

Numerical Analysis

Zhiping Li

LMAM and School of Mathematical Sciences
Peking University



一阶常微分方程组的初值问题的数值方法

考虑一阶常微分方程组的初值问题:

$$\begin{cases} \frac{d\mathbf{y}}{dx} = \mathbf{f}(x, \mathbf{y}), & a \leq x < b, \\ \mathbf{y}(a) = \boldsymbol{\eta}, \end{cases}$$

其中

$$\mathbf{y}(x) = \begin{pmatrix} y_1(x) \\ y_2(x) \\ \vdots \\ y_m(x) \end{pmatrix}, \quad \boldsymbol{\eta} = \begin{pmatrix} \eta_1(x) \\ \eta_2(x) \\ \vdots \\ \eta_m(x) \end{pmatrix}, \quad \mathbf{f}(x, \mathbf{y}) = \begin{pmatrix} f_1(x, \mathbf{y}) \\ f_2(x, \mathbf{y}) \\ \vdots \\ f_m(x, \mathbf{y}) \end{pmatrix}.$$

我们在前面针对单个方程导出的所有格式, 形式上都可以直接应用于方程组. 理论分析及其证明也都类似.



一阶常微分方程组的初值问题的数值方法的例

方程组的四级四阶古典 Runge-Kutta 公式:

$$\begin{cases} \mathbf{K}_1 = \mathbf{f}(x_n, \mathbf{y}_n), \\ \mathbf{K}_2 = \mathbf{f}(x_n + \frac{h}{2}, \mathbf{y}_n + \frac{h}{2}\mathbf{K}_1), \\ \mathbf{K}_3 = \mathbf{f}(x_n + \frac{h}{2}, \mathbf{y}_n + \frac{h}{2}\mathbf{K}_2), \\ \mathbf{K}_4 = \mathbf{f}(x_n + h, \mathbf{y}_n + h\mathbf{K}_3), \\ \mathbf{y}_{n+1} = \mathbf{y}_n + \frac{h}{6}(\mathbf{K}_1 + 2\mathbf{K}_2 + 2\mathbf{K}_3 + \mathbf{K}_4). \end{cases}$$

方程组的预估-校正改进的 Euler 公式:

$$\begin{cases} \mathbf{y}_{n+1}^* = \mathbf{y}_n + h\mathbf{f}(x_n, \mathbf{y}_n), \\ \mathbf{y}_{n+1} = \mathbf{y}_n + \frac{h}{2}[\mathbf{f}(x_n, \mathbf{y}_n) + \mathbf{f}(x_{n+1}, \mathbf{y}_{n+1}^*)]. \end{cases}$$



高阶常微分方程初值问题数值方法

考虑 m 阶常微分方程初值问题

$$\begin{cases} \frac{d^m y(x)}{dx} = f(x, y, y', \dots, y^{(m-1)}), & x \in [a, b] \\ y(a) = \eta^{(0)}, \\ y'(a) = \eta^{(1)}, \\ \dots\dots\dots \\ y^{(m-1)}(a) = \eta^{(m-1)}. \end{cases}$$

数值求解时, 可将其先化为以下等价的一阶方程组初值问题

$$\begin{cases} \frac{dy_i(x)}{dx} = y_{i+1}(x), & i = 1, \dots, m, & x \in [a, b] \\ \frac{dy_m(x)}{dx} = f(x, y_1, y_2, \dots, y_m), & x \in [a, b] \\ y_i(a) = \eta^{(i-1)}, & i = 1, \dots, m. \end{cases}$$



刚性常微分方程组

与单个方程的数值求解相比, 方程组的特殊困难之一是所谓的刚性问题. 我们通过以下例子来认识一下刚性及其可能带来的困难.

考虑时间发展方程组

$$\begin{pmatrix} y' \\ z' \end{pmatrix} = \begin{pmatrix} -1000 & 999 \\ 0 & -1 \end{pmatrix} \begin{pmatrix} y \\ z \end{pmatrix} + \mathbf{f}(t).$$

易见系数矩阵的特征值为 $\lambda_1 = -1$ 和 $\lambda_2 = -1000$, 解可写为

$$y(t) = C_1 e^{-t} + C_2 e^{-1000t} + \phi(t),$$

其中 $\phi(t)$ 是解相应于非齐次项 $\mathbf{f}(t)$ 的部分. 当 $t \gg 1$ 时, $y(t) \approx \phi(t)$. 我们常常希望用数值方法得到渐近解 $\phi(t)$.



