Marrakech Jan 2003

AN INTRODUCTION TO

DOMAIN DECOMPOSITION METHODS

Gérard MEURANT

CEA

Introduction

- Domain decomposition is a "divide and conquer" technique
- Natural framework to introduce parallelism in the solution of elliptic or parabolic PDE's
- General scheme:
    - Decompose the problems into subproblems
    - Solve the subproblems in parallel
    - Glue the (sub)solutions together to get the global solution
- Generally the subproblems correspond to partitioning the domain
- The modern view on DD is to construct preconditioners for Krylov iterative methods for solving linear systems (CG, GMRES, BiCG,... depending on the properties of the matrix)

- There are hundreds of variants of DD preconditioners
- Two main classes
    - methods with subdomain overlapping (Schwarz–like)
    - methods without overlapping (interface problems)
- Methods may differ also on other issues:
    - exact or inexact solvers for the subproblems
    - solving a reduced system or the global system
    - etc...
- Most DD methods for PDEs rely on mesh partitioning (in most cases we need a graph partitioner (METIS,...))

Books on DD

Proceedings of the DDM conferences (first in Paris 1987)

SMITH, BJORSTAD and GROPP, Domain decomposition: parallel multilevel methods for elliptic partial differential equations, Cambridge University Press (1996)

QUARTERONI and VALLI, Domain decomposition methods for partial differential equations, Clarendon Press (1999)

MEURANT, Computer solution of large linear systems, North–Holland (1999)

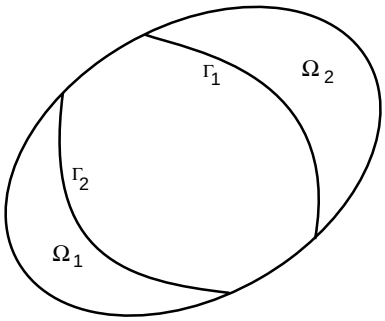
The classical Schwarz alternating method

H.A. SCHWARZ, Uber einige abbildungsaufgaben, Ges. Math. Abh. vol 11, (1869) pp 65-83

• Solve a 2^{nd} order elliptic PDE with Dirichlet b.c. in a bounded 2D domain Ω

• The domain Ω is split into two (or more) overlapping subdomains Ω_1 and Ω_2

• Γ_i, $i = 1, 2$, is the part of the boundary of Ω_i enclosed in Ω



◦ Guess a value for the unknowns on the inner boundary Γ_1

◦ Solve the problem exactly in Ω_1

◦ Use the computed values on the inner boundary Γ_2 to solve exactly in Ω_2

◦ Repeat the process until convergence

• This very simple method "almost always" converges

• Can be analyzed at the continuous or discrete level

• We consider the Poisson model problem but the results are more generally true for self–adjoint continuous bilinear forms

$$-\Delta u = f \ \text{ in } \Omega, \quad u|_{\partial\Omega} = 0.$$

• Basic Schwarz =

$$-\Delta u^{2k} = f \ \text{ in } \Omega_1, \quad u^{2k}|_{\Gamma_1} = u^{2k-1}|_{\Gamma_1},$$

$$-\Delta u^{2k+1} = f \ \text{ in } \Omega_2, \quad u^{2k+1}|_{\Gamma_2} = u^{2k}|_{\Gamma_2},$$

+ given boundary conditions on the other parts of the boundary

• The bilinear form a of the problem is defined as

$$a(u,v) = \int_\Omega \nabla u \cdot \nabla v \, dx.$$

• The model problem can be written in variational form as

$$a(u,v) = (f,v), \quad \forall v \in H_0^1(\Omega).$$

• Let $V_1 = H_0^1(\Omega_1)$ and $V_2 = H_0^1(\Omega_2)$ and the projectors P_1 and P_2 defined by

$$a(P_i v, w) = a(v, w), \ \forall w \in V_i, i = 1, 2.$$

• Functions defined only on subdomains are extended by 0 to $H_0^1(\Omega)$

$$a(u^{2k} - u, v_1) = 0, \quad \forall v_1 \in V_1, \quad u^{2k} - u^{2k-1} \in V_1,$$

$$a(u^{2k+1} - u, v_2) = 0, \quad \forall v_2 \in V_2, \quad u^{2k+1} - u^{2k} \in V_2.$$

• This gives

$$u - u^{2k} = (I - P_1)(u - u^{2k-1}),$$

$$u - u^{2k+1} = (I - P_2)(u - u^{2k}).$$

Therefore,

$$u - u^{2k+1} = (I - P_2)(I - P_1)(u - u^{2k-1}).$$

- This is known as the multiplicative Schwarz method
- We have to study the convergence of

$$v^0 \in V, \quad v^{2k} = (I - P_1)v^{2k-1}, \quad v^{2k+1} = (I - P_2)v^{2k}.$$

Theorem (P.L. Lions, DDM 1987)

If $V = \overline{V_1 + V_2}$, where the overbar denotes the closure of the set, then $v^k \longrightarrow 0$.

Moreover, if $V = V_1 + V_2$ then

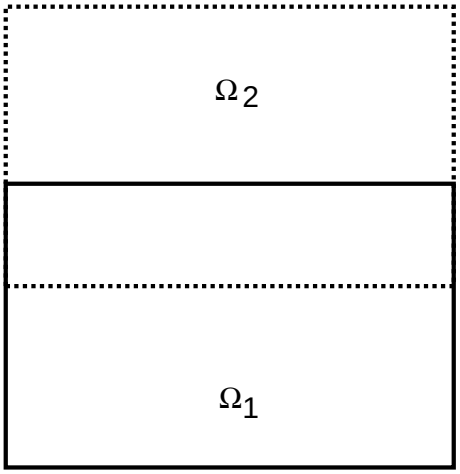
$$\|(I - P_2)(I - P_1)\| \leq c, \quad c < 1.$$

The matrix form of the Schwarz alternating method

• Solve a 2^{nd} order elliptic equation in a rectangle using a 5 point FD scheme with the natural (rowwise) ordering. In block form

$$A = \begin{pmatrix} D_1 & -B_2^T & & & \\ -B_2 & D_2 & -B_3^T & & \\ & \ddots & \ddots & \ddots & \\ & & -B_{m-1} & D_{m-1} & -B_m^T \\ & & & -B_m & D_m \end{pmatrix}.$$

Suppose the mesh is partitioned as



• The matrix $A^{(1)}$ corresponding to Ω_1 is

$$A^{(1)} = \begin{pmatrix} D_1 & -B_2^T & & & \\ -B_2 & D_2 & -B_3^T & & \\ & \ddots & \ddots & \ddots & \\ & & -B_{p-2} & D_{p-2} & -B_{p-1}^T \\ & & & -B_{p-1} & D_{p-1} \end{pmatrix},$$

• The matrix $A^{(2)}$ corresponding to Ω_2 is

$$A^{(2)} = \begin{pmatrix} D_{l+1} & -B_{l+2}^T & & & \\ -B_{l+2} & D_{l+2} & -B_{l+3}^T & & \\ & \ddots & \ddots & \ddots & \\ & & -B_{m-1} & D_{m-1} & -B_m^T \\ & & & -B_m & D_m \end{pmatrix}.$$

• Let us denote the matrix A in block form as

$$A = \begin{pmatrix} A^{(1)} & A^{(1,2)} \\ X & X \end{pmatrix} \text{ and } A = \begin{pmatrix} Y & Y \\ A^{(2,1)} & A^{(2)} \end{pmatrix},$$

and let b_1 and b_2 be the restrictions of the right hand side b to Ω_1 and Ω_2

• Note that $A^{(1,2)}$ has only one non–zero block in the left lower corner and $A^{(2,1)}$ is zero except for the upper right block

• We denote by x_1 and x_2 the unknowns in Ω_1 and Ω_2

• We extend the vectors x_1 and x_2 to Ω by completing with the components of the previous iterate

• The Schwarz alternating method is

$$A^{(1)}x_1^{2k} = b_1 + \begin{pmatrix} 0 \\ \vdots \\ 0 \\ B_p^T (x_2^{2k-1})_p \end{pmatrix}, \quad A^{(2)}x_2^{2k+1} = b_2 + \begin{pmatrix} B_{l+1}(x_1^{2k})_l \\ 0 \\ \vdots \\ 0 \end{pmatrix}.$$

$\Longrightarrow$

$$\begin{aligned} x_1^{2k} &= x_1^{2k-1} + (A^{(1)})^{-1}(b_1 - A^{(1)}x_1^{2k-1} - A^{(1,2)}x_{1,2}^{2k-1}), \\ x_2^{2k+1} &= x_2^{2k} + (A^{(2)})^{-1}(b_2 - A^{(2)}x_2^{2k} - A^{(2,1)}x_{2,1}^{2k}). \end{aligned}$$

$$\begin{aligned} x^{2k} &= x^{2k-1} + \begin{pmatrix} (A^{(1)})^{-1} & 0 \\ 0 & 0 \end{pmatrix}(b - Ax^{2k-1}), \\ x^{2k+1} &= x^{2k} + \begin{pmatrix} 0 & 0 \\ 0 & (A^{(2)})^{-1} \end{pmatrix}(b - Ax^{2k}). \end{aligned}$$

