East Asian Math. J.
Vol. 37 (2021), No. 3, pp. 295–305
http://dx.doi.org/10.7858/eamj.2021.020

EFFICIENCY ANALYSIS OF A DOMAIN DECOMPOSITION METHOD FOR THE TWO-DIMENSIONAL TELEGRAPH EQUATIONS

Younbae Jun

Abstract. In this paper, we analyze the efficiency of a domain decomposition method for the two-dimensional telegraph equations. We formulate the theoretical spectral radius of the iteration matrix generated by the domain decomposition method, because the rate of convergence of an iterative algorithm depends on the spectral radius of the iteration matrix. The theoretical spectral radius is confirmed by the experimental one using MATLAB. Speedup and operation ratio of the domain decomposition method are also compared as the two measurements of the efficiency of the method. Numerical results support the high efficiency of the domain decomposition method.

1. Introduction

The main interest of this paper is the two-dimensional telegraph equation given by

$$u_{tt} + 2\alpha u_t + \beta^2 u = u_{xx} + u_{yy} + f(x, y, t), (x, y, t) \in \Omega \times [0, T], \tag{1}$$

with the initial conditions

$$u(x, y, 0) = u_0(x, y), u_t(x, y, 0) = v_0(x, y), (x, y) \in \Omega, \tag{2}$$

and the boundary conditions

$$B[u(x, y, t)] = g(x, y, t), (x, y, t) \in \partial\Omega \times [0, T], \tag{3}$$

where Ω=[0,1]×[0,1], α and β are given constants, and B is the differential operator of the boundary condition.

The telegraph equation is a vital physical equation which is widely applied to the fields of electric signals, electromagnetic waves, fluid mechanics, and so on. The telegraph equations in the real-world problems usually include many complex data so that numerical methods are commonly used to the equations.

Received March 16, 2021; Accepted May 10, 2021.
2010 *Mathematics Subject Classification.* 65M06.
Key words and phrases. Domain decomposition, Finite difference method, Telegraph equation, Unconditional stability, Efficiency, Spectral radius.
This research was supported by Kumoh National Institute of Technology(2018-104-067).

(pISSN 1226-6973, eISSN 2287-2833)

In recent years, many numerical methods have been developed for solving telegraph equations. A Haar wavelet collocation approach has been proposed by Asif *et al.* [2], Aslefallah and Rostamy [3] researched an application of the singular boundary method, Dehghan and Salehi [5] worked on a method based on meshless approach, Jiwari *et al.* [6] developed a differential quadrature algorithm, Ma *et al.* [8] introduced a meshless collocation approach with barycentric rational interpolation, Mittal and Bhatia [10] studied a modified B-spline differential quadrature method. Most of numerical methods for solving Equation (1) focused on the whole domain in which the telegraph equation is defined.

Recently, instead of working on one whole domain, the domain decomposition (DD) technique is getting used to the areas involving the partial differential equations. The DD method is working on the decomposed subdomains. One of the main reasons using a DD method is its excellent efficiency. Jun [7] proposed a second-order implicit prediction domain decomposition (SIPDD) method for the telegraph equations and reported its excellent accuracy and powerful efficiency in the paper. However, in [7], the efficiency has been only illustrated by numerical experiments without any theoretical supports, even though its accuracy and unconditional stability have been analyzed.

In this paper, we analyze the algorithm of the SIPDD method [7] and provide several mathematical reasons of the excellent efficiency in terms of the spectral radii of the matrices generated by the SIPDD method. The outline of this paper is as follows. In Section 2, we present existing schemes of the finite difference methods for solving the two-dimensional telegraph equation (1). In Section 3, theorems and proofs for the efficiency of the domain decomposition method are provided. Speedup and operation ratio are compared in Section 4. Lastly, we make concluding remarks in Section 5.

