Scientific Computing WS 2018/2019

Lecture 7

Jürgen Fuhrmann

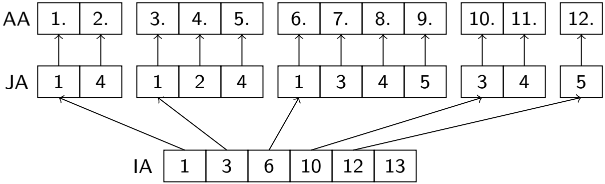
juergen.fuhrmann@wias-berlin.de

# Compressed Row Storage (CRS) format

(aka Compressed Sparse Row (CSR) or IA-JA etc.)

- real array `AA`, length nnz, containing all nonzero elements row by row
- integer array `JA`, length nnz, containing the column indices of the elements of `AA`
- integer array `IA`, length n+1, containing the start indizes of each row in the arrays `IA` and `JA` and `IA(n+1)=nnz+1`

$$A = \begin{pmatrix} 1. & 0. & 0. & 2. & 0. \\ 3. & 4. & 0. & 5. & 0. \\ 6. & 0. & 7. & 8. & 9. \\ 0. & 0. & 10. & 11. & 0. \\ 0. & 0. & 0. & 0. & 12. \end{pmatrix}$$

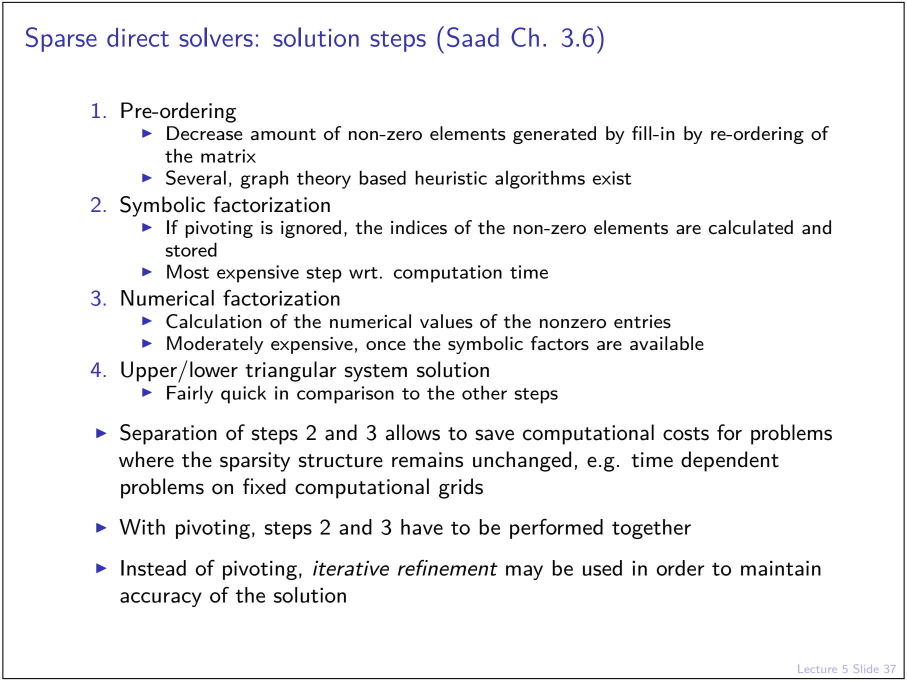


AA | 1. | 2. | | 3. | 4. | 5. | | 6. | 7. | 8. | 9. | | 10. | 11. | | 12.

JA | 1 | 4 | | 1 | 2 | 4 | | 1 | 3 | 4 | 5 | | 3 | 4 | | 5

IA | 1 | 3 | 6 | 10 | 12 | 13

- Used in most sparse matrix solver packages
- CSC (Compressed Column Storage) uses similar principle but stores the matrix column-wise.



## Sparse direct solvers: solution steps (Saad Ch. 3.6)

1. Pre-ordering
   - Decrease amount of non-zero elements generated by fill-in by re-ordering of the matrix
   - Several, graph theory based heuristic algorithms exist
2. Symbolic factorization
   - If pivoting is ignored, the indices of the non-zero elements are calculated and stored
   - Most expensive step wrt. computation time
3. Numerical factorization
   - Calculation of the numerical values of the nonzero entries
   - Moderately expensive, once the symbolic factors are available
4. Upper/lower triangular system solution
   - Fairly quick in comparison to the other steps

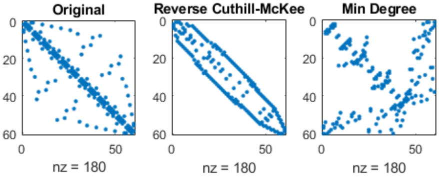
- Separation of steps 2 and 3 allows to save computational costs for problems where the sparsity structure remains unchanged, e.g. time dependent problems on fixed computational grids

- With pivoting, steps 2 and 3 have to be performed together

- Instead of pivoting, *iterative refinement* may be used in order to maintain accuracy of the solution

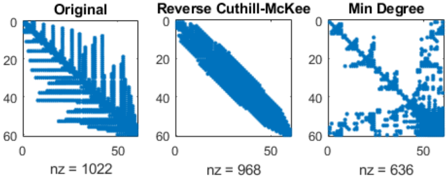
# Sparse direct solvers: influence of reordering

- Sparsity patterns for original matrix with three different orderings of unknowns – number of nonzero elements (of course) independent of ordering:



https://de.mathworks.com

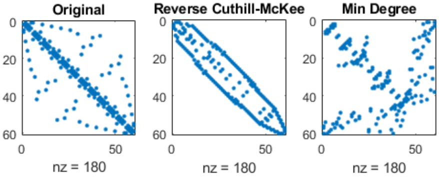
- Sparsity patterns for corresponding LU factorizations – number of nonzero elements depend original ordering!



https://de.mathworks.com

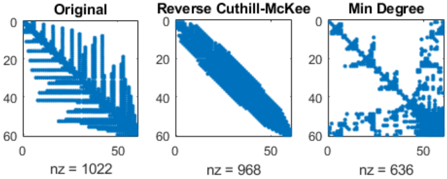
# Sparse direct solvers: influence of reordering

