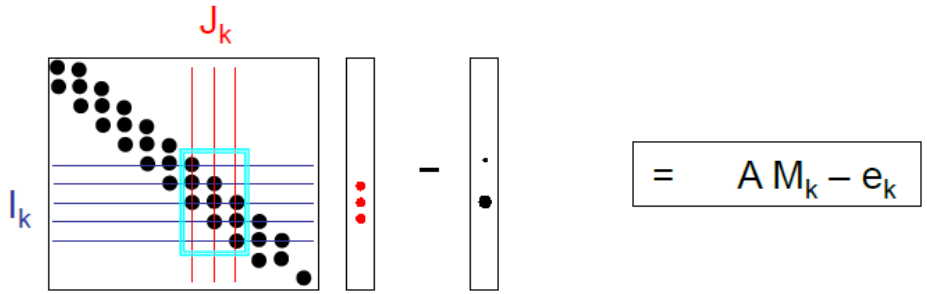
SPAI and Sparse Least Squares

- $A(:, \mathcal{J}_k)$ is a sparse rectangular matrix. $\mathcal{I}_k$ is index set called *shadow* of $\mathcal{J}_k$.
- Delete superfluous zeros in least squares problem:
  - For index set $\mathcal{J}_k$ in M_k keep only $A(:, \mathcal{J}_k)$
  - In $A(:, \mathcal{J}_k)$ keep only nonzero rows $A(\mathcal{I}_k, \mathcal{J}_k)$



- Hence, reduction of sparse LS to $A(\mathcal{I}_k, \mathcal{J}_k)$:

$$\min_{M_k \in \mathcal{P}_k} \|A(\mathcal{I}_k, \mathcal{J}_k)M(\mathcal{J}_k) - e(\mathcal{I}_k)\|_2^2 = \min_{M_k \in \mathcal{P}_k} \left\|\widetilde{A}\widetilde{M}_k - \widetilde{e}_k\right\|_2^2$$

- Solve small LS problem by Householder *QR* for

$$A(\mathcal{I}_k, \mathcal{J}_k) \to \widetilde{M}_k \to M_k$$

Sparsity Pattern: Static A Priori Choice

- A^{-1} will be no more sparse!
- As a priori choice of a good approximate sparsity pattern for M we can choose the pattern of
  - A^k or $(A^T)^k$
  - $(A^T A)^k A^T$ for some $k = 1, 2$
  - A_ϵ with sparsified A
  - a combination of above
- A_ϵ by sparsification of A: delete all entries with $|A_{ij}| < \epsilon$

Sparsity Pattern: Dynamic Pattern Finding

- Start with thin approximate pattern $\mathcal{J}_k$ for M_k
- Compute optimal column $M_{k,\text{opt}}(\mathcal{J}_k)$ by least squares
- Find new entry j for M_k such that $M_{k,\text{opt}}(\mathcal{J}_k) + \lambda e_j$ has smaller residual in the Frobenius norm.

$$\begin{aligned}\min_{M_k \in \mathcal{P}_k} \left\| A(M_k + \lambda e_j) - e_k \right\|_2^2 &= \min_{M_k \in \mathcal{P}_k} \left\| (AM_k - e_k) + \lambda A e_j \right\|_2^2 = \\ &= \min \left(\|r_k\|_2^2 + 2\lambda (r_k^T A_j) + \lambda^2 \|A_j\|_2^2 \right)\end{aligned}$$

For each possible index candidate j compute

$$\lambda_j = -\frac{r_k^T A_j}{\|A_j\|_2^2}$$

Choose index j with $r_k^T A_j \neq 0$ and $j = \arg\min(\lambda_j)$ since

$$\min = \|r_k\|_2^2 - \frac{(r_k^T A_j)^2}{\|A_j\|_2^2}.$$

Error Estimates

- Assume that for all columns holds

$$\|r_k\| = \|AM_k - e_k\| < \epsilon$$

- Then

$$\begin{aligned} \|AM - I\|_F &\leq \sqrt{n}\epsilon \\ \|AM - I\|_2 &\leq \sqrt{n}\epsilon \\ \|AM - I\|_1 &\leq \sqrt{p}\epsilon \end{aligned}$$

with p being the maximum number of nonzero entries in r_k, $k = 1, \ldots, n$.