2. Existing schemes

In this section, we review three existing finite difference schemes for solving the two-dimensional telegraph equations in order to compare each other in terms of stability, accuracy, and efficiency. Those schemes are the fully explicit scheme (FES), the fully implicit scheme (FIS), and a domain decomposition. These schemes are based on the finite difference operators which are useful to discretize the partial differential equation and the domain of the equation. For the rest of the paper, the positive integers L, M, and N are chosen so that $\Delta x = \frac{1}{L}$, $\Delta y = \frac{1}{M}$, and $\Delta t = \frac{T}{N}$. Let $x_i = i\Delta x$, $y_j = j\Delta y$, and $t_n = n\Delta t$, where $i = 0, \cdots, L, j = 0, \cdots, M$, and $n = 0, \cdots, N$. Let u_{ij}^n be the exact solution $u(x_i, y_j, t_n)$ and w_{ij}^n be the approximated solution at the grid point (x_i, y_j, t_n). We denote $f(x_i, y_j, t_n)$ by f_{ij}^n. Then, the central finite difference operators for the time level $t = t_n$ at the point (x_i, y_j, t_n) are defined by

$$w_{tt}^n = \frac{w_{i,j}^{n+1} - 2w_{ij}^n + w_{i,j}^{n-1}}{(\Delta t)^2}, w_t^n = \frac{w_{ij}^{n+1} - w_{ij}^{n-1}}{2\Delta t},$$

$$w_{xx}^n = \frac{w_{i+1,j}^n - 2w_{ij}^n + w_{i-1,j}^n}{(\Delta x)^2}, w_{yy}^n = \frac{w_{i,j+1}^n - 2w_{ij}^n + w_{i,j-1}^n}{(\Delta y)^2}.$$

Now, we describe the detailed scheme of each of the classical three-level schemes for solving the two-dimensional telegraph equation (1).

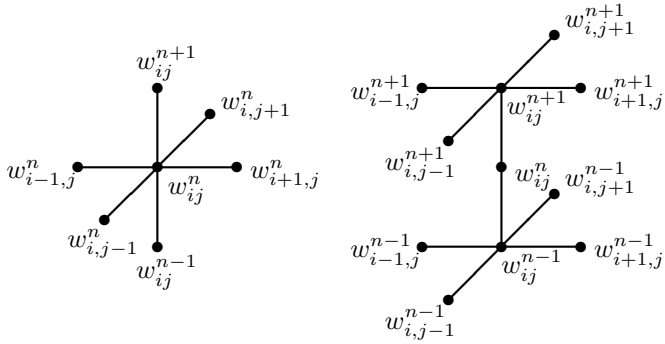
2.1. Fully explicit scheme (FES) and Fully implicit scheme (FIS)

The classical fully explicit three-level finite difference scheme and fully implicit scheme [1] for solving the two-dimensional telegraph equation (1) can be written as follows:

$$\text{FES: } w_{tt}^n + 2\alpha w_t^n + \beta^2 w_{ij}^n = w_{xx}^n + w_{yy}^n + f_{ij}^n, \tag{4}$$

and

$$\text{FIS: } w_{tt}^n + 2\alpha w_t^n + \beta^2 w_{ij}^n = \frac{1}{2}\left(w_{xx}^{n+1} + w_{xx}^{n-1}\right) + \frac{1}{2}\left(w_{yy}^{n+1} + w_{yy}^{n-1}\right) + f_{ij}^n. \tag{5}$$



(a) Fully explicit scheme (b) Fully implicit scheme

Figure 1. Stencils of three-level schemes

It is well-known [1] that the FES is easy to understand but conditionally stable for $\lambda = \left(\frac{\Delta t}{\Delta x}\right)^2 + \left(\frac{\Delta t}{\Delta y}\right)^2 \leq 1$ and that the FIS is unconditionally stable but not efficient. Stencils of those schemes are provided in Figure 1 as a reference. We will see later in Table 1, FES has significant stability problem, however FIS is unconditionally stable.

2.2. Domain decomposition method : Second-order implicit prediction scheme

In this section, we investigate the algorithm of a particular domain decomposition method for solving the equation (1) which is referred to as the second-order implicit prediction domain decomposition (SIPDD) method [7].

