Domain decomposition methods: a tool for the parallelization of large-scale simulations

An introduction, with a focus on scalable methods for Poisson's equation (two-level overlapping AS).

Bruno Degli Esposti

January 2020 - Topics on control and numerics of PDEs

Table of contents

The importance of being scalable

Linear, time indipendent PDE simulation pipeline

- Mesh generation or loading
- Linear system assembly ($Au = f$)
- Essential boundary conditions
- Preconditioner (optional)
- Iterative sparse linear system solver
- Postprocessing and visualization

Scalability in the context of HPC

Assumption: the linear system is too large to solve (or even assemble) on a single CPU. We need a way to distribute work among multiple processors. Moreover, we are looking for algorithms that satisfy a scalability property. Let:

- T be the total running time
- n be the size of the linear system $Au = f$
- n_{cores} be the number of parallel processors available

An algorithm is said to be (weakly) *scalable* if

$$\frac{n}{n_{\text{cores}}} \text{ constant} \implies T \text{ constant}$$

Scalability in the context of HPC

Bad news: parallel programming is hard...

Scalability in the context of HPC

Bad news: parallel programming is hard...

...because we are used to thinking in terms of sequential tasks.

Scalability in the context of HPC

Bad news: parallel programming is hard...

...because we are used to thinking in terms of sequential tasks.

An example? The pipeline itself!

Scalability in the context of HPC

Bad news: parallel programming is hard...

...because we are used to thinking in terms of sequential tasks.

An example? The pipeline itself!

Also, iterative methods for the solution of the linear system.

Scalability in the context of HPC

Without preconditioners, iterative methods cannot scale: as the size of the problem increases,

- The number of unknowns n in the linear system increases
- Hence, the condition number k of A increases
- In turn, the number of iterations n_{iter} required to get a small residual increases
- Since each iteration has a fixed cost, T increases even if n/n_{cores} remains constant.

The fix is a *scalable* preconditioner, i.e. one that can keep n_{iter} bounded as the size of the problem grows.

What is domain decomposition?

High-level overview of DDMs

All kinds of domain decomposition methods work roughly like this:

- Split the domain Ω into N subdomains Ω_i ($N \approx n_{\text{cores}}$)
- Assemble local matrices A_i for each subdomain
- Factorize each matrix A_i using a direct method (e.g. LU)
- Paste together the A_i blocks to form a preconditioner M
- Add a coarse space correction to M for scalability (optional)

Now $M^{-1}y$ can be quickly computed using the factorizations.

Linear, time indipendent PDEs simulation pipeline + DDM

- Mesh generation or loading
- **Mesh partitioning** (possibly automatic, with e.g. METIS)
- Linear system assembly ($Au = f$)
- Essential boundary conditions
- Preconditioner M **based on domain decomposition**
- Iterative sparse linear system solver
- Postprocessing and visualization

Domain decomposition is:

- A technique to calculate a scalable preconditioner in parallel
- Compatible with most discretization schemes (FE,FD,FV,...)
- Well-suited for heterogeneous or multiphysics problems
- A way to improve the robustness of iterative solvers
  (direct + iterative = hybrid, best of both worlds)

Domain decomposition is not:

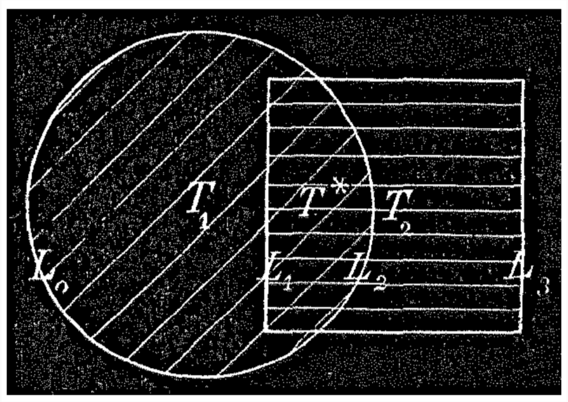
- The only way to parallelize the numerical solution of PDEs: two popular alternatives are multifrontal solvers (e.g. PARDISO) and multigrid methods (e.g. HYPRE).
- An end-to-end strategy for parallelization (the rest of the pipeline **must** be parallel as well, in accordance with Amdahl's law)
- A generic (i.e. purely algebraic, PDE-agnostic) preconditioner
- A theoretical tool for the solution of PDEs (even though proving convergence of DDM in the continuous setting is a classical topic of research in mathematical analysis).

