UNIVERSITY OF CAMBRIDGE

Numerical Analysis Reports

Combining Domain Decomposition With Waveform Relaxation

Sigitas Keras

DAMTP 1995/NA7
October 1995

Department of Applied Mathematics and Theoretical Physics
Silver Street
Cambridge CB3 9EW
England

Combining Domain Decomposition With Waveform Relaxation

Sigitas Keras

Abstract

In this paper we consider a waveform relaxation method applied to a semidiscretized parabolic equation of the form

$$\frac{du}{dt} + Au = f$$
$$u(0) = u_0$$

where the operator A is a space discretization of the elliptic part of a parabolic PDE. Waveform relaxation method is used in tandem with a domain decomposition method. We prove the convergence of the combined method, give estimates of the rate of the convergence and present the results of numerical experiments.

1 Introduction

There is no need to emphasise the importance of rendering numerical algorithms suitable for implementation on parallel computers. Several techniques with inherent parallelism are available for the solution of parabolic equations and among the most successful ones 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 computational time required to obtain the numerical solution of the equation.

DD method was developed as a *divide and conquer* technique for solving elliptic problems. The main ideas of the algorithm were suggested by Schwarz in 1869 [15], and recent interest has been motivated by work of Bramble, Dryja, Widlund, Kuznetsov and others (see [5], [7], [4] and also the review [6]). There are two approaches to DD methods. The first one is to divide the domain into several overlapping subdomains and then to solve the obtained equation iteratively, using the values on the overlapping parts as boundary values. This method is also known as a *Schwarz algorithm*. The other approach is to divide the region into nonoverlapping subdomains, to solve the equation (typically using an iterative procedure) on the boundaries of subdomains, and then to calculate the solution in their interior. This technique is known under the name of *Schur algorithm*.

It is evident that in both cases at least part of the computation is carried out on the subdomains and can be performed independently of each other. This explains the suitability of the algorithm for parallel computation: the process can be parallelised in space. A different class of techniques, which, however, also possess a similar feature, is

WR methods. Waveform relaxation, similar to the DD algorithm, was first suggested in a totally different setting in the late 19th century by Picard and Lindelöf ([14], [11]) 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 [10]. Recent work in this field include papers by Nevanlinna, Zennaro, Bjørhus and others (see [12], [2], [1]).

WR methods have been considered in combination with other methods. In particular, a lot of interest has been generated by *Multigrid Waveform Relaxation* (see [16]). It has also been considered together with DD methods for hyperbolic problems by Bjørhus [3].

The aim of this paper is to show that it is possible to combine WR and DD techniques into a single algorithm. The resulting method inherits parallelism of both methods while at the same time has an improved convergence rate compared to both methods separately. The paper is organised as follows. In the next section we present the background material about WR and DD methods which we use in Section 3 to prove the main results about the convergence of the method. Section 4 is devoted to the numerical examples which strongly support the theoretical results.

2 Background material

2.1 Domain Decomposition

In this section we present a formal introduction to Schur type DD methods. Consider the equation

$$Au = f \tag{2.1}$$

where u and f are vectors defined on a grid in the domain Ω. We assume that Ω is divided into two subdomains Ω_1 and Ω_2 separated by the boundary B. We subdivide vectors u and f

$$u = \begin{pmatrix} u_1 \\ u_2 \\ u_B \end{pmatrix}, \quad f = \begin{pmatrix} f_1 \\ f_2 \\ f_B \end{pmatrix}, \tag{2.2}$$

where indices $1, 2$ and B denote restrictions of the vectors to the domains Ω_1, Ω_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 Ω_1 and Ω_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 equation (2.1) can be written as a set of three independent equations

$$Su_B = \tilde{f}_B, \tag{2.3}$$

$$A_{11}u_1 = \tilde{f}_1, \tag{2.4}$$

$$A_{22}u_2 = \tilde{f}_2, \tag{2.5}$$

where

$$
\begin{aligned}
S &= A_{BB} - A_{B1}A_{11}^{-1}A_{1B} - A_{B2}A_{22}^{-1}A_{2B}, && (2.6)\\
\tilde{f}_B &= f_B - A_{B1}A_{11}^{-1}u_1 - A_{B2}A_{22}^{-1}u_2, && (2.7)\\
\tilde{f}_1 &= f_1 - A_{1B}u_B, && (2.8)\\
\tilde{f}_2 &= f_2 - A_{2B}u_B. && (2.9)
\end{aligned}
$$

The equation (2.3) is solved first, after which we can determine vectors $\tilde{f}_1$ and $\tilde{f}_2$ and solve the equations (2.4) and (2.5). Note that equations (2.4) and (2.5) are independent of each other and can be solved simultaneously.

