J. Korean Soc. Math. Educ. Ser. B: Pure Appl. Math. ISSN 1226-0657
Volume 17, Number 4 (November 2010), Pages 289–298

AN EFFICIENT SECOND-ORDER NON-ITERATIVE FINITE DIFFERENCE SCHEME FOR HYPERBOLIC TELEGRAPH EQUATIONS

YOUNBAE JUN [a] AND HONGTAEK HWANG [b]

ABSTRACT. In this paper, we propose a second-order prediction/correction (SPC) domain decomposition method for solving one dimensional linear hyperbolic partial differential equation $u_{tt} + a(x,t)u_t + b(x,t)u = c(x,t)u_{xx} + f(x,t)$. The method can be applied to variable coefficients problems and singular problems. Unconditional stability and error analysis of the method have been carried out. Numerical results support stability and efficiency of the method.

1. INTRODUCTION

Hyperbolic partial differential equations arise in many phenomena such as wave mechanics, vibrations, aerodynamic flows, flows of fluids and contaminants through a porous media, neutron diffusion and radiation transfer. In this paper, we consider the following one dimensional linear second-order hyperbolic partial differential equation, so-called a telegraph equation, of the form

$$u_{tt} + a(x,t)u_t + b(x,t)u = c(x,t)u_{xx} + f(x,t) \tag{1.1}$$

defined in $\{(x,t)|0 \le x \le 1, 0 \le t \le T\}$, with the initial conditions and the Dirichlet boundary conditions

$$u(x,0) = g_0(x), u_t(x,0) = g_1(x) \text{ for } 0 \le x \le 1, \tag{1.2}$$

$$u(0,t) = h_0(t), u(1,t) = h_1(t) \text{ for } t \ge 0. \tag{1.3}$$

We often use finite difference scheme [2, 15, 16, 17, 18] or finite element scheme [4, 8, 13, 19] to solve initial and boundary value problems. In particular, there are several unconditionally stable finite difference schemes for solving hyperbolic equations such as three-level implicit scheme [1], Mohanty's three-level scheme [15], compact difference scheme [6], and compact/collocation method [14].

Received by the editors April 8, 2010. Revised September 30, 2010. Accepted November 20, 2010.
2000 *Mathematics Subject Classification.* 65M06.
Key words and phrases. second-order accuracy, domain decomposition, finite difference method, hyperbolic telegraph equation, unconditional stability.

Due to recent powerful parallel computational capability, efficient parallel algorithm is one of the biggest issues in computational mathematics. Domain decomposition (DD) is known to be very efficient parallelism [5, 7, 9, 10, 11, 12]. Dawson and Dupont [5] proposed a non-iterative DD method related to the explicit/implicit conservative Galerkin method. This method is conditionally stable even though the constraint is less than the fully explicit method. Gander and Halpern [7] proposed an overlapping Schwarz method. Recently, Jun [12] has proposed a non-overlapping prediction/correction (PC) method for linear hyperbolic equations. The performance of the PC method [12] is much more efficient with respect to CPU time than the three-level implicit scheme [1]. However, it was observed that truncation error increases when the number of decomposed subdomains increases, since the prediction scheme of the PC method uses the boundary points. In this paper, we provide an algorithm using adjacent interface points instead of boundary points so that the error term of the interface scheme of the new method is of second order. By doing this way, the overall truncation error of the new method is getting less than the PC method.

In Section 2, we discuss existing DD algorithms and propose a second-order non-iterative DD algorithm, as well as we analyze stability of the new method. Numerical experiments and efficiency of the method are provided in Section 3. We make concluding remarks in Section 4.

2. SECOND-ORDER PREDICTION/CORRECTION ALGORITHM AND STABILITY

First, we describe existing non-domain decomposition algorithm [1], which is three-level implicit scheme, to solve the hyperbolic telegraph problem (1.1)–(1.3). Second, we describe the Prediction/Correction (PC) domain decomposition (DD) algorithm. Then, new second-order algorithm is introduced.

