

自适应网格技术在数值模式中的应用研究^{*}

I. 一维问题

康红文 王鹏云 徐祥德

(中国气象科学研究院, 北京 100081)

摘 要 利用变分原理的自适应网格技术被应用到三个有解析解的问题上, 它能根据问题的求解, 在解的大梯度区自动加密网格, 从而非常成功地算出了激波。通过分析发现自适应网格技术在提高精度、减少运行时间方面显示了优良的性能。

关键词: 自适应网格; 激波; 数值模拟

1 前言

中尺度数值模式在预报具有强对流的天气系统时, 不可避免地要遇到多尺度结构的问题, 比如: 强风暴、锋面降水等, 一方面要考虑其中的中小尺度结构特征; 另一方面, 还要考虑影响它们的天气背景的大尺度结构流场及其反馈作用。目前, 解决这一问题广泛采用的是套网格技术。然而, 20 多年来, 套网格技术中粗细网格之间插值所产生的误差及内边界所存在的伪反射问题, 一直令气象学家们头疼。

本文所介绍的自适应网格技术正是针对包含多尺度结构流场进行数值模拟提出的, 能从根本上去除套网格所带来的弊端。它是通过坐标变换, 把物理域上不规整网格变换到计算域上规整的网格, 使得网格的移动写进控制方程中, 并在每一步的求解中得以实现, 因而网格能连续、光滑地适应天气形式的变化而变化。

自适应网格技术首先从航空航天领域发展起来的, 80 年代在计算流体力学领域已有较广泛的应用。Dietachmayer 和 Droegemeier 在 1992 年称他们第一次将这一技术应用到气象学领域^[1], 实际上我国学者刘卓在 1991 年曾对这一技术在气象学上应用有过专门的研究^[2]。自适应网格技术分为两类: 一类是网格点数固定, 通过移动网格在解的大梯度区自动加密网格; 另一类是网格点数可以变化, 此法多用于有限元方法中。本文采用第一类方法, 它是由 Brackbill 和 Saltzman^[3]所提出, 其要点是: 根据各网格间解的梯度值取一个权函数 w , 并使得 $w \cdot \Delta x = \text{常数}$ 。这样, 某网格间上解的梯度值越大, 权函数就越大, 相应的 Δx 就越小。为达到这一目的, 先将其转化为一个变分问题, 通过求解其欧拉方程来得到新网格点位置^[4]。

以下我们用三个具有解析解的方程来进行试验。

1999-07-05 收到, 1999-09-25 收到再改稿

^{*} 国家自然科学基金资助项目 49735180、国家攀登专项“海峡两岸及邻近地区暴雨试验研究”95-专-03 和国家重点基础研究发展规划项目 G1998040907 共同资助

问题 A: 定常流动一维 Burgers 方程

$$\frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} = \frac{1}{Re} \frac{\partial^2 u}{\partial x^2}, \quad (1)$$

其中 $u \equiv u(x, t)$ (下同)。

初始条件可任取, 在此取线性分布

$$\text{当 } t = 0, \quad u = -x (-1 \leq x \leq 1),$$

边界条件

$$u = \begin{cases} 1, & x = -1 \\ -1, & x = 1 \end{cases}$$

其特解为^[5]

$$u = -\operatorname{th}\left(\frac{Re}{2}x\right)$$

本文取 $Re = 100.0$ 。

问题 B: 运动激波。

控制方程

$$\frac{\partial u}{\partial t} + \alpha \frac{\partial u}{\partial x} = \frac{1}{Re} \frac{\partial^2 u}{\partial x^2}, \quad (2)$$

初始条件

$$\text{当 } t = 0, \quad u = \begin{cases} 1, & x = 0 \\ 0, & 0 < x \leq 1 \end{cases}$$

边界条件

$$u = \begin{cases} 1, & x = 0 \\ 0, & x = 1 \end{cases}$$

其精确解为^[6]

$$u = \frac{1}{2} \left\{ \operatorname{erfc}\left(\frac{x - \alpha t}{2\sqrt{t/Re}}\right) + e^{\alpha x} \cdot \operatorname{erfc}\left(\frac{x + \alpha t}{2\sqrt{t/Re}}\right) \right\},$$

