

Parallel Iterative Algorithms with Accelerate Convergence for Solving Implicit Difference Equations

¹Qing-Fu Liu, ¹Wei-Jun Zhong and ²Mancho Manev

¹School of Economics and Management,
Southeast University, Nanjing, 210096, PR China

²Faculty of Mathematics and Informatics,
Paisii Hilendarski University of Plovdiv, 4000, Bulgaria

Abstract: This study focuses on the solution of implicit difference equations, which are very difficult to compute in parallel for the diffusion equation. For improving the convergence rates and the properties of gradual-approach convergence of Segment-Classic-Implicit-Iterative (SCII) and Segment-Crank-Nicolson-Iterative (SCNI) algorithms realizing efficient iterative computation in parallel by segmenting grid domains, SCII and SCNI algorithms with accelerated convergence are studied and improved through inserting classic implicit schemes and Crank-Nicolson schemes into them respectively. The SCII and SCNI algorithms with accelerated convergence, which can be decomposed into smaller strictly tri-diagonally dominant subsystems, are solved by using a double-sweep algorithm. In the present study, general structures of SCII and SCNI algorithms with accelerated convergence are constructed with matrix forms. The convergent rates are estimated and properties of gradual-approach convergence about diffusion equation are described by a splitting coefficient matrix in detail. These algorithms improve the convergence rates in iteration while making the properties of gradual-approach convergence reach two ranks. The efficiency of computation is greatly enhanced. In addition, the algorithms are extended to the case of two-dimensional problem by studying Peaceman-Rachford scheme into which classic implicit schemes are inserted alternately. Numerical computations employing SCII and SCNI algorithms with accelerated convergence are made to SGL/Challenge L with 8 CPUs as examples. Theoretical analyses and numerical exemplifications show that the parallel iterative algorithms with accelerated convergence for solving one-dimensional diffusion equations are more efficient in computation and have much better convergent rates and properties of gradual-approach convergence.

Key words: Diffusion equation, parallel iterative algorithm, convergent rate, property of gradual-approach convergence

INTRODUCTION

Recently, parallel algorithms with such good properties such unconditional stable schemes and higher accuracy schemes for solving implicit difference equations have been improved greatly. Both ASE-I algorithm and ASC-N algorithms, known as segment implicit methods, are set up to solve different implicit equations^[9]. They realize the principle of divide and rule and efficient computation in parallel by segmenting grid domains. It turns out that iterative methods are of convergent properties, which was proved by splitting coefficient matrix^[1]. Segment Classic Implicit Iterative (SCII) and Segment Crank-Nicolson Iterative (SC-NI) algorithms for solving one-dimensional diffusion equation, which can be decomposed into smaller tri-diagonal subsystems, are solved by using a double-sweep algorithm^[8]. The convergent rate is estimated and the property of gradual-approach convergence about one-dimensional implicit difference equations is

discussed. The algorithms can solve implicit difference equations and have an efficiency in parallel^[2,5]. For improving the convergence rates and the properties of gradual-approach convergence of the SCII and SCNI algorithms, SCII and SCNI algorithms with accelerated convergence which can be decomposed into smaller strictly tri-diagonally dominant subsystems and be solved by using a double-sweep algorithm are studied and improved through the inserting classic implicit scheme and Crank-Nicolson scheme into them respectively. General structures of SCII and SCNI algorithms with accelerated convergence are described by using matrix form. The improved convergent rates and properties of the gradual-approach are described by a splitting coefficient matrix in detail. By using the algorithms, it can save much time to solve implicit difference equations in parallel. In addition, the algorithms are extended to two-space dimensional problems by studying Peaceman-Rachford scheme into which classic implicit schemes are inserted alternately.

Corresponding Author: Qing-fu Liu, School of Economics and Management, Southeast University, Southeast University, Nanjing, China, Tel: +86-25-3689347

vector of $U^{[k+1]}$, and let $U^{[k+1]^{(0)}} = U^{[k]}$ in computing in this study.