By eliminating x^{2k} we obtain

$$x^{2k+1} = x^{2k-1} + [\begin{pmatrix} (A^{(1)})^{-1} & 0 \\ 0 & 0 \end{pmatrix} + \begin{pmatrix} 0 & 0 \\ 0 & (A^{(2)})^{-1} \end{pmatrix} - \begin{pmatrix} 0 & 0 \\ 0 & (A^{(2)})^{-1} \end{pmatrix} A \begin{pmatrix} (A^{(1)})^{-1} & 0 \\ 0 & 0 \end{pmatrix}] r^{2k-1},$$

$$r^{2k-1} = b - Ax^{2k-1}.$$

• The Schwarz alternating method is nothing else than a preconditioned Richardson iteration

• This method can also be written with another notation

◦ We introduce restriction operators R_1 and R_2

$$x_1^k = R_1 x^k, \quad x_2^k = R_2 x^k.$$

In our example R_1 is simply $\begin{pmatrix} I_{p-1} & 0 \end{pmatrix}$ and $R_2 = \begin{pmatrix} 0 & I_{m-l+1} \end{pmatrix}$

$$A^{(1)} = R_1 A R_1^T, \quad A^{(2)} = R_2 A R_2^T.$$

- The first step of the iteration is:
    - restriction by R_1
    - apply the inverse of $R_1 A R_1^T$
    - extension of the result by R_1^T

$$x^{2k} = x^{2k-1} + R_1^T (R_1 A R_1^T)^{-1} R_1 (b - A x^{2k-1}).$$

- The second step is

$$x^{2k+1} = x^{2k} + R_2^T (R_2 A R_2^T)^{-1} R_2 (b - A x^{2k}).$$

Proposition

The matrix $P_i = R_i^T (R_i A R_i^T)^{-1} R_i A, i = 1, 2$ is an orthogonal projection in the scalar product defined by A

If ε^k is the error, we have

$$\varepsilon^{2k} = (I - P_1)\varepsilon^{2k-1}, \quad \varepsilon^{2k+1} = (I - P_2)\varepsilon^{2k}.$$

- P_i is the discrete version of the projection operator we introduced earlier

The rate of convergence

• Let us consider a one dimensional Poisson model problem

$$A^{(1)} = \begin{pmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & -1 \\ & & & -1 & 2 \end{pmatrix}$$

of order $p-1$ and $A^{(2)}$ which is the same matrix but of order $n-l$

Proposition

We have

$$\varepsilon_i^{2k} = \frac{i}{p}\varepsilon_p^{2k}, i = 1, \ldots, p-1$$

$$\varepsilon_i^{2k+1} = \frac{n-i+1}{n-l+1}\varepsilon_l^{2k+1}, i = l+1, \ldots, n.$$

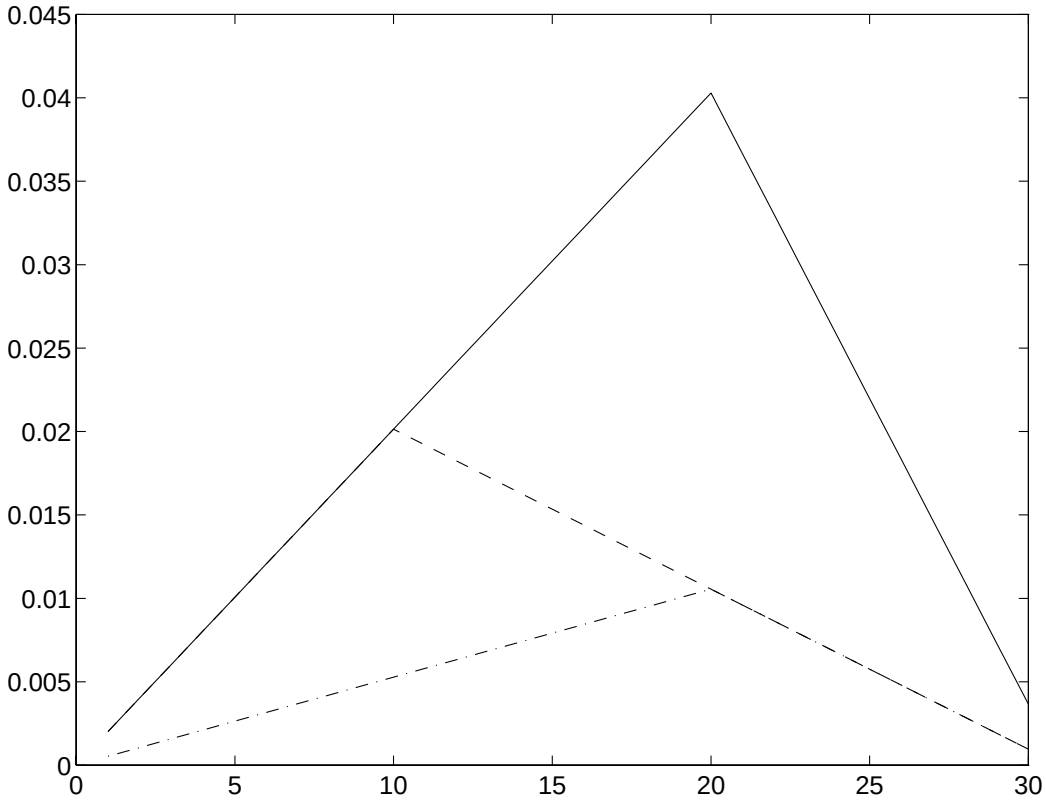
• This is because

$$A^{(1)} \begin{pmatrix} \varepsilon_1^{2k} \\ \vdots \\ \varepsilon_{p-1}^{2k} \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ \varepsilon_p^{2k-1} \end{pmatrix}$$

and $\varepsilon_p^{2k} = \varepsilon_p^{2k-1}$, $\varepsilon_l^{2k+1} = \varepsilon_l^{2k}$

• At the end of the first half step, the error is maximum for the node p and linear (being 0 at the ends of the interval)

• At the end of the second half step, the error is maximum for the node l and linear



Theorem

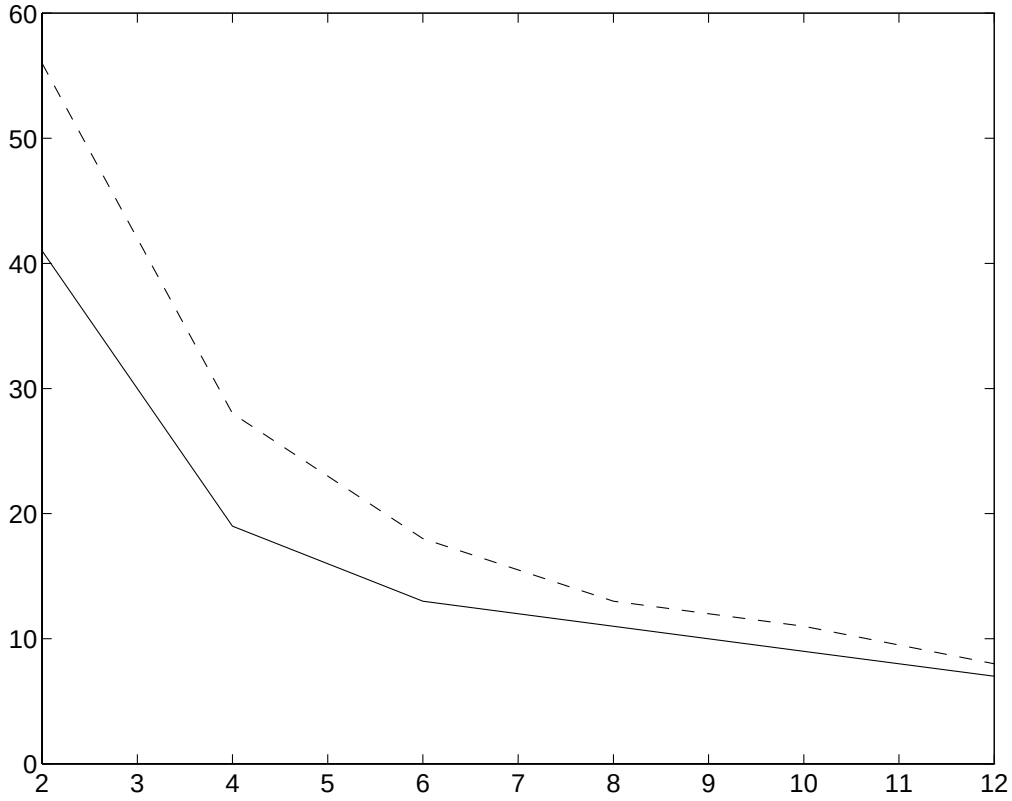
At odd steps, the maximum of the (absolute value) of the error is obtained for node l and

$$\|\varepsilon^{2k+1}\|_\infty = \frac{l}{p}\frac{n-p+1}{n-l+1}\|\varepsilon^{2k-1}\|_\infty.$$

• The larger the overlap $(p-l)$ the faster the convergence because $n-l+1 = n-p+1+(p-l)$

• The same analysis can be done on this problem for a larger number of subdomains since the error is still linear on each subdomain

• The rate of convergence is slower when we have a large number of subdomains



Number of iterations as a function of the overlap

solid line: two subdomains, dashed: three subdomains

Other boundary conditions

• A way to reduce the overlap while maintaining a good convergence rate is to use other inner boundary conditions than Dirichlet for the subproblems (W.P. Tang, M. Gander and al.)