- Sparsity patterns for original matrix with three different orderings of unknowns – number of nonzero elements (of course) independent of ordering:

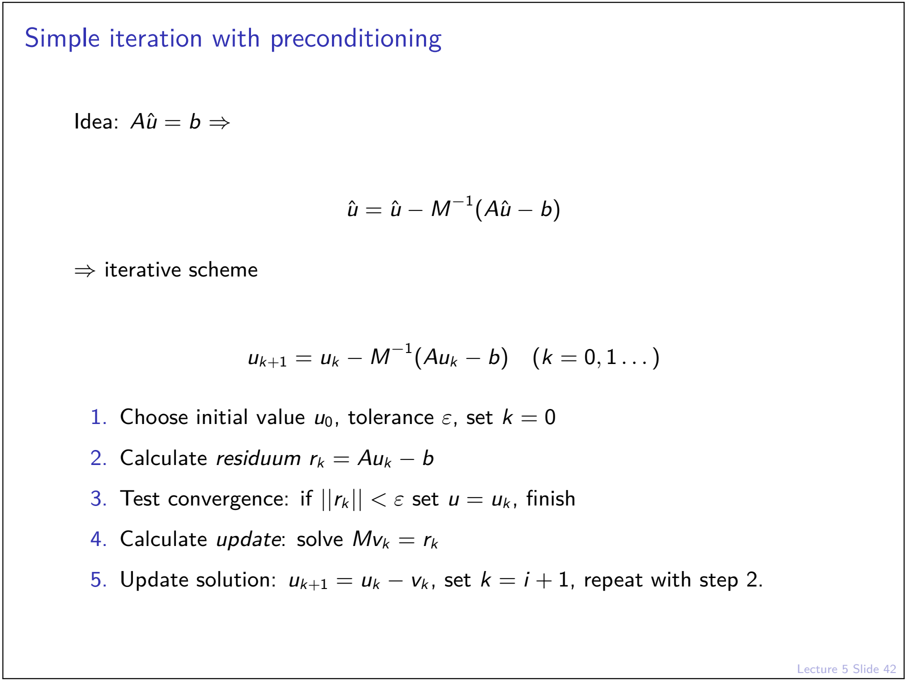


https://de.mathworks.com

- Sparsity patterns for corresponding LU factorizations – number of nonzero elements depend original ordering!



https://de.mathworks.com



## Simple iteration with preconditioning

Idea: $A\hat{u} = b \Rightarrow$

$$\hat{u} = \hat{u} - M^{-1}(A\hat{u} - b)$$

$\Rightarrow$ iterative scheme

$$u_{k+1} = u_k - M^{-1}(Au_k - b) \quad (k = 0, 1 \dots)$$

1. Choose initial value $u_0$, tolerance $\varepsilon$, set $k = 0$
2. Calculate *residuum* $r_k = Au_k - b$
3. Test convergence: if $||r_k|| < \varepsilon$ set $u = u_k$, finish
4. Calculate *update*: solve $Mv_k = r_k$
5. Update solution: $u_{k+1} = u_k - v_k$, set $k = i + 1$, repeat with step 2.

## The Jacobi method

- Let $A = D - E - F$, where $D$: main diagonal, $E$: negative lower triangular part $F$: negative upper triangular part
- Preconditioner: $M = D$, where $D$ is the main diagonal of $A \Rightarrow$

$$u_{k+1,i} = u_{k,i} - \frac{1}{a_{ii}}\left(\sum_{j=1\dots n} a_{ij}u_{k,j} - b_i\right) \quad (i = 1\dots n)$$

- Equivalent to the succesive (row by row) solution of

$$a_{ii}u_{k+1,i} + \sum_{j=1\dots n, j\neq i} a_{ij}u_{k,j} = b_i \quad (i = 1\dots n)$$

- Already calculated results not taken into account
- Alternative formulation with $A = M - N$:

$$\begin{aligned} u_{k+1} &= D^{-1}(E+F)u_k + D^{-1}b \\ &= M^{-1}Nu_k + M^{-1}b \end{aligned}$$

- Variable ordering does not matter

## The Gauss-Seidel method

- Solve for main diagonal element row by row
- Take already calculated results into account

$$a_{ii}u_{k+1,i} + \sum_{j<i} a_{ij}u_{k+1,j} + \sum_{j>i} a_{ij}u_{k,j} = b_i \qquad (i = 1 \dots n)$$

$$(D-E)u_{k+1} - Fu_k = b$$

- May be it is faster
- Variable order probably matters
- Preconditioners: forward $M = D - E$, backward: $M = D - F$
- Splitting formulation: $A = M - N$
  forward: $N = F$, backward: $M = E$
- Forward case:

$$\begin{aligned} u_{k+1} &= (D-E)^{-1}Fu_k + (D-E)^{-1}b \\ &= M^{-1}Nu_k + M^{-1}b \end{aligned}$$

# Block methods

- Jacobi, Gauss-Seidel, (S)SOR methods can as well be used block-wise, based on a partition of the system matrix into larger blocks,
- The blocks on the diagonal should be square matrices, and invertible
- Interesting variant for systems of partial differential equations, where multiple species interact with each other

## Convergence

- Let $\hat{u}$ be the solution of $Au = b$.
- Let $e_k = u_j - \hat{u}$ be the error of the $k$-th iteration step

$$\begin{aligned} u_{k+1} &= u_k - M^{-1}(Au_k - b) \\ &= (I - M^{-1}A)u_k + M^{-1}b \\ u_{k+1} - \hat{u} &= u_k - \hat{u} - M^{-1}(Au_k - A\hat{u}) \\ &= (I - M^{-1}A)(u_k - \hat{u}) \\ &= (I - M^{-1}A)^k(u_0 - \hat{u}) \end{aligned}$$

resulting in

$$e_{k+1} = (I - M^{-1}A)^k e_0$$

- So when does $(I - M^{-1}A)^k$ converge to zero for $k \to \infty$ ?