We choose the positive integers L and N so that $h = 1/L$ and $k = T/N$. Let w_i^n be the approximated value to the exact value u_i^n at the grid point (x_i, t_n) where $x_i = ih$ and $t_n = nk$. We denote $a(x_i,t_n)$, $b(x_i,t_n)$, $c(x_i,t_n)$, and $f(x_i,t_n)$ by a_i^n,b_i^n, c_i^n, and f_i^n, respectively. We define the difference operators w_{tt}^n, w_t^n, w_{xx}^n by

(2.1)
$$w_{tt}^n = \frac{w_i^{n+1} - 2w_i^n + w_i^{n-1}}{k^2}, \quad w_t^n = \frac{w_i^{n+1} - w_i^{n-1}}{2k}, \quad w_{xx}^n = \frac{w_{i+1}^n - 2w_i^n + w_{i-1}^n}{h^2}.$$

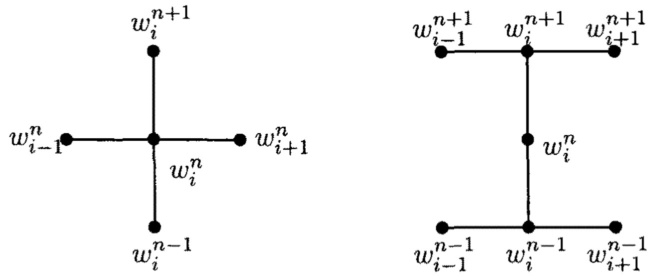
2.1. Three-level Implicit Scheme It is well-known [1] that the following three-level implicit scheme is unconditionally stable if we choose a free variable $\gamma \geq 1/4$, and conditionally stable if we use $\gamma < 1/4$:

$$w_{tt}^n + a_i^n w_t^n + b_i^n w_i^n = c_i^n \left(\gamma w_{xx}^{n+1} + (1-2\gamma) w_{xx}^n + \gamma w_{xx}^{n-1}\right) + f_i^n. \tag{2.2}$$

When $\gamma = 0$, the algorithm is referred to as the fully explicit scheme (FES). It is known [1] that FES is stable only if $\lambda \leq 1$, where the mesh ratio $\lambda = k/h = \Delta t/\Delta x$. Whereas, if $\gamma = 1/2$, the algorithm is referred to as the fully implicit scheme (FIS) which is unconditionally stable for $0 < \lambda < \infty$. We note that both FES and FIS are not domain decomposition method. Three-level stencils of FES and FIS are provided in Fig. 1. It is clear to see from Equation (2.2) that these schemes are written as follows:

Fully explicit scheme vs. Fully implicit scheme

- Fully explicit scheme (FES): $w_{tt}^n + a_i^n w_t^n + b_i^n w_i^n = c_i^n w_{xx}^n + f_i^n$
- Fully implicit scheme (FIS): $w_{tt}^n + a_i^n w_t^n + b_i^n w_i^n = c_i^n \frac{1}{2}\left[w_{xx}^{n+1} + w_{xx}^{n-1}\right] + f_i^n$



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

Figure 1. Three-level stencils of three-level implicit schemes

In Section 3, we demonstrate numerically that FES is conditionally stable and FIS is unconditionally stable.

2.2. Prediction/Correction Method The Prediction/Correction (PC) method has been proposed in [12] as a non-iterative unconditionally stable domain decomposition method for solving hyperbolic partial differential equations. The PC algorithm involves prediction and correction steps. The prediction of the interface values of the PC method is based on Taylor series expansion with the knowledge of boundary values. For a given function $v(x)$, we easily approximate $v''(x_i)$ as follows:

$$v''(x_i) \approx \frac{2[pv(x_L) - (p+q)v(x_i) + qv(x_0)]}{pq(p+q)H^2},$$

where $pH = x_i$, $qH = x_L - x_i$, and $H = 1/P$. Here P is the number of decomposed subdomains. Thus, we define an operator to approximate u_{xx} by

$$\bar{w}_{xx}^n = \frac{2[pu_L^n - (p+q)w_i^n + qu_0^n]}{pq(p+q)H^2}, \tag{2.3}$$

where u_L^n and u_0^n are boundary values.

