Acceleration of a Schwarz Waveform Relaxation Method for Parabolic Problems.

M. Garbey[†]
† CS dept. University of Houston
garbey@cs.uh.edu

Department of Computer Science
University of Houston
Houston, TX, 77204, USA
http://www.cs.uh.edu

Technical Report Number UH-CS-06-11

September 9, 2006

Keywords: Domain Decomposition, Aitken like acceleration, Elliptic Operators, additive Schwarz waveform relaxation, parabolic problems, parallel algorithm

Abstract

In this paper we generalize the Aitken-like acceleration method of the additive Schwarz algorithm for elliptic problems to the additive Schwarz waveform relaxation for parabolic problems. The domain decomposition is in space and time. The standard Schwarz waveform relaxation algorithm has a linear rate of convergence and low numerical efficiency. This algorithm is, however, friendly to cache use and scales with the memory in parallel environments. It also minimizes the number of messages sent in a parallel implementation and is therefore very insensitive to delays due to a high latency network. M. Gander and co-workers have shown that the convergence of this algorithm can be speed up by optimizing the transmission conditions. Our Aitken-like acceleration is an alternative method that consists of postprocessing the sequence of interfaces generated by the domain decomposition solver and might be combined to the method of Gander et al. We show that our technique (1) is a direct solver that requires at most four solves per sub-domain in the case of a one space dimension linear parabolic problem with time independent coefficients, (2) can be applied easily to multi-dimensional problems, provided that the operator is separable in space (3) is an efficient iterative procedure for parabolic problems that are weak nonlinear perturbations of linear operators with time independent coefficients, (4) provides a rigorous framework to optimize the parallel implementation on a slow network of computers.

Acceleration of a Schwarz Waveform Relaxation Method for Parabolic Problems.

M. Garbey[†]
† CS dept. University of Houston
garbey@cs.uh.edu

Abstract

In this paper we generalize the Aitken-like acceleration method of the additive Schwarz algorithm for elliptic problems to the additive Schwarz waveform relaxation for parabolic problems. The domain decomposition is in space and time. The standard Schwarz waveform relaxation algorithm has a linear rate of convergence and low numerical efficiency. This algorithm is, however, friendly to cache use and scales with the memory in parallel environments. It also minimizes the number of messages sent in a parallel implementation and is therefore very insensitive to delays due to a high latency network. M. Gander and co-workers have shown that the convergence of this algorithm can be speed up by optimizing the transmission conditions. Our Aitken-like acceleration is an alternative method that consists of postprocessing the sequence of interfaces generated by the domain decomposition solver and might be combined to the method of Gander et al. We show that our technique (1) is a direct solver that requires at most four solves per sub-domain in the case of a one space dimension linear parabolic problem with time independent coefficients, (2) can be applied easily to multi-dimensional problems, provided that the operator is separable in space (3) is an efficient iterative procedure for parabolic problems that are weak nonlinear perturbations of linear operators with time independent coefficients, (4) provides a rigorous framework to optimize the parallel implementation on a slow network of computers.

Index Terms

Domain Decomposition, Aitken like acceleration, Elliptic Operators, additive Schwarz waveform relaxation, parabolic problems, parallel algorithm

I. Introduction

In this paper we generalize the Aitken-like acceleration method of the additive Schwarz algorithm for elliptic problems to the additive Schwarz waveform relaxation for parabolic problems.

Currently, standard processors are becoming multi-cores and there is a strong incentive to make use of all these parallel resources while avoiding conflict in memory access. We also have an overwhelming abundance of parallel computers available when using grids. The Additive Schwarz (AS) method for elliptic problems or the Additive Schwarz Waveform Relaxation (ASWR) method for parabolic problems can be implemented easily in distributed computing environments and have very simple and systematic communication schemes. At the end of each sub-domain solve, the processor units exchange their interfaces with the ones that process neighboring subdomains. The sub-domain solver can be optimized independently of the overall implementation. These algorithms are friendly to memory cache use and scale with the memory in parallel environments. ASWR in particular minimizes the number of messages sent in a parallel implementation and is very insensitive to delays due to a high latency network. The main drawback of these methods is that they are one or several orders of magnitude slower than modern solvers such as multigrids. In the meantime, multigrids have poor parallel efficiency with high latency networks.

There have been two main classes of methods to speed up AS and ASWR. One is to introduce a coarse grid preconditioner. But a coarse grid operator reduces drastically the parallel efficiency on a slow network. A second option is to optimize the transmission conditions. This general avenue of work has been followed with success by numerous workers - see for example [22], [23], [7], [8], [11], [20], [24], [25], [4] and their references. We have introduced in [16] a different and somehow complementary approach that consists of accelerating the sequence of trace on the interface generated by the AS method. The advantage of our postprocessing algorithm, besides its simplicity, is that it has quasi-optimum arithmetic complexity for the Poisson equation discretized on Cartesian grid while offering unique parallel efficiency on the grid. This is the only example, to our knowledge, of a numerically efficient Poisson solver that performs well on a grid of computers [3].

Our method offers also a general framework to speed up elliptic and non-linear elliptic solvers in a broad variety of conditions [2], [3], [12], [13], [15], [16]. The salient feature of the method consists of approximating the main eigenvectors of the trace transfer operator in the AS algorithm and applying a standard acceleration technique based on the linear rate of convergence, component by component, of the trace generated by AS in the proper basis. This idea might be applicable to any sub-domain-wise relaxation technique that has a linear rate of convergence.

Our main objective in this paper is to present an extension of this technique to parabolic operators with domain decomposition in space *and* time. We show how one can construct the trace transfer operator in ASWR or a fast and accurate approximation of it, and eventually derive an efficient acceleration technique. Numerical experiments will demonstrate the efficiency of our new algorithm for the standard heat equation with a box scheme on a Cartesian grid. There are indeed dozens of *iterative* methods to solve parabolic problems with domain decomposition. We refer to the proceeding series listed in http://www.ddm.org to provide a representative set of methods. We will show that our algorithm for the heat equation with a finite difference space stencil on a Cartesian grid is a parallel *direct* solver with a quasi-optimal arithmetic complexity with a very low requirement on the latency of the network. This result does not hold in general for any parabolic operator. The separability of the operator in space is a critical assumption to get optimum performance. Nevertheless, we will illustrate the potential and drawback of our method with weakly nonlinear problems. Performance of our method on grid computing will be reported in a companion paper.

The plan of this article is as follows. Section 2 presents the Aitken-Additive Schwarz-Waveform Relaxation (AASWR) algorithm in the one space dimension case. We analyze also various numerical experiences with the Steffensen variant of the method applied to linear and non-linear diffusion problems. Section 3 generalizes the algorithm to multi-dimensional space problems. Section 4 is our conclusion.

II. Aitken-Schwarz Method for Linear Operators in One Space Dimension

For completeness, we review the basic Aitken-Additive-Schwarz (AAS) method for linear elliptic problems in one dimension. The details on the algorithm can be found for example in [16].

A. Aitken-Schwarz Algorithm

Let us consider a linear problem

$$L[u] = f \text{ in } \Omega = (0,1),\ U_{|\partial\Omega} = 0. \tag{1}$$

Let $\Omega_i = (y_i^l, y_i^r),\ i = 1..q$ be a partition of Ω with $y_2^l < y_1^r < y_3^l < y_2^r, ..., y_q^l < y_{q-1}^r$. We consider the additive Schwarz algorithm

Iterate on n until convergence

```
$for\ i = 1..q, do$
        $L[u_i^{n+1}] = f\ in\ \Omega_i,\ u_i^{n+1}(y_i^l) = u_{i-1}^n(y_i^l),\ u_i^{n+1}(y_i^r) = u_{i+1}^n(y_i^r),$
  $enddo$
```

Let us denote $u_i^{l,n+1} = u_i^n(y_i^l)$, $u_i^{r,n+1} = u_i^n(y_i^r)$ and $\tilde{u}^n$ (respectively $\tilde{u}$) be the n iterated (respectively exact) solution restricted at the interface, i.e

$$\tilde{u}^n = (u_2^{l,n}, u_1^{r,n}, u_3^{l,n}, u_2^{r,n}, ..., u_q^{l,n}, u_{q-1}^{r,n})$$

The operator $\tilde{u}^n \rightarrow \tilde{u}^{n+1}$ is linear. Let P be its matrix. P has the following pentadiagonal structure:

$$\left|\begin{array}{cccccccc}
0 & \delta_1^r & 0 & 0 & .... & & & \\
\delta_2^{l,l} & 0 & 0 & \delta_2^{l,r} & ... & & & \\
\delta_2^{r,l} & 0 & 0 & \delta_2^{r,r} & ... & & & \\
 & & & & & & & \\
 & & & ... & \delta_{q-1}^{l,l} & 0 & 0 & \delta_{q-1}^{l,r} \\
 & & & ... & \delta_{q-1}^{r,l} & 0 & 0 & \delta_{q-1}^{r,r} \\
 & & & ... & 0 & 0 & \delta_q^l & 0
\end{array}\right|$$

The sub-blocks $P_i = \begin{vmatrix} \delta_i^{l,l} & \delta_i^{l,r} \\ \delta_i^{r,l} & \delta_i^{r,r} \end{vmatrix}$ $i = 2..q-1$ can be computed as follows. Let v be the solution of

$$L[v] = 0 \; in \; \Omega_i, \; v(y_i^l) = 1, \; v(y_i^r) = 0, \tag{2}$$

and w be the solution of

$$L[w] = 0 \; in \; \Omega_i, \; w(y_i^l) = 0, \; w(y_i^r) = 1. \tag{3}$$

We have then $\delta_i^{l,l} = v(y_{i-1}^r)$, $\delta_i^{l,r} = v(y_{i+1}^l)$ $\delta_i^{r,l} = w(y_{i-1}^r)$ and $\delta_i^{r,r} = w(y_{i+1}^l)$. δ_1^r and δ_q^l can be computed in a similar way. We observe that this computation of the sub-blocks P_i can be done with embarrassing parallelism.