## Spectral radius and convergence

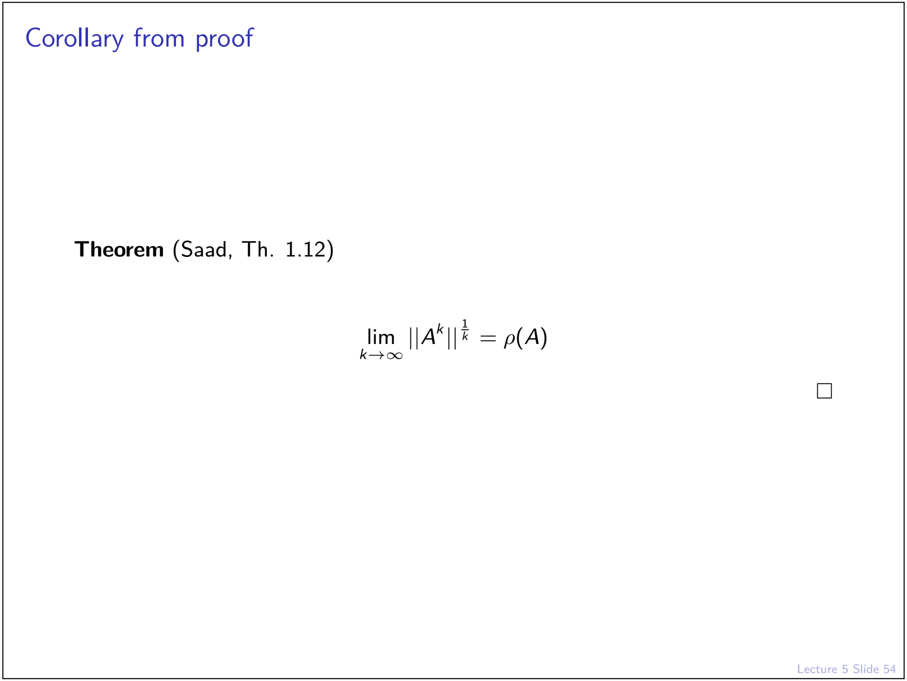
**Definition** The spectral radius $\rho(A)$ is the largest absolute value of any eigenvalue of $A$: $\rho(A) = \max_{\lambda \in \sigma(A)} |\lambda|$.

**Theorem** (Saad, Th. 1.10) $\lim\limits_{k\to\infty} A^k = 0 \Leftrightarrow \rho(A) < 1$.

**Proof**, $\Rightarrow$: Let $u_i$ be a unit eigenvector associated with an eigenvalue $\lambda_i$. Then

$$\begin{aligned}
Au_i &= \lambda_i u_i \\
A^2 u_i &= \lambda_i A_i u_i = \lambda^2 u_i \\
&\vdots \\
A^k u_i &= \lambda^k u_i \\
\text{therefore} \quad ||A^k u_i||_2 &= |\lambda^k| \\
\text{and} \quad \lim_{k\to\infty} |\lambda^k| &= 0
\end{aligned}$$

so we must have $\rho(A) < 1$



## Corollary from proof

**Theorem** (Saad, Th. 1.12)

$$\lim_{k\to\infty} ||A^k||^{\frac{1}{k}} = \rho(A)$$

□

# Back to iterative methods

Sufficient condition for convergence: $\rho(I - M^{-1}A) < 1$.

## Convergence rate

Assume $\lambda$ with $|\lambda| = \rho(I - M^{-1}A) < 1$ is the largest eigenvalue and has a single Jordan block of size $l$. Then the convergence rate is dominated by this Jordan block, and therein by the term with the lowest possible power in $\lambda$ which due to $E^l = 0$ is

$$\lambda^{k-l+1}\binom{k}{l-1}E^{l-1}$$

$$||(I - M^{-1}A)^k(u_0 - \hat{u})|| = O\left(|\lambda^{k-l+1}|\binom{k}{l-1}\right)$$

and the “worst case” convergence factor $\rho$ equals the spectral radius:

$$\begin{aligned}\rho &= \lim_{k\to\infty}\left(\max_{u_0}\frac{||(I - M^{-1}A)^k(u_0 - \hat{u})||}{||u_0 - \hat{u}||}\right)^{\frac{1}{k}}\\ &= \lim_{k\to\infty}||(I - M^{-1}A)^k||^{\frac{1}{k}}\\ &= \rho(I - M^{-1}A)\end{aligned}$$

Depending on $u_0$, the rate may be faster, though

# Richardson iteration, sufficient criterion for convergence

Assume $A$ has positive real eigenvalues $0 < \lambda_{min} \leq \lambda_i \leq \lambda_{max}$, e.g. $A$ symmetric, positive definite (spd),

- Let $\alpha > 0$, $M = \frac{1}{\alpha}I \Rightarrow I - M^{-1}A = I - \alpha A$
- Then for the eigenvalues $\mu_i$ of $I - \alpha A$ one has:

$$1 - \alpha\lambda_{max} \leq \mu_i \leq 1 - \alpha\lambda_{min}$$
$$\mu_i < 1$$

- We also need $1 - \alpha\lambda_{max} > -1$, so we must have $0 < \alpha < \frac{2}{\lambda_{max}}$.

**Theorem.** The Richardson iteration converges for any $\alpha$ with $0 < \alpha < \frac{2}{\lambda_{max}}$.

The convergence rate is $\rho = \max\left(|1 - \alpha\lambda_{max}|, |1 - \alpha\lambda_{min}|\right)$.

