有限体积法及关于有限差分法收敛性数值实验报告

孙泽轩
信息与计算科学专业
2016301000038

1 有限体积法

我们通过守恒律方法导出有限体积法的差分格式如下：

$$u_i^{n+1} = u_i^n + \Delta t F_i^n + \frac{\nu \Delta t}{(\Delta x)^2}(u_{i+1}^n - 2u_i^n + u_{i-1}^n).$$

对于网格的划分，我们一般有两种处理方法 — 单元中心和节点中心，这两种方法在处理边界条件时稍有不同。

1.1 Neumann 边界条件

这里我们只讨论左边界为 Neumann 边界条件的情况，即 $\partial_x v(0,t) = g(t), \forall t > 0$, 对于右边界为 Neumann 边界条件的情况可类似讨论。利用有限体积法采取单元中心的方法，在边界点我们得到如下格式：

$$u_1^{n+1} = u_1^n + \Delta t F_1^n + \frac{\nu \Delta t}{\Delta x^2}(u_2^n - u_1^n) - \frac{\nu \Delta t}{\Delta x} g^n,$$

通常情况下为了编程的方便我们增加一个 ghost point，使得边界点的更新格式与内部相同，即取

$$u_0^n = u_1^n - \Delta g^n.$$

而对于节点中心，同样我们容易得到以下格式：

$$u_0^{n+1} = u_0^n + \Delta t F_0^n + \frac{2\nu \Delta t}{\Delta x^2}(u_1^n - u_0^n) - \frac{2\nu \Delta t}{\Delta x} g^n,$$

同单元中心类似，为了编程的方便我们增加一个 ghost point，取

$$u_{-1}^n = u_1^n - 2\Delta g^n.$$

从两个格式 ghost point 的取法我们可以看出在理论上，对于 Neumann 边界条件我们采取单元中心可以得到更高的精度。

1.2 Dirichlet 边界条件

同上，对于 Dirichlet 边界条件我们也只考虑左边界为 Dirichlet 边界条件的情况。利用有限体积法采取单元中心的方法，在边界点我们得到如下格式：

$$u_1^{n+1} = u_1^n + \Delta t F_1^n + \frac{\nu \Delta t}{\Delta x^2}(u_2^n - 3u_1^n + 2g^n).$$

而对于节点中心，我们得到以下格式：

$$u_0^{n+1} = g^{n+1}.$$

显然对于 Dirichlet 边界条件采用节点中心我们可以得到精确的解，单元中心不可避免地会造成精度的损失。

1.3 算例

为了测试使用单元中心和节点中心在不同边界条件下的实际效果，我们对如下算例进行了数值实验：

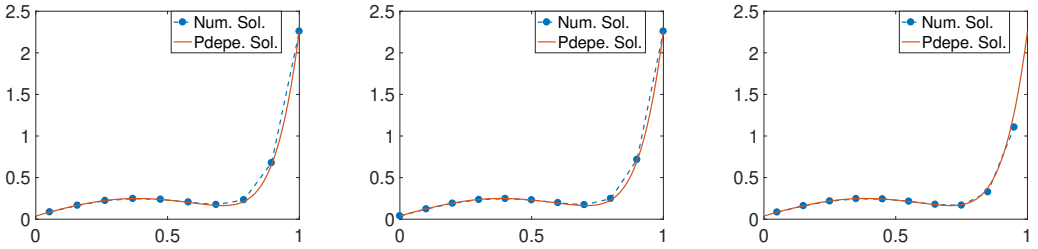
$$\begin{cases} v_t = \nu v_{xx} + F(x,t), & x \in (0,1), t > 0 \\ v(x,0) = f(x), & x \in (0,1) \\ v_x(0,t) = a(t), v(1,t) = b(t), & t \geq 0 \end{cases}$$

其中 $\nu = 0.1, f(x) = x(1-x), a(t) = 10sint, b(t) = 4sin6t, F(x,t) = sin2\pi x sin4\pi t.$ 我们主要测试了如下三种处理该算例边界条件的格式：

