AILU, Sweeping Preconditioner Domain Decomposition Methods Absorbing Boundary Conditions and PML: how does all this fit together ?

Martin J. Gander
martin.gander@unige.ch

In collaboration with Hui Zhang

University of Geneva

July 2013

INRIA Presentations 2007-2013

2007 PARALLÉLISME EN TEMPS: Que pouvons-nous prévoir du futur éloigné sans tout connaître du futur proche ?

2008 Les méthodes de Schwarz au fil de l'histoire

2009 Classical Multigrid Methods: Efficient for Diffusive Problems but Ineffective for Wave Propagation Problems

2010 Euler, Ritz, Galerkin, Courant: On the Road to the Finite Element Method

2011 Techniques adaptatives pour l'intégration des équations différentielles en espace temps

2012 A Different Access to Krylov Methods: Extrapolation Methods

2013 AILU, Sweeping Preconditioner, Domain Decomposition Methods, Absorbing Boundary Conditions and PML: how does all this fit together ?

Which Equation is the Helmholtz Equation?

$(\Delta + k^2)u = f$ **or** $(\Delta - \eta)u = f$, $\eta > 0$

Which Equation is the Helmholtz Equation?

$(\Delta + k^2)u = f$ **or** $(\Delta - \eta)u = f,\ \eta > 0$

Helmholtz (1871), Hertz ...

Which Equation is the Helmholtz Equation?

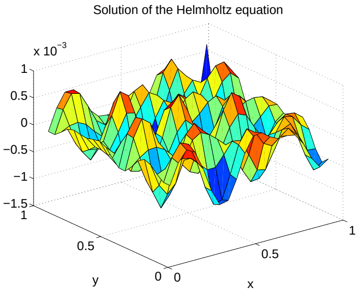
$(\Delta + k^2)u = f$ **or** $(\Delta - \eta)u = f,\ \eta > 0$

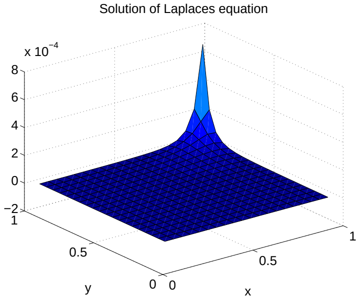
Helmholtz (1871), Hertz ... | Leslie, Bryant, McAveney (1973)
Rosmond and Faulkner (1976)

Which Equation is the Helmholtz Equation?

$(\Delta + k^2)u = f$ **or** $(\Delta - \eta)u = f,\ \eta > 0$

Helmholtz (1871), Hertz ... | Leslie, Bryant, McAveney (1973) Rosmond and Faulkner (1976)

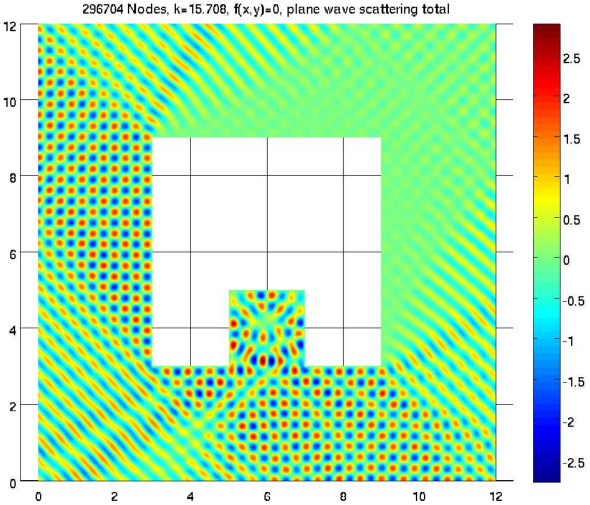




Which Equation is the Helmholtz Equation?

$(\Delta + k^2)u = f$ **or** $(\Delta - \eta)u = f,\ \eta > 0$

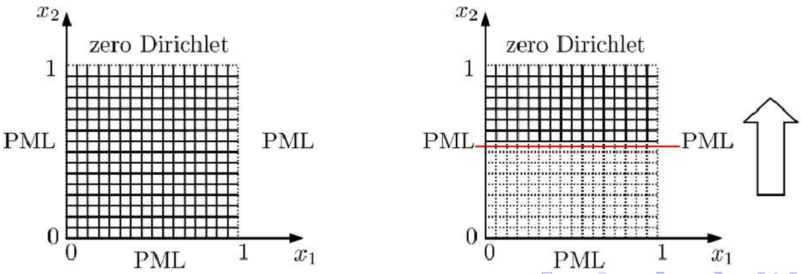
Helmholtz (1871), Hertz ... Leslie, Bryant, McAveney (1973)
Rosmond and Faulkner (1976)