In the PC algorithm [12], the interface value is computed using Equation (2.3). Then, the values of interior points in each subdomain are computed by the fully implicit scheme (FIS) and the interface points are updated using the FIS. The PC algorithm is as follows:

Prediction/Correction (PC) algorithm:

Step1: Interface prediction: $w_{tt}^n + a_i^n w_t^n + b_i^n w_i^n = c_i^n \frac{1}{2}\left[\bar{w}_{xx}^{n+1} + \bar{w}_{xx}^{n-1}\right] + f_i^n$, where $\bar{w}_{xx}^{n+1}$ and $\bar{w}_{xx}^{n-1}$ are defined in Equation (2.3)

Step2: Interior region: $w_{tt}^n + a_i^n w_t^n + b_i^n w_i^n = c_i^n \frac{1}{2}\left[w_{xx}^{n+1} + w_{xx}^{n-1}\right] + f_i^n$, which is the fully implicit scheme

Step3: Interface correction: $w_{tt}^n + a_i^n w_t^n + b_i^n w_i^n = c_i^n \frac{1}{2}\left[w_{xx}^{n+1} + w_{xx}^{n-1}\right] + f_i^n$, where w_{i+1}^{n+1} and w_{i-1}^{n+1} are computed values in Step2

Step4: Repeat Step1 through Step3 until last time level

2.3. Second-order Prediction/Correction Method A drawback of the PC method is that truncation error increases when the number of decomposed subdomains (P) increases, since the prediction scheme of the PC method uses the boundary points. In this section, we propose a new method in which truncation error decreases when P increases. One reason of reduced error is that we use adjacent interface points. Instead of computing the estimations of the interface points individually, which is used in the PC method, we use central finite difference scheme to compute the estimations on all the interface points at a time. The central difference scheme can be formulated as follows:

$$v''(x) = \frac{v(x+H) - 2v(x) + v(x-H)}{H^2} + O(H^2),$$

where H is distance of adjacent interface points. Also, H is an integer multiple of h. The accuracy of the new interface prediction scheme is of second order $O(H^2)$. Thus, it is clear that truncation error decreases when $P(= 1/H)$ increases. So, we define an operator to approximate u_{xx} by

$$\hat{w}_{xx}^n = \frac{w_{i+LH}^n - 2w_i^n + w_{i-LH}^n}{H^2}, \tag{2.4}$$

where w_{i+LH}^n and w_{i-LH}^n are adjacent interface points. We predict the interface values using the following scheme:

$$w_{tt}^n + a_i^n w_t^n + b_i^n w_i^n = c_i^n \frac{1}{2}\left[\hat{w}_{xx}^{n+1} + \hat{w}_{xx}^{n-1}\right] + f_i^n. \tag{2.5}$$

Once the interface values are predicted, we compute the interior points using FIS. Then, the interface points are updated using FIS. In the correction step, w_{i+1}^{n+1} and w_{i-1}^{n+1} are computed values in the interior step. Furthermore, systems which are generated by the prediction and interior schemes are only tri-diagonal linear systems and they are solved by non-iterative Crout factorization method [3]. Three-level stencils of the three steps of the new method are provided in Fig. 2. We call the new method the second-order prediction/correction (SPC) method that is summarized as follows:

Second-order Prediction/Correction (SPC) algorithm:

Step1: Interface prediction: $w_{tt}^n + a_i^n w_t^n + b_i^n w_i^n = c_i^n \frac{1}{2}\left[\hat{w}_{xx}^{n+1} + \hat{w}_{xx}^{n-1}\right] + f_i^n$, where $\hat{w}_{xx}^{n+1}$ and $\hat{w}_{xx}^{n-1}$ are defined in Equation (2.4)

Step2: Interior region: $w_{tt}^n + a_i^n w_t^n + b_i^n w_i^n = c_i^n \frac{1}{2}\left[w_{xx}^{n+1} + w_{xx}^{n-1}\right] + f_i^n$, which is the fully implicit scheme

Step3: Interface correction: $w_{tt}^n + a_i^n w_t^n + b_i^n w_i^n = c_i^n \frac{1}{2}\left[w_{xx}^{n+1} + w_{xx}^{n-1}\right] + f_i^n$, where w_{i+1}^{n+1} and w_{i-1}^{n+1} are computed values in Step2

Step4: Repeat Step1 through Step3 until last time level

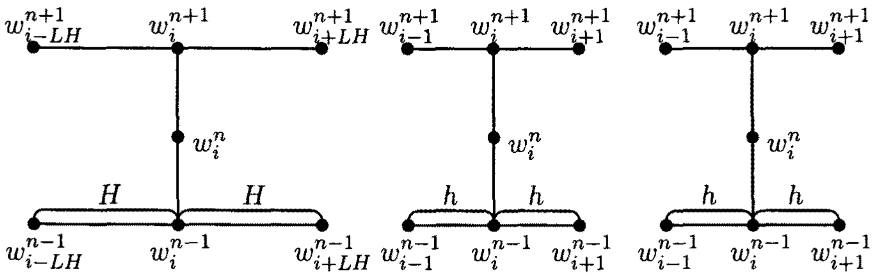


Figure 2. Three steps of the Second-order Prediction/Correction (SPC) algorithm

In order to obtain w_i^1 for $u(x,k)$, we use a Taylor series expansion with respect to time

$$u(x,k) = u(x,0) + ku_t(x,0) + \frac{k^2}{2}u_{tt}(x,0) + O(k^3).$$

Since $u(x,0) = g_0(x) = w_i^0$, $u_t(x,0) = g_1(x)$, and $u_{tt}(x,0) = c(x,0)u_{xx}(x,0) - a(x,0)u_t(x,0) - b(x,0)u(x,0) + f(x,0)$, the above $u(x,k)$ equation is approximated by

$$w_i^1 = w_i^0 + k(g_1)_i^0 + \frac{k^2}{2}\left[c_i^0 \frac{w_{i-1}^0 - 2w_i^0 + w_{i+1}^0}{h^2} - a_i^0(g_1)_i^0 - b_i^0 w_i^0 + f_i^0\right].$$

2.4. Stability and Error Analysis of the SPC Method We analyze stability and truncation error of the new method. As we see in the SPC algorithm, three-level stencils of the prediction scheme (Fig. 2(a)) is slightly different from those of the interior and correction schemes (Fig. 2(b),(c)). Thus, we prove two similar but different theorems.

Theorem 2.1. *The interface prediction scheme of the SPC method is unconditionally stable and the error term for the scheme is* $|w_i^n - u_i^n| = O(H^2 + k^2)$.

Proof. It is known [1] that the fully implicit scheme (FIS) is unconditionally stable. The SPC interface scheme is a FIS scheme, in which the step size of the x-direction is H. Thus, the SPC interface scheme is unconditionally stable. Furthermore, we have $|w_i^n - u_i^n| = O(H^2 + k^2)$, since $w_{tt}^n = u_{tt} + O(k^2)$, $w_t^n = u_t + O(k^2)$, $\hat{w}_{xx}^n = u_{xx} + O(H^2)$. □

With the same argument in the previous proof, we can see immediately the following theorem.

Theorem 2.2. *The interior and interface correction schemes of the SPC method are unconditionally stable and the error term of each scheme is* $|w_i^n - u_i^n| = O(h^2 + k^2)$.