Domain decomposition (DD) method is often used to solve the partial differential equations (PDEs), because the DD method is very efficient especially when a parallel computer is used. The basic idea of the domain decomposition is that the original domain is decomposed into two or more subdomains and the

PDE in each subdomain is solved in parallel manner. The way of decomposition depends on how we decompose the whole domain. For examples, overlapping or non-overlapping decomposition and stripwise or rectangular decomposition are possible.

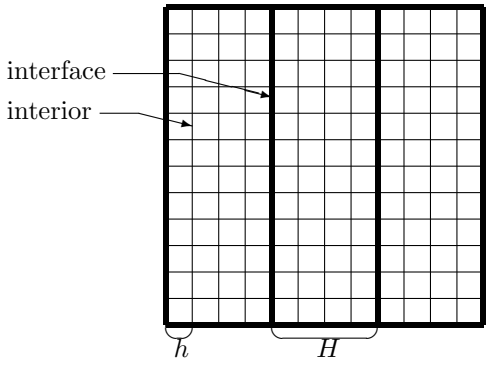


Figure 2. Non-overlapping vertical stripwise decomposition

The SIPDD [7] method uses a non-overlapping stripwise domain decomposition shown in Figure 2, in which the adjacent subdomains share an interface line. In order to solve the sub-problems on each subdomains concurrently in a parallel computing environment, the values at the grid points on the interface line have to be estimated in advance. Once the values on the interface lines are predicted, we solve the PDE on the interior region of each subdomain using the values at the interface lines. Suppose the whole domain is decomposed into P subdomains and let $H = 1/P$. The SIPDD algorithm consists of two steps: the interface prediction step and the interior region solver step, and then these two steps are repeated until the last time level.

2.2.1. *Step 1:* Interface prediction
In this step, the values at the grid points on the interface lines are estimated using the central finite difference operator $\hat{w}^n_{xx}$ on the interface lines which is defined by

$$\hat{w}^n_{xx} = \frac{w^n_{i+LH,j} - 2w^n_{ij} + w^n_{i-LH,j}}{H^2}, \tag{6}$$

where $w^n_{i+LH,j}$ and $w^n_{i-LH,j}$ are the unknown values on the adjacent interface lines. This implicit interface prediction scheme can be written as the following:

$$w^n_{tt} + 2\alpha w^n_t + \beta^2 w^n_{ij} = \frac{1}{2}\left(\hat{w}^{n+1}_{xx} + \hat{w}^{n-1}_{xx}\right) + \frac{1}{2}\left(w^{n+1}_{yy} + w^{n-1}_{yy}\right) + f^n_{ij},$$

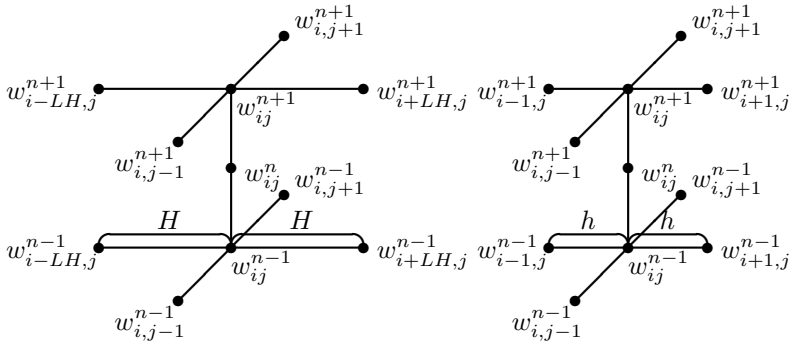
where $\hat{w}^{n+1}_{xx}$ and $\hat{w}^{n-1}_{xx}$ are defined in Equation (6)

2.2.2. *Step 2:* Interior region solver

This step, which seems to be the same as the fully implicit scheme, is the main and most time consuming step in the SIPDD method. The only difference is the number of unknowns of the initial boundary value problem. Since the whole domain is divided into the smaller subdomains, the total number of unknowns in the SIPDD scheme is much smaller than that in the fully implicit scheme. Therefore, the finite difference scheme itself has to be the same as in the FIS. This implicit interior region solver scheme is the following:

$$w_{tt}^n + 2\alpha w_t^n + \beta^2 w_{ij}^n = \frac{1}{2}\left(w_{xx}^{n+1} + w_{xx}^{n-1}\right) + \frac{1}{2}\left(w_{yy}^{n+1} + w_{yy}^{n-1}\right) + f_{ij}^n.$$

Stencils of these two steps of SIPDD scheme are provided in Figure 3.