- 左边界单元中心，右边界节点中心，我们简记为 lcrb
- 左右边界均为节点中心，我们简记为 lbrb
- 左右边界均为单元中心，我们简记为 lcrc

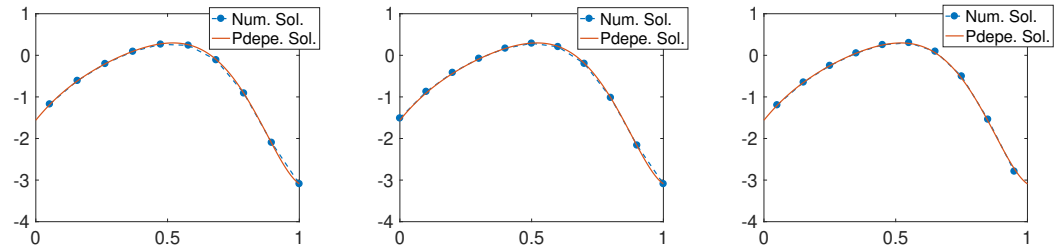
为了比较我们得到的解的精度，我们以 Matlab 的 pdepe 函数得到的解为基准进行比较。同时为了方便比较上述三种格式得到的解与 pdepe 得到的解，我们编写了 matlab_sol 函数封装了 pdepe 函数。我们编写了三个函数 lcrb(Dt,Tend,choice), lbrb(Dt,Tend,choice), lcrc(Dt,Tend,choice) 分别实现上述三种格式，其中 Dt 为时间步长，Tend 为终止时刻，choice 为 1 时作出当前格式得到的解与 pdepe 得到的解，choice 为 2 时作出当前格式取值点的值与 pdepe 在该点值的差的绝对值，即误差图，同时上述三个函数会输出当前格式网格点误差的平均值，即整体误差值除以网格点数。

首先我们直观地画出在 $t = 0.1, 0.9, 2.0,$ 时间步长为 0.01 条件下三种格式得到的解与 pdepe 得到的解的图像，得到的结果如下：



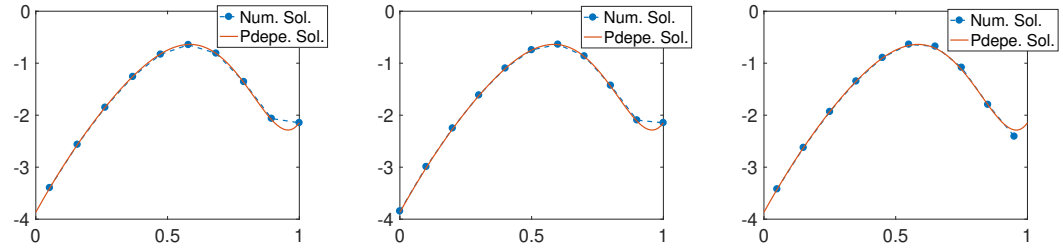
(a) lcrb (b) lbrb (c) lcrc

Figure 1: $t = 0.1$ 时, 时间步长为 0.01, 三种格式的结果与 pdepe 的结果比较图



(a) lcrb (b) lbrb (c) lcrc

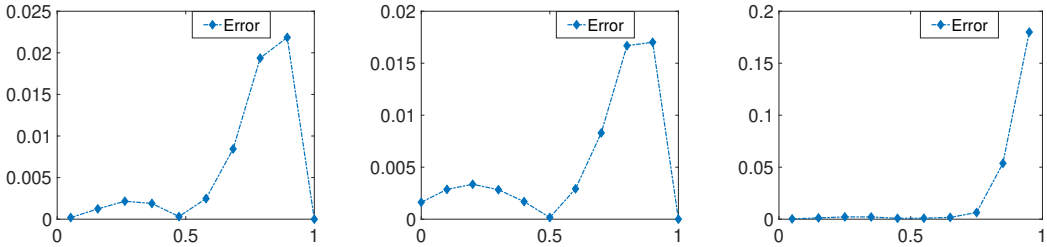
Figure 2: $t = 0.9$ 时, 时间步长为 0.01, 三种格式的结果与 pdepe 的结果比较图



(a) lcrb (b) lbrb (c) lcrc

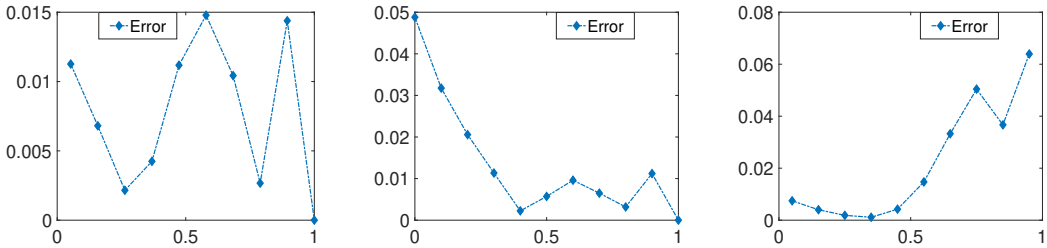
Figure 3: $t = 2.0$ 时, 时间步长为 0.01, 三种格式的结果与 pdepe 的结果比较图

