

文章编号: 1000_0887(2004)02_0213_08

Burgers 方程的一类交替分组方法^{*}

王文洽

(山东大学 数学与系统科学学院, 济南 250100)

(李继彬推荐)

摘要: 对于 Burgers 方程给出了一组新的 Saul'yev 型非对称差分格式, 并用这些差分格式构造了求解非线性 Burgers 方程的交替分组四点方法. 该算法把剖分节点分成若干组, 在每组上构造能够独立求解的差分方程. 因此算法具有并行本性, 能直接在并行计算机上使用. 文章还证明了所给算法线性绝对稳定. 数值试验表明, 该方法使用简便, 稳定性好, 有很好的精度.

关键词: Burgers 方程; Saul'yev 型非对称格式; 交替分组四点格式; 线性绝对稳定; 并行计算

中图分类号: O241 文献标识码: A

引 言

我们考虑非线性 Burgers 方程

$$\frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} = \varepsilon \frac{\partial^2 u}{\partial x^2}, \quad 0 < x < L, \quad 0 < t < T, \quad \varepsilon > 0; \quad (1)$$

初始条件和边界条件是

$$u(x, 0) = f(x), \quad 0 < x < L, \quad (2)$$

$$u(0, t) = g_1(t), \quad u(L, t) = g_2(t), \quad 0 < t < T, \quad (3)$$

这里 ε 是常数.

由于 Burgers 方程可以作为流体一类流动现象的数学模型, 又具有 Navier-Stokes 方程的一些性质, 因此讨论 (1) 的数值解法还有求解方程本身以外的学术价值. 求解 (1) 的有限差分方法和有限元方法已经很多, Evans 和张宝琳等人针对扩散方程和对流-扩散方程设计了一类有并行本性的 (Alternating Group Explicit) AGE 和 (Alternating Group Explicit-Implicit) AGE-I 方法^[1~5], Evans 和 Abdullah, 陆金甫和张宝琳等人还分别对方程 (1) 给出了 AGE 格式^[6~10]. 其中文献 [6, 9] 给出的是 (1) 的全隐格式, 而用的是交替分组显式迭代求解. 在这篇文章里, 我们用第二类 Saul'yev 型非对称格式, 构造了方程 (1) 的交替分组四点方法, 其中对流项的离散化不同于文献 [6~10]. 该方法线性绝对稳定, 使用简便, 适合在高性能多处理器的并行机上直接使用. 数值试验表明交替分组四点方法有很好的精度.

* 收稿日期: 2001_05_08; 修订日期: 2003_06_27

基金项目: 国家教育部博士点专项基金资助项目 (97042202); 山东省自然科学基金资助项目 (Y2003A04)

作者简介: 王文洽 (1950—), 男, 山东阳谷县人, 教授 (Fax: 86_531_8364652; E_mail: wangwq@sdu.edu.cn).

1 交替分组四点格式

令 h 和 Δt 分别表示 x 和 t 方向的网格步长, $h = L/m, m$ 为正整数. u_i^n 表示问题(1) ~ (3) 在网格点 (x_i, t_n) 的逼近解, 这里 $x_i = ih (i = 0, 1, \dots, m), t_n = n \Delta t$, 并记 $t_{n+1/2} = t_n + 0.5\Delta t (n = 0, 1, \dots, [T/\Delta t])$. 为简便起见, 用 (i, n) 表示节点 $(x_i, t_n), (i, n + 1/2)$ 表示 $(x_i, t_{n+1/2})$, 并定义下面的记号

$$\begin{aligned} \delta_x u_i^n &= (u_{i+1}^n - u_i^n)/h, \quad \delta_x u_i^n = (u_i^n - u_{i-1}^n)/h, \\ \delta_x u_i^n &= (u_{i+1}^n - u_{i-1}^n)/2h, \quad \delta_x u_i^n = (u_i^{n+1} - u_i^n)/\Delta t \end{aligned}$$

