Other parallel approaches

- Approximate inverses
- Domain decomposition methods

Approximate inverse preconditioning

Some references

- Kolotilina, L. Yu.; Yeremin, A. Yu. Factorized sparse approximate inverse preconditionings. I. Theory. SIAM J. Matrix Anal. Appl. 14 (1993), 45-58
- Kolotilina, L. Yu.; Nikishin, A. A.; Yeremin, A. Yu. Factorized sparse approximate inverse preconditionings. IV. Simple approaches to rising efficiency. Numer. Linear Algebra Appl. 6 (1999), 515-531
- Grote, Marcus J.; Huckle, Thomas Parallel preconditioning with sparse approximate inverses. SIAM J. Sci. Comput. 18 (1997), 838-853.
- Benzi, Michele; Meyer, Carl D.; Tuma, Miroslav A sparse approximate inverse preconditioner for the conjugate gradient method. SIAM J. Sci. Comput. 17 (1996), 1135-1149
- Benzi, Michele; Tuma, Miroslav A sparse approximate inverse preconditioner for nonsymmetric linear systems. SIAM J. Sci. Comput. 19 (1998), 968-994
- Huckle, Thomas Approximate sparsity patterns for the inverse of a matrix and preconditioning. Appl. Numer. Math. 30 (1999), 291-303
- Bröker, Oliver; Grote, Marcus J. Sparse approximate inverse smoothers for geometric and algebraic multigrid. Appl. Numer. Math. 41 (2002), 61-80

Some references

- Broker, Oliver; Parallel multigrid methods using sparse approximate inverses. Thesis (Dr.sc.)ž013Eidgenoessische Technische Hochschule Zuerich (Switzerland). 2003. 169 pp
- Holland, Ruth M.; Wathen, Andy J.; Shaw, Gareth J. Sparse approximate inverses and target matrices. SIAM J. Sci. Comput. 26 (2005),1000-1011
- Jia, Zhongxiao; Zhu, Baochen A power sparse approximate inverse preconditioning procedure for large sparse linear systems. Numer. Linear Algebra Appl. 16 (2009),259-299
- Wang, Shun; de Sturler, Eric Multilevel sparse approximate inverse preconditioners for adaptive mesh refinement. Linear Algebra Appl. 431 (2009), 409-426
- Malas, Tahir; Gürel, Levent Accelerating the multilevel fast multipole algorithm with the sparse-approximate-inverse (SAI) preconditioning. SIAM J. Sci. Comput. 31 (2009), 1968-1984
- Neytcheva, M; Bängtsson, Erik; Linnér, Elisabeth Finite-element based sparse approximate inverses for block-factorized preconditioners. Adv. Comput. Math. 35 (2011), 323-355

Some references

- Thomas Huckle,
  Modified Sparse Approximate Inverses
  http://www5.in.tum.de/wiki/index.php/MSPAI
  Factorized Sparse Approximate Inverses
  http://www5.in.tum.de/wiki/index.php/FSPAI

Approximate inverses: Explicit Methods

Given a sparse matrix $A = [a_{ij}] \in^{n\times n}$.
Let S be a sparsity pattern. We want to compute $G \in \mathcal{S}$, such that

$$(GA)_{ij} = \delta_{ij}\,,\ (i,j) \in \mathcal{S},$$

i.e.

$$\sum_{k:(i,k)\in S} g_{ik}a_{kj} = \delta_{ij}\,,\ (i,j) \in \mathcal{S}.$$

Some observations:

- $\oplus$ the elements in the ith row of G can be computed independently;
- $\ominus$ even if A is symmetric, G is not necessarily symmetric, because g_{ij} and g_{ji} are, in general, not equal.

How does this work?

Choose $\mathcal{S}$ to be the tridiagonal part of A,
$\mathcal{S} = \{(1,1),(1,2),\{(i,i-1),(i,i),(i,i+1)\}_{i=1}^{n},(n,n-1),(n,n)\}$.
Then, when computing the ith row of G we need only the entries of the matrix A, namely,

$$A^i = \begin{bmatrix} a_{i-1,i-1} & a_{i-1,i} & a_{i-1,i+1} \\ a_{i,i-1} & a_{i,i} & a_{i,i+1} \\ a_{i+1,i-1} & a_{i+1,i} & a_{i+1,i+1} \end{bmatrix}$$

Given $A \in R^{n\times n}$ and $\mathcal{S}$
for i=1:n,
    Extract from A the small matrix A^i, needed to compute the entries of $G(i,:)$
    Solve with A^i
    Store row $G(i,:)$
end

For all rows, the steps can be performed fully in parallel!

Example:

We want to find G with the same sparsity pattern as A, i.e.,

$$A = \begin{bmatrix} 2 & -1 & 0 & 0 \\ -1 & 3 & -2 & 0 \\ 0 & -2 & 4 & -1 \\ 0 & 0 & -1 & 2 \end{bmatrix} \qquad G = \begin{bmatrix} g_{11} & g_{12} & 0 & 0 \\ g_{21} & g_{22} & g_{23} & 0 \\ 0 & g_{32} & g_{33} & g_{34} \\ 0 & 0 & g_{43} & g_{44} \end{bmatrix}$$

$$G(1,:): \quad \begin{aligned} 2g_{11} - g_{12} &= 1 \\ -g_{11} + 3g_{12} &= 0 \end{aligned} \qquad G(2,:): \quad \begin{aligned} 2g_{21} - g_{22} &= 0 \\ -g_{21} + 3g_{22} - 2g_{23} &= 1 \\ -2g_{22} + 4g_{23} &= 0 \end{aligned}$$

Example, cont.

$$A^{-1} = \frac{1}{19}\begin{bmatrix} 13 & 7 & 4 & 2 \\ 7 & 14 & 8 & 4 \\ 4 & 8 & 10 & 5 \\ 2 & 4 & 5 & 12 \end{bmatrix}$$

$$G = \begin{bmatrix} \frac{3}{5} & \frac{1}{5} & 0 & 0 \\ \frac{1}{3} & \frac{2}{3} & \frac{1}{3} & 0 \\ 0 & \frac{4}{13} & \frac{6}{13} & \frac{3}{13} \\ 0 & 0 & \frac{1}{7} & \frac{4}{7} \end{bmatrix} \qquad GA = \begin{bmatrix} 1 & 0 & -0.40 & 0 \\ 0 & 1 & 0 & -0.33 \\ -0.31 & 0 & 1 & 0 \\ 0 & -0.28 & 0 & 1 \end{bmatrix}$$

Example, cont.

Note: the second row of G is the second row of the matrix

$$B = \begin{bmatrix} 2 & -1 & 0 & 0 \\ -1 & 3 & -2 & 0 \\ 0 & -2 & 4 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}^{-1}, \; GB = \begin{bmatrix} 1 & 0 & -0.4 & 0 \\ 0 & 1 & 0 & 0 \\ -0.31 & 0 & 1.2308 & 0.4615 \\ 0 & -0.2857 & 0.5714 & 1.1429 \end{bmatrix}.$$

However, if we compute AG then

$$AG = \begin{bmatrix} 0.8667 & -0.2667 & -0.3333 & 0 \\ 0.4000 & 1.1846 & 0.0769 & -0.4615 \\ -0.6667 & -0.1026 & 1.0366 & 0.3516 \\ 0 & -0.3077 & -0.1758 & 0.9121 \end{bmatrix}$$

i.e., the matrix G is computed as a left-side approximate inverse of A and as such is somewhat less accurate than as a right-side approximate inverse.
The drawback of the above method is that
in general even if A is symmetric, G is not!