从图像上我们可以看到三种格式均具有较高的精度，除了 lcrc 在最后一个取值点与 pdepe 的解有较明显的误差。这显然是由右边界采用单元中心格式造成的。为了更好的比较三种格式的精度，我们用编写的三个函数画出三种格式在 $t = 0.1, 0.9, 2.0$, 时间步长为 0.01 条件下取值点的误差图，结果如下：



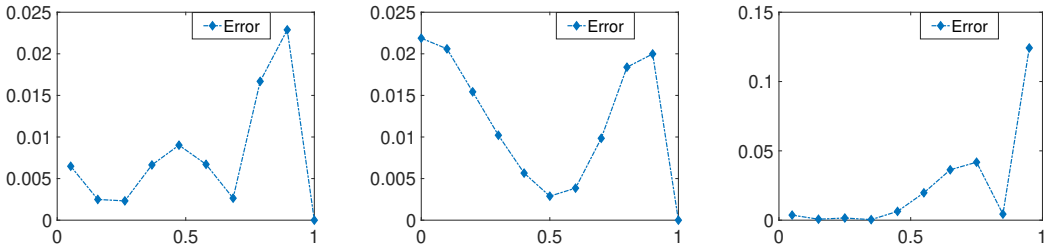
(a) lcrb (b) lbrb (c) lcrc

Figure 4: $t = 0.1$ 时, 时间步长为 0.01, 三种格式的误差图



(a) lcrb (b) lbrb (c) lcrc

Figure 5: $t = 0.9$ 时, 时间步长为 0.01, 三种格式的误差图



(a) lcrb (b) lbrb (c) lcrc

Figure 6: $t = 2.0$ 时, 时间步长为 0.01, 三种格式的误差图

从上述结果可以看出在 $t = 0.1$ 时，lcrb 和 lbrb 两种格式误差的最大值都差不多，大概在 0.02 左右，但 lcrb 与 lbrb 相比在左边界的取值点更小的误差，而 lcrc 格式由于在右边界采用单元中心格式导致了最后一个靠近右边界的取值点有较大的误差。在 $t = 0.9$ 时，lcrb 格式在三种格式中整体表现最优，所有点的误差都在 0.015 以下，而 lbrb 在左边界有较大误差，lcrc 在右边界处有更大的近 0.07 的误差。在 $t = 2.0$ 时，lcrb 和 lbrb 的效果差不多，但 lcrb 在左边界的误差

要小得多，而 lcrc 在右边界的巨大误差使得 lcrc 格式整体效果较前两种格式要差一些。

为了更好的比较三种格式在 $t = 0.1, 0.9, 2.0,$ 时间步长为 0.01 条件下整体误差值的大小，我们计算出三种格式在上述不同条件下网格点的平均误差值，得到的结果如下表：

格式	$t = 0.1$	$t = 0.9$	$t = 2.0$
lcrb	0.0058	0.0078	0.0076
lbrb	0.0052	0.0137	0.0117
lcrc	0.0249	0.0218	0.0239

Table 1: 三种格式在不同条件下的网格点平均误差值

从上表我们可以看出除了 $t = 0.1$ 时 lbrb 效果优于 lcrb，其余情况下 lcrb 均是三种格式中最优的，lbrb 次之，而 lcrc 在三个时间下误差都是最大的。$t = 0.1$lbrb 最优可能是由计算偶然性以及时间较短格式的缺点误差还未完全累积造成的。

1.4 实验结论

从上述实验我们可以看出对于该算例 lcrb 是最优的格式，lbrb 次之，lcrc 最差。这与我们的理论分析是一致的，即对于 Neumann 边界条件我们应采用单元中心格式，而对于 Dirichlet 边界条件我们应采用节点中心格式。

2 关于于有限差分法的收敛性

一般的我们称差分方程 $L_i^n u_i^n = F_i^n$ 对微分方程 $Lv = F$ 是逐点收敛的，如果对于任意 x 和 t，

$$\lim_{\substack{\triangle x\to 0,\triangle t\to 0\\(k\triangle x,(n+1)\triangle t)\to(x,t)}} u_i^n = v(x,t)$$

