12th International Conference on Domain Decomposition Methods
Editors: Tony Chan, Takashi Kako, Hideo Kawarada, Olivier Pironneau,

10. A Two-Stage Multi-Splitting Method for Non-Overlapping Domain Decomposition for Parabolic Equations

Daoud S. Daoud[1], Bruce A. Wade[2]

Domain Decomposition

In domain decomposition for parabolic partial differential equations (PDE) several approaches have been developed— breaking the domain into multiple subdomains of either overlapping or non-overlapping type, or using algebraic type splittings— *cf.* [CM94] for an overview. An important aspect is how to present the boundary conditions across interfaces or across common unknown points of subdomains, *cf.* [GS98, HT96, Tan92]. Towards parallelism, we divide the domain into subdomains with one grid point in common, adding an extra unknown at the interface to have effectively a non-overlapping decomposition.

In the present numerical method we have designed a one gridpoint overlap together with an extra equation in order to arrive at an effective multi-splitting approach. The transmission of data at the interface is through a discrete parametrized Robin boundary condition across interior interface points. A significant part of this report is the design and experimental study of optimizing boundary parameter coupled with particular choices of inner and outer splittings. We are interested here in extending some work of San and Tang [HT96] and Tang [Tan92] to parabolic problems. There is a parameter γ that acts like a feedback gain across the artificial interfaces. The primary aspect of this article is to construct and demonstrate effective multi-splitting methods as depending on the interface boundary condition.

Consider the numerical solution of parabolic problems of form:

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} + f(x,t), \quad x \in (0,L), t > 0, \tag{1}$$

subject to initial condition $u(x,0) = g(x)$, $x \in \Omega := (0,L)$ and boundary conditions $u(0,t) = u(1,t) = 0$.

With the usual $(\Delta x, \Delta t)$ mesh in (x,t) ($\Delta x = 1/M, x_i = i\Delta x, i{=}0,\cdots,M, t_j = j\Delta t, j = 0,\cdots,N$), let $\bar{x}$ be a point of interface by which the domain Ω is decomposed into two subdomains $\Omega_1 = \{x \in \Omega : x < \bar{x} + \Delta x\}$ and $\Omega_2 = \{x \in \Omega : x > \bar{x} - \Delta x\}$, where $\bar{x} = m\Delta x$, for some m with $1 < m < M-1$. We are utilizing just two subdomains to address the essential issues. Solutions u_1 and u_2 are restricted versions of u over the domains.

[1]Department of Mathematics, Eastern Mediterranean University, Famagusta, North Cyprus, Via Mersin 10, Turkey. daoud.daoud@emu.edu.tr.
[2]Department of Mathematical Sciences, University of Wisconsin– Milwaukee, Milwaukee, Wisconsin 53201-0413. wade@uwm.edu.

We introduce a boundary condition at $\bar{x}$ as follows

$$\alpha u_1 + (1-\alpha)\frac{\partial u_1}{\partial n}\bigg|_{\bar{x}-} = \alpha u_2 + (1-\alpha)\frac{\partial u_2}{\partial n}\bigg|_{\bar{x}+} \qquad \alpha \in [0,1]. \tag{2}$$

Then the global problem (1) is split over Ω_1 and Ω_2 in a natural fashion.

One can employ the method of lines with (1) through a second order central difference approximation [Smi85]: $\frac{\partial^2 u}{\partial x^2}\Big|_{(x_i,t_j)} = \Delta x^{-2}(u_{i-1,j} - 2u_{i,j} + u_{i+1,j}) + O(\Delta x^2)$. The normal derivative in the boundary condition (2) at $(\bar{x},t) = (x_m,t)$ is approximated by forward or backward differences for Ω_1, and Ω_2, respectively, as follows:

$$\begin{aligned} \alpha u_1|_{m,j} &+ (1-\alpha)\frac{1}{\Delta x}\left(u_1|_{m+1,j} - u_1|_{m,j}\right) = \\ \alpha u_2|_{m,j} &+ (1-\alpha)\frac{1}{\Delta x}\left(u_2|_{m+1,j} - u_2|_{m,j}\right), \qquad \bar{x} = x_m \in \Omega_1, \end{aligned} \tag{3}$$

$$\begin{aligned} \alpha u_2|_{m,j} &+ (1-\alpha)\frac{1}{\Delta x}\left(u_2|_{m,j} - u_2|_{m-1,j}\right) = \\ \alpha u_1|_{m,j} &+ (1-\alpha)\frac{1}{\Delta x}\left(u_1|_{m,j} - u_1|_{m-1,j}\right), \qquad \bar{x} = x_m \in \Omega_1. \end{aligned} \tag{4}$$