• WPT proposed using inner mixed boundary conditions like continuity of

$$\omega u + (1 - \omega) \frac{\partial u}{\partial n}.$$

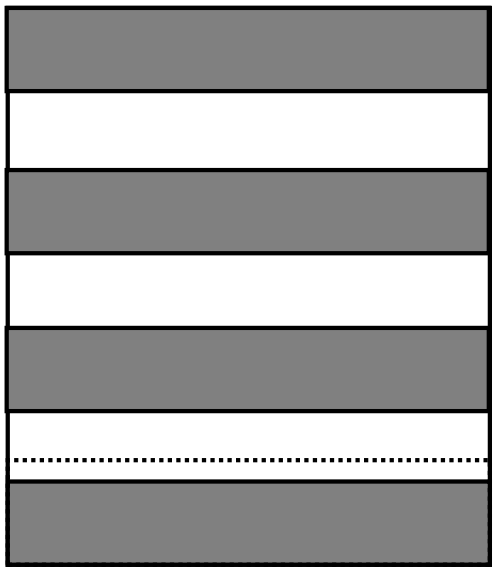
• Numerical results show that this can substantially improve the rate of convergence for small overlaps

Parallelizing Schwarz methods

• There is no parallelism in the Schwarz alternating method

• To get a parallel algorithm we may use a coloring of the subdomains such that a subdomain of one color is only connected to subdomains of other colors

• For strips a red–black ordering is used, every other strip is black, and red strips alternate with black strips



The additive Schwarz method

• The alternating (multiplicative) Schwarz method can be considered as a kind of `Gauss-Seidel` algorithm

• A way to get a parallel algorithm is to use instead a block `Jacobi`–like method

This is known as the Additive Schwarz method, (Dryja and Widlund)

$$x^{k+1} = x^k + \theta \sum_i R_i^T (R_i A R_i^T)^{-1} R_i (b - A x^k)$$

The preconditioner is

$$M^{-1} = \sum_i R_i^T (R_i A R_i^T)^{-1} R_i,$$

Notice there is no parameter

The iteration matrix is

$$T_{AS} = I - \theta M^{-1} A$$

where the summation is over the number of overlapping subdomains

• More generally, one can replace the exact solves for each subdomain by approximations and define

$$M^{-1} = \sum_i R_i^T M_i^{-1} R_i.$$

• Theory by Benzi, Frommer, Nabben and Szyld, Num. Math. v 89 (2001) pp 625–639

see also Frommer and Szyld, Num. Math. v 83 (1999) pp 259–278

Theorem (BFNS)

A M–matrix, $\theta < 1/q$, q max nb of ovl subdomains, then AS converges $\forall x^0$,

MS converges $\forall x^0$

Theorem (Nabben)

A M–matrix, MS converges always faster than AS

Adding a coarse mesh correction

• The rate of convergence of the multiplicative or additive Schwarz methods depends on the number of subdomains

• To improve on this we add a coarse grid correction

• The coarse grid generally corresponds to the interfaces in the partitioning

$$M^{-1} = \sum_i R_i^T (R_i A R_i^T)^{-1} R_i + R_0^T A_0^{-1} R_0,$$

• The coarse grid operator may be chosen as a Galerkin approximation $A_0 = R_0 A R_0^T$

• If the extent of overlap is kept proportional to the "sizes" of the subdomains the number of iterations is independent of n and of the number of subdomains

An additive Schwarz preconditioner for parabolic problems

$$\frac{\partial u}{\partial t} - \frac{\partial}{\partial x}(a(x,y)\frac{\partial u}{\partial x}) - \frac{\partial}{\partial y}(b(x,y)\frac{\partial u}{\partial y}) = f$$

$$\text{in } \Omega \subset \Re^2,$$

$$u\mid_{\partial\Omega} = 0, \quad u(x,0) = u_0(x).$$

The model problem that is considered is the heat equation :

$$\frac{\partial u}{\partial t} - \Delta u = f,$$

with Dirichlet boundary conditions, Ω being the unit square

- For stability and efficiency an implicit Crank–Nicolson scheme
- We discretize the space variables with FD

• Time is discretized with the usual C-N scheme : with $t \in [0, T]$, $k = \Delta t$ being the time step, p referring to the values of unknowns at time pk, we have

$$\frac{u^{p+1} - u^p}{k} + \frac{1}{2h^2}(Au^{p+1} + Au^p) = \frac{1}{2}(f^{p+1} + f^p),$$

• At every time step this gives a linear system to solve:

$$(2\frac{h^2}{k}I + A)u^{p+1} = 2\frac{h^2}{k}u^p - Au^p + h^2(f^{p+1} + f^p).$$

$$A_t x = h^2(f^{p+1} + f^p) - 2Au^p,$$

where $\theta = 2\frac{h^2}{k}$, and $A_t = \theta I + A$

Then $u^{p+1} = x + u^p$

• We remark that for our problem, A_t is a symmetric strictly diagonally dominant M–matrix

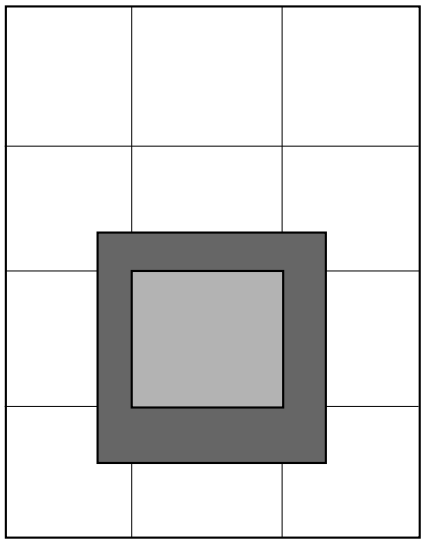
• This method was inspired by an algorithm of Y. Kuznetsov

• It belongs to the class of Additive Schwarz methods

• Contrary to Kuznetsov, we use `CG` to solve the linear system at each time step and DD only to provide a preconditioner

• The domain Ω is divided into non–overlapping subdomains Ω_i, $i = 1, \ldots, l$

• Each Ω_i is extended to a domain $\widehat{\Omega}_i$ that overlaps the neighbors of Ω_i (restricted of course to Ω)



• To solve $Mz = r$, the following steps are performed :

◦ for each subdomain $\widehat{\Omega}_i$, $i = 1, \ldots, l$, let $\widehat{A_i}$ be the $n_i \times n_i$ matrix arising from the discretization of the problem on $\widehat{\Omega}_i$ with homogeneous Dirichlet boundary conditions

◦ Let $\widehat{r_i}$ be the vector of length n_i that will be the right hand side on $\widehat{\Omega}_i$, whose entries are equal to those of r for components corresponding to mesh points in Ω_i and 0 elsewhere

◦ Then let $\widehat{z_i}$ be defined by solving a problem on $\widehat{\Omega}_i$:

$$\widehat{A_i}\widehat{z_i} = \widehat{r_i}.$$

◦ We extend $\widehat{z_i}$ to a vector z_i on the whole Ω with 0 components corresponding to mesh points outside $\widehat{\Omega}_i$

