

# 曲率流

李森 胡滌非 梁凯毅

March 10, 2021

## 1 曲率流问题介绍和研究方法

### 1.1 曲率流

如图1所示，曲率流是一种常见的物理现象，当不同流相相互接触时，由于分子间的作用力，由Young-Laplace定理

$$\Delta P \equiv P_{in} - P_{out} = \gamma(\kappa_1 + \kappa_2), \quad (1)$$

相交面的压力会和曲率相关，进而呈现出曲率流的现象。



图 1: Bubble in air

如图2所示，平均曲率流为曲率流的一种，即处在界面上的每一点，其运动方向沿法线方向，若为内凹则方向向外，外凸则向内，大小为该点处的平均曲率。对于二维情况，令 $s$ 为界面曲线的弦长参数， $\kappa$ 为带符号曲率， $\vec{n}$ 为单位外法向，则对曲线上每个点有如下方程

$$\frac{\partial \mathbf{X}}{\partial t} = \frac{\partial^2 \mathbf{X}}{\partial s^2} = -\kappa \vec{n}. \quad (2)$$

为了后续表达的方便，将点集原本的二维矩阵形式进行向量化，即

$$\mathbf{X} = \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix}, \quad \mathbf{f} = \begin{bmatrix} f_x \\ f_y \end{bmatrix} = \begin{bmatrix} -\kappa n_x \\ -\kappa n_y \end{bmatrix}. \quad (3)$$

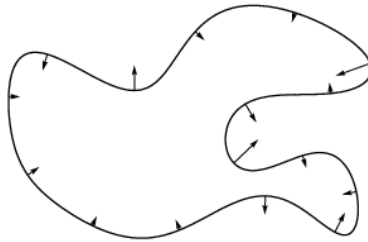


图 2: Mean curvature flow in 2D situation

## 1.2 研究方法

### 1.2.1 空间离散

为了处理方程(2)右端项，即对问题进行空间离散，我们可以从两个方向考虑，一种是计算曲率和外法向量，用到的方法是第2章中改进的HFES算法。另一种是将方程看作抛物方程处理，对右端项采用中心差分格式离散，具体见第4章。

### 1.2.2 界面追踪

将MARS方法与曲率流问题结合。MARS方法（如图3）利用殷集的映射和调整来追踪动态区域，通过调整示踪点与网格尺度关系，从而达到任意阶精度<sup>[5, 6]</sup>。

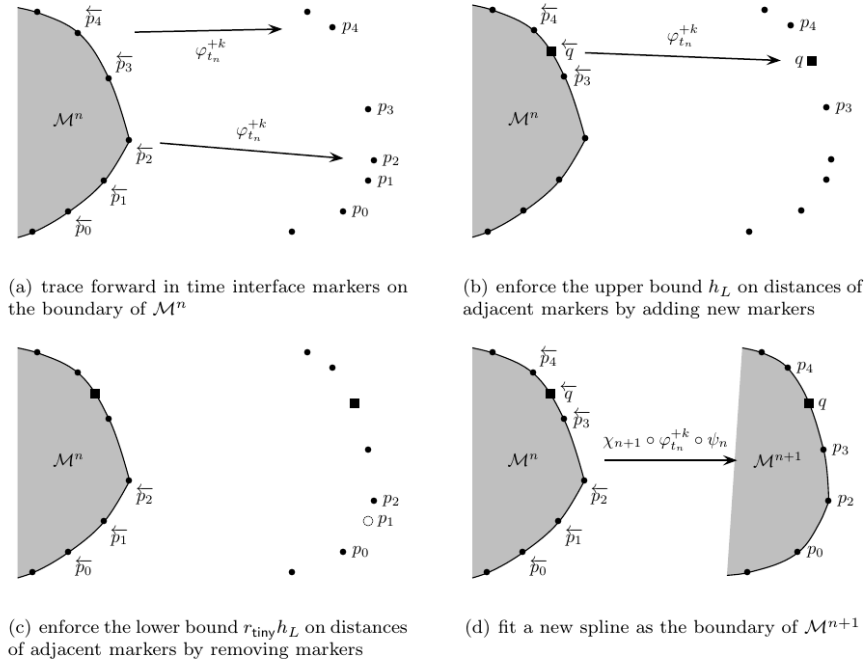


图 3: MARS方法基本步骤

### 1.2.3 时间积分格式

平均曲率流问题实际上为一个抛物方程，因此在时间步上积分时，普通的显式方法是低效且难以稳定的。若用向后欧拉方法，令  $\mathbf{X}^{n+1}$  为  $\mathbf{X}(t_{n+1})$  的近似， $k$  为时间步长，则在每一个时间步内，需要解非线性方程组

$$\frac{\mathbf{X}^{n+1} - \mathbf{X}^n}{k} = \mathbf{f}(\mathbf{X}^{n+1}, t_{n+1}). \quad (4)$$

为此，使用牛顿迭代，迭代格式为

$$\mathbf{X}_{i+1}^{n+1} = \mathbf{X}_i^{n+1} - \frac{k \mathbf{f}(\mathbf{X}_i^{n+1}, t_{n+1}) - \mathbf{X}_i^{n+1} + \mathbf{X}^n}{k \nabla \mathbf{f}(\mathbf{X}_i^{n+1}, t_{n+1}) - I}, \quad (5)$$

注意此处  $i$  表示迭代步数。

$\nabla f(\mathbf{X}, t)$ 即为 $f$ 关于 $\mathbf{X}$ 的 $2N \times 2N$ 的Jacobian矩阵, 其中 $N$ 为整条曲线上示踪点个数。即

$$J_f = \left[ \begin{array}{c|c} \left\{ \frac{\partial f_{x,i}}{\partial x_j} \right\} & \left\{ \frac{\partial f_{x,i}}{\partial y_j} \right\} \\ \hline \left\{ \frac{\partial f_{y,i}}{\partial x_j} \right\} & \left\{ \frac{\partial f_{y,i}}{\partial y_j} \right\} \end{array} \right]_{1 \leq i,j \leq N}. \quad (6)$$

其具体推导将在后文进行阐述。

由以上描述可以看出, 隐式方法的求解过程非常繁琐且耗时, 在第5章我们将另外介绍显式Runge-Kutta-Chebyshev (RKC) 方法来作为隐式方法的替代。

## 2 基于非定值 $h_\kappa$ 的HFES

本章将介绍基于非定值 $h_\kappa$ 的HFES, 用以计算方程(2)右端所需要的曲率和单位外法向量。

### 2.1 算法描述

为提高计算效率, 减少计算控制体体积的次数, 将原先的HFES方法改为,

1. 先将曲线切分成若干段,
2. 对每段做一次旋转, 使曲线段成为局部坐标系 $(\xi, \eta)$ 下纵坐标关于横坐标的一个函数 $H(\xi)$ ,
3. 通过对函数的积分计算该段曲线上的点在局部坐标下的一、二阶导数, 并利用一、二阶导数计算带符号曲率以及单位内法向,
4. 最后将局部坐标下的结果变回全局坐标下的结果。

在利用高度函数的积分计算一、二阶导数时, 先利用示踪点作为网格控制点在局部坐标系下生成一系列小控制体, 在计算该段曲线上每个点处的一二阶导数时, 对生成的小控制体进行拼接, 得到每部分宽度接近给定 $h_\kappa$ 的大控制体, 如图4, 利用得到的大控制体内物质体积计算所求点的导数。通过这种方式, 每个曲线片段只需要计算一次体积即可, 极大减少了计算量。但因为此时得到的控制体宽度与曲线示踪点位置相关, 为非定长且非对称, 因此系数需要根据具体得到的控制体宽度计算得到。

以2阶精度方法为例, 为了得到精度为2阶的曲率, 需要精度为2阶的一阶导数与二阶导数。为得到在局部坐标系下对 $H'(0)$ 的一个二阶的近似, 需要对局部坐标系下的曲线方程的泰勒展开式分别在如图5(a)三个控制体上进行积分, 使得 $C_1$ 与控制体宽度满足如方程(7)关系, 因为控制体为非对称, 对于中心控制体需要考虑左右宽度不同, 所以中心控制体宽度为 $h_{\kappa_0}^{0-} + h_{\kappa_0}^{0+}$

$$C_{1,-1} \int_{-h_{\kappa_0}^{0-} - h_{\kappa_1}^{0-}}^{-h_{\kappa_0}^{0-}} H(\xi) d\xi + C_{1,0} \int_{-h_{\kappa_0}^{0-}}^{h_{\kappa_0}^{0+}} H(\xi) d\xi + C_{1,1} \int_{h_{\kappa_0}^{0+}}^{h_{\kappa_0}^{0+} + h_{\kappa_1}^{0+}} H(\xi) d\xi = H'(0) + O(h_\kappa^2). \quad (7)$$