Schwarz's method and its limitations

Historical notes (see [2] for more details)

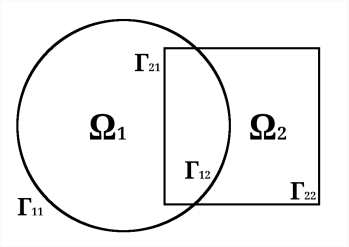
- Solution of Poisson's equation on a square (Fourier, 1807)
- Solution of Poisson's equation on a disk (Poisson, 1815)
- Dirichlet principle (well-known by 1850)
- Schwarz's domain decomposition method (1870)
- Direct method in the calculus of variations (Hilbert, 1904)
- Domain decomposition as a numerical tool (Miller, 1965)
- First scalable DDM for Poisson's equation (Dryja and Widlund, December 1987, [3])
- Independent discovery by Nicolaides in the context of deflation schemes for the conjugate gradients method (April 1987, [4])

Original Schwarz method



Original drawing by Schwarz

Original Schwarz method



The same drawing,
with modern notation

$$u_1^0 = 0 \quad \text{in } \overline{\Omega_1}$$

$$u_2^0 = 0 \quad \text{in } \overline{\Omega_2}$$

$$\begin{cases} -\Delta u_1^n = f & \text{in } \Omega_1 \\ u_1^n = 0 & \text{on } \Gamma_{11} \\ u_1^n = u_2^{n-1} & \text{on } \Gamma_{12} \end{cases}$$

$$\begin{cases} -\Delta u_2^n = f & \text{in } \Omega_2 \\ u_2^n = 0 & \text{on } \Gamma_{22} \\ u_2^n = u_1^n & \text{on } \Gamma_{21} \end{cases}$$

Original Schwarz method

Let u be the solution to Poisson's problem

$$\begin{cases} -\Delta u = f & \text{in } \Omega = \Omega_1 \cup \Omega_2 \\ u = 0 & \text{on } \partial\Omega = \Gamma_{11} \cup \Gamma_{22} \end{cases}$$

Schwarz proved that $u_1^n \to u_{|\Omega_1}$ and $u_2^n \to u_{|\Omega_2}$ in L^∞ at a geometric rate.

Original Schwarz method

Let u be the solution to Poisson's problem

$$\begin{cases} -\Delta u = f & \text{in } \Omega = \Omega_1 \cup \Omega_2 \\ u = 0 & \text{on } \partial\Omega = \Gamma_{11} \cup \Gamma_{22} \end{cases}$$

Schwarz proved that $u_1^n \to u_{|\Omega_1}$ and $u_2^n \to u_{|\Omega_2}$ in L^∞ at a geometric rate.

On the blackboard: if $u_1^n \to u_1$ and $u_2^n \to u_2$, then $u_1 = u_2$ on the overlap $\Omega_1 \cap \Omega_2$.

Original Schwarz method

Let u be the solution to Poisson's problem

$$\begin{cases} -\Delta u = f & \text{in } \Omega = \Omega_1 \cup \Omega_2 \\ u = 0 & \text{on } \partial\Omega = \Gamma_{11} \cup \Gamma_{22} \end{cases}$$

Schwarz proved that $u_1^n \to u_{|\Omega_1}$ and $u_2^n \to u_{|\Omega_2}$ in L^∞ at a geometric rate.

On the blackboard: if $u_1^n \to u_1$ and $u_2^n \to u_2$, then $u_1 = u_2$ on the overlap $\Omega_1 \cap \Omega_2$.

FreeFEM demo available in the file `original-schwarz.edp`

Original Schwarz method

List of 5 problems that we have to overcome:

(1) The algorithm is not parallel (as u_2^n depends on u_1^n)

(2) The unknowns on the overlaps are duplicated

(3) The rate of convergence depends on the size of the overlaps

(4) We're not using state-of-the-art iterative sparse linear solvers

(5) The algorithm is not scalable as $N \to \infty$.

