Domain decomposition

Lukas Einkemmer

Department of Mathematics
University of Innsbruck

May 19, 2015

Problem

Let us consider the model problem

$$\Delta u = f, u(0) = 0, u(1) = 1$$

discretized using finite differences a grid of N elements on $[0, 1]$.

The discretization results in a linear system of equations written in matrix form as

$$Au = f$$

with a large matrix A.

We can use fast Poisson solvers such as FFT based methods that are $\mathcal{O}(n \log n)$ but can only be applied to simple geometries and linear problems. Direct solvers (such as LU factorization) take $\mathcal{O}(N^3)$ operations. Basic iterative solvers (such as Jacobi or Gauss-Seidel) take $\mathcal{O}(N^3)$ operations. Krylov based iterative solvers (such as CG or GMRES) require $\mathcal{O}(N^2)$ operations.

Reminder

For $A = D + L + U$ the Jacobi iteration is given by

$$x_{n+1} = -D^{-1}(L + U)x_n + D^{-1}f$$

with iteration matrix $T = -D^{-1}(L + U)$.

The error is given by

$$\epsilon_n = u - x_n$$

and satisfies

$$\epsilon_n \leq \rho(T)^n \epsilon_0$$

Usually we have $\rho(T) = 1 - \mathcal{O}(N^2)$ which means that convergence is slow for large N.

Reminder

For $A = D + L + U$ the Gauss-Seidel iteration is given by

$$x_{n+1} = -(L + D)^{-1} U x_n + (L + D)^{-1} f$$

with iteration matrix $T = -(L + D)^{-1} U$.

There are block variants of the Jacobi and Gauss-Seidel iterations.

Domain decomposition

The fundamental idea of domain decomposition is that we solve a smaller problem on a subdomain $\Omega_i \subset \Omega$ which is an approximation to the large problem on Ω.

On the subdomains Ω_i we can use a standard methods (such as direct or iterative solvers).

Intuitively we would distribute the domain in non-overlapping parts that only interact via the boundary. But

- we know that in a parabolic problem a perturbation at one point affects every other point
- in each step of the domain decomposition method interaction only takes place on the boundary between the domain
- to propagate information from the left to the right requires $\mathcal{O}(n)$ steps

Schwarz methods

In practice it is not as bad. Influence of a perturbation at p_1 decreases as

$$\mathrm{e}^{-Dd(p_1,p_2)^2}$$

for a point p_2, where D is a constant and $d(p_1, p_2)$ is the distance between them (can be easily seen from the fundamental solution/Green's function).

Domain decomposition with overlap are called **Schwarz methods**.

For simplicity let us consider a domain Ω that is decomposed into Ω_1 and Ω_2. We denote the artificial boundaries by $\Gamma_1 = \Omega \backslash \Omega_2$ and $\Gamma_2 = \Omega \backslash \Omega_1$.

Alternating Schwarz method

Let us denote the unknowns in a domain by

$$u_{\Omega_i}$$

Then we can write the problem

$$Au = f$$

as a block matrix

$$\begin{bmatrix} A_{\Omega_1} & A_{\Gamma_1} I_{\Omega_2 \to \Gamma_1} \\ A_{\Gamma_2} I_{\Omega_1 \to \Gamma_2} & A_{\Omega_2} \end{bmatrix} \begin{bmatrix} u_{\Omega_1} \\ u_{\Omega_2} \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \end{bmatrix}.$$

Now employ a block Gauss-Seidel scheme

$$u_{\Omega_1}^{n+1} = A_{\Omega_1}^{-1}(f_1 - A_{\Gamma_1} I_{\Omega_2 \to \Gamma_1} u_{\Omega_2})$$

$$u_{\Omega_2}^{n+1} = A_{\Omega_2}^{-1}(f_2 - A_{\Gamma_2} I_{\Omega_1 \to \Gamma_2} u_{\Omega_1})$$

This algorithm is called the **alternating Schwarz method**.