From the equality

$$\tilde{u}^{n+1} - \tilde{u} = P(\tilde{u}^n - \tilde{u}),$$

one writes the generalized Aitken acceleration as follows:

$$\tilde{u}^\infty = (Id - P)^{-1}(\tilde{u}^{n+1} - P\tilde{u}^n). \tag{4}$$

If the additive Schwarz method converges, then $||P|| < 1$ and $Id - P$ is non-singular. However regardless of the convergence of the additive Schwarz algorithm, and provided that $Id - P$ is non singular, the AS algorithm is a direct solver.

The algorithm is then

- Step 1 : compute analytically or numerically in parallel each sub-block P_i from each sub-problems (2) and (3).
- Step 2: apply one additive Schwarz iterate.
- Step 3: apply generalized Aitken acceleration on the interfaces based on (4) with $n = 0$.
- Step 4: compute in parallel the solution for each sub-domain.

From the point of view of parallelism step 1 and step 4 does not requires any communication. Step2 requires local communication between sub-domains that overlap. Step 3 on the contrary requires global communication.

This algorithm might be applied to solve each step of the time integration of the following Initial Boundary Value Problem (IBVP):

$$\frac{\partial u}{\partial t} \; = \; L[u] \; + \; f(x,t), \; (x,t) \in \Omega = (0,1) \times (0,T), \tag{5}$$

$$u(x,0) \; = \; u_o(x), \; x \in (0,1), \tag{6}$$

$$u(0,t) = a(t), \; u(1,t) = b(t), \; t \in (0,T). \tag{7}$$

To illustrate the idea, let us consider the following semi-discretized problem with a first order Euler implicit scheme in time:

$$\frac{U^{k+1} - U^k}{dt} \; = \; L[U^{k+1}] \; + \; f(.,t^{k+1}), \; k = 0,..,M-1, \tag{8}$$

$$U^0 \; = \; u_o, \tag{9}$$

$$U_0^{k+1} = \; a(t^{k+1}), \; U_N^{k+1} = b(t^{k+1}), \; k = 0,..,M-1, \tag{10}$$

At each time step k we have to solve the following BVP problem:

$$-dt \, L[U^{k+1}] \; + \; U^{k+1} \; = \; U^k \; + \; dt \, f(.,t^{k+1}), \tag{11}$$

$$U_0^{k+1} = \; a(t^{k+1}), \; U_N^{k+1} = b(t^{k+1}), \; k = 0,..,M-1, \tag{12}$$

This problem can be solved with the AAS algorithm. While the AAS algorithm applied to this problem is parallel, it requires message passing every time step for Step 2 and 3. Let us denote by τ the overhead due to the latency time of the network for message passing at each time step. The total overhead due to latency is of order $M \; \tau$ where M is the number of time steps. This overhead is very significant for high latency networks and moderate size problems. The ASWR gives flexibility to decide how often in time one exchanges the interfaces between subdomains. Let us denote by $\tilde{M}$ the number of time steps computed at each ASWR iterate. The ASWR algorithm writes:

Iterate on n until convergence with the set of $\tilde{M}$ time steps

```
$for\ i = 1..q, do$
    $\frac{U^{k+1,n+1}-U^{k,n+1}}{dt} \ = \ L[U^{k+1,n+1}] \ + \ f(.,t^{k+1}), \ k=0,..,\tilde{M}-1,$
    $U_i^{k+1,n+1}(y_i^l) = U_{i-1}^{k+1,n}(y_i^l), \ U_i^{k+1,n+1}(y_i^r) = U_{i+1}^{k+1,n}(y_i^r), \ k=0,..,\tilde{M}-1,$
$enddo$
```

$\tilde{M}$ can be chosen in the range $1 \leq \tilde{M} \leq M$. This choice is balanced by two conflicting factors. $\tilde{M}$ should be small if the time step must be updated for accuracy purposes. $\tilde{M}$ is also limited by the number of time steps that can be efficiently accessed and stored in memory. On the other hand, the larger $\tilde{M}$ the less latency overhead one has. This key observation gives the main motivation to generalize the Aitken like acceleration method to ASWR.

Remark: *it is well known for the Helmholtz problem (11, 12), that the smaller the time step the faster might be the decay of the error in space introduced by artificial boundary conditions. In other words the cross terms $\delta^{l,r}, \delta^{r,l}$ in P might be asymptotically negligible. Message passing in the acceleration procedure might be then limited to processors linked by overlapped sub-domains. While this improves dramatically the performance of a parallel scheme, at the expense of some inaccuracy in the direct solver, one may still have a large overhead at each time step due to the high latency of the network combined with the fact that the computation time per time step is very small.*

We will now describe our AASWR algorithm for a domain decomposition in space *and* time, in the one dimensional space case for a linear parabolic problem.

B. Space-Time Aitken-Schwarz Algorithm

Let us consider the Initial Boundary Value Problem (IBVP):

$$\frac{\partial u}{\partial t} \ = \ L[u] \ + \ f(x,t), \ (x,t) \in \Omega = (0,1) \times (0,T), \tag{13}$$

$$u(x,0) \ = \ u_o(x), \ x \in (0,1), \tag{14}$$

$$u(0,t) = a(t), \ u(1,t) = b(t), \ t \in (0,T), \tag{15}$$

where L is a second order linear elliptic operator. We assume that the problem is well posed and has a unique solution.

We introduce the following discretization in space

$$0 = x_0 < x_1 < ... < x_{N-1} < x_N = 1, \ h_j = x_j - x_{j-1},$$

and time

$$t_k = k \ dt, \ k = 0 \dots M, \ dt = \frac{T}{M}.$$

Let us denote by X the column vector $X = (x_1, \dots, x_{N-1})^t$.

We introduce an approximation of the parabolic problem (13) with a first order Euler implicit scheme in time

$$\frac{U^{k+1}-U^k}{dt} \ = \ D \, U^{k+1} \ + \ f(X,t^{k+1}), \ k=0,..,M-1, \tag{16}$$

$$U^0 \ = \ u_o(X), \tag{17}$$

$$U_0^{k+1} = a(t^{k+1}), \ U_N^{k+1} = b(t^{k+1}), \ k=0,..,M-1, \tag{18}$$

where U^k is the column vector $U^k = (U_1^k, \ldots, U_{N-1}^k)^t$. We also introduce the notation U_j for the row vector $U_j = (U_j^1, \ldots, U_j^M)$. Matrix D might be time dependent, but we will not specify this dependence in our notation.

D is a square matrix that comes from a finite difference or a finite element approximation for example. We do not need to specify this approximation. We have chosen for simplicity a first order scheme for the time stepping, but we may use any *one step time integration* scheme, such as a Crank-Nicholson scheme for example.

If the scheme (16-18) converges to the exact continuous solution of the parabolic problem in some appropriate space, U^k is an approximation in space of u at time t^k, and U_j is an approximation in time of u at grid point x_j.

Our purpose is to compute efficiently the numerical solution of the discrete problem (16-18).

At each time step one solves the linear system

$$(Id \, - \, dt \, D)U^{k+1} = F(U^k), \tag{19}$$

where Id is the matrix of the identity operator.
We assume that the matrix $A \, = \, Id \, - \, dt \, D$ of the linear system (19) is regular.

Introducing the matrices $U = (U^1, ..., U^M)$ and $F = (F(U^1), ..., F(U^M))$, we have

$$A \, U \; = \; F, \; U_0 = (a(t^1), \ldots, a(t^M)), \; U_N = (b(t^1), ..., b(t^M)). \tag{20}$$

Let $\Omega_i = (y_i^l, y_i^r), \; i = 1..q$, be a partition of Ω with

$$x_0 = y_1^l < y_2^l < y_1^r < y_3^l < y_2^r, \ldots, y_q^l < y_{q-1}^r < y_q^r = x_N,$$

and

$$\{y_2^l, y_1^r, \ldots, y_q^l, y_{q-1}^r\} \subset \{x_1, \ldots, x_{N-1}\}.$$

One iteration of the ASWR algorithm writes

```
$for \; i = 1..q, do$
        $A_i \; V_i^{n+1} \; = \; F_i, \; in \; \Omega_i \times (0, T),$
        $V_i^{n+1}(y_i^l) = V_{i-1}^n(y_i^l), \; V_i^{n+1}(y_i^r) = V_{i+1}^n(y_i^r),$
$enddo$
```

where A_i is the appropriate sub-block of A corresponding to the discretization of the IBVP problem in $\Omega_i \times (0, T)$.

