16

# Combining Waveform Relaxation and Domain Decomposition

Sigitas Keras

## 1 Introduction

Several techniques with inherent parallelism are available for the solution of parabolic equations, and among the most successful are *Domain Decomposition* (DD) and *Waveform Relaxation* (WR) methods. The main goal of this paper is to demonstrate that it is possible to combine these techniques into a single algorithm, thus reducing the computational complexity and the time required to obtain the computational solution of parabolic equations.

Waveform relaxation was first suggested in the late 19th century by Picard and Lindelöf ([Pic93, Lin94]) and has been subjected to much recent interest as a practical method for the solution of stiff ODEs after the publication of the paper by Lelarasmee and coworkers [LRSV82]. Recent work in this field includes papers by Nevanlinna, Zennaro, Bjørhus and others (see [MN87a, Bjø95, BZ93, Lum92]). It can be described as an iterative method to solve an initial value problem

$$\frac{du}{dt} = f(u), \quad y(0) = y_0.$$

At each step we compute the solution of the equation

$$\frac{du}{dt}^{(n+1)} = \tilde{f}(u^{(n+1)}, u^{(n)}), \quad u^{(n+1)}(0) = y_0, \tag{1}$$

where the function $\tilde{f}$ satisfies the identity

$$\tilde{f}(v, v) = f(v).$$

In this paper we only consider the case where $f$ is a linear operator, which throughout will be denoted by $f(v) = -Av$, and $\tilde{f}(v, w) = -Pv + Qw$, where $P$ and $Q$ are linear operators and $A = P - Q$. In this case (1) lends itself to the following form:

$$\frac{du}{dt}^{(n+1)} + Pu^{(n+1)} = Qu^{(n)} + f, \quad u^{(n+1)}(0) = y_0. \tag{2}$$

*Ninth International Conference on Domain Decomposition Methods*
Editor Petter E. Bjørstad, Magne S. Espedal and David E. Keyes 

Solving (2) explicitly by integration of constants, $u^{(n+1)}$ can be formally written as

$$u^{(n+1)} = \mathcal{K}u^{(n)} + \phi,$$

where

$$\mathcal{K}u(t) = \int_0^t e^{(s-t)P} Qu(s)ds,$$
$$\phi(t) = e^{-tP}u_0 + \int_0^t e^{(s-t)P} f(s)ds,$$

and it follows from the Banach fixed point theorem that the method (2) converges for all $f$ and $u^0$ if and only if $\rho(\mathcal{K}) < 1$, where $\rho$ denotes the spectral radius of the operator.

An extensive theory for this iterative procedure and its discrete version has been developed in [MN87a] and [MN87b]. In particular, necessary and sufficient conditions for the convergence of this method have been established. The following result has been proved in [MN87b].

**Theorem 1.** Suppose that all the eigenvalues of $A$ and $P$ have positive real parts. Then the spectral radius of $\mathcal{K}$ can be represented by means of formula

$$\rho(\mathcal{K}) = \max_{\xi \in R} \rho((i\xi I + P)^{-1}Q). \tag{3}$$

This result can be applied to semidiscretized linear parabolic PDEs. The following theorem has been proved in [Ker95b].

**Theorem 2.** Consider the diffusion equation

$$u_t - \nabla(a(x)\nabla u(x)) = f, \quad (x,t) \in \Omega \times (0,\infty), \tag{4}$$
$$u(0,x) = u_0(x), \quad x \in \Omega, \tag{5}$$
$$u(t,x) = 0, \quad (x,t) \in \partial\Omega \times [0,\infty). \tag{6}$$

where $\Omega$ is a rectangular domain in $\mathbb{R}^d$. Let $A$ be a discretization of the elliptic operator $-\nabla(a(x)\nabla)$, $0 < a_- \leq a(x) \leq a_+ < \infty$ and $P$ be a discretization of the operator $-\nabla(b(x)\nabla)$ for some function $b$ such that $0 < b_- \leq b(x) \leq b_+ < \infty$ and let both $A$ and $P$ satisfy the following assumptions:

1. $A$ and $P$ are positive definite
2. $c\langle Pu, u\rangle < \langle Au, u\rangle < C\langle Pu, u\rangle$ for any vector $u \neq 0$, provided that $cb(x) < a(x) < Cb(x)$ for all $x \in \Omega$, where $c, C \in \mathbb{R}$

Then the method (2) converges if $\max\left|\frac{a(x)-b(x)}{b(x)}\right| < 1$ and $\rho(\mathcal{K}) < \max\left|\frac{a(x)-b(x)}{b(x)}\right|$.