In equation (3) u_2 is considered as given (known), $u_1|_{m,j}$ is an unknown, and $u_1|_{m+1,j}$ is a fictitious value for which this equation provides substitution. The situation is similar for equation (4). Re-write these equations by introducing a convenient parameter $\gamma := (1-\alpha(1+\Delta x))/(1-\alpha)$ as follows:

$$u_1|_{m+1,j} - u_2|_{m+1,j} = \gamma\left(u_1|_{m,j} - u_2|_{m,j}\right), \tag{5}$$

$$u_2|_{m-1,j} - u_1|_{m-1,j} = \gamma\left(u_2|_{m,j} - u_1|_{m,j}\right). \tag{6}$$

In equations (5) and (6) the parametrized discrete Robin boundary condition at the matrix level (below) amounts to a kind of error feedback where γ is the gain. If γ is first chosen, then α in (3) and (4) becomes $\alpha = (\gamma-1)/(\gamma-(1+\Delta x))$. We note the particular choice of $\gamma = 0$ ($\alpha = 1/(1+\Delta x)$), giving a variation on the SAM Dirichlet condition in which an extra column has been inserted to slide the entries over; we call this the Sliding Dirichlet Condition.

By substituting (3) and (4) in (2), a couple of first order systems of differential for u_1 and u_2 arise:

$$\frac{du_1}{dt} = B_1 u_1 + \begin{bmatrix} 0 \\ \vdots \\ 0 \\ \Delta x^{-2}(u_{2,m+1} - \gamma u_{2,m}) \end{bmatrix}, \tag{7}$$

$$\frac{du_2}{dt} = B_2 u_2 + \begin{bmatrix} \Delta x^{-2}(u_{1,m-1} - \gamma u_{1,m}) \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \tag{8}$$

where $B_1 \in \mathbf{R}^{m\times m}$, and $B_2 \in \mathbf{R}^{M-m\times M-m}$.

The i-th row of B_1 has form $\Delta x^{-2}[0,\cdots,0,1,-2,1,0,\cdots,0]$, $i=1,\cdots,m-1$, while the m-th row of B_1 is $\Delta x^{-2}[0,\cdots,1,-2+\gamma]$. For B_2 we retain the numbering from the spatial grid, so the i-th row is $\Delta x^{-2}[0,\cdots,0,1,-2,1,0,\cdots],i=m+1,\cdots,M-1$, and the m-th row (the first row in actuality) is $\Delta x^{-2}[-2+\gamma,1,0,...,0]$. Then we assemble the matrices B_1 and B_2 into a block matrix of coefficients A to represent the global system of ordinary differential equations. There are now M unknowns due to the extra unknown $u_{2,m}$ (formerly, there were $M-1$ unknowns). Setting $u=[u_1,u_2]^T$, we arrive at:

$$\frac{du}{dt} = Bu. \tag{9}$$

Here, $B \in \mathbf{R}^{M\times M}$ is given by:

$$B = \Delta x^{-2}\begin{bmatrix} -2 & 1 & & & & & & & \\ & \ddots & \ddots & \ddots & & & & & \\ & 1 & -2 & 1 & & & & & \\ & & 1 & -2+\gamma & -\gamma & 1 & & & \\ & & 1 & -\gamma & -2+\gamma & 1 & & & \\ & & & & 1 & -2 & 1 & & \\ & & & & & \ddots & \ddots & \ddots & \\ & & & & & & 1 & -2 & \end{bmatrix}. \tag{10}$$

The exact solution of the semi-discrete system (9) satisfies the following two-term recurrence relation, [Smi85, Var62]:

$$u(t+\Delta t) = e^{\Delta t B}u(t). \tag{11}$$

Several algorithms for the numerical solution of (11) can be generated through an approximation to the exponential $e^{\Delta t B}$; in particular, we shall use rational functions via implicit Padé approximations, focussing on (1,0) and (1,1) Padé schemes, *cf.* [Smi85].