应用显式 Euler 法数值求解时遇到的困境

如果应用显式 Euler 法数值求解以上初值问题, 由显式 Euler 格式的绝对稳定区域 $R = \{\mu \in \mathbb{C} : |1 + \mu| < 1\}$, 为保证 $\lambda_1 h \in R$, 即 $|1 - h| < 1$, 只需 $h \in (0, 2)$; 但为保证 $\lambda_2 h \in R$, 即 $|1 - 1000h| < 1$, 则需 $h \in (0, 0.002)$. 因此, 为了保证格式绝对稳定, 时间步长 h 应满足要求较高的条件, 即必须取 $h \in (0, 0.002)$.

另一方面, e^{-t} 收敛于零的速度远比 e^{-1000t} 慢, 要算到 (弛豫时间) T 使得 e^{-T} 小于允许误差后, 方可认为数值解有条件有效逼近渐近解了. 然而, 由于时间步长的限制, 这往往需要花费非常多的时间步, 有可能导致舍入误差的积累大幅增加, 从而影响数值结果的实际逼近精度.



刚性方程组的定义

定义： 对于线性常微分方程组 $\mathbf{y}' = \mathbf{A}\mathbf{y} + \phi(t)$, 如果 $\mathbf{A}$ 的特征值 λ_i , 满足 $\text{Re}(\lambda_i) < 0, i = 1, \dots, m$, 且 $\max_i |\text{Re}(\lambda_i)| \gg \min_i |\text{Re}(\lambda_i)|$, 则称该方程组为刚性方程组, 称

$$S = \frac{\max_i |\text{Re}(\lambda_i)|}{\min_i |\text{Re}(\lambda_i)|}$$

为刚性比. 对于非线性常微分方程组 $\mathbf{y}'(t) = \mathbf{f}(t, \mathbf{y}(t))$, 设 $\bar{\mathbf{y}}$ 为其精确解, 令 $\mathbf{J}(t) = \frac{\partial \mathbf{f}}{\partial \mathbf{y}}(t, \bar{\mathbf{y}})$, 若线性方程组 $\mathbf{z}' = \mathbf{J}(t)\mathbf{z}$ 是刚性的, 则称该非线性方程组为刚性的.

困境： 刚性系统中内含差别很大的时间尺度, 稳定性要求的时间步长一般由最小的时间尺度（系统变化最快的子过程）决定, 而允许误差要求的弛豫时间则由最大的时间尺度（系统变化最慢的子过程）决定.



A 稳定的数值方法

系统的弛豫时间一般由系统所模拟的实际应用过程所决定, 而时间步长则除了依赖于实际应用过程之外还与算法的绝对稳定区域有关. 因此, 为了有效求解刚性方程组, 一个自然的想法就是尽可能扩大算法的绝对稳定区域.

定义: 如果一个数值格式的绝对稳定区域包含了整个左半复平面 $\operatorname{Re}(\lambda h) < 0$, 则称该格式是 A -稳定的.

隐式 Euler 格式是 A -稳定的. 显然, 只要数值格式是 A -稳定的, 则不论方程组的刚性有多大, 格式的绝对稳定性都不会限制步长 h 的选取, 因此, 步长 h 的选取完全由精度要求决定.



关于 A -稳定格式的一些理论结果

定理：关于 A -稳定格式有以下结果：

- ① 任何显式线性多步法，以及任何显式 *Runge-Kutta* 方法，都不是 A -稳定的。
- ② 隐式线性多步法是 A -稳定的 $\Leftrightarrow$ (i) $\sigma(\xi)$ 的根都在单位圆内；
(ii) 在单位圆上 $\operatorname{Re}(\rho(\xi)/\sigma(\xi)) \geq 0$ 。
- ③ A -稳定的隐式线性多步法的精度不超过二阶。
- ④ 具有最小误差常数的二阶 A -稳定隐式线性多步法是梯形法。
- ⑤ *Gauss* 型的 m 级 $2m$ 阶 (及某些 m 级 $2m-1$ 和 $2m-2$ 阶) 的隐式 *Runge-Kutta* 方法是 A -稳定的。
- ⑥ 可构造出 A -稳定的 m 级 q 阶半隐式的 *Runge-Kutta* 方法。

