

改进的Ahmad-Cohen方法

——引力 N 体问题的一个积分方法

刘步林

(中国科学院北京天文台)

提 要

本文叙述由S. J. Aarseth发展的Ahmad-Cohen方法,以及在VAX780计算机上运行的程序。

一、概 述

五十年代末,出现高速电子计算机,随之产生了前所未有的计算机模拟方法。它在力学、等离子体物理、天文学等学科中有着广泛的应用。 N 体问题是天体力学、星系动力学和等离子体动力学等分支学科的基本问题之一。天体系统的实际情况有多重星(星数 $N=2-10$),疏散星团($N=10^2-10^3$),球状星团($N=10^5-10^7$),星系($N=10^{11}$ 以上),双星系或多重星系(星系数 $=2-10$),星系团(星系数 $\leq 10^2$),它们中的星数都很多。但是直到今天, N 体问题只有十个初积分,三体以上的问题都不可能得到完全解。所以要研究 N 这样大的天体组成的引力系统的运动规律、结构和演化过程,现在多数转而依赖于用电子计算机作数值模拟。

N 体问题计算机模拟的可行性,早在1960年由海德堡天文研究所Sebastian Von Hoerner提出利用计算机研究碰撞星系的动力演化的试探中,作出了肯定的结论^[1]。因此可以说Sebastian Von Hoerner是 N 体问题计算机模拟的先驱者之一。自那以后,1967年,国际天文学联合会在巴黎举行了“ N 体问题讨论会”^[2],着重讨论了用计算机对星系进行计算模拟的问题。1970年,国际天文学联合会又在剑桥大学召开“引力 N 体问题讨论会”^[3],广泛交流了用分析方法、数值方法和用计算机进行数值模拟方法等研究引力 N 体问题的技巧和结果,展示了计算机模拟是星系动力学和等离子体动力学研究中的有效手段之一。1972年,在美国奥斯汀又召开了关于常微分方程的数值解会议,讨论了多种数值模拟方法。近年来,关于 N 体问题的学术论文愈来愈多,学术活动日见频繁,这表明 N 体问题是国际科学界十分活跃的研究课题。

用计算机模拟求解引力 N 体问题,常用的有三种数值模拟方法。即数值积分法、网格差分法和蒙特卡罗法。Ahmad-Cohen方法是数值积分方法中的一个十分重要的方法。它是由纽约大学Ahmad和Cohen于1973年在《计算物理》杂志上发表的论文^[4],用于解引力 N 体问题的一个数值积分方法,简称AC方法。

N 体问题数值模拟的基本过程是, 当给定 t_0 时刻一个天体系统 N 颗星的质量、位置和速度, 通过数值积分方法求解牛顿运动方程, 计算出在以后任何时刻系统的状态, 有了该系统一系列不同时刻每颗星的位置和速度, 就可以用来研究该天体系统特殊的物理现象和力学特征。

利用数值积分方法讨论引力 N 体问题的主要优点是: 方程形式简单, 对问题的设想比较自由, 而且可以包括所有可能的作用因素在内^[4]。计算方法比较完善, 可以直接得到天体的坐标和速度, 因而便于对天体系统进行比较详细的讨论。

Ahmad-Cohen方法是在 Aarseth 单时步方法的基础上, 提出邻域球的概念, 把作用在第 i 颗星上的力分为规则力(邻域球外部天体作用力)和不规则力(邻域球内天体作用力), 不同的力采用不同的时步, 因而形成双时步方法。在这之后, Aarseth将 Ahmad-Cohen 方法又作了发展, 例如, 时步的选取, 邻域球外加壳层等, 致使 Ahmad-Cohen方法更加完善。下面将要介绍的是改进后的 Ahmad-Cohen方法。有的学者认为, 直接积分方法有可能用来解决恒星系统的稳定性问题, 这就使得直接积分方法的研究具有更深远的意义。但是由于直接方法计算非常繁琐, 占用机时和内存都很多, 直到现在, 用直接积分方法研究天体系统, 天体数目只能达到 10^2-10^3 量级, 所以方法本身还有待于进一步探讨。

二、改进的 Ahmad-Cohen 方法

1. 基本公式

在牛顿引力场中, 引力 N 体问题的运动方程可以表示为:

$$\ddot{\mathbf{r}}_i = -G \sum_{\substack{j=1 \\ j \neq i}}^N \frac{m_j (\mathbf{r}_i - \mathbf{r}_j)}{|\mathbf{r}_i - \mathbf{r}_j|^3} \quad i=1, 2, \dots, N \quad (1)$$