(a) Interface Prediction (b) Interior Region Solver

Figure 3. Stencils of the SIPDD scheme

2.2.3. *Summary.* As mentioned in [7], the performance of the SIPDD method is outstanding. Not only the method is unconditionally stable, but also it is as accurate as the FIS method. These facts are reported in Table 1. Let us denote $||w^N - u^N||_\infty$ by the maximum error between the approximated solution and the exact solution. Table 1 shows the maximum error of the telegraph equation $u_{tt} = u_{xx} + u_{yy}$ at the final time level $t = 1$ with the various λ ranging from 8 to 5000 of the three methods : FES, FIS, and SIPDD with 5 subdomains. In Table 1, we see that FES is not convergent for those λ, on the other hand, FIS and SIPDD(5) are both unconditionally stable. We note that SIPDD(5) is as accurate as FIS. Furthermore, the SIPDD method is very efficient, which will be analyzed in the next section.

Table 1. Maximum error at the various λ

$\Delta x(=\Delta y)$	Δt	λ	FES	FIS	SIPDD(5)
1/100	1/2	5000	∞	0.5862e-2	0.5827e-2
1/100	1/5	800	∞	0.1531e-2	0.1720e-2
1/100	1/10	200	∞	0.1472e-3	0.2934e-3
1/100	1/50	8	∞	0.8787e-4	0.2970e-4

3. Theorems for the efficiency of the domain decomposition method

In this section, we report some mathematical reasons for the efficiency of the second-order implicit prediction domain decomposition method [7]. The SIPDD method generates very large and sparse linear systems of the form $Au = b$. Iterative methods [11] are commonly used to solve such large and sparse linear systems. Some of iterative methods are Gauss-Seidel, SOR, SSOR, Incomplete Cholesky (IC), or Modified IC (MIC) with acceleration procedure [11]. In this paper, we use the Gauss-Seidel iterative method for the sake of simplicity of analysis. Using the GS iteration, we formulate the spectral radii of the iteration matrices of the SIPDD method and its efficiency.

It is well-known [4, 11] that the rate of convergence of an iterative algorithm depends on the spectral radius of the iteration matrix. The smaller spectral radius leads to faster convergence in the iterative method. Consequentially, we hope to show that the spectral radius of the SIPDD method is smaller than the one of the FIS method. Suppose the whole domain is decomposed into P subdomains. For simplicity of analysis, we consider the telegraph equation of the form $u_{tt} = u_{xx} + u_{yy}$ and let $h = \Delta x = \Delta y$, $H = 1/P, r = \Delta t/h, \delta = h/H$. Then, the spectral radii of the matrices generated by the GS iteration of each step of the SIPDD algorithm to solve $u_{tt} = u_{xx} + u_{yy}$ are formulated by the following theorems.

Theorem 3.1. *(Interior) Suppose the whole domain is decomposed into P subdomains. Let G_P be the matrix generated by the Gauss-Seidel iteration of the interior scheme of the SIPDD method. Then the spectral radius of G_P is*

$$\rho(G_P) = \left[\frac{r^2}{1+2r^2} \cdot (\cos P\pi h + \cos \pi h)\right]^2. \tag{7}$$