为了构造精度高计算量更小的求解刚性方程组的数值方法，有必要适当放松 A -稳定的限制条件。这就引出了 $A(\alpha)$ -稳定性和刚性稳定性的概念。

$A(\alpha)$ -稳定性和刚性稳定性

定义： 如果存在常数 $\alpha > 0$, 使得一个数值格式的绝对稳定区域 R 满足

$$R \supset \left\{ \mu \in \mathbb{C} : \operatorname{Re}(\mu) < 0, \arctan \frac{|\operatorname{Im}(\mu)|}{|\operatorname{Re}(\mu)|} \leq \alpha \right\},$$

则称该格式是 $A(\alpha)$ -稳定的.

定义： 如果存在常数 $a > 0, c > 0$, 使得一个数值格式的绝对稳定区域 R 满足

$$R \supset \left\{ \mu \in \mathbb{C} : \operatorname{Re}(\mu) \leq -a, \text{ 或 } \operatorname{Re}(\mu) < 0, |\operatorname{Im}(\mu)| \leq c \right\},$$

则称该格式是刚性稳定的.

注： 若一个数值格式是刚性稳定的, 令 $\alpha = \arctan \frac{c}{a}$, 则该格式是 $A(\alpha)$ -稳定的.



$A(\alpha)$ -稳定性和刚性稳定性数值方法的例 — Gear 方法

Gear 方法是一种 k 步 k 阶隐式线性多步法. 其形式为:

$$\mathbf{y}_{n+k} + \sum_{j=1}^k \alpha_{k,j} \mathbf{y}_{n+k-j} = h\beta_k \mathbf{f}(x_{n+k}, \mathbf{y}_{n+k}).$$

k	$\alpha_{k,1}$	$\alpha_{k,2}$	$\alpha_{k,3}$	$\alpha_{k,4}$	$\alpha_{k,5}$	$\alpha_{k,6}$	β_k
1	-1						1
2	$-\frac{4}{3}$	$\frac{1}{3}$					$\frac{2}{3}$
3	$-\frac{18}{11}$	$\frac{9}{11}$	$-\frac{2}{11}$				$\frac{6}{11}$
4	$-\frac{48}{25}$	$\frac{36}{25}$	$-\frac{16}{25}$	$\frac{3}{25}$			$\frac{12}{25}$
5	$-\frac{300}{137}$	$\frac{300}{137}$	$-\frac{200}{137}$	$\frac{75}{137}$	$-\frac{12}{137}$		$\frac{60}{137}$
6	$-\frac{360}{147}$	$\frac{450}{147}$	$-\frac{400}{147}$	$\frac{225}{147}$	$-\frac{72}{147}$	$\frac{10}{147}$	$\frac{60}{147}$



$A(\alpha)$ -稳定性和刚性稳定性数值方法的例 — Gear 方法

可以证明: Gear 方法是 $A(\alpha)$ -稳定和刚性稳定的 $\Leftrightarrow k \leq 6$.

Table: k 级 Gear 方法的 $A(\alpha)$ -稳定性和刚性稳定性常数

k	1	2	3	4	5	6
α_k	90°	90°	$88^\circ 27'$	$73^\circ 14'$	$51^\circ 50'$	$18^\circ 47'$
a_k	0	0	0.1	0.7	2.4	6.1

另一个刚性稳定性常数可由 $c_k = a_k \tan \alpha_k$ 给出.

注 1: Gear 方法的误差常数都不大.

注 2: 可以构造高阶的具有更好 $A(\alpha)$ -稳定性和刚性稳定性常数的数值方法. 如 Jain 方法和 Cryer 方法(特殊形式的线性隐式多步法).