Eliminating the boundary

It is more convenient to eliminate the boundary degree of freedom. We partition our degrees of freedom into

$$u = (u_{\Omega_1 \setminus \overline{\Omega}_2}, u_{\Gamma_2}, u_{\Omega_1 \cap \Omega_2}, u_{\Gamma_1}, u_{\Omega_2 \setminus \overline{\Omega}_1})^T$$

and partition

$$A = \begin{bmatrix} A_{\Omega_1} & A_{\Omega \setminus \Omega_1} \\ A_{\Omega \setminus \Omega_2} & A_{\Omega_2} \end{bmatrix} = \begin{bmatrix} A_{\Omega_1} & A_{\Gamma_1} & A_{\Omega_2 \setminus \overline{\Omega}_1} \\ A_{\Omega_1 \setminus \overline{\Omega}_2} & A_{\Gamma_2} & A_{\Omega_2} \end{bmatrix}$$

The alternating Schwarz method

$$u_{\Omega_1}^{n+1} = A_{\Omega_1}^{-1}(f_1 - A_{\Gamma_1} I_{\Omega_2 \to \Gamma_1} u_{\Omega_2})$$

$$u_{\Omega_2}^{n+1} = A_{\Omega_2}^{-1}(f_2 - A_{\Gamma_2} I_{\Omega_1 \to \Gamma_2} u_{\Omega_1})$$

can then be written (assuming only coupling at the boundary)

$$u_{\Omega_1}^{n+1} = A_{\Omega_1}^{-1}(f_1 - A_{\Omega \setminus \Omega_1} u_{\Omega \setminus \Omega_1}^{n})$$

$$u_{\Omega_2}^{n+1} = A_{\Omega_2}^{-1}(f_2 - A_{\Omega \setminus \Omega_2} u_{\Omega \setminus \Omega_2}^{n+1})$$

Coupling

In many numerical schemes coupling is not restricted to the boundary. For example, high order finite difference methods.

However, the scheme can still be defined and is called the **multiplicative Schwarz method**.

We have eliminated treatment of the boundary terms. Note that at the artificial boundary homogeneous (zero) Dirichlet boundary condition have to be specified.

Domain decomposition as a preconditioner

Let us introduce the restriction matrices

$$u_{\Omega_1} = R_1 u = [I\ 0] \begin{bmatrix} u_{\Omega_1} \\ u_{\Omega \setminus \Omega_1} \end{bmatrix}$$

$$u_{\Omega_2} = R_2 u = [0\ I] \begin{bmatrix} u_{\Omega \setminus \Omega_2} \\ u_{\Omega_2} \end{bmatrix}.$$

Now, we can write

$$A_{\Omega_1} = R_i A R_i^T$$

and the multiplicative Schwarz method is given by

$$u^{n+1/2} = u^n + R_1^T (R_1 A R_1^T)^{-1} R_1 (f - A u^n)$$

$$u^{n+1} = u^{n+1/2} + R_2^T (R_2 A R_2^T)^{-1} R_2 (f - A u^{n+1/2})$$

This is a purely theoretical tool. One should not assemble this matrices in an implementation.

Domain decomposition as a preconditioner

For $B_i = R_i^T (R_i A R_i^T)^{-1} R_i$ we can write the multiplicative Schwartz method as

$$u_{n+1} = u_n + (B_1 + B_2 - B_2 A B_1)(f - A u_n)$$

This is the Richardson iteration with

$$P^{-1} = B_1 + B_2 - B_2 A B_1$$

as preconditioner.

The **multiplicative** in the name of the method is due to the matrix product that appears in the preconditioner.

We can use domain decomposition as a preconditioner for Krylov subspace methods. In this case u_1 with $u_0 = 0$ gives the corresponding matrix-vector product.

Even if A is symmetric this is, in general, not true for the preconditioner.

Additive Schwarz method

We can also use block Jacobi instead of block Gauss-Seidel. In this case we obtain

$$u_{n+1} = u_n + \begin{bmatrix} A_{\Omega_1}^{-1} & 0 \\ 0 & A_{\Omega_2}^{-1} \end{bmatrix} (f - Au_n)$$

or

$$u_{n+1} = u_n + (B_1 + B_2)(f - Au_n)$$

This is called the **additive Schwarz method**.

This method can also be considered a simplification of the multiplicative Schwarz method where we drop the B_1AB_1 term in the preconditioner.

- Convergence is usually slower than for the multiplicative variant (often a factor of two)
- But can be applied in parallel

Domain decomposition as preconditioner

In practice domain decomposition is used as a preconditioner for a Krylov subspace method.