本文取 $\alpha = 10^4$, $Re = 1.0$ 。

问题 C: 非定常流动的一维 Burgers 方程

$$\frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} = \frac{1}{Re} \frac{\partial^2 u}{\partial x^2}, \quad (3)$$

初始条件

$$\text{当 } t = 0, \quad u = \frac{1}{Re} \left[x + \operatorname{tg}\left(\frac{x}{2}\right) \right], \quad \frac{1}{2} \leq x \leq \frac{3}{2},$$

边界条件

$$u = \begin{cases} \frac{1}{Re+t} \left[\frac{1}{2} + \operatorname{tg} \left(\frac{Re}{4(Re+t)} \right) \right], & x = \frac{1}{2} \\ \frac{1}{Re+t} \left[\frac{3}{2} + \operatorname{tg} \left(\frac{3Re}{4(Re+t)} \right) \right], & x = \frac{3}{2}, \end{cases}$$

其非定常流动的特解^[5]

$$u = \frac{1}{Re+t} \left[x + \operatorname{tg} \left(\frac{Re \cdot x}{2(Re+t)} \right) \right].$$

本文取 $Re = 100.0$ 。

前两个问题都是激波问题, 求解较为困难。问题 C 是一个不定常的粘性耗散的典型例子。因为它们都有解析解, 因此常用来检验一种新差分格式的优劣。在此用它们作为例子来考察自适应网格技术对解的精度是否有改进。

2 自适应网格技术

自适应网格技术包括两个方面的内容: 原参数方程的求解和网格方程的求解。

对于原参数方程, 我们以 Burgers 方程为例。首先将其从物理域坐标系转换到计算域坐标系 (ξ, τ) , 其中 $t = \tau$ 。此时方程 (1) 变成

$$\frac{\partial u}{\partial \tau} = -u_E \frac{\partial u}{\partial \xi} + \frac{1}{Re} \left(\frac{\partial x}{\partial \xi} \right)^{-2} \left[\frac{\partial^2 u}{\partial \xi^2} - \left(\frac{\partial x}{\partial \xi} \right)^{-1} \frac{\partial^2 x}{\partial \xi^2} \frac{\partial u}{\partial \xi} \right], \quad (4)$$

其中

$$u_E = \left(u - \frac{\partial x}{\partial \tau} \right) \left(\frac{\partial x}{\partial \xi} \right)^{-1} \quad (5)$$

是有效速度。从上式可以发现, 网格点的移动直接写进了方程中, 这样就避免了网格点因插值而带来的误差。

根据文献[5]对 8 个类型的差分格式比较来看, 程心一格式具有收敛较快、振幅较小的优点。本文几个问题的求解是用程心一格式。对于方程 (4), 其格式方程是

$$\begin{aligned} \tilde{u}_{j+1/2} = & \frac{1}{2} (u_j^n + u_{j+1}^n) - r(u_{j+1}^n - u_j^n) \left[\frac{1}{2} (u_j^n + u_{j+1}^n) - \frac{\partial x}{\partial \tau} \right] \left(\frac{\partial x}{\partial \xi} \right)^{-1} \\ & + \left(\frac{\partial x}{\partial \xi} \right)^{-2} \left[4s(u_{j+1}^n - 2\tilde{u}_{j+1/2} + u_j^n) \right. \\ & \left. - \frac{\Delta \tau}{Re \cdot \Delta \xi} \cdot \frac{\partial^2 x}{\partial \xi^2} \cdot (u_{j+1}^n - u_j^n) \cdot \left(\frac{\partial x}{\partial \xi} \right)^{-1} \right], \\ u_j^{n+1} = & u_j^n - r(\tilde{u}_{j+1/2} - \tilde{u}_{j-1/2}) \left[\frac{1}{2} (\tilde{u}_{j+1/2} + \tilde{u}_{j-1/2}) - \frac{\partial x}{\partial \tau} \right] \left(\frac{\partial x}{\partial \xi} \right)^{-1} \\ & + \left(\frac{\partial x}{\partial \xi} \right)^{-2} \left[4s(\tilde{u}_{j+1/2} - 2u_j^{n+1} + \tilde{u}_{j-1/2}) \right. \\ & \left. - \frac{\Delta \tau}{Re \cdot \Delta \xi} \cdot \frac{\partial^2 x}{\partial \xi^2} \cdot (\tilde{u}_{j+1/2} - \tilde{u}_{j-1/2}) \cdot \left(\frac{\partial x}{\partial \xi} \right)^{-1} \right], \end{aligned} \quad (6)$$