This algorithm generates a sequence of vectors $W^{k_s} = (V_2^{l,k_s}, V_1^{r,k_s}, V_3^{l,k_s}, V_2^{r,k_s}, \ldots, V_q^{l,k_s})$ corresponding to the boundary values on the set

$$\mathcal{S} = (y_2^l, y_1^r, y_3^l, y_2^r, \ldots, y_q^l, y_{q-1}^r) \times (t^1, ..., t^M)$$

of the V_i for each iterate k.

The proof of convergence of the additive Schwarz waveform relaxation on the continuous problem (13) with the heat equation given in [10] is based on the maximum principle.

The convergence of the ASWR algorithm at the discrete level follows from a discrete maximum principle as well: let us suppose that all discrete subproblems

$$A_i \; V \; = \; 0, \; V_0 = (a(t^1), \ldots, a(t^M)), \; V_N = (0, ..., 0), \tag{21}$$

satisfy the following condition:

$$||V_{N-1}||_\infty \; < \; ||a||_\infty, \tag{22}$$

and similarly for

$$A_i \; V \; = \; 0, \; V_0 = (0, \ldots, 0), \; V_N = (b(t^1), ..., b(t^M)), \tag{23}$$

$$||V_1||_\infty \; < \; ||b||_\infty. \tag{24}$$

Then it is straightforward to show that the sequence W^{k_s} converges to the trace of the exact solution of the discrete problem (16-18) on the interface $\mathcal{S}$.

Remark: *properties (22) and (24) hold for the classical finite difference scheme with the heat equation problem [26]:*

$$\frac{U_j^{k+1} - U_j^k}{dt} \; = \; \frac{1}{\bar{h}_j} \{ \frac{U_{j+1}^{k+1} - U_j^{k+1}}{h_{j+1}} \; - \; \frac{U_j^{k+1} - U_{j-1}^{k+1}}{h_j} \} \; + \; f(x_j, t^{k+1}), \tag{25}$$

$$\text{with } \bar{h}_j = \frac{1}{2}(h_{j+1} + h_j). \tag{26}$$

ASWR converges then for that scheme. We will see later on that the convergence of the ASWR scheme is not a necessary condition to construct the acceleration procedure AASWR

Because the parabolic problem (13) is linear, the trace transfer operator

$$W^{k_s+1} - W^{\infty} \rightarrow W^{k_s} - W^{\infty}$$

is linear. Let P denote its matrix. P has the following pentadiagonal structure:

$$\left| \begin{array}{cccccccc} 0 & P_1^r & 0 & 0 & .... & & & \\ P_2^{l,l} & 0 & 0 & P_2^{l,r} & ... & & & \\ P_2^{r,l} & 0 & 0 & P_2^{r,r} & ... & & & \\ & & & & & & & \\ & & & ... & P_{q-1}^{l,l} & 0 & 0 & P_{q-1}^{l,r} \\ & & & ... & P_{q-1}^{r,l} & 0 & 0 & P_{q-1}^{r,r} \\ & & & ... & 0 & 0 & P_q^l & 0 \end{array} \right|$$

P is a matrix of size $(2(q-1)(M-1))^2$ with the block $P_i^{l,l}, P_i^{l,r}, P_i^{r,l}, P_i^{r,r}$ that are square matrices of size $(M-1)^2$. Let us use the same generic notation Id for the matrix of the identity operator no matter the dimension of the matrix. If the matrix P is known and the matrix $Id - P$ is regular, one step of the ASWR provides enough information to reconstruct the exact interface values by solving the linear system

$$(Id - P)W^{\infty} \; = \; W^1 - P\, W^0. \tag{27}$$

We can then define the algorithm

Algorithm (I)

- Step 1 compute the first iterate of ASWR.
- Step 2 solve the linear problem (27).
- Step 3 compute the second iterate using the exact boundary value W^{∞}.

We show in Section 2.2 that under some circumstances one can easily construct the matrix P or an approximation $\tilde{P}$ of P.

We observe that this algorithm is a direct solver provided that $Id - P$ is regular, no matter the overlap, or the fact that ASWR converges or not. This method is a generalization of the Aitken-Schwarz algorithm described in [16] for the case of linear elliptic operators. We call *algorithm (I)* the Aitken-Additive Schwarz waveform relaxation algorithm. We have the following result

Theorem 2.1: If the ASWR algorithm converges, then AASWR is a direct solver.

Proof: If ASWR converges the sequence W^{k_s} converges. From

$$W^{k_s+1} - W^{\infty} \; = \; P\, (W^{k_s} - W^{\infty}),$$

we have $||P|| < 1$. Consequently $Id - P$ is a regular matrix. (27) has then a unique solution and AASWR is a direct solver □

We have then the following corollary of Theorem 2.1,

Corollary 1 *If each IBVP associated to matrix A_i in (21,23) satisfies the properties (22) and (24) then AASWR is a direct solver.*

This lemma can be used to show that AASWR is a direct solver for the standard three point scheme with the heat equation problem given in (25).

If one knows only an approximation $\tilde{P}$ of P, one may apply the following acceleration scheme

Algorithm (II)

Iterates on the following three step scheme

- Step 1 compute one iterate of ASWR.
- Step 2 solve the linear problem $(Id - \tilde{P})\tilde{W} \ = \ W^1 - \tilde{P}\, W^0$.
- Step 3 update the boundary values of each sub-problem in $\Omega_i \times (0,T)$ with $\tilde{W}$, i.e. $W^1 := \tilde{W}$.

We call this algorithm the Steffensen-Additive Schwarz Waveform Relaxation (SASWR) algorithm because we iterate on the Aitken acceleration that is inexact here [19], [29]. We have the following result

Theorem 2.2: SASWR algorithm converges if

$$\alpha = ||(Id - \tilde{P})^{-1}\, (P - \tilde{P})|| < 1.$$

The algorithm has then a linear rate of convergence with speed α.

Proof: from

$$W^1 - W^\infty \ = \ P\,(W^0 - W^\infty) \text{ and } W^1 - \tilde{W} \ = \ \tilde{P}\,(W^0 - \tilde{W}),$$

we have

$$\tilde{W} - W^\infty \ = \ P\,(W^0 - W^\infty) - \tilde{P}\,(W^0 - \tilde{W}),$$

$$\tilde{W} - W^\infty \ = \ (P - \tilde{P})\,(W^0 - W^\infty) + \tilde{P}\,(\tilde{W} - W^\infty),$$

and finally

$$\tilde{W} - W^\infty \ = \ (Id - \tilde{P})^{-1}\,(P - \tilde{P})\,(W^0 - W^\infty).$$

From $W^1 := \tilde{W}$, we have

$$||W^1 - W^\infty|| \ < \ \alpha\, ||W^0 - W^\infty||.$$

□

We will see in Section 2.5 how one can increase the sparsity of $\tilde{P}$ while keeping some fast convergence properties of the algorithm. But let us first present how one compute the coefficients of P.

C. Construction of Matrix P

Let us assume that the discretization of the temporal domain has only two time steps, i.e., $(t^0, t^1) = (0, T)$. The blocks $P_i^{\cdot,\cdot}$ are then scalars.

Let v be the solution of

$$A_i v = 0 \ in \ \Omega_i, \ v(y_i^l) = 1, \ v(y_i^r) = 0, \tag{28}$$

and w be the solution of

$$A_i w = 0 \ in \ \Omega_i, \ w(y_i^l) = 0, \ w(y_i^r) = 1. \tag{29}$$

AASWR then reduces to the exact same Aitken-Additive Schwarz algorithm for the Helmholtz operator $-dt\ D + Id$ described in Section 2.1 [16].

The construction of P with a temporal discretization of $(0, T)$ with an arbitrary number of time step $M > 1$ is done as follows.

We choose the following basis of functions

$$\delta_j^k = 1, \text{ if } j = k, \ 0 \text{ otherwise}, \ j, k \in \{1, .., M\}$$

to represent the trace of the solution on the interfaces

$$y_i^{l/r} \ \times \ \{t_1, ..., t_M\}, \ i = 1..q.$$

Let us consider the family of subproblems in $\Omega_i \times (0,T)$,

$$\frac{V_{i,j}^{k+1} - V_{i,j}^{k}}{dt} \;=\; D_i[V_{i,j}^{k+1}],\; k = 0, \ldots, M-1, \tag{30}$$

$$V_{i,j}^{0} \;=\; 0, \tag{31}$$

$$V_{i,j}^{k+1}(y_i^l) \;=\; 0,\; V_{i,j}^{k+1}(y_i^r) \;=\; \delta_j^{k+1},\; k = 0, \ldots, M-1. \tag{32}$$

Let $V_{i,j}$ denote the matrix that is the solution of the discrete problem (30-32). The j column vector of $P^{r,r}$, respectively $P^{r,l}$, is the trace of $V_{i,j}$ on y_{i+1}^l, respectively y_{i-1}^r. $P_i^{r,r}$ and $P_i^{r,l}$ are consequently lower triangular matrices. A similar construction can be done to compute the sub-block $P_i^{l,l}$ and $P_i^{l,r}$.

Let us assume now that all coefficients of the elliptic operator L in the IBVP (13) are time independent.