The equations above can easily be generalised to the case of l subdomains. We subdivide vectors into $l+1$ subvectors corresponding to the l subdomains and the boundary. Again, if we assume that matrix A is symmetric and $A_{ij} = A_{ji}^T = 0,\ i,j = 1,\ldots,l$, we can rewrite the initial equation as $l+1$ equations for the subvectors u_i inside of the subdomains and on the boundary,

$$
\begin{aligned}
Su_B &= \tilde{f}_B,\\
A_{ii}u_i &= \tilde{f}_i, \qquad i = 1,\ldots,l
\end{aligned}
$$

where S satisfies

$$
S = A_{BB} - \sum_{i=1}^{N} A_{Bi}A_{ii}^{-1}A_{iB} \qquad (2.10)
$$

and subvectors $\tilde{f}_B$, $\tilde{f}_i$ are calculated as follows,

$$
\begin{aligned}
\tilde{f}_B &= f_B - \sum_{i=1}^{n} A_{Bi}A_{ii}^{-1}u_i\\
\tilde{f}_i &= f_i - A_{iB}u_B, \qquad i = 1,\ldots,l
\end{aligned}
$$

As in the case of two subdomains, after solving the equation (2.10), vectors $u_i,\ i = 1,\ldots,l$) can be found simultaneously and independently of each other.

The most expensive part of the algorithm is solving the equation for the vector u_B. The matrix S is typically dense and expensive to calculate explicitly. In practise the equation for u_B is solved using 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 we refer to the review paper of Chan and Mathew [6].

In order to introduce the preconditioners, we make further assumptions about the structure of the grid on the boundary which we denote with a subscript B. We assume that it consists of k edges $E_i,\ i = 1,\ldots,k$ and vertices V (Fig. 1), so that matrix S can be decomposed as follows,

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

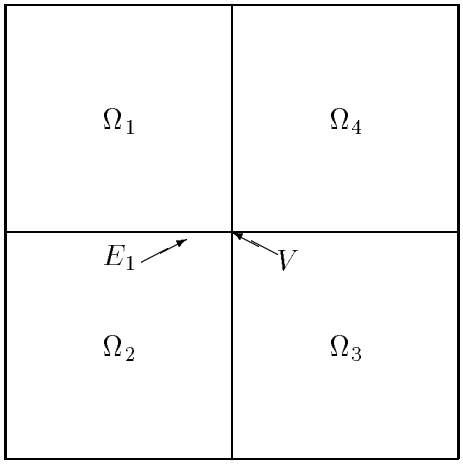

Figure 1: The domain Ω and its decomposition

The first preconditioner is a *block Jacobi* preconditioner

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

One may expect that the condition number of this preconditioner is dependant on the discretization step h as well as on the size of the subdomains H. The following result is due to Bramble *et al* [5],

Theorem 2.1 There exists a constant C independent of H and h, such that

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

The other preconditioner which was also introduced in [5] can be written as follows

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

where A_V is a matrix resulting from the discretization of the problem only on vertices V. Global coupling between the vertices, introduced by a matrix A_V, substantially reduces the condition number of the matrix S. The following result was proved in [5].

Theorem 2.2 There exists a constant C independent of H, h such that

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

Although the theory of DD methods was developed mainly to solve elliptic problems, it is easy to apply the method to the parabolic PDEs. In this case the equation is discretized in time so that at each time step one has to solve an elliptic problem. Since the resulting equations usually are better conditioned than typical elliptic problems, DD methods in this case converge very fast.

2.2 Waveform relaxation

Waveform relaxation is 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 equation

$$\frac{du}{dt}^{n+1} = \tilde{f}(u^{n+1}, u^n), \quad u^{n+1}(0) = y_0, \tag{2.11}$$

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 the paper 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 the equation (2.11) lends itself to the following form

$$\frac{du}{dt}^{n+1} + Pu^{n+1} = Qu^n, \quad u^{n+1}(0) = y_0, \tag{2.12}$$

Solving (2.12) 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 (2.3),(2.4) converges for all f and u^0 if and only if $\rho(\mathcal{K}) < 1$, where ρ denotes the spectral radius of the operator.

An extensive theory for this iterative procedure and its discrete version has been developed in [12] and [13]. In particular, necessary and sufficient conditions for the convergence of this method have been established. The following result has been proved in [13].

Theorem 2.3 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

$$\rho(\mathcal{K}) = \max_{\xi \in R} \rho((i\xi I + P)^{-1} Q). \tag{2.13}$$

This result can be applied to semidiscretized linear parabolic PDEs. The following theorem has been proved in [9].

Theorem 2.4 Consider the diffusion equation

$$u_t - \nabla(a(x)\nabla u(x)) = f, \quad (x,t) \in \Omega \times (0,\infty), \tag{2.14}$$

$$u(0,x) = u_0(x), \quad x \in \Omega, \tag{2.15}$$

$$u(t,x) = 0, \quad (x,t) \in \partial\Omega \times [0,\infty). \tag{2.16}$$

where Ω 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 (2.12) converges if $\max\left|\frac{a(x)}{b(x)} - 1\right| < 1$ and $\rho(\mathcal{K}) < \max\left|\frac{a(x)}{b(x)} - 1\right|$.