Implicit Methods

Let A be in a factored form.

Suppose $A = LD^{-1}U$ is a triangular matrix factorization of A. If A is a band matrix then L and U are also band matrices.

Let $\boxed{L = I - \widetilde{L};\ U = I - \widetilde{U},}$ where $\widetilde{L}$ and $\widetilde{U}$ are strictly lower and upper triangular matrices correspondingly.

Lemma 1 *Using the above notations it can be shown that*
(i) $A^{-1} = DL^{-1} + \widetilde{U}A^{-1}$, *(ii)* $A^{-1} = U^{-1}D + A^{-1}\widetilde{L}$.

Proof

$$A = LD^{-1}U \Longrightarrow A^{-1} = U^{-1}DL^{-1}$$

$$\Longrightarrow (I - \widetilde{U})A^{-1} = DL^{-1} \Longrightarrow A^{-1} = DL^{-1} + \widetilde{U}A^{-1}.$$

Also

$$A^{-1}(I - \widetilde{L}) = U^{-1}D \Longrightarrow A^{-1} = U^{-1}D + A^{-1}\widetilde{L}.$$

■

Algorithm to compute A^{-1}

for $r = n, n-1, \cdots, 1$

$$(A^{-1})_{r,r} = D_{r,r} + \sum_{s=1}^{min(q,n-r)} \widetilde{U}_{r,r+s}(A^{-1})_{r+s,r}$$

for $k = 1, 2, \cdots, q$

$$(A^{-1})_{r-k,r} = \sum_{s=1}^{min(q,n-r+k)} \widetilde{U}_{r-k,r-k+s}(A^{-1})_{r-k+s,r} \rightsquigarrow (i)$$

$$(A^{-1})_{r,r-k} = \sum_{t=1}^{min(q,n-r+k)} (A^{-1})_{r,r-k+t}\widetilde{L}_{r-k+t,r-k} \rightsquigarrow (ii)$$

endfor

endfor

q is the bandwidth.

A drawback:

Consider an spd matrix

$$A = \begin{bmatrix} 1 & -2 & 1 \\ -2 & 5 & -3 \\ 1 & -3 & 4 \end{bmatrix}. \quad \text{Then} \quad A_{band} = \begin{bmatrix} 1 & -2 & 0 \\ -2 & 5 & -3 \\ 0 & -3 & 4 \end{bmatrix}$$

is indefinite.

A general framework for computing approximate inverses

Frobenius norm minimization

$$||A||_I = \sqrt{\sum_{i=1}^{n}\sum_{i=1}^{n} a_{ij}^2} = \sqrt{tr(AA^H)}$$

Let a sparsity pattern $\mathcal{S}$ be given. Consider the functional

$$F_W(G) = ||I - GA||_W^2 = tr(I - GA)W(I - GA)^T,$$

where the weight matrix W is spd If $W \equiv I$ then $||I - GA||_I$ is the Frobenius norm of $I - GA$.
Clearly $F_W(G) \geq 0$. If $G = A^{-1}$ then $F_W(G) = 0$. Hence, we want to compute the entries of G in order to minimize $F_W(G)$, i.e. to find $\hat{G} \in S$, such that

$$||I - \hat{G}A||_W \leq ||I - GA||_W, \ \forall G \in S.$$

The following properties of $tr(\cdot)$ will be used:

$$trA = trA^T, \ tr(A + B) = trA + trB.$$

$$
\begin{aligned}
F_W(G) &= tr(I - GA)W(I - GA)^T \\
&= tr(W - GAW - W(GA)^T + GAW(GA)^T) \qquad (1)\\
&= trW - trGAW - tr(GAW)^T + trGAWA^TG^T.
\end{aligned}
$$

Minimize F_W w.r.t. G, consider the entries $g_{i,j}$ as variables. The necessary condition for a minimizing point are

$$
\frac{\partial F_W(G)}{\partial g_{ij}} = 0, \ (i,j) \in \mathcal{S}. \qquad (2)
$$

From (1) and (2) we get $\boxed{-2(WA^T)_{ij} + 2(GAWA^T)_{ij} = 0,}$ or

$$
(GAWA^T)_{ij} = (WA^T)_{ij}, \ (i,j) \in \mathcal{S}. \qquad (3)
$$

The equations (3) may or may not have a solution, depending on the particular matrix A and the choice of $\mathcal{S}$ and W.

Choices of W:

Choise 1: Let A be spd Choose $W = A^{-1}$ which is also spd

$$\Longrightarrow (GA)_{ij} = \delta_{ij}\,, \ (i, j) \in S,$$

i.e. the formula for the explicit method can be seen as a special case of the more general framework for computing approximate inverses using weighted Frobenius norms.

Choise 2: Let $W = (A^T A)^{-1}$.

$$\Longrightarrow (G)_{ij} = (A^{-1})_{ij}\,, \ (i, j) \in S,$$

which is the formula for the implicit method. In this case the entries of G are the corresponding entries of the exact inverse.

Improvement via diagonal compensation

Let A be symmetric and five-diagonal. Suppose we know that the two of the off-diagonals contain small entries. Such matrix appears if we solve the anisotropic problem, for instance:

$$-\frac{\partial^2 u}{\partial x^2} - \varepsilon \frac{\partial^2 u}{\partial y^2} = f,$$

where $\varepsilon > 0$ is small.
We choose a tridiagonal sparsity pattern $\mathcal{S}_3$ for G, where the the two nonzero off-diagonals will correspond to the off-diagonals of A, containing bigger elements, i.e. they are not necessarily next to the main diagonal. Then we construct an approximate inverse in the following way:

Step 1: Let $\widetilde{A}$ be A with deleted small entries, i.e. $\widetilde{A} \in \mathcal{S}_3$.

Step 2: Compute $\widetilde{G}$: $(\widetilde{G}A)_{ij} = \delta_{ij}, \ (i,j) \in S_3$.

Step 3: Find $G = \bar{G} + D$, where $\bar{G} = \frac{1}{2}(\widetilde{G} + \widetilde{G}^T)$ and D is diagonal, computed from the following imposed condition on G, i.e.

$$GA\mathbf{e} = \mathbf{e},$$

and $\mathbf{e} = (1, 1, \cdots, 1)^T$.

The diagonal compensation technique prescribes the spd property of A.

Constructing an spd approximate inverse

The methods described till now do not guarantee that G will be such a matrix.

We want now to compute an spd approximate inverse of an spd matrix.