We notice that all $V_{i,j}$ are obtained from $V_{i,1}$ by a translation in time, i.e.,

$$V_{i,j}(X_i, t) \;=\; V_{i,1}(X_i, t - t_{j-1}),\; t \in \{t_j, \ldots, t_M\}, \tag{33}$$

and

$$V_{i,j}(X_i, t) \;=\; 0,\; t \in \{t_0, t_{j-1}\}. \tag{34}$$

The first column vector of $P^{r,r}$, respectively $P^{r,l}$, is the trace of $V_{i,1}$ on y_{i+1}^l, respectively y_{i-1}^r. From (33) we see that all columns of $P_i^{r,r}$, respectively $P_i^{r,l}$, are obtained from the first column of matrix $P_i^{r,r}$, respectively $P_i^{r,l}$, with no additional computation.

To conclude, the construction of the matrix P of the trace transfer operator is achieved if one computes once and for all the solution of the two following sub-problems in $\Omega_i \times (0,T)$,

$$\frac{V_{i,j}^{k+1} - V_{i,j}^{k}}{dt} \;=\; D_i[V_{i,j}^{k+1}],\; k = 0, \ldots, M-1, \tag{35}$$

$$V_{i,j}^{0} \;=\; 0, \tag{36}$$

$$V_{i,j}^{k+1}(y_i^l) \;=\; \delta_1^{k+1},\; V_{i,j}^{k+1}(y_i^r) \;=\; 0,\; k = 0, \ldots, M-1, \tag{37}$$

and

$$\frac{V_{i,j}^{k+1} - V_{i,j}^{k}}{dt} \;=\; D_i[V_{i,j}^{k+1}],\; k = 0, \ldots, M-1, \tag{38}$$

$$V_{i,j}^{0} \;=\; 0, \tag{39}$$

$$V_{i,j}^{k+1}(y_i^l) \;=\; 0,\; V_{i,j}^{k+1}(y_i^r) \;=\; \delta_1^{k+1},\; k = 0, \ldots, M-1. \tag{40}$$

Sub-problems (35-37) and (38-40) are the analogue of (28) and (29) in the Aitken-Schwarz algorithm for the AASWR algorithm.

Remark: *All sub-problems listed above needed for the construction of the trace transfer operator matrix can be solved with embarrassing parallelism.*

We are going now to illustrate the method with the classical finite difference approximation (25) for the one dimensional heat equation.

D. Application to the One Dimensional Heat Equation

Our linear test case is the heat equation problem in the domain $(0,1) \times (0,T)$. The grid in space is regular with constant space step h. We keep the number of grid points per sub-domain to be $N_b = 20$, and the time step is $dt = h$. Further the overlap is kept minimum, that is a one mesh interval.

The total number of grid points is $N \;=\; N_b \;+\; (q-1)\;(N_b - 1)$, where q is the number of sub-domains.

The Standard Method (SM) to integrate (25) is to solve each time step with a direct tridiagonal solver. The LU decomposition of the tridiagonal system can be computed once, since the same linear system is solved at every time step. The arithmetic complexity of the SM is then $n_1 = C_1\; N\; M$, where C_1 is an integer. $C_1 = 5$ for Gaussian elimination. Let us compare the arithmetic complexity of the AASWR algorithm with that of SM. The arithmetic complexity of one iterate of the ASWR algorithm is $n_q \;=\; C_1\; M\; (N + q - 1)$ which is asymptotically equivalent to n_1.

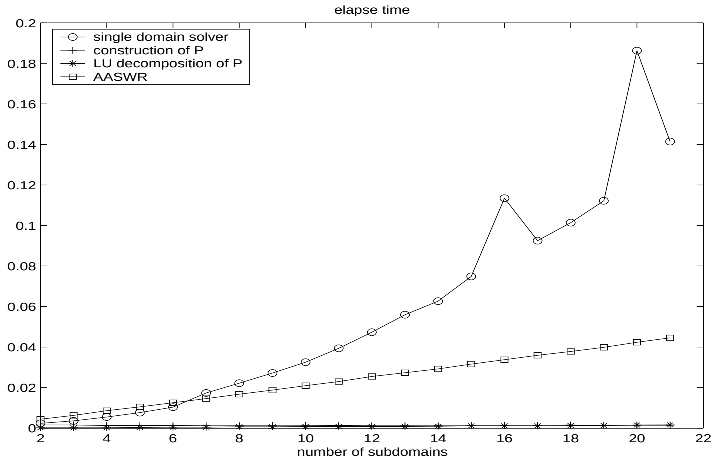

Fig. 1. Convergence of ASWR and AASWR for the heat equation.

All subdomains correspond to the same finite difference operator and this operator has no time dependent coefficient. Consequently, the construction of the matrix P requires to solve one sub-domain problem (35-37) or (38-40). The arithmetic complexity of the construction of P is then $C_1\ M\ \frac{N+q-1}{q}$ and can be neglected against n_q.

The acceleration step requires to solve the sparse linear system (27). Each block is a triangular matrix of dimension M the number of time steps. The solution can be computed forward in time by solving at each time step a penta-diagonal system analogous to P in Section 2.1, with a linear combination of the trace of the solution on the interface $\mathcal{S}$ at all previous time steps on the right hand side .

The arithmetic complexity of this blockwise backward elimination procedure is then asymptotically $n_{interface} = C_2 M[(q-1)^2 + O(q)]$ floating point operations [1]. $n_{interface}$ is small compare to n_q as long as

$$q << \sqrt{N}. \tag{41}$$

Overall and assuming (41), the number of floating point operations (flops) for the AASWR procedure is about twice the number of flops for the standard SM with no domain decomposition. However modern computer architecture do not perform linearly with the number of flops. The domain decomposition algorithm might still have a better elapse time performance than the SM thanks to memory access limitations and cache memory effect. To illustrate this concept we have performed the computation with both algorithm SM and AASWR on a PC running matlab with a Pentium 4 2.66GHz. This PC has one GB of main memory. With moderate number of time steps and large problem size, the advantage of the AASWR algorithms over the SM becomes clear. Figure 1 provides some comparison between both algorithm with ten time steps, i.e $M = 10$, $N_b = 20$ and a number of subdomains that varies from 2 to 20. The elapsed time is given in seconds and averages the measurement provided by one hundred runs. We remind here that the size of the problems grows linearly with the number of domains according to $N\ =\ N_b\ +\ (q-1)\ (N_b-1)$. Overall the construction of P and the acceleration step has negligible elapse time. In AASWR the elapse time grows linearly with the number of subdomains. AASWR performs better than SM for $q > 6$. We believe that the cache size is responsible for the two peaks in the curve giving the performance of the SM. On the contrary the AASWR seems to be insensitive to the cache size for the dimension of the sub-domain that has been chosen here.

To verify if the Aitken-like acceleration impacts the stability of the AASWR or not, we have checked the accuracy of our solution with larger problem size than in the previous test case, that is $M = 20$ and $N_d = 40$. The RHS and boundary conditions in the heat equation are such that the exact solution is the traveling wave:

$$u(x,t)\ =\ \exp\left(-20\ z^2\right),\ \text{where}\ z = x - 1/3 - 1/5\ t. \tag{42}$$

Figure 2 gives the condition number of the matrix $(Id - P)$ used in the acceleration step. This condition number grows linearly with the number of subdomains, which is proportional to the problem size in space N. Figure 3 provides some indication of the accuracy of the solver. We choose to measure the error in the discrete L_∞ norm for all the domain of computation $\Omega \times (0, M\ dt)$. The numerical accuracy of AASWR is overall compatible with the condition number of the linear discrete problem (25), and the acceleration procedure does not seems to impact significantly the accuracy of our exact solver.

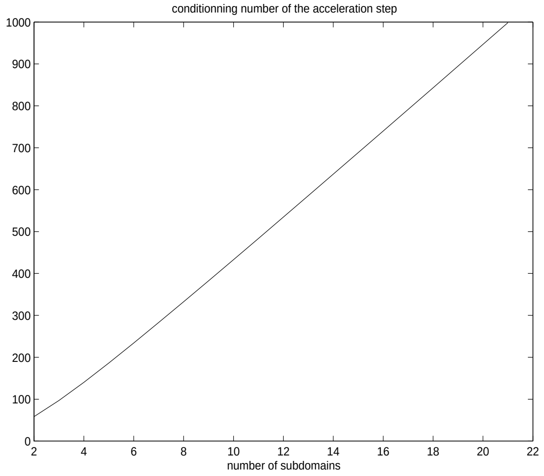

Fig. 2. Conditionning number of the linear system (27).

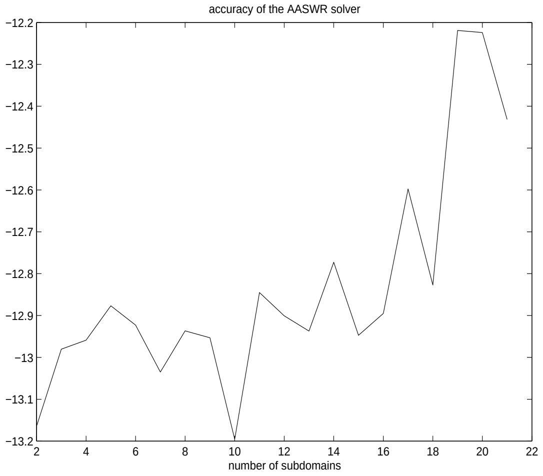

Fig. 3. Difference in maximum norm between the solution of SM and AASWR solvers

We discuss in the next section the application of the method to more general one D heat equation problems such as weakly nonlinear equation.