此时对 $H(\xi)$ 进行泰勒展开, 使得 $H(0), H''(0)$ 项系数为0,  $H'(0)$ 项系数为1, 此时 $C_1$ 需与 $h_\kappa^{-2}$ 同阶, 即可满足误差为 $O(h_\kappa^2)$ 。

而对于二阶导数, 需要满足

$$\begin{aligned} & C_{2,-1} \int_{-h_{\kappa_0}^{0-} - h_{\kappa_1}^{0-}}^{-h_{\kappa_0}^{0-}} f(x) dx + C_{2,0-} \int_{-h_{\kappa_0}^{0-}}^0 f(x) dx \\ & + C_{2,0+} \int_0^{h_{\kappa_0}^{0+}} f(x) dx + C_{2,1} \int_{h_{\kappa_0}^{0+}}^{h_{\kappa_0}^{0+} + h_{\kappa_1}^{0+}} f(x) dx \\ & = f''(0) + O(h_\kappa^2). \end{aligned} \quad (8)$$

注意到这里使用了四个控制体, 因为当 $H''(0)$ 项系数为1, 则 $C_2$ 与 $h_\kappa^{-3}$ 同阶, 为达到2阶精度, 则需要 $H(0), H'(0), H^{(3)}(0)$ 项的系数均为0, 所以需要四个控制体, 如图5(b)。

对于更高阶数, 方法都是相似的, 只需要增加控制体个数并在其上做积分即可, 此处不再详细展开。

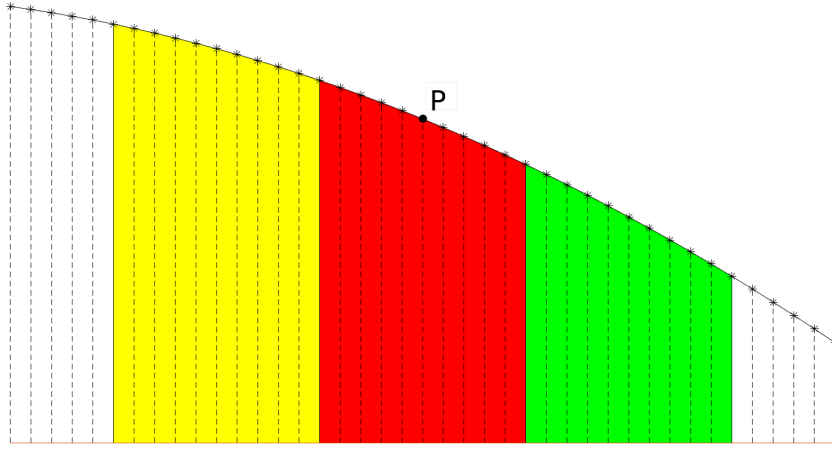


图 4: 曲线上每个\*代表曲线的示踪点, 用其作为控制体网络的生成点得到一系列小控制体, 而涂红黄绿三色区域的代表利用有限体积方法计算导数时用到的控制体, 其由先前生成的小控制体拼接而成, 从而对于每一个曲线片段, 只需要计算一次控制体体积即可, 计算每个点处的导数时, 用到的控制体由小控制体拼接得到。

## 2.2 曲率估算测试结果

以下为使用4阶方法计算得到的曲率

### 2.2.1 单位圆测例

第一个测例为单位圆, 即半径为1的圆。结果如表1所示, 达到了4阶收敛, 且计算速度提高明显, 5000个点的曲率计算用时0.88秒。

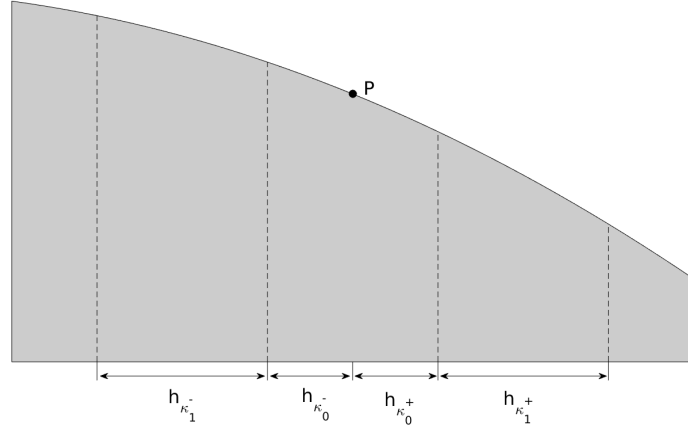
$h_\kappa$	1/20	收敛率	1/40	收敛率	1/80
曲率	6.52e-5	3.83	4.58e-6	3.71	3.49e-7
单位内法向量	3.19e-5	4.02	1.97e-6	4.01	1.22e-7

Table 1: 表中为四阶方法计算单位圆上点曲率以及单位内法向的误差以及收敛阶, 记录的误差为绝对误差的无穷范数, 一共使用了5000个示踪点表示圆。

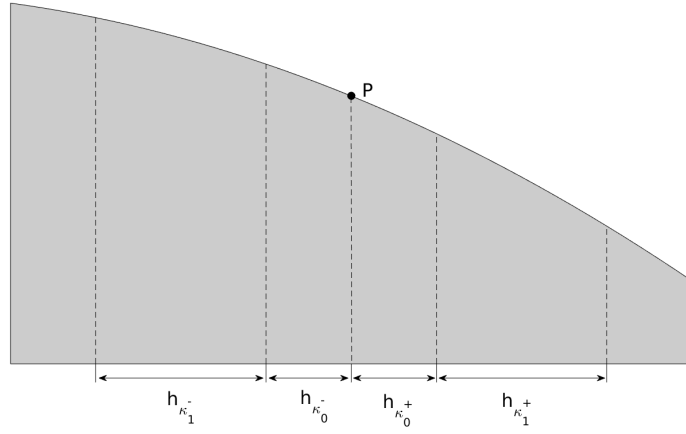
### 2.2.2 星形图案

除了圆之外, 计算了星形图案的曲率以及单位法向量。星形线的参数方程可通过方程(9)得到, 其图像为图6。计算结果如表2所示, 也达到了四阶收敛, 计算50000个点用时约为35秒。

$$\begin{cases} x = (1 + 0.3 \cos(7t)) \cos(t) \\ y = (1 + 0.3 \cos(7t)) \sin(t) \\ t \in [0, 2\pi] \end{cases} \quad (9)$$



(a) 在计算一阶导数时，为了误差达到二阶精度，需满足 $H(0)$ ， $H''(0)$ 系数为0， $H'(0)$ 系数为1，有三个方程，因此需要三个控制体



(b) 计算二阶导数时，为了误差达到二阶精度，需满足 $H(0)$ ， $H'(0)$ ， $H^{(3)}(0)$ 系数为0， $H''(0)$ 系数为1，因此需要四个方程，需要四个控制体，可以将原本计算一阶导数时的中心控制体一分为二，变为两个控制体

图 5: 二阶精度变长度HFES所用控制体示意图

$h_{\kappa}$	1/160	收敛率	1/320	收敛率	1/640
曲率	0.17	3.64	1.35e-2	3.83	9.55e-4
单位内法向量	5.72e-3	3.60	4.71e-4	3.87	3.22e-5

Table 2: 表中为四阶方法计算如图6星形图案上点的曲率以及单位内法向的误差以及收敛阶，记录的误差为绝对误差的无穷范数，一共使用了50000个示踪点表示星形。