□

# Richardson iteration, choice of optimal parameter

- We know that

$$
\begin{aligned}
-(1-\alpha\lambda_{max}) &> -(1-\alpha\lambda_{min})\\
+(1-\alpha\lambda_{min}) &> +(1-\alpha\lambda_{max})
\end{aligned}
$$

- Therefore, in reality we have $\rho = \max\left((1-\alpha\lambda_{max}), -(1-\alpha\lambda_{min})\right)$.
- The first curve is monotonically decreasing, the second one increases, so the minimum must be at the intersection

$$
\begin{aligned}
1-\alpha\lambda_{max} &= -1+\alpha\lambda_{min}\\
2 &= \alpha(\lambda_{max}+\lambda_{min})
\end{aligned}
$$

**Theorem.** The optimal parameter is $\alpha_{opt} = \frac{2}{\lambda_{min}+\lambda_{max}}$.
For this parameter, the convergence factor is

$$\rho_{opt} = \frac{\lambda_{max}-\lambda_{min}}{\lambda_{max}+\lambda_{min}} = \frac{\kappa-1}{\kappa+1}$$

where $\kappa = \kappa(A)\frac{\lambda_{max}}{\lambda_{min}}$ is the spectral condition number of $A$. □

# Spectral equivalence

**Theorem.** $M$, $A$ spd. Assume the *spectral equivalence estimate*

$$0 < \gamma_{min}(Mu,u) \leq (Au,u) \leq \gamma_{max}(Mu,u)$$

Then for the eigenvalues $\mu_i$ of $M^{-1}A$ we have

$$\gamma_{min} \leq \mu_{min} \leq \mu_i \leq \mu_{max} \leq \gamma_{max}$$

and $\kappa(M^{-1}A) \leq \frac{\gamma_{max}}{\gamma_{\mathrm{min}}}$

**Proof**. Let the inner product $(\cdot,\cdot)_M$ be defined via $(u,v)_M = (Mu,v)$. In this inner product, $C = M^{-1}A$ is self-adjoint:

$$\begin{aligned}(Cu,v)_M &= (MM^{-1}Au,v) = (Au,v) = (M^{-1}Mu,Av) = (Mu,M^{-1}Av) \\ &= (u,M^{-1}A)_M = (u,Cv)_M\end{aligned}$$

Minimum and maximum eigenvalues can be obtained as Ritz values in the $(\cdot,\cdot)_M$ scalar product

$$\begin{aligned}\mu_{min} &= \min_{u\neq 0} \frac{(Cu,u)_M}{(u,u)_M} = \min_{u\neq 0} \frac{(Au,u)}{(Mu,u)} \geq \gamma_{min} \\ \mu_{max} &= \max_{u\neq 0} \frac{(Cu,u)_M}{(u,u)_M} = \max_{u\neq 0} \frac{(Au,u)}{(Mu,u)} \leq \gamma_{max}\end{aligned}$$

□

# Matrix preconditioned Richardson iteration

$M$, $A$ spd.

- Scaled Richardson iteration with preconditoner $M$

$$u_{k+1} = u_k - \alpha M^{-1}(Au_k - b)$$

- Spectral equivalence estimate

$$0 < \gamma_{min}(Mu, u) \leq (Au, u) \leq \gamma_{max}(Mu, u)$$

- $\Rightarrow \gamma_{min} \leq \lambda_i \leq \gamma_{max}$
- $\Rightarrow$ optimal parameter $\alpha = \frac{2}{\gamma_{max} + \gamma_{min}}$
- Convergence rate with optimal parameter: $\rho \leq \frac{\kappa(M^{-1}A) - 1}{\kappa(M^{-1}A) + 1}$
- This is one possible way for convergence analysis which at once gives convergence rates
- But ... how to obtain a good spectral estimate for a particular problem ?

# Richardson for 1D heat conduction

- Regard the $n \times n$ 1D heat conduction matrix with $h = \frac{1}{n-1}$ and $\alpha = \frac{1}{h}$ (easier to analyze).

$$A = \begin{pmatrix} \frac{2}{h} & -\frac{1}{h} & & & & & \\ -\frac{1}{h} & \frac{2}{h} & -\frac{1}{h} & & & & \\ & -\frac{1}{h} & \frac{2}{h} & -\frac{1}{h} & & & \\ & \ddots & \ddots & \ddots & \ddots & & \\ & & -\frac{1}{h} & \frac{2}{h} & -\frac{1}{h} & & \\ & & & -\frac{1}{h} & \frac{2}{h} & -\frac{1}{h} \\ & & & & -\frac{1}{h} & \frac{2}{h} \end{pmatrix}$$

- Eigenvalues (tri-diagonal Toeplitz matrix):

$$\lambda_i = \frac{2}{h}\left(1 + \cos\left(\frac{i\pi}{n+1}\right)\right) \quad (i = 1 \dots n)$$

Source: A. Böttcher, S. Grudsky: Spectral Properties of Banded Toeplitz Matrices. SIAM,2005

- Express them in $h$: $n + 1 = \frac{1}{h} + 2 = \frac{1+2h}{h} \Rightarrow$

$$\lambda_i = \frac{2}{h}\left(1 + \cos\left(\frac{ih\pi}{1+2h}\right)\right) \quad (i = 1 \dots n)$$

# Richardson for 1D heat conduction: spectral bounds

- For $i = 1 \dots n$, the argument of cos is in $(0, \pi)$
- cos is monotonically decreasing in $(0, \pi)$, so we get $\lambda_{max}$ for $i = 1$ and $\lambda_{min}$ for $i = n = \frac{1+h}{h}$
- Therefore:

$$\lambda_{max} = \frac{2}{h}\left(1 + \cos\left(\pi \frac{h}{1+2h}\right)\right) \approx \frac{2}{h}\left(2 - \frac{\pi^2 h^2}{2(1+2h)^2}\right)$$

$$\lambda_{min} = \frac{2}{h}\left(1 + \cos\left(\pi \frac{1+h}{1+2h}\right)\right) \approx \frac{2}{h}\left(\frac{\pi^2 h^2}{2(1+2h)^2}\right)$$

Here, we used the Taylor expansion

$$cos(\delta) = 1 - \frac{\delta^2}{2} + O(\delta^4) \quad (\delta \to 0)$$

$$cos(\pi - \delta) = -1 + \frac{\delta^2}{2} + O(\delta^4) \quad (\delta \to 0)$$

and $\frac{1+h}{1+2h} = \frac{1+2h}{1+2h} - \frac{h}{1+2h} = 1 - \frac{h}{1+2h}$

# Richardson for 1D heat conduction: Jacobi

- The Jacobi preconditioner just multiplies by $\frac{h}{2}$, therefore for $M^{-1}A$:

$$\mu_{max} \approx 2 - \frac{\pi^2 h^2}{2(1+2h)^2}$$
$$\mu_{min} \approx \frac{\pi^2 h^2}{2(1+2h)^2}$$

- Optimal parameter: $\alpha = \frac{2}{\lambda_{max}+\lambda_{min}} \approx 1 \ (h \to 0)$
- Good news: this is independent of $h$ resp. $n$
- No need for spectral estimate in order to work with optimal parameter
- Is this true beyond this special case ?