我们已经证明了对于如下的偏微分方程初边值问题：

$$\begin{cases} v_t = \nu v_{xx}, & x\in(0,1), t>0,\\ v(x,0) = f(x), & x\in[0,1],\\ v(0,x) = v(1,t) = 0, & t\geq 0 \end{cases}$$

其中 $\nu = 0.1, f(x) = sin2\pi x$。当我们采用格式：

$$\begin{cases} u_i^{n+1} = (1-2r)u_i^n + r(u_{i+1}^n + u_{i-1}^n), & n\geq 0, i = 1,2,...,N-1\\ u_0^{n+1} = u_N^{n+1}, & n\geq 0\\ u_i^0 = f(i\triangle x) \end{cases}$$

其中 $r = \nu\triangle/\triangle x^2, 0\leq r\leq \frac{1}{2}$。则差分格式逐点收敛到微分方程的解。接下来我们想通过数值实验来验证这一事实。

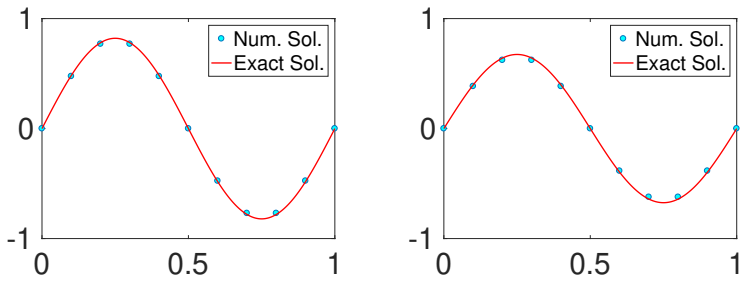
2.1 数值实验

我们只计算该微分方程在 $t = 0.05$ 和 $t = 0.1$ 时的数值结果。我们只在如下三种网格步长和时间步长下进行模拟并比较结果：

- $\triangle x = 0.1, \triangle t = 0.05$
- $\triangle x = 0.05, \triangle t = 0.00125$
- $\triangle x = 0.1, \triangle t = 0.0005$

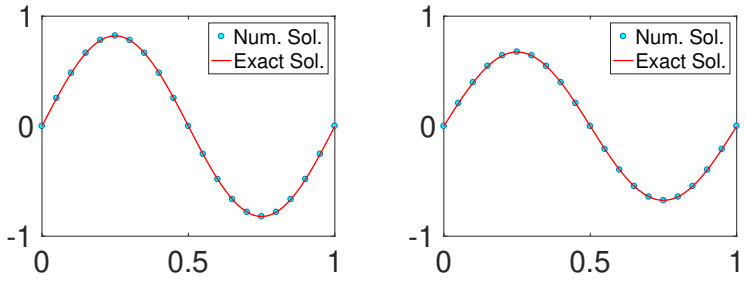
我们编写了 FDM _eg3(DX,DT,TEND,choice) 函数来实现差分方法，其中 DX 为网格步长，DT 为时间步长，TEND 为终止时间，choice 取 1 时作出数值解与精确解的比较图，为 2 时作出网格点的误差图，同时该函数会输出网格点的平均的误差值，即总误差值除以格点数。