The (1,0) and (1,1) Padé approximations are given (respectively) by

$$e^{\Delta t B} = (I-\Delta t B)^{-1} + O(\Delta t)\ \&\ = (I-0.5\Delta t B)^{-1}(I+0.5\Delta t B) + O(\Delta t^2). \tag{12}$$

For each scheme of (12) the recurrence relation for $u(t+\Delta t)$ in (11) gives the following linear system to solve:

$$(I-\sigma B)u(t+\Delta t) = R_m(\Delta t B)u(t), \tag{13}$$

where (for (1,0) and (1,1), respectively)

$$R_1(\Delta t B) = I,\ \sigma = \Delta t,\ \&\ R_2(\Delta t B) = I+\sigma B,\quad \sigma = 0.5\Delta t.$$

Let $A = I-\sigma B$ and $\tau = \sigma\Delta x^{-2}$. Then A is illustrated as follows:

$$A = I-\sigma B = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}, \tag{14}$$

where

$$A_{11} = \begin{bmatrix} 1+2\tau & -\tau & 0 & \cdots & 0 \\ -\tau & 1+2\tau & -\tau & & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & & -\tau & 1+2\tau & -\tau \\ 0 & \cdots & 0 & -\tau & 1+(2-\gamma)\tau \end{bmatrix},$$

$$A_{12} = \begin{bmatrix} 0 & 0 & \cdots & 0 & 0 \\ \vdots & & & & \vdots \\ 0 & 0 & \cdots & 0 & 0 \\ \gamma\tau & \tau & 0 & 0 & 0 \end{bmatrix},$$

$$A_{21} = \begin{bmatrix} 0 & 0 & \cdots & \tau & \gamma\tau \\ 0 & 0 & \cdots & 0 & 0 \\ \vdots & & & & \vdots \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$

$$A_{22} = \begin{bmatrix} 1+(2-\gamma)\tau & -\tau & 0 & \cdots & 0 \\ -\tau & 1+2\tau & -\tau & & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & & -\tau & 1+2\tau & -\tau \\ 0 & \cdots & 0 & -\tau & 1+2\tau \end{bmatrix}.$$

Due to the implicit nature of the finite difference scheme, we desire to avoid undue restriction on the size of Δt. Thus, we assume only that $\Delta t/\Delta x \leq 1$, so that τ is at most $O(\Delta x^{-1})$ and is expected to be of order Δx^{-1}. We desire to choose α (or γ) to assure that A is an M-matrix or an H-matrix. It is easy to check that one must have $\gamma \leq 1$ for $\alpha \in [0,1]$; if $0 < \gamma \leq 1$ and $2\gamma\tau \leq 1$, then A is an H-matrix; and if $\gamma \leq 0$ then A is an M-matrix. Since τ depends on the mesh size, we want to reject the condition $2\gamma\tau \leq 1$. Therefore, we are now concentrating on the situation where $\gamma \leq 0$, in which case A is an M-matrix. This is easy to check upon consulting [Var62, p. 85].

The Two-Stage Multi-splitting Algorithm

The main advantage in the proposed boundary condition (3, 4) at interfaces is that it leads to a partitioning of the global matrix A in (14) into particularly beneficial submatrix blocks, which are amenable to solution of multi-splitting iterative type. The parallel multi-splitting method of O'leary and White [OW85] to solve a linear system $Au = b$ is defined by considering multi-splittings from the decomposition $A = M_k - N_k, k = 1, \cdots, K$, such that each M_k is invertible; we can form an iterative method, as follows:

$$u^{i+1} = M_k^{-1} N_k u^i + M_k^{-1} b. \tag{15}$$

The multi-splitting iterates in (15) can be computed concurrently by introducing the non-negative diagonal matrices E_k, $\sum_{k=1}^{K} E_k = I$, and using them as follows:

$$u_{i+1} = Hu_i + c, \tag{16}$$

where $H = \sum E_k M_k^{-1} N_k$ and $c = \sum E_k M_k^{-1} b$.