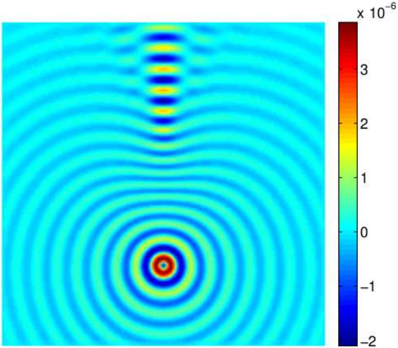
Sweeping Preconditioner

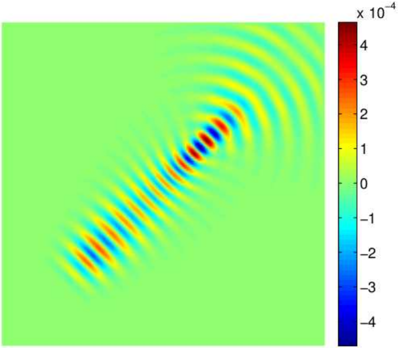
Enquist, Ying 2010: Sweeping Preconditioner for the Helmholtz Equation

"The paper introduces the sweeping preconditioner, which is highly efficient for iterative solutions of the variable coefficient Helmholtz equation including very high frequency problems. The first central idea of this novel approach is to construct an approximate factorization of the discretized Helmholtz equation by sweeping the domain layer by layer, starting from an absorbing layer or boundary condition."



Numerical Experiment (Enquist, Ying 2010)





				Test 1		Test 2	
$\omega/(2\pi)$	q	$N = n^3$	T_{setup}	N_{iter}	T_{solve}	N_{iter}	T_{solve}
5	8	39^3	4.80e+00	11	4.53e+00	11	4.63e+00
10	8	79^3	6.37e+01	11	4.92e+01	11	4.93e+01
20	8	159^3	8.27e+02	12	5.53e+02	12	5.94e+02

Source Transfer Domain Decomposition

Chen, Xiang 2012: A Source Transfer Domain Decomposition Method for Helmholtz Equations in Unbounded Domain

"The method is based on the decomposition of the domain into non-overlapping layers and the idea of source transfer which transfers the sources equivalently layer by layer so that the solution in the final layer can be solved using a PML method defined locally outside the last two layers."

Algorithm 2.1. (Source transfer for PML problem in $\mathbb{R}^2$)

1° *Let* $\bar{f}_1 = f_1$ *in* $\mathbb{R}^2$;

2° *For* $i = 1, 2, \cdots, N-2$, *compute* $\bar{f}_{i+1} = f_{i+1} + \Psi_{i+1}(\bar{f}_i)$, *where*

$$\Psi_{i+1}(\bar{f}_i) = \begin{cases} J^{-1}\nabla\cdot(A\nabla(\beta_{i+1}u_i)) + k^2(\beta_{i+1}u_i) & in \quad \Omega_{i+1}, \\ 0 & in \quad \mathbb{R}^2\backslash\bar{\Omega}_{i+1}, \end{cases}$$

and u_i *is given by*

$$u_i(x) = \int_{\Omega_i} \bar{f}_i(y)\tilde{G}(x,y)dy.$$

Numerical Experiment (Chen, Xiang 2012)

$k/2\pi$	q	DOF	N_{iter}	T_{solve}
30	10	300^2	5	2.43
60	10	600^2	5	9.77
120	10	1200^2	6	44.58
240	10	2400^2	7	225.15
480	10	4800^2	8	1122.37
960	10	9600^2	12	8047.68

TABLE 4.1

Numerical results for different wave numbers k when $q = 10$, where N_{tier} is the number of iterations of the preconditioned GMRES method and T_{solve} is the overall solution time in seconds.

$k/2\pi$	q	NOF	N_{iter}	T_{solve}
30	20	600^2	3	8.11
60	20	1200^2	3	26.58
120	20	2400^2	4	127.94
240	20	4800^2	5	676.45

TABLE 4.2

Numerical results for different wave numbers k when $q = 20$, where N_{tier} is the number of iterations of the preconditioned GMRES method and T_{solve} is the overall solution time in seconds.