By splitting off (8), a SCII algorithm with accelerated convergence can be expressed as:

$$MU^{[k+1]^{(s)}} = NU^{[k+1]^{(s)}} + b \tag{9}$$

To balance the computing in parallel, m_i for $i = 1, 2, \dots, k$ is often made equal in performing. Since A_i for $i = 1, 2, \dots, k$ has been strictly tri-diagonally dominant and N_i for $i = 1, 2, \dots, k-1$ has only one non-zero element, massive computing in parallel about (9) is not difficult by parallel segmented double-sweep algorithm^[7,8].

Analyses of convergent rate and property of gradual-approach convergence: In this section, stability, convergent rate and property of gradual approach convergence of (9) are analyzed. We firstly introduce a famous lemma as follows^[3,4].

Lemma: If $M = (m_{i,j})$ is an $n \times n$ matrix, $N = (n_{i,j})$ is an $n \times m$ matrix and:

$$|m_{i,i}| > \sum_{j \neq i} |m_{i,j}| \text{ for } i = 1, 2, \dots, n$$

Then:

$$\|M^{-1}N\|_{\infty} \leq \max_i \left\{ \sum_{j=1}^m |n_{i,j}| / (|m_{i,i}| - \sum_{j \neq i} |m_{i,j}|) \right\}$$

It is easy to prove that (7) is unconditionally stable according to the above lemma. And the estimate result in the iteration of (9) is what follows:

$$\|M^{-1}N\|_{\infty} \leq \frac{r^2}{1+3r+r^2} \text{ for } 1 < m_i < m-1$$

Then, we have $\rho(M^{-1}N) \leq \|M^{-1}N\|_{\infty} < 1$. So SCII algorithm with accelerated convergence (9) is convergent. Obviously, the convergence rate of (9) is better than that of SCII algorithm^[2].

The property of gradual approach convergence will be discussed in detail in the following. For balancing the computing in parallel, we suppose $A_i = A_p$ for $i = 1, 2, \dots, k$. So we have:

$$M^{-1} = \text{diag}(A_p^{-1}, A_p^{-1}, \dots, A_p^{-1})$$

$$M^{-1}N = \frac{r^2}{1+2r} \begin{pmatrix} H_1 & A_p^{-1}N & & & & \\ A_p^{-1}M & H_2 & A_p^{-1}N & & & \\ & O & O & O & & \\ & & A_p^{-1}M & H_{k-1} & A_p^{-1}N & \\ & & A_p^{-1}M & H_k & & \end{pmatrix}$$

Then:

$$\|M^{-1}N\|_{\infty} = \frac{r^2}{1+2r} \max \left\{ \|A_p^{-1}(M_p + N_p)\|_{\infty}, \|A_1^{-1}N_p\|_{\infty}, \|A_k^{-1}M_p\|_{\infty} \right\} \tag{10}$$

The result of $\|A_p^{-1}(M_p + N_p)\|_{\infty}$ will be estimated in the following section.

Since $A_p^{-1} > 0$, $M \geq 0$ and $N \geq 0$, supposing $X = (x_{i,j})_{p \times p} = A_p^{-1}$, Then we have:

$$A_p^{-1}(M + N) = X(M + N)$$

$$= \begin{pmatrix} 0 & x_{1,p} & 0 & L & 0 & x_{1,1} & 0 \\ 0 & x_{2,p} & 0 & L & 0 & x_{2,1} & 0 \\ M & M & & & M & M & \\ 0 & x_{p-1,p} & 0 & L & 0 & x_{p-1,1} & 0 \\ 0 & x_{p,p} & 0 & L & 0 & x_{p,1} & 0 \end{pmatrix}$$

Namely:

$$\frac{r^2}{1+2r} \|A_p^{-1}(M + N)\|_{\infty} = \frac{r^2}{1+2r} \max_i (x_{i,1} + x_{i,p}) \text{ for } i = 1, 2, \dots, p \tag{11}$$