For the linear system of concern here the first stage of splitting is according to the subdomains Ω_k (*cp.* [FS97]):

$$M_i = \begin{bmatrix} D_{11} & & & & \\ & \ddots & & & \\ & & Aii & & \\ & & & \ddots & \\ & & & & D_{KK} \end{bmatrix}, \tag{17}$$

where the matrices D_{kk} are the diagonals of A_{kk}, respectively, or any dummy diagonal matrices to make $M_k, k = 1, \cdots, K$ invertible. We take $N_k = M_k - A$.

To calculate u^{n+1} from u^n (where $u = [u_1, u_2, \cdots, u_K]^T$), use:

$$\begin{aligned} M_k w_k^{i+1} &= N_k u^i + b, \quad k = 1, \cdots, K, \\ u^{n+1} &= \sum_{k=1}^{K} E_k w_k^{i+1}. \end{aligned} \tag{18}$$

This is rewritten as a single splitting with $M = \text{diag}\,(A_{ii})$.

With the splitting above one could implement a nice parallel algorithm. There are a wide variety of iterative schemes to consider. As in [FS94], it is useful to examine the possibility of introducing a second state of multi-splitting methods, this time splitting the matrices M_k according to their convenient algebraic struture [OW85].

The second stage of splitting will be considered for the matrices M_k, $k = 1, \cdots, K$. For this article, we deal with three rather standard approaches for inner iteration: the Jacobi, Gauss-Seidel, and SOR. Each subdomain block has been constructed to be of form only slightly different from a standard type matrix, and it is natural to test these schemes to gain insight into the proposed boundary treatment.

For brevity, we shall only describe the procedure, assuming the standard version of each algorithm is well known. The outer iteration (also called the first stage) consists of splitting the matrix A as diagrammed earlier. Then we split the submatrices A_{ij} to the right hand side whenever $i \neq j$ and on processor i we split all other A_{jj} submatrices to the right, leaving only their diagonals.

The Parallel Jacobi (or PJacobi) version of the inner iteration (or second stage) is to iterate s times on processor i for unknowns corresponding to the i^{th} subdomain, splitting the upper and lower diagonal parts to the right hand side as well.

The Parallel Gauss-Seidel (or PGauss-Seidel) inner iteration is to iterate s times on processor i for unknowns corresponding to the i^{th} subdomain, splitting the upper diagonal parts within that block to the right hand side.

Parallel Successive Over-Relaxation (or PSOR) does s iterations for i^{th} subdomain unknowns of standard SOR (using ω as the parameter).

Numerical Experiments

In this section we demonstrate the performance of the algorithms described earlier and give empirical support to claim of convergence.

Let's work with the following prototype parabolic (heat) equation:

$$u_t = u_{xx} + x(1-x)\pi \sin(\pi t) - 2\cos(\pi t), \quad x \in (0,1), t > 0$$

$$u(x,0) = x(x-1), \quad x \in (0,1),$$

with boundary conditions $u(0,t) = u(1,t) = 0$. The exact solution is $x(x-1)\cos(\pi t)$. We will compute approximations to the solution at target time $t_* = 1.0$. All iterations have a halting signal of $||u^{n+1} - u^n|| \leq 0.5\Delta t^2 \Delta x ||u^n||$, which means smaller step sizes require more iterations.

The first numerical experiment to report, see Figure 1, is with inner splittings of Jacobi type, different values of K (the number of subdomains), two particular choices of γ, and $s = 1$. The computations were done in FORTRAN 90 on a Penitum III (not in parallel). The case $\gamma = 0.0$ is called the Sliding Dirichlet Condition because it looks like a Dirichlet condition, shifted over by one column. It arises from $\alpha = 1/(1 + \Delta x)$ in the discrete parametrized Robin condition, so it is not the ordinary Schwarz Alternating Method Dirichlet condition. It is observed in Figure 1 that the Sliding Dirichlet Condition allows the ordinary Jacobi algorithm to be parallelized in an efficient manner, in that the spectral radius appears to be nearly identical over any number of domain splittings (the numbers shown are rounded). Thus, our Parallel Jacobi method (PJacobi) would run a factor of K times faster on a parallel machine with K processors. (Here, communication is neglected because it is relatively insignificant.)