E. Generalization and Construction of Approximations of P

In the most general situation where the elliptic operator L has coefficients that are time dependent, the construction of the operator P becomes numerically intense. In principle, for each element of the vector basis that one uses to represent the trace of the solution on the interface $\mathcal{S}$, one has to compute a solution of a sub-problem similar to (35-37). ASWR will not be cost effective (on a sequential computer) unless one uses a domain decomposition with very few time steps q, or a good low cost approximation of P.

Remark: *because all subproblems (35-37) can be computed prior to the time stepping with embarrassing parallelism, the general situation might be still manageable on a grid of computers.*

To apply our technique to more general heat equation problems, we can try to construct cost effective approximations of the trace transfer operator.

We discuss four complementary ways of approximating the trace transfer operator P. The impact on the efficiency of the scheme can be *a posteriori* assessed by the estimate in Theorem 2.

• (i) First, one can neglect a set of $p < M$ first rows of the sub-block $P_i^{\cdot,\cdot}$ in matrix P. This minimizes the number of sub-problems to be solved when the elliptic operator L has time dependent coefficients. This is equivalent to neglecting the time dependency of the solution $u(x,t)$ in Ω_i on artificial boundary conditions $u(y_i^{l/r},t)$ for $t \in (0, p\, dt)$. For the heat equation the error can be analyzed analytically as in [10] using the $erfc$ function. This approximation of P might be effective if and only if one is interested in the accuracy of the final solution at the end time $t = T$, rather than on the whole time interval $(0,T)$. As a matter of fact we still have $||P - \tilde{P}||$ of order one in this situation.

• (ii) Second, one can neglect the cross terms $P^{r,l}$ and $P^{l,r}$ in P. The quality of this approximation depends on the fast decay of the error in space. For the heat equation, for example, we have seen that each time step corresponds to the solution of a Helmholtz problem,

$$-\, dt\; u'' \,+ u \;=\; rhs.$$

The influence of the error at an artificial boundary y decays as the function

$$\exp(-\frac{|x-y|}{\sqrt{dt}})$$

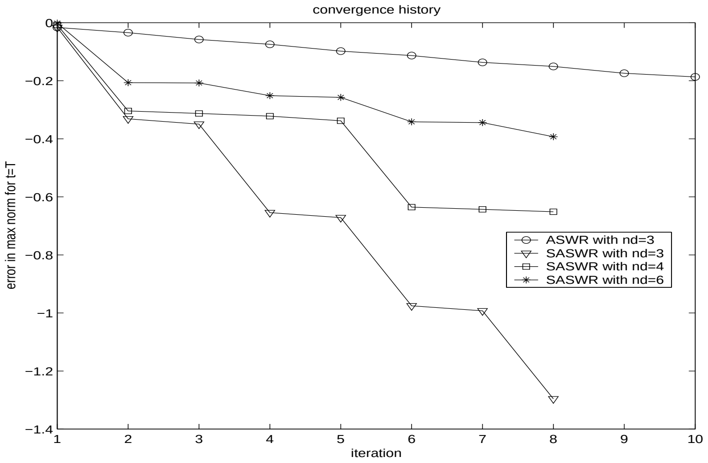

Fig. 4. Neglecting the cross terms in matrix P with the heat equation $\partial_t u = \partial_{xx} u + f$.

at every time step. The smaller the time step, the better the approximation. The main advantage of this approximation relies on the fact that the algorithm does not require any global communication to solve the acceleration step as in AASWR. This is, however, not lowering the complexity of the construction of the approximation $\tilde{P}$ when the elliptic operator has time dependent coefficients.

• (iii) Third, one can use a coarse global representation of the trace on the artificial interface $\mathcal{S}$ following the same approach as in [12]. This method will not be developed here.

• (iv) Fourth, one can use a constant coefficient approximation of L in the linear case or an average in time approximation of the Jacobian of the operator in the nonlinear case. Thus the construction of $\tilde{P}$ requires the parallel computation of two local problems analogous to (35-38).

Let us illustrate the performance of the Steffensen variation of the algorithm, i.e., *algorithm (II),* when one uses an approximation of the trace transfer operator.

In all numerical experiments below we force the solution of the IBVP to be the traveling wave (42).

In our experiments we evaluate the impact of neglecting the non-local domain dependencies as described in (ii). We take the block diagonal approximation of P and neglect the cross terms $P^{l,r}, \; P^{r,l}$. Figure 2 reports on the performance of SAWR with the heat equation problem, while Figure 3 applies to the reaction-diffusion problem

$$\partial_t u = \partial_{xx} u - \mu \, u + f, \text{ with } \mu = N. \tag{43}$$

The convergence speed deteriorates dramatically when the number of subdomains grows for the heat equation test case. The situation improves for the Helmholtz problem (43) with large μ. This observation will be used for multi-dimensional problems in see Section 3.2.

Finally, let us consider a nonlinear problem as follows

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} + f(u). \tag{44}$$

Following argument (iv) above, we use as an approximation of P the trace transfer operator that corresponds to the heat equation operator $\partial_t - \partial_{xx}$ that has time independent coefficients.

In our experiment we choose

$$f(u) \; = \; \mu \, tanh(u).$$

If L_t^h denotes the Jacobian of the discrete non-linear operator corresponding to the time integration of (44), $f(u)$ is a second order perturbation of this operator, i.e $||f(u)|| = \mu << ||L_t^h|| \sim h^{-2}$. In Figures 4 and 5, we take $\mu = 1$ and $\mu = 5$, respectively. We observe a linear rate of convergence of SASWR. The speed of convergence deteriorates rapidly as μ grows. We speculate that the speed of convergence of the SAWR algorithms will be quadratic if one uses L_t^h instead of the heat operator. It is, however, impractical to compute the matrix of the corresponding trace transfer operator because of the time dependency of the coefficient of the Jacobian.

We are going to show that most of the results obtained in this section can be extended to multi-dimensional parabolic problems provided L is separable or a weak perturbation of a separable operator.

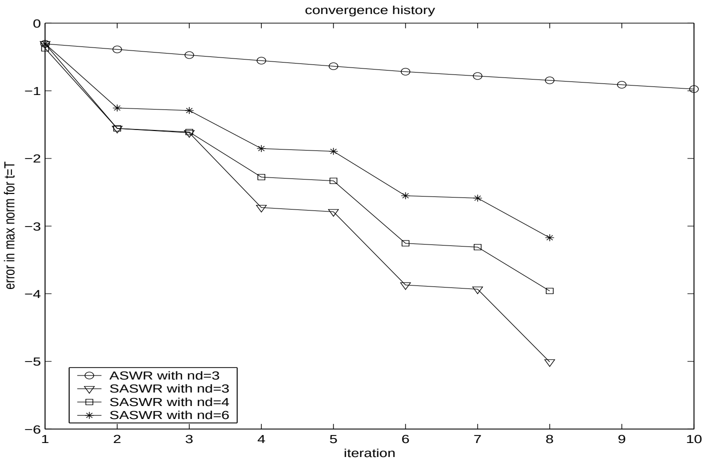

Fig. 5. Neglecting the cross terms in matrix P with the reaction-diffusion equation $\partial_t u = \partial_{xx} u - N\ u + f$.

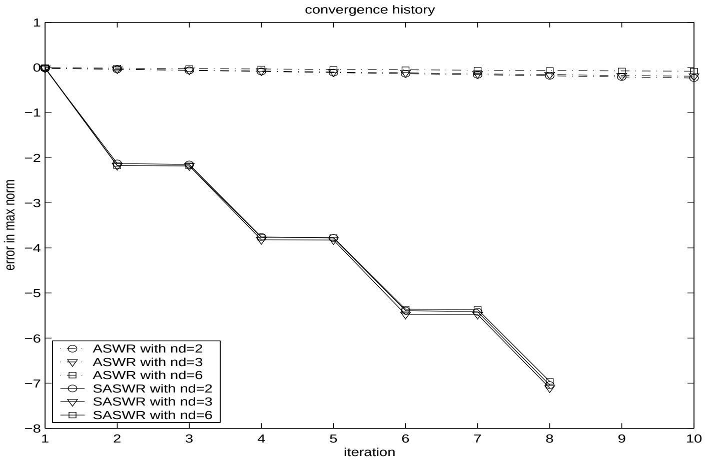

Fig. 6. Performance of an SASWR algorithm for a solution of the equation $\partial_t u = \partial_{xx} u + tanh(u)$.