If denoting $f(i,p) = x_{i,1} + x_{i,p}$ and $x_j = (x_{1,j}, x_{2,j}, \dots, x_{p,j})^T$ for $j = 1, 2, \dots, p$, the result of (11) can be obtained by solving the first and last linear system of:

$$A_p X_1 = e_1, \quad A_p X_p = e_p$$

By Grammer's rule, it is easy to get the two vectors X_1 and X_p as:

$$x_{i,1} = r^{i-1} \bar{D}_{p-i} / D_p, \quad x_{i,p} = r^{p-i} \bar{D}_{i-1} / D_p \text{ for } i = 1, 2, \dots, p$$

Where:

$$D_1 = \det A_1,$$

$$\bar{D}_i = \begin{vmatrix} 1+2r & -r & & & & \\ -r & 1+2r & -r & & & \\ & & \ddots & \ddots & \ddots & \\ & & & -r & 1+2r & -r \\ & & & & -r & 1+2r - \frac{r^2}{1+2r} \end{vmatrix}_{i \times i}$$

$$\bar{D}_i = \begin{vmatrix} 1+2r - \frac{r^2}{1+2r} & -r & & & & \\ -r & 1+2r & -r & & & \\ & & \ddots & \ddots & \ddots & \\ & & & -r & 1+2r & -r \\ & & & & -r & 1+2r \end{vmatrix}_{i \times i}$$

Then:

$$f(i,p) = (r^{i-1} \bar{D}_{p-i} + r^{p-i} \bar{D}_{i-1}) / D_p$$

Since, $\bar{\mathbf{D}}_i = (1 + 2r)\bar{\mathbf{D}}_{i-1} - r^2\bar{\mathbf{D}}_{i-2}$ for $0 \leq i \leq p$ and denoting $\alpha + \beta = 1 + 2r, \alpha\beta = r^2$ and $\bar{\mathbf{D}}_0 = 1$, then we have $\bar{\mathbf{D}}_i = (\alpha^{i+2} - \beta^{i+2}) / (\alpha^2 - \beta^2)$

Similar to the method above, $\bar{\bar{\mathbf{D}}}_i$ is equal to $\bar{\mathbf{D}}_i$. So we have:

$$\mathbf{D}_1 = (1 + 2r - \frac{r^2}{1+2r})\bar{\mathbf{D}}_{1-1} - r^2\bar{\mathbf{D}}_{1-2} = (\alpha^{1+3} - \beta^{1+3}) / (\alpha^2 - \beta^2) / (\alpha + \beta)$$

Therefore:

$$f(i, p) = [r^{i-1}(\alpha^{p-i+2} - \beta^{p-i+2}) + r^{p-i}(\alpha^{i+1} - \beta^{i+1})] / (\alpha + \beta) / (\alpha^{p+3} - \beta^{p+3})$$

Taking i as continuous and differentiable in $f(i, p)$, we have:

$$f'(i, p) = [(\frac{\alpha}{r})^{i+1} + (\frac{\beta}{r})^{i+1} - (\frac{\alpha}{r})^{p-i+2} - (\frac{\beta}{r})^{p-i+2}] r^{p+1} (\alpha + \beta) \ln \frac{\alpha}{r} / (\alpha^{p+3} - \beta^{p+3})$$

Assuming $\alpha = r + (1 + \sqrt{1 + 4r}) / 2$ which can be calculated by $\alpha + \beta = 1 + 2r$ and $\alpha\beta = r^2$, it is not difficult to find that:

$$f'_i(i, p) = \begin{cases} < 0, (i < \frac{p+1}{2}) \\ = 0, (i = \frac{p+1}{2}) \\ > 0, (i > \frac{p+1}{2}) \end{cases}$$

Therefore, $f(i, p)$ is strictly decreasing when $i < \frac{p+1}{2}$ and strictly increasing when $i > \frac{p+1}{2}$ for variable i . So $f(i, p)$ reaches its maximum at $i = 1, p$ and can be rewritten as:

$$f(p) = \max_i f(i, p)$$

Supposing p is a continuous and differentiable again, we have:

$$f'(p) = [(\frac{\alpha}{r})^{p+3} + (\frac{\beta}{r})^{p+3} - 2(\frac{\alpha}{r})^{\frac{p+3}{2}} (\frac{\beta}{r})^{\frac{p+3}{2}}] r^{p+2} (\alpha + \beta)^2 (\alpha - \beta) \ln \frac{\alpha}{\alpha} < 0$$