其中

$$r = \frac{\Delta\tau}{\Delta\xi}, \quad s = \frac{\Delta\tau}{Re \cdot \Delta\xi^2}.$$

$\partial x / \partial \xi$ 、 $\partial^2 x / \partial \xi^2$ 用中心差分格式， $\partial x / \partial \tau$ 用后差格式。因为网格间距是不断变化的，因此在求解过程中时间步长也是变化的。文献[5]给出了程心一格式方程的稳定性条件

$$r_c \leq \sqrt{\frac{1}{2}(1+4s)(1+8s)}, \quad (7)$$

其中

$$r_c = \frac{u_c \cdot \Delta\tau}{\Delta\xi}, \quad s = \frac{\Delta\tau}{Re \cdot \Delta\xi^2}.$$

对于 Burgers 方程， u_c 取 u_E 的最大值。对于问题 B， $u_c = a$ 。由此可定出每一迭代步最大许可的时间步长。我们在计算中将式 (7) 以下面更严格的形式替换，

$$r_c \leq \sqrt{\frac{1}{2}}. \quad (8)$$

并对时间步长取一个安全系数 s_t ，并要求 $0 < s_t < 1$ ，我们取 $s_t = 0.6$ 。

关于网格方程，先考虑一个变分问题，

$$\min \int \left(\frac{\partial x}{\partial \xi} \right)^2 d\xi.$$

因为 $\Delta\xi = c$ ，其欧拉方程对应一个解 $\Delta x = \text{常数}$ 。为了对网格方程加进自适应性，根据各网格间解的梯度值取一个权函数 w ，用 $w(\partial x / \partial \xi)$ 代替 $\partial x / \partial \xi$ ，这样其对应的变分问题的欧拉方程解为 $w \cdot \Delta x = \text{常数}$ 。然而，当时 $w = 0$ ，导致网格间距无穷大。因此我们重新考虑一个变分问题

$$\min \int (1 + \lambda w^2) \left(\frac{\partial x}{\partial \xi} \right)^2 d\xi, \quad (9)$$

其中 λ 是用来调节自适应程度的一个参数。其欧拉方程是

$$(1 + \lambda w^2) \frac{\partial^2 x}{\partial \xi^2} + \lambda \cdot w \frac{\partial w}{\partial \xi} \frac{\partial x}{\partial \xi} = 0, \quad (10)$$

它等价于方程

$$\frac{\partial}{\partial \xi} \left[(1 + \lambda w^2) \frac{\partial^2 x}{\partial \xi^2} \right] = 0. \quad (11)$$

令

$$f = (1 + \lambda w^2) \frac{\partial^2 x}{\partial \xi^2}, \quad (12)$$

用以下差分形式来求解方程 (11)，

$$f_{i+1/2} - f_{i-1/2} = 0, \quad (13)$$

即

$$[1 + \lambda(w_{i+1} + w_i)^2 / 4](x_{i+1} - x_i)^2 - [1 + \lambda(w_i + w_{i-1})^2 / 4](x_i - x_{i-1})^2 = 0. \quad (14)$$

令上式的左边为 R , 根据 Brackill 和 Saltzman^[3]迭代法, 有

$$R^{(k+1)} = R^{(k)} + \frac{\partial R^{(k)}}{\partial x_i^{(k)}} (x_i^{(k+1)} - x_i^{(k)}), \quad (15)$$

