JOURNAL OF COMPUTATIONAL PHYSICS **88**, 362–380 (1990)

Multigrid and Adaptive Algorithm for Solving the Nonlinear Schrödinger Equation

QIANSHUN CHANG AND GUOBIN WANG

Institute of Applied Mathematics, Academia Sinica, Beijing, China

Received January 18, 1989; revised July 18, 1989

In this paper, a conservative difference scheme for generalized nonlinear Schrödinger equations is given. We apply multigrid method and adaptive algorithm to solve the equations. Numerical results are presented and compared. They demonstrate that the multigrid and adaptive algorithm are efficient and can considerably relax the restrict on step size of time, which is caused by nonlinear iteration. © 1990 Academic Press, Inc.

1. INTRODUCTION

In order to ensure computational stability, we often employ unconditionally stable implicit schemes for nonlinear differential equations. The schemes are nonlinear algebraic equations, which always are solved by means of iterative algorithms. The nonlinear iterative algorithms require more computing time and their iterative convergence depends on step size of time. Thus, the step size of time is restricted, though unconditionally stable schemes are used.

Multigrid method can efficiently solve the algebraic equations arising in discretizing boundary-value problem and enormously reduce the amount of computational work. Adaptive algorithm is useful for problems in which different scales of discretization are needed in different parts of the domain [3, 4].

In this paper, we consider application of multigrid and adaptive algorithm to nonlinear Schrödinger (NLS) equation. Conservative difference scheme for the NLS equation has been given in [1, 2]. This is a nonlinear algebraic equations. By means of theoretical analysis and test computation, a multigrid procedure for solving the NLS equation is presented. The NLS equation possesses soliton solution, which is located at a small region. Therefore, the adaptive algorithm can be efficiently employed in solving the NLS equation. In view of Brandt's idea [3, 6], we deduce a formula on relation between truncation error τ^K and quantity τ_{K+1}^K, which is computed in the multigrid procedure. The quantity τ_{K+1}^K is used in grid adaptiation.

Numerical results of applying the multigrid and adaptive algorithm are given and compared with ones of iterative method. In previous papers, it has been discussed that the multigrid method can decrease the amount of computational work and

0021-9991/90 $3.00

save CPU time. But, our computational results demonstrate the iteration of the multigrid method also can relax the restrict on the step size of time, which is given by nonlinear iteration.

2. Differential Equation and Difference Scheme

We consider the following initial-boundary value problem of generalized NLS equation

$$i\mathbf{u}_t - \frac{\partial}{\partial x} A(x) \frac{\partial \mathbf{u}}{\partial x} + \beta(x)\, q(|\mathbf{u}|^2)\mathbf{u} + F(x, t)\mathbf{u}$$

$$= \mathbf{G}(x, t), \qquad t > 0,\ x_L < x < x_R, \tag{2.1}$$

$$\mathbf{u}|_{x=x_L} = 0, \qquad \mathbf{u}|_{x=x_R} = 0, \qquad t > 0 \tag{2.2}$$

$$\mathbf{u}|_{t=0} = \mathbf{u}_0(x), \qquad x_L < x < x_R, \tag{2.3}$$

where $\mathbf{u}(x, t)$ is an unknown complex functional vector, $A(x) = (a_m(x))_{M\times M}$ is a real diagonal matrix, $F = (f_{l,m}(x, t))_{M\times M}$ is a symmetrical real matrix, $\beta(x)$ and $q(s)$ are real functions, $\mathbf{u}_0(x)$ and $\mathbf{G}(x, t)$ are complex vectors.

It is easy to obtain two conservation laws of the problem (2.1)–(2.3), namely,

$$\|\mathbf{u}(x, t)\|^2_{L_2} = \|\mathbf{u}(x, 0)\|^2_{L_2} + 2\int_0^t \operatorname{Im}(\mathbf{G}(x, \tau), \mathbf{u}(x, \tau))\, d\tau, \tag{2.4}$$

and

$$\left(A(x)\frac{\partial \mathbf{u}(x, t)}{\partial x}, \frac{\partial \mathbf{u}(x, t)}{\partial x}\right) + (\beta(x), Q(|\mathbf{u}(x, t)|^2))$$

$$= \left(A(x)\frac{\partial \mathbf{u}(x, 0)}{\partial x}, \frac{\partial \mathbf{u}(x, 0)}{\partial x}\right) + (\beta(x), Q(|\mathbf{u}(x, 0)|^2))$$

$$- \sum_{l=1}^{M}\sum_{m=1}^{M}\int_0^t \left(f_{lm}(x, \tau), \operatorname{Re}\left[\frac{\partial}{\partial \tau}(u_l(x, \tau)\cdot \overline{u_m(x, \tau)})\right]\right) d\tau$$

$$+ 2\operatorname{Re}\int_0^\tau \left(\mathbf{G}(x, \tau), \frac{\partial \mathbf{u}(x, \tau)}{\partial \tau}\right) d\tau, \tag{2.5}$$

where inner product $(\mathbf{f}(x, t), \mathbf{g}(x, t)) = \sum_{m=1}^{M} \int_{x_L}^{x_R} f_m(x, t)\cdot \overline{g_m(x, t)}\, dx$ and $Q(s) = \int_0^s q(z)\, dz$.

The problem (2.1)–(2.3) can be approximated by conservation difference scheme

$$i(u^{n+1}_{m,j})_t - \frac{1}{2}\{a_{m,j+1/2}[(u^{n+1}_{m,j})_x + (u^n_{m,j}]\}_{\bar{x}}$$

$$+ \frac{\beta_j}{2}\frac{Q(|\mathbf{u}^{n+1}_j|^2) - Q(|\mathbf{u}^n_j|^2)}{|\mathbf{u}^{n+1}_j|^2 - |\mathbf{u}^n_j|^2}(u^{n+1}_{m,j} + u^n_{m,j}) + \frac{1}{2}\sum_{l=1}^{M} f^{n+1/2}_{m,l,j}(u^{n+1}_{l,j} + u^n_{l,j})$$

$$= G^{n+1/2}_{m,j}, \qquad 1 \leqslant m \leqslant M,\ 1 \leqslant j \leqslant J-1,\ n = 0, 1, ..., \tag{2.6}$$

$$u^n_{m,0} = u^n_{m,J} = 0, \qquad 1 \leqslant m \leqslant M, \quad n = 0, 1, ..., \tag{2.7}$$

$$u^0_{m,j} = u_{0,m}(x_j), \qquad 1 \leqslant m \leqslant M, \quad 1 \leqslant j \leqslant J-1, \tag{2.8}$$

where

$$(f^n_j)_{\bar{x}} \equiv \frac{f^n_j - f^n_{j-1}}{h}, \qquad (f^n_j)_x = \frac{f^n_{j+1} - f^n_j}{h};$$

$h = (x_R - x_L)/J$ and τ are the step sizes of space and time, respectively.