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

$$\rho(\mathcal{K}) \leq \max\left|\frac{1}{C}a(x) - 1\right| \tag{2.17}$$

and

$$\min_{C\in\mathbb{R}^+} \rho(\mathcal{K}) \leq \frac{a_+ - a_-}{a_+ + a_-} \tag{2.18}$$

In rectangular domains this choice of $b(x)$ makes the matrix P Toeplitz or block Toeplitz, which allows a fast solution of the subproblem (2.12). One may ask what can be other choices of $b(x)$. In this paper we consider a natural generalisation where $b(x)$ 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. We present the algorithm in the next section.

3 The algorithm

Our method is a direct generalisation of the approach described by present author in [9]. It consists of solving the equation (2.14)–(2.16) using the WR method as described in the Theorem 2.4, but, instead of choosing $b(x)$ constant on the domain Ω, we divide the domain into subdomains Ω_i and chose $b(x)$ to be constant on each. One can easily prove the following.

Theorem 3.1 Consider the WR method with a splitting as described in Theorem 2.4, where the domain Ω is divided into subdomains Ω_i and function $b(x)$ is constant in each of the subdomains Ω_i. Then

$$\rho(\mathcal{K}) < \max_i \frac{a^i_{\max} - a^i_{\min}}{a^i_{\max} + a^i_{\min}}, \tag{3.1}$$

where

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

Proof. We choose

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

then the estimate (3.1) follows from the Theorem 2.4. □

One can easily deduce from the inequality 3.1 that piecewise constant coefficients allow noticeable improvement in the convergence rate. For instance, if $a(x)$ is a linear function,

$$a(x) = \alpha x + \beta, \ \alpha > 0, \beta > 0$$

and $\Omega = [0, 1]$, then defining

$$b(x) = \frac{2\alpha + \beta}{2}$$

we can achieve

$$\rho(\mathcal{K}) \leq \frac{\beta}{2\alpha + \beta}.$$

However, if we subdivide the interval $[0, 1]$ into n subintervals $\Omega_i = [(i-1)/n, i/n]$ and define

$$b(x) = \frac{(2i-1)\alpha + n\beta}{2n},$$

we obtain the drastically improved bound

$$\rho(\mathcal{K}) \leq \frac{\beta}{(2\alpha + \beta)n}.$$

There is, however, price to pay. The main advantage of the WR method is due to the fact that matrix the P is easily invertible, so that equation (2.12) 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 solved very fast using FFT techniques. In case of a piecewise constant function $b(x)$, this structure is destroyed. This is why we want to employ the DD method. There are several reasons why we believe in the promise of this approach. Firstly, as we noted in the previous section, the main computational cost of the DD algorithm consists of solving the equation in each subdomain. This can be implemented in parallel, and the case of $b(x)$ being constant in each subdomain results in solving equations with block Toeplitz matrices. Secondly, we employ the DD method in order to solve the equation for u^{n+1}. Since this is not the solution we are seeking but only an iterate, we need not to solve the DD equations exactly. In fact, as we will demonstrate, it is often enough to employ only one iteration of the DD method and to use it as a new iterate of the WR method. But first we establish some facts about iterative solutions of the differential equation.

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 θ method

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

and at each time step the resulting linear equation

$$u_{n+1} = \left(\frac{1}{\Delta t} I + \theta P\right)^{-1} \left(\left(\frac{1}{\Delta t} I - (1-\theta) P\right) u_n + f_n\right) \tag{3.2}$$

is solved using an iterative method with a linear rate of convergence ρ, i.e. after m iterations the following estimate holds

$$\begin{aligned} v_0 &= u_0, \\ v_{n+1}^0 &= v_n, \\ ||v_{n+1} - \bar{v}_{n+1}|| &< \rho^m ||v_{n+1}^0 - \bar{v}_{n+1}||, \end{aligned}$$

and we set

$$v_{n+1} = v_n^m.$$

where $\bar{v}_{n+1}$ is an exact solution of the equation

$$\bar{v}_{n+1} = \left(\frac{1}{\Delta t} I + \theta P\right)^{-1} \left(\left(\frac{1}{\Delta t} I - (1-\theta) P\right) v_n + f_n\right) \tag{3.3}$$

Theorem 3.2 If m is large enough, then the error $e_n = v_n - u_n$ satisfies the inequality

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

where the constant C is independent of m and ρ.

Proof. For simplicity we denote

$$T = \left(\frac{1}{\Delta t} I + \theta P\right)^{-1} \left(\frac{1}{\Delta t} I - (1-\theta) P\right),$$

and

$$z_n = v_n - \bar{v}_{n+1}.$$

We can deduce from the definition of e_n that

$$||e_{n+1}|| = ||v_{n+1} - u_{n+1}|| \leq ||v_{n+1} - \bar{v}_{n+1}|| + ||\bar{v}_{n+1} - u_{n+1}|| \leq \rho^m ||z_n|| + ||T e_n||, \tag{3.4}$$

Since

$$z_n = u_n - e_n + u_{n+1} - T e_n,$$

we derive the estimate

$$||z_n|| < ||u_n - u_{n+1}|| + ||e_n - T e_n||. \tag{3.5}$$

Substituting this into equation (3.4) we obtain an estimate on the norm of e_{n+1}.