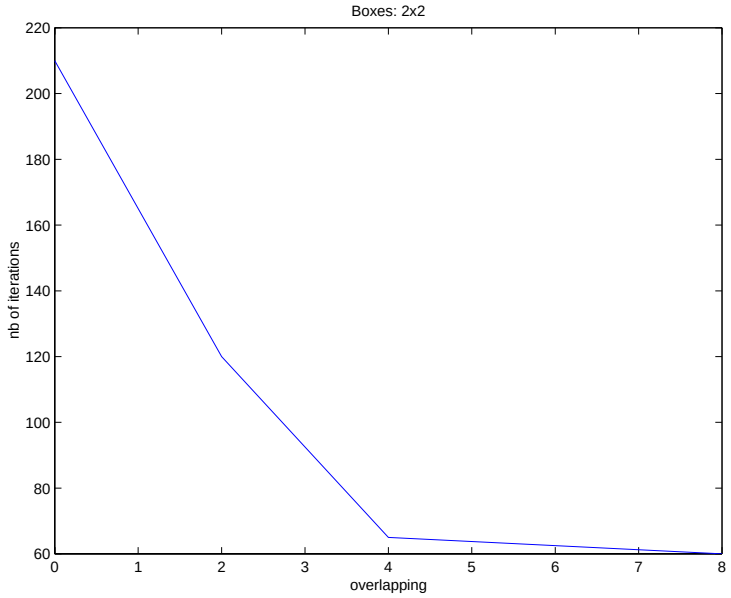
◦ The solution z is simply defined as

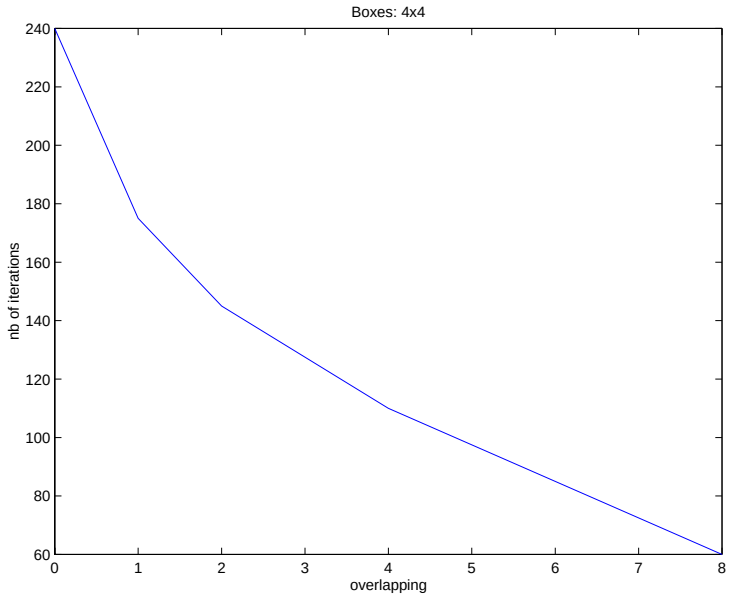
$$z = \sum_{i=1}^{l} \widehat{z_i}.$$

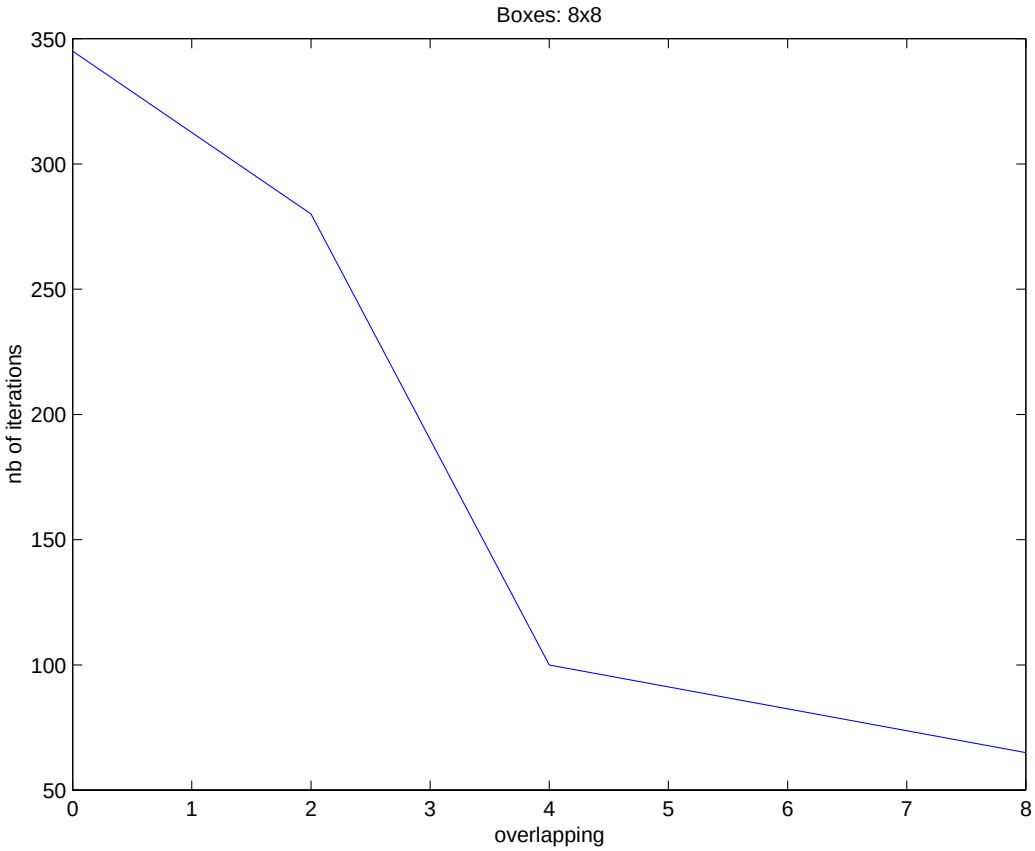
• The main problem is to know where to put the artificial boundaries

Numerical results

Heat equation in the unit square, $u = \sin(\pi t)xy(1-x)(1-y)\exp(xy)$, $T = 0.5$, $h = 1/32$, $k = \Delta t = 1/60$





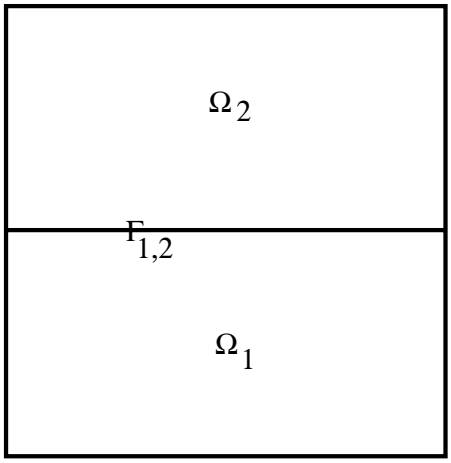

Boxes: 8x8
nb of iterations
overlapping
350
300
250
200
150
100
50
0
1
2
3
4
5
6
7
8

Algebraic domain decomposition methods without overlapping

• We consider a square domain Ω decomposed into two subdomains

• An elliptic second order PDE in a rectangle discretized by FD

• Let Ω_1 and Ω_2 be the two subdomains and $\Gamma_{1,2}$ the interface which is a mesh line



• We denote by m_1 (resp. m_2) the number of mesh lines in Ω_1 (resp. Ω_2), each mesh line having m mesh points ($m = m_1 + m_2 + 1$)

• We renumber the unknowns in Ω

Let x_1 (resp. x_2) be the vector of unknowns in Ω_1 (resp. in Ω_2) and $x_{1,2}$ be the vector of the unknowns on the interface

$$\begin{pmatrix} A_1 & 0 & E_1 \\ 0 & A_2 & E_2 \\ E_1^T & E_2^T & A_{12} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_{1,2} \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ b_{1,2} \end{pmatrix}.$$

$$E_1 = (0\ 0\ \ldots\ 0\ E_1^{m_1})^T, \qquad E_2 = (E_2^1\ 0\ \ldots\ 0)^T,$$

where $E_1^{m_1}$ and E_2^1 are diagonal matrices

• Most algebraic DD methods are based on block Gaussian elimination (or approximate block Gaussian factorization) of the matrix

• Basically, we have two possibilities depending on the fact that we can or cannot (or do not want to) solve linear systems corresponding to subproblems like

$$\begin{cases} A_1 y_1 = c_1 \\ A_2 y_2 = c_2 \end{cases}$$

"exactly" with a direct method (or with a fast solver)

Exact solvers for the subdomains

• We eliminate the unknowns x_1 and x_2 in the subdomains

This gives a reduced system for the interface unknowns

$$Sx_{1,2} = \overline{b_{1,2}},$$

with

$$\overline{b_{1,2}} = b_{1,2} - E_1^T A_1^{-1} b_1 - E_2^T A_2^{-1} b_2$$

and

$$S = A_{12} - E_1^T A_1^{-1} E_1 - E_2^T A_2^{-1} E_2.$$

The matrix S is the Schur complement of A_{12} in A

• Constructing and factoring S is costly

• A more economical solution is to solve the reduced system with matrix S on the interface with an iterative method

Theorem

For the Poisson model problem the condition number of the Schur complement is

$$\kappa(S) = O(\frac{1}{h})$$

• The product, Sp can be computed easily as

$$Sp = A_{1,2}p - E_1^T A_1^{-1} E_1 p - E_2^T A_2^{-1} E_2 p,$$

p being a vector defined on the interface

$$E_1 p = (0 \ \ldots 0 \ E_1^{m_1})^T p = (0 \ \ldots 0 \ E_1^{m_1} p)^T,$$

$$E_2 p = (E_2^1 \ 0 \ \ldots 0 \ )^T p = (E_2^1 p \ 0 \ \ldots 0)^T.$$

Then $w^1 = A_1^{-1} E_1 p$ is computed by solving

$$A_1 w^1 = E_1 p,$$

This is solving a linear system corresponding to a problem in Ω_1

• Note that only the last block of the right hand side is different from 0 and because we only need $E_1^T w^1$, the last block $w^1_{m_1}$ of the solution w^1 is what we must compute