The following theorems have been proved in [2, Theorems 1–4].

THEOREM 1. *Suppose that* $|f_{m,l}(x, t)| \leqslant c$,

$$\left|\frac{\partial f_{m,l}(x, t)}{\partial t}\right| \leqslant c, \qquad \left\|\frac{\partial G_m(x, t)}{\partial t}\right\|_{L_2} \leqslant c,$$

$u_{0,m}(x) \in H^1_0[x_L, x_R]$, $1 \leqslant m, l \leqslant M$, $q(s) \in C^1[0, \infty)$, *and assume that one of the following conditions are satisfied*

(i) $0 < c_1 \leqslant a_m(x) \leqslant c$, $0 \leqslant \beta(x) \leqslant c$, $Q(s) \geqslant 0$, $s \in [0, \infty)$;

(ii) $0 < c_1 \leqslant a_m(x) \leqslant c$ *or* $0 < c_0 \leqslant -a_m(x) \leqslant c$, $|\beta(x)| \leqslant c$, $|q'(s)| \leqslant c$, $s \in [0, \infty)$;

(iii) $0 < c_1 \leqslant a_m(x) \leqslant c$ *or* $0 < c_0 \leqslant -a_m(x) \leqslant c$, $|\beta(x)| \leqslant c$, $q(s) = s^p$, $0 \leqslant p < 2$,

where c_1 *and* c *are positive constants. Then the scheme* (2.6)–(2.8) *is stable in* L_2 *norm for initial values.*

THEOREM 2. *Suppose that the conditions of Theorem* 1 *are satisfied, and assume that for the solution of problem* (2.1)–(2.3), $u(x, t) \in c^{(4,3)}$, $a_m(x) \in c^3$. *Then the solution* $\mathbf{u}_h$ *of the difference problem* (2.6)–(2.8) *converges to the solution* $\mathbf{u}$ *of the problem* (2.1)–(2.3) *in* L_2 *norm and* $\|\mathbf{u} - \mathbf{u}_h\|_{L_2} = O(\tau^2 + h^2)$.

THEOREM 3. *Assume the conditions of Theorem* 1 *are satisfied, and* $u_{0,m}(x) \in H^2$. *Then there exists the generalized solution of the problem* (2.1)–(2.3) *and it is unique.*

If the condition

(iv) $0 < c_0 \leqslant -a_m(x) \leqslant c$, $0 \leqslant -\beta(x) \leqslant c$, $Q(s) \geqslant 0$, $s \in [0, \infty)$,

is satisfied instead of conditions (i), (ii), (iii) in Theorems 1, 2, and 3, then these theorems can still be obtained by means of the proof idea given in [2].

The scheme (2.6)–(2.8) possesses discrete conservation laws:

$$h \sum_{m=1}^{M} \sum_{j=1}^{J-1} |u^{n+1}_{m,j}|^2 = h \sum_{m=1}^{M} \sum_{j=1}^{J-1} |u^0_{m,j}|^2 + h \sum_{k=0}^{n} \sum_{m=1}^{M} \sum_{j=1}^{s-1} \operatorname{Im}[G^{k+1/2}_{m,j}(\bar{u}^{k+1}_{m,j} + \bar{u}^k_{m,j})] \tag{2.9}$$

and

$$h \sum_{m=1}^{M} \sum_{j=0}^{J-1} a_{m,j+1/2} |(u_{m,j}^{n+1})_x|^2 + h \sum_{j=1}^{J-1} \beta_j Q_j^{n+1}$$
$$= h \sum_{m=1}^{M} \sum_{j=0}^{J-1} a_{m,j+1/2} |(u_{m,j}^{0})_x|^2 + h \sum_{j=1}^{J-1} \beta_j Q_j^{0}$$
$$- h \sum_{k=0}^{n} \sum_{m=1}^{M} \sum_{l=1}^{M} \sum_{j=1}^{J-1} f_{m,l,j}^{k+1/2} \cdot \operatorname{Re}(u_{l,j}^{k+1} \cdot \bar{u}_{m,j}^{k+1})_{\bar{t}}$$
$$+ 2 \cdot \operatorname{Re} \sum_{k=0}^{n} \sum_{m=1}^{M} \sum_{j=1}^{J-1} G_{m,j}^{k+1/2} (\bar{u}_{m,j}^{k+1})_{\bar{t}}. \qquad (2.10)$$

Comparing (2.9) and (2.10) with (2.4) and (2.5), we know that the difference scheme (2.6)–(2.8) keeps two conservation laws that the differential problem (2.1)–(2.3) possesses.

Nonlinear iterative algorithms for the scheme (2.6)–(2.8) are discussed in [1, 2]. It has been proved that when $\tau \leqslant \text{const} \cdot h^{\alpha}$, $\alpha = 1$ or 2, the iterative algorithms are convergent.

3. Application of the Multigrid Method

We shall now consider the application of the multigrid method to an initial-boundary value problem of basic NLS equation

$$iu_t + u_{xx} + 2|u|^2 u = 0, \qquad t > 0, \quad x_L < x \leqslant x_R, \qquad (3.1)$$

$$u|_{x=x_L}, \qquad u|_{x=x_R} = 0, \qquad t > 0, \qquad (3.2)$$

$$u|_{t=0} = u_0(x), \qquad x_L < x < x_R, \qquad (3.3)$$

The conservation difference scheme that approximates (3.1)–(3.3) may be written as

$$Au_{j+1}^{n+1} + B_j^{n+1} u_j^{n+1} + Au_{j-1}^{n+1} - F_j^{n+1} = 0, \quad 1 \leqslant j \leqslant J-1, \; n = 0, 1, ..., \qquad (3.4)$$

$$u_0^n = u_J^n = 0, \qquad n = 0, 1, ..., \qquad (3.5)$$

$$u_j^0 = u_0(x_j), \qquad 1 \leqslant j \leqslant J-1. \qquad (3.6)$$

where

$$A = \frac{\tau}{2h^2},$$

$$B_j^{n+1} = i - \frac{\tau}{h^2} + \frac{\tau}{2} (|u_j^{n+1}|^2 + |u_j^n|^2),$$

$$F_j^{n+1} = iu_j^n - \frac{\tau}{2h^2} (u_{j+1}^n - 2u_j^n + u_{j-1}^n) - \frac{\tau}{2} (|u_j^{n+1}|^2 + |u_j^n|^2) u_j^n. \qquad (3.7)$$

They are obviously nonlinear algebraic equations and are solved by full-approximation scheme (FAS) of the multigrid method [3, 6].