Moreover, there are better choices of γ. As shown in Figure 1, $\gamma = -1.0$ speeds up the PJacobi multi-splitting scheme quite significantly. (The value $\gamma = -1.0$ is not the absolute optimal from experiment, but is nearly so, and this is chosen for convenience.) This corresponds to a discrete parametrized Robin boundary condition with $\alpha = 2/(2 + \Delta x)$.

Using old values from neighboring domains and a single inner iteration ($s = 1$), we want to have a look at the Gauss-Seidel scheme on each subdomain. The Parallel Gauss-Seidel scheme performs best when γ is zero (or slightly positive); see Figure 2.

References

[CM94]Tony F. Chan and Tarek P. Mathew. Domain decomposition algorithms. In *Acta Numerica 1994*, pages 61–143. Cambridge University Press, 1994.

[FS94]A. Frommer and D.B. Szyld. Asynchronous two stage iterative methods. *Numer. Math.*, 69:141–153, 1994.

[FS97]A. Frommer and H. Schwandt. A unified representation and theory of algebraic additive schwarz and multisplitting methods. *SIAM J. Matrix Anal. Appl.*, 18:893–912, 1997.

[GS98]M.J. Gander and A.M. Stuart. Space time continuous analysis of waveform relaxation for the heat equation. *SIAM J.*, 19:2014–2031, 1998.

[HT96]H.San and W.P. Tang. An overdetermined schwarz alternating method. *SIAM J. Sci. Comput.*, 7:884–905, 1996.

K	γ	Avg. No. Iter. Per Step $\Delta x = 0.003125$	Avg. No. Iter. Per Step $\Delta x = 0.0015625$
1	N/A	4407	9856
	N/A		
2	0.0	4407	9856
	-1.0	4058	9436
4	0.0	4407	9856
	-1.0	3097	8070
8	0.0	4407	9856
	-1.0	1632	5164
16	0.0	4407	9856
	-1.0	1468	2935
32	0.0	4407	9856
	-1.0	1486	2953

Figure 1: Parallel Jacobi showing the effect of γ and K ($s = 1$).

K	Avg. No. Iter. Per Step $\Delta x = 0.003125$	Avg. No. Iter. Per Step $\Delta x = 0.0015625$
1	941	1895
2	948	1905
4	948	1905
8	950	1906
16	957	1912
32	971	1927

Figure 2: Parallel Gauss-Seidel showing the effect of K with $\gamma = 0.0$ and $s = 1$.

[OW85]D. P. O'Leary and R.E. White. Multi-splittings of matrices and parallel solution of linear systems. *SIAM J. Alg. Disc. Meth.*, 6:630–640, 1985.

[Smi85]G.D. Smith. *Numerical Solution of Partial Differential Equations: Finite Difference Methods*. Oxford Univ. Press, second edition, 1985.

[Tan92]W.P. Tang. Generalized schwarz splittings. *SIAM J. Sci. Stat. Comput.*, 13:573–595, 1992.

[Var62]R. Varga. *Matrix Iterative Analysis*. Prentice Hall, first edition, 1962.

K	Avg. No. Iter. Per Step (ω) $\Delta x = 0.003125$	Avg. No. Iter. Per Step (ω) $\Delta x = 0.0015625$
1	108 (1.90)	167 (1.92)
2	118 (1.90)	214 (1.90)
4	142 (1.87)	217 (1.90)
8	170 (1.84)	243 (1.89)
16	231 (1.78)	321 (1.85)
32	365 (1.64)	510 (1.75)

Figure 3: Parallel SOR (at near optimal ω) showing the effect of K with $\gamma = 0.0$ and $s = 1$.