In particular, when $b(x)$ is a constant, one can deduce that

$$\rho(\mathcal{K}) \leq \max\left|\frac{a(x) - C}{C}\right| \tag{7}$$

and

$$\min_{C \in \mathbb{R}^+} \rho(\mathcal{K}) \leq \frac{a_+ - a_-}{a_+ + a_-}. \tag{8}$$

In rectangular domains this choice of $b(x)$ makes the matrix $P$ Toeplitz or block Toeplitz, which allows a fast solution of the subproblem (2). One may ask what other choices can be made for $b(x)$. In this paper we consider a natural generalization in which $b(x)$ is a piecewise constant function. In order to exploit the structure of the matrix $P$, we propose to solve the WR equations using domain decomposition techniques.

## 2 Nonoverlapping Domain Decomposition

Consider the equation

$$Au = f, \tag{9}$$

where $u$ and $f$ are vectors defined on a grid in the domain $\Omega$. We assume that $\Omega$ is divided into two subdomains $\Omega_1$ and $\Omega_2$ separated by the boundary $B$. We subdivide vectors $u$ and $f$

$$u = \begin{pmatrix} u_1 \\ u_2 \\ u_B \end{pmatrix}, \quad f = \begin{pmatrix} f_1 \\ f_2 \\ f_B \end{pmatrix}, \tag{10}$$

where indices $1, 2$ and $B$ denote restrictions of the vectors to the domains $\Omega_1$, $\Omega_2$ and the boundary $B$ respectively. Similarly, we can write

$$A = \begin{pmatrix} A_{11} & A_{12} & A_{1B} \\ A_{21} & A_{22} & A_{2B} \\ A_{B1} & A_{B2} & A_{BB} \end{pmatrix},$$

where submatrices $A_{ij}$ satisfy the relationship $\sum_j A_{ij} u_j = f_i$. We assume that matrix $A$ is symmetric, $A = A^T$, and also that $A_{12} = A_{21}^T = O$, which means that there is no interaction between subdomains $\Omega_1$ and $\Omega_2$ other than through the boundary. These assumptions are satisfied by matrices arising from the discretization of elliptic operators which are considered in the present paper. In this case (9) can be written as a set of three independent equations

$$\begin{aligned} Su_B &= \tilde{f}_B, && (11) \\ A_{11}u_1 &= \tilde{f}_1, && (12) \\ A_{22}u_2 &= \tilde{f}_2, && (13) \end{aligned}$$

where

$$\begin{aligned} S &= A_{BB} - A_{B1}A_{11}^{-1}A_{1B} - A_{B2}A_{22}^{-1}A_{2B}, && (14) \\ \tilde{f}_B &= f_B - A_{B1}A_{11}^{-1}u_1 - A_{B2}A_{22}^{-1}u_2, && (15) \\ \tilde{f}_1 &= f_1 - A_{1B}u_B, && (16) \\ \tilde{f}_2 &= f_2 - A_{2B}u_B. && (17) \end{aligned}$$

Equation (11) is solved first, after which we can determine vectors $\tilde{f}_1$ and $\tilde{f}_2$ and solve (12) and (13). Note that (12) and (13) are independent of each other and can be solved simultaneously.

An important part of the algorithm is to be able to solve the equation for the vector $u_B$ efficiently. The matrix $S$ is typically dense and expensive to calculate explicitly. In practice the equation for $u_B$ is solved using the preconditioned conjugate gradient method (PCG), and the rate of convergence depends on the condition number of the preconditioned matrix $S$. We present here two examples of preconditioners for two-dimensional elliptic problems, for a more elaborate discussion refer to the review paper of Chan and Matthew [CM94].