Let $\mathcal{S}$ be a symmetric sparsity pattern. We seek G of the form

$$G = L_G^T L_G, L_G \in \mathcal{S}_L.$$

Clearly G will be spd

Theorem 1 *A matrix G of the form $G = L_G^T L_G$ which is an spd approximation of A^{-1} can be computed from the following relation:*

$$min_{X \in \mathcal{S}_L} \frac{\frac{1}{n} tr X A X^T}{(det(X A X^T))^{\frac{1}{n}}} = \frac{\frac{1}{n} tr L_G A L_G^T}{\left(det(L_G A L_G^T)\right)^{\frac{1}{n}}}. \quad (4)$$

Proof:

$X \in \mathcal{S}_L$ is lower triangular. Let $X = D(I - \widetilde{X})$, where $\widetilde{X} \in \mathcal{S}_{\widetilde{L}}$ is strictly lower triangular. Then $\widetilde{X} = I - D^{-1}X$. Let denote also $D = diag(d_1, d_2, \cdots, d_n)$. Then

$$\frac{\frac{1}{n} tr XAX^T}{(det(XAX^T))^{\frac{1}{n}}} = \frac{\frac{1}{n}\sum_i (XAX^T)_{ii}}{(det(X)^2 det(A))^{\frac{1}{n}}}$$

$$= \frac{\frac{1}{n}\sum_i \left(D(I-\widetilde{X})A(I-\widetilde{X})^T D\right)_{ii}}{(det(X)^2 det(A))^{\frac{1}{n}}} = \frac{\frac{1}{n}\sum_i d_i^2 \left((I-\widetilde{X})A(I-\widetilde{X})^T\right)_{ii}}{\left(\prod_i d_i^2\right)^{\frac{1}{n}} (det(A))^{\frac{1}{n}}}$$

$$= \frac{\frac{1}{n}\sum_i \alpha^2}{\left(\prod_i \alpha^2\right)^{\frac{1}{n}}} \cdot \frac{\left(\prod_i ((I-\widetilde{X})A(I-\widetilde{X})^T)_{ii}\right)^{\frac{1}{n}}}{(det(A))^{\frac{1}{n}}} \qquad (5)$$

$$= Expression\_A \cdot Expression\_B.$$

In the above notations $\alpha_i^2 = d_i^2 \left((I - \widetilde{X})A(I - \widetilde{X})^T \right)_{ii}$.

$Expression\_B$ does not depend on d_i. The problem of minimizing $Expression\_B$ is a particular case of the already considered problem of minimizing the functional $F_W(G)$ with a special choice of the corresponding matrices - $W = A,\ A = I,\ G = \widetilde{X}$. In other words, the solution of the problem

$$min_{\widetilde{X} \in S_{\widetilde{L}}} \prod_i \left((I - \widetilde{X})A(I - \widetilde{X})^T \right)_{ii} = min_{\widetilde{X} \in S_{\widetilde{L}}} tr(I - \widetilde{X})A(I - \widetilde{X})^T \quad (6)$$

will be also the solution of minimizing $Expression\_B$.

Further, $Expression\_A \geq 1,\ \forall \alpha$, being the ratio of the arithmetic and geometric mean, and takes the value 1 when $\alpha_i^2 = 1$.

Thus, we minimize $Expression\_A$ computing

$$d_i = \frac{1}{\left((I - \widetilde{X})A(I - \widetilde{X})^T\right)_{ii}^{\frac{1}{2}}}. \tag{7}$$

Let $\widetilde{L}_G$ be the solution of (7). Note that it is strictly lower triangular. Let the entries d_i of D are computed from the relations (7), where instead of $\widetilde{X}$ $\widetilde{L}_G$ is used. Then the matrix $L_G^T L_G$, where $L_G = D(I - \widetilde{L}_G)$, will be the searched approximation of A^{-1}:

- $(L_G A L_G^T)_{ii} = 1$ by construction;
- The equality (4) gives a measure of the quality of the approximate inverse constructed (the K-condition number (Igor Kaporin).

Let $A = tridiag(-1, 4, -1)$. Find $L_G^T L_G$ - an approximate inverse of A, where L_G is bidiagonal. Thus, $\mathcal{S}_{\widetilde{L}} = \{\{(i-1, i)\}_{i=2}^n\}$.

First we compute a strictly lower bidiagonal matrix $\widetilde{L}$ from the condition

$$(\widetilde{L}A)_{i,j} = (A)_{i,j}, \; i, j \in \mathcal{S}_{\widetilde{L}},$$

which gives us

$$\widetilde{L} = \begin{bmatrix} 0 & & & & & \\ \frac{1}{4} & 0 & & & & \\ & \frac{1}{4} & 0 & & & \\ & & & \ddots & & \\ & & & \frac{1}{4} & 0 & \\ & & & & \frac{1}{4} & 0 \end{bmatrix}.$$

Then d_i are found to be

$$d_1 = \frac{1}{2}, d_i = \frac{2}{\sqrt{15}}, \; i = 1, 2, \cdots, n.$$

$$L_G = \begin{bmatrix} \frac{1}{2} & 0 & \cdots & 0 & 0 \\ \frac{1}{2\sqrt{15}} & \frac{2}{\sqrt{15}} & \cdots & 0 & 0 \\ & & \ddots & & \\ & & & \ddots & \\ 0 & 0 & \cdots & \frac{1}{2\sqrt{15}} & \frac{2}{\sqrt{15}} \end{bmatrix}, \quad L_G^T L_G = \begin{bmatrix} \frac{4}{15} & \frac{1}{15} & 0 & \cdots & 0 \\ \frac{1}{15} & \frac{17}{60} & \frac{1}{15} & \cdots & 0 \\ & & \ddots & & \\ & & & \ddots & \\ 0 & \cdots & & \frac{1}{15} & \frac{17}{60} \end{bmatrix},$$

and

$$L_G^T L_G A = \begin{bmatrix} 1 & 0 & -\frac{1}{15} & \cdots & 0 \\ \frac{7}{15} & 1 & \frac{7}{60} & \cdots & 0 \\ & & \ddots & & \\ & & & \ddots & \\ 0 & \cdots & & \frac{7}{60} & 1 \end{bmatrix}.$$

Extensions

- When minimizing $||I - AG||_F$, minimize the 2-norm of each column separately, $||\mathbf{e}_k - A\mathbf{g}_k||_F, k = 1, \cdots, n$
- use adaptive $\mathcal{S}$ (much more expensive)
- used the sparsity pattern of powers of A
- Modified SPAI: combines
  - Frobenius norm minimization
  - MILU
  - vector probing

MSPAI

Consider the formulation:

$$\min_G \|CG - B\|_F = \min_G \left\| \begin{bmatrix} C \\ \rho \mathbf{e}^T C \end{bmatrix} G - \begin{bmatrix} B_0 \\ \rho \mathbf{e}^T B_0 \end{bmatrix} \right\|_F$$

$\rho = 0, C_0 = A, B_0 = I$ - the original form
$C_0 = I, B_0 = A$ - explicit approximation of A
$\rho = [1, 1, \cdots, 1]$ - MILU
Improve existing approximations:

$$\min_U \left\| \begin{bmatrix} L \\ \rho \mathbf{e}^T L \end{bmatrix} U - \begin{bmatrix} A \\ \rho \mathbf{e}^T A \end{bmatrix} \right\|_F$$

