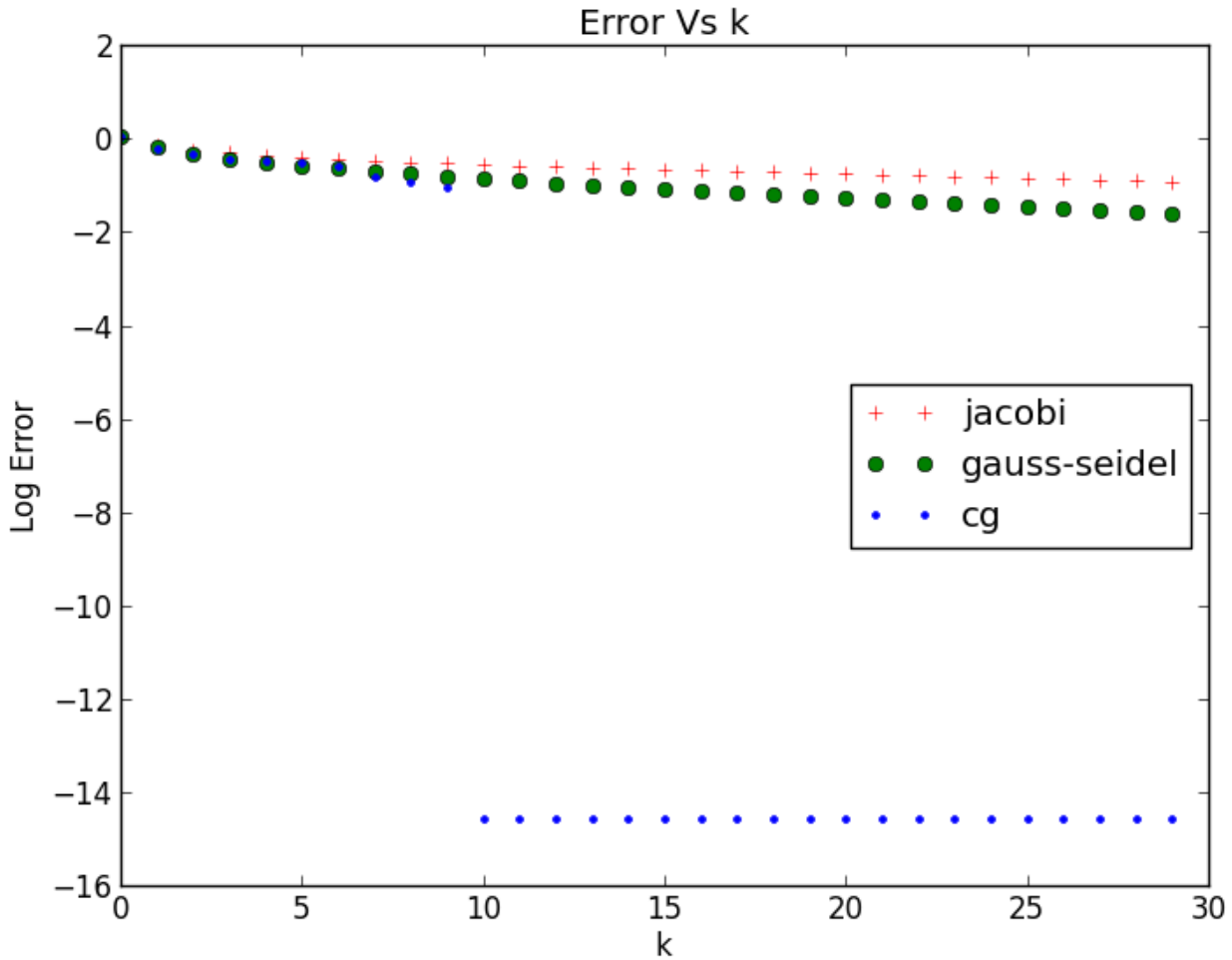
or

$$\beta = -r_1^T A s_0 / s_0^T A s_0.$$

- Holds for $s_{k+1}$ in terms of $r_k + \beta_k s_k$
- Further simplification (which is *not* simple to carry out) yields a simple method that requires only one matrix-vector product per step:

# Conjugate Gradients

1 $x_0 =$ `initial guess`
2 $r_0 = b - Ax_0$
3 $s_0 = r_0$
4 `for` $k = 0, 1, 2, \ldots$
5 $\quad \alpha_k = \frac{r_k^T r_k}{s_k^T A s_k}$
6 $\quad x_{k+1} = x_k + \alpha_k s_k$
7 $\quad r_{k+1} = r_k - \alpha_k A s_k$
8 $\quad \beta_{k+1} = r_{k+1}^T r_{k+1} / r_k^T r_k$
9 $\quad s_{k+1} = r_{k+1} + \beta_{k+1} s_k$


Error Vs k
Log Error
k
jacobi
gauss-seidel
cg
2
0
−2
−4
−6
−8
−10
−12
−14
−16
0
5
10
15
20
25
30