Now, we consider a brief description of the FAS mode applied to the equation

$$LU = G, \qquad \text{in } \Omega.$$

Its approximation solution u^M on the finest grid Ω^M satisfies

$$L^M U^M = G^M, \qquad \text{in } \Omega^M. \tag{3.8}$$

Assume that the grid Ω^{K+1} is finer than the grid Ω^K and the ratio of their step size is $h_{K+1}/h_K = 1/2$. In the FAS mode, the U^K satisfies the modified equation

$$L^K U^K = \bar{G}^K, \qquad \text{in } \Omega^K, \tag{3.9}$$

where

$$\bar{G}^K = L^K(I^K_{K+1} U^{K+1}) + I^K_{K+1}(\bar{G}^{K+1} - L^{K+1}U^{K+1}),, \qquad K = 0, 1, ..., M-1,$$
$$\bar{G}^M = G^M.$$

Let

$$\tau^K_{K+1} = L^K(I^K_{K+1} U^{K+1}) - I^K_{K+1}(L^{K+1} U^{K+1}); \tag{3.10}$$

then

$$\bar{G}^K = I^K_{K+1} \bar{G}^{K+1} + \tau^K_{K+1}, \qquad K = 0, 1, ..., M-1,$$
$$\bar{G}^M = G^M,$$

where I^K_{K+1} and I^{K+1}_K denote restriction and interpolation operators, respectively.

In interpolating correction to the finer grid, the formula is

$$U^{K+1(\text{new})} = U^{K+1(\text{old})} + I^{K+1}_K(U^K - I^K_{K+1} U^{K+1(\text{old})}). \tag{3.11}$$

A flow chart of FAS mode is given in Fig. 1.

In view of theoretical analysis and numerical computation, we choose that the restriction operator is

$$U^K_j = (I^K_{K+1} U^{K+1})_j = U^{K+1}_{2j} \tag{3.12}$$

and interpolation operators are written as

$$U^{K+1}_{2j} = (I^{K+1}_K U^K)_j = U^K_j, \tag{3.13}$$

$$U^{K+1}_{2j+1} = (I^{K+1}_K U^K)_{2j+1} = \frac{25}{128}(U^K_j + U^K_{j+1}) - \frac{25}{256}(U^K_{j-1} + U^K_{j+2})$$
$$+ \frac{3}{256}(U^K_{j-2} + U^K_{j+3}), \qquad 2 \leqslant j \leqslant J-3, \tag{3.14}$$

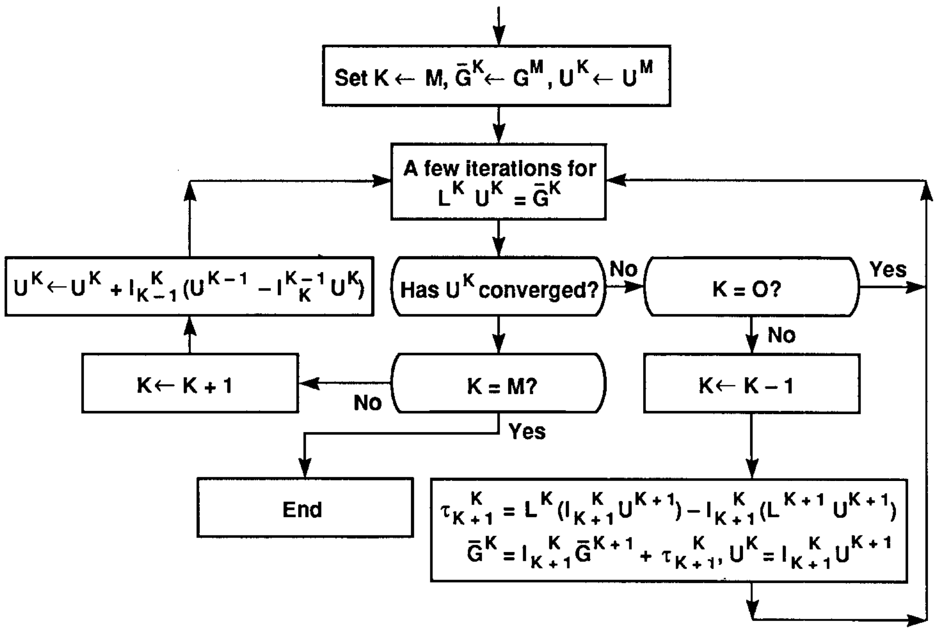

FIG. 1. Flow chart for FAS mode.

The following interpolation formulas are used near the left boundary:

$$U_1^{K+1} = \tfrac{5}{16}U_0^K + \tfrac{15}{16}U_1^K - \tfrac{5}{16}U_2^K + \tfrac{1}{16}U_3^K, \tag{3.15}$$

$$U_3^{K+1} = \tfrac{9}{16}(U_1^K + U_2^K) - \tfrac{1}{16}(U_0^K + U_3^K). \tag{3.16}$$

The interpolation formulas near the right boundary are similar to (3.15) and (3.16).

Considering that τ is a small quantity and $|B_j^{n+1}| \geqslant 2A$, a Seidel-type iterative formula is employed

$$u_j^{n+1(s+1)} = \frac{F_j^{n+1(s)} - Au_{j+1}^{n+1(s)} - Au_{j-1}^{n+1(s+1)}}{B_j^{n+1(s)}}, \qquad u_j^{n+1(0)} = u_j^n, \tag{3.17}$$

By means of linearization of Eq. (3.17), a convergence factor is approximately obtained

$$\mu(\theta) = \left| \frac{Ae^{i\theta}}{B + Ae^{-i\theta}} \right|,$$

where $B \approx B_j^{n+1(s)} = i - \tau/h^2$. Let $\alpha = 2h^2/\tau$, we have

$$\mu(\theta) = \frac{1}{\sqrt{5 + \alpha^2 - 2\alpha \sin\theta - 4\cos\theta}}$$

and smoothing factor

$$\bar{\mu} = \max_{\pi/2 \leqslant \theta \leqslant \pi} \mu(\theta) = \frac{1}{\sqrt{4 + (\alpha - 1)^2}}.$$

Its values are given in Table I.