Hence, $f(p)$ is strictly decreasing for variable p according to the above discussion. That is to say, $\|A_p^{-1}(M_p + N_p)\|_\infty$ decreases with the increasing of net point number p in each segment. We also have:

$$\|M^{-1}N\|_V = \frac{r^2}{1+2r} f(p) = [r^2 - r^2 x^{-2p-2} + (\alpha^2 - \beta^2) x^{-p-1}] / (\alpha^2 - \beta^2 x^{-2p-2}) \xrightarrow{p \rightarrow +\infty} (\frac{r}{\alpha})^2 \quad (12)$$

In the following section, $\|A_1^{-1}N_p\|_\infty$ and $\|A_k^{-1}M_p\|_\infty$ will be estimated.

Since $A_1^{-1} > 0, N \geq 0$, and supposing $Y = (y_{i,j})_{p \times p} = A_1^{-1}$, we have:

$$A_1^{-1}N = YN = \begin{pmatrix} 0 & y_{1,p} & 0 & \dots & 0 \\ 0 & y_{2,p} & 0 & & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & y_{p-1,p} & 0 & \dots & 0 \\ 0 & y_{p,p} & 0 & \dots & 0 \end{pmatrix}$$

Namely:

$$\|A_1^{-1}N\|_\infty = \max_i (y_{i,p}) \text{ for } i = 1, 2, \dots, p$$

Let $f(i, p) = y_{i,p}$ and $Y_j = (y_{1,j}, y_{2,j}, \dots, y_{p,j})^T$ for $i = 1, 2, \dots, p$. We have:

$$A_1 Y_p = e_p$$

And:

$$y_{i,p} = r^{p-i} G_{i-1} / \bar{D}_p \text{ for } i = 1, 2, \dots, p$$

Where:

$$G_i = \begin{vmatrix} 1+2r & -r & & & \\ -r & 1+2r & -r & & \\ & & \ddots & \ddots & \ddots \\ & & & -r & 1+2r & -r \\ & & & & -r & 1+2r \end{vmatrix}_{i \times i}$$

Then:

$$f(i, p) = r^{p-i} (\alpha^i - \beta^i) / (\alpha - \beta) / \bar{D}_p$$

Taking i as continuous in form, we have:

$$f'(i, p) = [(\frac{\alpha}{r})^i + (\frac{\beta}{r})^i] r^p \ln \frac{\alpha}{r} / (\alpha - \beta) / \bar{D}_p > 0$$

Therefore, $f(i, p)$ is strictly increasing about i in area $[1, p]$ and so we have:

$$f(p) = \max_i f(i, p) = G_{p-1} / \bar{D}_p = (1 + 2r)(\alpha^p - \beta^p) / (\alpha^{p+2} - \beta^{p+2})$$

Again differentiating function $f(p)$ for variable p , then:

$$f'(p) = 2(1 + 2r)r^{2p}(\alpha^2 - \beta^2) \ln \frac{\alpha}{r} / (\alpha^{p+2} - \beta^{p+2})^2 > 0$$

So $f(p)$ is strictly increasing and we have:

$$\frac{r^2}{1+2r} \|A_1^{-1}N\|_\infty = r^2 (1 - x^{-2p} / (\alpha^2 - \beta^2 x^{-2p})) \xrightarrow{p \rightarrow +\infty} (\frac{r}{\alpha})^2 \quad (13)$$

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \quad (21)$$

with the boundary conditions:

$$\begin{aligned} u(0, y, t) &= f_1(y, t), \quad u(1, y, t) = f_2(y, t) \\ u(x, 0, t) &= f_3(x, t), \quad u(x, 1, t) = f_4(x, t) \end{aligned} \quad (22)$$

and the initial condition:

$$u(x, y, 0) = f(x, y) \quad (23)$$

Let Δx , Δy and Δt be the grid spaces in the x, y and t directions; where $\Delta x = 1/m$ and $\Delta y = 1/n$ in which m and n are positive integers. The approximate value $u_{i,j}^k$ of the solution $u(x,y,t)$ for the problems above are to be computed at the grid points (x_i, y_j, t_k) , where :