# Richardson for 1D heat conduction: Convergence factor

- Condition number + spectral radius

$$\kappa(M^{-1}A) = \kappa(A) = \frac{4(1+2h)^2}{\pi^2 h^2} - 1$$
$$\rho(I - M^{-1}A) = \frac{\kappa - 1}{\kappa + 1} = 1 - \frac{\pi^2 h^2}{2(1+2h)^2}$$

- Bad news: $\rho \to 1 \quad (h \to 0)$
- Typical situation with second order PDEs:

$$\kappa(A) = O(h^{-2}) \quad (h \to 0)$$
$$\rho(I - D^{-1}A) = 1 - O(h^2) \quad (h \to 0)$$

- Mean square error of approximation $||u - u_h||_2 < h^{\gamma}$, in the simplest case $\gamma = 2$.

# Iterative solver complexity I

- Solve linear system iteratively until $||e_k|| = ||(I - M^{-1}A)^k e_0|| \leq \epsilon$

$$\begin{aligned} \rho^k e_0 &\leq \epsilon \\ k \ln \rho &< \ln \epsilon - \ln e_0 \\ k \geq k_\rho &= \left\lceil \frac{\ln e_0 - \ln \epsilon}{\ln \rho} \right\rceil \end{aligned}$$

- $\Rightarrow$ we need at least $k_\rho$ iteration steps to reach accuracy $\epsilon$
- Optimal iterative solver complexity - assume:
  - $\rho < \rho_0 < 1$ independent of $h$ resp. $N$
  - $A$ sparse ($A \cdot u$ has complexity $O(N)$)
  - Solution of $Mv = r$ has complexity $O(N)$.

$\Rightarrow$ Number of iteration steps $k_\rho$ independent of $N$
$\Rightarrow$ Overall complexity $O(N)$

# Iterative solver complexity II

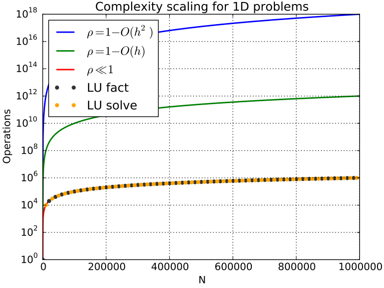
- Assume
  - $\rho = 1 - h^\delta \Rightarrow \ln\rho \approx -h^\delta \to k_\rho = O(h^{-\delta})$
  - $d$: space dimension $\Rightarrow h \approx N^{-\frac{1}{d}} \Rightarrow k_\rho = O(N^{\frac{\delta}{d}})$
  - $O(N)$ complexity of one iteration step (e.g. Jacobi, Gauss-Seidel)

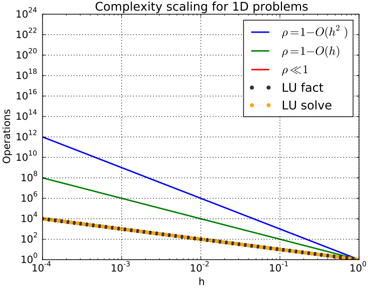
  $\Rightarrow$ Overall complexity $O(N^{1+\frac{\delta}{d}})$=$O(N^{\frac{d+\delta}{d}})$
- Jacobi: $\delta = 2$
- Hypothetical "Improved iterative solver" with $\delta = 1$ ?
- Overview on complexity estimates

| dim | $\rho = 1 - O(h^2)$ | $\rho = 1 - O(h)$ | LU fact. | LU solve |
|---|---|---|---|---|
| 1 | $O(N^3)$ | $O(N^2)$ | $O(N)$ | $O(N)$ |
| 2 | $O(N^2)$ | $O(N^{\frac{3}{2}})$ | $O(N^{\frac{3}{2}})$ | $O(N\log N)$ |
| 3 | $O(N^{\frac{5}{3}})$ | $O(N^{\frac{4}{3}})$ | $O(N^2)$ | $O(N^{\frac{4}{3}})$ |

# Solver complexity scaling for 1D problems

| dim | $\rho = 1 - O(h^2)$ | $\rho = 1 - O(h)$ | LU fact. | LU solve |
|---|---|---|---|---|
| 1 | $O(N^3)$ | $O(N^2)$ | $O(N)$ | $O(N)$ |

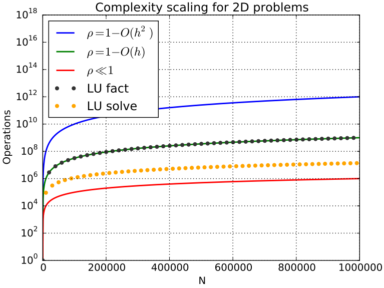


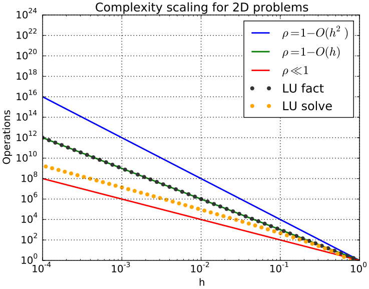


- Direct solvers significantly better than iterative ones

# Solver complexity scaling for 2D problems

| dim | $\rho = 1 - O(h^2)$ | $\rho = 1 - O(h)$ | LU fact. | LU solve |
|---|---|---|---|---|
| 2 | $O(N^2)$ | $O(N^{\frac{3}{2}})$ | $O(N^{\frac{3}{2}})$ | $O(N \log N)$ |