Proof. Let A_P be the coefficient matrix of the five-diagonal linear system which is generated from the interior region solver scheme of the SIPDD. Then it can be written as

$$A_P = \left(1 + \left(\frac{\Delta t}{\Delta x}\right)^2 + \left(\frac{\Delta t}{\Delta y}\right)^2\right) I_K - 4 \cdot \left(\frac{1}{2}r^2\right) R_K = (1+2r^2)I_K - 2r^2 R_K,$$

where I_K is the identity matrix of order K and the matrix R_K is a block tri-diagonal matrix such as

$$R_K = \frac{1}{4}\begin{bmatrix} S & I & O & \cdots & O \\ I & S & I & \ddots & \vdots \\ O & \ddots & \ddots & \ddots & O \\ \vdots & \ddots & I & S & I \\ O & \cdots & O & I & S \end{bmatrix} \text{ and } S = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 1 & 0 & 1 & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & 1 & 0 & 1 \\ 0 & \cdots & 0 & 1 & 0 \end{bmatrix}.$$

The order K of the square matrix R_K of the interior scheme is $(\frac{L}{P} - 1) \cdot (M - 1)$. It is easy to see that the spectral radius $\rho(R_K)$ of the interior scheme for $K =$

$(\frac{L}{P}-1)\cdot(M-1)$ is $\rho(R_K) = \frac{1}{2}\left(\cos\frac{\pi}{L/P}+\cos\frac{\pi}{M}\right) = \frac{1}{2}(\cos P\pi h+\cos\pi h)$. Then, the Jacobi iteration matrix $G_{P,J}$ of the interior scheme is

$$G_{P,J} = I_K - \frac{1}{1+2r^2}\{(1+2r^2)I_K - 2r^2R_K\} = \frac{2r^2}{1+2r^2}R_K.$$

Hence, the spectral radius of $G_{P,J}$ is

$$\rho(G_{P,J}) = \frac{2r^2}{1+2r^2}\rho(R_K) = \frac{2r^2}{1+2r^2}\cdot\frac{1}{2}(\cos P\pi h+\cos\pi h).$$

It is well known [11] that the spectral radius of the GS iteration matrix is the square of the one of the Jacobi iteration matrix. Thus, the spectral radius of the GS iteration matrix of the interior scheme of the SIPDD method is

$$\rho(G_P) = \rho(G_{P,J})^2 = \left[\frac{r^2}{1+2r^2}\cdot(\cos P\pi h+\cos\pi h)\right]^2.$$

□

Theorem 3.2. *(Prediction) Suppose the whole domain is decomposed into P subdomains and $H = 1/P$. Let G_H be the matrix generated by the Gauss-Seidel iteration of the prediction scheme of the SIPDD method. Then the spectral radius of G_H is*

$$\rho(G_H) = \left[\frac{r^2}{1+\delta^2r^2+r^2}\cdot(\delta^2\cos\pi H+\cos\pi h)\right]^2. \tag{8}$$

Proof. With the same argument in Theorem 3.1, the coefficient matrix A_H with the five-diagonal linear system of the interface prediction scheme can be written as

$$A_H = \left(1+\left(\frac{\Delta t}{H}\right)^2+\left(\frac{\Delta t}{\Delta y}\right)^2\right)I_{\hat{K}} - 4\cdot\left(\frac{1}{2}r^2\right)R_{\hat{K}} = (1+\delta^2\cdot r^2)I_{\hat{K}} - 2r^2R_{\hat{K}},$$

where

$$R_{\hat{K}} = \frac{1}{4}\begin{bmatrix} \hat{S} & I & O & \cdots & O \\ I & \hat{S} & I & \ddots & \vdots \\ O & \ddots & \ddots & \ddots & O \\ \vdots & \ddots & I & \hat{S} & I \\ O & \cdots & O & I & \hat{S} \end{bmatrix} \text{ and } \hat{S} = \delta^2\begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 1 & 0 & 1 & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & 1 & 0 & 1 \\ 0 & \cdots & 0 & 1 & 0 \end{bmatrix}.$$