$A(\alpha)$ -稳定性在实际应用中的意义

设刚性系统的特征值为 $\lambda_j = a_j + ib_j$, $a_j < 0$, $b_j \in \mathbb{R}^1$, $j = 1, \dots, m$, 且有 $|a_1| \leq |a_2| \leq \dots \leq |a_m|$. 则刚性方程组的解的各分量可表示为

$$y_k(t) = \sum_{j=1}^m C_j^{(k)} e^{(a_j + ib_j)t} + \phi_j(t), \quad k = 1, \dots, m,$$

其中 $C_j^{(k)}$, $k, j = 1, \dots, m$ 为由初值和方程组右端项决定的常数.

- 若系统的刚性比 $S \gg 1$, 但有 $|b_j|/|a_j| \leq \tan \alpha$, $j = 1, 2, \dots, m$, 则此时无论如何选时间步长 h , $\xi_j = \lambda_j h$, $j = 1, \dots, m$, 都落在 $A(\alpha)$ -稳定区域内.



刚性稳定性在实际应用中的意义

- 若系统的刚性比 $S \gg 1$, 但有 $|b_j|/|a_j| \leq \tan \frac{c}{a} = \tan \alpha$, $1 \leq j \leq m_1$, $|a_j| \gg 1$, $m_1 < j \leq m$. 则此时无论如何选时间步长 h , $\xi_j = \lambda_j h$, $j = 1, \dots, m_1$, 都落在 $A(\alpha)$ - 稳定区域内, 因此也落在刚性稳定区域内.

另一方面, 对应于 $j = m_1, \dots, m$ 的分量随着 t 的增长迅速衰减, 因此这些分量对解的贡献变得微乎其微. 此时, 只要时间步长 h 满足 $|a_j|h \geq a$, 即 $\lambda_j h$ 落在刚性稳定区域内, $m_1 < j \leq m$, 则相应分量的数值解尽管可能相对精度并不高, 但其绝对误差 (含舍入误差) 都将是微乎其微的, 因此它们对方程组数值整体解的影响也是很小的.



$A(\alpha)$ -稳定性和刚性稳定性都不适用的情况

- 若系统的刚性比 $S \gg 1$, 且有 $|a_j| \sim O(1)$, 但 $|b_j| \gg 1$, $1 \leq j \leq m_1 \leq m$. 则 $A(\alpha)$ -稳定性和刚性稳定性都不适用. 此时, 只有采用 A -稳定的数值格式.
- 在实际应用中, 在可能的情况下, 应该尽可能将时间 (或空间) 尺度差别很大的子系统解耦, 使每个子系统的刚性不太大, 或退一步, 使每个子系统可以应用 $A(\alpha)$ -稳定或刚性稳定的数值格式.
- 许多刚性问题也可从建模层面入手, 建立多尺度模型和多尺度数值方法.



N 个分子的理想的经典力学体系

与数值求解单个常微分方程相比, 数值求解常微分方程组的另一个主要困难来自于问题的规模. 对于大规模常微分方程组的数值求解, 计算量和内存需求量都成为挑战.

以分子动力学为例. 将 N 个分子的运动体系视为一个具有 N 个质点的理想的经典力学体系, 考虑建立并数值求解 N 个分子的运动方程.

一个有 N 个质点的经典力学体系的 Hamilton 量等于其总能量:

$$H(\mathbf{r}, \mathbf{p}) = K(\mathbf{p}) + U(\mathbf{r}) = \sum_{i=1}^N \frac{p_i^2}{2m_i} + U(\mathbf{r}),$$

其中 $K(\mathbf{p})$ 是系统的总动能, $U(\mathbf{r})$ 是系统的势能, p_i , m_i , r_i , 分别是粒子 i 的动量、质量和位置.



N 个分子的运动方程

N 个质点的经典力学体系的运动方程可用其 Hamilton 量表示为

$$\begin{cases} \frac{dr_i}{dt} = \frac{\partial H(\mathbf{r}, \mathbf{p})}{\partial p_i}, \\ \frac{dp_i}{dt} = -\frac{\partial H(\mathbf{r}, \mathbf{p})}{\partial r_i}. \end{cases}$$

在三维空间中, 这是一个有 $6N$ 个未知量的一阶常微分方程组.