当 $R^{(k+1)} \rightarrow 0$ 时, $x_i^{(k+1)}$ 即为新坐标点位置。

在式 (9) 中的权函数 w 定义如下:

$$w = w_1 \left| \frac{\partial u}{\partial x} \right| + w_2 \left| \frac{\partial^2 u}{\partial x^2} \right|, \quad (16)$$

w_1 、 w_2 为可调系数。为使得网格变化更趋光滑, 对权函数 w 在空间上进行光滑处理

$$w_{i,\text{new}} = \frac{1}{2} w_{i,\text{old}} + \frac{1}{4} (w_{i-1,\text{old}} + w_{i+1,\text{old}}). \quad (17)$$

在编制程序时, 规定一个整数 m , 来确定光滑次数。为便于用 λ 对自适应程度进行调节, 我们把 w 的变化范围限制在 0 到 1 之间, 即: 先找出所有网格点上权函数的最大值, 然后分别用各网格点上的权函数与这一最大值相比, 即为该时间步上各网格点上的权函数。

至此, 整个问题就可求解。

假设已知第 n 步网格点上的 x^n 、 u^n , 由式 (8) 可求出新的时间步 Δt , 由式 (16) 和式 (17) 可求出新的权函数 w , 由式 (15) 可迭代求出新网格点位置 x^{n+1} , 由式 (6) 可求出新时间步上的 u^{n+1} 。由此又可推进求解第 $n+2$ 步上的 x 、 u 值。

3 数值结果

在计算中, 我们用记号

$$\varepsilon_{\max} = \max_{1 \leq j \leq N} |(u_j^n)_{\text{exact}} - (u_j^n)_{\text{comp}}| \quad (18)$$

和

$$\tilde{\varepsilon} = \max_{1 \leq j \leq N} |(u_j^n)_{\text{exact}}| / \varepsilon_{\max} \quad (19)$$

表示在 $n\Delta t$ 时刻数值解的最大绝对误差和最大相对误差, N 是网格点数。

先考察问题 A。

图 1 给出用不同格点数计算结果与精确解的比较。显然, 网格点太少 ($N=7$ 、 11 时), 计算值波动较大; 当 $N=21$ 时, 计算值在激波附近还有一定的抖动; 当 $N=41$ 时, 计算结果与精确解符合的已较满意。由此, 可以设想: 在激波间断区加密网格, 而

在其他区取较疏的网格, 这样即使网格点数达不到 41, 也会取得较好的结果。

图 2 给出了三个算例。 λ 表示加权程度。 $\lambda=0$ 表示没有用自适应网格。从图 2 可以看出, 当 $N=21$ 时, 如果没有采用自适应网格, 计算值在激波附近还有一定的抖动; 但当采用自适应网格技术后, 计算值已与解析解符合的很好。图 2 中横线上的 * 号表示采用自适应网格后 21 个网格点在空间上的分布, 可以发现在激波区集中了大部分网格, 而外缘无速度梯度处, 网格点很稀疏, 这也使得边界上的精确条件能很快传到计算域中。图 2 中还给出了 $N=57$, 不加自适应网格技术时的计算情况。因为我们在“ $N=21$ 、 $\lambda=100$ ”的算例中, 找出其最小网格间距为 $\Delta x_{\min}=0.035\ 22$, 如果以此作为等网格分布的间距, 需要 57 个网格点。