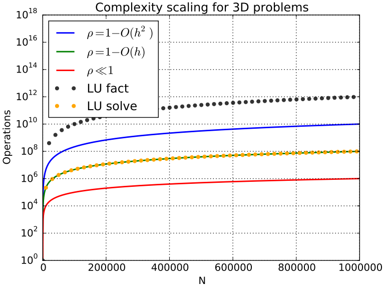


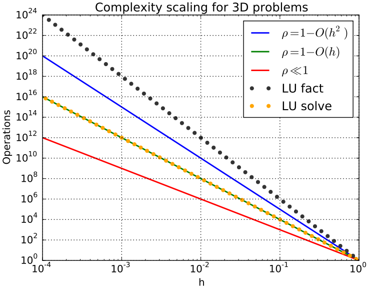


- Direct solvers better than simple iterative solvers (Jacobi etc.)
- On par with improved iterative solvers

# Solver complexity scaling for 3D problems

| dim | $\rho = 1 - O(h^2)$ | $\rho = 1 - O(h)$ | LU fact. | LU solve |
|---|---|---|---|---|
| 3 | $O(N^{\frac{5}{3}})$ | $O(N^{\frac{4}{3}})$ | $O(N^2)$ | $O(N^{\frac{4}{3}})$ |





- LU factorization is extremly expensive
- LU solve on par with improved iterative solvers

# What could be done ?

- Find optimal iterative solver with $O(N)$ complexity
- Find "improved preconditioner" with $\kappa(M^{-1}A) = O(h^{-1}) \Rightarrow \delta = 1$
- Find "improved iterative scheme": with $\rho = \frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}$:

  For Jacobi, we had $\kappa = X^2 - 1$ where $X = \frac{2(1+2h)}{\pi h} = O(h^{-1})$.

$$
\begin{aligned}
\rho &= 1 + \frac{\sqrt{X^2-1}-1}{\sqrt{X^2-1}+1} - 1 \\
&= 1 + \frac{\sqrt{X^2-1}-1-\sqrt{X^2-1}-1}{\sqrt{X^2-1}+1} \\
&= 1 - \frac{1}{\sqrt{X^2-1}+1} \\
&= 1 - \frac{1}{X\left(\sqrt{1-\frac{1}{X^2}}+\frac{1}{X}\right)} \\
&= 1 - O(h)
\end{aligned}
$$

$\Rightarrow \delta = 1$

# Generalization of iteration schemes

- Simple iterations converge slowly
- For most practical purposes, Krylov subspace methods are used.
- We will introduce one special case and give hints on practically useful more general cases
- Material after J. Shewchuk: An Introduction to the Conjugate Gradient Method Without the Agonizing Pain“

# Solution of SPD system as a minimization procedure

Regard $Au = f$ ,where $A$ is symmetric, positive definite. Then it defines a bilinear form $a : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$

$$a(u, v) = (Au, v) = v^T Au = \sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij} v_i u_j$$

As $A$ is SPD, for all $u \neq 0$ we have $(Au, u) > 0$.

For a given vector $b$, regard the function

$$f(u) = \frac{1}{2} a(u, u) - b^T u$$

What is the minimizer of $f$ ?

$$f'(u) = Au - b = 0$$

- Solution of SPD system $\equiv$ minimization of $f$.

# Method of steepest descent

- Given some vector $u_i$, look for a new iterate $u_{i+1}$.
- The direction of steepest descend is given by $-f'(u_i)$.
- So look for $u_{i+1}$ in the direction of $-f'(u_i) = r_i = b - Au_i$ such that it minimizes f in this direction, i.e. set $u_{i+1} = u_i + \alpha r_i$ with $\alpha$ choosen from

$$\begin{aligned}
0 &= \frac{d}{d\alpha} f(u_i + \alpha r_i) = f'(u_i + \alpha r_i) \cdot r_i \\
&= (b - A(u_i + \alpha r_i), r_i) \\
&= (b - Au_i, r_i) - \alpha(Ar_i, r_i) \\
&= (r_i, r_i) - \alpha(Ar_i, r_i) \\
\alpha &= \frac{(r_i, r_i)}{(Ar_i, r_i)}
\end{aligned}$$

# Method of steepest descent: iteration scheme

$$r_i = b - Au_i$$

$$\alpha_i = \frac{(r_i, r_i)}{(Ar_i, r_i)}$$

$$u_{i+1} = u_i + \alpha_i r_i$$

Let $\hat{u}$ the exact solution. Define $e_i = u_i - \hat{u}$, then $r_i = -Ae_i$

Let $||u||_A = (Au, u)^{\frac{1}{2}}$ be the *energy norm* wrt. A.

**Theorem** The convergence rate of the method is

$$||e_i||_A \leq \left(\frac{\kappa - 1}{\kappa + 1}\right)^i ||e_0||_A$$

where $\kappa = \frac{\lambda_{max}(A)}{\lambda_{min}(A)}$ is the spectral condition number.

# Method of steepest descent: advantages

- Simple Richardson iteration $u_{k+1} = u_k - \alpha(Au_k - f)$ needs good eigenvalue estimate to be optimal with $\alpha = \frac{2}{\lambda_{max}+\lambda_{min}}$
- In this case, asymptotic convergence rate is $\rho = \frac{\kappa-1}{\kappa+1}$
- Steepest descent has the same rate without need for spectral estimate

# Conjugate directions

For steepest descent, there is no guarantee that a search direction $d_i = r_i = -Ae_i$ is not used several times. If all search directions would be orthogonal, or, indeed, $A$-orthogonal, one could control this situation.

So, let $d_0, d_1 \dots d_{n-1}$ be a series of $A$-orthogonal (or conjugate) search directions, i.e. $(Ad_i, d_j) = 0$, $i \neq j$.