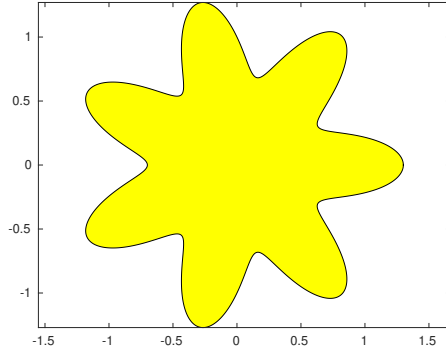


图 6: 图为通过方程(9)得到的图形, 黄色部分为其内部

### 2.3 Jacobian矩阵的计算

为了通过牛顿法迭代求解, 需要计算Jacobian矩阵, 如果令第 $i$ 点处曲率流作用的力 $F_i$ 为

$$F_i = \begin{bmatrix} f_i \\ g_i \end{bmatrix}. \quad (10)$$

为了求得Jacobian矩阵, 则需要对每个 $F_i$ 与第 $j$ 点求得

$$\begin{bmatrix} \frac{\partial f_i}{\partial x_j} & \frac{\partial f_i}{\partial y_j} \\ \frac{\partial g_i}{\partial x_j} & \frac{\partial g_i}{\partial y_j} \end{bmatrix}. \quad (11)$$

在计算第 $i$ 点处曲率时, 为了得到合适的高度函数, 需要将第 $i$ 点所处的曲线段做一次旋转, 将全局坐标 $(x_j, y_j)$ 通过旋转矩阵 $R$ 的作用, 变为局部坐标 $(\xi_j, \eta_j)$ , 从而将曲线段变为局部坐标下的曲线函数, 并利用第一部分中提到的方法, 通过高度函数计算局部坐标下的单位内法向量和带符号曲率, 再将得到的结果变为全局坐标下的结果。因此计算Jacobian矩阵时也需要利用这一过程进行计算。

为了后续表示方便, 此处令 $H'_i, H''_i$ 分别为在计算第 $i$ 点处曲率和单位法向量时, 所用到的局部坐标下高度函数的一、二阶导数。此时可以将局部坐标下点 $i$ 的带符号曲率 $\kappa_i$ 和单位内法向 $\mathbf{n}_i$ 分别写为

$$\kappa_i = \frac{H''_i}{(1 + (H'_i)^2)^{3/2}}, \quad \mathbf{n}_i = \frac{1}{(1 + (H'_i)^2)^{1/2}} \begin{bmatrix} H'_i \\ -1 \end{bmatrix}. \quad (12)$$

因此在局部坐标系下曲率流的力为

$$\begin{bmatrix} f_i^{local} \\ g_i^{local} \end{bmatrix} = -\kappa_i \mathbf{n}_i = \begin{bmatrix} \frac{-H''_i H'_i}{(1 + (H'_i)^2)^2} \\ \frac{H''_i}{(1 + (H'_i)^2)^2} \end{bmatrix}. \quad (13)$$

因为先前令从全局坐标 $(x_j, y_j)$ 到局部坐标 $(\xi_j, \eta_j)$ 的旋转矩阵为 $R$ , 则有

$$\begin{bmatrix} f_i \\ g_i \end{bmatrix} = R' * \begin{bmatrix} f_i^{local} \\ g_i^{local} \end{bmatrix}. \quad (14)$$

同时有

$$\begin{bmatrix} \frac{\partial \xi_j}{\partial x_j} & \frac{\partial \xi_j}{\partial y_j} \\ \frac{\partial \eta_j}{\partial x_j} & \frac{\partial \eta_j}{\partial y_j} \end{bmatrix} = R. \quad (15)$$

由上述关系式，得到对于式（11），可以将其写为

$$\begin{bmatrix} \frac{\partial f_i}{\partial x_j} & \frac{\partial f_i}{\partial y_j} \\ \frac{\partial g_i}{\partial x_j} & \frac{\partial g_i}{\partial y_j} \end{bmatrix} = R' * \begin{bmatrix} \frac{\partial f_i^{local}}{\partial \xi_j} & \frac{\partial f_i^{local}}{\partial \eta_j} \\ \frac{\partial g_i^{local}}{\partial \xi_j} & \frac{\partial g_i^{local}}{\partial \eta_j} \end{bmatrix} * R. \quad (16)$$

而对于中间部分，即局部坐标下的Jacobian矩阵

$$\begin{bmatrix} \frac{\partial f_i^{local}}{\partial \xi_j} & \frac{\partial f_i^{local}}{\partial \eta_j} \\ \frac{\partial g_i^{local}}{\partial \xi_j} & \frac{\partial g_i^{local}}{\partial \eta_j} \end{bmatrix}. \quad (17)$$

可以利用链式法则将其写为

$$\begin{aligned} \frac{\partial f_i^{local}}{\partial \xi_j} &= \frac{\partial f_i^{local}}{\partial H_i''} \cdot \frac{\partial H_i''}{\partial \xi_j} + \frac{\partial f_i^{local}}{\partial H_i'} \cdot \frac{\partial H_i'}{\partial \xi_j}, \\ \frac{\partial f_i^{local}}{\partial \eta_j} &= \frac{\partial f_i^{local}}{\partial H_i''} \cdot \frac{\partial H_i''}{\partial \eta_j} + \frac{\partial f_i^{local}}{\partial H_i'} \cdot \frac{\partial H_i'}{\partial \eta_j}, \\ \frac{\partial g_i^{local}}{\partial \xi_j} &= \frac{\partial g_i^{local}}{\partial H_i''} \cdot \frac{\partial H_i''}{\partial \xi_j} + \frac{\partial g_i^{local}}{\partial H_i'} \cdot \frac{\partial H_i'}{\partial \xi_j}, \\ \frac{\partial g_i^{local}}{\partial \eta_j} &= \frac{\partial g_i^{local}}{\partial H_i''} \cdot \frac{\partial H_i''}{\partial \eta_j} + \frac{\partial g_i^{local}}{\partial H_i'} \cdot \frac{\partial H_i'}{\partial \eta_j}. \end{aligned} \quad (18)$$

通过式(13)中对 $f_i^{local}, g_i^{local}$ 的定义可知

$$\begin{aligned} \frac{\partial f_i^{local}}{\partial H_i''} &= -\frac{H_i'}{(1 + (H_i')^2)^2} \\ \frac{\partial f_i^{local}}{\partial H_i'} &= \frac{3H_i''(H_i')^2 - H_i''}{(1 + (H_i')^2)^3} \\ \frac{\partial g_i^{local}}{\partial H_i''} &= \frac{1}{(1 + (H_i')^2)^2} \\ \frac{\partial g_i^{local}}{\partial H_i'} &= -\frac{4H_i''H_i'}{(1 + (H_i')^2)^3} \end{aligned} \quad (19)$$

对于链式求导的第二部分，参考第2.1节所述，局部坐标下的一、二阶导数是通过以所求点为中心展开的一系列控制体体积乘上对应系数之后加和得到的，也即为

$$H_i'' = \sum_l C_{2,l} \langle H \rangle_{i,l}, \quad H_i' = \sum_l C_{1,l} \langle H \rangle_{i,l}. \quad (20)$$

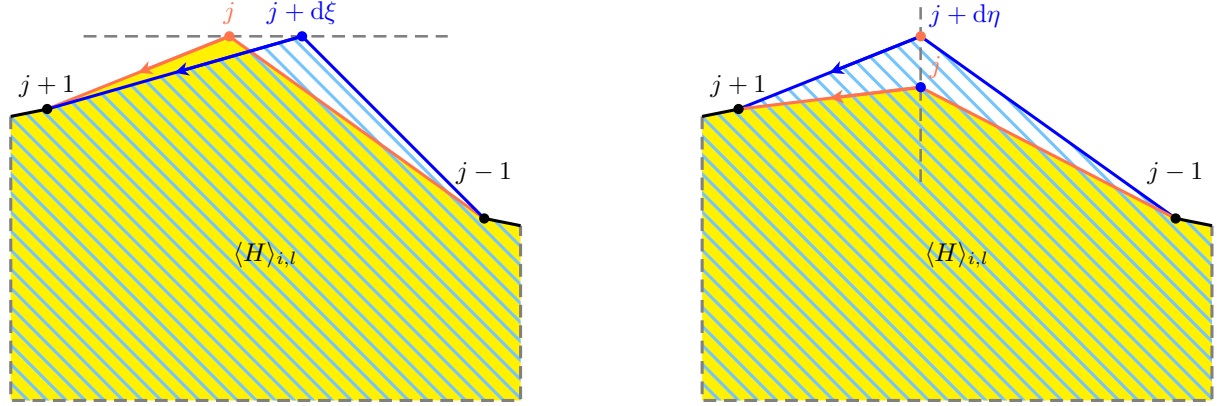
其中 $l$ 为展开的控制体的编号， $\langle H \rangle_{i,l}$ 为对应控制体的体积。为了叙述方便，不妨假设展开的一系列控制体可恰好覆盖当前曲线段。从而对于曲线段上的每个示踪点，其必然属于某一个控制体内部或为两个控制体的交界点（对于曲线段上计算点 $i$ 导数时没有用到的点，可认为其所属控制体对应的系数为0）。因此，若假设当前所求点 $j$ 属于控制体 $l_1$ 内部，对于 $l_2 \neq l_1$ ，则有

$$\frac{\partial \langle H \rangle_{i,l_2}}{\partial \xi_j} = 0, \quad \frac{\partial \langle H \rangle_{i,l_2}}{\partial \eta_j} = 0. \quad (21)$$