Overview Following SPAI Variants

1. Block SPAI
2. Iterative SPAI
3. Factorized SPAI (FSPAI)
4. Modified SPAI (MSPAI)

Further Variants of SPAI

1. Block SPAI

- Partition the matrix M in small blocks (e.g., 2×2 or 3×3) and apply the Frobenius norm minimization with blockwise pattern.
- Advantages:
  - Underlying block structure will also appear in the pattern of $A^{-1} \rightarrow$ improved pattern
  - Block operations are more efficient
  - Less least squares problems to solve

2. Iterative SPAI

- Start with pattern of $A \rightarrow M_1$
- Construct M_2 relative to new matrix AM_1
- Construct M_3 relative to new matrix AM_1M_2
- ...
- Advantage: cheaper (but inferior approximation)

3. Factorized SPAI, FSPAI

- Parallel Preconditioner for SPD matrices
- Approximate the inverse Cholesky factor of $A = L_A^T L_A$

$$A^{-1} = (L_A^T L_A)^{-1} = L_A^{-1} L_A^{-T} \approx L L^T$$

$$L_A^{-1} \approx L \qquad \rightarrow \qquad \min \| L_A L - I \|_F$$

- Normal equations give $A(\mathcal{J}_k, \mathcal{J}_k) L(\mathcal{J}_k) = \alpha e_k$

4. Modified SPAI, MSPAI: Combining Targeting and Probing in classical SPAI formulation.

Targeting (for MSPAI)

$$\min_{M \in \mathcal{P}} \|AM - T\|_F$$

- Assume, T is a good sparse preconditioner for A. Improve T by computing M and solving the above Frobenius norm minimization.
- Application: Assume that A is given by two parts, e.g. an advection part and a diffusion part.

  We can choose T as Laplacian relative to the diffusion part (easy to solve) and then we add M for improving T relative to the advection part.

Probing (for MSPAI)

- Find preconditioner of special form (tridiag, band) for preconditioning a matrix that is not given explicitly, but only by its action on certain (probing-)vectors, e.g.

$$e^T S = f^T.$$

  Example: S is Schur complement or general matrix.

- Choose, e.g. $e = (1, 1, \ldots, 1)^T$, $e = (1, -1, 1, -1, \ldots)^T$, $\ldots$

- Disadvantage: Can use only very special pattern for M and special probing vectors e.
  Example: tridiagonal probing

4. Modified SPAI: SPAI with Targeting and Probing

Generalize Frobenius norm minimization to

$$\min_{M\in\mathcal{P}} \|CM - B\|_F = \min_{M\in\mathcal{P}} \left\| \begin{pmatrix} C_0 \\ \rho u^T \end{pmatrix} M - \begin{pmatrix} B_0 \\ \rho v^T \end{pmatrix} \right\|_F$$

For example: original SPAI extended by an additional norm minimization to deliver especially good results on vector e:

$$\min_{M\in\mathcal{P}} \|CM - B\|_F = \min_{M\in\mathcal{P}} \left\| \begin{pmatrix} A \\ \rho e^T A \end{pmatrix} M - \begin{pmatrix} I \\ \rho e^T \end{pmatrix} \right\|_F$$

Outlook: Further Research

1. Apply MSPAI with the probing feature to multigrid or regularization problems (smoother, preconditioner).
   In connection with domain decomposition:
   - In the small domains use MSPAI as smoother
   - Use MSPAI as preconditioner for Schur complement

2. Find good block pattern
   - that reflects the physical connections
   - combines columns with very similar pattern to one LS problem

3. Factorized SPAI for indefinite matrices
   - Use a priori permutations (perfect matching)
   - and blocking

   to avoid breakdowns.

Available Code

- SPAI, MSPAI, FSPAI available in parallel version (MPI, C/C++)
  - SPAI (University of Basel):
    `http://www.computational.unibas.ch/software/spai/`
  - SPAI, MSPAI (TUM):
    `http://www5.in.tum.de/wiki/index.php/MSPAI`
  - FSPAI (TUM):
    `http://www5.in.tum.de/wiki/index.php/FSPAI`

    Publications on SPAI, MSPAI, FSPAI available on these websites!
- Block SPAI also available for GPU and shared memory (OpenMP, C/C++).
- High priority to port FSPAI to (parallel) block case.

Parallel Numerics, WT 2013/2014

6 Domain Decomposition

Contents

Concept of Domain Decomposition

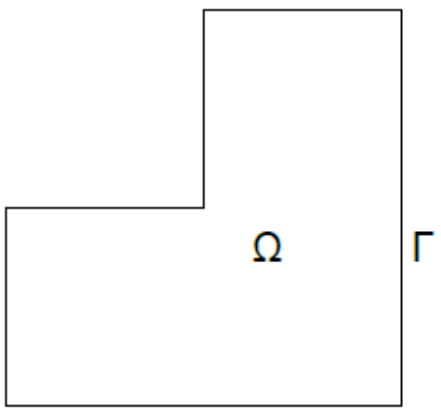
Consider elliptic PDE on region Ω with boundary Γ, e.g. with Dirichlet boundary conditions.