The order $\hat{K}$ of the square matrix $R_{\hat{K}}$ of the interface prediction scheme is $(P-1)\cdot(M-1)$. It can be easily obtained that the spectral radius $\rho(R_{\hat{K}})$ of the interface prediction scheme is $\rho(R_{\hat{K}}) = \frac{1}{2}\left(\delta^2\cos\frac{\pi}{P}+\cos\frac{\pi}{M}\right) = \frac{1}{2}\left(\delta^2\cos\pi H+\cos\pi h\right)$. Then, the Jacobi iteration matrix $G_{H,J}$ of the interface prediction scheme is

$$G_{H,J} = I_{\hat{K}} - \frac{1}{1+\delta^2r^2+r^2}\{(1+\delta^2r^2+r^2)I_{\hat{K}} - 2r^2R_{\hat{K}}\} = \frac{2r^2}{1+\delta^2r^2+r^2}R_{\hat{K}}.$$

and

$$\rho(G_{H,J}) = \frac{2r^2}{1+\delta^2 r^2 + r^2} \cdot \frac{1}{2}(\delta^2 \cos \pi H + \cos \pi h).$$

Thus, the spectral radius of the GS iteration matrix of the SIPDD interface scheme is

$$\rho(G_H) = \rho(G_{H,J})^2 = \left[\frac{r^2}{1+\delta^2 r^2 + r^2} \cdot (\delta^2 \cos \pi H + \cos \pi h) \right]^2.$$

□

Corollary 3.3. *Let G_{FIS} be the matrix generated by the Gauss-Seidel iteration of the fully implicit scheme (FIS). Then the spectral radius of G_{FIS} is*

$$\rho(G_{FIS}) = \left[\frac{2r^2}{1+2r^2} \cdot \cos \pi h \right]^2. \tag{9}$$

Proof. The result is immediately obtained by Theorem 3.1. □

Using the equations 3.1 through 3.3, Table 2 shows the theoretical spectral radii of the GS iteration matrices of the interface prediction scheme and the interior region solver scheme, respectively, of the SIPDD method at the various number of subdomains P for $u_{tt} = u_{xx} + u_{yy}$ with $\Delta x = \Delta y = 0.01$ and $\Delta t = 0.02$. In Table 2, we see that the spectral radius of the FIS method is 0.7893, however, the spectral radius of the interior scheme of the SIPDD method decreases significantly as P increases. This fact supports the efficiency of the SIPDD method, because the smaller spectral radius leads to faster convergence in the iterative method. Note that the spectral radius of the interface scheme decreases a little. because $\delta = h/H$ is relatively small. We also point out that there is no interface prediction scheme needed, when $P = 1$ which is the FIS scheme.

Table 2. Theoretical spectral radii at the various P of the SIPDD

P	1(=FIS)	2	5	10	20	25
$\rho(G_H)$ Prediction	N/A	**0.6390**	**0.6394**	**0.6413**	**0.6487**	**0.6541**
$\rho(G_P)$ Interior	**0.7893**	**0.7882**	**0.7800**	**0.7515**	**0.6461**	**0.5753**

Table 3 shows the actual and experimental first 6 largest absolute eigenvalues computed by MATALB [9] of the GS iteration matrices of the SIPDD method at the various number of subdomains P for $u_{tt} = u_{xx} + u_{yy}$ with $\Delta x = \Delta y = 0.01$ and $\Delta t = 0.02$. We can see in Table 3 that the largest experimental eigenvalue is exactly the same as the theoretical eigenvalue in Table 2.

Table 3. Actual first 6 largest eigenvalues computed by MATLAB

P	1(=FIS)	2	5	10	20	25
		0.6390	**0.6394**	**0.6413**	**0.6487**	**0.6541**
		0.6371	0.6378	0.6395	0.6469	0.6523
Prediction	N/A	0.6339	0.6375	0.6394	0.6469	0.6523
		0.6295	0.6359	0.6376	0.6451	0.6505
		0.6239	0.6358	0.6367	0.6439	0.6493
		0.6171	0.6344	0.6363	0.6438	0.6493
	0.7893	**0.7882**	**0.7800**	**0.7515**	**0.6461**	**0.5753**
	0.7882	0.7870	0.7789	0.7504	0.6450	0.5743
Interior	0.7882	0.7851	0.7769	0.7485	0.6433	0.5727
	0.7870	0.7835	0.7742	0.7459	0.6408	0.5703
	0.7862	0.7824	0.7708	0.7425	0.6377	0.5674
	0.7862	0.7804	0.7666	0.7383	0.6338	0.5638