Original Schwarz method

On the blackboard: two examples taken from Chapter 1.5 of [1], for which we compute the exact rates of convergence and show their strong dependence on the size of the overlaps.

In the second example we also see how low-frequency modes of the error converge to zero at a much slower rate than the high-frequency ones, foreshadowing the need for a two-level method (as we will see shortly).

Jacobi-Schwarz, RAS, ASM

Jacobi-Schwarz method

If problem (1) is that u_2^n depends on u_1^n, then we can fix it by simply making it depend on u_1^{n-1} instead:

$$u_1^0 = 0 \quad \text{in } \overline{\Omega_1} \qquad u_2^0 = 0 \quad \text{in } \overline{\Omega_2}$$

$$\begin{cases} -\Delta u_1^n = f & \text{in } \Omega_1 \\ u_1^n = 0 & \text{on } \Gamma_{11} \\ u_1^n = u_2^{n-1} & \text{on } \Gamma_{12} \end{cases} \qquad \begin{cases} -\Delta u_2^n = f & \text{in } \Omega_2 \\ u_2^n = 0 & \text{on } \Gamma_{22} \\ u_2^n = u_1^{n-1} & \text{on } \Gamma_{21} \end{cases}$$

We can again prove L^∞ convergence at a (usually larger) geometric rate.

FreeFEM demo available in the file `jacobi-schwarz.edp`

Jacobi-Schwarz method

Now we switch from the "decompose, then discretize" approach to "discretize, then decompose". What happens at the linear algebra level? Consider the system $Au = f$ associated with a three-point finite differences scheme in 1D with step size h, and consider the following domain decomposition with minimal overlap:

$$\Omega = (0,1), \quad \Omega_1 = (0,(l+1)h), \quad \Omega_2 = (lh,1), \quad l \in \mathbb{N} \cap [0,h^{-1}).$$

We then partition all vectors and matrices accordingly (the first block has size $l+1$, the other $h^{-1}-l$):

$$A = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix} \quad D = \begin{pmatrix} A_{11} & 0 \\ 0 & A_{22} \end{pmatrix} \quad u^n = \begin{pmatrix} u_1^n \\ u_2^n \end{pmatrix} \quad f = \begin{pmatrix} f_1 \\ f_2 \end{pmatrix}$$

Jacobi-Schwarz method

Each Jacobi-Schwarz iteration corresponds to

$$\begin{cases} A_{11} u_1^n = f_1 - A_{12} u_2^{n-1} \\ A_{22} u_2^n = f_2 - A_{21} u_1^{n-1} \end{cases}$$

$$\begin{pmatrix} A_{11} & 0 \\ 0 & A_{22} \end{pmatrix} \begin{pmatrix} u_1^n \\ u_2^n \end{pmatrix} = \begin{pmatrix} A_{11} & 0 \\ 0 & A_{22} \end{pmatrix} \begin{pmatrix} u_1^{n-1} \\ u_2^{n-1} \end{pmatrix} + \begin{pmatrix} f_1 \\ f_2 \end{pmatrix} - \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix} \begin{pmatrix} u_1^{n-1} \\ u_2^{n-1} \end{pmatrix}$$

$$Du^n = Du^{n-1} + f - Au^{n-1} =: Du^{n-1} + r^{n-1}$$

$$u^n = u^{n-1} + D^{-1} r^{n-1}$$

In numerical linear algebra, this iteration is known as the block Jacobi algorithm. Hence the name Jacobi-Schwarz.

Jacobi-Schwarz method

In order to proceed further, we need to introduce additional notation.

Let $n = \dim(A),\ m_i = \dim(A_{ii})$.

Let $R_i\colon \mathbb{R}^n \to \mathbb{R}^{m_i} \subset \mathbb{R}^n$ be the projections (or restrictions) to the subspace associated with the degrees of freedom inside Ω_i.