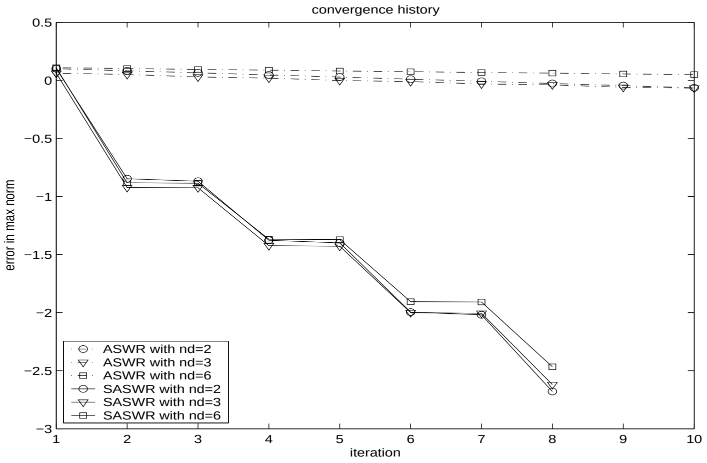

Fig. 7. Same as Figure 7 but with $\partial_t u = \partial_{xx} u + 5\ tanh(u)$.

III. Aitken-Schwarz Method for Linear Operators in the Multidimensional Case

To simplify the notations we will restrict ourselves to two space dimensions. We further assume that the domain Ω is a square discretized by a rectangular Cartesian grid with arbitrary space steps in each direction. The general situation with arbitrary shape domains might be treated following the line of [12] and will be reported elsewhere.

Let us consider the IBVP:

$$\frac{\partial u}{\partial t} \;=\; L[u] \;+\; f(x,y,t),\; (x,y,t)\in\Omega=(0,1)^2\times(0,T), \tag{45}$$

$$u(x,y,0) \;=\; u_o(x,y),\; (x,y)\in(0,1)^2, \tag{46}$$

$$u(0,y,t)=a(y,t),\; u(1,y,t)=b(y,t),\; y\in(0,1),\; t\in(0,T), \tag{47}$$

$$u(x,0,t)=c(x,t),\; u(x,1,t)=d(x,t),\; x\in(0,1),\; t\in(0,T), \tag{48}$$

where L is a second order linear elliptic operator. We assume that the problem is well posed and has a unique solution. Using an appropriate shift in space we can restrict ourselves to the special case:

$$c(x,t)=d(x,t)=0,\;\forall(x,t)\in\;(0,1)\times(0,T). \tag{49}$$

We introduce the following discretization in space

$$0=x_0<x_1<\ldots<x_{_{Nx-1}}<x_{_{Nx}}=1,\; h_{xj}=x_j-x_{j-1},$$

$$0=y_0<y_1<\ldots<y_{_{Ny-1}}<y_{_{Ny}}=1,\; h_{yj}=y_j-y_{j-1},$$

and time

$$t_k=k\;dt,\; k\in\{0,\ldots,M\},\; dt=\frac{T}{M}.$$

The domain $\Omega=(0,1)^2$ is decomposed into q overlapping strips $\Omega_i=(y_i^l,y_i^r)\times(0,1)$.

We first present the general algorithm when L is a separable linear operator and refer to the theoretical framework established in [2] for elliptic operator.

A. The General Algorithm for Separable Operator

Let us assume that the elliptic operator L is a separable second order operator as follows:

$$L=L_1+L_2$$

$$L_1=e_1\partial_{xx}+f_1\partial_x+g_1,\;\; L_2=e_2\partial_{yy}+f_2\partial_y+g_2.$$

e_1,f_1,g_1 are functions of x only, and e_2,f_2,g_2 are functions of y only. We write the discretized problem as follows

$$\frac{U^{k+1}-U^k}{dt} \;=\; D_{xx}[U^{k+1}] \;+\; D_{yy}[U^{k+1}] \;+\; f(X,Y,t^{k+1}),\; k=0,\ldots,M-1, \tag{50}$$

with appropriate boundary conditions corresponding to (46-48).

Our main objective is to rewrite the discretized problem in such a way that we can reuse the results of Section 2 that is for the one space dimension case. Let us assume that D_{yy} has a family of (N_y-1) independent eigenvectors $\Phi_j,\; j=1,..,N_y$ in $\mathbb{R}^{N_y-1}$ with corresponding eigenvalues μ_j.

The Φ_j are implicitly the numerical approximation in $(0,1)$ of the solutions of the following continuous eigenvector problems:

$$L_2[v(y)] \;=\; \mu\, v(y),\; v(0)=v(1)=0. \tag{51}$$

Let us introduce the decomposition

$$U^k(X,Y,t)=\sum_{j=1}^{N_y-1}\Lambda_j^k(X,t)\Phi_j(Y),\; u_o(X,Y)=\sum_{j=1}^{N_y-1}\lambda_j^k(X)\Phi_j(Y),$$

$$f(X,Y,t^k) = \sum_{j=1}^{N_y-1} f_j^k(X,t^k)\Phi_j(Y),$$

and

$$a(Y,t^k) = \sum_{j=1}^{N_y-1} a_j(t^k)\Phi_j(Y),\ b(Y,t^k) = \sum_{j=1}^{N_y-1} b_j(t^k)\Phi_j(Y).$$

The discrete solution of (50) satisfies the following set of $(Ny-1)$ uncoupled problems

$$\frac{\Lambda_j^{k+1}-\Lambda_j^k}{dt} \ = \ D_{xx}[\Lambda_j^{k+1}] \ + \ \mu_j \ \Lambda^{k+1} \ + \ f_j(X,t^{k+1}),\ k=0,\ldots,M-1, \tag{52}$$

$$\Lambda_j^0 \ = \ \lambda_j(X), \tag{53}$$

$$\Lambda^{k+1}(x_0) = a_j(t^{k+1}),\ \Lambda^{k+1}(x_{N_x}) = b_j(t^{k+1}),\ k=0,\ldots,M-1. \tag{54}$$

Let us extend the notation of Section 2 for the trace transfer operator

$$W^{k_s} \ - W^{\infty} \ \rightarrow \ W^{k_s+1} \ - W^{\infty},$$

to the two dimensional space case.

This operator can also be decomposed into (N_y-1) independent trace transfer operators

$$W_j^{k_s} \ - W_j^{\infty} \ \rightarrow \ W_j^{k_s+1} \ - W_j^{\infty},$$

that apply to each component of the trace of the solution expanded in the eigenvector basis $E = \{\Phi_j,\ j = 1,\ldots,(N_y-1)\}$. Let Q_j be the matrix of this linear operator. The matrix P has now a (N_y-1) diagonal block structure, where each block is the matrix Q_j. *Algorithm (I)* can be generalized as follows

Algorithm (I bis)

- Step 1 compute the first iterate of ASWR for the two space dimension parabolic problem (45).
- Step 2 expand the trace of the solution in the eigenvector basis E and solve the linear problem component wise
$$(Id-Q_j)W_j^{\infty} \ = \ W_j^1 - Q_j\ W_j^0,\ \forall j \in \{1,\ldots,(Ny-1)\}. \tag{55}$$
Assemble the boundary condition $W^{\infty} = \sum_{j=1,\ldots,Ny-1} W_j^{\infty}\ \Phi_j$.
- Step 3 compute the second iterate using the exact boundary value W^{∞}.

Let us emphasize that the sub-domain problems in $\Omega_j \times (0,T)$ can be integrated by any existing efficient numerical solver. It is only the acceleration step 2 that requires the decomposition of the *trace* of the solution into the eigenvector basis E.

Because all eigenvector components of the solution are independents, we have then as in the one dimension space case

Theorem 3.1: If the ASWR algorithm converges, then AASWR is a direct solver.

The construction of the Q_j can be done exactly as in the one space dimension case.

This preprocessing step consists of solving $2\ q\ (Ny-1)$ problems analogous to (35-37) in $(y_i^l, y_i^r) \times (0,T)$,

$$\frac{V_i^{k+1}-V_i^k}{dt} \ = \ D_{xx}[V_i^{k+1}] \ + \ \mu_j\ V_i^{k+1},\ k=0,\ldots,M-1, \tag{56}$$

$$V_i^0 \ = \ 0, \tag{57}$$

$$V_i^{k+1}(y_i^l) \ = \ 0,\ V_i^{k+1}(y_i^r) \ = \ \delta_1^{k+1},\ k=0,\ldots,M-1, \tag{58}$$

and

$$\frac{V_i^{k+1}-V_i^k}{dt} \ = \ D_{xx}[V_i^{k+1}] \ + \ \mu_j\ V_i^{k+1},\ k=0,\ldots,M-1, \tag{59}$$

$$V_i^0 \ = \ 0, \tag{60}$$

$$V_i^{k+1}(y_i^l) \ = \ \delta_1^{k+1},\ V_i^{k+1}(y_i^r) \ = \ 0,\ k=0,\ldots,M-1. \tag{61}$$

for each eigenvalue $\mu_j,\ j = 1, \ldots, N_y - 1$ and each sub-domain $i = 1, \ldots, q$.

These sub-problems (56-58) and (59-61) can be computed with embarrassing parallelism.

Thanks to the block diagonal decomposition of P one can generalize *algorithm (II)* to two space dimensions as well and repeat Theorem 2.2. We will see in Section 3.2 how one can take advantage of the eventual large eigenvalue $|\mu_j| >> 1$ of D_{yy} in a parallel implementation. The extension of the method presented here to the three dimensional case under the assumption of separability of the discrete elliptic operator is straightforward.

We now illustrate our method for the heat equation problem.