借由这一结论，可以得到当点 $j$ 为控制体 $l$ 内部点时，

$$\begin{aligned} \frac{\partial H_i''}{\partial \xi_j} &= C_{2,l} \frac{\partial \langle H \rangle_{i,l}}{\partial \xi_j}, \\ \frac{\partial H_i''}{\partial \eta_j} &= C_{2,l} \frac{\partial \langle H \rangle_{i,l}}{\partial \eta_j}, \\ \frac{\partial H_i'}{\partial \xi_j} &= C_{1,l} \frac{\partial \langle H \rangle_{i,l}}{\partial \xi_j}, \\ \frac{\partial H_i'}{\partial \eta_j} &= C_{1,l} \frac{\partial \langle H \rangle_{i,l}}{\partial \eta_j}. \end{aligned} \quad (22)$$

若曲线为利用示踪点线性插值得到的，则其计算如图7所示。最后将(22)、(19)与链式法则(18)相结合就可以得到



(a) 当点 $j$ 沿着 $\xi$ 方向正向移动时，增加的面积为 $\frac{d\xi(\eta_j - \eta_{j-1})}{2} + \frac{d\xi(\eta_{j+1} - \eta_j)}{2}$ ，因此变化的面积为 $\frac{d\xi(\eta_{j+1} - \eta_{j-1})}{2}$ ，即 $\frac{\partial \langle H \rangle_{i,l}}{\partial \xi_j} = \frac{\eta_{j+1} - \eta_{j-1}}{2}$

(b) 当点 $j$ 沿着 $\eta$ 方向正向移动时，增加的面积为 $\frac{d\eta(\xi_{j-1} - \xi_j)}{2} + \frac{d\eta(\xi_j - \xi_{j+1})}{2}$ ，因此变化的面积为 $\frac{d\eta(\xi_{j-1} - \xi_{j+1})}{2}$ ，即 $\frac{\partial \langle H \rangle_{i,l}}{\partial \xi_j} = \frac{\xi_{j-1} - \xi_{j+1}}{2}$

图 7: 当曲线为利用示踪点线性插值得到的时， $\langle H \rangle_{i,l}$ 的变化即为局部小三角形面积的增减，因为规定了曲线的左侧为内部，利用这一局部性质，即可以得到上面两个小图的结论。判断面积的增减即是判断变化得到的小三角形是否在曲线的右侧，以图7(a)中点 $j, j+1, j+d\xi$ 组成的三角形为例，若其在由 $j$ 指向 $j+1$ 的有向线段右侧，则 $\langle H \rangle_{i,l}$ 会增加这部分面积，若在其左侧，则会减去这部分面积。因为 $d\xi$ 方向已知，三角形与有向线段的关系可以通过 $\eta_{j+1} - \eta_j$ 的值的正负来判断。其余四个三角形同理，因此可以得到如上形式。

矩阵(17)中所有元素的值，通过式(16)，可以得到第 $i$ 点处曲率流的作用力关于点 $j$ 的Jacobian矩阵的四项，也即矩阵(11)。当 $i, j$ 遍历所有点时，就可得到如(6)所示的整体的Jacobian矩阵。

### 3 稳定性分析

本章我们将从最简单的例子来简单回顾一下稳定性分析的相关概念，并对基于第2章算法的平均曲率流问题稳定性进行分析。

#### 3.1 初值问题

定义初值问题:

**Definition 1.** Given  $T > 0$ ,  $\mathbf{f} : \mathbb{R}^N \times [0, T] \rightarrow \mathbb{R}^N$ , and  $\mathbf{u}_0 \in \mathbb{R}^N$ , the *initial value problem* (IVP) is to find  $\mathbf{u}(t) \in \mathcal{C}^1$  such that

$$\begin{cases} \mathbf{u}(0) = \mathbf{u}_0, \\ \mathbf{u}'(t) = \mathbf{f}(\mathbf{u}(t), t), \forall t \in [0, T]. \end{cases} \quad (23)$$

##### 3.1.1 零稳定性和绝对稳定性

对于稳定性分析，我们考虑标量初值问题:

$$u'(t) = \lambda u(t) \quad (24)$$

设时间步长为 $k$ ，时间 $t_n = nk$ ， $U^n$ 为 $t_n$ 时刻的估计： $U^n \approx u(t_n)$ ，零稳定性 (zero stability) 意味着

$$\text{固定 } T, \quad \lim_{k \rightarrow 0} \|U^{T/k}(T)\| < +\infty,$$



而绝对稳定性 (absolute stability) 则是

$$\text{固定 } k, \quad \lim_{T \rightarrow +\infty} \|U^{T/k}(T)\| < +\infty.$$

以Euler法为例,

$$U^{n+1} = (1 + k\lambda)U^n$$

则当

$$|1 + k\lambda| \leq 1. \quad (25)$$

时满足绝对稳定性, 即绝对稳定域为

$$\{z \in \mathbb{C} : |1 + z| \leq 1\}. \quad (26)$$

而对于  $\lambda > 0$  的情况, 显然无法满足绝对稳定性, 此时我们只能考虑零稳定性, 因为

$$\lim_{N \rightarrow +\infty} \left(1 + \frac{\lambda T}{N}\right)^N = e^{\lambda T},$$

当  $k \rightarrow 0$  时, 解是有界的。但是值得注意的是, 当  $\lambda$  很大的时候, 在实际计算时  $e^{\lambda T}$  仍会大得没有实际意义。其他数值方法 (LMM、RK) 的分析思想也与此相通, 此处不做赘述。

## 3.2 热方程

一维热方程定义如下:

**Definition 2.** The *one-dimensional heat equation* is a parabolic PDE of the form

$$u_t = \nu u_{xx} \text{ in } \Omega := (0, 1) \times (0, T), \quad (27)$$

where  $x \in (0, 1)$  is the spatial location,  $t \in (0, T)$  the time and  $\nu > 0$  the dynamic viscosity; the equation has to be supplemented with an *initial condition*

$$u(x, 0) = \eta(x), \text{ on } (0, 1) \times \{0\} \quad (28)$$

and appropriate boundary conditions at  $\{0, 1\} \times (0, T)$ .

### 3.2.1 MOL

首先对空间-时间定义域进行离散,  $h$  为空间步长,  $k$  为时间步长, 则

$$x_i = ih, \quad t_n = nk, \quad (29)$$

$U_i^n$  表示在  $(x_i, t_n)$  处的未知量。

为了更好的理解上一小节时间相关ODE的稳定性理论与本小节时间相关PDE的稳定性理论的关系, 我们考虑Method of Lines (MOL)。

**Definition 3.** *Method of Lines* (MOL) 通过以下步骤解决PDE问题:

- (a) 对空间导数进行离散, 得到时间连续的半离散ODE方程;
- (b) 用初值问题的数值方法求解上述ODE方程。

**Example 1.** 对热方程(27)空间采用中心差分可得到如下半离散系统:

$$\mathbf{U}'(t) = A\mathbf{U}(t) + g(t), \quad (30)$$

其中

$$A = \frac{\nu}{h^2} \begin{bmatrix} -2 & +1 & & & \\ +1 & -2 & +1 & & \\ & +1 & -2 & +1 & \\ & & \ddots & \ddots & \ddots \\ & & & +1 & -2 & +1 \\ & & & & +1 & -2 \end{bmatrix}, \quad (31)$$

$$\mathbf{U}(t) := \begin{bmatrix} U_1(t) \\ U_2(t) \\ U_3(t) \\ \vdots \\ U_{m-1}(t) \\ U_m(t) \end{bmatrix}, \quad g(t) = \frac{\nu}{h^2} \begin{bmatrix} g_0(t) \\ 0 \\ 0 \\ \vdots \\ 0 \\ g_1(t) \end{bmatrix}. \quad (32)$$

$g(t)$ 表示Dirichlet边界条件。

对时间采用向前Euler可以得到FTCS方法:

**Definition 4.** The *FTCS (forward in time, centered in space) method* solves the heat equation (27) by

$$\frac{U_i^{n+1} - U_i^n}{k} = \frac{\nu}{h^2} (U_{i-1}^n - 2U_i^n + U_{i+1}^n), \quad (33)$$

or, equivalently

$$U_i^{n+1} = U_i^n + 2r(U_{i-1}^n - 2U_i^n + U_{i+1}^n), \quad (34)$$

where  $r := \frac{k\nu}{2h^2}$ .