首先我们作出在 $t = 0.05$ 和 $t = 0.1$ 时在三种步长下数值解与精确解的比较图，结果如下：



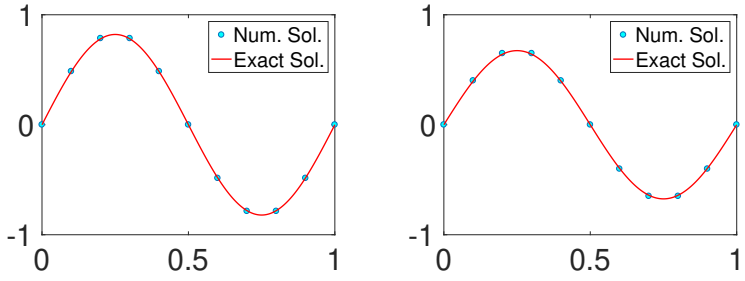
(a) $t = 0.05$ (b) $t = 0.1$

Figure 7: $\triangle x = 0.1, \triangle t = 0.05$，两个时间下数值解与精确解的比较图



(a) $t = 0.05$ (b) $t = 0.1$

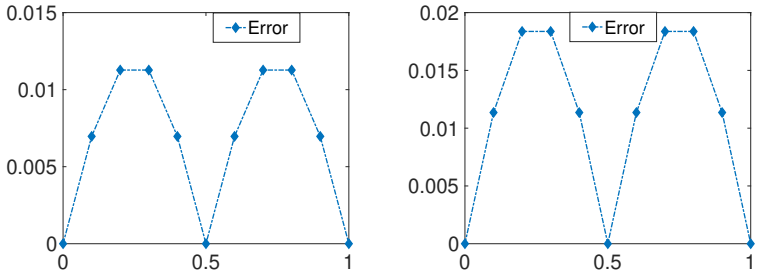
Figure 8: $\triangle x = 0.05, \triangle t = 0.00125$，两个时间下数值解与精确解的比较图



(a) $t = 0.05$ (b) $t = 0.1$

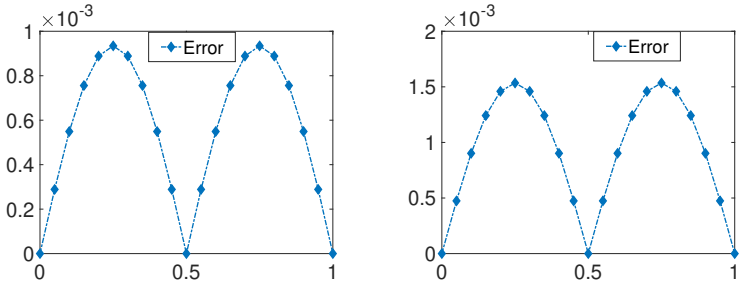
Figure 9: $\triangle x = 0.1, \triangle t = 0.0005$，两个时间下数值解与精确解的比较图

从图上看三种步长都具有较好的精度，为了更好地比较三种步长的精度，我们画出误差图，结果如下：



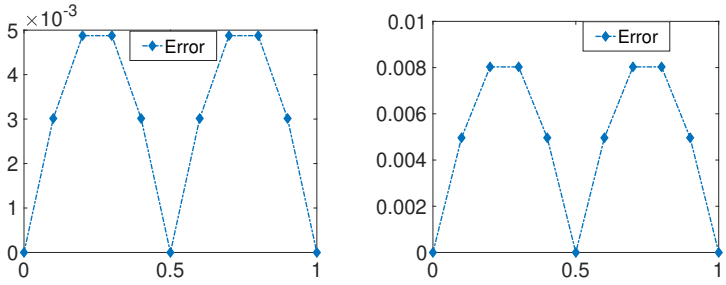
(a) $t = 0.05$ (b) $t = 0.1$

Figure 10: $\triangle x = 0.1, \triangle t = 0.05$，两个时间下的误差图



(a) $t = 0.05$ (b) $t = 0.1$

Figure 11: $\triangle x = 0.05, \triangle t = 0.00125$，两个时间下的误差图



(a) $t = 0.05$ (b) $t = 0.1$

Figure 12: $\triangle x = 0.1, \triangle t = 0.0005$，两个时间下的误差图

我们可以看到对于三种步长，随着时间的增加误差在增大。而第二种步长相比较第一种步长同时减小了网格步长和时间步长后误差更小；第三种步长相

比较第一种步长在只减小了时间步长的情况下也使得误差减小。第二种步长整体来看比第三种步长精度更高。为了更直观地比较三种步长的误差大小，我们计算出不同情况下不同步长网格点的平均误差值，结果如下表：

步长种类	$t = 0.05$	$t = 0.1$
$\triangle x = 0.1, \triangle t = 0.05$	0.0066	0.0108
$\triangle x = 0.05, \triangle t = 0.00125$	5.62e-04	9.23e-04
$\triangle x = 0.1, \triangle t = 0.0005$	0.0029	0.0047

Table 2: $t = 0.05$ 和 $t = 0.1$ 时，三种步长网格点的平均误差值

从表中数据可以看出第二种和第三种步长要明显优于第一种步长，且第二种步长得到误差最小，第三种次之，第一中最大。

2.2 实验结论

从我们做的数值实验可以看出，当 $\triangle x \to 0, \triangle t \to 0$ 时，差分格式得到的数值解的确会逐点收敛到微分方程的解。