B. Application to the Heat Equation

We consider the standard heat equation problem discretized in space on a five point stencil with central finite differences on a regular Cartesian mesh. We have in this situation

$$\Phi_j = \sin(j\ \pi\ y),\ \mu_j\ =\ -\frac{4}{h_y^2}\ \sin^2(j\pi\frac{h_y}{2}).$$

The time stepping:

$$\frac{U_{i,j}^{k+1} - U_{i,j}^{k}}{dt} = \frac{U_{i+1,j} - 2U_{i,j} + U_{i-1,j}}{h_x^2} + \frac{U_{i,j+1} - 2U_{i,j} + U_{i,j-1}}{h_y^2}\ +\ F_i, j^{k+1},$$

can be processed fairly efficiently using the Fourier sin transform in space. By following a fast Fourier like procedure to compute the data in the $Ny - 1$ uncoupled IBVP (52 - 54) with a tridiagonal solver for each set of linear equations. Using notation similar to the one in Section 2.4, this SM involves

$$n_1\ =\ C_3\ N_x\ M\ N_y\ \log(N_y)\ \text{flops},$$

where C_3 is an integer constant.

Following the same steps as in Section 2.4, one can show that AASWR requires roughly two times as many floating point operations. But as stated before the AASWR algorithm is a parallel algorithm fairly tolerant to high latency networks. We have verified also that AASWR performs better than SM on a scalar processor with small number of time steps and large problem size.

We have checked the accuracy of our AASWR solver on a two dimensional problem with singular sources as follows:

$$F(x,y) = \sum_{j=1..3} f_j(t)\delta(x_j, y_j, r_j).$$

The delta functions are equal to one in a disk of center (x_j, y_j) and radius r_j, and zero elsewhere. Figure 8 and 9, gives a representation of the solution with three different time periodic source terms. In this simulation the time interval is $(0, \pi)$. The space grid is given by $N_x = 200$ and $N_y = 197$. We use four strip subdomains. The difference in the discrete maximum norm on $(0,1)^2 \times (0, \pi)$ between the SM solution and the AASWR is of the order of 10^{-13}.

Remark: *our result can be easily generalized to tensorial products of a one dimensional grid with adaptive space stepping. The key hypothesis is the separability of the discrete operator $D_{xx} + D_{yy}$ on the tensorial product of grid. Because h_y is not a constant, the eigenvectors Φ_j are not known analytically and should be computed numerically as in [2].*

We have shown in [3] for ASWR that it was not necessary to accelerate with the exact Aitken'like acceleration formula all sin components of the trace of the solution on the artificial interface. A similar technique applies to the AASWR algorithm for our benchmark problems. We observe that the propagation of the error in space for each component in the eigenvector basis E decays monotonically as $|\mu_j|$ grows. Further one usually has simultaneously $||\Lambda_j^k|| << 1$, for large j components [6].

On can check that the linear convergence rate of ASWR for the sin wave component $\Phi_j(y)$ of the solution is approximatively $\gamma_j(h)$ with

$$\gamma_j(x) = \exp(-\ dist(x)\ (\frac{dt}{1\ -\ \mu_j\ dt})^{\frac{1}{2}}),$$

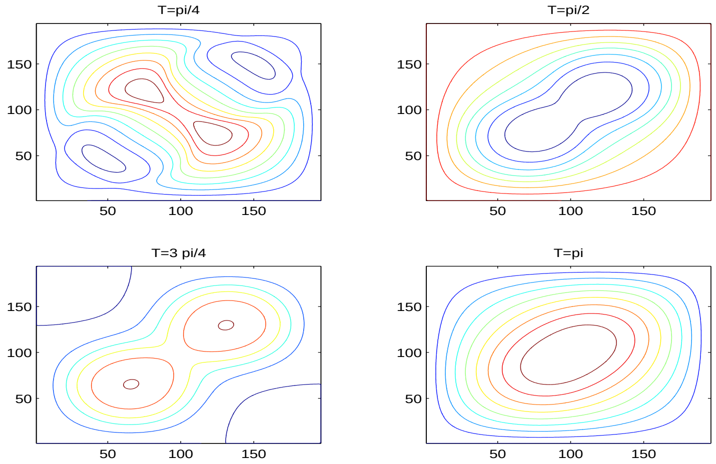

Fig. 8. Contour line of the solution at four different times.

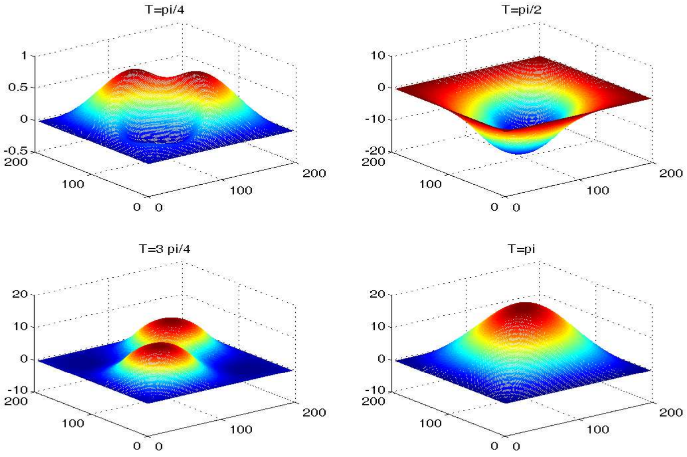

Fig. 9. Representation of the solution at four different times.

and $dist(x)$ is the distance of the grid point of first coordinate x to the artificial interface of $\mathcal{S}$. Further we have $||Q^{r,l}_{j,i}||_\infty$, $||Q^{l,r}_{j,i}||_\infty \sim \gamma_j(1/q-h)$.

For the trace components that correspond to large eigenvalues μ_j, ASWR has a fast linear rate of convergence, and the acceleration may not be necessary. For intermediate values, one may neglect completely the global coupling of all domains in (55), that is to neglect the cross terms $Q^{r,l}_{j,i}$ and $Q^{l,r}_{j,i}$ of the block pentadiagonal matrix Q_j - see Figure 5. For all other eigenvalue components that correspond to the slow decay of the error in space, one must solve the complete linear system as in (55). As observed in [3] a parallel implementation can take advantage of this feature by reducing dramatically the size of the messages between the sub-domains.

We will now discuss some details of the parallel implementation of our method that are specific to space *and* time decomposition.

IV. Discussion of the Parallel Algorithm Implementation

First, let us discuss the properties of AASWR that should be used in a parallel implementation of the method. We will describe theoretically the parallel algorithm using the general framework of message passing on a distributed network of processors and show that our new method provides a rigorous framework to optimize the use of a parallel architecture.

To formalize the algorithm, one introduces the following notations: $V^k(\delta_{right})$ and $V^k(\delta_{left})$, are the solutions at time step $t = t^k$ of the set of problems (35-40) in the one space dimension case, (56-61) respectively in the two space dimension case. $U(W^{k_s}(t^k))$ is the solution of the set of subproblems of the domain decomposition algorithm at iterate k_s with boundary conditions W^{k_s} on $\mathcal{S}$ at time level $t = t^k$.

Prior to the AASWR algorithm one computes in parallel $V^k(\delta_{left})$, and $V^k(\delta_{right})$, in order to construct the matrix P of (27) or its equivalent with multi-dimension problems. This can be done, in the one space dimension case, with no message passing required, on $2 \times q$ processors where q is the number of sub-domains. In the two dimensional case one can distribute the process on $2 \times q \times N_y$ processors since all eigenvector components are given by the solution of a set of N_y decoupled (one space dimension) parabolic problems.

The first step of AASWR is the parallel computation of the solution of the IBVP (13) or (45) in each sub-domain $\Omega_i \times (0, T),\ i = 1 \dots q$. It is important to notice that the acceleration of the interface components of $\mathcal{S}$ at time step t_{k_1} does not depend on the interface components of $\mathcal{S}$ at later time steps $t_{k_2},\ k_2 > k_1$. This property is reflected in the fact that the blocks of the pentadiagonal matrix P are lower tridiagonal matrices. Consequently the resolution of the linear system in the acceleration step (27) or its equivalent with multi-dimension problems can progress with the time stepping on $U(W_0(t^k))$. A message passing parallel implementation of the algorithm should then distribute each time step of the first iterate of W^1 as soon as they are computed.

Similarly as soon as the accelerated time level t^k of W^∞ is produced by Step 2 of *algorithm (I)*, the exact discrete solution at time level t^k can be computed.

To summarize, the AASWR algorithm gives rise to four *parallel* time stepping processes:

- (i) compute V_{left}^{k+1},
- (ii) compute V_{right}^{k+1},
- (iii) compute $U(W^0(t^{k+1}))$,
- (iv) compute $W^\infty(t^k)$ based on (i-iii) at time step t^k and then compute $U(W^\infty(t^k))$.

These four parallel computations need the first time step for (i-iii) to be achieved. In the one space dimensional case, AASWR may run on $4 \times q$ processors!

Let us notice that the parallel implementation of the acceleration step itself (27) is not obvious. We refer to [30] for a discussion of such parallel algorithms. However, the solution of the linear system in (27) is at least an order $\frac{N_x}{q}$ times cheaper than the rest of the calculation in Algorithm I or Ibis. It can be done then redundantly on the q processors that execute (iv).