The multigrid method is applied to the problem (3.1)–(3.3) subject to the following conditions:

(i) One-soliton solution

$$u_0(x) = \text{Sech}(x+10) \cdot \exp(2i(x+10)), \quad x_L = -16, \quad x_R = 16; \tag{3.18}$$

(ii) Collision of two solitons, which move in the opposite direction

$$u_0(x) = \text{Sech}(x+10) \cdot \exp(2i(x+10) + \text{Sech}(x-10) \cdot \exp(-2i(x-10)), \quad x_L = -16, \quad x_R = 16; \tag{3.19}$$

(iii) Collision of two solitons, which move in the same direction

$$u_0(x) = \text{Sech}\left(\frac{\sqrt{2}}{2} x\right) \cdot \exp\left(\frac{1}{2} ix\right) + \text{Sech}\left(\frac{\sqrt{2}}{2}(x-25)\right) \cdot \exp\left(\frac{1}{20} i(x-25)\right), \quad x_L = -20, \quad x_R = 80. \tag{3.20}$$

In computation, we use the conservation scheme (3.4)–(3.6) and require that the iterative error $E \equiv \max_j |u_j^{n+1(s+1)} - u_j^{n+1(s)}| \leqslant 10^{-5}$ in every step of time. The soliton solutions are computed from $t = 0$ to $t = T$ on the Micro Vax II computer. We take that $h = 0.1$, $\tau = 0.01$, 0.02, 0.05, 0.1, 0.2, and 0.5, $T = 5$ for the case (i); $h = 0.1$, $\tau = 0.01$, 0.02, 0.05, 0.1, and 0.2, $T = 5$ for the case (ii); and $h = 0.25$, $\tau = 0.0625$, 0.125, and 0.25, $T = 45$ for the case (iii), respectively. The computational results are given in the Tables II–IV.

TABLE I

Values of the Smoothing Factor

τ	h	α	$\bar{\mu}$
0.01	0.1	2.0	0.4472
0.02	0.1	1.0	0.5000
0.05	0.1	0.4	0.4789
0.1	0.1	0.2	0.4642

TABLE II

Numerical Results for One-Soliton Solution

Method	τ	0.01	0.02	0.05	0.1	0.2	0.5
Seidel-type iteration	Number of iterations	13	41	385	D	D	D
	CPU time	32 min 19 s	39 min 42 s	—	—	—	—
Multigrid method	Number of levels in the multigrid	2	3	3	3	3	3
	Number of iterations	1	2	3	3–4	3–4	D
	CPU time	18 min 18 s	16 min 19s	10 min 35 s	4 min 49s	—	

Note. "D" denotes divergent.

4. Application of the Adaptive Algorithm

It is obvious that special refinement of the grid is required near wave crest of the soliton and the coarser grid can be used in other parts of the domain. Therefore, it is valuable to consider adaptive algorithm at the base of the multigrid method.

An important feature of the adaptive algorithm is adaptivity. The grid may change during the solution process, adapting itself to the evolving solution. The key of the algorithm is how to choose the domain requiring local refinement of the grid.

TABLE III

Numerical Results for Collision of Two Solitons Which Move in Opposite Directions

Method	τ	0.01	0.02	0.05	0.1	0.2
Seidel-type iteration	Number of iterations	13–22	64–200	D	D	D
	CPU time	34 min 39 s	—	—	—	—
Multigrid method	Number of levels in the multigrid	3	3	3	3	3
	Number of iterations	$1 \sim 2$	2	3	3–10	D
	CPU time	19 min 59 s	17 min 11 s	11 min 26 s	6 min 51 s	—

TABLE IV

Numerical Results for Collision of Two Solitons Which Move in the Same Direction

Method	τ	0.0625	0.125	0.25
Seidel-type iteration	Number of iterations	15–18	D	D
	CPU time	1 h 0 min 36 s	—	—
Multigrid method	Number of levels in the multigrid	2	3	3
	Number of iterations	3	4–8	D
	CPU time	32 min 24 s	25 min 27 s	—

In general, truncation error can represent error of the solution. The bigger the truncation error, the bigger is the solution error. Hence, the domains requiring local refinement of grid are chosen by means of the truncation error, which is defined as

$$\tau^K = L^K(\hat{I}^K U) - I^K(LU), \tag{4.1}$$

where L is the differential operator, L^K is the difference operator on the grid Ω^K, U is the true differential solution, and $\hat{I}^K$ and I^K are two continuum-to-grid Ω^K projection operators.

In view of Eqs. (3.9) and (3.10) of FAS mode, we have

$$\begin{gathered} L^K U^K = I^K_{K+1} \bar{G}^{K+1} + \tau^K_{K+1}, \qquad K = 0, 1, ..., M-1, \\ L^M U^M = G^M. \end{gathered} \tag{4.2}$$

Assume that the operators $\hat{I}^K$ and I^K possess properties

$$I^K_{K+1} I^{K+1} = I^K, \qquad I^K_{K+1} \hat{I}^{K+1} = \hat{I}^K, \tag{4.3}$$

and

$$\hat{I}^{K+1} U = U^{K+1}, \tag{4.4}$$

when the difference solution U^K converges to the differential solution U. Thus, it follows from (4.1) that

$$\begin{aligned} I^K_{K+1} \tau^{K+1} &= I^K_{K+1} L^{K+1} U^{K+1} - I^K(LU) \\ &= I^K_{K+1} L^{K+1} U^{K+1} - L^K(\hat{I}^K U) + \tau^K = -\tau^K_{K+1} + \tau^K; \end{aligned}$$

i.e.,

$$\tau_{K+1}^{K} = \tau^{K} - I_{K+1}^{K} \tau^{K+1}. \tag{4.5}$$

Making Taylor's expansion, we obtain, from (4.1),

$$\tau^{K} = c^{K}(x) \cdot h_{K}^{p} + O(h_{K}^{p+1}), \tag{4.6}$$

where $c^{K}(x)$ is independent of h_K. Let $h_K = 2h_{K+1}$ as a general rule; then it follows from (4.5) that

$$\tau_{K+1}^{K} = C^{K}(x)\, h_{K}^{p} - I_{K+1}^{K} C^{K+1}(x)\, h_{K+1}^{p} + O(h_{K}^{p+1}).$$

It is easy that I_{K+1}^{K} is chosen to satisfy

$$I_{K+1}^{K} C^{K+1}(x) = C^{K}(x) + O(h_K).$$

We have

$$\tau_{K+1}^{K} = C^{K}(x)\, h_{k}^{p} - c^{K}(x)\, h_{K+1}^{p} + O(h_{K}^{p+1}) = C^{K}(x)[1 - (\tfrac{1}{2})^{p}]\, h_{K}^{p} + O(h_{K}^{p+1}). \tag{4.7}$$

Combining (4.6) with (4.7) yields

$$\tau^{K} = 2^{p}(2^{p} - 1)^{-1} \cdot \tau_{K+1}^{K} + O(h_{K}^{p+1}). \tag{4.8}$$

Therefore, the quantity τ_{K+1}^{K} is proportional to the truncation error τ^{K}. We can choose the domains requiring local refinement of the grid by means of the quantity τ_{K+1}^{K}. This adaptive algorithm at the base of the multigrid is efficient and economical.