同样后续稳定性分析也基于FTCS来阐述基本的思想。

### 3.2.2 稳定性

因为MOL已经将原方程转化成半离散ODE方程, 因此稳定性的分析也与初值问题的一致。

**Lemma 5.** (31)中 $A$ 的特征值 $\lambda_p$ 和特征向量 $\mathbf{w}^p$ 为

$$\lambda_p = -\frac{4\nu}{h^2} \sin^2\left(\frac{p\pi h}{2}\right), \quad (35)$$

$$w_j^p = \sin(p\pi jh), \quad (36)$$

其中 $p, j = 1, 2, \dots, m$ ,  $h = \frac{1}{m+1}$ 。

为了使FTCS方法(33) 满足绝对稳定性, 对(35)中每个特征值都必须满足 $|1 + k\lambda| \leq 1$ , 即 $-2 \leq -4\nu k/h^2 \leq 0$ , 因此时间步长将受限于 $h^2$ 的大小:

$$k \leq \frac{h^2}{2\nu}. \quad (37)$$

然而上述分析的基础是绝对稳定性, 从Lemma 5也可以看到特征值实部是小于0的。而对特征值实部大于0的情况, 我们需要考虑的就是零稳定性, 由上一章结论可以看到, 当 $Re(\lambda) > 0$ 且非常大时, 零稳定性也失去了实际应用意义。而对于热方程, 特征值 $\lambda$ 往往是 $O(\frac{1}{h^2})$ 的, 一旦大于0, 将很难处理。

### 3.3 基于非定值HFES算法的平均曲率流问题稳定性

对于二维平均曲率流方程(2) 进行空间离散可以得到如下半离散形式

$$\frac{d\mathbf{X}}{dt} = \mathbf{f}(\mathbf{X}, t), \quad (38)$$

其中 $\mathbf{f}$ 为采用非定值HFES算法计算得到的右端项。为了进行稳定性分析，我们需要考察Jacobian矩阵(6) 的特征值分布。

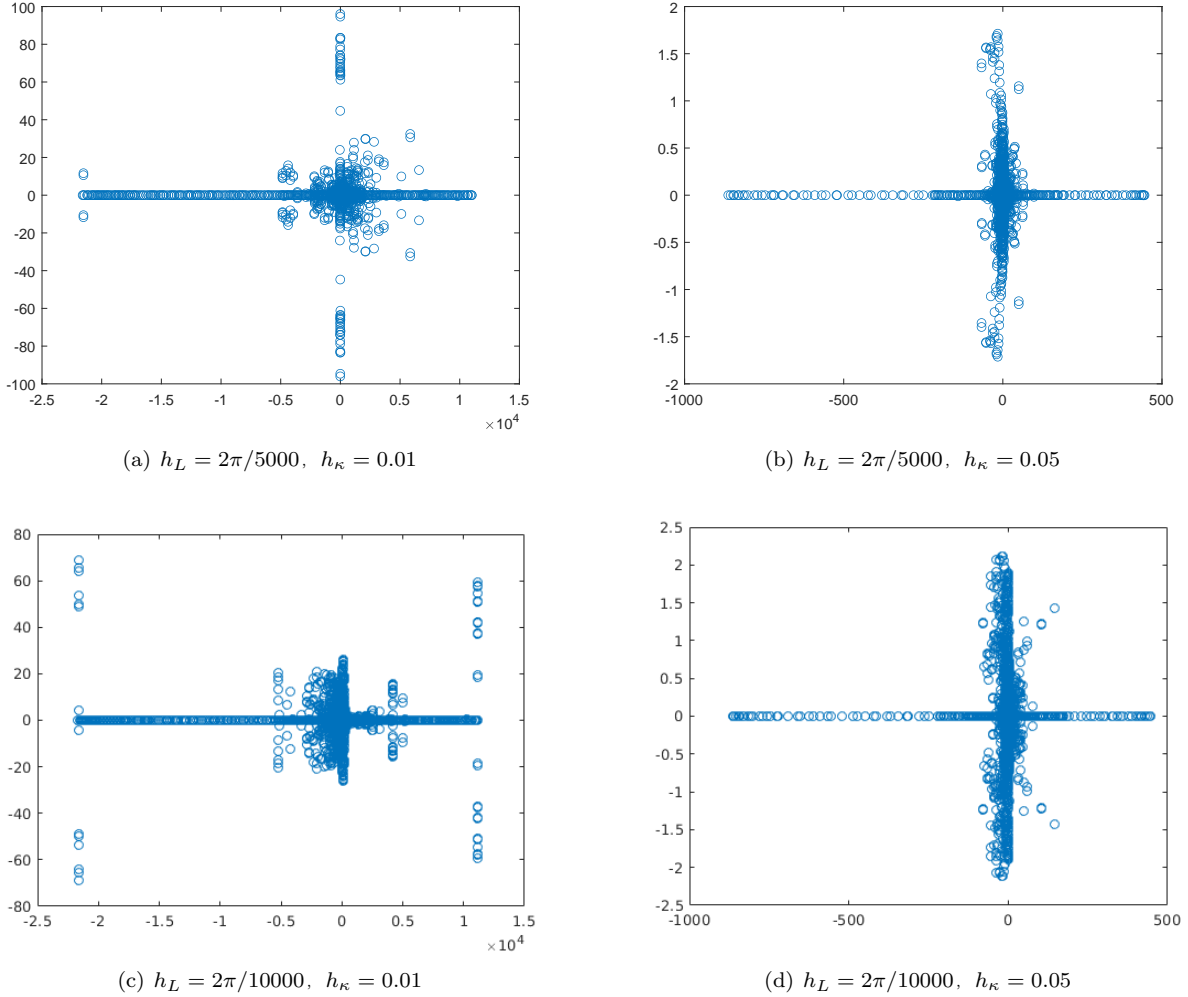


图 8: Jacobian矩阵 $J_{\mathbf{f}}$ 特征值分布, 选取的闭合曲线为单位圆。每行左右两图 $h_L$ 不变, 右图 $h_\kappa$ 为左图的五倍, 每列上下两图 $h_\kappa$ 不变, 上图 $h_L$ 为下图的两倍, 此时, (a)、(c)谱半径都约为 $2.16 \times 10^4$ , (b)、(d)谱半径都约为863,

从图8可以看出, Jacobian矩阵 $J_{\mathbf{f}}$ 的分布基本沿着实轴, 但是存在特征值 $Re(\lambda) > 0$ 的情况, 且特征值的大小满足 $O(h_\kappa^{-2})$ , 那么由上一章的分析, 无论是绝对稳定性还是零稳定性都很难满足。

## 4 基于中心差分法的空间离散

基于上一节不太美妙的稳定性结果, 我们现考虑采用中心差分来处理空间离散。

对于二维平均曲率流方程(2) 在空间上对其做二阶中心差分获得曲线上一点 $\mathbf{X}_i$ 关于该处弧长 $s_i$ 的二阶导数, 即

$$\frac{\partial^2 \mathbf{X}_i}{\partial s_i^2} = \frac{(\mathbf{X}_{i+1} - \mathbf{X}_i)/h_i - (\mathbf{X}_i - \mathbf{X}_{i-1})/h_{i-1}}{(h_i + h_{i-1})/2}, \quad (39)$$

其中 $h_i$ 为 $\mathbf{X}_i$ 至 $\mathbf{X}_{i+1}$ 间的弧长。为了使用三次样条曲线的结果计算弧长, 对于以累积弦长 $s$ 为参数得到的 $x(l), y(l)$ , 可得方程(40),

$$\nabla s_i = \int_{l_i}^{l_{i+1}} \sqrt{x'(l)^2 + y'(l)^2} dl. \quad (40)$$

对于该方程, 可以用Gauss-Legendre积分进行求解。

设对 $\partial \mathbf{X}^2 / \partial^2 s$ 的二阶中心差分为 $\mathbf{F}(\mathbf{X})$ , 那么接下来我们需要对半离散格式

$$\frac{d\mathbf{X}}{dt} = \mathbf{F}(\mathbf{X}) = A\mathbf{X} = \begin{bmatrix} A_1 & \\ & A_1 \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix} \quad (41)$$