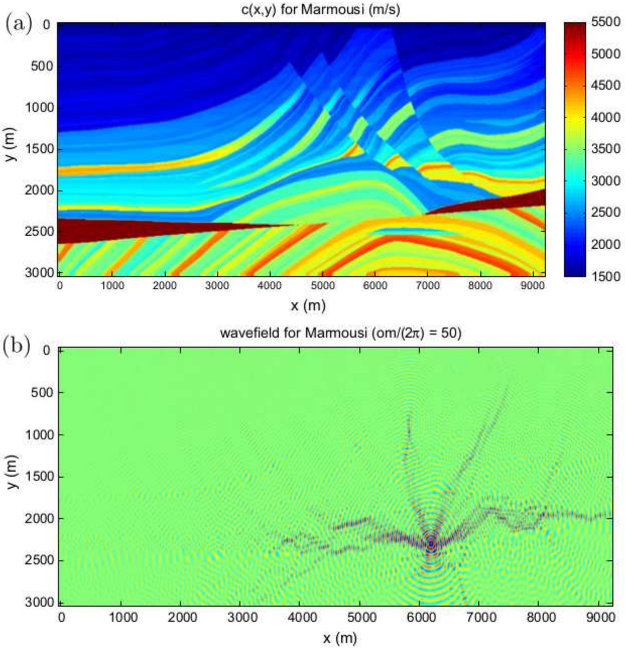
PML Domain Decomposition

Stolk 2013: A rapidly converging domain decomposition method for the Helmholtz equation

"A new domain decomposition method is introduced for the heterogeneous 2-D and 3-D Helmholtz equations. Transmission conditions based on the perfectly matched layer (PML) are derived that avoid artificial reflections and match incoming and outgoing waves at the subdomain interfaces."

"Our most remarkable finding concerns the situation where the domain is split into many thin layers along one of the axes, say J subdomains numbered from 1 to J. Following [3] we will also call these quasi 2-D subdomains. Generally, an increase in the number of subdomains leads to an increase in the number of iterations required for convergence. Here we propose and study a method where the number of iterations is essentially independent of the number of subdomains."

Numerical Experiments (Stolk 2013)



Numerical Experiments (Stolk 2013)

Table 2
Convergence results for Example 1. Displayed is the number of iterations for reduction of the residual by 10^{-6} as a function of the size of the domain and the number of subdomains.

$N_x \times N_y$	h (m)	$\frac{\omega}{2\pi}$ (Hz)	Number of x-subdomains				
			3	10	30	100	300
600×212	16	12.5	4	5	6		
1175×400	8	25	5	6	7		
2325×775	4	50	6	6	7	9	
4625×1525	2	100	6	6	7	8	
9225×3025	1	200		7	8	9	13 (8) (*)

(*) 13 was obtained for $w_{\text{pml}} = 5$, 8 for $w_{\text{pml}} = 6$.

Table 5
Comparison of convergence between Robin and PML-based transmission conditions for a constant medium.

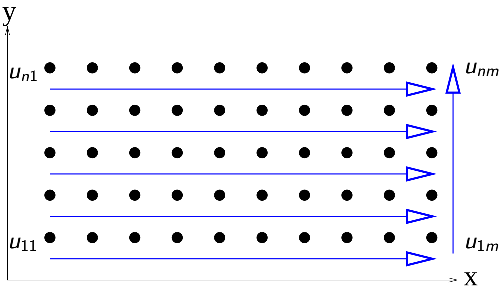
$N_x \times N_y$	h	$\frac{\omega}{2\pi}$	J	PML	Robin
100×100	0.01	10	10	3	9
200×200	0.005	20	20	4	13
400×400	0.0025	40	40	4	20
800×800	0.00125	80	80	5	42
1600×1600	0.000625	160	160	7	103

Table 6
Comparison of convergence between Robin and PML-based transmission conditions for the random medium displayed in Fig. 3.

$N_x \times N_y$	h	$\frac{\omega}{2\pi}$	J	PML	Robin
100×100	0.01	7.14	10	7	11
200×200	0.005	14.29	20	6	14
400×400	0.0025	28.57	40	6	20
800×800	0.00125	57.14	80	7	34
1600×1600	0.000625	114.3	160	8	74

What are these new Methods ?

$$(\Delta + k^2)u = f \qquad \text{in } \Omega = (0,1)\times(0,\pi)$$



$$A\mathbf{u} = \begin{bmatrix} D_1 & L_{1,2} & & \\ L_{2,1} & D_2 & \ddots & \\ & \ddots & \ddots & L_{n-1,n} \\ & & L_{n,n-1} & D_n \end{bmatrix} \begin{pmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \\ \vdots \\ \mathbf{u}_n \end{pmatrix} = \begin{pmatrix} \mathbf{f}_1 \\ \mathbf{f}_2 \\ \vdots \\ \mathbf{f}_n \end{pmatrix}$$