Finite element setting:

$$A = \sum_{k=1}^{M} R_k^T A_k R_k,$$

with R_k being the Boolean matrices which prescribe the local-to-global correspondence of the numbered degrees of freedom.
Is this of interest?

$$B^{-1} = \sum_{k=1}^{M} R_k^T A_k^{-1} R_k.$$

B^{-1} and A^{-1} are spectrally equivalent, namely, for some $0 < \alpha_1 < \alpha_2$ there holds

$$\alpha_1 A_{11}^{-1} \leq B_{11}^{-1} \leq \alpha_2 A_{11}^{-1},$$

Finite element setting:

Consider spd matrices.

$$\min_M(\lambda_{min}(A_k)) \leq \lambda(A) \leq p \max_M(\lambda_{max}(A_k)),$$

where p is the maximum degree of the graph representing the discretization mesh. Similarly, there holds

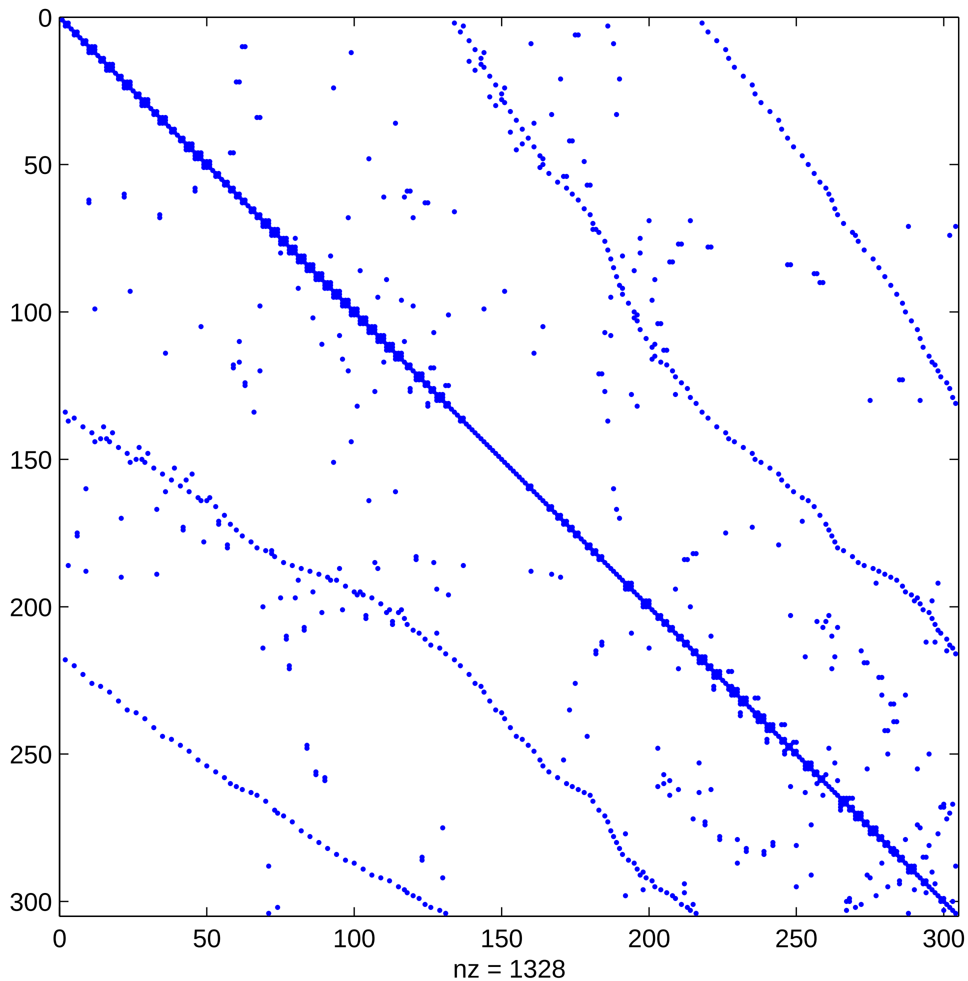
$$\min_M(\lambda_{min}(A_k)^{-1}) \leq \lambda(B^{-1}) \leq p \max_M(\lambda_{max}(A_k)^{-1}).$$

Then we obtain

$$\frac{\min(\lambda_{min}(A_k))}{\max(\lambda_{max}(A_k))} \leq \frac{\mathbf{x}^T B^{-1}\mathbf{x}}{\mathbf{x}^T A^{-1}\mathbf{x}} \leq \frac{\max(\lambda_{max}(A_k))}{\min(\lambda_{min}(A_k))}$$

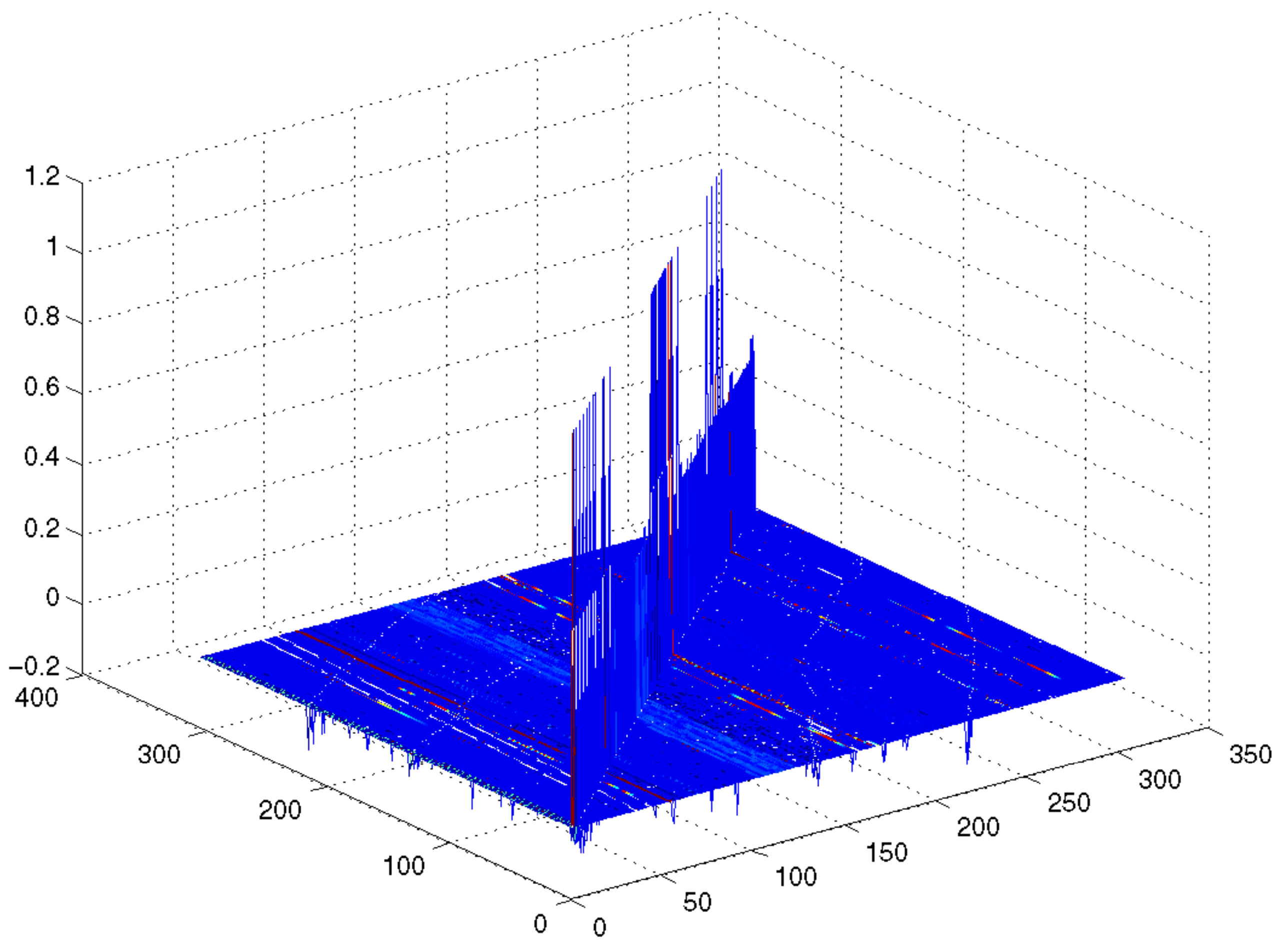
Thus, the spectral equivalence constants do not depend on the mesh parameter h but they are in general robust neither with respect to problem and mesh-anisotropies, nor to jumps in the problem coefficients as the eigenvalues of A_k depend on those.