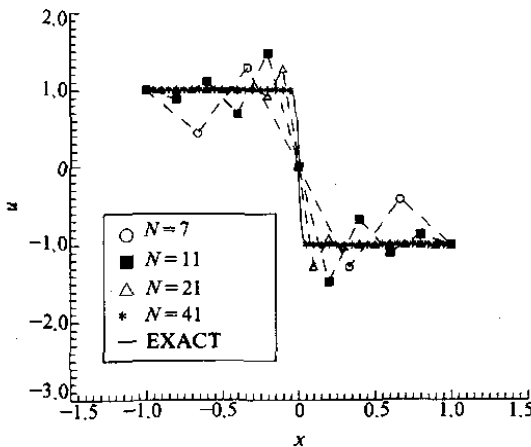


图 1 问题 A, 固定网格

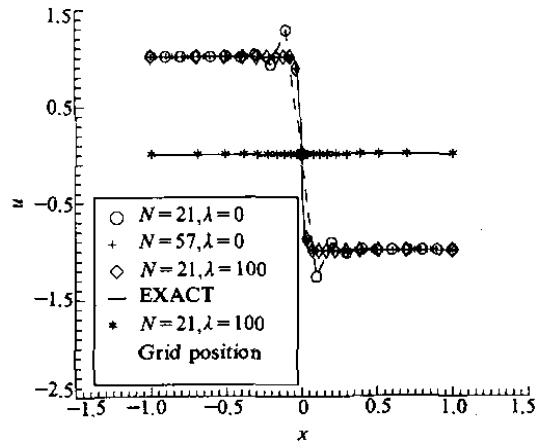


图 2 问题 A, 自适应网格算例及网格点位置

从图 2 中看不出算例“ $N=57$ 、 $\lambda=0$ ”和算例“ $N=21$ 、 $\lambda=100$ ”的优劣, 我们根据式 (19) 将其相对误差计算出来, 如图 3 所示, 可以发现在 $t>2\text{ s}$ 后, 两个算例都趋于稳定, 而且误差基本相等。根据文献[5], 我们知道程心一格式的计算误差与网格间距紧密相关, 由于图 3 中两个算例最小间距基本相等, 因而它们的计算误差基本相等。以上是两个算例相同的地方, 不同的地方是: 算例“ $N=21$ 、 $\lambda=100$ ”的收敛速度更快。作者认为这是由于边界上的精确值能更快地传递到计算域的缘故。还有一点不同的是: 算例“ $N=21$ 、 $\lambda=100$ ”的用时要远小于算例“ $N=57$ 、 $\lambda=0$ ”。因为算例“ $N=21$ 、 $\lambda=100$ ”在每一时间步上要求 21 个点上的值, 而算例“ $N=57$ 、 $\lambda=0$ ”则需求 57 个点上的值, 如果只求解一个变量, 后者运算时间是前者的 3 倍, 如果是 10 个变量的情况, 则运算量要增加 30 倍。还有一点必须得考虑: 根据式 (5) 和式 (8), 有

$$\Delta\tau \propto \Delta x_{\min},$$

每一步的时间步长 Δt 由最小网格间距来确定, Δx 越大, Δt 就越大。算例“ $N=21$ 、 $\lambda=100$ ”在激波形成之前, 网格间距要远大于 $\Delta x_{\min}=0.035\ 22$, 因而这之前其时间步长要远大于算例“ $N=57$ 、 $\lambda=0$ ”的时间步长。取 2 s 作为截止时间, 两个算例运算的时间步数分别为 154、6 210。二者相差 40 倍。另外, 如果将数组开大, 将不得不考虑内存的限制, 以及数据在计算机内部传输所需的时间。从以上分析可以看出, 自适应网格在

用时减少方面的巨大的优越性。

问题是自适应网格的加权系数如何确定。在图 4 中对此进行了考察，当 $\lambda=50$ 、 $\lambda=100$ 、 $\lambda=500$ 时计算值与解析解都符合较好。当然， λ 取值越大，网格点越集中在激波附近，但趋势渐缓。从图 4 可见，在激波附近网格间距减小到一定程度后，激波已能较为理想地算出，因而 λ 再取大已没多大的意义。如果设想在激波区外还有另外一个有速度梯度的区，那么这时就不能将网格点过于集中在激波附近，还必须使得另外一个有速度梯度的区有足够的网格以保证那里的计算精度。重要的是：对于激波、强对流系统的中小尺度结构，究竟网格间距取到多小才能较为理想地计算出它们的特征。我们知道，对于一个波长为 l 的正弦谐波，只有当网格间距取到 $(l/4)$ 或更小时，才能显示出其特征。因此，对于一个实际问题，我们在知道它的特征尺度后，通过调节 λ 来调整网格间距。这对于问题 A 来说易于调节，因为它只有一个大速度梯度区。但是，如果流场有两个或更多的大速度梯度区，那就需要综合考虑。因而，自适应网格的加权系数是与实际问题紧密联系在一起的。