Block Factorization

$$A = \begin{bmatrix} D_1 & L_{1,2} & & \\ L_{2,1} & D_2 & L_{2,3} & \\ & \ddots & \ddots & \ddots \\ & & L_{n,n-1} & D_n \end{bmatrix}$$

$$= \begin{bmatrix} T_1 & & & \\ L_{2,1} & T_2 & & \\ & \ddots & \ddots & \\ & & L_{n,n-1} & T_n \end{bmatrix} \begin{bmatrix} I & T_1^{-1}L_{1,2} & & \\ & I & T_2^{-1}L_{2,3} & \\ & & \ddots & \ddots \\ & & & I \end{bmatrix}$$

$$T_i = \begin{cases} D_1 & i = 1, \\ D_i - L_{i,i-1}T_{i-1}^{-1}L_{i-1,i} & 1 < i \leq n. \end{cases}$$

Block Factorization

$$A = \begin{bmatrix} D_1 & L_{1,2} & & \\ L_{2,1} & D_2 & L_{2,3} & \\ & \ddots & \ddots & \ddots \\ & & L_{n,n-1} & D_n \end{bmatrix}$$

$$= \begin{bmatrix} T_1 & & & \\ L_{2,1} & T_2 & & \\ & \ddots & \ddots & \\ & & L_{n,n-1} & T_n \end{bmatrix} \begin{bmatrix} T_1^{-1} & & & \\ & T_2^{-1} & & \\ & & \ddots & \\ & & & T_n^{-1} \end{bmatrix} \begin{bmatrix} T_1 & L_{1,2} & & \\ & T_2 & \ddots & \\ & & \ddots & L_{n-1,n} \\ & & & T_n \end{bmatrix}$$

$$T_i = \begin{cases} D_1 & i = 1, \\ D_i - L_{i,i-1} T_{i-1}^{-1} L_{i-1,i} & 1 < i \leq n. \end{cases}$$

Block Factorization

$$A = \begin{bmatrix} D_1 & L_{1,2} & & \\ L_{2,1} & D_2 & L_{2,3} & \\ & \ddots & \ddots & \ddots \\ & & L_{n,n-1} & D_n \end{bmatrix}$$

$$= \begin{bmatrix} T_1 & & & \\ L_{2,1} & T_2 & & \\ & \ddots & \ddots & \\ & & L_{n,n-1} & T_n \end{bmatrix} \begin{bmatrix} T_1^{-1} & & & \\ & T_2^{-1} & & \\ & & \ddots & \\ & & & T_n^{-1} \end{bmatrix} \begin{bmatrix} T_1 & L_{1,2} & & \\ & T_2 & \ddots & \\ & & \ddots & L_{n-1,n} \\ & & & T_n \end{bmatrix}$$

$$T_i = \begin{cases} D_1 & i = 1, \\ D_i - L_{i,i-1} T_{i-1}^{-1} L_{i-1,i} & 1 < i \leq n. \end{cases}$$

For symmetric PDEs, this factorization is called the block LDL^T factorization

Forward and Backward Substitution

$$A\mathbf{u} = LDL^T\mathbf{u} = \mathbf{f} \quad \Longleftrightarrow \quad L\mathbf{v} = \mathbf{f}, \quad L^T\mathbf{u} = D^{-1}\mathbf{v}$$

Forward substitution:

$$\begin{bmatrix} T_1 & & & \\ L_{2,1} & T_2 & & \\ & \ddots & \ddots & \\ & & L_{n,n-1} & T_n \end{bmatrix} \begin{pmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \\ \vdots \\ \mathbf{v}_n \end{pmatrix} = \begin{pmatrix} \mathbf{f}_1 \\ \mathbf{f}_2 \\ \vdots \\ \mathbf{f}_n \end{pmatrix}$$

$$\begin{aligned} \mathbf{v}_1 &= T_1^{-1}\mathbf{f}_1 \\ \mathbf{v}_2 &= T_2^{-1}(\mathbf{f}_2 - L_{2,1}\mathbf{v}_1) = T_2^{-1}(\mathbf{f}_2 - L_{2,1}T_1^{-1}\mathbf{f}_1) = T_2^{-1}\bar{\mathbf{f}}_2 \\ \mathbf{v}_3 &= T_3^{-1}(\mathbf{f}_3 - L_{3,2}\mathbf{v}_2) = T_3^{-1}(\mathbf{f}_3 - L_{3,2}T_2^{-1}\bar{\mathbf{f}}_2) = T_3^{-1}\bar{\mathbf{f}}_3 \\ \vdots & \qquad\qquad\qquad \vdots \qquad\qquad\qquad\qquad \vdots \end{aligned}$$

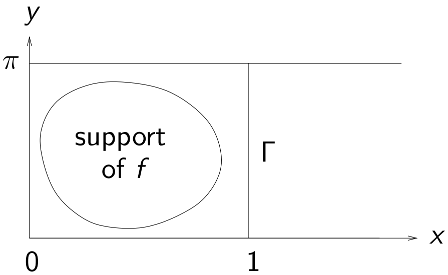
$$\boxed{\textbf{Source transfer: } \bar{\mathbf{f}}_j = \mathbf{f}_j - L_{j,j-1}T_{j-1}^{-1}\bar{\mathbf{f}}_{j-1} \text{ and } \mathbf{v}_n = \mathbf{u}_n}$$