In order to introduce the preconditioners, we make further assumptions about the structure of the grid on the boundary, which is denoted with a subscript $B$. Assume that the domain $\Omega$ is divided into rectangular subdomains so that the boundary $B$ consists of $k$ edges $E_i$, $i = 1, \ldots, k$ and vertices $V$, and the matrix $S$ can be decomposed as follows,

$$S = \begin{pmatrix} S_{E_1E_1} & \ldots & S_{E_1E_k} & S_{E_1V} \\ \vdots & \ddots & \vdots & \vdots \\ S_{E_kE_1} & \ldots & S_{E_kE_k} & S_{E_kV} \\ S_{VE_1} & \ldots & S_{VE_k} & S_{VV} \end{pmatrix}.$$

The first preconditioner is a *block Jacobi* preconditioner

$$M_1 = \begin{pmatrix} S_{E_1E_1} & 0 & \ldots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & S_{E_kE_k} & 0 \\ 0 & \ldots & 0 & S_{VV} \end{pmatrix}.$$

One may expect that the condition number of the preconditioned system, $M_1^{-1}S$, is dependent on the discretization size $h$ as well as on the size of the subdomains $H$. This is in fact the case as shown in [BPS86],

**Theorem 3.** There exists a constant $C$ independent of $H$ and $h$, such that

$$\mathrm{cond}(M_1^{-1}S) \le CH^{-2}(1 + \ln(H/h)).$$

The other preconditioner which was also introduced in [BPS86] can be written in the form

$$M_2 = \begin{pmatrix} S_{E_1E_1} & 0 & \ldots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & S_{E_kE_k} & 0 \\ 0 & \ldots & 0 & A_V \end{pmatrix}.$$

Here $A_V$ is a matrix resulting from the discretization of the problem only on vertices $V$. Global coupling between the vertices, introduced by the matrix $A_V$, substantially reduces the condition number of the matrix $S$. The following result has been proved in [BPS86].

**Theorem 4.** There exists a constant $C$ independent of $H$ and $h$ such that

$$\mathrm{cond}(M_2^{-1}S) \le C(1 + \ln^2(H/h)).$$

## 3 Domain Decomposition for WR Equations

Consider the waveform relaxation method with a splitting as described in Theorem 2, where the domain $\Omega$ is divided into subdomains $\Omega_i$ and function $b(x)$ is constant in each of the subdomains $\Omega_i$. Then it is easy to show that

$$\rho(\mathcal{K}) < \max_i \frac{a^i_{\max} - a^i_{\min}}{a^i_{\max} + a^i_{\min}}, \tag{18}$$

where

$$a^i_{\max} = \max_{x\in\Omega_i} a(x), \qquad a^i_{\min} = \min_{x\in\Omega_i} a(x).$$

Indeed, we choose

$$b(x) = \frac{a^i_{\max} + a^i_{\min}}{2}, \quad x \in \Omega_i; \tag{19}$$

then the estimate (18) follows from Theorem 2.

In this way, we obtain a WR method with an improved radius of convergence. There is, however, a price to pay. The main advantage of the waveform relaxation method is due to the fact that the matrix $P$ is easily invertible, so that (2) is easy to solve. For instance, in case of a constant function $b(x)$ we have obtained a block Toeplitz matrix $P$ which can be inverted very fast using FFT techniques. In case of a piecewise constant function $b(x)$ this structure is destroyed. To overcome this difficulty we propose to employ the domain decomposition method. There are several reasons why this can be a promising approach. Firstly, as we have already noted in the previous section, the main computational cost of domain decomposition consists of solving the equation in each subdomain. This can be implemented in parallel. Since $b(x)$ is constant in each subdomain, this results in solving equations with block Toeplitz matrices. Secondly, we employ the domain decomposition method in order to solve the equation for $(n+1)$st iteration of the WR method $u^{(n+1)}$. Since this is not the solution we are seeking but only an iteration, we need not to solve the domain decomposition equations exactly. In other words, we propose to use an inner-outer iteration scheme where inner iteration is performed using DD algorithm and outer iteration is performed using WR method.

We start by stating a result about the convergence of iterative methods for time dependent iterative schemes. Consider a linear system of ordinary differential equations

$$\frac{du}{dt} + Pu = f(t), \qquad u(0) = u_0,$$

where P is a symmetric positive definite matrix. We solve it on a finite time interval $[0, t^*]$ with the $\theta$ method

$$\frac{u_{n+1} - u_n}{\Delta t} + \theta P u_{n+1} + (1-\theta) P u_n = f_n, \qquad \theta \geq 1/2,$$

At each time step the resulting linear equation

$$u_{n+1} = \left(\frac{1}{\Delta t} I + \theta P\right)^{-1} \left[\left(\frac{1}{\Delta t} I - (1-\theta) P\right) u_n + f_n\right] \tag{20}$$