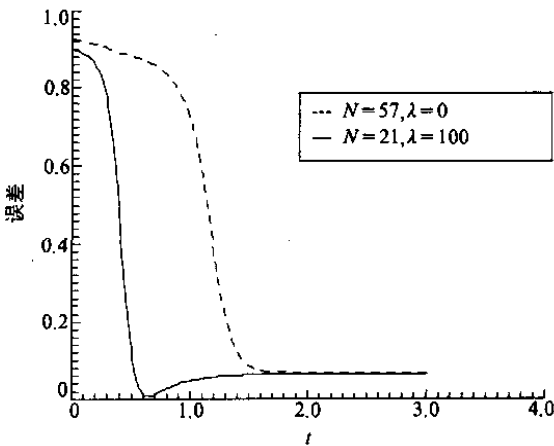


图 3 问题 A，误差分析

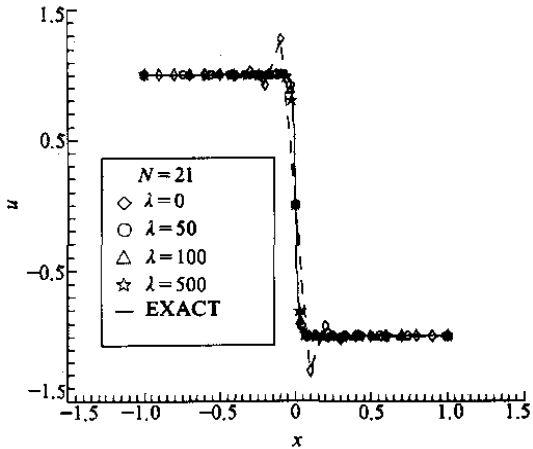


图 4 问题 A，自适应网格加权系数

现在来考察问题 B。

这是一个运动激波问题，它相当于左边界不断有单位污染浓度输入时，在高对流速度 ($\alpha=10^4$) 的带动下的对流扩散问题。

图 5 给出在等网格间距下 $t=0.000\ 03\text{ s}$ 时数值求解情况。可以发现当 $N=201$ 时，计算值在激波附近仍有较大的数值震荡。

图 6 显示出如果采用自适应网格技术时，即使 $N=51$ ，也已符合得比较好。

图 7 给出了在 $t=0.000\ 01\text{ s}$ 、 $t=0.000\ 03\text{ s}$ 、 $t=0.000\ 06\text{ s}$ 时，采用自适应网格技术与没采用自适应网格技术的计算结果与解析解对比的情况。可见随着时间的推进，不采用自适应网格技术的计算结果振荡越来越大，而采用自适应网格技术的计算结果与精确解一直符合得较好。

图 8 给出了网格间距随时间的变化。它清楚地显示出高密度网格区紧紧跟随着激波向前推进，这有效地减少了大梯度区的网格间距，因而能提高精度，非常成功地算出运动的激波。