Transparent Boundary Conditions

For the model problem

$$\begin{aligned} (\eta - \Delta)u &= f \qquad \text{in } \Omega = (0,\infty)\times(0,\pi) \\ u(x,0) = u(x,\pi) &= 0 \\ u(0,y) &= 0 \end{aligned}$$

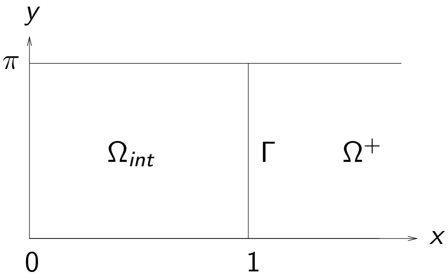
with f compactly supported in $\Omega_{int} = (0,1)\times(0,\pi)$, and u bounded at infinity.



In order to solve this problem on a computer, the computational domain needs to be truncated at $x = 1$, and an artificial boundary condition needs to be imposed.

Construction of Transparent Boundary Conditions

Based on the decomposition of $\Omega = \Omega_{int} \cup \Gamma \cup \Omega^+$



and the equivalent coupled problems

$$\begin{array}{rcll} (\eta - \Delta)v & = & f & \text{in } \Omega_{int} \\ \partial_x v & = & \partial_x v^+ & \text{on } \Gamma \\ v^+ & = & v & \text{on } \Gamma \\ (\eta - \Delta)v^+ & = & 0 & \text{in } \Omega^+ \end{array}$$

with homogeneous conditions at $y = 0$ and $y = \pi$.

Equivalent Solution on Ω_{int}

The problem on Ω^+ is independent of the source f. For any Dirichlet data g on Γ, we can solve it to obtain $v^+(x,y,g)$. Taking a normal derivative on Γ we obtain

$$\mathrm{DtN}(g) := \partial_x v^+(1,y,g).$$

We can thus solve only on Ω_{int} the problem

$$\begin{array}{rcll} (\eta-\Delta)v & = & f & \text{in } \Omega_{int} \\ \partial_x v & = & \mathrm{DtN}(v) & \text{on } \Gamma \end{array}$$

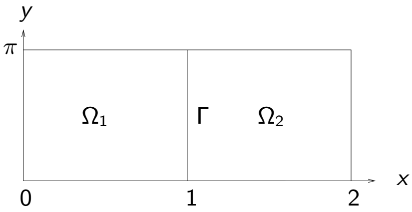
since this v is identical to the solution v of the coupled problem

$$\begin{array}{rcll} (\eta-\Delta)v & = & f & \text{in } \Omega_{int} \\ \partial_x v & = & \partial_x v^+ & \text{on } \Gamma \\ v^+ & = & v & \text{on } \Gamma \\ (\eta-\Delta)v^+ & = & 0 & \text{in } \Omega^+ \end{array}$$

DtN is called the Dirichlet to Neumann operator, and $\partial_x v - \mathrm{DtN}(v) = 0$ transparent boundary condition (TBC)

An Optimal Schwarz Algorithm

Based on the decomposition of $\Omega = \Omega_1 \cup \Gamma \cup \Omega_2$



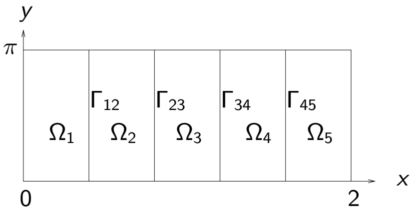
The optimal Schwarz algorithm is

$$
\begin{array}{rcll}
(\eta - \Delta) u_1^n & = & f & \text{in } \Omega_1 \\
\partial_x u_1^n - \mathrm{DtN}_2(u_1^n) & = & \partial_x u_2^{n-1} - \mathrm{DtN}_2(u_2^{n-1}) & \text{on } \Gamma \\
(\eta - \Delta) u_2^n & = & f & \text{in } \Omega_2 \\
\partial_x u_2^n - \mathrm{DtN}_1(u_2^n) & = & \partial_x u_1^{n-1} - \mathrm{DtN}_1(u_1^{n-1}) & \text{on } \Gamma
\end{array}
$$