则方程(1)在节点 $(x_i, t_{n+1/2})$ 处的 4 个第二类 Saul'yev 型格式是

$$\delta_x u_i^n + \frac{u_i^{n+1/2}}{2} \left\{ \frac{u_{i+1}^{n+1} - u_{i-1}^n}{2h} + \delta_x u_i^n \right\} = \frac{\varepsilon}{2} (\delta_x (\delta_x u)_i^n + h^{-1} (\delta_x u_i^{n+1} - \delta_x u_i^n)), \quad (4)$$

$$\delta_x u_i^n + \frac{u_i^{n+1/2}}{2} \left\{ \frac{u_{i+1}^n - u_{i-1}^{n+1}}{2h} + \delta_x u_i^n \right\} = \frac{\varepsilon}{2} (\delta_x (\delta_x u)_i^n + h^{-1} (\delta_x u_i^n - \delta_x u_i^{n+1})), \quad (5)$$

$$\delta_x u_i^n + \frac{u_i^{n+1/2}}{2} \left\{ \frac{u_{i+1}^{n+1} - u_{i-1}^n}{2h} + \delta_x u_i^{n+1} \right\} = \frac{\varepsilon}{2} (\delta_x (\delta_x u)_i^{n+1} + h^{-1} (\delta_x u_i^{n+1} - \delta_x u_i^n)), \quad (6)$$

$$\delta_x u_i^n + \frac{u_i^{n+1/2}}{2} \left\{ \frac{u_{i+1}^n - u_{i-1}^{n+1}}{2h} + \delta_x u_i^{n+1} \right\} = \frac{\varepsilon}{2} (\delta_x (\delta_x u)_i^{n+1} + h^{-1} (\delta_x u_i^n - \delta_x u_i^{n+1})). \quad (7)$$

为了对(4) ~ (7) 进行线性化, 式中 $u_i^{n+1/2}$ 按文[9]的思想处理, 如图 1 取

$$u_i^{n+1/2} = u_i^c \approx \frac{u_i^n}{1 + \frac{\Delta t}{2h}(u_i^n - u_{i-1}^n)} \equiv a_i^n, \quad (8)$$

记 $r = \Delta t/(2h^2)$, 于是(4) ~ (7) 式可以改写成下面的形式

$$(1 + r\varepsilon) u_i^{n+1} - r(\varepsilon - a_i^n h/2) u_{i+1}^n = r(2\varepsilon + a_i^n h) u_{i-1}^n + (1 - 3r\varepsilon) u_i^n + r(\varepsilon - a_i^n h/2) u_{i+1}^n, \quad (9)$$

$$- r(\varepsilon + a_i^n h/2) u_{i-1}^{n+1} + (1 + r\varepsilon) u_i^{n+1} = r(\varepsilon + a_i^n h/2) u_{i-1}^n + (1 - 3r\varepsilon) u_i^n + r(2\varepsilon - a_i^n h/2) u_{i+1}^n, \quad (10)$$

$$- r(\varepsilon + a_i^n h/2) u_{i-1}^{n+1} + (1 + 3r\varepsilon) u_i^{n+1} - r(2\varepsilon - a_i^n h) u_{i+1}^n = r(\varepsilon + a_i^n h/2) u_{i-1}^n + (1 - r\varepsilon) u_i^n, \quad (11)$$

$$- r(2\varepsilon + a_i^n h) u_{i-1}^{n+1} + (1 + 3r\varepsilon) u_i^{n+1} - r(\varepsilon - a_i^n h/2) u_{i+1}^n = (1 - r\varepsilon) u_i^n + r(\varepsilon - a_i^n h/2) u_{i+1}^n. \quad (12)$$

在讨论交替分组四点方法之前, 先给出本文使用的几种基本组上的差分格式. 假设第 n 层上的数值解 u_i^n 已经知道, 为了求第 $(n + 1)$ 层上的值 u_i^{n+1} , 我们在由节点 $(k, n + 1) (k = i, i + 1, i + 2, i + 3)$ 组成的四点组上, 分别使用差分方程(9)、(11)、(12) 和 (10). 记 $u_j^n = (u_i^n, u_{i+1}^n, u_{i+2}^n, u_{i+3}^n)^T, b^{(j)} = (2rd_i u_{i-1}^n, 0, 0, 2rc_{i+3} u_{i+4}^n)^T, d_i^n = \varepsilon + a_i^n h/2, c_i^n = \varepsilon - a_i^n h/2$, 这里, j 是四点组的序号, d_i^n 和 c_i^n 都是与 n 有关的量. 为简便起见, 在下文中记 $d_i = d_i^n, c_i = c_i^n$, 于是可以得到四点组上差分格式的矩阵形式

