

Recall: Iterative solvers

Gundolf Haase

Karl-Franzens University Graz

Isfahan, Jan 2019

MONT-BLANC

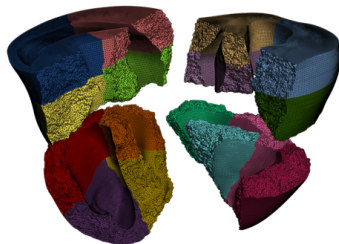
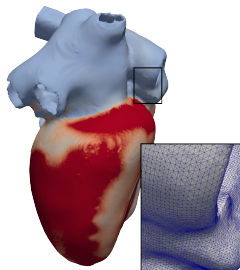


BioTechMed[®]
GRAZ

What we want to solve

We want to solve **partial differential equations** (PDEs)

- on parallel computers (distributed memory + shared memory)
- with $10^3 - 10^4$ CPU cores
- for $10^6 - 10^8$ discretization points.



Considered Problem Classes

$$\begin{array}{ll} \text{Find } u : & Lu(x) = f(x) \quad \forall x \in \Omega \\ & lu(x) = g(x) \quad \forall x \in \partial\Omega \end{array}$$

variational \Downarrow formulation

$$\text{Find } u \in \mathbb{V} : \quad a(u, v) = \langle F, v \rangle \quad \forall v \in \mathbb{V}$$

FEM, FDM \Downarrow FVM, FIT

$$\text{Solve } K_h \cdot \underline{u}_h = \underline{f}_h \quad \underline{u}_h \in \mathbb{R}^{N_h}$$

With K_h

- symmetric, i.e., $K_h \equiv K_h^T$
- positive definite, i.e., $\langle K_h \underline{u}_h, \underline{u}_h \rangle > 0$
 $\forall \underline{u}_h \notin \ker(K_h)$

- (linear) 2nd order problem.
 - ▶ Poisson equation (temperature)
 - ▶ Lamé equation (deformation)
 - ▶ Maxwell's equations (magnetic field)
- Matrix K_h is sparse, positive definite (symmetric, large dimension)
- non-linear and time-dependent problems.

Richardson iteration to solve $Ku = f$

- Richardson iteration:

$$\underline{u}^{k+1} := \underline{u}^k + \tau \left(\underline{f} - K \cdot \underline{u}^k \right)$$

$$\text{with } \tau_{opt} = \frac{2}{\lambda_{max}(K) + \lambda_{min}(K)}.$$

- Preconditioned Richardson iteration:

$$\underline{u}^{k+1} := \underline{u}^k + C^{-1} \left(\underline{f} - K \cdot \underline{u}^k \right)$$

with $C \approx K$ and $C\underline{u} = \underline{f}$ can be faster solved than $K\underline{u} = \underline{f}$.

Choice $C := \frac{1}{\tau} I$ (scaled Identity) results in classical Richardson.

Richardson iteration: algorithm

Algorithm 1 Preconditioned Richardson for $Ku = f$

- 1: Choose $u^0 = 0$; $k = 0$ ▷ initial guess
 - 2: $s0 = \langle C^{-1} \cdot f, f \rangle$ ▷ initial error²
 - 3: **repeat**
 - 4: $r := f - K \cdot u^k$ ▷ defect
 - 5: Solve $Cw = r$ ▷ correction
 - 6: $u^{k+1} := u^k + w$ ▷ update
 - 7: **until** $\langle w, r \rangle > s0 \cdot \varepsilon^2$ ▷ relative accuracy
-

Basic operations:

- sparse matrix-vector product
- vector operations, inner product
- preconditioning system $Cw = r$

Jacobi iteration

Special Richardson iteration with preconditioner

$$C = \frac{1}{\omega} D$$

denoting $D := \text{diag}(K)$.

- Jacobi iteration (solver: $\omega := 1$):

$$\underline{u}^{k+1} := \underline{u}^k + \omega D^{-1} \left(\underline{f} - K \cdot \underline{u}^k \right)$$

- with the preconditioning step

$$w := \omega D^{-1} \cdot r$$

ILU iteration

Special Richardson iteration with preconditioner

$$C = L \cdot U \quad \Longleftrightarrow \quad L \cdot U \cdot w = r$$

resulting from an incomplete factorization $K = L \cdot U + R$, with a remainder R .

- ILU iteration:

$$\underline{u}^{k+1} := \underline{u}^k + U^{-1}L^{-1} \left(\underline{f} - K \cdot \underline{u}^k \right)$$

- with the preconditioning steps

$$\text{solve} \quad L \cdot v = r$$

$$\text{solve} \quad U \cdot w = v$$

Preconditioned cg iteration to solve $Ku = f$

Choose \underline{u}^0

$$\underline{r} := \underline{f} - K \cdot \underline{u}^0$$

$$\underline{w} := C^{-1} \cdot \underline{r}$$

$$\underline{s} := \underline{w}$$

$$\sigma := \sigma_{\text{old}} := \sigma_0 := (\underline{w}, \underline{r})$$

repeat

$$\underline{v} := K \cdot \underline{s}$$

$$\alpha := \sigma / (\underline{s}, \underline{v})$$

$$\underline{u} := \underline{u} + \alpha \cdot \underline{s}$$

$$\underline{r} := \underline{r} - \alpha \cdot \underline{v}$$

$$\underline{w} := C^{-1} \cdot \underline{r}$$

$$\sigma := (\underline{w}, \underline{r})$$

$$\beta := \sigma / \sigma_{\text{old}}$$

$$\sigma_{\text{old}} := \sigma$$

$$\underline{s} := \underline{w} + \beta \cdot \underline{s}$$

until $\sigma < \varepsilon^2 \cdot \sigma_0$

Preconditioning opportunities

Many opportunities for choosing C in $Cw = r \iff w = C^{-1}r$

- $C = I$: no preconditioning, or scaled as $C = \frac{1}{\omega}I$
- $C = \frac{1}{\omega}D$: Jacobi preconditioning / diagonal scaling
(one Jacobi iteration with initial guess $u^0 = 0$)
- $C = L \cdot U$: ILU-preconditioning
- Approximativ inverse as SPAI
- Additive/multiplicative Schwarz methods (ASM/MSM) incl.
 - ▶ Multilevel (ASM)
 - ▶ Multigrid (MSM)
 - ▶ Domain decomposition methods (DDM)
- Multigrid: $C^{-1} = (I_\ell - M_\ell)K^{-1}$ with

$$M_k = S_k^{post} (I_k - P_k(I_{k-1} - M_{k-1})K_{k-1}^{-1}R_kK_k) S_k^{pre}$$

or as two-grid: $M_1 = S_1^{post} (I_1 - P_1K_0^{-1}R_1K_1) S_1^{pre}$