Result: This algorithm converges in two iterations,

An Optimal Schwarz Algorithm

Based on the decomposition of $\Omega = \Omega_1 \cup \Gamma \cup \Omega_2$



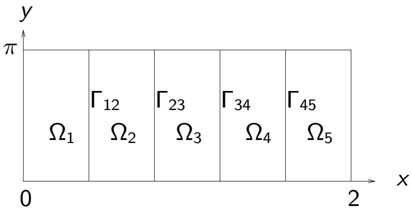
The optimal Schwarz algorithm is

$$\begin{array}{rcll}
(\eta - \Delta)u_1^n & = & f & \text{in } \Omega_1 \\
\partial_x u_1^n - \mathrm{DtN}_2(u_1^n) & = & \partial_x u_2^{n-1} - \mathrm{DtN}_2(u_2^{n-1}) & \text{on } \Gamma \\
(\eta - \Delta)u_2^n & = & f & \text{in } \Omega_2 \\
\partial_x u_2^n - \mathrm{DtN}_1(u_2^n) & = & \partial_x u_1^{n-1} - \mathrm{DtN}_1(u_1^{n-1}) & \text{on } \Gamma
\end{array}$$

Result: This algorithm converges in two iterations, and with N subdomains in N iterations,

An Optimal Schwarz Algorithm

Based on the decomposition of $\Omega = \Omega_1 \cup \Gamma \cup \Omega_2$



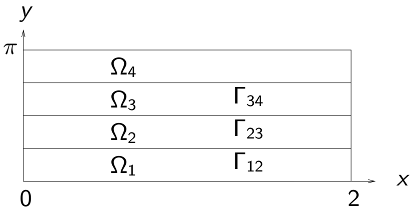
The optimal Schwarz algorithm is

$$\begin{array}{rcll} (\eta-\Delta)u_1^n & = & f & \text{in } \Omega_1 \\ \partial_x u_1^n - \mathrm{DtN}_2(u_1^n) & = & \partial_x u_2^{n-1} - \mathrm{DtN}_2(u_2^{n-1}) & \text{on } \Gamma \\ (\eta-\Delta)u_2^n & = & f & \text{in } \Omega_2 \\ \partial_x u_2^n - \mathrm{DtN}_1(u_2^n) & = & \partial_x u_1^{n-1} - \mathrm{DtN}_1(u_1^{n-1}) & \text{on } \Gamma \end{array}$$

Result: This algorithm converges in two iterations, and with N subdomains in N iterations, or in two when sweeping back and forth once, independently of N.

An Optimal Schwarz Algorithm

Based on the decomposition of $\Omega = \Omega_1 \cup \Gamma \cup \Omega_2$



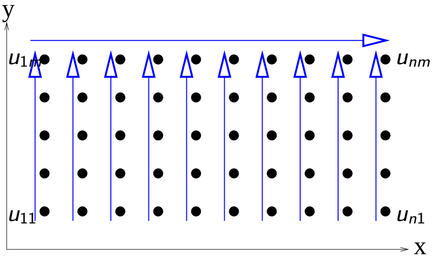
The optimal Schwarz algorithm is

$$
\begin{array}{rcll}
(\eta - \Delta) u_1^n & = & f & \text{in } \Omega_1 \\
\partial_x u_1^n - \mathrm{DtN}_2(u_1^n) & = & \partial_x u_2^{n-1} - \mathrm{DtN}_2(u_2^{n-1}) & \text{on } \Gamma \\
(\eta - \Delta) u_2^n & = & f & \text{in } \Omega_2 \\
\partial_x u_2^n - \mathrm{DtN}_1(u_2^n) & = & \partial_x u_1^{n-1} - \mathrm{DtN}_1(u_1^{n-1}) & \text{on } \Gamma
\end{array}
$$

Result: This algorithm converges in two iterations, and with N subdomains in N iterations, or in two when sweeping back and forth once, independently of N.

Discrete Transparent Boundary Conditions

$$(\eta - \Delta)u \quad = \quad f \qquad \text{in } \Omega = (0,\infty) \times (0,\pi)$$



$$A\mathbf{u} = \begin{bmatrix} D_1 & L_{1,2} & & & \\ L_{2,1} & D_2 & \ddots & & \\ & \ddots & \ddots & L_{p-1,p} & \\ & & L_{p,p-1} & D_{p,p} & \ddots \\ & & & \ddots & \ddots \end{bmatrix} \begin{pmatrix} \mathbf{u}_1 \\ \vdots \\ \mathbf{u}_{p-1} \\ \mathbf{u}_p \\ \vdots \end{pmatrix} = \begin{pmatrix} \mathbf{f}_1 \\ \vdots \\ \mathbf{f}_{p-1} \\ \mathbf{0} \\ \vdots \end{pmatrix}$$