**Table 1** The performance of the standard parallel solver for the test problem with $\alpha = 0.5$ on large grids.

| Grid size | Number of nodes | Time (sec) |
|---|---|---|
| $64 \times 64 \times 50$ | 1 | 76.22 |
| $64 \times 64 \times 50$ | 2 | 41.61 |
| $64 \times 64 \times 50$ | 4 | 24.45 |
| $64 \times 64 \times 50$ | 8 | 15.63 |
| $64 \times 64 \times 50$ | 16 | 12.08 |
| $64 \times 64 \times 50$ | 32 | 11.44 |
| $128 \times 128 \times 50$ | 1 | 584.1 |
| $128 \times 128 \times 50$ | 2 | 300.1 |
| $128 \times 128 \times 50$ | 4 | 158.64 |
| $128 \times 128 \times 50$ | 8 | 88.83 |
| $128 \times 128 \times 50$ | 16 | 54.25 |
| $128 \times 128 \times 50$ | 32 | 39.83 |
| $128 \times 128 \times 50$ | 64 | 31.58 |

is solved using an iterative method with a linear rate of convergence $\rho$,

$$||u_n^{k+1} - u_n|| \leq \rho ||u_n^k - u_n||, \;\; k = 1, 2, \ldots,$$

where the superscript $k$ denotes the iteration number. We construct an approximate solution by applying $m$ iterations at each time step and using the new value $v_n = u_{n+1}^m$ in the right hand side of (20). Then the following theorem holds.

**Theorem 5.** If $m$ is large enough, then the error at time step $n$, $e_n = v_n - u_n$, satisfies the inequality

$$||e_n|| < C\rho^m$$

where the constant $C$ is independent of $m$ and $\rho$.

The above theorem can be applied to our problem of combining the waveform relaxation and domain decomposition methods together. Consider the equation

$$\frac{du}{dt} + Au = f,$$

which is solved using the waveform relaxation method (2) as described in Theorem 2, and at each iteration the resulting equation is solved using $m$ iterations of the domain decomposition method. Let $\Omega$ be divided into subdomains $\Omega_i$ and let $b(x)$ be defined as in (19).

**Theorem 6.** If $m$ is large enough, then the combined waveform relaxation – domain decomposition method converges.

The proofs of Theorems 5 and 6 are rather technical and are given in full detail in [Ker95a].

**Table 2** The performance of the waveform relaxation with Toeplitz splitting for the test problem with $\alpha = 0.5$ on large grids.

| Grid size | Number of nodes | Number of iterations | Time (sec) |
|---|---|---|---|
| $64 \times 64 \times 50$ | 2 | 10 | 40.92 |
| $64 \times 64 \times 50$ | 4 | 10 | 26.74 |
| $64 \times 64 \times 50$ | 8 | 10 | 20.35 |
| $64 \times 64 \times 50$ | 16 | 10 | 12.44 |
| $64 \times 64 \times 50$ | 32 | 10 | 8.09 |
| $128 \times 128 \times 50$ | 2 | 10 | 171.8 |
| $128 \times 128 \times 50$ | 4 | 10 | 111.5 |
| $128 \times 128 \times 50$ | 8 | 10 | 83.46 |
| $128 \times 128 \times 50$ | 16 | 10 | 49.44 |
| $128 \times 128 \times 50$ | 32 | 10 | 27.43 |

## 4 A Numerical Example

A new numerical method can only be justified if it performs comparably to or better than existing methods. In this section we present a numerical example. All the calculations were carried out on an Intel Paragon computer. Our test problem is a parabolic equation with variable coefficients,

$$\begin{aligned}\frac{\partial u}{\partial t} &= \frac{\partial}{\partial x}\left((1+\alpha \sin 4\pi x \sin 4\pi y)\frac{\partial u}{\partial x}\right) \\ &+ \frac{\partial}{\partial y}\left((1+\alpha \sin 4\pi x \sin 4\pi y)\frac{\partial u}{\partial y}\right), \qquad (21)\\ &\qquad (x,y,t) \in \Omega \times (0,1), \quad \Omega = (0,1)\times(0,1)\end{aligned}$$

$$u(x,y,t) = 0, \quad (x,y,t) \in \partial\Omega \times (0,1), \qquad (22)$$

$$u(x,y,0) = \sin \pi x \sin \pi y. \qquad (23)$$

We solve the above equation using three methods. The first method is a standard Crank–Nicolson scheme. Since this is an implicit scheme, the parallel solver of sparse linear systems is used. The second method is the WR method with Toeplitz splitting as described in [Ker95b]. In this case the matrix $P$ in (2) is block Toeplitz so that parallel FFT solvers are used. Finally, the third method is the combined waveform relaxation domain decomposition method. The parallelization is done by assigning subdomain problems to different processors as well as performing WR iterations on different processors. The preconditioner described in Theorem 3 is used for the solution of resulting Schur problems. While it is not asymptotically optimal, it is easy to implement and provided a good estimate of the method. A better convergence is expected if a better preconditioner is used.