Using the difference scheme (3.4)–(3.7) and the restriction operator (3.12), we obtain

$$\begin{aligned}
(L^{K} U^{K})_j &= A^{K} u_{j+1}^{K} + B_{j}^{K} u_{j}^{K} + A^{K} u_{j-1}^{K}, \\
(L^{K} I_{K+1}^{K} u^{K+1})_j &= A^{K} (I_{K+1}^{K} u^{K+1})_{j+1} + B_{j}^{K} (I_{K+1}^{K} u^{K+1})_j + A^{K} (I_{K+1}^{K} u^{K+1})_{j-1} \\
&= A^{K} u_{2j+2}^{K+1} + B_{j}^{K} u_{2j}^{K+1} + A^{K} U_{2j-2}^{K+1}, \\
[I_{K+1}^{K} (L^{K+1} u^{K+1})]_j &= (L^{K+1} u^{K+1})_{2j} \\
&= A^{K+1} u_{2j+1}^{K+1} + B_{2j}^{K+1} u_{2j}^{K+1} + A^{K+1} u_{2j-1}^{K+1}.
\end{aligned}$$

Thus, it follows from the formula (3.10) that

$$\begin{aligned}
(\tau_{K+1}^{K})_j &= A^{K} (u_{2j+2}^{K+1} + u_{2j-2}^{K+1}) \\
&\quad - A^{K+1} (u_{2j+1}^{K+1} - u_{2j-1}^{K+1}) + (B_{j}^{K} - B_{2j}^{K+1})\, u_{2j}^{K+1} \\
&= A^{K+1} (\tfrac{1}{4} u_{2j+2}^{K+1} + \tfrac{1}{4} u_{2j-2}^{K+1} - u_{2j+1}^{K+1} - u_{2j-1}^{K+1}) \\
&\quad + \left(-\frac{\tau}{4h_{K+1}^{2}} + \frac{\tau}{h_{K+1}^{2}} \right) u_{2j}^{K+1} \\
&= A^{K+1} (\tfrac{1}{4} u_{2j+2}^{K+1} + \tfrac{1}{4} u_{2j-2}^{K+1} - u_{2j+1}^{K+1} - u_{2j-1}^{K+1} + \tfrac{3}{2} u_{2j}^{K+1}).
\end{aligned}$$

The quantity τ_{K+1}^{K} is linear, although the difference scheme (3.4)–(3.7) is nonlinear.

TABLE V

Numerical Results of the Adaptive Algorithm for One-Soliton Solution

Method \ τ	0.01	0.02	0.05	0.1
Adaptive algorithm	7 min 53 s	6 min 24 s	4 min 16 s	3 min 49 s
Multigrid method	18 min 18 s	16 min 19 s	10 min 45 s	5 min 35 s

Using the adaptive algorithm, one-soliton solutions for the NLS equations (3.1)–(3.3) are computed. We take $x_L = -16$, $x_R = 16$, $T = 5$, $h = 0.1$, $\tau = 0.01, 0.02, 0.05$, and 0.1. At first, the approximation solution in the interval $[-16, 16]$ is computed by the multigrid of two levels, in which step sizes of space are $h_1 = 0.4$ and $h_2 = 0.2$. In this process, values of the quantity τ_2^1 are stored. Then, maximum value $(\tau_2')_{j_0} = \max_j (\tau_2')_j$ is chosen and domain Ω^*, where $(\tau_2')_j \geqslant 0.1 \cdot (\tau_2')_{j_0}$ is found. Computational experience shows that j_0 is at wave crest and the domain Ω^* is only a interval $j_1 \leqslant j \leqslant j_2$. In order to ensure soliton is symmetrical on j_0, we adjust the domain Ω^* to domain Ω^4 which is a interval $j_0 - \Delta j \leqslant j \leqslant j_0 + \Delta j$, $\Delta j = \max(j_0 - j_1, j_2 - j_0)$. Finally, the approximation solution is computed by the multigrid in step size $h_4 = 0.1$ and $h_2 = 0.2$ at the domain Ω^4. The results are given in Table V. In view of experience, we know that the domain Ω^4 is about the one fifth of the interval $[-16, 16]$ and moves with the soliton.

5. Discussion of Computational Results

(1) We know that convergence of nonlinear iteration requires too small step size of time τ in computation, when nonlinear evolution equation is solved by implicit scheme. For example, the number of iterations in $\tau \leqslant 0.05$ increases sharply with increments of the step size of time τ and iteration process is divergent for $\tau > 0.05$, when one-soliton solution of NLS equation is computed. Our computational results demonstrate a new advantage of the multigrid method: it can considerably relax the restrict on step size of time, which is given by nonlinear iteration. For example, the Seidel-type iteration method is convergent only for $\tau = 0.05$. But, iteration process of the multigrid method still is convergent for $\tau = 0.2$ and CPU time is only 4 min 49s in computing the one-soliton solution. This advantage can encourage the multigrid method to be applied more widely.

(2) We know from the computational results that the adaptive algorithm may combine conveniently with the multigrid method and local refinement of the grid can be chosen by the quantity τ_{K+1}^K. For example, the adaptive algorithm for

one-soliton solution and $\tau = 0.05$ decreases CPU time from 10 min 45 s, which is taken by the multigrid method, to 4 min 16 s and the refined domain is one-fifth of the interval $[-16, 16]$. This means that the adaptive algorithm is suitable and efficient.

(3) The interpolation operator I_K^{K+1} is important for the multigrid method. We try to take various forms of the operator and compare them. When the order of the operator is less than six, the correction on the coarser grid cannot be transmitted to the finest grid, and the iterative error on the finest grid cannot be decreased to less than 10^{-5}. When the order of the operator is too large, there are added components of high frequencies of the error in the finer grids that increase the number of the iteration. We also consider to take cubic splines as the operator. Thus, smooth correction can be obtained with the finer grid. But, computational experience demonstrates that this does not decrease the number of iteration and may increase CPU time, since complication of computing cubic spline functions. The formulas (3.12)–(3.16) are suitable for the NLS equation in view of our experience.

(4) We consider computational accuracy in various step sizes of time. Let a and v denote maximum amplitude and speed of the soliton, respectively. Their numerical results are given in Tables VI and VII for cases (i) and (ii), respectively.

