Scientific Computing WS 2017/2018

Lecture 7

Jürgen Fuhrmann

juergen.fuhrmann@wias-berlin.de

## numcxx

numcxx is a small C++ library developed for and during this course which implements the concepts introduced

- Shared smart pointers vs. references
- 1D/2D Array class
- Matrix class with LAPACK interface
- Expression templates
- Interface to triangulations
- Sparse matrices + UMFPACK interface
- Iterative solvers
- Python interface

# numcxx availability

- UNIX pool installation in `/net/wir/numcxx`
- Code home page
  `https://www.wias-berlin.de/people/fuhrmann/numcxx.html`
  - Documentation incl. installation instructions
  - Zip files with code for download

## numcxx classes

- `TArray1`: templated 1D array class
  `DArray1`: 1D double array class
- `TArray2`: templated 2D array class
  `DArray2`: 2D double array class
- `TMatrix`: templated dense matrix class
  `DMatrix`: double dense matrix class
- `TSolverLapackLU`: LU factorization based on LAPACK
  `DSolverLapackLU`

# CRS again

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

```
AA: 1. 2. 3. 4. 5. 6. 7. 8. 9. 10. 11. 12.
JA: 0 3 0 1 3 0 2 3 4 2 3 4
IA: 0 2 4 0 11 12
```

- some package APIs provide the possibility to specify array offset
- index shift is not very expensive compared to the rest of the work

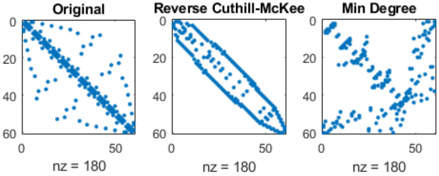
# numcxx Sparse matrix class

`numcxx::TSparseMatrix<T>`

- Class characterized by IA/JA/AA arrays
- How to create these arrays ?
- Common way (e.g. Eigen) : from a list triples $i, j, a_{ij}$. In practice, this can be expensive because in FEM assembly we will have many triplets repeating with the same $i, j$ but different $a_{ij}$
- Remedy:
  - Internally create and update an intermediate datas structure which maintains a list of already available entries
  - Hide this behind the facade $A(i, j) = x$

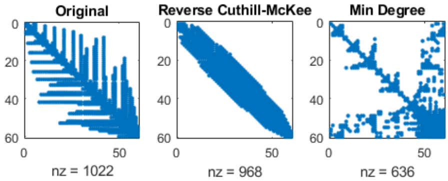
## Sparse direct solvers: influence of reordering

- Sparsity patterns for original matrix with three different orderings of unknowns unknowns:




https://de.mathworks.com

- Sparsity patterns for corresponding LU factorizations unknowns:




https://de.mathworks.com

## 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
   - Not very 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: Complexity

- Complexity estimates depend on storage scheme, reordering etc.
- Sparse matrix - vector multiplication has complexity $O(N)$
- Some estimates can be given for from graph theory for discretizations of heat eqauation with $N = n^d$ unknowns on close to cubic grids in space dimension $d$
  - sparse LU factorization:

| d | work | storage |
|---|---|---|
| 1 | $O(N) \mid O(n)$ | $O(N) \mid O(n)$ |
| 2 | $O(N^{\frac{3}{2}}) \mid O(n^3)$ | $O(N \log N) \mid O(n^2 \log n)$ |
| 3 | $O(N^2) \mid O(n^6)$ | $O(N^{\frac{4}{3}}) \mid O(n^4)$ |

  - triangular solve: work dominated by storage complexity

| d | work |
|---|---|
| 1 | $O(N) \mid O(n)$ |
| 2 | $O(N \log N) \mid O(n^2 \log n)$ |
| 3 | $O(N^{\frac{4}{3}}) \mid O(n^4)$ |

Source: J. Poulson, PhD thesis,http://hdl.handle.net/2152/ETD-UT-2012-12-6622

## 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

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

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

- 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:

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

## 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) $\lim_{k\to\infty} A^k = 0 \Leftrightarrow \rho(A) < 1$.

**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 $\rho(A) < 1$

# 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:
  $1 - \alpha\lambda_{max} \leq \mu_i \leq 1 - \alpha\lambda_{min}$
  and $\mu_i < 1$ due to $\lambda_{min} > 0$
- We also need $1 - \alpha\lambda_{max} > -1 \Rightarrow 0 < \alpha < \frac{2}{\lambda_{max}}$.