• Similarly, $w^2 = A_2^{-1} E_2 p$ is computed by solving

$$A_2 w^2 = E_2 p,$$

a problem in Ω_2

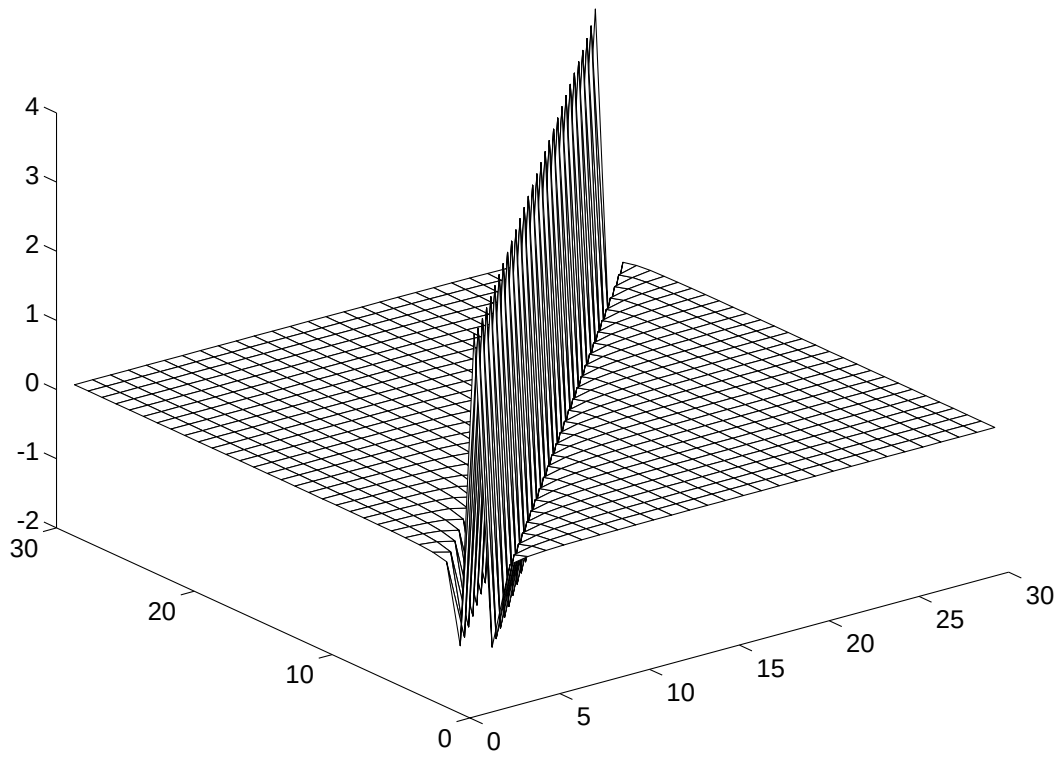
Finally, we have

$$Sp = A_{1,2} p - w^1_{m_1} - w^2_1.$$

• To improve the convergence rate of `CG` on the reduced system, a preconditioner M is needed

• The main problem is:

Find an approximation of the Schur complement S



Schur complement, 2 subdomains

Approximate solvers for the subdomains

• Let us choose M in the form

$$M = L \begin{pmatrix} M_1^{-1} & & \\ & M_2^1 & \\ & & M_{1,2}^{-1} \end{pmatrix} L^T,$$

where M_1 (resp. M_2) is of the same order as A_1 (resp. A_2) and $M_{1,2}$ is of the same order as $A_{1,2}$. L is block lower triangular

$$L = \begin{pmatrix} M_1 & & \\ 0 & M_2 & \\ E_1^T & E_2^T & M_{1,2} \end{pmatrix}$$

• At each PCG iteration, we must solve a linear system like

$$Mz = M \begin{pmatrix} z_1 \\ z_2 \\ z_{1,2} \end{pmatrix} = r = \begin{pmatrix} r_1 \\ r_2 \\ r_{1,2} \end{pmatrix}.$$

• This is done by first solving $Ly = r$, where the first parallel two steps are

$$M_1 y_1 = r_1, \quad M_2 y_2 = r_2.$$

- Finally, we solve for the interface

$$M_{1,2}y_{1,2} = r_{1,2} - E_1^T y_1 - E_2^T y_2.$$

- To obtain the solution, we have a backward solve step as

$$\begin{pmatrix} I & 0 & M_1^{-1}E_1 \\ & I & M_2^{-1}E_2 \\ & & I \end{pmatrix} \begin{pmatrix} z_1 \\ z_2 \\ z_{1,2} \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \\ y_{1,2} \end{pmatrix}.$$

This implies that $z_{1,2} = y_{1,2}$ and

$$M_1 w_1 = E_1 z_{1,2}, \quad z_1 = y_1 - w_1,$$

$$M_2 w_2 = E_2 z_{1,2}, \quad z_2 = y_2 - w_2.$$

- How to choose the approximations M_1, M_2 and $M_{1,2}$?

$$M = \begin{pmatrix} M_1 & 0 & E_1 \\ 0 & M_2 & E_2 \\ E_1^T & E_1^T & M_{1,2}^* \end{pmatrix},$$

where

$$M_{1,2}^* = M_{1,2} + E_1^T M_1^{-1} E_1 + E_2^T M_2^{-1} E_2.$$

• We would like M to be an approximation of A, it makes sense to choose

$$M_1 \approx A_1, \quad M_2 \approx A_2,$$

and

$$M_{1,2}^* \approx A_{1,2} \Longrightarrow M_{1,2} \approx A_{12} - E_1^T M_1^{-1} E_1 - E_2^T M_2^{-1} E_2.$$

• We are back to the same problem as before; that is to say, $M_{1,2}$ must be an approximation to the Schur complement S

Approximate Schur complements in the two subdomains case

- We suppose that A has a block tridiagonal structure

$$A = \begin{pmatrix} D_1 & -B_2^T & & & \\ -B_2 & D_2 & -B_3^T & & \\ & \ddots & \ddots & \ddots & \\ & & -B_{m-1} & D_{m-1} & -B_m^T \\ & & & -B_m & D_m \end{pmatrix},$$

$$A_1 = \begin{pmatrix} D_1 & -B_2^T & & & \\ -B_2 & D_2 & -B_3^T & & \\ & \ddots & \ddots & \ddots & \\ & & -B_{m_1-1} & D_{m_1-1} & -B_{m_1}^T \\ & & & -B_{m_1} & D_{m_1} \end{pmatrix},$$

$$A_2 = \begin{pmatrix} D_{m_1+2} & -B_{m_1+3}^T & & & \\ -B_{m_1+3} & D_{m_1+3} & -B_{m_1+4}^T & & \\ & \ddots & \ddots & \ddots & \\ & & -B_{m-1} & D_{m-1} & -B_m^T \\ & & & -B_m & D_m \end{pmatrix},$$

and

$$A_{1,2} = D_{m_1+1}, \quad E_1^{m_1} = -B_{m_1+1}^T, \quad E_2^1 = -B_{m_1+2}.$$

• We consider a block twisted factorization of A, the block in the center being $j = m_1 + 1$.

$$A_1 = (\Delta + L_1)\Delta^{-1}(\Delta + L_1^T),$$

where Δ is a block diagonal matrix and L_1 is the block lower triangular part of A_1 which is of block order m_1

$$A_2 = (\Sigma + L_2^T)\Sigma^{-1}(\Sigma + L_2),$$

where Σ is a block tridiagonal matrix and L_2 is the block lower triangular part of A_2

• We denote the diagonal blocks of Σ by $\Sigma_{m_1+2}, \ldots, \Sigma_n$

$$\begin{cases} \Delta_1 = D_1, \\ \Delta_i = D_i - B_i\Delta_{i-1}^{-1}B_i^T, \quad i = 2, \ldots, m_1 \end{cases}$$

and

$$\begin{cases} \Sigma_m = D_m, \\ \Sigma_i = D_i - B_{i+1}^T\Sigma_{i+1}^{-1}B_{i+1}, \quad i = m-1, \ldots, m_1+2 \end{cases}$$

Theorem

$$S = D_{m_1+1} - B_{m_1+1}\Delta_{m_1}^{-1}B_{m_1+1}^T - B_{m_1+2}^T\Sigma_{m_1+2}^{-1}B_{m_1+2}.$$

• Now we consider the eigenvalues of the Schur complement for separable problems

$$A = \begin{pmatrix} T & -I & & & \\ -I & T & -I & & \\ & \ddots & \ddots & \ddots & \\ & & -I & T & -I \\ & & & -I & T \end{pmatrix},$$

$$T = Q\Lambda Q^T,$$

Q being such that $QQ^T = I$ and Λ being a diagonal matrix whose diagonal elements are the eigenvalues of T

We denote these eigenvalues by $\lambda_l, l = 1, \ldots, m$

Theorem

The spectral decompositions of matrices Δ_i and Σ_i are

$$\Delta_i = Q\Lambda_i Q^T, \quad \Sigma_i = Q\Pi_i Q^T, \forall i$$

where Λ_i and Π_i are diagonal matrices whose diagonal elements are given for $l = 1, \ldots, m$ by

$$\begin{cases} (\Lambda_1)_{l,l} = \Lambda_{l,l} = \lambda_l, \\ (\Lambda_i)_{l,l} = \lambda_l - \dfrac{1}{(\Lambda_{i-1})_{l,l}}, \quad i = 2, \ldots, m_1 \end{cases}$$

and

$$\begin{cases} (\Pi_m)_{l,l} = \lambda_l, \\ (\Pi_i)_{l,l} = \lambda_l - \dfrac{1}{(\Pi_{i+1})_{l,l}}, \quad i = m-1, \ldots, m_1+2 \end{cases}$$