对于一个有 N 个质点的经典力学体系, 已知系统的势能 $U(\mathbf{r})$, 则作用在粒子 i 上的力为 $F_i = -\nabla_i U(\mathbf{r})$, 因此, 由牛顿第二定律 $F = ma$, 我们也可以得到系统中每个粒子 i 的运动方程

$$m_i \frac{d^2 r_i}{dt^2} = -\nabla_i U(\mathbf{r}), \quad i = 1, \dots, N.$$

在三维空间中, 这是一个有 $3N$ 个未知量的二阶常微分方程组.



常用的分子动力学数值方法— 蛙跳格式

理论上说, 本章中介绍的各种常微分方程数值方法都可应用于求解分子动力学方程组. 但当 $N \gg 1$ 时, 为节省计算量和内存, 人们还是专门设计出了一些适用于大规模常微分方程组的数值方法. 例如, 常用于计算分子动力学问题的 Verlet 方法、蛙跳格式和速度 Verlet 方法. 首先介绍蛙跳格式:

$$\begin{cases} \mathbf{r}_{n+1} = \mathbf{r}_n + \mathbf{v}_{n+\frac{1}{2}} \Delta t, \\ \mathbf{v}_{n+\frac{1}{2}} = \mathbf{v}_{n-\frac{1}{2}} + \frac{\mathbf{F}_n}{\mathbf{m}} \Delta t, \end{cases}$$

其中 $\mathbf{v}_{n+\frac{1}{2}}$ 表示 $t_{n+\frac{1}{2}}$ 时刻的速度场, $\mathbf{F}_n/\mathbf{m}$ 表示按分量相除后得到的向量. 容易看出蛙跳格式关于位移场和速度场都具有二阶精度. 由于位移场和速度场是在交错网格上计算的, 因此, 要同时获得粒子在某时刻的位移和速度时, 需要对数值结果做后处理, 这往往会增加运算量并损失精度.



F 为常向量时蛙跳格式的稳定性

记 $\mathbf{e}_n^r = \mathbf{r}_n - \mathbf{r}(t_n)$, $\mathbf{e}_{n+\frac{1}{2}}^v = \mathbf{v}_{n+\frac{1}{2}} - \mathbf{v}(t_{n+\frac{1}{2}})$. 记 $\mathbf{R}_n^r$ 和 $\mathbf{R}_{n+\frac{1}{2}}^v$ 为格式的误差（含局部截断误差和舍入误差），则由定义有

$$\mathbf{e}_n \triangleq \begin{pmatrix} \mathbf{e}_n^r \\ \mathbf{e}_{n+\frac{1}{2}}^v \end{pmatrix} = \begin{pmatrix} I & \Delta t \cdot I \\ 0 & I \end{pmatrix} \begin{pmatrix} \mathbf{e}_{n-1}^r \\ \mathbf{e}_{n-\frac{1}{2}}^v \end{pmatrix} + \begin{pmatrix} \mathbf{R}_n^r \\ \mathbf{R}_{n+\frac{1}{2}}^v \end{pmatrix} = A\mathbf{e}_{n-1} + \mathbf{R}_n,$$

其中 I 为 $3N$ 维的单位矩阵. 于是, 归纳地有

$$\|\mathbf{e}_n\|_\infty \leq \|A^n \mathbf{e}_0\|_\infty + \sum_{k=0}^{n-1} \|A^k \mathbf{R}_{n-k}\|_\infty.$$

易证

$$A^k = \begin{pmatrix} I & k \cdot \Delta t \cdot I \\ 0 & I \end{pmatrix}.$$



F 为常向量时蛙跳格式的稳定性（续）

因此, 当 $n\Delta t \leq T$ 时, 记 $R_T = \max_{k \leq T/\Delta t} \|\mathbf{R}_k\|_\infty$, 有

$$\|\mathbf{e}_n^v\|_\infty \leq \|\mathbf{e}_0^v\|_\infty + n \max_{k \leq T/\Delta t} \|\mathbf{R}_k^v\|_\infty \leq \|\mathbf{e}_0^v\|_\infty + (\Delta t)^{-1} T R_T^v.$$

$$\begin{aligned} \|\mathbf{e}_n^r\|_\infty &\leq \|\mathbf{e}_0^r\|_\infty + n\Delta t \|\mathbf{e}_0^v\|_\infty + \sum_{k=0}^{n-1} (R_T^r + k\Delta t R_T^v) \\ &\leq \|\mathbf{e}_0^r\|_\infty + T \|\mathbf{e}_0^v\|_\infty + (\Delta t)^{-1} T (R_T^r + \frac{1}{2} T R_T^v). \end{aligned}$$