在时间步上进一步离散, 其中

$$A_1 = \begin{bmatrix} -\frac{2}{h_N+h_1}(\frac{1}{h_N} + \frac{1}{h_1}) & \frac{2}{h_N+h_1} \frac{1}{h_1} & & & \frac{2}{h_N+h_1} \frac{1}{h_N} \\ \frac{2}{h_1+h_2} \frac{1}{h_1} & -\frac{2}{h_1+h_2}(\frac{1}{h_1} + \frac{1}{h_2}) & \frac{2}{h_1+h_2} \frac{1}{h_2} & & \\ & \ddots & \ddots & \ddots & \\ & & \frac{2}{h_{N-2}+h_{N-1}} \frac{1}{h_{N-2}} & -\frac{2}{h_{N-2}+h_{N-1}}(\frac{1}{h_{N-2}} + \frac{1}{h_{N-1}}) & \frac{2}{h_{N-2}+h_{N-1}} \frac{1}{h_{N-1}} \\ \frac{2}{h_{N-1}+h_N} \frac{1}{h_N} & & \frac{2}{h_{N-1}+h_N} \frac{1}{h_{N-1}} & -\frac{2}{h_{N-1}+h_N}(\frac{1}{h_{N-1}} + \frac{1}{h_N}) & -\frac{2}{h_{N-1}+h_N} \frac{1}{h_N} \end{bmatrix}. \quad (42)$$

下面我们将介绍显式RKC方法, 并分析如何将其应用于平均曲率流方程。

## 5 显式RKC方法描述

### 5.1 方法概述

对于一般的抛物方程, 通过空间离散我们可以得到如下常微分方程组:

$$\frac{d\mathbf{U}}{dt} = \mathbf{F}(\mathbf{U}, t), \quad 0 < t \leq T, \quad \mathbf{U}(0) = \mathbf{U}^0, \quad (43)$$

其中 $\mathbf{U}^0$ 是给定初值向量, 且(43)已包含边界条件。

因为显式Runge-Kutta-Chebyshev方法的稳定区域是沿着负实轴分布的狭长区域, 因此需要满足以下条件:

- (i) Jacobian矩阵 $J_{\mathbf{F}} = \partial \mathbf{F} / \partial \mathbf{U}$  的特征值需要分布在沿着负实轴的狭长区域内, 最好的当然是正好分布在负实轴上。
- (ii)  $J_{\mathbf{F}}$ 是接近正规 (normal) 的。

特别的, 当 $J_{\mathbf{F}}$ 对称且负定时, 显然满足这两个条件。

相比隐式方法而言, 显式RKC方法的优势在于不要求解大型复杂的非线性方程组。对于刚性问题, 会需要很小的时间步长, 而在相同条件下, 在后文我们可以看到, RKC方法的级数 $s$ 的平方与时间步长 $k$ 成正比, 即随着 $k$ 的减小,  $s$ 也会相应减小。这也说明了该方法的有效性, 具体描述见5.4节。

### 5.2 稳定多项式

对于 $s$ 级显式Runge-Kutta方法

$$\begin{aligned} \mathbf{Y}_0 &= \mathbf{U}^n, \\ \mathbf{Y}_j &= \mathbf{U}^n + k \sum_{l=0}^{j-1} a_{jl} \mathbf{F}(\mathbf{Y}_l, t_n + c_l k), \quad 1 \leq j \leq s, \\ \mathbf{U}^{n+1} &= \mathbf{Y}_s, \end{aligned} \quad (44)$$

其中  $c_j = a_{j0} + \dots + a_{jj-1}$ ,  $k = t_{n+1} - t_n$  是时间步长, 绝对稳定性 (absolute stability) 考察的是由初值  $\mathbf{U}^0$  的扰动带来的误差传播, 而对于级数  $s$  很大的 RK 方法, 还需要考虑每个积分步内产生的误差传播, 也就是所谓的内部稳定性 (internal stability) 分析。

我们通过引入扰动来考察稳定性:

$$\begin{aligned}\tilde{\mathbf{Y}}_0 &= \tilde{\mathbf{U}}^n, \\ \tilde{\mathbf{Y}}_j &= \tilde{\mathbf{U}}^n + k \sum_{l=0}^{j-1} a_{jl} \mathbf{F}(\tilde{\mathbf{Y}}_l, t_n + c_l k) + \tilde{\mathbf{r}}_j, \quad 1 \leq j \leq s, \\ \tilde{\mathbf{U}}^{n+1} &= \tilde{\mathbf{Y}}_s,\end{aligned}\tag{45}$$

其中  $\tilde{\mathbf{U}}^n$  表示  $\mathbf{U}^n$  扰动后的值,  $\tilde{\mathbf{r}}_j$  表示在第  $j$  步引入的扰动 (例如舍入误差), 设  $\mathbf{e}^n = \tilde{\mathbf{U}}^n - \mathbf{U}^n$ ,  $\varepsilon_j = \tilde{\mathbf{Y}}_j - \mathbf{Y}_j$ 。由此可得到如下误差形式:

$$\varepsilon_j = R_j(kJ_{\mathbf{F}})\mathbf{e}^n + \sum_{l=1}^j Q_{jl}(kJ_{\mathbf{F}})\tilde{\mathbf{r}}_l, \quad 1 \leq j \leq s,\tag{46}$$

其中  $R_j(kJ_{\mathbf{F}})$ 、 $Q_{jl}(kJ_{\mathbf{F}})$  分别为  $j$  次和  $j-l$  多项式, 特别的,

$$\mathbf{e}^{n+1} = R_s(kJ_{\mathbf{F}})\mathbf{e}^n + \sum_{j=1}^s Q_{sj}(kJ_{\mathbf{F}})\tilde{\mathbf{r}}_j.\tag{47}$$

$R = R_s$  就是绝对稳定多项式, 而  $Q_{sj}$  ( $1 \leq j \leq s$ ) 则是内部稳定多项式。

由  $J_{\mathbf{F}}$  的正规性, 可以得到误差的范数关系如下:

$$\begin{aligned}\|\mathbf{e}^{n+1}\| &\leq \|R(kJ_{\mathbf{F}})\| \|\mathbf{e}^n\| + \sum_{j=1}^s \|Q_{sj}(J_{\mathbf{F}})\| \|\tilde{\mathbf{r}}_j\| \\ &= \rho(R(kJ_{\mathbf{F}})) \|\mathbf{e}^n\| + \sum_{j=1}^s \rho(Q_{sj}(J_{\mathbf{F}})) \|\tilde{\mathbf{r}}_j\| \\ &= \max_{z=k\lambda} |R(z)| \|\mathbf{e}^n\| + \sum_{j=1}^s \max_{z=k\lambda} |Q_{sj}(z)| \|\tilde{\mathbf{r}}_j\|\end{aligned}\tag{48}$$

其中  $\lambda$  为矩阵  $J_{\mathbf{F}}$  的特征值,  $R(z)$  为  $s$  次稳定多项式:

$$R(z) = d_0 + d_1 z + d_2 z^2 + \dots + d_s z^s.\tag{49}$$

则绝对稳定区域为:

$$\mathcal{S} := \{z \in \mathbb{C} : |R(z)| \leq 1\}.\tag{50}$$

对于给定的  $s$ , 在满足  $p$  阶精度的条件下, 我们希望绝对稳定区域  $\mathcal{S}$  能够包含负实轴区间  $[-\beta(s), 0]$  且稳定边界  $\beta(s)$  尽可能大。

基于上述目的得到的最优或近最优稳定多项式能够保证  $\beta(s)$  有二次形式

$$\beta(s) = c(s)s^2,\tag{51}$$

其中  $c(s)$  接近常数函数, 且随着阶数  $p$  的增大而减小。接下来我们将对一、二阶条件下的最优或近最优  $R(z)$  进行简单的推导和总结。