FEM-SPAI



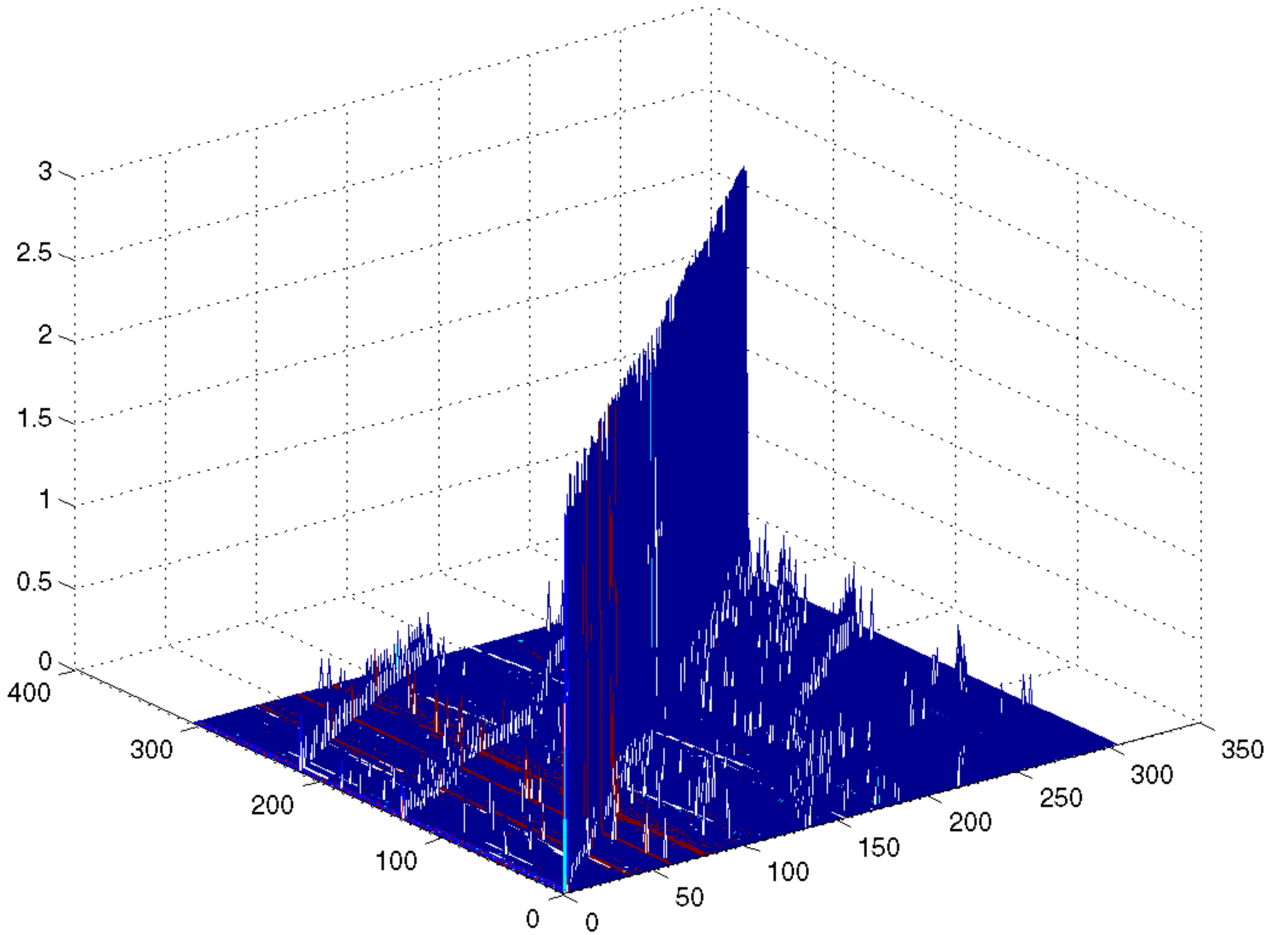
The matrix itself (`spy(A)`)