Proposition

If $\lambda_l \neq 2$, then

$$(\Lambda_i)_{l,l} = \frac{(r_l)_+^{i+1} - (r_l)_-^{i+1}}{(r_l)_+^{i} - (r_l)_-^{i}}, \quad i = 1, \ldots, m_1$$

$$(\Pi_j)_{l,l} = \frac{(r_l)_+^{m-j+2} - (r_l)_-^{m-j+2}}{(r_l)_+^{m-j+1} - (r_l)_-^{m-j+1}}, \quad j = m, \ldots, m_1+2$$

where $(r_l)_\pm = \frac{\lambda_l \pm \sqrt{\lambda_l^2 - 4}}{2}$

Theorem

The spectral decomposition of the Schur complement is

$$S = Q\Theta Q^T,$$

where Θ is a diagonal matrix whose diagonal elements θ_l are given by

$$\theta_l = \lambda_l - \frac{(r_l)_+^{m_1} - (r_l)_-^{m_1}}{(r_l)_+^{m_1+1} - (r_l)_-^{m_1+1}} - \frac{(r_l)_+^{m_2} - (r_l)_-^{m_2}}{(r_l)_+^{m_2+1} - (r_l)_-^{m_2+1}}, \quad l = 1, \ldots, m$$

where $(r_l)_\pm = \frac{\lambda_l \pm \sqrt{\lambda_l^2 - 4}}{2}$

• We do not need to explicitly know the eigenvectors Q to compute the eigenvalues

Proposition

Let $\lambda_l = 2 + \sigma_l$ and $\gamma_l = \left(1 + \frac{\sigma_l}{2} - \sqrt{\sigma_l + \frac{\sigma_l^2}{4}}\right)^2$, then

$$\theta_l = \left(\frac{1 + \gamma_l^{m_1+1}}{1 - \gamma_l^{m_1+1}} + \frac{1 + \gamma_l^{m_2+1}}{1 - \gamma_l^{m_2+1}}\right) \sqrt{\sigma_l + \frac{\sigma_l^2}{4}}, \quad \forall l = 1, \ldots, m$$

• We note that if we assume $\lambda_l > 2, \forall l$, as $(r_l)_\pm > 0$ and $(r_l)_+ > (r_l)_-$, we have $0 < \gamma_l < 1$.

• Consider the case of the model problem

$$\lambda_l = 2 + \sigma_l = 4 - 2\cos(l\pi h), \quad l = 1 \ldots, m$$

where $h = \frac{1}{m+1}$

$$\sigma_{min} = 2 - 2\cos(\pi h) = 2\pi^2 h^2 + O(h^4),$$

$$\sigma_{max} = 2 - 2\cos(\pi \frac{m}{m+1}) = 4 - O(h^2).$$

$$\theta_{min} = C_1 h + O(h^2), \quad \theta_{max} = C_2 + O(h^2),$$

where C_1 and C_2 are two constants independent of h and

$$\kappa(S) = O\left(\frac{1}{h}\right).$$

• Let us now look at the eigenvalues of S when, for a fixed h, the domains Ω_1 and Ω_2 extend to infinity

Theorem

If $\lambda_l > 2$,

$$\theta_l \to 2\sqrt{\sigma_l + \frac{\sigma_l^2}{4}} \text{ when } \; m_i \to \infty, \; i = 1, 2.$$

Let Σ_2 be the diagonal matrix of the eigenvalues

$$\sigma_i = 2 - 2\cos(i\pi h), \quad i = 1, \ldots, m$$

and Q such that

$$q_{i,j} = \sqrt{\frac{2}{m+1}} \sin(ij\pi h), \quad i, j = 1, \ldots, m.$$

Dryja's preconditioner:

Let T_2 be the matrix corresponding to finite difference discretization of the one–dimensional Laplacian, $T_2 = Q_2 \Sigma_2 Q_2^T$

$$M_D = Q_2 \sqrt{\Sigma_2} Q_2^T.$$

Golub and Mayers' preconditioner:

$$M_{GM} = Q_2 \sqrt{\Sigma_2 + \frac{\Sigma_2^2}{4}} Q_2^T.$$

Theorem

For the Poisson model problem $\kappa(M^{-1}S) = O(1)$, for Dryja and Golub–Mayers

The Neumann–Dirichlet preconditioner

• This preconditioner was introduced by Bjørstad and Widlund

$$\begin{pmatrix} A_1 & 0 & E_1 \\ 0 & A_2 & E_2 \\ E_1^T & E_2^T & A_{1,2} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_{1,2} \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ b_{1,2} \end{pmatrix},$$

• We can distinguish what in $A_{1,2}$ comes from subdomain Ω_1 and what comes from Ω_2

$$A_{1,2} = A_{1,2}^{(1)} + A_{1,2}^{(2)}.$$

Since we know that

$$S = A_{1,2} - E_1^T A_1^{-1} E_1 - E_2^T A_2^{-1} E_2,$$

we can define

$$S^{(1)} = A_{1,2}^{(1)} - E_1^T A_1^{-1} E_1, \quad S^{(2)} = A_{1,2}^{(2)} - E_2^T A_2^{-1} E_2,$$

and $S = S^{(1)} + S^{(2)}$

• The Neumann–Dirichlet preconditioner is defined as

$$M_{ND} = S^{(1)}.$$

Note, that we could also have chosen $S^{(2)}$ instead of $S^{(1)}$

Theorem

For the Poisson model problem $\kappa(M_{ND}^{-1}S) = O(1)$.

Note that if $m_2 = m_1$, the preconditioner is exact

Why is this preconditioner called "Neumann–Dirichlet"?

$$S^{(1)} y_{12} = (A_{1,2}^{(1)} - E_1^T A_1^{-1} E_1) y_{1,2} = c_{1,2},$$

is equivalent to solving

$$\begin{pmatrix} A_1 & E_1 \\ E_1^T & A_{1,2}^{(1)} \end{pmatrix} \begin{pmatrix} y_1 \\ y_{1,2} \end{pmatrix} = \begin{pmatrix} 0 \\ c_{1,2} \end{pmatrix}.$$

For second order elliptic PDEs, it is easy to see that this is simply solving a problem in Ω_1 with given Neumann boundary conditions on the interface

When the solution is known on the interface, it is enough to solve a Dirichlet problem in Ω_2

The Neumann–Neumann preconditioner

- This preconditioner was introduced by Le Tallec

$$M_{NN}^{-1} = \frac{1}{2}\left[(S^{(1)})^{-1} + (S^{(2)})^{-1}\right]$$

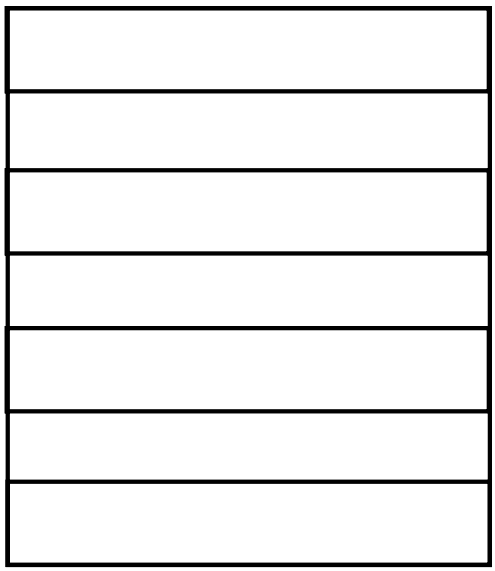
Note that we directly define the inverse of the preconditioner as an average of inverses of "local" (to each subdomain) inverses of Schur complements.

Theorem

For the Poisson model problem $\kappa(M_{NN}^{-1}S) = O(1)$.

Approximations of Schur complements with many subdomains

• We extend the results for two subdomains by considering now the domain Ω being divided into k strips



• We denote by x_i the unknowns in subdomain Ω_i and by $x_{i,i+1}$ the unknowns on the interface between Ω_i and Ω_{i+1}

$$A = \begin{pmatrix} A_1 & & & & & C_1 & & & \\ & A_2 & & & & E_2 & C_2 & & \\ & & A_3 & & & & E_3 & \ddots & \\ & & & \ddots & & & & \ddots & C_{k-1} \\ & & & & A_k & & & & E_k \\ C_1^T & E_2^T & & & & A_{1,2} & & & \\ & C_2^T & E_3^T & & & & B_{2,3} & & \\ & & \ddots & \ddots & & & & \ddots & \\ & & & C_{k-1}^T & E_k^T & & & & A_{k-1,k} \end{pmatrix},$$

$$x = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_k \\ x_{1,2} \\ x_{2,3} \\ \vdots \\ x_{k-1,k} \end{pmatrix}, \quad b = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \\ \vdots \\ b_k \\ b_{1,2} \\ b_{2,3} \\ \vdots \\ b_{k-1,k} \end{pmatrix}.$$

$$A_i = \begin{pmatrix} D_i^1 & (A_i^2)^T & & & \\ A_i^2 & D_i^2 & (A_i^3)^T & & \\ & \ddots & \ddots & \ddots & \\ & & A_i^{m_i} & D_i^{m_i-1} & (A_i^{m_i})^T \\ & & & A_i^{m_i} & D_i^{m_i} \end{pmatrix}, \quad i = 1, \ldots, k$$