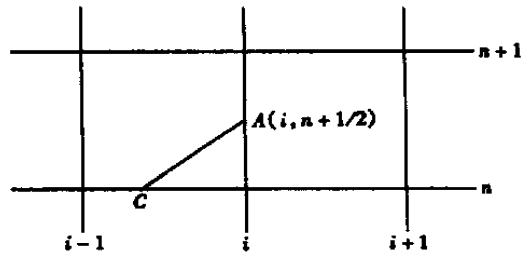


图 1 质点轨迹

$$(I + rQ_4^{(j)}) u_j^{n+1} = (I - rP_4^{(j)}) u_j^n + b^{(j)}, \quad (13)$$

$$Q_4^{(j)} = \begin{bmatrix} \varepsilon & -c_i & & \\ -d_{i+1} & 3\varepsilon & -2c_{i+1} & \\ & -2d_{i+2} & 3\varepsilon & -c_{i+2} \\ & & -d_{i+3} & \varepsilon \end{bmatrix},$$

$$P_4^{(j)} = \begin{bmatrix} 3\varepsilon & -c_i & & \\ -d_{i+1} & \varepsilon & & \\ & & \varepsilon & -c_{i+2} \\ & & -d_{i+3} & \varepsilon \end{bmatrix}.$$

显然, (13) 是一个独立系统, 我们可以求出 $(I + rQ_4^{(j)})^{-1}$, 显式地表示出 u_j^{n+1} , 也可以通过解方程组 (13) 求 u_j^{n+1} .

其次, 我们在由靠近左边界的 2 点 $(1, n+1)$ 和 $(2, n+1)$ 组成的二点组上分别使用格式 (12) 和 (10), 于是可以得到

$$\begin{bmatrix} 1+3r\varepsilon & -rc_1 \\ -rd_2 & 1+r\varepsilon \end{bmatrix} \begin{bmatrix} u_1^{n+1} \\ u_2^{n+1} \end{bmatrix} = \begin{bmatrix} 1-r\varepsilon & rc_1 \\ rd_2 & 1-3r\varepsilon \end{bmatrix} \begin{bmatrix} u_1^n \\ u_2^n \end{bmatrix} + \begin{bmatrix} 2rd_1 u_0^{n+1} \\ 2rc_2 u_3^n \end{bmatrix}. \quad (14)$$

同样, 我们也可以在靠近右边界的二点组上建立差分格式

$$\begin{bmatrix} 1+r\varepsilon & -rc_{m-2} \\ -rd_{m-1} & 1+3r\varepsilon \end{bmatrix} \begin{bmatrix} u_{m-2}^{n+1} \\ u_{m-1}^{n+1} \end{bmatrix} = \begin{bmatrix} 1-3r\varepsilon & rc_{m-2} \\ rd_{m-1} & 1-r\varepsilon \end{bmatrix} \begin{bmatrix} u_{m-2}^n \\ u_{m-1}^n \end{bmatrix} + \begin{bmatrix} 2rd_{m-2} u_{m-3}^n \\ 2rc_{m-1} u_m^{n+1} \end{bmatrix}. \quad (15)$$

记 $c = (1+r\varepsilon)(1+3r\varepsilon) - r^2 d_2 c_1$, 从 (14) 可以解出 u_1^{n+1} 和 u_2^{n+1}

$$\begin{cases} u_1^{n+1} = (2rd_1(1+r\varepsilon)u_0^{n+1} + (1-r^2\varepsilon^2 + r^2d_2c_1)u_1^n + \\ \quad 2rd_2(1-r\varepsilon)u_2^n + 2r^2c_1c_2u_3^n)/c, \\ u_2^{n+1} = (2r^2d_1d_2^2u_0^{n+1} + 2rc_1(1+r\varepsilon)u_1^n + (1-9r^2\varepsilon^2 + \\ \quad r^2d_2c_1)u_2^n + 2rc_2(1+3r\varepsilon)u_3^n)/c. \end{cases}$$