Then $R_i^{-1} = R_i^T$ and $A_{ii} = R_i A R_i^T$, so

$$D^{-1} = \sum_{i=1}^{N} R_i^T A_{ii}^{-1} R_i = \sum_{i=1}^{N} R_i^T (R_i A R_i^T)^{-1} R_i$$

However, this is no longer true if the overlap is not minimal, as the blocks A_{ii} will not be disjoint anymore. We double-count (or worse) on the overlaps $\Omega_i \cap \Omega_j$.

Jacobi-Schwarz method

We can solve this problem by introducing a *discrete partition of unity*: a set of matrices $D_i : \mathbb{R}^{m_i} \to \mathbb{R}^{m_i}$ such that

$$D_i \geq 0 \quad \forall i = 1, \dots, N$$

$$I = \sum_{i=1}^{N} R_i^T D_i R_i$$

Now

$$D^{-1} = \sum_{i=1}^{N} R_i^T D_i A_{ii}^{-1} R_i = \sum_{i=1}^{N} R_i^T D_i (R_i A R_i^T)^{-1} R_i$$

is always true, even if the blocks A_{ii} overlap.

Additive Schwarz Method (ASM) and Restricted Additive Schwarz (RAS) preconditioners

The iterative method $u^n = u^{n-1} + D^{-1} r^{n-1}$ is known as *overlapping block Jacobi*. Since Krylov-space methods (e.g. CG) always converge faster than fixed point iterations (like block Jacobi), we define the two following preconditioners:

$$M_{\text{RAS}}^{-1} = \sum_{i=1}^{N} R_i^T D_i (R_i A R_i^T)^{-1} R_i$$

$$M_{\text{ASM}}^{-1} = \sum_{i=1}^{N} R_i^T (R_i A R_i^T)^{-1} R_i$$

Note that block Jacobi is equivalent to Richardson's method with one of these preconditioners.

Additive Schwarz Method (ASM) and Restricted Additive Schwarz (RAS) preconditioners

M_{ASM}^{-1} is not as good as M_{RAS}^{-1}, but it's still useful in practice because it's symmetrical (and so it can be used with the conjugate gradients method, just to give an example).

The preconditioning approach greatly reduces the coupling between the convergence ratio and the size of the overlaps (see Figure 3.1 in [1]), so we can consider problems (2), (3) and (4) solved by now.

Regarding the eigenvalues of the preconditioned system, we can bound λ_{max} from above with a constant that doesn't depend on N (as in Lemma 5.9 of [1]), but it's still not possible to bound λ_{min} from below without additional hypothesis. Hence problem (5) is still open.

Two-level methods to the rescue

The need for a two-level method

The preconditioner M_{ASM}^{-1} is still not scalable. When applied to Richardson's scheme, it's easy to see that information from one subdomain flows only to its direct neighbors. As the number of subdomains grows, so does the minimum number of iterations required to get a global exchange of information. The preconditioned matrix $M^{-1}A$ has a few small eigenvalues that cause the norm of the residual to plateau when using an iterative scheme. Solution: couple all subdomains by introducing a new, non-localized term to M (a so-called *coarse-space correction*).

Nicolaides coarse space

In the case of Poisson's equation, the small eigenvalues of $M^{-1}A$ come from piecewise constant functions (i.e. the kernel of the Laplace operator on the subdomains).

How can we use this information to get a better M?

Nicolaides coarse space

Let Z be the matrix whose columns form a basis for the vector space of piecewise constants functions on the N subdomains (the *Nicolaides coarse space*):

$$Z_i = R_i^T D_i R_i \mathbf{1}$$

Let y be an approximate solution to the linear system $Au = f$, and let r be the residual $f - Ay$. Consider the problem of finding the best correction for y (in the energy norm) among the space of piecewise constants functions:

$$\min_{\beta \in \mathbb{R}^N} \|y + Z\beta - u\|_A$$

Nicolaides coarse space

Since A is SPD (discretization of $-\Delta$), the minimum is given by

$$\beta = (Z^T A Z)^{-1} Z^T r,$$

and the best correction for y is

$$Z\beta = Z(Z^T A Z)^{-1} Z^T r.$$

Notice how similar this looks to the expression for M_{ASM}^{-1}!