Proposition

The Schur complement matrix S is block tridiagonal and we denote

$$S = \begin{pmatrix} A'_{12} & F_2^T & & & \\ F_2 & A'_{23} & F_3^T & & \\ & \ddots & \ddots & \ddots & \\ & & F_{k-2} & A'_{k-2,k-1} & F_{k-1}^T \\ & & & F_{k-1} & A'_{k-1,k} \end{pmatrix}.$$

$$A'_{i,i+1} = A_{i,i+1} - C_i^T A_i^{-1} C_i - E_{i+1}^T A_{i+1}^{-1} E_{i+1},$$

$$F_i = -C_i^T A_i^{-1} E_i.$$

Theorem

The eigenvalues of S are

$$\omega_{l,j} = \frac{2}{1-\gamma_l^{m+1}} \sqrt{\sigma_l + \frac{\sigma_l^2}{4}} \left(1 + \gamma_l^{m+1} - 2\gamma_l^{\frac{m+1}{2}} \cos\left(\frac{j\pi}{k}\right)\right),$$
$$l = 1, \ldots, m \quad j = 1, \ldots k-1$$

$$\kappa(S) = O\left(\frac{k}{h}\right).$$

- If k is fixed and $h \to 0$, κ increases as $1/h$
- If h is given and we increase the number of subdomains, κ increases as k
- The problem is to define preconditioners for S

• First idea: use block diagonal preconditioners, the diagonal blocks arising from the two subdomains case

• Ex: Dryja's; This preconditioner removes the h dependency but not that on k and, in fact, we have

$$\kappa(M_D^{-1}S) = O(k^2).$$

• The same result is true for the Golub and Mayers' preconditioner although the condition number is a little smaller

• It is more difficult to generalize the Neumann–Dirichlet preconditioner to many subdomains

If we do this in the obvious way, we have $\kappa = O(k^2)$

• The Neumann–Neumann preconditioner can be easily extended to many subdomains, the inverses of partial Schur complements have to be weighted by the inverse of the number of subdomains which share a given node

Inexact subdomain solvers

• If we cannot solve exactly for the subproblems, we are not able to use an iterative method with S as we cannot compute the matrix$\times$vector product Sv

• We need a global parallel preconditioner

$$M = L \begin{pmatrix} M_1^{-1} & & & & & & \\ & M_2^{-1} & & & & & \\ & & \ddots & & & & \\ & & & M_k^{-1} & & & \\ & & & & M_{1,2}^{-1} & & \\ & & & & & \ddots & \\ & & & & & & M_{k-1,k}^{-1} \end{pmatrix} L^T$$

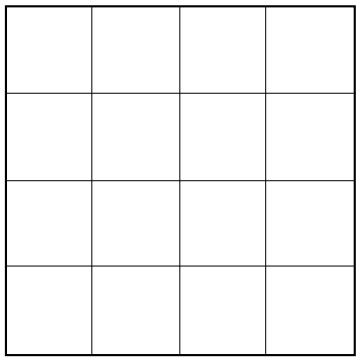
$$L = \begin{pmatrix}
M_1 & & & & & & & & \\
& M_2 & & & & & & & \\
& & \ddots & & & & & & \\
& & & \ddots & & & & & \\
& & & & M_k & & & & \\
C_1^T & E_2^T & & & & M_{1,2} & & & \\
& C_2^T & E_3^T & & & H_2 & M_{2,3} & & \\
& & \ddots & & & & \ddots & \ddots & \\
& & & C_{k-1}^T & E_k^T & & & H_{k-1} & M_{k-1,k}
\end{pmatrix}$$

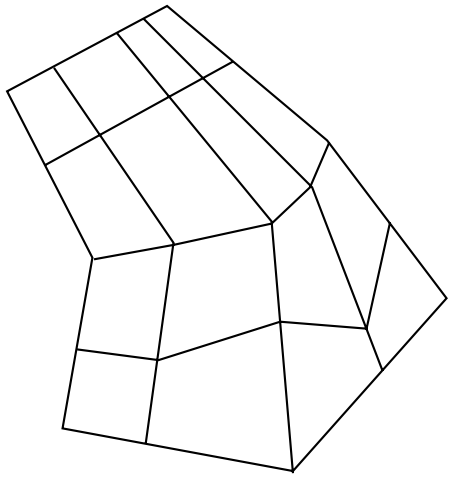
- The matrices M_i can be chosen as for the two subdomains case
- For matrices $M_{i,i+1}$ and H_i, we have many possible choices

Domain decomposition with boxes

• A domain decomposition with strips can be done for more general domains by finding pseudo–peripheral nodes and constructing the level structure corresponding to one of these nodes

• However, except for very large problems, when partitioning in this way, we cannot use many subdomains. A way to partition with many subdomains is to use so–called boxes





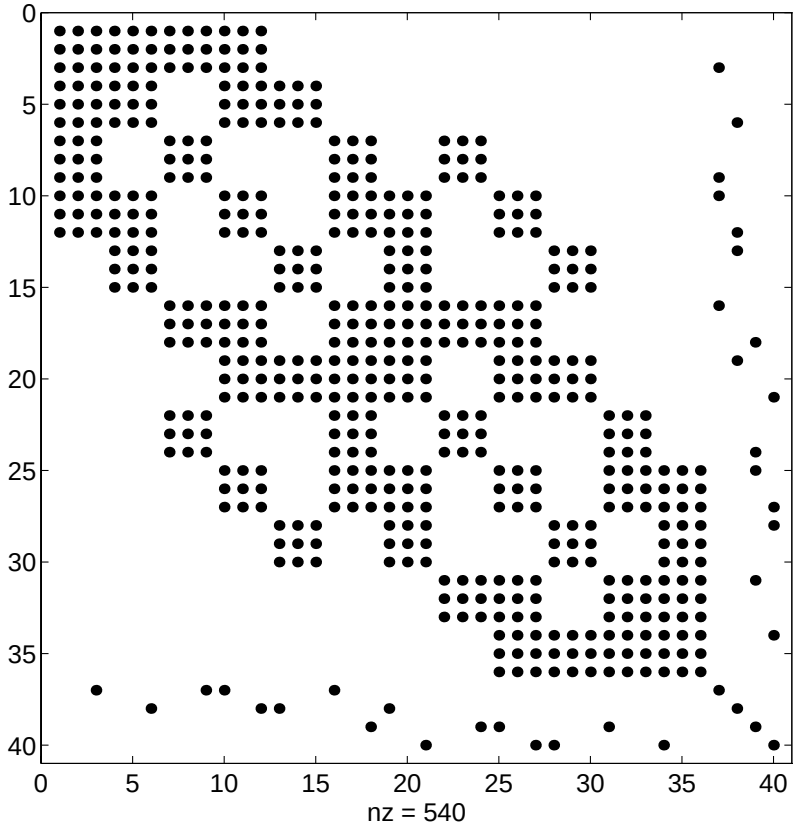
• Interfaces = two kinds of sets: edges and cross points

• We use an index E_i for the edges, $E = \cup E_i$ and V for the vertices

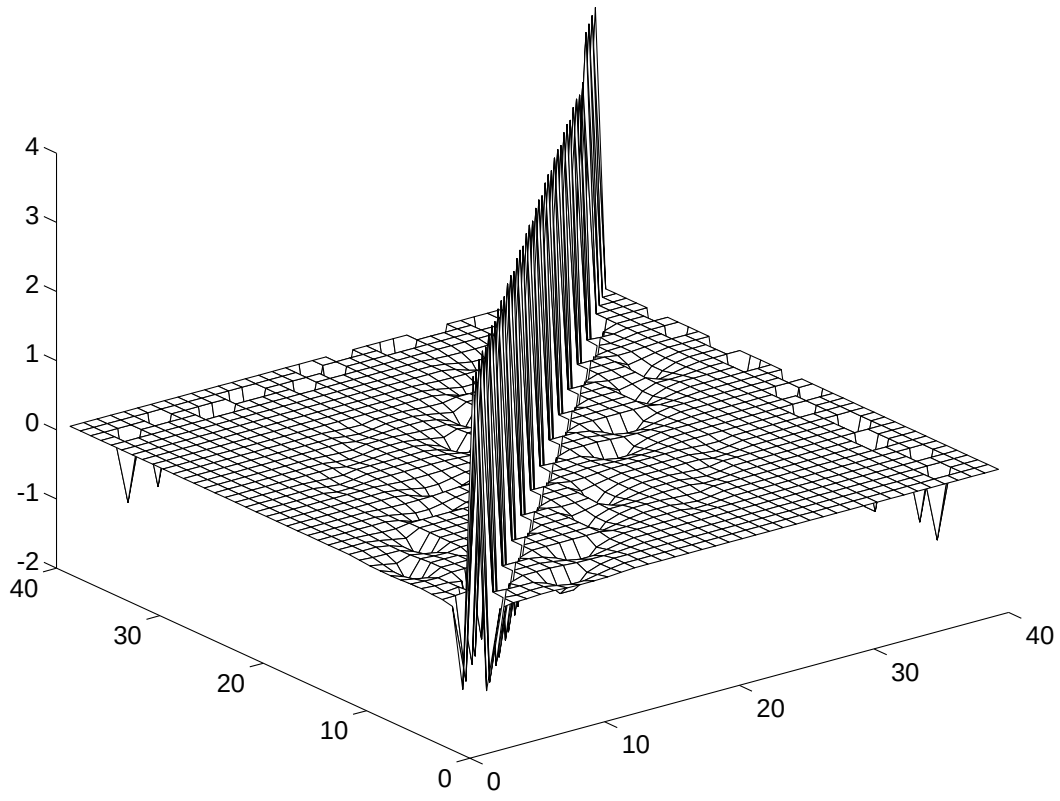
• Example: model problem with 121 nodes and 9 subdomains arranged as 3×3, each subdomain having 3×3 nodes.