For the multiplicative Schwarz method compute

$$y = Au$$

$$v_{\Omega_1} = A_{\Omega_1}^{-1} y_{\Omega_1}, \qquad v_{\Omega_2} = A_{\Omega_2}^{-1} R_2 (y - Av)$$

which corresponds to an initial choice of $v = 0$.

Many domains

In practice we consider a decomposition into many domains Ω_i for $1 \leq i \leq p$.

The Schwarz methods can be readily extended to this case but the multiplicative variant is entirely sequential

$$u_{n+1/p} = u_n + B_1(f - Au_n)$$

$$u_{n+2/p} = u_{n+1/p} + B_2(f - Au_{n+1/p})$$

...

$$u_{n+1} = u_{n+(p-1)/p} + B_p(f - Au_{n+(p-1)/p})$$

Coloring

Define a coloring such that each subdomain has a color but no two subdomains that share a grid point have the same color.

For each color we perform a multiplicative Schwarz method

The domains of separate color are independent and can be computed in parallel.

Convergence rate is worse than the full algorithm but better than for the additive Schwarz method.

In 2d 4 colors are sufficient, in 3d 8 colors (for a rectangular grid).

Ideally, 4 or 8 times as many subdomains as there are processors.

Convergence

$\mathcal{O}(1)$ domain length
$\mathcal{O}(h)$ grid spacing
$\mathcal{O}(H)$ subdomain length
$\mathcal{O}(\delta)$ overlap

For parabolic problems the Schwarz methods used as a preconditioner for Krylov subspace methods usually satisfies

- The number of iterations grow as $1/H$
- If $\delta \propto H$ the number of iterations is bounded independent of h and H/h
- Convergence is poor for $\delta = 0$ but increases rapidly as the overlap δ increases.

Limitations

Let us try to understand the $1/H$ dependence on the number of iterations.

Note that the row sum of A (from the discretization of a parabolic PDE) for interior points is zero.

Let us assume that ϵ_n is constant over a region that includes a subdomain Ω_i.
The correction is given by

$$A_{\Omega_i}^{-1}(f - Au_n) = A_{\Omega_i}^{-1} A\epsilon_n = 0$$

Note that for small Ω_i the error is

$$\epsilon = \overline{\epsilon} + \epsilon_r,$$

where $\epsilon_r/\overline{\epsilon}$ is small.
Since the average error is not damped the error decreases only slowly.

Two-level Schwarz method

This consideration limits domain decomposition to a small number of domains if computational efficiency is desired.

But we know how to solve this issue.

Let us consider a fine mesh on which we consider the system

$$A_F u_F = f$$

Now a correction is computed

$$u_F = u_F + R^T A_C^{-1} R(f - A_F u_F)$$

which is the same as in a two-grid scheme

- calculate residual on the fine grid
- restrict the residual to the coarse grid
- solve the coarse grid problem
- interpolate the coarse grid solution to the fine grid

Two-level Schwarz method

We already know that this is not sufficient as high frequencies errors are not damped at all.
In a classical multigrid scheme we use a weighted Jacobi iteration to remove the high frequency errors on the fine grid.

- We have seen that the choice of this method is important (simple Jacobi does not work very well for very high frequencies)
- But note that a local method is sufficient in this case

Algorithm:

- First apply

$$u_F^{n+1} = u_F^n + B_C(f - A_F u_F^n)$$

  long range but computationally cheap correction.

- Then apply

$$u_F^{n+1} = u_F^{n+1/2} + B_F(f - A_F u_F^{n+1/2})$$

  localized and parallelizable but expensive method that removes high frequencies. Use Domain decomposition.

Two-level Schwarz method

The two-level Schwarz method can be seen as an extension of classic multigrid scheme where we replace the simple iterative solver by the (more robust) domain decomposition approach.

As for multigrid methods we usually would use this as a preconditioner for a Krylov subspace method.

Jacobi iteration is just the special case where $\Omega_i = \{i\}$; i.e. each domain consists only of a single grid point.

Convergence behavior:

- For $\delta \propto H$ the number of iterations is independent of h, H, and H/h.
- Convergence is poor for $\delta = 0$ but improves rapidly as the overlap increases.

Many of the limitations for multigrid schemes also remain for the two-level Schwarz methods.

Literature

Smith, Bjorstad, Gropp, Domain Decomposition, 1996

Saad, Iterative methods for sparse linear systems, 2004