$$||e_{n+1}|| < ||Te_n|| + \rho^m||(1-T)e_n|| + \rho^m||u_n - u_{n+1}||. \tag{3.6}$$

Since we assumed our matrix P to be symmetric and positive definite and $\theta < 1/2$, the spectral radius of the matrix T obeys the inequality $\rho(T) < 1$, hence, for sufficiently large m the term $||Te_n|| + \rho^m||(1-T)e_n||$ can bounded above by $||e_n||$, in which case we finally obtain

$$||e_{n+1}|| < ||e_n|| + \rho^m||u_n - u_{n+1}||. \tag{3.7}$$

Applying the inequality (3.7) recursively yields

$$||e_{n+1}|| < \rho^m \sum_{i=0}^{n} ||u_n - u_{n+1}||. \tag{3.8}$$

Since for fixed t^* the right hand side of the inequality is bounded, this completes the proof of the theorem. □

Next we apply the above theorem to our problem of combining the WR and DD methods together.

Theorem 3.3 Consider a parabolic equation (2.14)–(2.16) which is solved using the WR method (2.12) as described in the Theorem 2.4 and at each iteration the resulting equation is solved using m iterations of the DD method. Let Ω be divided into subdomains Ω_i and let $b(x)$ be defined as in Theorem 3.1. Then the error of the of the solution satisfies the inequality

$$||e_n|| \leq \rho^n||e_0|| \tag{3.9}$$

where ρ is defined as follows

$$\rho = C(1+\rho_1)\rho_2^m + \rho_1,$$

C is a constant independent of ρ_1 and ρ_2, and ρ_1 and ρ_2 are convergence rates for the WR and DD methods respectively

$$\rho_1 = \rho(\mathcal{K}) = \max_i \frac{a^i_{\max} - a^i_{\min}}{a^i_{\max} + a^i_{\min}},$$
$$\rho_2 = 1 - \frac{2}{1 + \sqrt{\mathrm{cond}(SM^{-1})}}.$$

Proof. First we will prove that the DD method satisfies the conditions of the Theorem 3.2. Recall that, as we noted in the previous section, a DD method in its practical implementation is essentially an iterative method, where the equation on the boundary is solved using the PCG method, and the solution inside subdomains is calculated using the solution on the boundary. As is well known (see, for instance, [8]), the error of the PCG method satisfies the following bound

$$||e^m_B||_S \leq 2\rho_2^m||e^0_B||_S \tag{3.10}$$

where $||\cdot||_S = \langle S\cdot,\cdot\rangle$ is an S-norm. The error inside of the subdomain Ω_i satisfies the equation

$$A_{ii}e_i^m = -A_{iB}e_B^m,$$

hence

$$||e_i^m|| \leq 2C_i\rho_2^m||e^0||_S$$

where each constant C_i depends only on matrices A_{ii} and A_{iB}. Using the fact that all the norms are equivalent in $\mathbb{R}^n$, this allows us to bound the global error by

$$||e^m|| \leq C_1\rho_2^m||e^0|| \tag{3.11}$$

Applying Theorem 3.2 we can deduce that the global error of the solution of the equation (2.12) can also be bounded by a multiple of ρ_2^m.

Next we estimate the global error $\epsilon^n = u^n - u$, where u^n was obtained by solving the equation (2.12) with a DD method. The bound of the error of the exact solution of the equation (2.12) satisfies an inequality

$$||u_{\text{exact}}^{n+1} - u|| \leq \rho_1||u^n - u|| = \rho_1||\epsilon^n||$$

The solution u^{n+1} obtained after m iterations of the DD method using u^n as a starting value satisfies the inequality (3.11) which can be yielded as follows

$$||u_{\text{exact}}^{n+1} - u^{n+1}|| \leq C\rho_2^m||u_{\text{exact}}^{n+1} - u^n|| \leq C\rho_2^m(1+\rho_1)||\epsilon^n||.$$

Combining two last inequalities results in the estimate

$$||\epsilon^{n+1}|| = ||u^{n+1} - u|| \leq (\rho_1 + C\rho_2^m(1+\rho_1))||\epsilon^n||$$

which proves the theorem. □

While the above bounds prove the convergence of the method, they give very little insight on the speed of convergence of the method. In the next section we present the results of numerical experiments which support our assumption that only few iterations of the DD iteration are needed in order to obtain the convergence of the algorithm.

4 Numerical examples

In this section we present the numerical examples of the solution of the test problem using our proposed algorithm. Our test problem was equation (2.14)–(2.16) solved in the square $\Omega = (0,1) \times (0,1)$. The coefficient $a(x,y)$ was chosen to be linear

$$a(x,y) = \alpha + x + y, \quad \alpha > 0$$

For the DD algorithm we divided Ω into 9 squares Ω_i of the size $(1/3) \times (1/3)$. The equation was discretized in space using the five point difference scheme, and in time using the backward Euler rule. In this case, according to the Theorem 2.1, the Toeplitz WR method with a constant function $b(x)$ has the rate of convergence

$$\rho_T = \frac{1}{\alpha + 1}$$

The graph of ρ_T is presented in Figure 2. The combined Toeplitz–DD method, where $b(x)$ is defined as piecewise constant in each subdomain Ω_i has the rate of convergence

$$\rho_1 = \frac{1}{3\alpha + 1}.$$

It is displayed for different values of α in Figure 3.