记 $c = (1+r\varepsilon)(1+3r\varepsilon) - r^2 d_{m-1} c_{m-2}$, 从 (15) 可以解出 u_{m-2}^{n+1} 和 u_{m-1}^{n+1}

$$\begin{cases} u_{m-2}^{n+1} = (2rd_{m-2}(1+3r\varepsilon)u_{m-3}^n + (1-9r^2\varepsilon^2 + r^2d_{m-1}c_{m-2})u_{m-2}^n + \\ \quad 2rc_{m-2}(1+r\varepsilon)u_{m-1}^n + 2r^2c_{m-1}c_{m-2}u_m^{n+1})/c, \\ u_{m-1}^{n+1} = (2r^2d_{m-1}d_{m-2}u_{m-3}^n + 2rd_{m-1}(1-r\varepsilon)u_{m-2}^n + \\ \quad (1-r^2\varepsilon^2 + r^2d_{m-1}c_{m-2})u_{m-1}^n + 2rc_{m-1}(1+r\varepsilon)u_m^{n+1})/c. \end{cases}$$

现在我们讨论交替分组四点方法. 设 n 为偶数, 分别考虑分点数 m 是奇数和偶数的情况. 当 m 是奇数时, 内点数 $m-1$ 是偶数, 我们分两种情况讨论节点的分组模式.

1) 对于 $m-1 = 4J+2$, J 是正整数, 我们在 t_{n+1} 层上划分 $(J+1)$ 个独立计算组. 从左至右, 第 1 组至第 J 组, 每组由 4 个内点组成, 其差分格式由联立方程组 (13) 表示; 在靠近右边界的两点 $(m-2, n+1)$ 和 $(m-1, n+1)$ 上, 其格式由联立方程组 (15) 表示. 类似地, 在 t_{n+2} 层上也划分 $(J+1)$ 个独立计算组, 从左至右, 第 1 组为两点组, 其格式由联立方程组 (14) 表示; 第 2 组至第 $(J+1)$ 组是四点组, 每组上的差分格式由 (13) 式表示. 交替使用 t_{n+1} 和 t_{n+2} 层上的差分格式, 于是得到交替分组四点格式的矩阵形式是

$$\begin{cases} (I + rG_1^{(n)}) U^{n+1} = (I - rG_2^{(n)}) U^n + F_1, \\ (I + rG_2^{(n+1)}) U^{n+2} = (I - rG_1^{(n+1)}) U^{n+1} + F_2, \end{cases} \quad (16)$$

式中

$$U^n = (u_1^n, u_2^n, \dots, u_{m-1}^n)^T, \quad F_1 = (2rd_1^n u_0^n, 0, \dots, 0, 2rc_{m-1}^n u_m^{n+1})^T, \\ F_2 = (2rd_1^{n+1} u_0^{n+2}, 0, \dots, 0, 2rc_{m-1}^{n+1} u_m^{n+1})^T.$$

U^n, F_1, F_2 都是 $(m-1)$ 维向量。

$$G_1^{(n)} = \begin{bmatrix} Q_4^{(1)} & & & & \\ & Q_4^{(2)} & & & \\ & & \ddots & & \\ & & & Q_4^{(J)} & \\ & & & & Q_2^{(r)} \end{bmatrix}, \quad G_2^{(n)} = \begin{bmatrix} Q_2^{(l)} & & & & \\ & Q_4^{(2)} & & & \\ & & \ddots & & \\ & & & Q_4^{(J)} & \\ & & & & Q_4^{(J+1)} \end{bmatrix}. \quad (17)$$

$$Q_2^{(r)} = \begin{bmatrix} \varepsilon & -c_{m-2} \\ -d_{m-1} & 3\varepsilon \end{bmatrix}, \quad Q_2^{(l)} = \begin{bmatrix} 3\varepsilon & -c_1 \\ -d_2 & \varepsilon \end{bmatrix}.$$

2) 对于 $m-1=4J$, 我们在两个时间层上交替构造上分组差分格式, 在 t_{n+1} 层上划分 J 个独立计算的四点组, 每组上都使用格式(13)。在 t_{n+2} 层上划分 $(J+1)$ 个独立计算组, 从左至右, 第 1 组包含 2 个内点, 应用格式(14); 从第 2 组至第 J 组, 每组都由 4 个内点组成, 使用格式(13); 靠近右边界的第 $(J+1)$ 组由内点 $(m-2, n+2)$ 和 $(m-1, n+2)$ 组成, 应用格式(15)。交替使用 t_{n+1} 层和 t_{n+2} 层上的差分格式, 于是得到另一个交替分组四点格式, 其差分格式的矩阵形式是

$$\begin{cases} (I + rG_1^{(n)}) U^{n+1} = (I - rG_2^{(n)}) U^n + F_1, \\ (I + rG_2^{(n+1)}) U^{n+2} = (I - rG_1^{(n+1)}) U^{n+1} + F_2 \end{cases} \quad (18)$$