4. Speedup and Operation ratio of the domain decomposition method

In this section, we provide two measurements for the efficiency of the SIPDD algorithm. We will see that each measurement is consistent to the efficiency of the SIPDD method shown in the previous section. A common measurement of the efficiency of a parallel algorithm is the speedup that is defined by

$$\text{Speedup} = \frac{\text{Execution time for a single processor}}{\text{Execution time using } P \text{ processors}}.$$

Since the algorithm is simulated with one processor, the true parallel execution time using P processors is roughly equivalent to the total CPU time in seconds (T_P) obtained by the simulation being divided by P. Thus, the speedup S_P can be simply written as

$$S_P = \frac{T_1}{T_P/P}.$$

In this paper, the Gauss-Seidel iterative scheme is used to solve the large and sparse linear systems generated by the SIPDD method. The stopping criterion in the GS iterative procedure is given by

$$\frac{||w^{(n)} - w^{(n-1)}||_2}{||w^{(n)}||_2} < \epsilon$$

where $w^{(n)}$ is the estimate at the nth iteration and ϵ is a preset small value. We choose $\epsilon = 10^{-6}$ for our experiments. All of the numerical experiments are carried out on a desktop computer with Intel(R) Core(TM) i7-8700 CPU at 3.20GHz with 8.0GB RAM.

Table 4 shows the maximum error, total CPU time (T_P), and Speedup (S_P) of the two model problems such as MP1: $u_{tt} = u_{xx} + u_{yy}$ and MP2: $u_{tt} + 20u_t + 25u = u_{xx} + u_{yy} + f(x, y, t)$ at the final time level $t = 1$ with the

various P of the SIPDD method, where $\Delta x = \Delta y = 0.01$, $\Delta t = 0.02$, and $f(x, y, t) = 4e^{-t} \sinh x \sinh y$. In Table 4, we see that the SIPDD method is very efficient with excellent speedups when P is large.

Table 4. Maximum error and Speedup at the various P of the SIPDD

	P	1(=FIS)	2	5	10	20	25
	Error	0.87e-4	0.31e-3	0.29e-4	0.29e-4	0.51e-4	0.43e-4
MP1	T_P	5.5781	5.4219	5.2656	4.5938	3.2813	2.9531
	S_P	**1**	**2.0576**	**5.2967**	**12.1429**	**34.0000**	**47.2222**
	Error	0.55e-4	0.83e-4	0.98e-5	0.11e-4	0.23e-4	0.17e-4
MP2	T_P	4.6875	4.6719	4.5313	3.9844	3.0313	2.6875
	S_P	**1**	**2.0067**	**5.1724**	**11.7647**	**30.9278**	**43.6047**

Another measurement of the efficiency of a parallel algorithm is the operation ratio. Suppose the whole domain is decomposed into P subdomains. Let J_H and J_P be the number of iterations of the Gauss-Seidel iterative process of the prediction scheme and the interior scheme of the SIPDD algorithm, respectively. Let O_P be the total number of operations where P is the number of subdomains of the SIPDD algorithm. Then the total number of operations O_P of the SIPDD algorithm can be written as the followings.

(1) Prediction requires $(N-1) \cdot J_H \cdot (50MP - 50M - 50P + 50)$ operations.
(2) Interior requires $(N-1) \cdot J_P \cdot (50LM - 50MP - 50L + 50P)$ operations.
(3) Total operations are $O_P = (N-1) \cdot J_H \cdot (50MP - 50M - 50P + 50) + (N-1) \cdot J_P \cdot (50LM - 50MP - 50L + 50P)$.