For the DD method we have used the preconditioner M_1. While this preconditioner has poor asymptotic properties when the size of the subdomains H tends to zero, it still provided a good rate of convergence. The spectral radius of the DD method was calculated experimentally and is depicted in Figures 4 and 5.

Numerical experiments show that it is enough to take two iterations of the DD method in order to ensure the convergence of the method. Figures 6 and 7 show the spectral radius of the combined Toeplitz–DD method with two and three iterations of the DD method. These calculation were carried out on a grid with 900 nodes ($m = n = 30$). Comparing this graph with the graph in Figure 2 one can see the improved rate of convergence.

5 Conclusions

In this paper we demonstrate that it is possible to combine the Toeplitz waveform relaxation method with domain decomposition techniques and that the resulting algorithm has a superior rate of convergence. Several issues were not addressed in this paper and we intend to return to them in the subsequent work. Among them, the question of choosing the size H of the subdomains into which we divide Ω. When H is decreasing, the spectral radius ρ_1 is decreasing, while ρ_2 is increasing, which means that some balancing between them is required. Another important question is the possibility of dividing Ω into subdomains of a different size. One approach, for instance, would be to chose the subdomains so that to minimise ρ_1. This, however, will affect the condition number and ρ_2. Again the strategy of balancing between those two spectral radii remains unclear.

6 Acknowledgments

The author would like to thank his supervisor Dr Arieh Iserles for many useful comments on this work. The author was supported by Leslie Wilson Scholarship from Magdalene College, Cambridge and ORS award.

References

[1] A Bellen and M Zennaro. The use of runge-kutta formulae in waveform relaxation methods. *Appl. Numer. Math.*, **11**:95–114, 1993.

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

[3] M Bjørhus. Semi-discrete subdomain iteration for hyperbolic systems. Technical report, The University of Trondheim, Norway, 1995.

[4] P E Bjørstad and O B Widlund. Iterative methods for the solution of elliptic problems on regions partitioned into substructures. *SIAM J. Numer. Anal.*, **23**:1093–1120, 1986.

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

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

[7] M Dryja. A finite element-capacitance method for elliptic problems on regions partitioned into subregions. *Numer. Math.*, **44**:153–168, 1984.

[8] G H Golub and C F van Loan. *Matrix Computation*. North Oxford Academic, 1986.

[9] S Keras. Waveform relaxation method with Toeplitz operator splitting. Technical Report NA4, DAMTP, University of Cambridge, 1995.

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

[11] E Lindelöf. 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, 1894.

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

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

[14] É Picard. 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, 1893.

[15] H A Schwartz. Über einige abbildungsaufgaben. *Ges. Math. Abh.*, **11**:65–83, 1869.

[16] S Vandewalle. *Parallel Multigrid Waveform Realxation for Parabolic Problems*. B. G. Teubner, Stuttgart, 1993.