$x_i = i\Delta x$ for $i = 0, 1, \dots, m$, $y_j = j\Delta y$ for $j = 0, 1, \dots, n$ and $t_k = k\Delta t$ for $k = 0, 1, \dots$. For simplicity, we take $\Delta x = \Delta y = h$ and $\Delta t = \tau$, where $h = 1/m$. And sometimes denote points (I, j, k) by (x_i, y_j, t_k) .

Among the finite difference method for the numerical solution above, the well-known Peaceman-Rachford scheme is unconditionally stable and has truncation error $O(\Delta t^2 + \Delta x^2)$ as follows:

$$\begin{cases} -ru_{i+1,j}^{k+\frac{1}{2}} + (1+2r)u_{i,j}^{k+\frac{1}{2}} - ru_{i-1,j}^{k+\frac{1}{2}} \\ = u_{i,j}^k + r(u_{i,j+1}^k - 2u_{i,j}^k + u_{i,j-1}^k) \\ -ru_{i,j+1}^{k+1} + (1+2r)u_{i,j}^{k+1} - ru_{i,j-1}^{k+1} \\ = u_{i,j}^{k+\frac{1}{2}} + r(u_{i+1,j}^{k+\frac{1}{2}} - 2u_{i,j}^{k+\frac{1}{2}} + u_{i-1,j}^{k+\frac{1}{2}}) \end{cases} \quad (24)$$

with the boundary conditions:

$$\begin{aligned} u_{0,j}^k &= f_1(j, k), \quad u_{1,j}^k = f_2(j, k) \\ u_{i,0}^k &= f_3(i, k), \quad u_{i,1}^k = f_4(i, k) \end{aligned} \quad (25)$$

for $I, j = 0, 1 \dots m$ and the initial condition:

$$u_{i,j}^k = f(i, j) \quad \text{for } I, j = 0, 1, \dots, m \quad (26)$$

where, $\tau/h^2 = r$.

So the difference Eq, 24-26 for the Eq. 21-23 consist of solving $(m-1) \times (m-1)$ system, which can be written in the matrix form as:

$$\begin{cases} AU_j^{k+\frac{1}{2}} = b_1 \\ AU_i^{k+1} = b_2 \end{cases} \quad (27)$$

where, A is defined as:

$$\mathbf{A} = \begin{pmatrix} 1+2r & -r & & & 0 \\ -r & 1+2r & -r & & \\ & \ddots & \ddots & \ddots & \\ & & -r & 1+2r & -r \\ 0 & & & -r & 1+2r \end{pmatrix}$$

$$U_j^k = (u_{1,j}^k, u_{2,j}^k, \dots, u_{m-1,j}^k)^T \quad \text{for } j = 1, 2, \dots, m-1$$

$$U_i^k = (u_{i,1}^k, u_{i,2}^k, \dots, u_{i,m-1}^k)^T \quad \text{for } i = 1, 2, \dots, m-1$$

It is well known that Peaceman-Rachford schemes can be divided into two processes in alternate directions. So parallel iterative algorithm with accelerated convergence of (21)-(23) can be expressed as:

$$\begin{cases} \mathbf{M}U_j^{[k+\frac{1}{2}](s+1)} = \mathbf{N}U_j^{[k+\frac{1}{2}](s)} + \mathbf{b}_1 \\ \mathbf{M}U_i^{[k+1](s+1)} = \mathbf{N}U_i^{[k+1](s)} + \mathbf{b}_2 \end{cases} \quad (28)$$

where, M and N are defined as (11). By a similar method in one-dimensional problems, the convergence rate and property of gradual approach convergence are estimated as follows:

$$\begin{cases} \left\| \mathbf{M}_i^{-1} \mathbf{N}_i \right\|_{\infty}^{[k+\frac{1}{2}]} \leq \frac{r^2}{1+3r+r^2} \quad \text{for } 1 < mi < m-1 \\ \left\| \mathbf{M}_i^{-1} \mathbf{N}_i \right\|_{\infty}^{[k+1]} \leq \frac{r^2}{1+3r+r^2} \end{cases} \quad (29)$$

$$\begin{cases} \left\| \mathbf{M}^{-1} \mathbf{N} \right\|_{\infty}^{[k+\frac{1}{2}]} \xrightarrow{p \rightarrow \infty} \left(\frac{r}{\alpha}\right)^2 \\ \left\| \mathbf{M}^{-1} \mathbf{N} \right\|_{\infty}^{[k+1]} \xrightarrow{p \rightarrow \infty} \left(\frac{r}{\alpha}\right)^2 \end{cases} \quad \text{for}$$

$$\forall m_i = p \quad \text{and} \quad \alpha = r + \frac{1+\sqrt{1+4r}}{2} \quad (30)$$

It is not difficult to see that parallel iterative algorithm with accelerate convergence (28) for two-dimensional problems are convergent. In addition, it has a better convergent rate and property of gradual convergence than those of parallel iterative algorithm about two-dimensional diffusion problem^[5]. Thus, the parallel iterative method with accelerated convergence for one-dimensional problem is extended to the computation of multi-dimensional problem.

RESULTS

Numerical examples and numerical results:

One-dimension example: Consequently, numerical experiments are made for problems of (1)-(3) in which:

$$f(x) = 4x(1-x), \quad g_0(t) = g_1(t) = 0$$

The exact solution of the problem is:

$$U(x, t) = 32/\pi^3 \cdot \sum_{k=1,3,5,\dots}^{\infty} e^{-k^2\pi^2 t} \sin(k\pi x)/k^3$$

Table 1: Numerical results of SCII algorithm with Accelerated Convergence (SCIIAC) and SCNI algorithm with Accelerated Convergence (SCNIAC)

r = 10, Δ = 10, Δx = 0.01, k = 200, error = 10 ⁻⁸							
x _i	U(10 ⁻¹)	U _i ^k (10 ⁻¹)		AE(10 ⁻⁴)		PE(10 ⁻⁴)	
		SCIIAC	SCNIAC	SCIIAC	SCNIAC	SCIIAC	SCNIAC
0.1	2.67084	2.67218	2.67170	1.34212	0.86076	5.02508	3.22281
0.2	5.04109	5.04260	5.04178	1.50410	0.69047	2.98368	1.36969
0.3	6.87319	6.87433	6.87362	1.13072	0.42760	1.64511	0.62213
0.4	8.01911	8.01978	8.01933	0.67003	0.22392	0.83554	0.27924
0.5	8.40767	8.40816	8.40783	0.48855	0.16352	0.58108	0.19449
0.6	8.01911	8.01978	8.01933	0.67003	0.22392	0.83554	0.27924
0.7	6.87319	6.87433	6.87362	1.13072	0.42760	1.64511	0.62213
0.8	5.04109	5.04260	5.04178	1.50410	0.69047	2.98368	1.36969
0.9	2.67084	2.67218	2.67170	1.34212	0.86076	5.02508	3.22281

Table 2: Iterative degrees of SCII algorithm and SCII algorithm with accelerated convergence (SCIIAC)

r = 10, m = 100, k = 200, error = 10 ⁻⁸ , is the iterative degree						
p	3	5	10	20	25	50
S _{SCII}	14	11	8	4	2	1
S _{SCIIAC}	7	5	4	2	1	0