- Look for $u_{i+1}$ in the direction of $d_i$ such that it minimizes f in this direction, i.e. set $u_{i+1} = u_i + \alpha_i d_i$ with $\alpha$ choosen from

$$\begin{aligned}
0 &= \frac{d}{d\alpha} f(u_i + \alpha d_i) = f'(u_i + \alpha d_i) \cdot d_i \\
&= (b - A(u_i + \alpha d_i), d_i) \\
&= (b - Au_i, d_i) - \alpha(Ad_i, d_i) \\
&= (r_i, d_i) - \alpha(Ad_i, d_i) \\
\alpha_i &= \frac{(r_i, d_i)}{(Ad_i, d_i)}
\end{aligned}$$

# Conjugate directions II

$e_0 = u_0 - \hat{u}$ (such that $Ae_0 = -r_0$) can be represented in the basis of the search directions:

$$e_0 = \sum_{i=0}^{n-1} \delta_j d_j$$

Projecting onto $d_k$ in the $A$ scalar product gives

$$\begin{aligned}
(Ae_0, d_k) &= \sum_{i=0}^{n-1} \delta_j (Ad_j, d_k) \\
&= \delta_k (Ad_k, d_k) \\
\delta_k &= \frac{(Ae_0, d_k)}{(Ad_k, d_k)} = \frac{(Ae_0 + \sum_{i<k} \alpha_i d_i, d_k)}{(Ad_k, d_k)} = \frac{(Ae_k, d_k)}{(Ad_k, d_k)} \\
&= \frac{(r_k, d_k)}{(Ad_k, d_k)} \\
&= -\alpha_k
\end{aligned}$$

# Conjugate directions III

Then,

$$e_i = e_0 + \sum_{j=0}^{i-1} \alpha_j d_j = -\sum_{j=0}^{n-1} \alpha_j d_j + \sum_{j=0}^{i-1} \alpha_j d_j$$
$$= -\sum_{j=i}^{n-1} \alpha_j d_j$$

So, the iteration consists in component-wise suppression of the error, and it must converge after $n$ steps. Let $k \leq i$. $A$-projection on $d_k$ gives

$$(Ae_i, d_k) = -\sum_{j=i}^{n-1} \alpha_j (Ad_j, d_k) = 0$$

Therefore, $r_i = Ae_i$ is orthogonal to $d_0 \dots d_{i-1}$.

# Conjugate directions IV

Looking at the error norm $||e_i||_A$, the method yields the element with the minimum energy norm from all elements of the affine space $e_0 + \mathcal{K}_i$ where $\mathcal{K}_i = \mathrm{span}\{d_0, d_1 \dots d_{i-1}\}$

$$(Ae_i, e_i) = \left(\sum_{j=i}^{n-1} \delta_j d_j, \sum_{j=i}^{n-1} \delta_j d_j\right) = \sum_{j=i}^{n-1}\sum_{k=i}^{n-1} \delta_j \delta_k (d_j, d_k)$$
$$= \sum_{j=i}^{n-1} \delta_j^2 (d_j, d_j) = \min_{e \in e_0 + \mathcal{K}_i} ||e||_A$$

Furthermore, we have

$$\begin{aligned} u_{i+1} &= u_i + \alpha_i d_i \\ e_{i+1} &= e_i + \alpha_i d_i \\ Ae_{i+1} &= Ae_i + \alpha_i Ad_i \\ r_{i+1} &= r_i - \alpha_i Ad_i \end{aligned}$$

By what magic we can obtain these $d_i$?

# Gram-Schmidt Orthogonalization

- Assume we have been given some linearly independent vectors $v_0, v_1 \dots v_{n-1}$.
- Set $d_0 = v_0$
- Define

$$d_i = v_i + \sum_{k=0}^{i-1} \beta_{ik} d_k$$

- For $j < i$, A-project onto $d_j$ and require orthogonality:

$$\begin{aligned} (Ad_i, d_j) &= (Av_i, d_j) + \sum_{k=0}^{i-1} \beta_{ik}(Ad_k, d_j) \\ 0 &= (Av_i, d_j) + \beta_{ij}(Ad_j, d_j) \\ \beta_{ij} &= -\frac{(Av_i, d_j)}{(Ad_j, d_j)} \end{aligned}$$

- If $v_i$ are the coordinate unit vectors, this is Gaussian elimination!
- If $v_i$ are arbitrary, they all must be kept in the memory

# Conjugate gradients (Hestenes, Stiefel, 1952)

As Gram-Schmidt builds up $d_i$ from $d_j$, $j < i$, we can choose $v_i = r_i$, i.e. the residuals built up during the conjugate direction process.

Let $\mathcal{K}_i = \mathrm{span}\{d_0 \dots d_{i-1}\}$. Then, $r_i \perp \mathcal{K}_i$

But $d_i$ are built by Gram-Schmidt from the residuals, so we also have $\mathcal{K}_i = \mathrm{span}\{r_0 \dots r_{i-1}\}$ and $(r_i, r_j) = 0$ for $j < i$.

From $r_i = r_{i-1} - \alpha_{i-1} A d_{i-1}$ we obtain

$\mathcal{K}_i = \mathcal{K}_{i-1} \cup \mathrm{span}\{A d_{i-1}\}$

This gives two other representations of $\mathcal{K}_i$:

$$\begin{aligned}\mathcal{K}_i &= \mathrm{span}\{d_0, Ad_0, A^2 d_0, \dots, A^{i-1} d_0\}\\ &= \mathrm{span}\{r_0, Ar_0, A^2 r_0, \dots, A^{i-1} r_0\}\end{aligned}$$

Such type of subspace of $\mathbb{R}^n$ is called *Krylov subspace*, and orthogonalization methods are more often called *Krylov subspace methods*.

# Conjugate gradients II

Look at Gram-Schmidt under these conditions. The essential data are (setting $v_i = r_i$ and using $j < i$) $\beta_{ij} = -\frac{(Ar_i, d_j)}{(Ad_j, d_j)} = -\frac{(Ad_j, r_i)}{(Ad_j, d_j)}$.