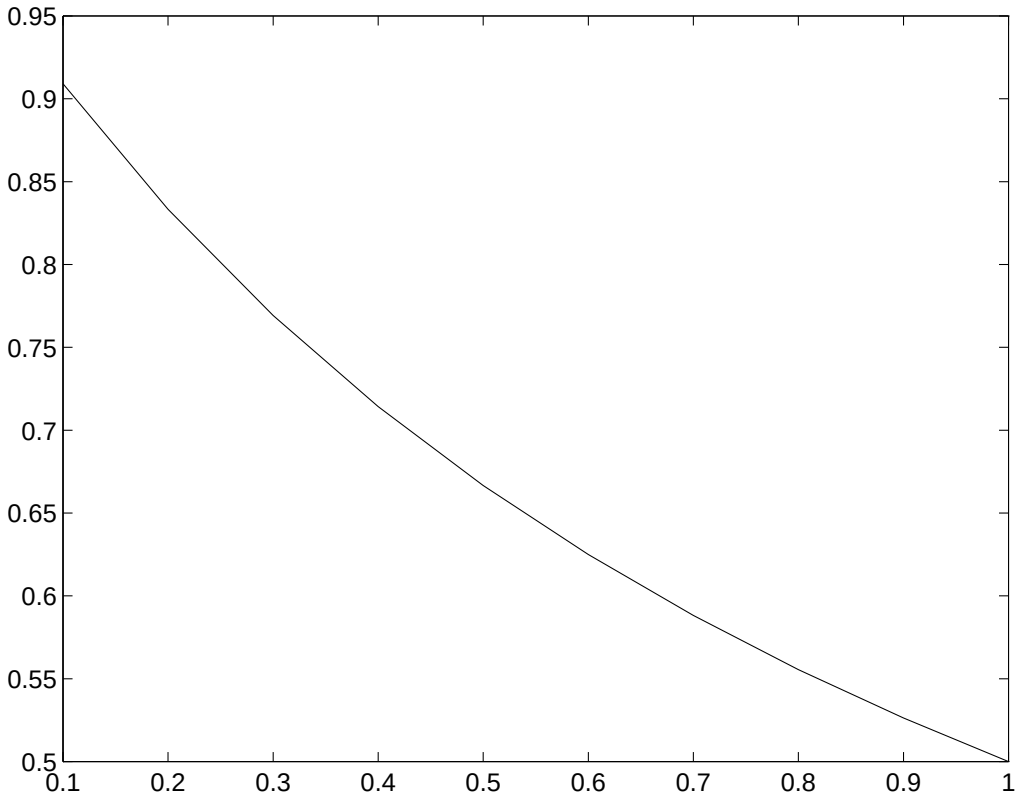

Figure 2: The plot of the ρ_T against α.

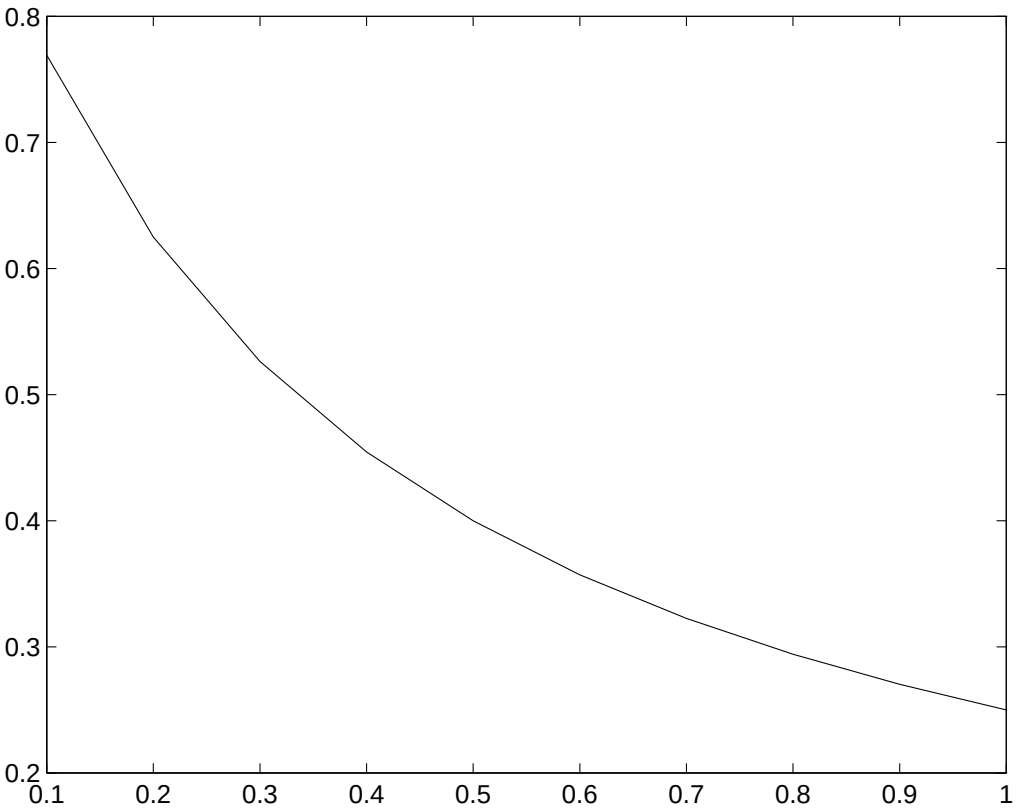

Figure 3: The plot of the ρ_2 against α.