The results of the computations are presented in Tables 1 through 3. They suggest

**Table 3** The performance of the combined waveform relaxation domain decomposition method (4 subdomains) for the test problem with $\alpha = 0.5$ on large grids.

| Grid size | Number of nodes | Number of iterations | Time (sec) |
|---|---|---|---|
| $64 \times 64 \times 50$ | 2 | 8 | 55.91 |
| $64 \times 64 \times 50$ | 4 | 8 | 35.87 |
| $64 \times 64 \times 50$ | 8 | 8 | 28.63 |
| $64 \times 64 \times 50$ | 16 | 8 | 23.71 |
| $64 \times 64 \times 50$ | 32 | 8 | 18.34 |
| $128 \times 128 \times 50$ | 2 | 8 | 202.1 |
| $128 \times 128 \times 50$ | 4 | 8 | 134.1 |
| $128 \times 128 \times 50$ | 8 | 8 | 91.23 |
| $128 \times 128 \times 50$ | 16 | 8 | 60.23 |
| $128 \times 128 \times 50$ | 32 | 8 | 38.1 |

that both WR and WRDD methods have better asymptotic properties compared to the standard scheme when the grid size of the space domain increases. In particular, for the grids of the given size, the WR method outperforms the standard solver and the WRDD method performs comparably to it. The number of the iterations required for the convergence is independent on the grid size and the WRDD method requires less iterations than the WR method. The other important feature is that DD can be used as an inner iteration method, so that only several iterations of the DD are needed. In our numerical example we have performed only 3 DD iterations which was sufficient for the convergence of the method.

## Acknowledgement

Numerical experiments described in this paper were carried out on Intel Paragon computer at the California Institute of Technology. The author also acknowledges financial support he has received from Magdalene College, Cambridge and EPSRC.

## REFERENCES

[Bjø95] Bjørhus M. (1995) A note on the convergence of discretized dynamic iteration. *BIT* 35: 291–296.

[BPS86] Bramble J. H., Pasciak J. E., and Shatz A. H. (1986) An iterative method for elliptic problems on regions partitioned into substructures. *Math. Comp.* 46: 361–369.

[BZ93] Bellen A. and Zennaro M. (1993) The use of Runge-Kutta formulae in waveform relaxation methods. *Appl. Numer. Math.* 11: 95–114.

[CM94] Chan T. F. and Mathew T. P. (1994) Domain decomposition algorithms. *Acta Numerica* 3: 61–143.

[Ker95a] Keras S. (1995) Combining domain decomposition and waveform relaxation. Technical Report NA6, DAMTP, University of Cambridge.

[Ker95b] Keras S. (1995) Waveform relaxation method with Toeplitz operator splitting. Technical Report NA4, DAMTP, University of Cambridge. Submitted to SIAM J. Numer. Anal.

[Lin94] Lindelöf E. (1894) Sur l'application des méthodes d'approximations successives á l'etude des intégrales réeles des équations différentielles ordinaires. *Journal de Mathématiques Pures et Appliquées* 10: 117–128.

[LRSV82] Lelarasmee E., Ruehli A. E., and Sangiovanni-Vincentelli A. L. (1982) The waveform relaxation method for time-domain analysis of large scale integrated circuits. *IEEE Trans. on CAD of IC and Syst.* 1: 131–145.

[Lum92] Lumsdaine A. (1992) *Theoretical and Practical Aspects of Parallel Numerical Algorithms for Initial Value Problems, with Applications.* PhD thesis, Massachusetts Institute of Technology.

[MN87a] Miekkala U. and Nevanlinna O. (1987) Convergence of dynamic iteration methods for initial value problems. *SIAM J. Sci. Stat. Comput.* 8(4): 459–482.

[MN87b] Miekkala U. and Nevanlinna O. (1987) Sets of convergence and stability regions. *BIT* 27: 554–584.

[Pic93] Picard E. (1893) Sur l'application des méthodes d'approximations successives á l'etude de certaines équations différentielles ordinaires. *Journal de Mathématiques Pures et Appliquées* 9: 217–271.