It follows from (2.4) and (2.5) that there are two conserved quantities in basic NLS equations (3.1)–(3.3). They are

$$\int_{x_L}^{x_R} |u(x, t)|^2 \, dx = \text{const.},$$

$$\int_{x_L}^{x_R} \left(|u(x, t)|^4 - \left| \frac{\partial u(x, t)}{\partial x} \right|^2 \right) dx = \text{const.}$$

TABLE VI

Computational Results of Amplitude and Speed of the Soliton for Casr (ii)

t	τ	0.01	0.02	0.05	0.1	0.2
1	a	1.008	1.013	1.023	1.057	1.024
	v	4.000	4.000	4.000	3.900	3.700
2	a	1.016	1.019	1.033	1.094	1.075
	v	4.000	4.000	4.000	3.900	3.650
3	a	1.009	1.014	1.026	1.086	1.093
	v	4.000	4.000	3.950	3.850	3.650
4	a	1.007	1.010	1.021	1.064	1.051
	v	4.000	3.950	3.950	3.800	3.600
5	a	1.005	1.007	1.018	1.045	0.982
	v	4.000	3.950	3.950	3.750	3.400

TABLE VII

Computational Results of Amplitude and Speed of the Solitons for Case (ii)

t	τ	0.01		0.05	
		First soliton	Second soliton	First soliton	Second soliton
1	a	1.009	1.009	1.023	1.023
	v	4.000	4.000	4.000	4.000
2	a	1.005	1.005	1.029	1.029
	v	4.000	4.000	4.000	4.000
2.5	a	2.045	2.045	2.080	2.080
	v	4.000	4.000	4.000	4.000
3	a	0.9992	0.9992	1.033	1.033
	v	4.000	4.000	4.000	4.000
4	a	1.010	1.010	1.026	1.026
	v	4.000	4.000	4.000	4.000
5	a	1.005	1.005	1.014	1.014
	v	4.000	4.000	4.000	4.000

Errors of the conserved quantities are denoted by p_1 and p_2; i.e.,

$$p_1 = \left| \frac{q_1 - q_{10}}{q_{10}} \right|, \qquad p_2 = \left| \frac{q_2 - q_{20}}{q_{20}} \right|,$$

where q_{10} and q_{20} are exact values of $\int_{x_L}^{x_R} |u|^2 \, dx$ and $\int_{x_L}^{x_R} (|u|^4 - |\partial u/\partial x|^2) \, dx$, respectively, and q_1 and q_2 are the calculated values of the quantities. It follows from the computational results that p_1 and p_2 are very small for $\tau \leqslant 0.1$. For example, in computation of one-soliton solution there are

$$p_1 \leqslant 0.0001, \qquad p_2 \leqslant 0.005, \qquad \text{for} \quad \tau = 0.01;$$

$$p_1 \leqslant 0.004, \qquad p_2 \leqslant 0.01, \qquad \text{for} \quad \tau = 0.05.$$

In view of the Tables VI, VII and the values of p_1 and p_2, we can find that it is better to take $\tau = 0.05$ for the case (i) and (ii), and comparison of CPU times of various algorithms are given in Table VIII. The results for the case (i) and $\tau = 0.05$ is drawn in Fig. 2.

Theorefore, the multigrid method is efficient for solving the nonlinear evolution equation and the adaptive algorithm can combine with the multigrid method for the problem.

Comparing the adaptive solution with the solution of the multigrid method, we know that the difference between them is less than 10^{-5}. This result is satisfactory, because the iterative error E also is less than 10^{-5}.

TABLE VIII

Comparison of CPU Time for Cases (i) and (ii)

Case	Seidel-type iteration	Multigrid method	Adaptive algorithm
(i)	32 min 19 s	10 min 45 s	4 min 16 s
(ii)	34 min 29 s	11 min 29 s	—

Note. $\tau = 0.01$ for Seidel-type iteration and $\tau = 0.05$ for other methods.

(5) We try to use a greater number of levels and coarser grid in the multigrid method. The results for the one-soliton solution, $\tau = 0.5$ and $\tau = 5$, are given in Table IX. Other results indicate a similar property. The computational results show that the multigrid procedure presented in this paper is convergent for a greater number of levels and coarser grid. But, the coarsest grid should be fine enough to provide rough approximation.

(6) In order to compare CPU time between the multigrid method and the Seidel-type iteration, we require that $E \equiv \max_j |u_j^{n+1(s+1)} - u_j^{n+1(s)}| \leqslant 10^{-5}$. It is possible that the actual error is smaller in the multigrid than in the Seidel-type, because in the latter the convergence is much slower than in the former. Therefore, we consider two criteria of convengence in the multigrid

$$E^{n+1} \equiv \max_j |u_j^{n+1(s+1)} - u_j^{n+1(s)}| \leqslant 10^{-5}$$

and

$$E_*^{n+1} \equiv \max_j |u_j^{n+1(s+1)} - u_j^*| \leqslant 10^{-5},$$

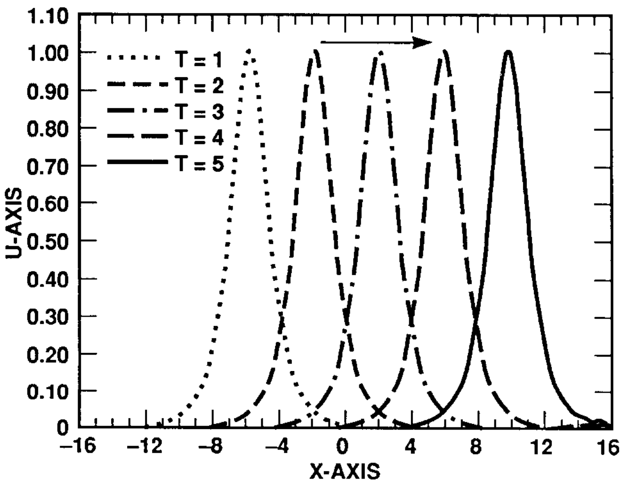

FIG. 2. Soliton solution for the case (i) and the $\tau = 0.005$.

TABLE IX

Comparison of Various Number of Levels for One-Soliton Solution, $\tau = 0.5$ and $T = 5$

Number of levels in the multigrid	2	3	4	5	6
Number of iterations	3	3	3	3–4	4–4
CPU time	11 min 29 s	10 min 45 s	10 min 54 s	12 min 44 s	13 min 37 s

where u_j^* denotes value on the finest grid before Seidel-type iteration which is to smooth error out in the multigrid. The values E^{n+1} and E_*^{n+1} for one-soliton solution and $\tau = 0.05$ are given in Table X. It follows from the Table X that the error E^{n+1} is slight smaller than $E_*{}^{n+1}$, when they are less than 10^{-5}. Therefore, we can use $E^{n+1} \leqslant 10^{-5}$ rough to compare CPU time between the multigrid and the Seidel-type iteration.