这里 m_i 和 \mathbf{r}_i 分别表示第 i 颗星的质量和坐标。 G 是万有引力常数, 适当选取长度、质量和时间单位使 $G=1$ 。(1) 式右端求和是在其他 $N-1$ 颗星成员中进行的, 这是个非线性强对偶力的系统。有两个基本因素使得(1) 式积分存在困难, 一是两颗星紧密接近, 导致系统的不稳定。二是因为作用在每颗星上的力依赖于其他所有的质点。因此, 计算力所需的时间随着天体数目 N 的平方增加而增加, 这就限制了较大数目天体系统的研究。这些困难限制了常规数值方法(例如龙格-库特方法等)的使用, 迫使人们去寻找新的方法, Ahmad-Cohen 方法就是一个比较有效的方法。

用 Taylor 级数展开的方法, 将天体的坐标和速度从 t_0 到 t 展开, 得出 4 阶的 Taylor 级数多项式为:

$$\mathbf{r}_i(t) = \mathbf{r}_i(t_0) + \mathbf{v}_i(t_0)\Delta t + \frac{1}{2} \frac{\mathbf{F}_i(t_0)}{m_i} (\Delta t)^2 + \sum_{k=1}^4 \frac{\mathbf{F}_i^{(k)}(t_0) (\Delta t)^{k+2}}{m_i (k+2)!} \quad (2)$$

$$\mathbf{v}_i(t) = \mathbf{v}_i(t_0) + \frac{\mathbf{F}_i(t_0)}{m_i} \Delta t + \sum_{k=1}^4 \frac{\mathbf{F}_i^{(k)}(t_0) (\Delta t)^{k+1}}{m_i (k+1)!} \quad (3)$$

其中 $\Delta t = t - t_0$, $\mathbf{F}_i(t_0)$ 是 t_0 时刻作用在第 i 个天体上力的总和, $\mathbf{F}_i^{(k)}(t_0)$ 是 t_0 时刻 $\mathbf{F}_i(t_0)$ 对时间 t 的第 k 次导数。

作用在第 i 颗星上的力 $F(t)$ 可以展成 Taylor 级数多项式或均差多项式 (the divided difference polynomial), 四阶均差多项式定义为:

$$\begin{aligned} F(t) = & F(t_0) + D^1[t_0, t_1](t - t_0) \\ & + D^2[t_0, t_2](t - t_0)(t - t_1) \\ & + D^3[t_0, t_3](t - t_0)(t - t_1)(t - t_2) \\ & + D^4[t, t_3](t - t_0)(t - t_1)(t - t_2)(t - t_3) \end{aligned} \quad (4)$$

这里 t_3, t_2, t_1, t_0 是最后四个精确力的计算时间, t_0 是最近的, $D^k (k=1, 2, 3, 4)$ 称为 k 阶均差, $k=1$ 时可以不写。各阶均差的定义如下:

$$\left. \begin{aligned} D^1[t_0, t_1] &= \frac{F(t_0) - F(t_1)}{t_0 - t_1} \\ D^2[t_0, t_2] &= \frac{D^1[t_0, t_1] - D^1[t_1, t_2]}{t_0 - t_2} \\ D^3[t_0, t_3] &= \frac{D^2[t_0, t_2] - D^2[t_1, t_3]}{t_0 - t_3} \\ D^4[t, t_3] &= \frac{D^3[t, t_2] - D^3[t_0, t_3]}{t - t_3} \end{aligned} \right\} \quad (5)$$

从(5)式可以看出, 均差和差分的产生有些类似, 若知道 t_3, t_2, t_1, t_0, t 时刻的力 $F(t_3), F(t_2), F(t_1), F(t_0), F(t)$, 则一阶至四阶均差的产生方法如下:

$$\begin{array}{ccccccc} t_3, F(t_3) & & & & & & \\ & \searrow & & & & & \\ t_2, F(t_2) & & D^1[t_2, t_3] & & & & \\ & \searrow & & & & & \\ t_1, F(t_1) & & D^1[t_1, t_2] & & D^2[t_1, t_3] & & \\ & \searrow & & & \searrow & & \\ t_0, F(t_0) & & D^1[t_0, t_1] & & D^2[t_0, t_2] & & D^3[t_0, t_3] \\ & \searrow & & & \searrow & & \searrow \\ t, F(t) & & D^1[t, t_0] & & D^2[t, t_1] & & D^3[t, t_2] \\ & & & & & & \searrow \\ & & & & & & D^4[t, t_3] \end{array} \quad (6)$$