FEM-SPAI



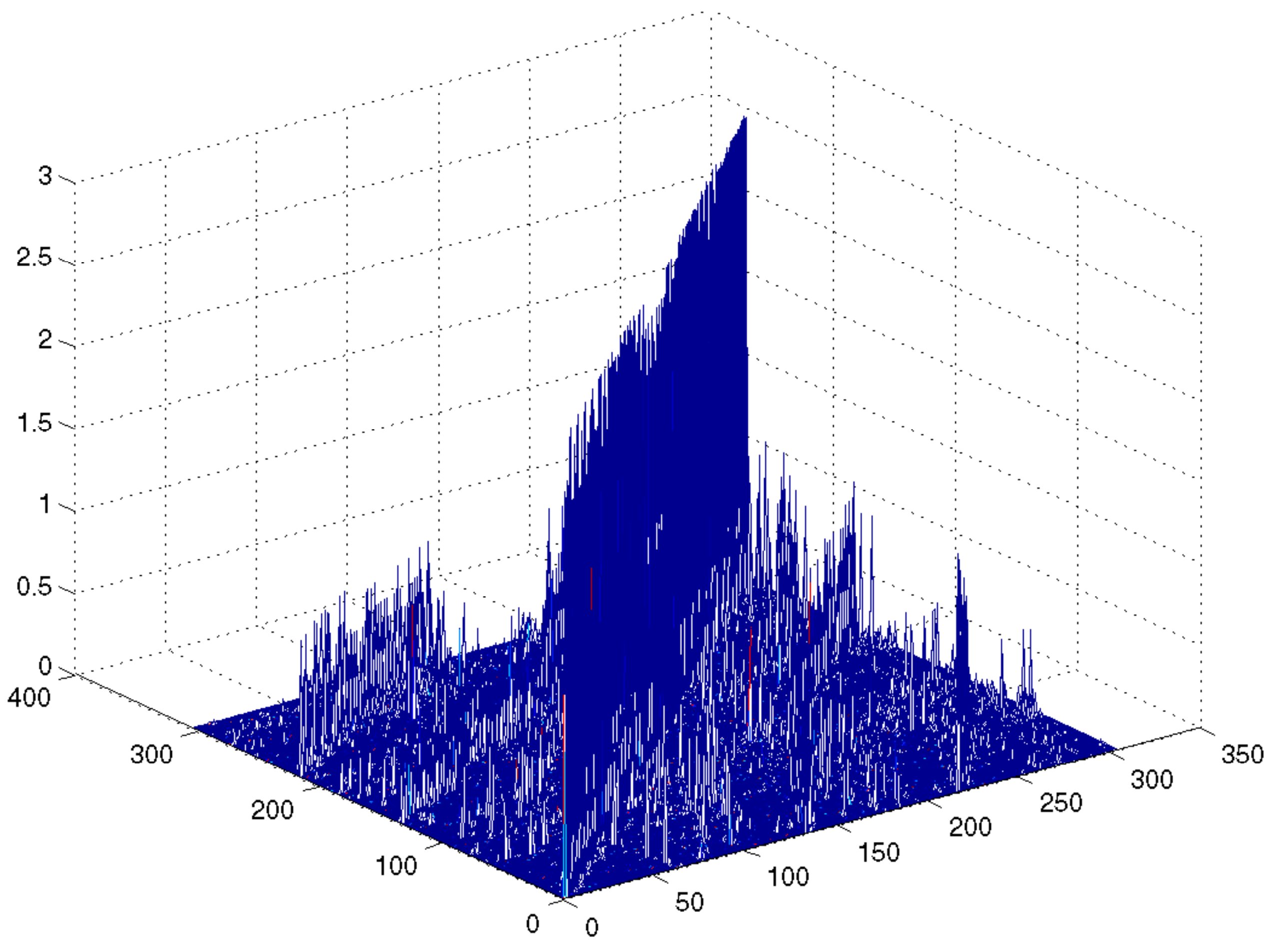
The matrix itself (`mesh(A)`)

FEM-SPAI



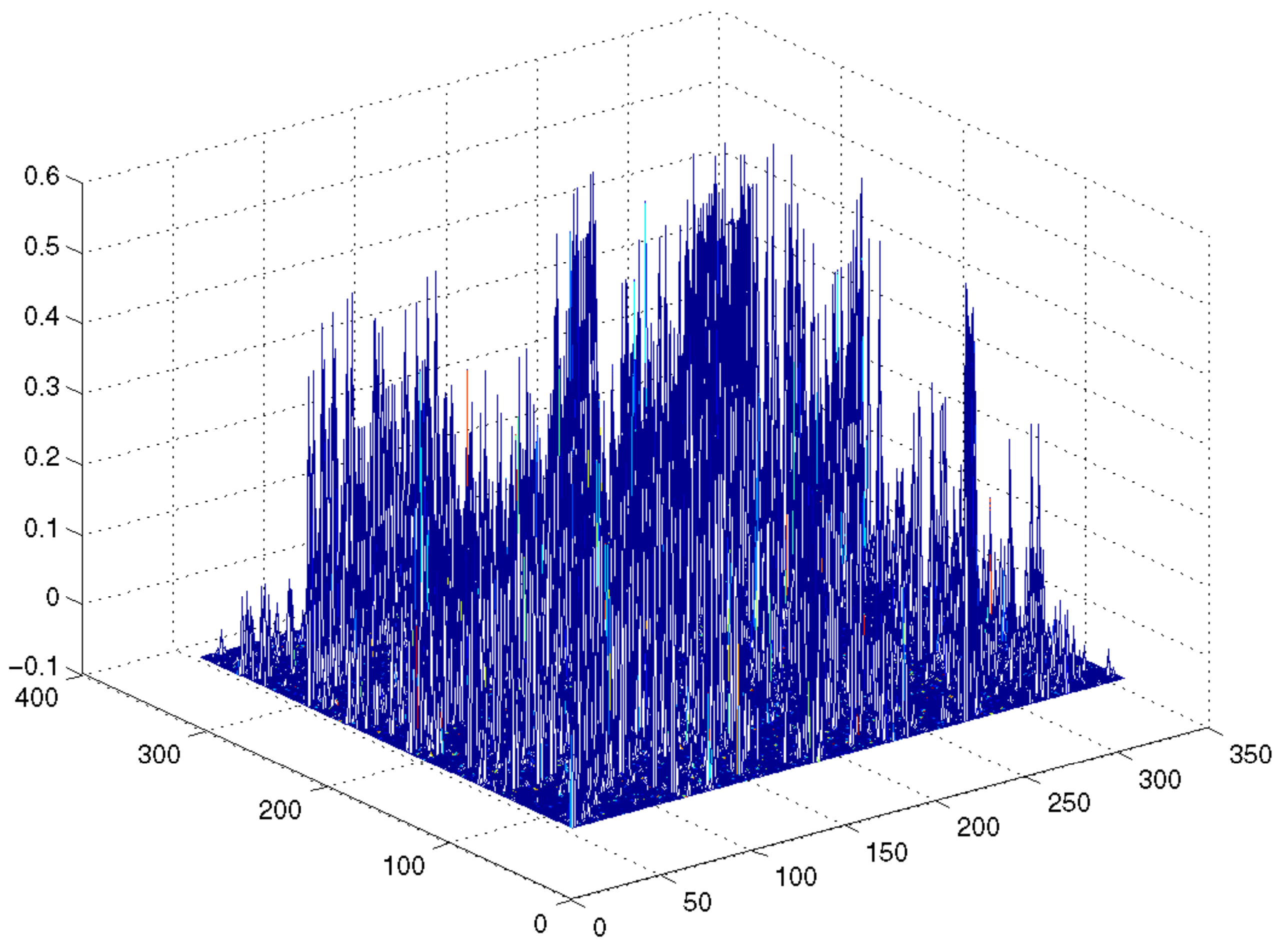
The approximate inverse (`mesh(AI)`)