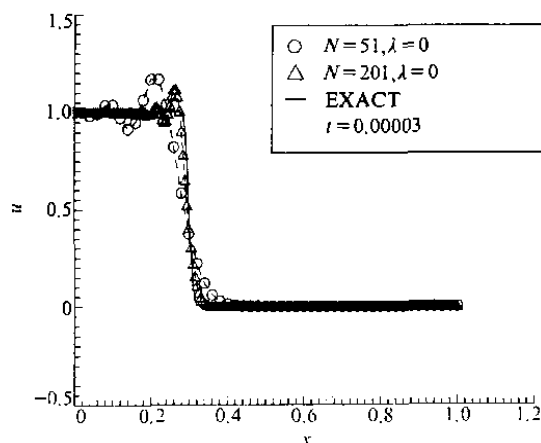


图 5 问题 B, 固定网格

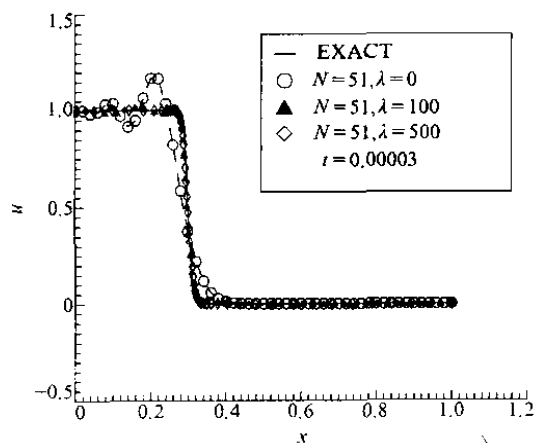


图 6 问题 B, 自适应网格与固定网格比较

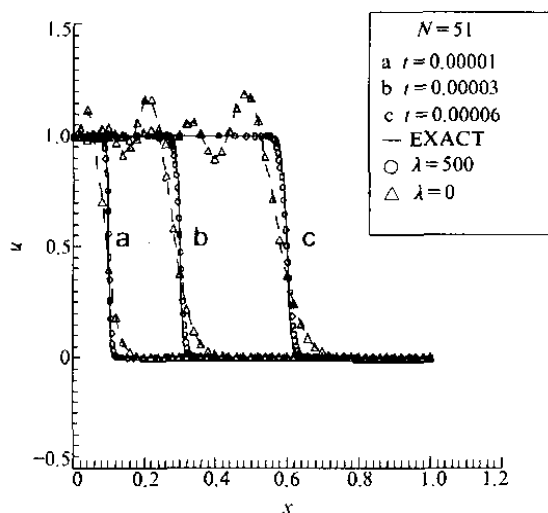


图 7 问题 B, 自适应网格

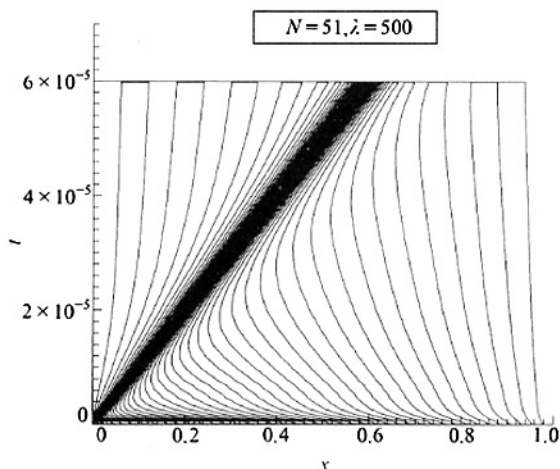


图 8 问题 B, 自适应网格点位置随时间的变化

我们又考察了不同计算格式对结果的影响。图 9 是在等网格距下, 用迎风格式和用程心一格式计算激波的情况, 发现都很不理想。当采用自适应网格技术后, 结果如图 10 所示, 都比较理想。仔细观察会发现程心一格式计算地更好些。

从以上两个问题的求解发现, 自适应网格技术对大梯度问题的求解有很大的优越性。那么对小梯度问题的求解情况又怎样呢? 我们再来考察问题 C。

这是一个不定常的粘性耗散的例子。图 11 给出了在 $t=10$ s、 $t=50$ s、 $t=100$ s 时, 采用自适应网格技术与没采用自适应网格技术的计算结果与解析解对比的情况。由于梯度不大, 在计算初始阶段, 自适应网格还没来得及把网格调整过来, 因而有一定的偏差, 但仍比没采用自适应网格技术的计算结果好, 随着计算的推进, 网格更好地调整, 计算值与解析解的符合已是很满意了。