### 5.2.1 一阶多项式

为了达到一阶精度, 多项式  $R(z)$  需要满足条件

$$d_0 = d_1 = 1.\tag{52}$$

此时最优多项式可由第一类移位Chebyshev多项式 (shifted Chebyshev polynomial) 给出:

$$R(z) = T_s \left( 1 + \frac{z}{s^2} \right), \quad (53)$$

其中 $T_s(\cdot)$ 为 $s$ 次第一类Chebyshev多项式:

$$T_s(x) = \cos(s \arccos(x)), \quad -1 \leq x \leq 1, \quad (54)$$

或表示成递推形式:

$$T_0(x) = 1, \quad T_1(x) = x, \quad T_j(x) = 2xT_{j-1}(x) - T_{j-2}(x), \quad 2 \leq j \leq s. \quad (55)$$

由此可得稳定边界为

$$\beta(s) = 2s^2. \quad (56)$$

### 5.2.2 二阶多项式

为了达到二阶精度, 多项式 $R(z)$ 除了(52)之外还需要满足条件

$$d_2 = \frac{1}{2}. \quad (57)$$

下面将给出两个近似最优多项式。

第一种由Bakker<sup>[1]</sup>提出, 也就是Bakker-Chebyshev多项式:

$$R(z) = \frac{2}{3} + \frac{1}{3s^2} + \left( \frac{1}{3} - \frac{1}{3s^2} \right) T_s \left( 1 + \frac{3z}{s^2 - 1} \right), \quad s \geq 2. \quad (58)$$

其稳定边界

$$\beta(s) \approx \frac{2}{3}(s^2 - 1) \quad (59)$$

大约能达到80%的最优区间。

第二种由Van der Houwen和Sommeijer<sup>[2]</sup>提出

$$R(z) = \frac{2}{2-z} - \frac{z}{2-z} T_s \left( \cos \frac{\pi}{s} + \frac{z}{2} \left( 1 - \cos \frac{\pi}{s} \right) \right), \quad s \geq 2, \quad (60)$$

其结果非常接近最优多项式, 即有

$$\beta(s) = \frac{2}{(\tan(\pi/2))^2} \approx 8 \frac{s^2}{\pi^2} \approx 0.81s^2. \quad (61)$$

该多项式相比第一种有更好的结果, 但相应的形式也比前者更加复杂, 也因为第一种多项式提出的更早, 就目前查阅的文献资料来看前者的运用更为广泛, 在后续行文中我们也主要以Bakker-Chebyshev多项式为基础进行阻尼引入和二阶方法的推导。

### 5.2.3 阻尼

以一阶移位Chebyshev多项式(53)为例, 我们考察其相应的稳定区域 $\mathcal{S}$ 形状, 如图9所示, 可以看到虽然稳定区域包含区间 $[-\beta, 0]$ , 但每当实轴上 $|R(z)| = 1$ 的时候,  $\mathcal{S}$ 都会收缩成轴上的一个点, 因此我们希望能够适当地修改前面得到的 $R(z)$  从而得到更具实用价值的绝对稳定区域, 即我们要让 $|R(z)|$ 在实轴 $[-\beta, 0]$ 上的上界略小于1, 而其代价是需要略微地缩小 $\beta$ 的值。

对于一阶方法, 我们用以下多项式来替代(53):

$$R(z) = \frac{T_s(\omega_0 + \omega_1 z)}{T_s(\omega_0)}, \quad \omega_1 = \frac{T_s(\omega_0)}{T'_s(\omega_0)}. \quad (62)$$

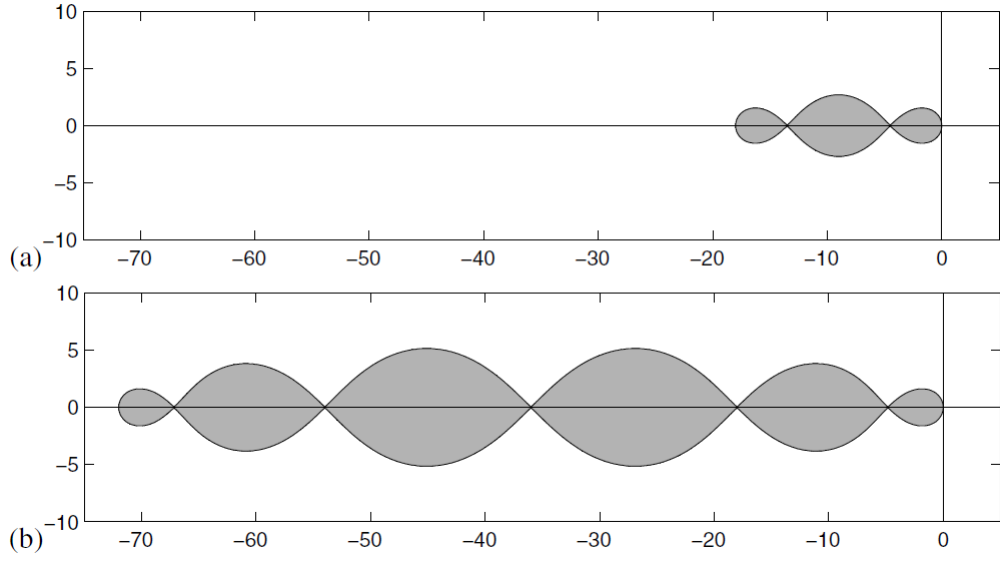


图 9: 以(53)为稳定多项式的一阶Runge-Kutta-Chebyshev方法绝对稳定区域。(a)  $s = 3$ , (b)  $s = 6$ 。

$\omega_0 > 1$ 为阻尼参数,  $\omega_1$ 的选择用以保证一阶精度(52)。此时 $R(z)$ 在 $[-\beta, 0]$ 上的取值位于 $-T_s^{-1}(\omega_0)$ 和 $T_s^{-1}(\omega_0) < 1$ 之间,  $\beta(s)$ 的取值由方程 $\omega_0 - \omega_1\beta = -1$ 给出, 取阻尼参数 $\omega_0 = 1 + \epsilon/s^2$ , 其中 $\epsilon$ 是一个小正数, 则通过在 $z = 1$ 处Taylor展开我们可以得到:

$$\beta(s) = \frac{(\omega_0 + 1)T'_s(\omega_0)}{T_s(\omega_0)} \approx 2s^2 \left[ 1 - \frac{4s^2 - 1}{6}(\omega_0 - 1) \right] \approx (2 - \frac{4}{3}\epsilon)s^2. \quad (63)$$

当取 $\epsilon = 0.05$ 时, 上界 $T_s^{-1}(\omega_0) \approx 1 - \epsilon$ 意味着5%的阻尼效果, 而 $\beta \approx 1.93s^2$ 并不会减少很多。图10显示了此时对应的稳定区域形状。

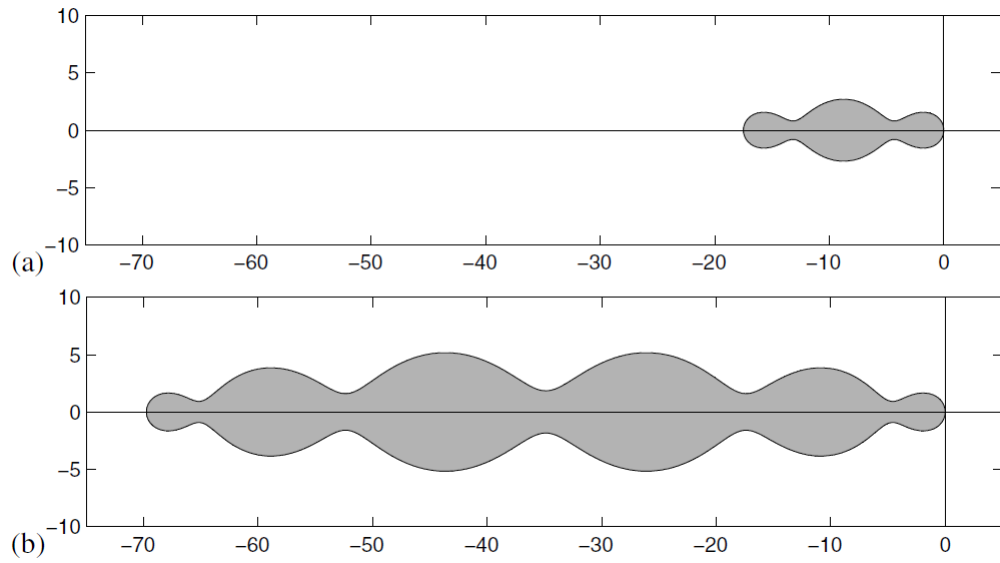


图 10: 以(62)为稳定多项式的一阶阻尼Runge-Kutta-Chebyshev方法绝对稳定区域。(a)  $s = 3$ , (b)  $s = 6$ 。