FEM-SPAI



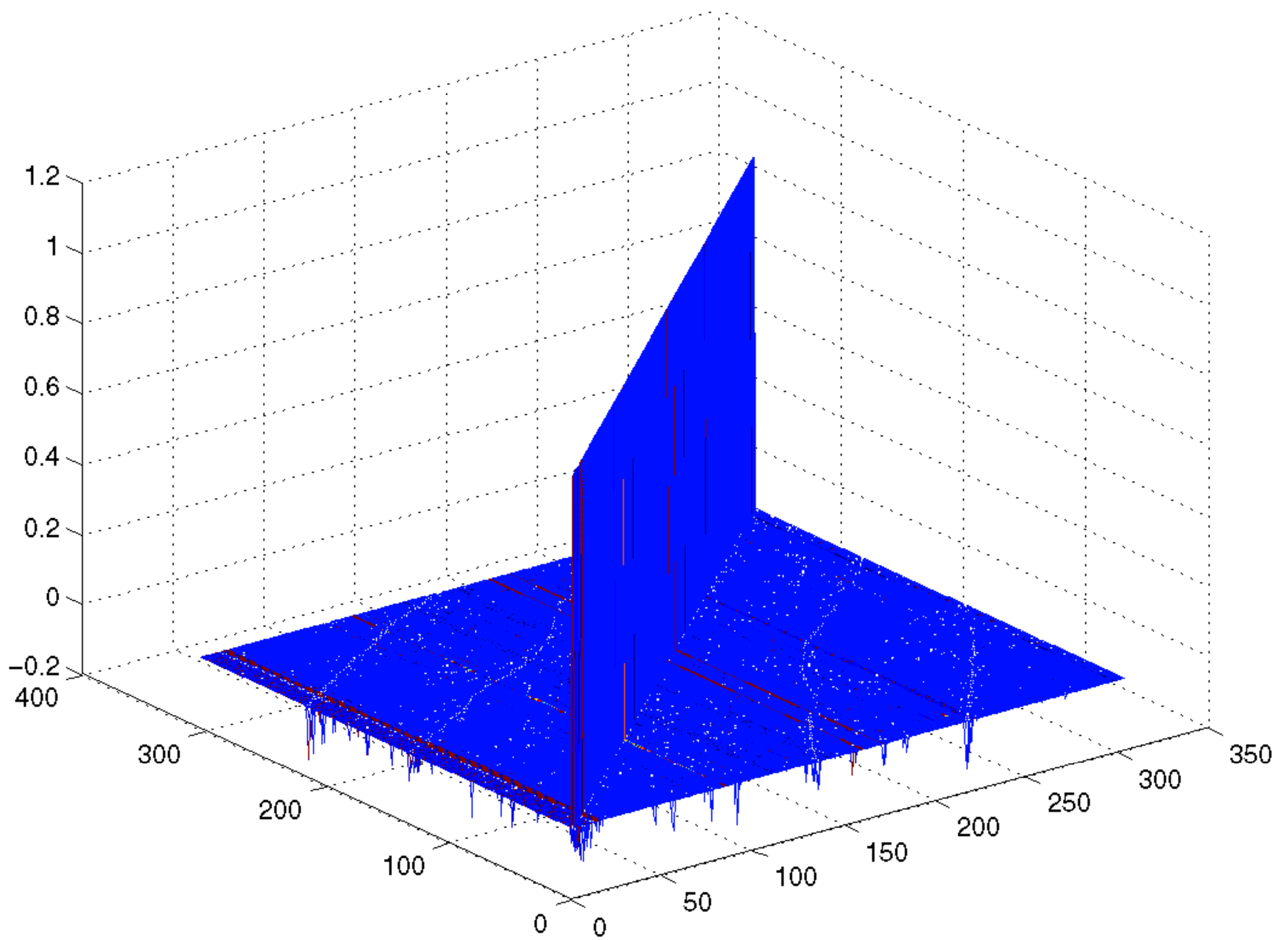
The exact inverse matrix (`mesh(inv(A))`)

FEM-SPAI



The difference (`mesh(inv(A)-AI)`)

FEM-SPAI



The product with A (`mesh(AI*A)`)

FEM-SPAI: Scalability figures: Constant problem size

$\#proc$	n_{fine}	$t_{B_{11}^{-1}}/t_A$	t_{repl} [s]	$t_{solution}$ [s]	$\#\,iter$
4	197129	0.005	0.28	7.01	5
16	49408	0.180	0.07	0.29	5
64	12416	0.098	0.02	0.03	5

Problem size: 787456
Solution method: *PCG*
Relative stopping criterium: $< 10^{-6}$

FEM-SPAI: Scalability figures: Constant load per processor

$\#proc$	$t_{B_{11}^{-1}}/t_A$	t_{repl} [s]	$t_{solution}$ [s]	$\# \, iter$
1	0.0050	-	0.17	5
4	0.0032	0.28	7.01	5
16	0.0035	0.24	4.55	5
64	0.0040	0.23	12.43	5

Local number of degrees of freedom: 197129
Solution method: *PCG*
Relative stopping criterium: $< 10^{-6}$

Domain decomposition methods

Many different interpretations within the PDE community:

- Parallel computing: data decomposition (independent of numerical method)
- Asymptotic analysis: separation of physical domain into regions with possibly different models
- Preconditioning methods: solution of a large linear system arising from the discretization of the PDE on the whole domain by DDM as a solver or a preconditioner:
  - Overlapping domain decomposition
  - Non-overlapping domain decomposition

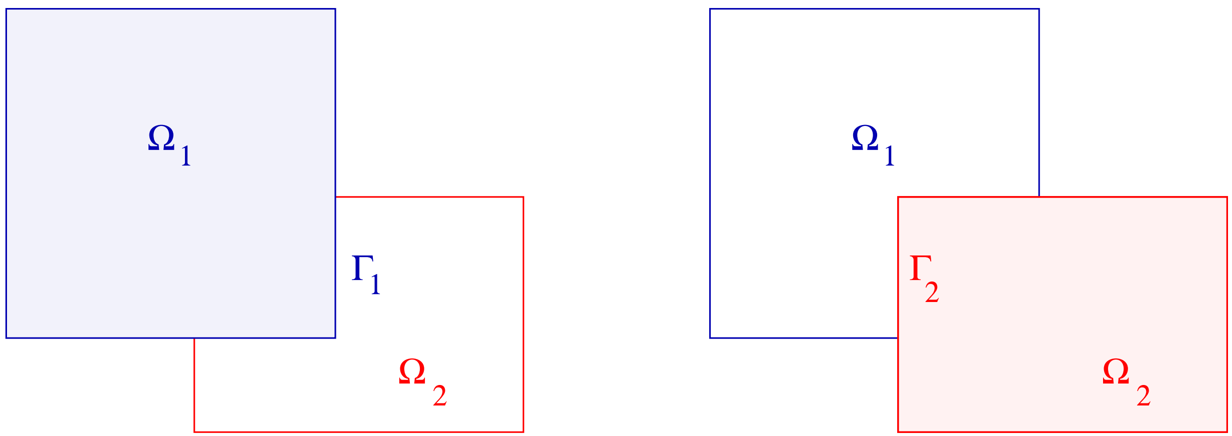
Domain decomposition:

Domain decomposition - decomposition of the spatial domain into several subdomains. Search the global true solution through (iteratively) solving subproblems while enforcing suitable continuity requirements between neighbor subdomains.