APPENDIX: Uniqueness and Convergence of the Nonlinear Iteration (3.17)

In this section, we consider the difference scheme (3.4)–(3.7) and the Seidel-type iteration (3.17). We use the notations

$$\|u^n\|^2 = h \sum_{j=1}^{J-1} |u_j^n|^2, \qquad \|u_x^n\|^2 = h \sum_{j=0}^{J-1} |(u_j^n)_x|^2,$$

$$\|u^n\|_\infty = \sup_{1 \leqslant j \leqslant J-1} |u_j^n|.$$

TABLE X

Values E^{n+1} and E_*^{n+1} for one-soliton solution and $\tau = 0.05$

n	Number of iterations	E^{n+1}	E_*^{n+1}
0	First	0.167	4.812×10^{-4}
	Second	4.208×10^{-5}	1.293×10^{-5}
	Third	9.537×10^{-7}	1.073×10^{-6}
1	First	0.168	4.481×10^{-4}
	Second	4.262×10^{-5}	1.192×10^{-5}
	Third	1.073×10^{-6}	1.252×10^{-6}
4	First	0.171	5.097×10^{-4}
	Second	5.037×10^{-5}	1.419×10^{-5}
	Third	1.132×10^{-6}	1.371×10^{-6}

LEMMA 1 [7, Lemma 2.1]. *Let $\varepsilon > 0$ be a given constant, then there exists a constant c_1 depending only on ε such that*

$$\max_{0 \leqslant x \leqslant 1} |y(x)|^2 \leqslant \varepsilon \left\| \frac{dy(x)}{dx} \right\|_{L_2}^2 + c \| y(x) \|_{L_2}^2,$$

where $y(x) \in L_\infty[0, 1]$.

LEMMA 2 [8, Lemma 4.2]. *For any h, there exists such operator I_h: $L_2^h \to L_2[0, 1]$ that if $y^h \in L_2^h$ and $y(x) = I_h y^h$, then $y(x_j) = y^h(x_j)$ and $y(x)$ is analytic, I_h commutes with shifts and differences, and there is estimate*

$$\left(\frac{2}{\pi}\right)^m \left\| \frac{d^m y(x)}{dx^m} \right\|_{L_2} \leqslant \| D_+^m y^h \| \leqslant \left\| \frac{d^m y(x)}{dx^m} \right\|_{L_2},$$

where

$$L_2^h = \left\{ y^h : h \sum_{j=1}^{J-1} |y_j^h|^2 < \infty,\ y_0^h = y_J^h = 0 \right\}, \qquad (D^+ y^h)_j = \frac{1}{h} (y_{j+1}^h - y_j^h).$$

LEMMA 3. *Let $\varepsilon > 0$ be a given constant, then there exists a constant c_2 depending only on ε such that*

$$\|u^n\|_\infty \leqslant \varepsilon \|u_x^n\| + c_2 \|u^n\|.$$

Proof. It is immediate by Lemma 1 and Lemma 2.

THEOREM 4. *Assume that $u_0(x) \in H^1$, then there are estimates for the solution of the difference scheme* (3.4)–(3.7)

$$\|u^n\| \leqslant c_3, \qquad \|u_x^n\| \leqslant c_4, \qquad \|u^n\|_\infty = c_5.$$

Proof. Taking $G_{m,j}^{K+1/2} = 0$, $f_{m,l,j}^{K+1/2} = 0$, $\beta_j = 2$, $a_{m,j+1/2} = -1$, and $Q_j^{n+1} = \frac{1}{2} |u_j^{n+1}|^4$ in the conservation laws (2.9) and (2.10), we have for the solution of the difference scheme (3.4)–(3.7)

$$\|u^n\|^2 = \|u^0\|^2, \tag{A.1}$$

$$-\|(u^{n+1})_x\|^2 + h \sum_{j=1}^{J-1} |u_j^{n+1}|^4 = -\|u_x^0\|^2 + h \sum_{j=1}^{J-1} |u_j^0|^4. \tag{A.2}$$

Without loss of generality, we can assume that h is chosen so small that there are

$$\|u^0\| \leqslant 2 \|u_0(x)\|_{L_2}, \qquad \|u_x^0\| \leqslant 2 \left\| \frac{du_0(x)}{dx} \right\|_{L_2},$$

$$h \cdot \sum_{j=1}^{J-1} |u_j^0|^4 \leqslant 2 \int_{x_L}^{x_R} [u_0(x)]^4 \, dx \equiv 2 \|u_0(x)\|_{L_4}^4.$$

Thus, it follows from (A.1) and (A.2) that

$$\|u^n\| \leqslant 2\|u_0(x)\|_{L_2} \equiv c_3, \tag{A.3}$$

$$\|u_x^{n+1}\|^2 \leqslant 4\left\|\frac{du_0(x)}{dx}\right\|_{L_2}^2 + 2\|u_0(x)\|_{L_4}^4 + h\sum_{j=1}^{J-1}|u_j^{n+1}|^4. \tag{A.4}$$

From Lemma 3 and (A.3), we have

$$\begin{aligned} h\sum_{j=1}^{J-1}|u_j^{n+1}|^4 &\leqslant \max_{1\leqslant j\leqslant J-1}|u_j^{n+1}|^2 \cdot h\sum_{j=1}^{J-1}|u_j^{n+1}|^2 \\ &\leqslant c_3^2[\varepsilon\|u_x^{n+1}\| + c_2\|u^{n+1}\|]^2 \\ &\leqslant 2c_3^2[\varepsilon^2\|u_x^{n+1}\|^2 + c_2^2\|u^{n+1}\|^2], \end{aligned}$$

where ε is chosen such that $\varepsilon \cdot c_3 < \frac{1}{2}$. On combining this with (A.3) and (A.4), we deduce that

$$\|u_x^{n+1}\|^2 \leqslant 2\left[4\left\|\frac{du_0(x)}{dx}\right\|_{L_2}^2 + 2\|u_0(x)\|_{L_4}^4 + 2c_3^4c_2^2\right] \equiv c_4^2. \tag{A.5}$$

Thus, it follows from Lemma 3, (A.3), and (A.5) that

$$\|u^n\|_\infty \leqslant c_5.$$

THEOREM 5. *Assume $u_0(x) \in H^1$. If we use iterative initial value $u_j^{n+1(0)} = u_j^n$, $1 \leqslant j \leqslant J-1$ and the step size of space and time satisfy*

$$h \leqslant \frac{1}{\sqrt{5}\,c_5}, \qquad \tau \leqslant c_6 \cdot h^2,$$

where c_6 is a positive constant depending on c_5. Then the Seidel-type iterative algorithm (3.17) *is convergent and solution of the difference scheme is unique.*

Proof. Let $\varepsilon_j^{n+1(s)} = u_j^{n+1} - u_j^{n+1(s)}$. From (3.4) and (3.17) we have

$$A\varepsilon_{j+1}^{n+1(s)} + B_j^{n+1(s)}\varepsilon_j^{n+1(s+1)} + A\varepsilon_{j-1}^{n+1(s+1)} = H_j^{n+1(s)}, \qquad 1 \leqslant j \leqslant J-1, \tag{A.6}$$

where

$$H_j^{n+1(s)} = -\frac{\tau}{2}(u_j^{n+1} + u_j^n)(|u_j^{n+1}| + |u_j^{n+1(s)}|)\,\varepsilon_j^{n+1(s)}.$$

Now, we prove by contradiction that

$$\max_{1\leqslant j\leqslant J-1}|\varepsilon_j^{n+1(s+1)}| \leqslant \delta \max_j |\varepsilon_j^{n+1(s)}|, \tag{A.7}$$

where $0 < \delta < 1$, independing on s.