式中

$$F_1 = (2rd_1^n u_0^n, 0, \dots, 0, 2rc_{m-1}^n u_m^n)^T, \quad F_2 = (2rd_1^{n+1} u_0^{n+2}, 0, \dots, 0, 2rc_{m-1}^{n+1} u_m^{n+2})^T$$

是与边界条件有关的 $(m-1)$ 维向量。

$$G_1^{(n)} = \begin{bmatrix} Q_4^{(1)} & & & & \\ & Q_4^{(2)} & & & \\ & & \ddots & & \\ & & & Q_4^{(J-1)} & \\ & & & & Q_4^{(J)} \end{bmatrix}, \quad G_2^{(n)} = \begin{bmatrix} Q_2^{(l)} & & & & \\ & Q_4^{(2)} & & & \\ & & \ddots & & \\ & & & Q_4^{(J)} & \\ & & & & Q_2^{(r)} \end{bmatrix}. \quad (19)$$

对于 m 为偶数的情况, 内点数 $m-1$ 为奇数。同样, 我们也分两种情况讨论交替分组四点方法。

3) 对于 $m-1=4J+1$, 我们在 t_{n+1} 层上划分 J 个独立计算组。从左至右, 第 1 组至第 $(J-1)$ 组, 每组由 4 个内点组成, 其差分格式由可以独立计算的联立方程组(13)表示; 为了避免单点上的显格式影响整体精度, 我们把最后 5 个点组成一个组。同时为了与前面使用的 Saul'yev 格式匹配, 保证方法的线性绝对稳定性, 我们在最后 2 点 $(m-2, n+1)$ 和 $(m-1, n+1)$ 上使用 Crank_Nicolson 格式。类似地, 在 t_{n+2} 上层也划分 $(J+1)$ 个独立计算组, 从左至右, 第 1 组为两点组, 其格式由联立方程组(14)表示; 第 2 组至第 J 组, 每组由 4 个内点组成, 其差分格式仍然由(13)式表示; 第 $(J+1)$ 组为三点组, 同样, 象在 t_{n+1} 层上那样, 我们在 $(m-2, n+2)$ 和 $(m-1, n+2)$ 2 个点上使用 Crank_Nicolson 格式。交替使用 t_{n+1} 层和 t_{n+2} 层上的差分格式, 于是得到

$$\begin{cases} (I + rG_1^{(n)}) U^{n+1} = (I - rG_2^{(n)}) U^n + F_1, \\ (I + rG_2^{(n+1)}) U^{n+2} = (I - rG_1^{(n+1)}) U^{n+1} + F_2, \end{cases} \quad (20)$$

式中

$$\begin{aligned} F_1 &= (2rd_1^n u_0^n, 0, \dots, 0, rc_{m-1}^n (u_m^n + u_m^{n+1}))^T, \\ F_2 &= (2rd_1^{n+1} u_0^{n+2}, 0, \dots, 0, rc_{m-1}^{n+1} (u_m^{n+1} + u_m^{n+2}))^T. \end{aligned}$$

F_1, F_2 是 $(m-1)$ 维向量。

$$G_1^{(n)} = \begin{bmatrix} Q_4^{(1)} & & & & \\ & Q_4^{(2)} & & & \\ & & \ddots & & \\ & & & Q_4^{(J-1)} & \\ & & & & Q_5^{(r)} \end{bmatrix}, \quad G_2^{(n)} = \begin{bmatrix} Q_2^{(l)} & & & & \\ & Q_4^{(2)} & & & \\ & & \ddots & & \\ & & & Q_4^{(J)} & \\ & & & & Q_3^{(r)} \end{bmatrix}. \quad (21)$$

$$Q_5^{(r)} = \begin{bmatrix} \varepsilon & -c_{m-5} & & & \\ -d_{m-4} & 3\varepsilon & -2c_{m-4} & & \\ & -2d_{m-3} & 3\varepsilon & -c_{m-3} & \\ & & -d_{m-2} & 2\varepsilon & -c_{m-2} \\ & & & -d_{m-1} & 2\varepsilon \end{bmatrix},$$

$$Q_3^{(r)} = \begin{bmatrix} 3\varepsilon & -c_{m-3} & & \\ -d_{m-2} & 2\varepsilon & -c_{m-2} & \\ & -d_{m-1} & 2\varepsilon & \end{bmatrix}.$$