Elimination of the Outer Variables

$$\begin{bmatrix} D_1 & L_{1,2} & & \\ L_{2,1} & \ddots & \ddots & \\ & \ddots & D_{n-1} & L_{n-1,n} \\ & & L_{n,n-1} & D_n \end{bmatrix} \begin{pmatrix} \vdots \\ \vdots \\ \mathbf{u}_{n-1} \\ \mathbf{u}_n \end{pmatrix} = \begin{pmatrix} \vdots \\ \vdots \\ \mathbf{0} \\ \mathbf{0} \end{pmatrix}$$

Eliminate variables from the end using Schur complements:

$$\begin{bmatrix} D_1 & L_{1,2} & & \\ L_{2,1} & \ddots & \ddots & \\ & \ddots & D_{n-2} & L_{n-2,n-1} \\ & & L_{n-1,n-2} & D_{n-1} - L_{n-1,n} D_n^{-1} L_{n,n-1} \end{bmatrix}$$

Elimination of the Outer Variables

$$\begin{bmatrix} D_1 & L_{1,2} & & & \\ L_{2,1} & \ddots & \ddots & & \\ & \ddots & D_{n-1} & L_{n-1,n} \\ & & L_{n,n-1} & D_n \end{bmatrix} \begin{pmatrix} \vdots \\ \vdots \\ \mathbf{u}_{n-1} \\ \mathbf{u}_n \end{pmatrix} = \begin{pmatrix} \vdots \\ \vdots \\ \mathbf{0} \\ \mathbf{0} \end{pmatrix}$$

Eliminate variables from the end using Schur complements:

$$\begin{bmatrix} D_1 & L_{1,2} & & & \\ L_{2,1} & \ddots & \ddots & & \\ & \ddots & D_{n-2} & L_{n-2,n-1} \\ & & L_{n-1,n-2} & T_{n-1} \end{bmatrix}$$

Elimination of the Outer Variables

$$\begin{bmatrix} D_1 & L_{1,2} & & & \\ L_{2,1} & \ddots & \ddots & & \\ & \ddots & D_{n-1} & L_{n-1,n} \\ & & L_{n,n-1} & D_n \end{bmatrix} \begin{pmatrix} \vdots \\ \vdots \\ \mathbf{u}_{n-1} \\ \mathbf{u}_n \end{pmatrix} = \begin{pmatrix} \vdots \\ \vdots \\ \mathbf{0} \\ \mathbf{0} \end{pmatrix}$$

Eliminate variables from the end using Schur complements:

$$\begin{bmatrix} D_1 & L_{1,2} & & \\ L_{2,1} & \ddots & \ddots & \\ & \ddots & D_{n-2} & L_{n-2,n-1} \\ & & L_{n-1,n-2} & T_{n-1} \end{bmatrix}$$

The same recurrence relation as in block ILU

$$T_i = \begin{cases} D_n & i = n, \\ D_i - L_{i,i+1} T_{i+1}^{-1} L_{i+1,i} & i = n-1, n-2, \ldots \end{cases}$$

Discrete Transparent Boundary Condition

To truncate the domain at block p:

$$\tilde{A} = \begin{bmatrix} D_1 & L_{1,2} & & & \\ L_{2,1} & D_2 & \ddots & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & D_{p-1,p-1} & L_{p-1,p} \\ & & & L_{p,p-1} & T_p \end{bmatrix}$$

where

$$T_i = \begin{cases} D_n & i = n, \\ D_i - L_{i,i+1} T_{i+1}^{-1} L_{i+1,i} & i = n-1, n-2, \dots, p \end{cases}$$

What happens if the outer domain is infinite ?

Discrete TBC for Infinite Domain

Assume the outer problem has constant coefficients

$$A = \begin{bmatrix} D_1 & L_{1,2} & & & & \\ L_{2,1} & D_2 & \ddots & & & \\ & \ddots & \ddots & L_{p-1,p} & & \\ & & L_{p,p-1} & D_{p,p} & L & \\ & & & L & D & \ddots \\ & & & & \ddots & \ddots \end{bmatrix}$$

Then the recurrence relation for T_i is

$$T_i = \begin{cases} D & i = n \text{ very large}, \\ D - LT_{i+1}^{-1}L & i = n-1, n-2, \dots, p \end{cases}$$

and in the limit, when n goes to infinity $T_p = T_\infty$ where T_∞ satisfies

$$T_\infty = D - LT_\infty^{-1}L$$

Equivalence Between Discrete and Continuous

Result: The equivalent of the continuous problem

$$\begin{aligned} (\eta - \Delta)u &= f && \text{in } \Omega_{int} \\ \partial_x u - \mathrm{DtN}(u) &= 0 && \text{on } \Gamma \end{aligned}$$