由此可见格式是零稳定的, 且当无舍入误差时关于速度和位移的收敛阶均为 $O(\Delta t^2)$. 但考虑舍入误差时格式不具有 $\Delta t/T \rightarrow 0$ 时的稳定性. 因此, 实际计算时需注意格式精度与 $(\Delta t)^{-1} T$ 倍的舍入误差间的平衡 (对刚性很大的问题, 麻烦大了!).



常用的分子动力学数值方法— Verlet 方法

Verlet 方法：由 Taylor 展开有

$$\mathbf{r}_{n+1} = \mathbf{r}_n + \mathbf{v}_n \Delta t + \frac{1}{2} \frac{\mathbf{F}_n}{\mathbf{m}} \Delta t^2 + \frac{1}{6} \frac{d^3 \mathbf{r}_n}{dt^3} \Delta t^3 + O(\Delta t^4),$$

$$\mathbf{r}_{n-1} = \mathbf{r}_n - \mathbf{v}_n \Delta t + \frac{1}{2} \frac{\mathbf{F}_n}{\mathbf{m}} \Delta t^2 - \frac{1}{6} \frac{d^3 \mathbf{r}_n}{dt^3} \Delta t^3 + O(\Delta t^4),$$

两式分别相加和相减得

$$\mathbf{r}_{n+1} = 2\mathbf{r}_n - \mathbf{r}_{n-1} + \frac{\mathbf{F}_n}{\mathbf{m}} \Delta t^2 + O(\Delta t^4)$$

$$\mathbf{v}_n = \frac{\mathbf{r}_{n+1} - \mathbf{r}_{n-1}}{2\Delta t} + O(\Delta t^2).$$



F 为常向量时Verlet 方法的稳定性

记 $\mathbf{e}_n^r = \mathbf{r}_n - \mathbf{r}(t_n)$. 记 $\mathbf{R}_n^r$ 为格式的误差 (含局部截断误差和舍入误差), 则由定义得误差方程

$$\mathbf{e}_{n+1}^r - 2\mathbf{e}_n^r + \mathbf{e}_{n-1}^r = \mathbf{R}_n^r,$$

该差分方程的特征方程为 $\lambda^2 - 2\lambda + 1 = 0$, 所以有 $\lambda_{1,2} = 1$. 由此知差分方程的通解可以表示为

$$\mathbf{e}_n^r = \mathbf{c}_1 + \mathbf{c}_2(n-1) + \sum_{k=1}^{n-1} (n-k)\mathbf{R}_k^r,$$

其中 $\mathbf{c}_1 = \mathbf{e}_0^r$, $\mathbf{c}_2 = (\mathbf{e}_1^r - \mathbf{e}_0^r) \triangleq \Delta t \mathbf{e}_0^v$. 由此得

$$\|\mathbf{e}_n^r\|_\infty \leq (\|\mathbf{e}_0^r\|_\infty + n\Delta t \|\mathbf{e}_0^v\|_\infty) + \sum_{k=1}^{n-1} k \|\mathbf{R}_{n-k}^r\|_\infty.$$



F 为常向量时Verlet 方法的稳定性（续）

当 $n\Delta t \leq T$ 时, 记 $R_T^r = \max_{k \leq T/\Delta t} \|\mathbf{R}_k^r\|_\infty$, 则有

$$\|\mathbf{e}_n^r\|_\infty \leq (\|\mathbf{e}_0^r\|_\infty + T\|\mathbf{e}_0^v\|_\infty) + \frac{1}{2}(\Delta t)^{-2}T^2R_T^r.$$

由此可见格式是零稳定的, 当无舍入误差时关于位移和速度的收敛阶分别为 $O(\Delta t^2)$, $O(\Delta t)$. 但有舍入误差时格式不具有 $\Delta t/T \rightarrow 0$ 时的稳定性, 因此, 实际计算时需注意格式精度与 $(\Delta t)^{-2}T^2$ 倍的舍入误差间的平衡. 这显然比蛙跳格式更为苛刻.