Proof. The interior and correction schemes of the SPC method are FIS, where the step size of the x-direction is h. Thus, they are clearly unconditionally stable. Also, $w_{tt}^n = u_{tt} + O(k^2)$, $w_t^n = u_t + O(k^2)$, and $w_{xx}^n = u_{xx} + O(h^2)$. Therefore, $|w_i^n - u_i^n| = O(h^2 + k^2)$. □

All of the interface prediction, interior, and interface correction schemes of the SPC method are FIS methods. Hence, the SPC method is unconditionally stable for $0 < \lambda < \infty$.

3. Numerical Examples and Efficiency of the SPC Method

In this section, we test stability and efficiency of the new method using three model problems. The exact solution for each model problem is $u(x,t) = e^{-2t}\sinh x$. The initial and boundary conditions and $f(x,t)$ are derived from the exact solution. We note that the exact solution is unknown, in general. However, we use the exact solution for the comparison of numerical experiment in this paper.

First, we compare four different methods for stability: (1) the fully explicit scheme (FES); (2) the fully implicit scheme (FIS); (3) the Prediciton/Correction method of five subdomains (PC(5)); (4) the Second-order Prediciton/Correction method of five subdomains (SPC(5)). The FIS method is used as the benchmark for stability, since it is well-known unconditionally stable method. All of the numerical experiments are carried out on a desktop computer with Intel Core2 Duo CPU at 2.93GHz with 4.0 GB of RAM.

Second, we test efficiency of the SPC method. Total CPU time in seconds with various number of subdomains is provided in this section. Since the algorithm is simulated with one processor, the true parallel execution time using P processors is roughly equivalent to the total CPU time divided by P. Thus, the parallel execution time is referred to as parallel CPU time.

Example 3.1. Model Problem 1 (MP1):

$$u_{tt} + 2e^{x+t}u_t + \sin^2(x+t)u = (1+x^2)u_{xx} + f(x,t)$$

Table 1 shows the maximum error of MP1 between the approximated solution and the exact solution $||w^N - u^N||_\infty$ at the final time $T = 1$ with various $\lambda(= k/h)$ ranging from 1/2 to 200 using four different methods: FES, FIS, PC(5), and SPC(5). As we can see in Table 1, FES is conditionally stable and FIS is unconditionally stable, as expected. Both PC and SPC methods are unconditionally stable, but we can see that the SPC is more accurate than PC. Table 2 shows the maximum error and CPU time of SPC with various P for $h = 1/2000$ and $k = 1/1000$. We can see in Table 2 that the SPC method is unconditionally stable not only at $P = 5$, but also at various P. Also, we see that PCPU time in seconds decreases when P

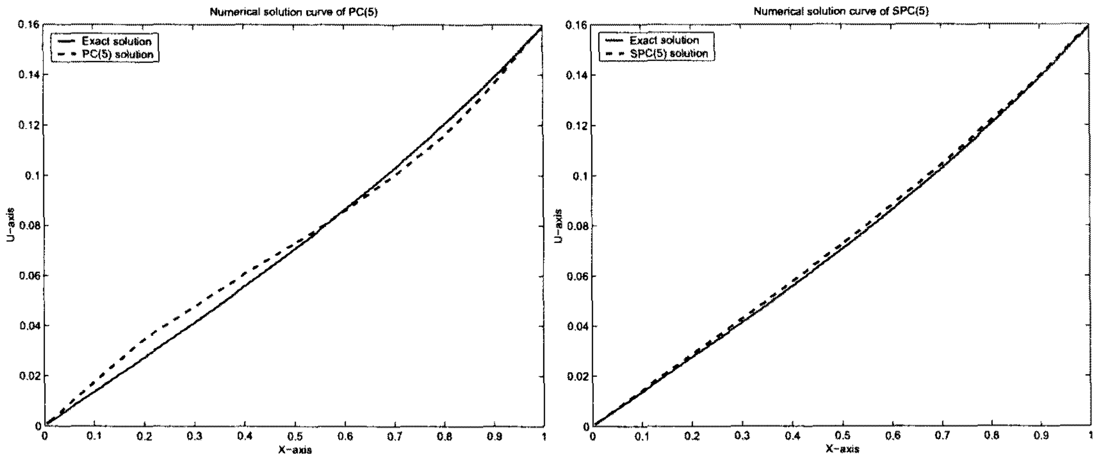
increases. We note that when $P = 1$ is used, the SPC method is equivalent to FIS, which is non-domain decomposition. Fig. 3 shows numerical solution curves of MP1 using two schemes PC(5) and SPC(5) at the final time $t = 1$ with $h = 1/2000$ and $k = 1/1000$. Exact solution curve is provided as reference. As we can see in Fig. 3, SPC(5) curve is closer to the exact solution curve than PC(5) curve, which supports that accuracy of the SPC method is better than PC.