- Flexible - localized treatment of complex and irregular geometries, singularities etc.
- Efficient - often optimal convergence rate
- Easy to parallelize (coarse grain parallelization)

Schwarz 1870 (alternating method)

$A\mathbf{u} = \mathbf{f}$



Matrix form of Alternating Schwarz

Decompose A as $A_i A_{\partial\Omega_i\setminus\Gamma_i} A_{\Gamma_i}$.
Let $I_{\Omega_i\to\Gamma_j}$ be the discrete operator that interpolates the nodes in the interior of Ω_i to Γ_j. Then:

$$\begin{aligned} A_{\Omega_1}\mathbf{u}^k_{\Omega_1} &= \mathbf{f}_1 - A_{\Gamma_1} I_{\Omega_2\to\Gamma_1}\mathbf{u}^{k-1}_{\Omega_2} \\ A_{\Omega_2}\mathbf{u}^k_{\Omega_2} &= \mathbf{f}_2 - A_{\Gamma_2} I_{\Omega_1\to\Gamma_2}\mathbf{u}^{k}_{\Omega_1} \end{aligned}$$

Gauss-Seidel method for the system

$$\begin{bmatrix} A_{\Omega_1} & A_{\Gamma_1} I_{\Omega_2\to\Gamma_1} \\ A_{\Gamma_2} I_{\Omega_1\to\Gamma_2} & A_{\Omega_2} \end{bmatrix} \begin{bmatrix} \mathbf{u}_{\Omega_1} \\ \mathbf{u}_{\Omega_2} \end{bmatrix} = \begin{bmatrix} \mathbf{f}_1 \\ \mathbf{f}_2 \end{bmatrix}$$

Rearrange as a simple iteration:

$$\mathbf{u}^k_{\Omega_1} = \mathbf{u}^{k-1}_{\Omega_1} + A^{-1}_{\Omega_1}(\mathbf{f}_1 - A_{\Omega_1}\mathbf{u}^{k-1}_{\Omega_1} - A_{\Gamma_1} I_{\Omega_2\to\Gamma_1}\mathbf{u}^{k-1}_{\Omega_2})$$

$$\mathbf{u}^k_{\Omega_2} = \mathbf{u}^{k-1}_{\Omega_2} + A^{-1}_{\Omega_2}(\mathbf{f}_2 - A_{\Omega_2}\mathbf{u}^{k-1}_{\Omega_2} - A_{\Gamma_1} I_{\Omega_1\to\Gamma_2}\mathbf{u}^{k}_{\Omega_2})$$

Additive and multiplicative Schwarz methods:

$$\mathbf{u}^k_{\Omega_1} = \mathbf{u}^{k-1}_{\Omega_1} + A^{-1}_{\Omega_1}(\mathbf{f}_1 - A_{\Omega_1}\mathbf{u}^{k-1}_{\Omega_1} - A_{\Omega\setminus\overline{\Omega}_1}\mathbf{u}^{k-1}_{\Omega\setminus\overline{\Omega}_1})$$

$$\mathbf{u}^k_{\Omega_2} = \mathbf{u}^{k-1}_{\Omega_2} + A^{-1}_{\Omega_2}(\mathbf{f}_2 - A_{\Omega_2}\mathbf{u}^{k-1}_{\Omega_2} - A_{\Omega\setminus\overline{\Omega}_2}\mathbf{u}^{k-1}_{\Omega\setminus\overline{\Omega}_2})$$

$$\Uparrow$$

$$\mathbf{u}^k_{\Omega\setminus\overline{\Omega}_2}$$

For the whole system: two-step algorithm

$$\mathbf{u}^{k+1/2} = \mathbf{u}^k + \begin{bmatrix} A_{\Omega_1}^{-1} & 0 \\ 0 & 0 \end{bmatrix} (\mathbf{f} - A\mathbf{u}^k)$$

$$\mathbf{u}^{k+1} = \mathbf{u}^{k+1/2} + \begin{bmatrix} 0 & 0 \\ 0 & A_{\Omega_2}^{-1} \end{bmatrix} (\mathbf{f} - A\mathbf{u}^{k+1/2})$$

Final form

Denote: $\mathbf{u}_{\Omega_1} = R_1\mathbf{u} = \begin{bmatrix} I & 0 \end{bmatrix} \begin{bmatrix} \mathbf{u}_{\Omega_1} \\ \mathbf{u}_{\Omega\setminus\Omega_1} \end{bmatrix}$

$\mathbf{u}_{\Omega_2} = R_2\mathbf{u} = \begin{bmatrix} 0 & I \end{bmatrix} \begin{bmatrix} \mathbf{u}_{\Omega\setminus\Omega_2} \\ \mathbf{u}_{\Omega_2} \end{bmatrix}$

Then $A_{\Omega_i} = R_i A R_i^T$; let $B_i = R_i^T (R_i A R_i^T)^{-1} R_i$.

$$\mathbf{u}^{k+1} = \mathbf{u}^k + (B_1 + B_2 - B_2 A B_1)(\mathbf{f} - A\mathbf{u}^k)$$

$$\mathbf{u}^{k+1} = \mathbf{u}^k + (B_1 + B_2)(\mathbf{f} - A\mathbf{u}^k)$$

Final form, many subdomains

$$\mathbf{u}^{k+1} = \mathbf{u}^k + \sum_{i=1}^{p} B_i(\mathbf{f} - A\mathbf{u}^k)$$

$$\begin{aligned}
\mathbf{u}^{k+1/p} &= \mathbf{u}^k + B_1(\mathbf{f} - A\mathbf{u}^k) \\
\mathbf{u}^{k+2/p} &= \mathbf{u}^{k+1/p} + B_2(\mathbf{f} - A\mathbf{u}^{k+1/p}) \\
&\vdots \\
\mathbf{u}^{k+1} &= \mathbf{u}^{k+(p-1)/p} + B_p(\mathbf{f} - A\mathbf{u}^{k+(p-1)/p})
\end{aligned}$$

$$\mathbf{u}^{k+1} = \mathbf{u}^k + (I - (I - B_p A(\cdots(I - B_A))))A^{-1}(\mathbf{f} - A\mathbf{u}^k)$$

Problem:

Too slow Convergence deteriorates as p increases (H decreases).
Reason: The only global communication of information between subdomains are through overlapping regions. **Too slow!**
How to speed up? **Coarse grid correction.**

$$\mathbf{u}^{fine} = \mathbf{u}^{fine} + R^T A_C^{-1} R(\mathbf{f} - A\mathbf{u}^{fine})$$

Two-level additive Schwarz method:

$$\mathbf{u}^{k+1} = \mathbf{u}^k + \left(R^T A_C^{-1} R + \sum_{i=1}^{p} R^T A_i^{-} R \right) \mathbf{r}^k$$

Final keywords

- multiplicative is faster than additive
- overlapping or nonoverlapping; large overlap is better for convergence
- deteorating convergence when increasing the number of subdomains (if implemented straightforwardly)
- stabilization with a coarse grid corrrection, nearly optimal convergence
- used as a preconditiner
- used in a Multigrid setting as a smoother
- attractive for parallel computations (FETI, BETI, ...)