In practice one should minimize the overhead due to the latency of the network. The communication step should be executed every n_q time steps where n_q is a small integer to be optimized experimentally. One then uses non-blocking communications to overlap communication of sub-domain boundaries by computations of sub-domain solutions. The optimal choice of n_q depends critically on the ratio between the network performance and the floating point performance of the processors. The detailed specific implementation depends strongly on the computer architecture. An experimental study, with particular emphasis on grid computing will be published in a companion paper.

V. Conclusion

In this paper we have shown how to generalize the Aitken-like acceleration method of the additive Schwarz algorithm for elliptic problems to the additive Schwarz waveform relaxation for parabolic problems. This new domain decomposition algorithm is in space *and* time. Since the concept of our acceleration technique is general and might be applied in principle to any block-wise relaxation scheme, we expect that it can be combined with some optimized transmission conditions for the same PDE problem. We have shown that our technique

(1) is a direct solver that requires at most four solves per sub-domain in the case of a one space dimension linear parabolic problem with time independent coefficients, (2) can be applied easily to multi-dimensional problems, provided that the operator is separable in space, (3) is an efficient iterative procedure for parabolic problems that are weak nonlinear perturbations of linear operators with time independent coefficients, (4) provides a rigorous framework to optimize the parallel implementation on a slow network of computers.

A further step in the development of our methodology would be to consider unstructured meshes, and approximate the trace transfer operator with for example, the coarse grid interface approximation presented in [12].

Acknowledgement : This work was supported in part by NSF Grant ACI-0305405.

REFERENCES

[1] P.ARBENTZ AND W.GANDER, *A Survey of Direct Parallel Algorithms for banded Linear Systems*, October 1994 ETH Zurich Departement ftp.inf.ethz.ch/pub/publications/tech-reports/2xx/221.ps.gz

[2] J. BARANGER, M. GARBEY, AND F. OUDIN-DARDUN, *On Aitken like acceleration of Schwarz domain decomposition method using generalized Fourier*, in Domain Decomposition Methods in Science and Engineering, I. Herrera et al., eds., Universidad Nacional Autónoma de México, Mexico City, 2003, pp. 341–348.

[3] N. Barberou, M. Garbey, M. Hess, M. Resch, T. Rossi, J. Toivanen, and D. Tromeur-Dervout, ON THE EFFICIENT META-COMPUTING OF LINEAR AND NONLINEAR ELLIPTIC PROBLEMS, J. Parallel Distrib. Comput. (special issue on grid computing), 63 (2003), pp. 564–577.

[4] Y.Boubendir and A.Bendali, DOMAIN DECOMPOSITION METHODS FOR SOLVING SCATTERING PROBLEMS BY A BOUNDARY ELEMENT METHOD, 13th Int. Conf. on Domain Decomposition Methods DD13, Domain Decomposition Methods in Science and Engineering, CIMNE, Barcelona, N.Debit et Al edt, pp319-326, 2002.

[5] D.Braess, FINITE ELEMENTS, THEORY, FAST SOLVERS, AND APPLICATION IN SOLID MECHANICS, Cambridge Univ. Press, 1997.

[6] M.O. Deville, P.F. Fischer, and E.H. Mund, HIGH-ORDER METHODS FOR INCOMPRESSIBLE FLUID FLOW, Cambridge University Press (2002).

[7] B.Despres, DOMAIN DECOMPOSITION METHOD AND THE HELMHOLTZ PROBLEM, Mathematical and Numerical Aspects of Wave Phenomena, pp 44-52, SIAM, 1991.

[8] B.Despres, DÉCOMPOSITION DE DOMAINE ET PROBLÈME DE HELMHOLTZ, C.R.Acad.Sci.,Paris,t.311, Série I, pp 313-316, 1990.

[9] I.Faille, A CONTROL VOLUME METHOD TO SOLVE AN ELLIPTIC EQUATION ON TWO-DIMENSIONAL IRREGULAR MESHING, Comp.Meth.Appl.Mech.Engrg.,100, pp.275-290, 1992.

[10] M.Gander and H.Zhao, OVERLAPPING SCHWARZ WAVEFROM RELAXATION FOR THE HEAT EQUATION IN N-DIMENSIONS, BIT, Vol 40, No 4, pp 001-004, 2000.

[11] M. Gander, L. Halpern, F. Nataf, OPTIMIZED SCHWARZ METHODS, Domain decomposition methods in sciences and engineering (Chiba, 1999), DDM.org, Augsburg, 1527 (electronic), 2001.

[12] M.Garbey, ACCELERATION OF THE SCHWARZ METHOD FOR ELLIPTIC PROBLEM, SIAM J. of Scientific Computing, Vol 26, No 6, pp 1871-1893, 2005.

[13] M.Garbey, B.Hadri and W.Shyy, FAST ELLIPTIC SOLVER FOR INCOMPRESSIBLE NAVIER STOKES FLOW AND HEAT TRANSFER PROBLEMS ON THE GRID, , 43rd Aerospace Sciences Meeting and Exhibit Conference, Reno January 2005, AIAA-2005-1386, 2005.

[14] M. Garbey and D. Tromeur-Dervout, TWO LEVEL DOMAIN DECOMPOSITION FOR MULTICLUSTERS, Domain Decomposition in Sciences and Engineering, T. Chan & Al editors, published by DDM.org, pp325-339, 2001.

[15] M. Garbey and D. Tromeur-Dervout, AITKEN-SCHWARZ ALGORITHM ON CARTESIAN GRID, 13th Int. Conf. on Domain Decomposition Methods DD13, Domain Decomposition Methods in Science and Engineering, CIMNE, Barcelona, N.Debit et Al edt, pp53-66, 2002.

[16] M.Garbey and D. Tromeur Dervout, ON SOME AITKEN LIKE ACCELERATION OF THE SCHWARZ METHOD, Int. J. for Numerical Methods in Fluids. 40 (12), pp 1493-1513, 2002.

[17] G.Golub and C.F.Van Loan, MATRIX COMPUTATIONS, 2sd Edt Johns Hopkins University Press, 89.

[18] D. Gottlieb and Chi-Wang Shu, ON THE GIBBS PHENOMENON AND ITS RESOLUTION, SIAM review, Vol39, No 4, 644-668, 1997.

[19] P. Henrici, ELEMENTS OF NUMERICAL ANALYSIS, John Wiley & Sons Inc, New York-London-Sydney, 1964.

[20] C.Japhet, F.Nataf and F.Rogier, THE OPTIMIZED ORDER 2 METHOD. APPLICATION TO CONVECTION-DIFFUSION PROBLEMS, Future Generation Computer Systems FUTURE 18, 2001.

[21] D. E. Keyes, HOW SCALABLE IS DOMAIN DECOMPOSITION IN PRACTICE?, Proceedings of the 11th Intl. Conf. on Domain Decomposition Methods (C.-H. Lai, et al, eds.), pp. 286-297, 1998.

[22] P.L.Lions, ON THE SCHWARZ ALTERNATING METHOD I, First International Symposium on Domain Decomposition Methods for Partial Differential Equations, Roland Glowinski and Gene H. Golub and Gérard A. Meurant and Jacques Périaux edt. SIAM, Philadelphia, PA, pp 1-42, 1988.

[23] P.L.Lions, ON THE SCHWARZ ALTERNATING METHOD II, Second International Symposium on Domain Decomposition Methods for Partial Differential Equations, Tony Chan and Roland Glowinski and Jacques Périaux and Olof Widlund edt. SIAM, Philadelphia, PA, pp 47-70, 1989.

[24] V.Martin, AN OPTIMIZED SCHWARZ WAVEFORM RELAXATION METHOD FOR UNSTEADY CONVECTION-DIFFUSION EQUATION, Applied Num. Math., 52 (4), pp 401-428, 2005.

[25] F.Nataf and F.Rogier, FACTORIZATION OF THE CONVECTION-DIFFUSION OPERATOR AND A (POSSIBLY) NON OVERLAPPING SCHWARZ METHOD, M3AS, 5, No 1, pp 67-93, 1995.

[26] A. SAMARSKI AND V. ANDRÉEV, METHODES AUX DIFFÉRENCES POUR ÉQUATIONS ELLIPTIQUES, Éditions MIR, Moscow, 1978.

[27] H.A.Schwarz, GESAMMELTE MATHEMATISCHE ABHANDLUNGEN,2, 2nd ed.,, Springer Berlin, pp.133-143, 1980. *Vierteljahrsschrift der Naturforschenden Gesellschaf*, Zurich, Vol15, 1st ed. pp.272-286, 1870.

[28] B. Smith, P. Bjorstad and W.Gropp, DOMAIN DECOMPOSITION, PARALLEL MULTILEVEL METHODS FOR ELLIPTIC PARTIAL DIFFERENTIAL EQUATIONS, Cambridge University Press, 1996.

[29] J. Stoer and R. Burlish, INTRODUCTION TO NUMERICAL ANALYSIS, TAM 12 Springer, 1980.

[30] S.Wright, PARALLEL ALGORITHMS FOR BANDED LINEAR SYSTEMS, SIAM J. Sci.Stat.Comput. 12, pp 824-842, 1991.