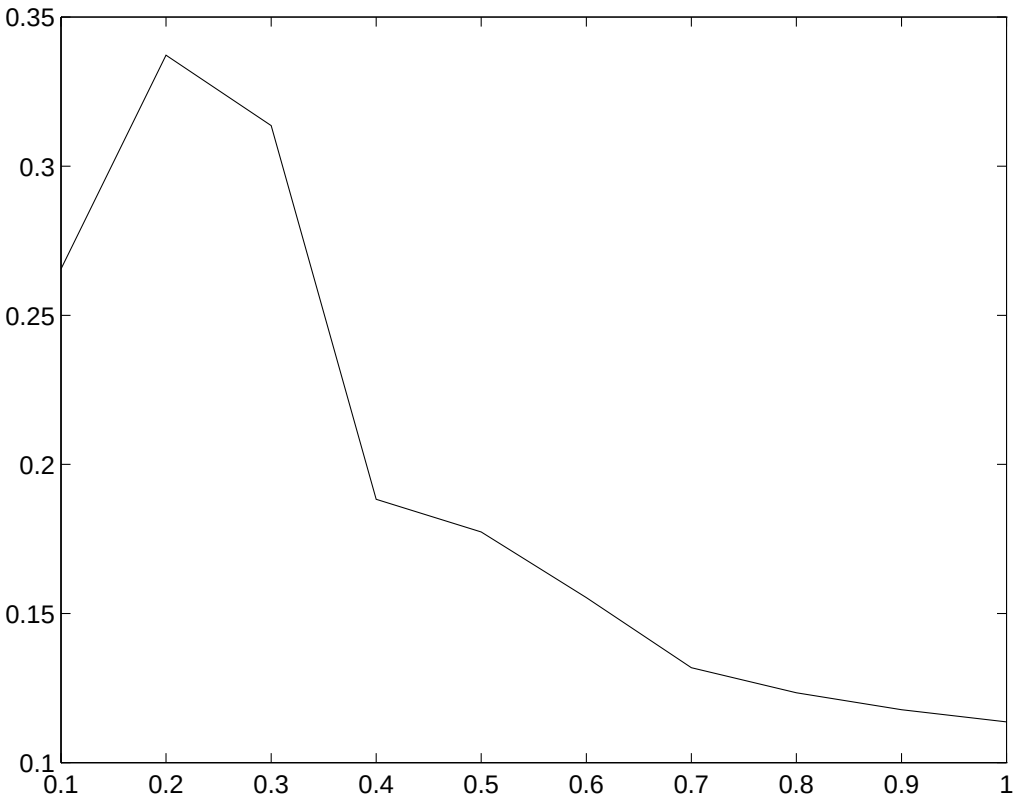

Figure 4: The plot of the ρ_2 against α on a grid of 900 nodes ($m = n = 30$)

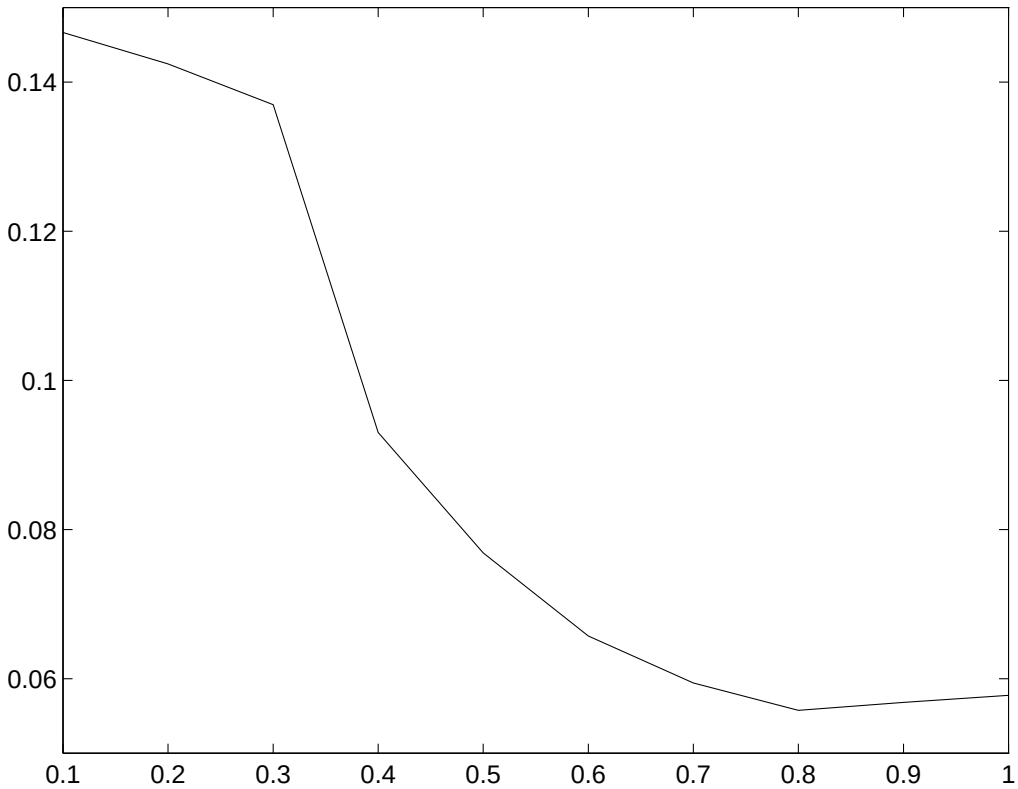

Figure 5: The plot of the ρ_2 against α on a grid with 144 nodes ($m = n = 12$)