Then, for $j \leq i$:

$$\begin{aligned}
r_{j+1} &= r_j - \alpha_j A d_j \\
(r_{j+1}, r_i) &= (r_j, r_i) - \alpha_j (Ad_j, r_i) \\
\alpha_j (Ad_j, r_i) &= (r_j, r_i) - (r_{j+1}, r_i) \\
(Ad_j, r_i) &= \begin{cases} -\frac{1}{\alpha_j}(r_{j+1}, r_i), & j+1 = i \\ \frac{1}{\alpha_j}(r_j, r_i), & j = i \\ 0, & \text{else} \end{cases} = \begin{cases} -\frac{1}{\alpha_{i-1}}(r_i, r_i), & j+1 = i \\ \frac{1}{\alpha_i}(r_i, r_i), & j = i \\ 0, & \text{else} \end{cases}
\end{aligned}$$

For $j < i$:

$$\beta_{ij} = \begin{cases} \frac{1}{\alpha_{i-1}} \frac{(r_i, r_i)}{(Ad_{i-1}, d_{i-1})}, & j+1 = i \\ 0, & \text{else} \end{cases}$$

# Conjugate gradients III

For Gram-Schmidt we defined (replacing $v_i$ by $r_i$):

$$\begin{aligned} d_i &= r_i + \sum_{k=0}^{i-1} \beta_{ik} d_k \\ &= r_i + \beta_{i,i-1} d_{i-1} \end{aligned}$$

So, the new orthogonal direction depends only on the previous orthogonal direction and the current residual. We don't have to store old residuals or search directions. In the sequel, set $\beta_i := \beta_{i,i-1}$.

We have

$$\begin{aligned} d_{i-1} &= r_{i-1} + \beta_{i-1} d_{i-2} \\ (d_{i-1}, r_{i-1}) &= (r_{i-1}, r_{i-1}) + \beta_{i-1}(d_{i-2}, r_{i-1}) \\ &= (r_{i-1}, r_{i-1}) \\ \beta_i &= \frac{1}{\alpha_{i-1}} \frac{(r_i, r_i)}{(Ad_{i-1}, d_{i-1})} = \frac{(r_i, r_i)}{(d_{i-1}, r_{i-1})} \\ &= \frac{(r_i, r_i)}{(r_{i-1}, r_{i-1})} \end{aligned}$$

# Conjugate gradients IV - The algorithm

Given initial value $u_0$, spd matrix A, right hand side $b$.

$$\begin{aligned} d_0 &= r_0 = b - Au_0 \\ \alpha_i &= \frac{(r_i, r_i)}{(Ad_i, d_i)} \\ u_{i+1} &= u_i + \alpha_i d_i \\ r_{i+1} &= r_i - \alpha_i Ad_i \\ \beta_{i+1} &= \frac{(r_{i+1}, r_{i+1})}{(r_i, r_i)} \\ d_{i+1} &= r_{i+1} + \beta_{i+1} d_i \end{aligned}$$

At the i-th step, the algorithm yields the element from $e_0 + \mathcal{K}_i$ with the minimum energy error.

**Theorem** The convergence rate of the method is

$$||e_i||_A \leq 2 \left( \frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1} \right)^i ||e_0||_A$$

where $\kappa = \frac{\lambda_{max}(A)}{\lambda_{min}(A)}$ is the spectral condition number.

# Preconditioning

Let $M$ be spd, and spectrally equivalent to $A$, and assume that $\kappa(M^{-1}A) << \kappa(A)$.

Let $E$ be such that $M = EE^T$, e.g. its Cholesky factorization. Then, $\sigma(M^{-1}A) = \sigma(E^{-1}AE^{-T})$:

Assume $M^{-1}Au = \lambda u$. We have

$$(E^{-1}AE^{-T})(E^Tu) = (E^TE^{-T})E^{-1}Au = E^TM^{-1}Au = \lambda E^Tu$$

$\Leftrightarrow$ $E^Tu$ is an eigenvector of $E^{-1}AE^{-T}$ with eigenvalue $\lambda$.

# Preconditioned CG I

Now we can use the CG algorithm for the preconditioned system

$$E^{-1}AE^{-T}\tilde{x} = E^{-1}b$$

with $\tilde{u} = E^T u$

$$\begin{aligned}
\tilde{d}_0 &= \tilde{r}_0 = E^{-1}b - E^{-1}AE^{-T}u_0 \\
\alpha_i &= \frac{(\tilde{r}_i, \tilde{r}_i)}{(E^{-1}AE^{-T}\tilde{d}_i, \tilde{d}_i)} \\
\tilde{u}_{i+1} &= \tilde{u}_i + \alpha_i \tilde{d}_i \\
\tilde{r}_{i+1} &= \tilde{r}_i - \alpha_i E^{-1}AE^{-T}\tilde{d}_i \\
\beta_{i+1} &= \frac{(\tilde{r}_{i+1}, \tilde{r}_{i+1})}{(\tilde{r}_i, \tilde{r}_i)} \\
\tilde{d}_{i+1} &= \tilde{r}_{i+1} + \beta_{i+1}\tilde{d}_i
\end{aligned}$$

Not very practical as we need $E$

# Preconditioned CG II

Assume $\tilde{r}_i = E^{-1} r_i$, $\tilde{d}_i = E^T d_i$, we get the equivalent algorithm

$$
\begin{aligned}
r_0 &= b - Au_0 \\
d_0 &= M^{-1} r_0 \\
\alpha_i &= \frac{(M^{-1} r_i, r_i)}{(Ad_i, d_i)} \\
u_{i+1} &= u_i + \alpha_i d_i \\
r_{i+1} &= r_i - \alpha_i A d_i \\
\beta_{i+1} &= \frac{(M^{-1} r_{i+1}, r_{i+1})}{(r_i, r_i)} \\
d_{i+1} &= M^{-1} r_{i+1} + \beta_{i+1} d_i
\end{aligned}
$$

It relies on the solution of the preconditioning system, the calculation of the matrix vector product and the calculation of the scalar product.

# A few issues

Usually we stop the iteration when the residual $r$ becomes small. However during the iteration, floating point errors occur which distort the calculations and lead to the fact that the accumulated residuals

$$r_{i+1} = r_i - \alpha_i A d_i$$

give a much more optimistic picture on the state of the iteration than the real residual

$$r_{i+1} = b - A u_{i+1}$$