4) 对于 $m-1 = 4J+3$, 我们在 t_{n+1} 层上划分 $(J+1)$ 个独立计算组。从左至右, 第 1 组至第 J 组, 每组由 4 个内点组成, 其差分格式由可以独立计算的联立方程组(13) 表示; 靠近右边界的第 $(J+1)$ 组为三点组, 并且在 $(m-2, n+1)$ 和 $(m-1, n+1)$ 2 点上使用 Crank_Nicolson 格式。在 t_{n+2} 层上划分 $(J+1)$ 个独立计算组, 从左至右, 第 1 组由 2 个内点组成, 其格式由联立方程组(14) 表示; 第 2 组至第 J 组, 每组由 4 个内点组成, 其差分格式由(13) 式表示; 第 $(J+1)$ 组是五点组, 同样, 我们在 $(m-2, n+2)$ 和 $(m-1, n+2)$ 2 个点上使用 Crank_Nicolson 格式。交替使用 t_{n+1} 层和 t_{n+2} 层上的差分格式, 于是得到

$$\begin{cases} (I + rG_1^{(n)}) U^{n+1} = (I - rG_2^{(n)}) U^n + F_1, \\ (I + rG_2^{(n+1)}) U^{n+2} = (I - rG_1^{(n+1)}) U^{n+1} + F_2, \end{cases} \quad (22)$$

式中

$$\begin{aligned} F_1 &= (2rd_1^n u_0^n, 0, \dots, 0, rc_{m-1}^n (u_m^n + u_m^{n+1}))^T, \\ F_2 &= (2rd_1^{n+1} u_0^{n+2}, 0, \dots, 0, rc_{m-1}^{n+1} (u_m^{n+1} + u_m^{n+2}))^T. \end{aligned}$$

F_1, F_2 是 $(m-1)$ 维向量。

$$G_1^{(n)} = \begin{bmatrix} Q_4^{(1)} & & & & \\ & Q_4^{(2)} & & & \\ & & \ddots & & \\ & & & Q_4^{(J)} & \\ & & & & Q_3^{(r)} \end{bmatrix}, \quad G_2^{(n)} = \begin{bmatrix} Q_2^{(l)} & & & & \\ & Q_4^{(2)} & & & \\ & & \ddots & & \\ & & & Q_4^{(J)} & \\ & & & & Q_5^{(r)} \end{bmatrix}. \quad (23)$$

2 线性稳定性分析和数值例子

在以下稳定性分析中,我们假定(16)中的系数 $a_i^n = a$ 为常数,在这种情况下, $G_1^{(n)} = G_1^{(n+1)} = G_1$, $G_2^{(n)} = G_2^{(n+1)} = G_2$. 消去 U_i^{n+1} , 公式(16)可以改写成

$$U^n = G U^{n-2},$$

此处 G 为增长矩阵, $G = (I + rG_2)^{-1}(I - rG_1)(I + rG_1)^{-1}(I - rG_2)$. 令

$$G = (I + rG_2) G (I + rG_2)^{-1} = (I - rG_1)(I + rG_1)^{-1}(I - rG_2)(I + rG_2)^{-1},$$

可以证明,式中矩阵 G_1 和 G_2 是非负实阵,由 Kellogg 引理^[11]可得

$$\|(I - rG_i)(I + rG_i)^{-1}\|_2 \leq 1 \quad (i = 1, 2).$$

于是有

$$\rho(G) = \rho(G) \leq \|(I - rG_1)(I + rG_1)^{-1}\|_2 \|(I - rG_2)(I + rG_2)^{-1}\|_2 \leq 1,$$

即格式(16)线性绝对稳定. 此处, $\rho(G)$ 和 $\rho(G)$ 分别表示矩阵 G 和 G 的谱半径.

用同样的方法可以证明格式(18), (20), (22)线性绝对稳定.