**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

$$-(1-\lambda_{max}\alpha) > -(1-\lambda_{min}\alpha)$$
$$+(1-\lambda_{min}\alpha) > +(1-\lambda_{max}\alpha)$$

- 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

$$1-\alpha\lambda_{max} = -1+\alpha\lambda_{min}$$
$$2 = \alpha(\lambda_{max}+\lambda_{min})$$

**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 $\lambda_i$ of $M^{-1}A$ we have

$$\gamma_{min} \leq \lambda_{min} \leq \lambda_i \leq \lambda_{max} \leq \gamma_{max}$$

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

**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

$$\lambda_{min} = \min_{u\neq 0} \frac{(Cu,u)_M}{(u,u)_M} = \min_{u\neq 0} \frac{(Au,u)}{(Mu,u)} \geq \gamma_{min}$$

$$\lambda_{max} = \max_{u\neq 0} \frac{(Cu,u)_M}{(u,u)_M} = \max_{u\neq 0} \frac{(Au,u)}{(Mu,u)} \leq \gamma_{max}$$

□

# 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 $M^{-1}A$:

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

- 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:

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

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

# Iterative solver complexity I

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

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

- Assume $\rho < \rho_0 < 1$ independent of $h$ resp. $N$, $A$ sparse and solution of $Mv = r$ has complexity $O(N)$.
  $\Rightarrow$ Number of iteration steps $k_\rho$ independent of $N$
  $\Rightarrow$ Overall complexity $O(N)$.

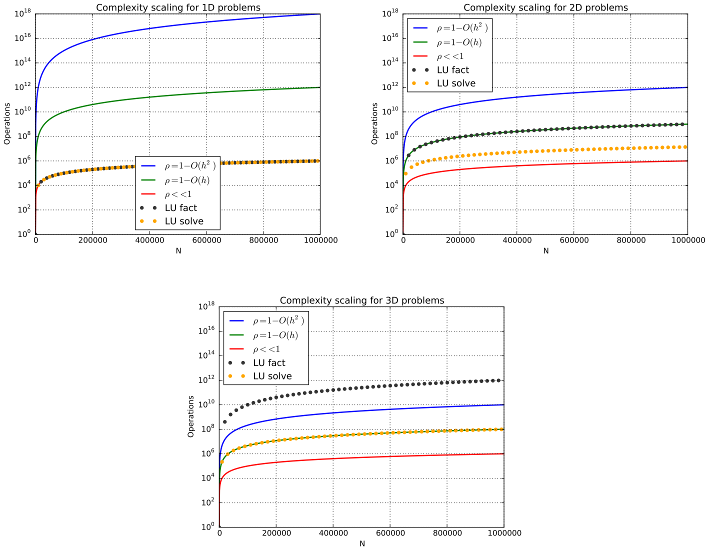
# Iterative solver complexity II

- Assume $\rho = 1 - h^\delta \Rightarrow \ln \rho \approx -h^\delta$
- $k = O(h^{-\delta})$
- $d$: space dimension, then $h \approx N^{-\frac{1}{d}} \Rightarrow k = O(N^{\frac{\delta}{d}})$
- Assume $O(N)$ complexity of one iteration step
  $\Rightarrow$ Overall complexity $O(N^{\frac{d+\delta}{d}})$
- Jacobi: $\delta = 2$, something better with at least $\delta = 1$ ?

| 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 | $O(N^{\frac{5}{3}})$ | $O(N^{\frac{4}{3}})$ | $O(N^2)$ | $O(N^{\frac{4}{3}})$ |

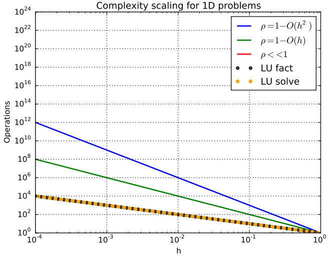
- In 1D, iteration makes not much sense
- In 2D, we can hope for parity
- In 3D, beat sparse matrix solvers with $\rho = 1 - O(h)$ ?

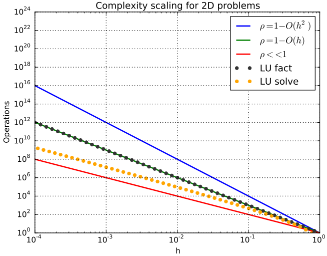
# Solver complexity: scaling with problem size

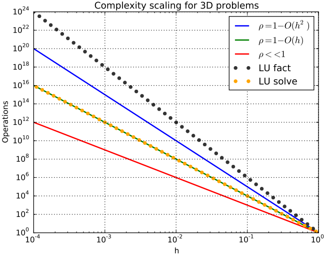



Scaling with problem size.

# Solver complexity: scaling with accuracy




- Accuracy of numerial solutions is proportional to some power of $h$.
- Amount of operations for to reach a given accuracy.

# What could be done ?

- Find a better preconditioner with $\kappa(M^{-1}A) = O(h^{-1})$ or independent of $h$
- Find a better iterative scheme:
  Assume e.g. $\rho = \frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}$. Let $\kappa = X^2 - 1$ where $X = \frac{2(1+2h)}{\pi h} = O(h^{-1})$.

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

- Here, we would have $\delta = 1$. Together with a good preconditioner ...