同理，我们可以得到二阶阻尼Bakker-Chebyshev多项式：

$$\begin{aligned} R(z) &= a_s + b_s T_s(\omega_0 + \omega_1 z), \\ \omega_1 &= \frac{T'_s(\omega_0)}{T''_s(\omega_0)}, \quad a_s = 1 - b_s T_s(\omega_0), \quad b_s = \frac{T''_s(\omega_0)}{(T'_s(\omega_0))^2}. \end{aligned} \quad (64)$$

此时 $\beta(s)$ 为

$$\beta(s) = \frac{(\omega_0 + 1)T''_s(\omega_0)}{T'_s(\omega_0)} \approx \frac{2}{3}(s^2 - 1)(1 - \frac{2}{15}\epsilon). \quad (65)$$

当取 $\epsilon = 2/13$ 时，可以达到5%的阻尼效果（ $a_s + b_s \approx 1 - \epsilon/3$ ），与此同时 $\beta$ 大约会减少2%。

## 5.3 RKC方法

### 5.3.1 推导

因为 $R(z)$ 并不唯一确定最终的Runge-Kutta算法，所以在满足绝对稳定性的前提下，我们需要选择Runge-Kutta算法的构造方式来提高内部稳定性。

Van der Houwen和Sommeije<sup>[3]</sup>提出了一类基于三项Chebyshev递推式(55)构造的RK方法，即Runge-Kutta-Chebyshev (RKC) 方法。其构造的主要思想是运用三项Chebyshev递推式定义(46)中RK方法每一中间步的多项式 $R_j$ ，从而可以得到最后第 $s$ 步的稳定多项式 $R = R_s$ 。假设所有 $R_j(0 \leq j \leq s)$ 有如下形式：

$$R_j(z) = a_j + b_j T_j(\omega_0 + \omega_1 z), \quad a_j = 1 - b_j T_j(\omega_0), \quad (66)$$

其中 $R = R_s$ 就是之前所定义的某一稳定多项式。换言之， $\omega_0$ 、 $\omega_1$ 和 $b_s$ 都是已知的，而 $b_j(0 \leq j < s)$ 是未知变量。

运用三项递推式(55)和性质 $R_j(0) = 1$ ，可以得到 $R_j$ 满足如下形式：

$$\begin{aligned} R_0(z) &= 1, \\ R_1(z) &= 1 + \tilde{\mu}_1 z, \\ R_j(z) &= (1 - \mu_j - \nu_j) + \mu_j R_{j-1}(z) + \nu_j R_{j-2}(z) + \tilde{\mu}_j R_{j-1}(z)z + \tilde{\gamma}_j z, \quad j = 2, \dots, s, \end{aligned} \quad (67)$$

其中

$$\begin{aligned} \tilde{\mu}_1 &= b_1 \omega_1, \\ \mu_j &= \frac{2b_j \omega_0}{b_{j-1}}, \quad \nu_j = \frac{-b_j}{b_{j-2}}, \quad \tilde{\mu}_j = \frac{2b_j \omega_1}{b_{j-1}}, \quad \tilde{\gamma}_j = -a_{j-1} \tilde{\mu}_j, \quad j = 2, \dots, s. \end{aligned} \quad (68)$$

则由多项式和算法的关系可以得到RKC方法：

$$\begin{aligned} \mathbf{Y}_0 &= \mathbf{U}^n, \\ \mathbf{Y}_1 &= \mathbf{Y}_0 + \tilde{\mu}_1 k \mathbf{F}(\mathbf{Y}_0, t_n), \\ \mathbf{Y}_j &= (1 - \mu_j - \nu_j) \mathbf{Y}_0 + \mu_j \mathbf{Y}_{j-1} \\ &\quad + \nu_j \mathbf{Y}_{j-2} + \tilde{\mu}_j k \mathbf{F}(\mathbf{Y}_{j-1}, t_n + c_{j-1}k) + \tilde{\gamma}_j k \mathbf{F}(\mathbf{Y}_0, t_n), \quad j = 2, \dots, s, \\ \mathbf{U}^{n+1} &= \mathbf{Y}_s. \end{aligned} \quad (69)$$

对于一阶方法，可以取参数 $b_j$ 为：

$$b_j = T_j^{-1}(\omega_0), \quad j = 0, \dots, s. \quad (70)$$

对于二阶方法，可取

$$b_0 = b_1 = b_2, \quad b_j = \frac{T''_j(\omega_0)}{(T'_j(\omega_0))^2}, \quad j = 2, \dots, s, \quad (71)$$

如此所有 $\mathbf{Y}_j(2 \leq j \leq s)$ 都可以达到二阶精度。



### 5.3.2 内部稳定性

上述推导都是基于满足绝对稳定性的条件，而当我们考虑内部稳定性，则对于上述一阶二阶方法，无论是否进行阻尼操作，都可以得到如下误差范围：

$$\|\mathbf{e}^{n+1}\| \leq \|\mathbf{e}^n\| + \frac{1}{2}s(s+1)C \max_j \|\tilde{\mathbf{r}}_j\|, \quad (72)$$

其中 $C$ 是与 $\mathbf{J}_{\mathbf{F}}$ 、 $k$ 、 $s$ 无关的常数，具体推导见Verwer<sup>[4]</sup>的文章。

由此可得到相应的内部稳定性结论，在每一时间步中的例如舍入误差的内部扰动的积累与 $\mathbf{J}_{\mathbf{F}}$ 的特征值无关。由(72)可以看出，内部扰动至多关于 $s$ 呈二次增长。以舍入误差为例，假设 $s = 1000$ ，那么局部扰动最多 $\approx 10^6 \|\tilde{\mathbf{r}}_j\|$ ，若计算机的机器精度约为14位，则该局部扰动仍然能留下8位的精度。

### 5.4 RKC方法效率的数值结果

以一维热方程为例，若空间采用二阶中心差分，空间步长为 $h$ ，那么右端矩阵谱半径为 $\rho = O(\frac{1}{h^2})$ ，若时间步长 $k = O(h^2)$ ，则级数 $s = O(\sqrt{k\rho}) = O(1)$ ，这种情况下 $s$ 并不会很大。若时间步长 $k = O(h)$ ，则级数 $s = O(\sqrt{k\rho}) = O(\frac{1}{\sqrt{h}})$ ，这种情况下， $h = O(10^{-4})$ ， $s$ 大概会上百， $h = O(10^{-6})$ ， $s$ 大概会上千。

### 5.5 将显式RKC方法运用于曲率流

由(41)，若将RKC方法运用于曲率流问题，则对每一时间步我们可以简单总结出以下实行步骤：

- (i) 计算 $A$ 谱半径 $\rho(A)$ ；
- (ii) 由 $k\rho(A) \leq \beta$  得到稳定边界 $\beta$ ；
- (iii) 根据 $\beta$ 与级数 $s$ 关系（由所使用的RKC方法对应的稳定多项式确定），结合设定的阻尼参数，得到需要实行的RKC级数 $s$ ；
- (iv) 将最终确定好的 $s$ 级RKC方法运用于(41)在该时间步的计算。

## 参考文献

- [1] M. Bakker, Analytical aspects of a minimax problem, Technical Note TN 62, Mathematical Centre, Amsterdam (1971) (in Dutch).
- [2] P. J. van der Houwen, The development of Runge-Kutta methods for partial differential equations, CWI Report NM-R9420, CWI, Amsterdam (1994); also: *Proceedings of the 1994 IMACS World Congress* (Atlanta, 1994), to appear.
- [3] P. J. van der Houwen and B. P. Sommeijer. On the internal stability of explicit  $m$ - stage Runge–Kutta methods for large values of  $m$ . *Z. Angew. Math. Mech.*, 60:479–485, 1980 (in German).
- [4] J. G. Verwer. Explicit Runge-Kutta methods for parabolic differential equations. *Appl. Numer. Math.*, 22:359, 1996.
- [5] Qinghai Zhang. Fourth-and higher-order interface tracking via mapping and adjusting regular semianalytic sets represented by cubic splines. *SIAM Journal on Scientific Computing*, 40(6):A3755-A3788, 2018.
- [6] Qinghai Zhang and Aaron Fogelson. MARS: an analytic framework of interface tracking via mapping and adjusting regular semialgebraic sets. *SIAM Journal on Numerical Analysis*, 54(2):530-560, 2016.