is at the algebraic level

$$\tilde{A}\mathbf{u} = \begin{bmatrix} D_1 & L_{1,2} & & \\ L_{2,1} & D_2 & \ddots & \\ & \ddots & \ddots & L_{p-1,p} \\ & & L_{p,p-1} & T_\infty \end{bmatrix} \begin{pmatrix} \mathbf{u}_1 \\ \vdots \\ \mathbf{u}_{p-1} \\ \mathbf{u}_p \end{pmatrix} = \begin{pmatrix} \mathbf{f}_1 \\ \vdots \\ \mathbf{f}_{p-1} \\ \mathbf{f}_p \end{pmatrix}$$

where

$$T_\infty = D - L T_\infty^{-1} L$$

Approximations at the Continuous Level

For the model problem

$$\begin{aligned}(\eta - \Delta)u &= f && \text{in } \Omega_{int}\\ \partial_x u - \mathrm{DtN}(u) &= 0 && \text{on } \Gamma\end{aligned}$$

one can show that the Fourier transform along the interface of the DtN operator is

$$\widehat{\mathrm{DtN}}(u) = -\sqrt{\eta + k^2}$$

Approximations are

- the so called absorbing boundary conditions (ABCs)

$$\sqrt{\eta + k^2} \approx p + qk^2.$$

- and the perfectly matched layers (PMLs)

$$\sqrt{\eta + k^2} \approx \frac{P(k)}{Q(k)}.$$

Approximations at the Discrete Level

Have to approximate the dense matrices T_i in

$$T_i = \begin{cases} D & i = n \text{ very large}, \\ D - LT_{i+1}^{-1}L & i = n-1, n-2, \ldots, p \end{cases}$$

by sparse matrices.

Can use the same techniques as at the continuous level!

- approximation by a tridiagonal matrix (ABCs),

$$\Longleftrightarrow \sqrt{\eta + k^2} \approx p + qk^2.$$

- approximation using the PML
- approximation using H-matrix techniques

Algorithms Based on Block Factorization

Analytic Incomplete LU, AILU (G, Nataf 1998,2002,2005) is an approximate LDL^T factorization based on approximating

$$T_i \approx T_\infty \approx \sqrt{\eta + k^2} \approx p + qk^2 \approx \text{Tridiag}$$

	QMR		ILU('0')		ILU(1e-2)		AILU('0')	
k	it	Mflops	it	Mflops	it	Mflops	it	Mflops
10	737	1858.2	370	1489.3	80	421.4	36	176.2
15	1775	10185.2	2000	18133.2	220	2615.1	43	475.9
20	2000	20335.1	—	—	2000	42320.1	64	1260.2
30	–	–	–	–	–	–	90	3984.1
50	–	–	–	–	–	–	285	24000.4

Theorem (G, Nataf 2000:)

Frequency Filtering (Wittum 1991) is an AILU with discrete tridiagonal approximation of T_i, exact for 2 frequencies

Theorem (G, Zhang 2012:)

The Sweeping Preconditioner (Enquist, Ying (2010,...) is an AILU approximating T_i by PML or H-matrices

Algorithms Based on Domain Decomposition

Optimal Schwarz methods use the *DtN* as transmission conditions (Nataf, Rogier, de Sturler 1994, G, Kwok 2010)

Optimized Schwarz methods use approximations of the *DtN* in the transmission condition (G, Halpern, Nataf 2000)

Theorem (G, Zhang 2013:)

The source transfer domain decomposition method (Chen, Xiang 2012) is an optimized Schwarz method using a PML approximation of the DtN in the transmission condition on one side of the subdomains, and Dirichlet on the other.

Theorem (G, Zhang 2013:)

The PML domain decomposition method (Stolk 2013) is an optimized Schwarz method based on one forward and one backward sweep using a PML approximation of the DtN in the transmission condition.

Conclusions

- The Dirichlet to Neumann operator (DtN) and its discrete equivalent T_i appear naturally both in strip domain decomposition methods and block factorizations

- Any approximation of the DtN or its discrete equivalent T_i can be used to obtain AILU preconditioners or optimized Schwarz methods

Conclusions

- The Dirichlet to Neumann operator (DtN) and its discrete equivalent T_i appear naturally both in strip domain decomposition methods and block factorizations
- Any approximation of the DtN or its discrete equivalent T_i can be used to obtain AILU preconditioners or optimized Schwarz methods
- Optimized Schwarz and AILU are very much related, just use strip domains instead of single layers !

Conclusions

- The Dirichlet to Neumann operator (DtN) and its discrete equivalent T_i appear naturally both in strip domain decomposition methods and block factorizations
- Any approximation of the DtN or its discrete equivalent T_i can be used to obtain AILU preconditioners or optimized Schwarz methods
- Optimized Schwarz and AILU are very much related, just use strip domains instead of single layers !

Preprints are available at www.unige.ch/∼gander