Table 1. Maximum error of MP1 with various λ

h	k	λ	FES	FIS	PC(5)	SPC(5)
1/2000	1/10	200	∞	0.1968e-2	0.7440e-2	0.2009e-2
1/2000	1/20	100	∞	0.4878e-3	0.6334e-2	0.5659e-3
1/2000	1/1000	2	∞	0.1987e-6	0.1828e-4	0.1312e-5
1/2000	1/4000	1/2	0.9716e-8	0.1292e-7	0.1156e-5	0.8360e-7

Table 2. Maximum error and CPU time of SPC with various P (TCPU=Total CPU time, PCPU=Parallel CPU time)

P	1=FIS	2	5	10	20
Error	0.1987e-6	0.2539e-5	0.1312e-5	0.7693e-6	0.4785e-6
TCPU	0.375	0.45313	0.56250	0.76563	1.125
PCPU	0.375	0.22657	0.1125	0.07656	0.05625



(a) Solution curve of PC(5) (b) Solution curve of SPC(5)

Figure 3. Solution curves of PC(5) and SPC(5) to MP1

Example 3.2. Model Problem 2 (MP2):

$$u_{tt} + \frac{2}{x^2}u_t + \frac{1}{x^2}u = (1 + x^2)u_{xx} + f(x, t)$$

Maximum error and CPU time in seconds of the SPC method to MP2 at the final time $t = 1$ with $h = 1/2000$ and $k = 1/1000$ are provided in Tables 3 and 4.

Table 3. Maximum error of MP2 with various λ

h	k	λ	FES	FIS	PC(5)	SPC(5)
1/2000	1/10	200	∞	0.1217e-2	0.4658e-2	0.1362e-2
1/2000	1/20	100	∞	0.3272e-3	0.3107e-2	0.3604e-3
1/2000	1/1000	2	∞	0.1275e-6	0.1564e-4	0.6171e-6
1/2000	1/4000	1/2	0.6871e-8	0.8180e-8	0.9858e-6	0.3929e-7

Table 4. Maximum error and CPU time of SPC to MP2 (TCPU=Total CPU time, PCPU=Parallel CPU time)

P	1=FIS	2	5	10	20
Error	0.1275e-6	0.1294e-5	0.6171e-6	0.3761e-6	0.2467e-6
TCPU	0.375	0.45313	0.56250	0.75	1.125
PCPU	0.375	0.22657	0.1125	0.075	0.05625

Example 3.3. Model Problem 3 (MP3):

$$u_{tt} + 20u_t + 25u = u_{xx} + f(x,t)$$

Maximum error and CPU time in seconds of the SPC method to MP3 at the final time $t = 1$ with $h = 1/2000$ and $k = 1/1000$ are provided in Tables 5 and 6.