为了检验交替分组四点方法的有效性,我们给出两个数值例子.

例 1 取 Burgers 方程的精确解为^[6]

$$u(x, t) = (0.1e^{-A} + 0.5e^{-B} + e^{-C}) / (e^{-A} + e^{-B} + e^{-C}), \quad 0 \leq x \leq 1, t \geq 0, \quad (24)$$

式中

$$A = \frac{0.05}{\varepsilon}(x - 0.5 + 4.95t),$$

$$B = \frac{0.25}{\varepsilon}(x - 0.5 + 0.75t),$$

$$C = \frac{0.5}{\varepsilon}(x - 0.375).$$

表 1 例 1 中的数值解在 $t = 1.0$ 时的绝对误差 ($m = 10(h = 0.1)$, $\varepsilon = 0.1$)

x_i	方程(20) $\lambda = 5$	(D) AGE ^[6] $\lambda = 1.0$	PR AGE_MP ^[6] $\lambda = 1.0$	DR AGE_CN ^[6] $\lambda = 1.0$	精确解
0.1	2.81×10^{-4}	1.83×10^{-4}	2.9×10^{-2}	1.60×10^{-4}	0.932 745
0.2	8.00×10^{-4}	3.26×10^{-4}	6.1×10^{-2}	2.66×10^{-4}	0.911 271
0.3	7.00×10^{-7}	3.88×10^{-4}	9.4×10^{-2}	2.59×10^{-4}	0.883 314
0.4	2.69×10^{-4}	2.93×10^{-4}	1.236×10^{-1}	7.17×10^{-5}	0.847 514
0.5	5.65×10^{-4}	9.6×10^{-5}	$1.449 8 \times 10^{-1}$	3.3×10^{-4}	0.802 758
0.6	6.18×10^{-4}	6.94×10^{-4}	1.542×10^{-1}	8.94×10^{-4}	0.748 601
0.7	3.58×10^{-3}	1.19×10^{-3}	$1.476 9 \times 10^{-1}$	1.45×10^{-3}	0.685 736
0.8	3.25×10^{-3}	1.51×10^{-3}	$1.222 9 \times 10^{-1}$	1.71×10^{-3}	0.616 304
0.9	2.42×10^{-3}	1.22×10^{-3}	7.485×10^{-2}	1.34×10^{-3}	0.543 775
迭代次数		3	5	13	

例 2 取 Burgers 方程的精确解为^[9]

$$u(x, t) = \frac{\varepsilon}{1 + \varepsilon} \left(x + \tan \frac{x}{2(1 + \varepsilon)} \right), \quad 0 \leq x \leq 2, t \geq 0. \quad (25)$$

为了便于比较,我们与文[6,9]取相同的参数,对于例1,分别就 $\varepsilon = 0.1$, $h = 0.1$ 和 $\varepsilon = 0.003$, $h = 0.01$ 进行数值试验,并在表1、2中与 Evans 在[6]中用(D) AGE 方法和交替方向分组迭代方法得到的数值结果进行了比较。若记网比 $\lambda = \Delta t/h^2$,结果表明,本文的方法使用大的时间步长和网比时,仍然有很好的精度。该方法并行本性好,可以在并行机上直接使用。对于例2,数值试验结果与文[9]提供的数据比较(表3),绝对误差与文[9]高精度方法有相同的数量级,比用文[10]的方法误差小1个数量级。需要指出的是,本文提供的方法使用简便,避免了文[6,9]中存在的迭代参数的选择问题。表1~3中除去用本文提供的方法计算的数据外,其它数据均取自文[6,9]。

表2 例1在 $t = 0.5$ 时的数值解 ($m = 100(h = 0.01)$, $\varepsilon = 0.003$)

x_i	方程(22) $\lambda = 25.0$	(D) AGE ^[6] $\lambda = 1.0$	PR AGE_MP ^[6] $\lambda = 1.0$	DR AGE_CN ^[6] $\lambda = 1.0$	精确解
0.1	1.000 0	1.000 0	0.999 999	1.000 000	1.000 000
0.2	1.000 0	1.000 0	0.999 999	0.999 999	1.000 000
0.3	1.000 0	1.000 0	0.999 995	0.999 999	1.000 000
0.4	1.000 0	1.000 0	0.992 646	0.999 999	1.000 000
0.5	1.000 0	1.000 0	0.620 463	1.000 001	0.999 985
0.6	0.950 7	0.955 2	0.360 375	0.953 063	0.941 313
0.7	0.114 4	0.114 5	0.109 650	0.114 373	0.113 837
0.8	0.100 0	0.100 0	0.100 049	0.100 026	0.100 018
0.9	0.100 0	0.100 0	0.100 000	0.100 000	0.100 000
迭代次数			2	10	