61	62	63	114	70	71	72	117	79	80	81
58	59	60	113	67	68	69	116	76	77	78
55	56	57	112	64	65	66	115	73	74	75
103	104	105	120	106	107	108	121	109	110	111
34	35	36	99	43	44	45	102	52	53	54
31	32	33	98	40	41	42	101	49	50	51
28	29	30	97	37	38	39	100	46	47	48
88	89	90	118	91	92	93	119	94	95	96
7	8	9	84	16	17	18	87	25	26	27
4	5	6	83	13	14	15	86	22	23	24
1	2	3	82	10	11	12	85	19	20	21

- With this ordering the structure of S is given as



S looks like



The Bramble, Pasciak and Schatz preconditioner

$$a(u,v) = (f,v),\ \forall v \in V$$

where a is a coercive bilinear form arising from a second order elliptic PDE, V being a Hilbert space, say $H_0^1(\Omega)$ for homogeneous Dirichlet boundary conditions

• We want to construct another spectrally equivalent bilinear form $b(u,v)$ such that

$$\lambda_0 b(v,v) \leq a(v,v) \leq \lambda_1 b(v,v),\ \forall v \in V$$

and to use b as a preconditioner

• Ω is divided into non–overlapping subdomains Ω_k, the edges between two subdomains being denoted as $\Gamma_{l,m}$

• Another intermediate form is introduced to eventually allow for some averaging of the coefficients,

$$\tilde{a}(u,v) = \sum_k \sum_{i,j} \int_{\Omega_k} a_{i,j}^k \frac{\partial u}{\partial x_i} \frac{\partial v}{\partial x_j}\ dx = \sum_k \tilde{a}_k(u,v).$$

• The method separates interior, edges and vertices unknowns in the following way:

$$u = u_P + u_H,$$

where u_P is in $\sum_{\oplus} V^0(\Omega_k)$ where functions in $V^0(\Omega_k)$ have homogeneous Dirichlet boundary conditions and

$$u_P = 0 \text{ on } \Gamma_{l,m}.$$

u_P is defined by

$$\tilde{a}_k(u_P, \phi) = \tilde{a}_k(u, \phi), \ \forall \phi \in V^0(\Omega_k).$$

• This takes care of the right hand side and u_H is defined by

$$\tilde{a}_k(u_H, \phi) = 0 \ \forall \phi \in V^0(\Omega_k).$$

• The method goes one step further and decomposes u_H on the interfaces as

$$u_H = u_E + u_V,$$

where u_E stands for edge unknowns, u_V for vertices unknowns, $u_V(v_j) = u(v_j)$ and $u_V|_{\Gamma_{ij}}$ is linear, $u_E(v_j) = 0$

• BPS defined an operator l_0 on the edges: $V^0(\Gamma_{i,j}) \to V^0(\Gamma_{i,j})$ by

$$\int_{\Gamma_{i,j}} c^{-1} l_0(w)\phi = \int_{\Gamma_{i,j}} cw'\phi',\ \forall \phi \in V^0(\Gamma_{i,j}),$$

where c is piecewise constant

This defines something which behaves like the one dimensional Laplace operator

$$\begin{aligned} b(w,\phi) =& \tilde{a}(u_P, \phi_P) \\ &+ \sum_{\Gamma_{i,j}} \int_{\Gamma_{i,j}} \alpha_{i,j} c^{-1} l_0^{1/2}(u_E)\phi_E \\ &+ \sum_{\Gamma_{i,j}} (u_V(v_i) - u_V(v_j))(\phi_V(v_i) - \phi_V(v_j)). \end{aligned}$$

• The basis functions that are used are the usual ones for the interior nodes, one dimensional hat functions for the edges (vanishing at the vertices) and functions which are linear on each edge, 1 at one vertex, 0 at the other ones for the vertices

• BPS requires us to perform the following steps:

1) solve Dirichlet problems on each subdomain in parallel for $\longrightarrow u_P$,

2) solve one dimensional edge equations in parallel for $\longrightarrow u_E$

3) solve a coarse mesh system on vertices for $\longrightarrow u_V$. From u_E and u_V we obtain the boundary values of u_H

4) solve Dirichlet problems on each subdomain in parallel for $\longrightarrow u_H$. The solution is $u_P + u_H$

Theorem

Under suitable hypotheses, the condition number for the preconditioned system in two dimensions is

$$\kappa \leq C \left(1 + \log^2 \left(\frac{H}{h}\right)\right),$$

where H is the coarse mesh size.

Variants of the BPS preconditioner can also be denoted as

$$M^{-1}v = \sum_{edges} R_{E_i}^T (\alpha_i M_i)^{-1} R_{E_i} v + R_H^T A_H^{-1} R_H v,$$

where R_{E_i} denotes the restriction to the edge E_i and R_H is a weighted restriction onto the coarse mesh, M_i being one of the preconditioners for two subdomain case: either Dryja or Golub–Mayers

Vertex space preconditioners

• A way to improve on BPS is to allow for some coupling between the vertices and the edge nodes

• Some points are considered around each vertex on each of the edges Let V_k be this set of points. Then the preconditioner is defined as

$$M^{-1}v =$$

$$R_H^T A_H^{-1} R_H v + \sum_{edges} R_{E_i}^T (M_{E_i})^{-1} R_{E_i} v + \sum_{vertices} R_{V_k}^T (M_{V_k})^{-1} R_{V_k} v$$

This includes some coupling between neighboring edges

• The edge preconditioner can be chosen as a weighting of Dryja's or Golub–Mayers' preconditioners

• The restriction to the edges is tridiagonal and an edge is only linked to the crosspoint and to the two nodes adjacent to the crosspoint on the neighboring edges

• If enough points are used around each vertex, then the condition number is independent of h and of the number of subdomains

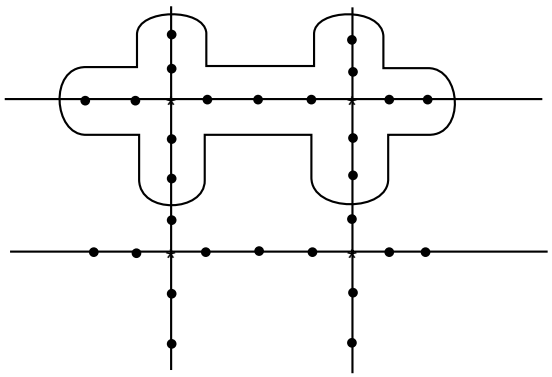
The vertex space algorithm was developed by B. Smith

• In his Ph.D. thesis L. Carvalho considered some preconditioners whose spirit is quite close to the vertex space preconditioners

• Because they involve some kind of overlapping between the edge and vertex parts, they are denoted as algebraic additive Schwarz (AAS)

• He studied several local block preconditioners for the subdomains and several coarse space preconditioners

• For one of the local preconditioners, the main difference with the vertex space preconditioner is that the edge and the adjacent vertices are considered together



• Another proposal was to consider the complete boundary of one subdomain, to be able to retrieve all the couplings between the edge nodes and the vertices when the interior nodes are eliminated

• It is necessary to add a coarse space component in the algorithm

• A restriction operator R_0 is defined (depending on the choice of the coarse part of the preconditioner)

• The coarse component of the preconditioner is defined as $R_0^T A_0^{-1} R_0$ where A_0 is the Galerkin coarse space operator $A_0 = R_0 S R_0^T$

• Several possibilities were considered:

◦ i) a subdomain–based coarse space where all the boundary points of a subdomain are considered. The coarse space is spanned by vectors which have non–zero components for the points around a subdomain, for all subdomains.

◦ ii) a vertex–based coarse space where the vertices and some few adjacent edge points are considered.

◦ iii) an edge–based coarse space where the points of an edge and the adjacent vertices are considered.

• When combining these coarse space preconditioners with the local parts, a preconditioner for which the condition number is insensitive to the mesh size or the number of subdomains is obtained except for very highly anisotropic problems

Numerical experiments

- 16×16 mesh for each subdomain
- Pb 1: Poisson equation

nb of subd	4×4	8×8	16×16
M_E	13	28	51
M_{VE}	12	22	40
M_S	11	19	32
M_{C-E}	9	11	11
M_{C-VE}	10	12	12
M_{C-S}	10	10	11

• Pb 2: Isotropic discontinuous pb on the Scottish flag, coefficients $1, 10^3, 10^{-3}$

nb of subd	4×4	8×8	16×16
M_{C-E}	11	11	15
M_{C-VE}	12	12	16
M_{C-S}	10	11	14

• Pb 3: Anisotropic and discontinuous pb on the Scottish flag, coefficient 1 in x, same as before in y

nb of subd	4×4	8×8	16×16
M_{C-E}	25	65	103
M_{C-VE}	23	80	141
M_{C-S}	20	43	79