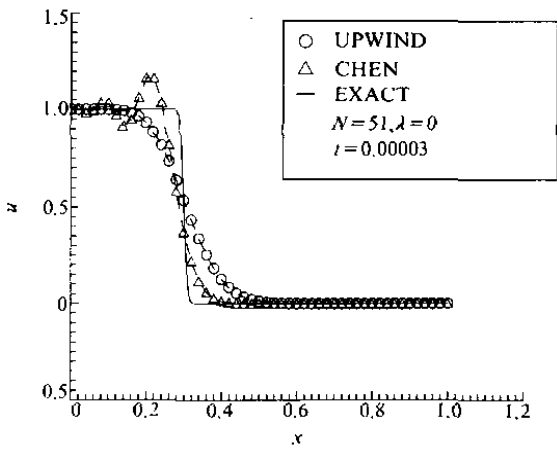


图 9 问题 B, 固定网格的不同格式的结果

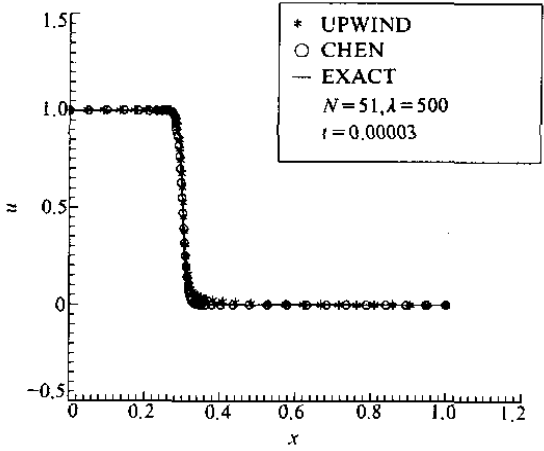


图 10 问题 B, 自适应网格的不同格式的结果

4 结论

从以上的计算结果, 有如下结论:

- (1) 自适应网格技术能有效地减少大梯度区的网格间距, 因而能减小误差, 提高精度;
- (2) 自适应网格技术能显著地降低运行时间, 并节省内存;
- (3) 自适应网格技术使得梯度不大的区域网格间距大, 因而能更快地传递边值信息, 使收敛加快;
- (4) 自适应网格技术不论对线性的对流扩散问题、还是对非线性的 Burgers 方程, 不论对定常问题、还是对非定常问题, 不论对有大梯度解的情况, 还是对没有大梯度解的情况, 其计算结果均显示出了其优良的性能。

当然, 这些结论都是在一维情况下得到的。关于此方法在二维问题上的应用, 将在另文发表。

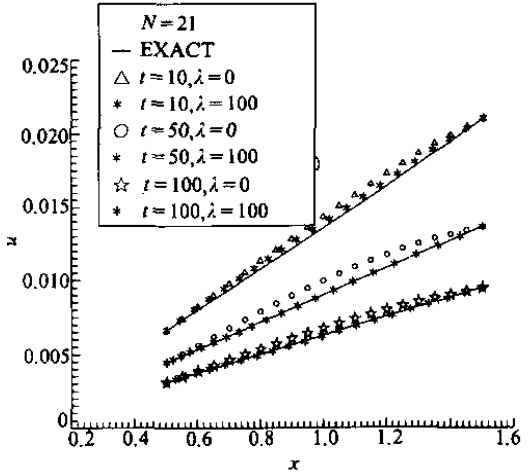


图 11 问题 C, 自适应网格与固定网格比较

参 考 文 献

- 1 Dietachmayer, G. S. and K. K. Droegemeier, Application of continuous dynamic rgrid adaptive techniques to meteorological modelling, *Mon. Wea. Rev.*, 1992, **120**, 1675~1722.
- 2 刘卓, 自适应网格在计算地球流体力学中的应用, 中国科学研究院大气物理研究所博士学位论文, 1991.
- 3 Brackbill, J. U. and J. S. Saltzman, Adaptive zoning for singular problems in two dimensions., *J. Comput. Phys.*, 1982, **46**, 342~368.
- 4 Thompson, J. F., Grid generation techniques in computational fluid mechanics, *AIAA J.*, 1984, **22**, 1505~1523.
- 5 忻孝康、朱士灿、张慧生, 一维Burgers方程的各种差分格式研究, *上海力学*, 1980, **1**, 61~80.

- 6 忻孝康、黄光伟, 对流-扩散型方程的一种简单、有效的欧拉-拉格朗日分裂格式, 空气动力学报, 1986, 4, 65~72.

A Study on Applications of Adaptive Grids Technique in Numerical Model Part I: One-Dimensional Case Test

Kang Hongwen, Wang Pengyun and Xu Xiangde

(Chinese Academy of Meteorological Science, Beijing 100081)

Abstract Adaptive grid techniques through a variational approach are applied to analytically solve three problems. More gridpoints are redistributed in the large gradient solution regions in response to numerical solutions and thus the shock can be captured successfully. The results show the techniques have excellent qualities in improving accuracy of numerical solutions and reducing CPU time.

Key words: adaptive grid techniques; shock; numerical simulation