Example:

$$\Delta u = u_{xx} + u_{yy} = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = f(x,y) \in \Omega$$

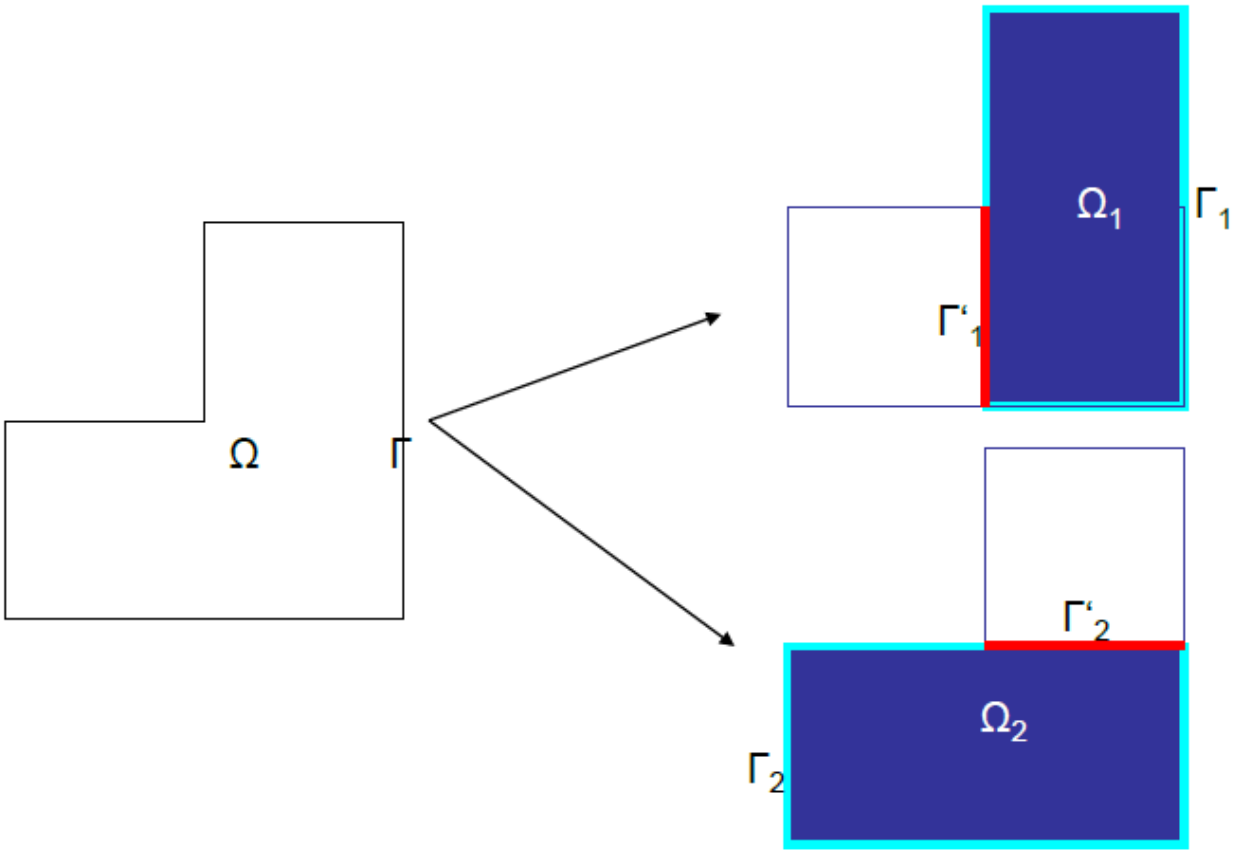
$$u(x,y)|_\Gamma = g(x,y) \in \Gamma$$

Parallel solution?



6.1. Overlapping Domain Decomposition

Partition region Ω in two regions Ω_1 and Ω_2, with new boundaries Γ_1 and Γ_2 which are partially given by old boundary Γ and some unknown parts Γ_1' and Γ_2' :

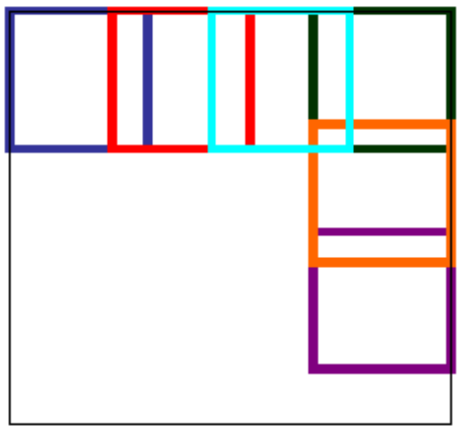


Overlapping Domain Decomposition (cont.)