Table 3: Numerical results of two-dimensional parallel iterative algorithm with accelerated convergence

r = 1.0, Δx = Δy = Δz = 0.01, x = 0.5, k = 1000 error = 10 ⁻⁸									
y	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
U(10 ⁻¹)	0.40824	0.77653	1.06881	1.25646	1.32112	1.25646	1.06881	0.77653	0.40824
U _{ij} ^k (10 ⁻¹)	0.40810	0.77625	1.06842	1.25601	1.32064	1.25601	1.06842	0.77625	0.40810
AE(10 ⁻⁴)	1.46697	2.79034	3.84058	4.51488	4.74722	4.51488	3.84058	2.79034	1.46697
PE(10 ⁻³)	3.59332	3.59332	3.59332	3.59332	3.59332	3.59332	3.59332	3.59332	3.59332

The numerical results are shown in Table 1 by putting the Absolute Error (AE):

$$|AE = e_i^k = u_i^k - U_i^k|$$

and Percentage Error (PE):

$$PE = R_i^k = e_i^k / U_i^k \times 100\%$$

at each point along the mesh line, where u and U are numerical solutions and exact solutions respectively. The numerical results of SCII and SC-NI algorithms with accelerated convergence obtained in SGL/Challenge L with 8 CPUs for these methods are more accurate in computing in parallel. It is shown in Table 2 that the iterative degree of SCII algorithm with accelerated convergence is less than that of SCII algorithm and it decreases with the increasing of net point number in each segment (Table 3).

Two-dimension example: In closing, we give the numerical experiment made for the problem (21-23), in which:

$$f(x, y) = \sin \pi x \sin \pi y$$

$$f_1(y, t) = f_2(y, t) = 0, \quad 0 \leq y \leq 1, 0 \leq t \leq T$$

$$f_3(x, t) = f_4(x, t) = 0, \quad 0 \leq x \leq 1, 0 \leq t \leq T$$

The exact solution of the problem is:

$$U(x, y, t) = e^{-2\pi^2 t} \sin \pi x \sin \pi y$$

CONCLUSION

By reconstructing differential equations, the SCII algorithm with accelerated convergence for solving one-dimensional diffusion equation is developed in this study. It is convergent in iteration and has a better convergent rate and property of gradual approach convergence than those of SCII algorithm. Furthermore, the SC-NI algorithm with accelerated convergence is discussed. The convergent rate, property of gradual-approach convergence and accuracy are much better than those of SCII algorithm with accelerated convergence. In addition, the algorithm is tended to two-space dimensional problem by studying Peaceman-Rachford schemes.

REFERENCES

1. Chen, J. and B.L. Zhang, 1992. A class of alternation block cranks-Nicolson method [J]. Int. J. Comput. Math, 45: 89-112.
2. Chen, J. and B.L. Zhang, 1994. Massively parallel algorithm for parabolic equation: one dimensional problem [J]. Computat. Phys. Sinica, 4: 372-382.

3. Evans, D.J. and W.S. Yousif, 1990. The solution of unsymmetric tridiagonal Toeplitz systems of the stride reduction algorithm [J]. *Parallel Comput.*, 20: 787-798
4. Hu, J.G., 1984. The Estimates of $\|M^{-1}N\|_{\infty}$ and the optimally scaled matrix [J]. *J. Computat. Math.*, 2: 122-129.
5. Liu, Q.F., 2002. Parallel iterative algorithms for solving the implicit difference equations [J]. *Guizhou Sci.*, 6: 29-36.
6. Zhang, B.L., 1991. An alternation segment explicit-implicit method for the diffusion equation [J]. *J. Numeric. Method. Comput. Appl.*, 4: 245-253.
7. Zhang, B.L., 1996. One finite difference methods for solving initial-boundary value problems in parallel. *Comm. Appl. Math. Comput.*, 6: 53-65.
8. Zhang, B.L., T.X. Gu and Z.Y. Mo, 1999, *Principles and Methods of Numerical Parallel Computation* [M]. Beijing: National Defense Industry Press, pp: 88-90
9. Zhang, B.L. and X.M. Su, 1991. Alternating block explicit-implicit method for two-dimensional diffusion equation [J]. *Int. J. Comput. Math.*, 38: 241-255
10. Zhang, B.L. and X.M. Su, 1992, Parallel algorithms for solving the implicit difference equations [J]. *Computat. Phys. Sinica*, 9: 250-256.