Now, we define the operation ratio O_R by

$$O_R = \frac{O_1}{O_P/P}.$$

Table 5 shows the total number of operations and operation ratio (O_R) of the model problems $u_{tt} = u_{xx} + u_{yy}$ and $u_{tt} + 20u_t + 25u = u_{xx} + u_{yy} + f(x, y, t)$ at the final time level $t = 1$ with the various P of the SIPDD method, where $\Delta x = \Delta y = 0.01$, $\Delta t = 0.02$, so that $\lambda = 8$. In Table 5, we see that the operation ratio O_R is very similar to the speedup S_P. In other words, the experimental speedup is close to the theoretical operation ratio of the SIPDD algorithm, which shows the SIPDD method is very efficient.

Table 5. Total operations and operation ratio at the various P of the SIPDD

P	1(=FIS)	2	5	10	20	25
(1) Prediction	N/A	7.48e+6	2.99e+7	6.78e+7	1.47e+8	1.89e+8
(2) Interior	1.40e+9	1.37e+9	1.28e+9	1.05e+9	6.13e+8	4.54e+8
(3) O_P	1.40e+9	1.38e+9	1.31e+9	1.12e+9	7.60e+8	6.44e+8
(4) O_R	**1**	**2.0224**	**5.3470**	**12.4815**	**36.8706**	**54.4354**

((1) Pred. operations, (2) Int. operations, (3) Total operations, (4) Operation ratio)

5. Conclusion

The second-order implicit prediction domain decomposition (SIPDD) method is known to be an accurate, unconditionally stable, and efficient numerical method for solving the two-dimensional telegraph equations. It is well known that the rate of convergence of an iterative algorithm depends on the spectral radius of the iteration matrix. In this paper, we formulate the theoretical spectral radius of the iteration matrix generated by the SIPDD method. We see that the spectral radius decreases when the number of decomposed subdomains increases, which is confirmed by the numerical experiments. In addition, experimental speedup and theoretical operation ratio of the SIPDD method are compared as the measurements of the efficiency of the method. Numerical results show that the method is very efficient with the excellent speedup which is confirmed by the theoretical operation ratio of the domain decomposition method.

References

[1] W.F. Ames, *Numerical methods for partial differential equations*, Academic Press, 1992.
[2] M. Asif, N. Haider, Q. Al-Mdallal, I. Khan, *A Haar wavelet collocation approach for solving one and two-dimensional second-order linear and nonlinear hyperbolic telegraph equations*, Numer. Methods Partial Differential Equations **36** (2020), 1962–1981.
[3] M. Aslefallah, D. Rostamy, *Application of the singular boundary method to the two-dimensional telegraph equation on arbitrary domains*, J. Engrg. Math. **118** (2019), 1–14.
[4] R.L. Burden, J.D. Faires, *t Numerical Analysis*, Thomson Brooks/Cole, 2005.
[5] M. Dehghan, R. Salehi, *A method based on meshless approach for the numerical solution of the two-space dimensional hyperbolic telegraph equation*, Math. Methods Appl. Sci. **35** (2012), 1220–1233.
[6] R. Jiwari, S. Pandit, R.C. Mittal, *A differential quadrature algorithm to solve the two dimensional linear hyperbolic telegraph equation with Dirichlet and Neumann boundary conditions*, Appl. Math. Comput. **218** (2012), 7279–7294.
[7] Y. Jun, *An unconditionally stable implicit high-order domain decomposition method for two-dimensional second-order hyperbolic problems*, Appl. Math. Sci. **6** (2012), 3301–3312.
[8] W. Ma, B. Zhang, H. Ma, *A meshless collocation approach with barycentric rational interpolation for two-dimensional hyperbolic telegraph equation*, Appl. Math. Comput. **279** (2016), 236–248.
[9] MATHWORKS, Inc. http://www.mathworks.com.
[10] R.C. Mittal, R. Bhatia, *A numerical study of two dimensional hyperbolic telegraph equation by modified B-spline differential quadrature method*, Appl. Math. Comput. **244** (2014), 976–997.
[11] D.M. Young, *Iterative Solution of Large Linear Systems*, Dover, New York, 2003.

Younbae Jun
Department of Applied Mathematics, Kumoh National Institute of Technology, Gyeongbuk 39177, Republic of Korea
E-mail address: yjun@kumoh.ac.kr