常用的分子动力学数值方法— 速度 Verlet 方法

Verlet 方法的速度场是用位移场的数值结果做数值微分得到的, 因此实际计算时会损失计算精度 (丢失有效位的个数). 如果对速度场做中心差商逼近, 则会有利于改善计算精度.

速度 Verlet 方法:

$$\begin{cases} \mathbf{r}_{n+1} = \mathbf{r}_n + \mathbf{v}_n \Delta t + \frac{1}{2} \frac{\mathbf{F}_n}{\mathbf{m}} \Delta t^2, \\ \mathbf{v}_{n+1} = \mathbf{v}_n + \frac{1}{2} \left(\frac{\mathbf{F}_n}{\mathbf{m}} + \frac{\mathbf{F}_{n+1}}{\mathbf{m}} \right) \Delta t, \end{cases}$$

速度 Verlet 方法关于位移场和速度场都具有二阶精度. 与蛙跳格式和 Verlet 方法相比速度 Verlet 方法的长处在于其更好的稳定性和较高的速度场精度, 不足之处则是较大的计算量.



F 为常向量时速度 Verlet 方法的稳定性

记 $\mathbf{e}_n^r = \mathbf{r}_n - \mathbf{r}(t_n)$, $\mathbf{e}_n^v = \mathbf{v}_n - \mathbf{v}(t_n)$. 记 $\mathbf{R}_n^r$ 和 $\mathbf{R}_n^v$ 为格式的误差 (含局部截断误差和舍入误差), 则由定义有

$$\mathbf{e}_{n+1} \triangleq \begin{pmatrix} \mathbf{e}_{n+1}^r \\ \mathbf{e}_{n+1}^v \end{pmatrix} = \begin{pmatrix} I & \Delta t \cdot I \\ 0 & I \end{pmatrix} \begin{pmatrix} \mathbf{e}_n^r \\ \mathbf{e}_n^v \end{pmatrix} + \begin{pmatrix} \mathbf{R}_n^r \\ \mathbf{R}_n^v \end{pmatrix} = A\mathbf{e}_n + \mathbf{R}_n.$$

于是有 $\|\mathbf{e}_{n+1}\| \leq \|A^n \mathbf{e}_1\| + \sum_{k=1}^n \|A^k \mathbf{R}_{n-k}\|$. 不难验证

$$A^k = \begin{pmatrix} I & k\Delta t \cdot I \\ 0 & I \end{pmatrix}.$$



F 为常向量时速度 Verlet 方法的稳定性(续)

因此, 当 $n\Delta t \leq T$ 时, 记 $R_T = \max_{k \leq T/\Delta t} \|\mathbf{R}_k\|_\infty$, 有

$$\|\mathbf{e}_n^v\|_\infty \leq \|\mathbf{e}_0^v\|_\infty + n \max_{k \leq T/\Delta t} \|\mathbf{R}_k^v\|_\infty \leq \|\mathbf{e}_0^v\|_\infty + (\Delta t)^{-1} TR_T^v.$$

$$\begin{aligned} \|\mathbf{e}_n^r\|_\infty &\leq \|\mathbf{e}_0^r\|_\infty + n\Delta t \|\mathbf{e}_0^v\|_\infty + \sum_{k=0}^{n-1} (R_T^r + k\Delta t R_T^v) \\ &\leq \|\mathbf{e}_0^r\|_\infty + T \|\mathbf{e}_0^v\|_\infty + (\Delta t)^{-1} T (R_T^r + \frac{1}{2} TR_T^v). \end{aligned}$$

由此可见格式是零稳定的, 且当无舍入误差时关于速度和位移的收敛阶均为 $O(\Delta t^2)$. 但格式不具有 $\Delta t/T \rightarrow 0$ 时的数值稳定性, 因此, 实际计算时需注意格式精度与 $(\Delta t)^{-1} T$ 倍的舍入误差间的平衡. 该格式比 Verlet 方法稳定性更好, 比蛙跳格式精度更高, 实际计算时(数值)稳定性和精度也确实更好, 尤其是 T 很大时.



习题六：9； 上机习题六：2, 4.

Thank You!