Assume (A.7) to be false, then for any $1 > \varepsilon > 0$ there exists $s_0 \geqslant 0$ such that

$$|\varepsilon_{j0}^{n+1(s_0+1)}| = \max_{1 \leqslant j \leqslant J-1} |\varepsilon_j^{n+1(s_0+1)}| \geqslant (1-\varepsilon) \max_{1 \leqslant j \leqslant J-1} |\varepsilon_j^{n+1(s_0)}|, \tag{A.8}$$

and it holds for $s < s_0$:

$$\max_{1 \leqslant j \leqslant J-1} |\varepsilon_j^{n+1(s+1)}| \leqslant \delta \max_{1 \leqslant j \leqslant J-1} |\varepsilon_j^{n+1(s)}|. \tag{A.9}$$

Thus, (A.6) yields

$$\begin{aligned} |B_{j0}^{n+1(s_0)}| &\leqslant \frac{|H_{j0}^{n+1(s_0)}|}{|\varepsilon_{j0}^{n+1(s_0+1)}|} + A \frac{|\varepsilon_{j0+1}^{n+1(s_0)}|}{|\varepsilon_{j0}^{n+1(s_0+1)}|} + A \frac{|\varepsilon_{j0-1}^{n+1(s_0+1)}|}{|\varepsilon_{j0}^{n+1(s_0+1)}|} \\ &\leqslant \frac{|H_{j0}^{n+1(s_0)}|}{|\varepsilon_{j0}^{n+1(s_0+1)}|} + A \cdot \frac{1}{1-\varepsilon} + A; \end{aligned}$$

i.e.,

$$\max_{1 \leqslant j \leqslant J-1} |\varepsilon_j^{n+1(s_0+1)}| \leqslant \frac{|H_{j0}^{n+1(s_0)}|}{|B_{j0}^{n+1(s_0)}| - A - A \cdot (1/(1-\varepsilon))}. \tag{A.10}$$

From $u_j^{n+1(0)} = u_j^n$ and (A.9), we have

$$\max_{1 \leqslant j \leqslant J-1} |\varepsilon_j^{n+1(s)}| \leqslant \max_{1 \leqslant j \leqslant J-1} |u_j^{n+1} - u_j^n|, \qquad s \leqslant s_0. \tag{A.11}$$

It follows from the Theorem 4 that

$$|\varepsilon_j^{n+1(s)}| \leqslant 2c_5, \qquad s \leqslant s_0. \tag{A.12}$$

Considering $h \leqslant 1/\sqrt{5}\, c_5$, we have estimate

$$\begin{aligned} &|B_{j0}^{n+1(s_0)}| - A - A \cdot \frac{1}{1-\varepsilon} \\ &= \sqrt{1 + \frac{\tau^2}{h^4}\left[1 - \frac{h^2}{2}\left(|u_{j0}^{n+1(s_0)}|^2 + |u_{j0}^n|^2\right)\right]^2} - \frac{\tau}{2h^2} - \frac{\tau}{2h^2(1-\varepsilon)} \\ &\geqslant \sqrt{1 - \frac{\tau^2}{4h^2}} - \frac{\tau}{2h^2} - \frac{\tau}{2h^2(1-\varepsilon)}. \end{aligned}$$

Choosing τ such that

$$\tau \leqslant \left\{\left[c_0 + \frac{1}{2h^2}\left(1 + \frac{1}{1-\varepsilon}\right)\right]^2 - \frac{1}{4h^2}\right\}^{-1/2} \tag{A.13}$$

where c_0 is a positive constant, which can be definite in the sequel, we get

$$|B_{j0}^{n+1(s_0)}| - A - A\frac{1}{1-\varepsilon} \geqslant c_0\tau. \tag{A.14}$$

In view of the Theorem 4 and (A.12), it holds

$$|H_{j0}^{n+1(s_0)}| \leqslant 3c_5^2\tau\,|\varepsilon_{j0}^{n+1(s_0)}|. \tag{A.15}$$

Combining (A.10), (A.14), and (A.15), we have

$$\max_{1\leqslant j\leqslant J-1} |\varepsilon_j^{n+1(s_0+1)}| \leqslant \frac{3c_5^2}{c_0} \max_{1\leqslant j\leqslant J-1} |\varepsilon_j^{n+1(s_0)}|, \tag{A.16}$$

c_0 can be chosen such that

$$\frac{3c_5^2}{c_0} < 1-\varepsilon. \tag{A.17}$$

Thus,

$$\max_{1\leqslant j\leqslant J-1} |\varepsilon_j^{n+1(s_0+1)}| < (1-\varepsilon) \max_{1\leqslant j\leqslant J-1} |\varepsilon_j^{n+1(s_0)}|.$$

Since this leads to a contradiction, then (A.7) is true. Therefore, the iterative algorithm (3.17) is convergent and the solution is unique.

Combining (A.13) and (A.17), we know that τ should satisfy the following condition,

$$\tau \leqslant c_6 \cdot h^2,$$

where c_6 is a constant depending on c_5.

From the Theorem 5, we known that h can be chosen big enough and the restriction on τ is serious in the iterative algorithm (3.17).

References

1. Q. Chang, *Sci. Sinica* (*Ser. A*) **16**, 687 (1983).
2. Q. Chang and L. Xu, *J. Comput. Math.* **4**, 191 (1986).
3. A. Brandt, *Math. Comput.* **31**, 333 (1977).
4. W. Hackbusch, *Multigrid Methods and Applications* (Springer-Verlag, New York, 1985).
5. T. R. Tana and M. J. Ablowitz, *J. Comput. Phys.* **55**, 203 (1984).
6. A. Brandt, *Multigrid Techniques: Guide with Applications to Fluid Dynamics* (GMD-Stidier No. 85, Bonn, 1984).
7. A. Sjoberg, *J. Math. Anal. Appl.* **29**, 569 (1970).
8. A. Manikoff, *Commun. Pure Appl. Math.* **25**, 407 (1972).