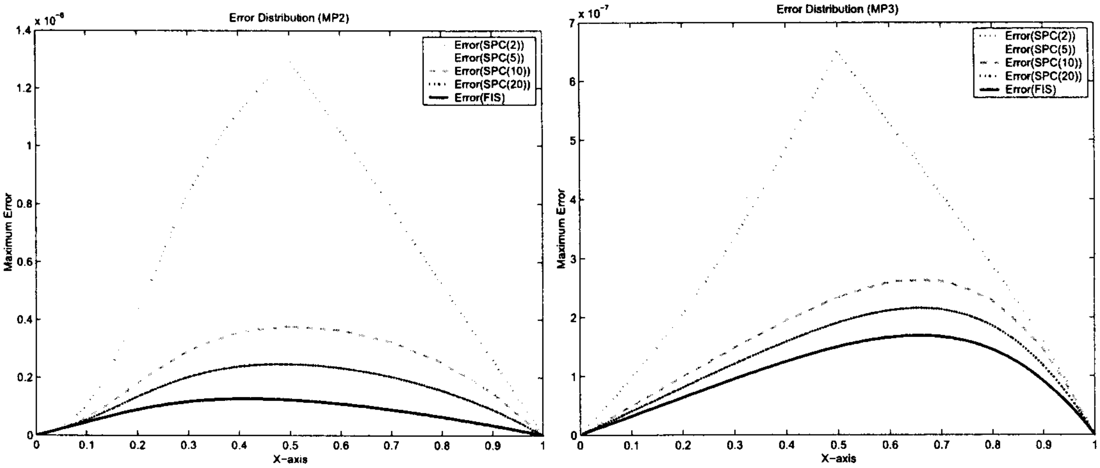
Table 5. Maximum error of MP3 with various λ

h	k	λ	FES	FIS	PC(5)	SPC(5)
1/2000	1/10	200	∞	0.1677e-2	0.2023e-2	0.1658e-2
1/2000	1/20	100	∞	0.4207e-3	0.1618e-2	0.4267e-3
1/2000	1/1000	2	∞	0.1688e-6	0.9290e-5	0.3577e-6
1/2000	1/4000	1/2	0.9935e-8	0.1066e-7	0.5870e-6	0.2268e-7

Table 6. Maximum error and CPU time of SPC to MP3 (TCPU=Total CPU time, PCPU=Parallel CPU time)

P	1=FIS	2	5	10	20
Error	0.1688e-6	0.6506e-6	0.3577e-6	0.2624e-6	0.2154e-6
TCPU	0.375	0.45313	0.56250	0.75	1.14063
PCPU	0.375	0.22657	0.1125	0.075	0.05703

Fig. 4 shows distribution of the error $|w_i^n - u_i^n|$ at the final time $t = 1$ of MP2 and MP3 using the SPC(P) method with various P, where $h = 1/2000$ and $k = 1/1000$. As you can see in Fig. 4 that truncation error decreases when P increases, which is consistent to our theory. FIS is the benchmark which is unconditionally stable and non-domain decomposition.



(a) Error distribution to MP2 (b) Error distribution to MP3

Figure 4. Error distribution of SPC with various P

4. Conclusion

In this paper, we present a second-order prediction/correction (SPC) non-iterative finite difference scheme using domain decomposition for solving one dimensional linear hyperbolic partial differential equations. The interface prediction scheme of the new SPC method is of order $O(H^2 + k^2)$. The interior and interface correction schemes of SPC are of order $O(h^2 + k^2)$. The SPC scheme is unconditionally stable for $0 < \lambda < \infty$ and can be used to variable coefficients problems and singular problems. Numerical experiments show that the SPC method is stable for any choice of $\lambda(= h/k)$ and efficient domain decomposition method. It is hoped that the SPC scheme for one dimensional telegraph equations is easily applied to two dimensional problems.

Acknowledgement

The authors thank the referees for their valuable comments.