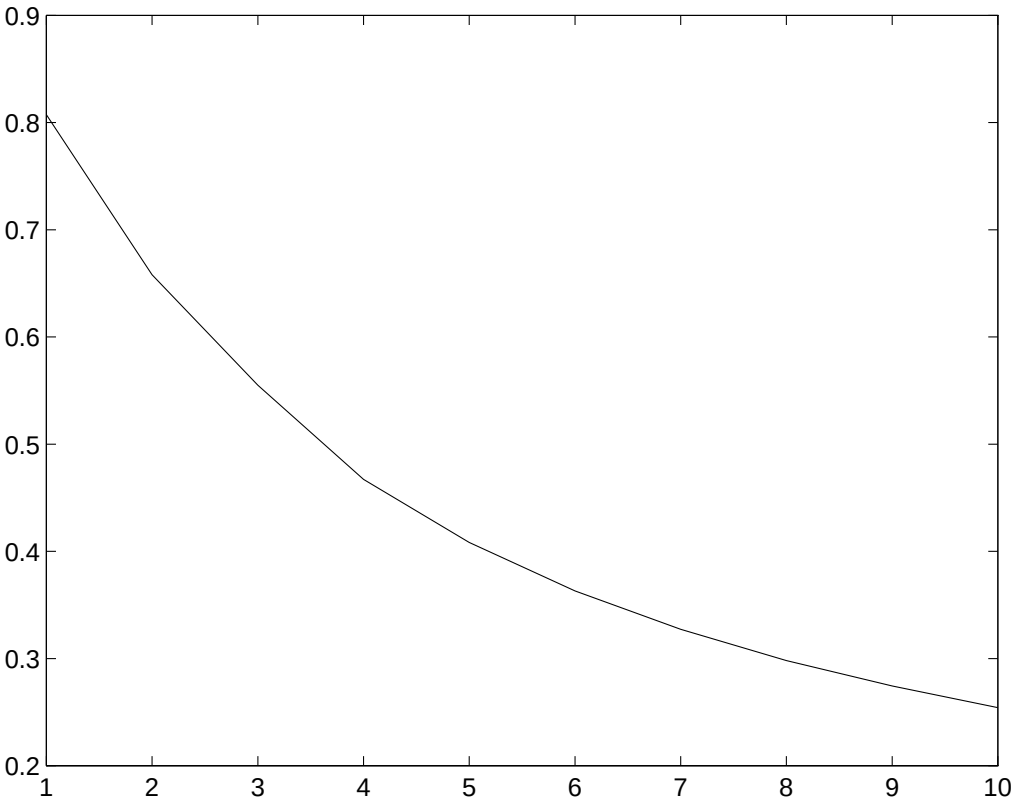

Figure 6: The plot of ρ against α on a grid of 900 nodes ($m = n = 30$) with 2 DD iterations.

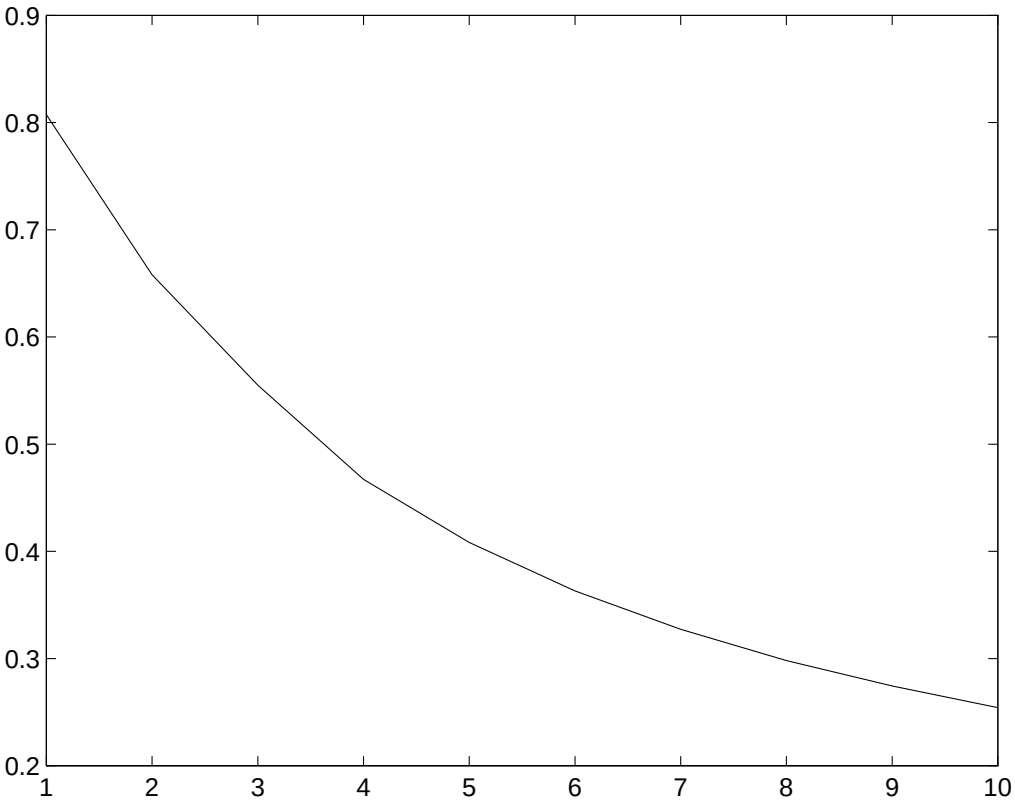

Figure 7: The plot of ρ against α on a grid of 900 nodes ($m = n = 30$) with 3 DD iterations.