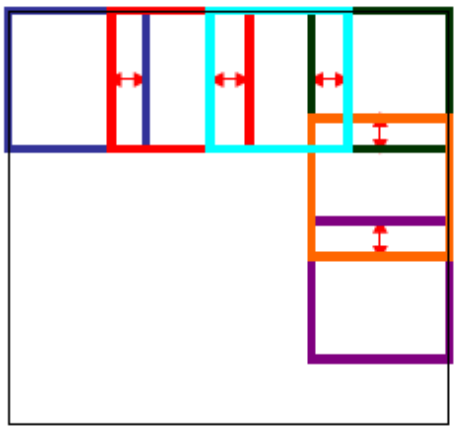
- We discretize and solve given PDE on Ω_1 with boundary Γ_1, but we need the values of $u(x, y)$ on new artificial boundary Γ_1'.
- First, we assume any values for Γ_1', e.g. $u(x, y) = 0$.
- Then we solve the linear system relative to region Ω_1.
- The same can be done in parallel for Ω_2.
- Values of resulting solution of 1) on Γ_2' can be used to compute solution in the next step for region Ω_2 with boundary Γ_2.
- In same way we use the resulting solution of 2) on Γ_1', to compute the solution for region Ω_1 with boundary Γ_1.
- So we generate solutions on partial regions which provide us with approximate values for unknown boundary values of the other partial solution.
- The sequence of solutions converges in each region against solution on Ω.

Overlapping Domain Decompos. (cont. 2)

1. Solve in parallel the PDE on all small subdomains with certain boundary cond.
2. Exchange boundary values



1st step



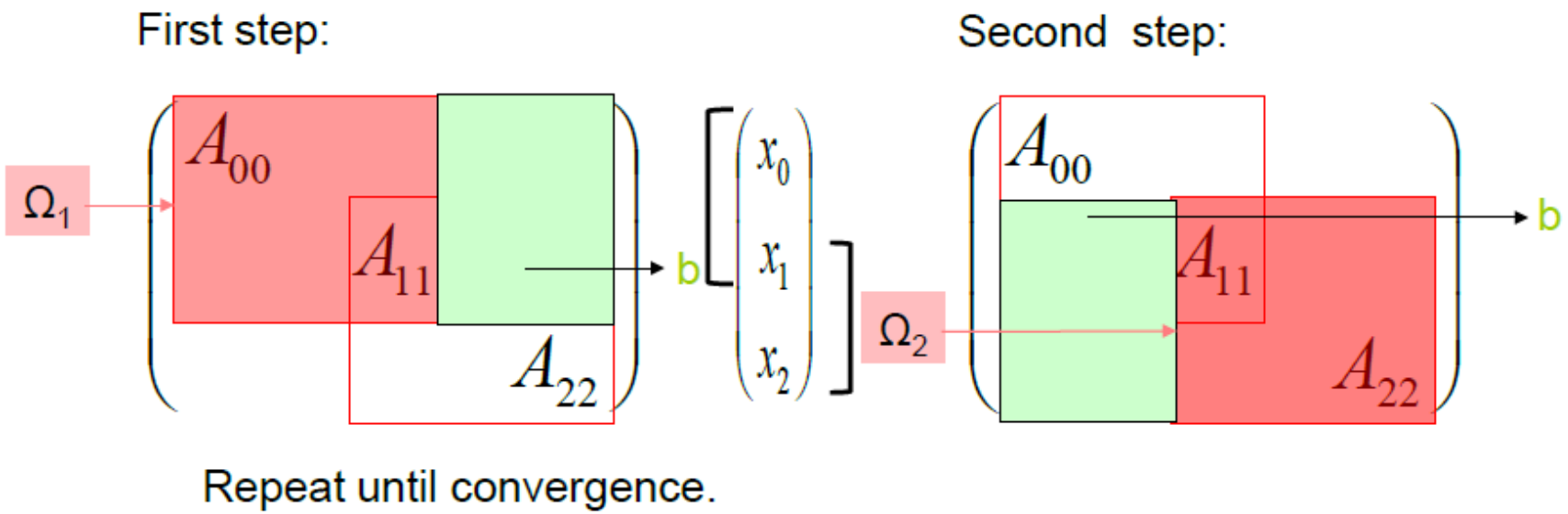
2nd step

3. Repeat until convergence.

For getting better initial values for interior boundary nodes: solve the whole problem on a coarse mesh and interpolate.

Overlapping Domain Decompos. (cont. 3)

Matrix representation of overlapping domain decomposition:



Repeat until convergence.

Green parts are related to the other domain and we asssume to know the related components in the vector *x* of unknowns. They are moved to the right-hand side *b*.