References

1. W.F. Ames: *Numerical methods for partial differential equations.* Academic Press, 1992.
2. J.J. Benito, F. Ureña & L. Gravete: Solving parabolic and hyperbolic equations by the generalized finite difference method. *J. Comput. Appl. Math.* **209** (2007), 208–233.
3. R.L. Burden & J.D. Faires: *Numerical Analysis*, Thomson Brooks/Cole, 2005.
4. R. Codina: Finite element approximation of the hyperbolic wave equation in mixed form. *Comput. Methods Appl. Mech. Engrg.* **197** (2008), 1305–1322.
5. C.N. Dawson & T.F. Dupont: Noniterative domain decomposition for second order hyperbolic problems. *Contemp. Math.* **157** (1994), 45–52.

6. H. Ding & Y. Zhang: A new unconditionally stable compact difference scheme of $O(\tau^2 + h^4)$ for the 1D linear hyperbolic equation. *Appl. Math. Comput.* **207** (2009), 236–241.
7. M.J. Gander & L. Halpern: Absorbing boundary conditions for the wave equation and parallel computing. *Math. Comp.* **74** (2005), 153–176.
8. X. He & T. Lü: A finite element splitting extrapolation for second order hyperbolic equations. *SIAM J. Sci. Comput.* **31** (2009/10), 4244–4265.
9. Y. Jun & T.Z. Mai: ADI method – Domain decomposition. *Appl. Numer. Math.* **56** (2006), 1092–1107.
10. ______: Rectangular domain decomposition method for parabolic problems. *J. Korea Soc. Math. Educ. Ser. B: Pure and Appl. Math.* **13** (2006), 281–294.
11. Y. Jun: An efficient domain decomposition decomposition method for three-dimensional parabolic problems. *Appl. Math. Comput.* **215** (2009), 2815–2825.
12. ______: A stable noniterative Prediction/Correction domain decomposition method for hyperbolic problems. *Appl. Math. Comput.* **216** (2010), 2286–2292.
13. Y. Liu & H. Li: H^1-Galerkin mixed finite element methods for pseudo-hyperbolic equations. *Appl. Math. Comput.* **212** (2009), 446–457.
14. A. Mohebbi & M. Dehghan: High order compact solution of the one-space-dimensional linear hyperbolic equation. *Numer. Methods Partial Differential Equations* **24** (2008), 1222–1235.
15. R.K. Mohanty: An unconditinally stable finite difference formula for a linear second order one space dimensional hyperbolic equation with variable coefficnents. *Appl. Math. Comput.* **165** (2005), 229–236.
16. ______: Stability interval for explicit difference schemes for multi-dimensional second-order hyperbolic equations with significant first-order space derivative terms. *Appl. Math. Comput.* **190** (2007), 1683–1690.
17. M. Ramezani, M. Dehghan & M. Razzaghi: Combined finite difference and spectral methods for the numerical solution of hyperbolic equation with an integral condition. *Numer. Methods Partial Differential Equations* **24** (2008), 1–8.
18. K.K. Sharma & P. Singh: Hyperbolic partial differential-difference equation in the mathematical modeling of neuronal firing and its numerical solution. *Appl. Math. Comput.* **201** (2008), 229–238.
19. Z. Zhang & D. Deng: A new alternating-direction finite element method for hyperbolic equation. *Numer. Methods Partial Differential Equations* **23** (2007), 1530–1559.

[a]DEPARTMENT OF APPLIED MATHEMATICS, KUMOH NATIONAL INSTITUTE OF TECHNOLOGY, GUMI, GYEONGBUK 730-701, KOREA
Email address: `yjun@kumoh.ac.kr`

[b]DEPARTMENT OF APPLIED MATHEMATICS, KUMOH NATIONAL INSTITUTE OF TECHNOLOGY, GUMI, GYEONGBUK 730-701, KOREA
Email address: `hht51@kumoh.ac.kr`