Two-level additive Schwarz preconditioner

This observation suggests the following definition of a two-level additive Schwarz preconditioner:

$$M_{\mathrm{ASM},2}^{-1} = Z(Z^T A Z)^{-1} Z^T + \sum_{i=1}^{N} R_i^T (R_i A R_i^T)^{-1} R_i$$

It's now possible to prove that $M_{\mathrm{ASM},2}^{-1}$ is a scalable preconditioner (see e.g. Corollary 5.12 in [1]).

FreeFEM demo available in the file `AS2-PCG.edp`

Two-level additive Schwarz preconditioner

Has problem (5) been solved?

Did we finally get an algorithm that scales, as $n \to \infty$?

Two-level additive Schwarz preconditioner

Has problem (5) been solved?
Did we finally get an algorithm that scales, as $n \to \infty$?

Almost. The devil is in the details...

Two-level additive Schwarz preconditioner

Has problem (5) been solved?
Did we finally get an algorithm that scales, as $n \to \infty$?

Almost. The devil is in the details...

We need to factorize $Z^T A Z$, an $N \times N$ matrix. As $n \to \infty$ implies $N \to \infty$, this a very difficult problem to solve in a scalable way (because it's just as hard as factorizing A in the first place).

Two-level additive Schwarz preconditioner

Has problem (5) been solved?
Did we finally get an algorithm that scales, as $n \to \infty$?

Almost. The devil is in the details...

We need to factorize $Z^T A Z$, an $N \times N$ matrix. As $n \to \infty$ implies $N \to \infty$, this a very difficult problem to solve in a scalable way (because it's just as hard as factorizing A in the first place).

Three-level methods to the rescue?

Two-level additive Schwarz preconditioner

Has problem (5) been solved?
Did we finally get an algorithm that scales, as $n \to \infty$?

Almost. The devil is in the details...

We need to factorize $Z^T A Z$, an $N \times N$ matrix. As $n \to \infty$ implies $N \to \infty$, this a very difficult problem to solve in a scalable way (because it's just as hard as factorizing A in the first place).

Three-level methods to the rescue?

Not necessarily, unless N is greater than the number of degrees of freeedom assigned to each subdomain Ω_i. How large is the problem that we want to solve, anyway?

Beyond Poisson's equation

Beyond Poisson's equation

Here are some advanced concepts that we didn't cover during this presentation, but nevertheless deserve to be mentioned:

- Optimized Schwarz methods (subdomains share information through Robin boundary conditions instead of Dirichlet)
- Spectral coarse spaces (a generalization of Nicolaides spaces, defined by harmonic extensions of the eigenvectors corresponding to small eigenvalues of the Dirichlet-to-Neumann map in each subdomain)
- Neumann–Neumann and FETI algorithms (they require nonoverlapping domain decompositions)
- The GenEO method (can build robust and efficient coarse spaces by solving generalized eigenvalue problems)

Beyond Poisson's equation

Besides Poisson's equation, domain decomposition methods have been successfully applied to:

- Helmholtz equation
- Biharmonic equation
- Equilibrium equations of linear elasticity
- Stokes equations
- Maxwell's equations in time-harmonic form

and more. Understanding how DDM can be best applied to different kinds of PDEs is currently the subject of ongoing research.

Domain decomposition, in practice

HPDDM is a C++ library that implements various domain decomposition methods, with a focus on performance. It's well integrated with PETSc, Trilinos and FreeFEM (using the `ffddm` plugin). The first version of the software was written by Pierre Jolivet and Frédéric Nataf, two of the authors of [1].

Bibliography

[1] Dolean, V., Jolivet, P., & Nataf, F. (2015). An introduction to domain decomposition methods: algorithms, theory, and parallel implementation (Vol. 144). SIAM.

[2] Gander, M. J., & Wanner, G. (2014). The origins of the alternating Schwarz method.

[3] Widlund, O., & Dryja, M. (1987). An additive variant of the Schwarz alternating method for the case of many subregions.

[4] Nicolaides, R. A. (1987). Deflation of conjugate gradients with applications to boundary value problems.

[5] Lions, P. L. (1989). On the Schwarz alternating method II.

That’s all folks

Thank you for your attention!

Are there any questions?