表3 例2中的数值解在 $t = 2.5$ 时的绝对误差 ($m = 40(h = 0.05)$, $\varepsilon = 0.1$)

x_i	方程(20) $\lambda = 5.0$	方程(20) $\lambda = 20.0$	Jin ^[9] $\lambda = 20.0$	Wang ^[10] $\lambda = 20.0$	精确解
0.2	0.000 000 94	0.000 018	0.000 00	0.000 24	0.022 41
0.4	0.000 001 9	0.000 035	0.000 01	0.000 39	0.044 91
0.6	0.000 002 8	0.000 051	0.000 01	0.000 46	0.067 58
0.8	0.000 003 6	0.000 064	0.000 02	0.000 50	0.090 51
1.0	0.000 004 5	0.000 073	0.000 02	0.000 52	0.113 82
1.2	0.000 005 5	0.000 078	0.000 03	0.000 52	0.137 65
1.4	0.000 006 6	0.000 077	0.000 03	0.000 50	0.162 16
1.6	0.000 007 7	0.000 068	0.000 03	0.000 45	0.187 56
1.8	0.000 008 8	0.000 046	0.000 02	0.000 32	0.214 17

[参 考 文 献]

- [1] Evans D J, Abdullah A R B. Group explicit methods for parabolic equations[J]. Internat J Computer Math, 1983, 14(1): 73—105.
- [2] Evans D J. Alternating group explicit method for the diffusion equations[J]. Appl Math Modelling, 1985, 9(3): 201—206.

- [3] Evans D J, Abdullah A R B. A new explicit method for the diffusion-convection equation[J]. Comput Math Appl, 1985, **11**(1/3): 145—154.
- [4] ZHANG Bao_lin, SU Xiu_min. Alternating block explicit-implicit method for two-dimensional diffusion equation[J]. Internat J Computer Math, 1991, **38**(3/4): 241—255.
- [5] ZHANG Bao_lin, LI Wen_zhi. On alternating segment Crank-Nicolson scheme[J]. Parallel Computing, 1994, **20**(8): 897—902.
- [6] Evans D J, Sahimi M S. The numerical solution of Burgers' equations by the alternating group explicit (AGE) method[J]. Internat J Computer Math, 1989, **29**(1): 39—64.
- [7] Evans D J, Abdullah A R B. The group explicit method for the Burgers' equations[J]. Computing, 1984, **32**(3): 239—252.
- [8] 陈金甫, 张宝琳, 徐涛. 二维 Burgers 方程的 AGE 方法与并行计算[J]. 计算物理, 1998, **15**(2): 225—233.
- [9] 金承日, 刘家琦. Burgers 方程的交替分组显示迭代方法[J]. 计算物理, 1998, **15**(5): 607—613.
- [10] 王子丁, 陆金甫, 肖世江. Burgers 方程的一个分组显式格式[J]. 计算物理, 1993, **10**(4): 479—487.
- [11] Kellogg R B. An alternating direction method for operator equations[J]. J Soc Indust Appl Math (SIAM), 1964, **12**(4): 848—854.

Class of Alternating Group Method of Burgers' Equation

WANG Wen_qia

(School of Mathematics and System Science, Shandong University,
Jinan 250100, P. R. China)

Abstract: Some new Saul'yev type asymmetric difference schemes for Burgers' equation is given, by the use of the schemes, a kind of alternating group four points method for solving nonlinear Burgers' equation is constructed here. The basic idea of the method is that the grid points on the same time level is divided into a number of groups, the difference equations of each group can be solved independently, hence the method with intrinsic parallelism can be used directly on parallel computer. The method is unconditionally stable by analysis of linearization procedure. The numerical experiments show that the method has good stability and accuracy.

Key words: Burgers' equation; Saul'yev type asymmetric difference scheme; alternating group four points scheme; linear unconditional stability; parallel computation