均差多项式只不过是内插多项式的变形罢了, 实际上就是内插多项式。

力 F 的 Taylor 级数展开式(4阶)为:

$$F(t) = F(t_0) + \sum_{k=1}^4 F^{(k)}(t_0) \frac{(t - t_0)^k}{k!} \quad (7)$$

欲要以均差表示力的导数, 可以令(4)式力的均差多项式和(7)式力的 Taylor 多项式相等, 然后令等式两边 $(t - t_0)$ 的等阶项的系数相等, 于是得到:

$$\left. \begin{aligned} F^{(1)} &= D^1[t_0, t_1] + t_1' D^2[t_0, t_2] + t_1' t_2' D^3[t_0, t_3] + t_1' t_2' t_3' D^4[t_1, t_3] \\ F^{(2)} &= 2! \{ D^2[t_0, t_2] + (t_1' + t_2') D^3[t_0, t_3] \\ &\quad + (t_1' t_2' + t_2' t_3' + t_1' t_3') D^4[t, t_3] \} \\ F^{(3)} &= 3! \{ D^3[t_0, t_3] + (t_1' + t_2' + t_3') D^4[t, t_3] \} \\ F^{(4)} &= 4! D^4[t, t_3] \end{aligned} \right\} \quad (8)$$

其中 $t_k' = t_0 - t_k, k=1, 2, 3$ 。(8)式表明四阶均差 D^4 对每一阶力的导数都有贡献, 而且这一

贡献仅仅是在每一个公式后面加上一项。从均差产生方式(6)式来看,此项只是在预测到 t 时刻的力 $F(t)$ 后才可得到,这称为半迭代过程(semi-iteration procedure),半迭代过程稍许多花费些机时,但能提高精度,而不需要过多的贮存。上述为数值积分的主要公式。

下面叙述在置初值过程中,各阶导数的求得。在给定初始条件 $m_i, r_i, v_i, i=1, 2, \dots, N$ 时,(7)式各阶Taylor级数的系数 $F^{(i)}$ ($i=1, 2, 3,$)可以通过对方程(1)连续求导数产生^[5]。在下面表达式中的量是以相对坐标 $R=r_i - r_j$ 和相对速度 $v=v_i - v_j$ 表示的。

$$\left. \begin{aligned} F_{ij} &= -m_j R/R^3 \\ F_{ij}^{(1)} &= -m_j v/R^3 - 3aF_{ij} \end{aligned} \right\} \quad (9)$$

$$\left. \begin{aligned} F_{ij}^{(2)} &= -m_j (F_i - F_j)/R^3 - 6aF_{ij}^{(1)} - 3bF_{ij} \\ F_{ij}^{(3)} &= -m_j (F_i^{(1)} - F_j^{(1)})/R^3 - 9aF_{ij}^{(2)} - 9bF_{ij}^{(1)} - 3cF_{ij} \end{aligned} \right\} \quad (10)$$

其中:

$$a = R \cdot v/R^2$$

$$b = (v/R)^2 + R \cdot (F_i - F_j)/R^2 + a^2$$

$$c = 3v \cdot (F_i - F_j)/R^2 + R \cdot (F_i^{(1)} - F_j^{(1)})/R^2 + a(3b - 4a^2)$$

假定已经知道一个适当的初始时步 δt_i (下面说明如何决定),对 $t_0=0$,置反向时间, $t_k = -k\delta t, k=3, 2, 1$,那么均差初始值通过对方程(8)求逆得到:

$$\left. \begin{aligned} D^1 &= F^{(1)} - (F^{(2)}/2 - F^{(3)}/6)t_1' \\ D^2 &= F^{(2)}/2 - F^{(3)}/6(t_1' + t_2') \\ D^3 &= F^{(3)}/6 \end{aligned} \right\} \quad (11)$$

这里只到 $F^{(3)}$ 的项。只要记住力的两类多项式的形式及联系,利用半迭代过程,考虑到四阶多项式是很容易的。在(9)、(10)两式的计算中,形式为 $m/(R^2 + \epsilon^2)^{1/2}$ 的软势(soft potential)很容易被引进来。软势的引入,使得两颗非常接近的星变得容易处理。

2. 单时步算法

单时步算法是力的多项式的一个简单的数值积分方法,单时步算法就是标准多项式方法,它是Ahmad-Cohen方法(双时步算法)的基础。应用单时步算法,可以在比较宽广的时间尺度上得到一致解(consistent solutions),而且能保持相当的精度,因此能够使得每颗星的位置以最大的时步计算。它的特点是在积分过程中,必须对整个坐标作中间时刻的预测。

主要计算步骤归纳如下:

(1) 用产生伪随机数的方法,产生系统中每一个天体的质量(或等质量)、坐标和速度,再换算成相对于天体系统质量中心的坐标和速度。

(2) 用(9)和(10)式计算系统中每一天的力和力的导数, $F, F^{(1)}, F^{(2)}, F^{(3)}$ 。

(3) 计算每一天的积分时步:

$$\delta t = \left[\eta \frac{|F| |F^{(2)}| + |F^{(1)}|^2}{|F^{(1)}| |F^{(3)}| + |F^{(2)}|^2} \right]^{1/2} \quad (12)$$

这个公式包含力和力的导数,是个混合表达式,它使得所有导数在时步测定中起作用。因此它能适应一些特殊的情况,例如在考虑潮汐力时,就可能出现合力 $F=0$ 的情况,或者 $F^{(1)}=F^{(2)}=0$ 的情况。尽管出现这些情况,从(12)式可以看出,并不影响时步的计算。时步

计算公式不同于 Ahmad-Cohen 文章^[4]中的计算公式, 这是由 Aarseth 经过多次试验之后采用的表达式。

虽然连续时步是以平滑方式正常变化, 但仍需要设法限制它增长, 对于一个标准积分来说, 采用一个稳定因子, 限制积分时步的增长。式中 η 是个精确的无量纲参数, 有了 η 就可控制总的精度, η 起到一个相对判据的作用。一般情况下, 标准值是 $\eta=0.03$ 。

(4) 决定均差多项式中力的四个时刻 t_k , $k=3, 2, 1, 0$, 以 t 表示积分历元 (初始时刻 $t=0$)。

$$t_3 = t - 3\delta t_i, \quad t_2 = t - 2\delta t_i$$

$$t_1 = t - \delta t_i, \quad t_0 = t$$

(5) 用(11)式计算均差 D^1, D^2, D^3 。

(6) 计算初始时步目录间隔 $DTLIST (= 0.5 \sum_{i=1}^N \delta t_i / N)$, 为了避免对系统中 N 个天体逐一搜索, 引进被积分天体时步目录 $NLIST(i)$, $i=1, 2, \dots, n$, 凡 $\delta t_i < DTLIST$ 的天体为被积分天体。天体序号存放于被积分天体的时步目录 $NLIST(i)$ 中, $i=2, 3, \dots, n$, 一般 $n=N^{1/2}$ 。 $NLIST(1)$ 为存放被积分天体的总数。

决定下一次修改时步目录的时间 $TLIST$ 。 $TLIST = TLIST + DTLIST$, $TLIST$ 的初值为零。

(7) 判别历元 t 是否等于零, 等于零则表明还未进入积分, 进入第(8)步; 不等于零则用(2)式和(3)式进行位置和速度的预测。

(8) 计算系统的势能和动能, 及确定标准穿透时间 TCR 。输出结果: 积分时间 t , 单时步积分记数 $NSTEPN$, 能量 $ENERGY$, 以及部分或全部天体的质量 m , 时步 δt , 位置 r 和速度 v 。

(9) 主要积分循环是从决定下一个被积分天体开始的, 也就是确定 $i = \min_j (t_j + \delta t_j)$, $j=1, 2, \dots, N$ 的那一颗天体, 然后决定被积分天体 i 的当前的积分历元 t , $t = t_i + \delta t_i$ 。

如果积分历元 t 大于等于修改时步目录的时间 $TLIST(i)$, 则调整时步目录间隔 $DTLIST$, 形成新的被积分天体的时步目录 $NLIST$, 重新进行积分循环。

(10) 用下式将系统中 N 个天体坐标预测到一阶,

$$r_i = [(F^{(1)}/6\delta t_j + F/2)\delta t_j + v_j]\delta t_j + r_i(t_j) \quad (13)$$

用(2)式和(3)式将第 i 天体坐标和速度预测到三阶。

(11) 计算作用在第 i 天体上的力 [用(1)式], 修改时间 t_k , 即 $t_3 = t_2$, $t_2 = t_1$, $t_1 = t_0$, $t_0 = t$, 将第 i 天体的位置和速度预测到四阶 (半迭代过程), 得到新的坐标和速度, 代替原来的坐标和速度。计算第 i 天体新的均差 D^1, D^2, D^3 。

(12) 计算第 i 天体新的时步 δt_i , 用(12)式。

(13) 如果积分时间 t 大于、等于结果输出时间 $TNEXT$ ($TNEXT = TNEXT + DELTAT$, $DELTAT$ 为输出时间间隔, 是输入值, $TNEXT$ 初值为零), 则转到第(8)步, 否则转到第(9)步。

单时步方法每颗星需要33个变量: $m, r(t_0), r(t), v(t_0), v(t), F, F^{(1)}, D^1, D^2, D^3, \delta t, t_0, t_1, t_2, t_3$ 。当计算机的字长的十进制有效位数小于10位时, 例如 VAX 机和 IBM 370 等计算机, 为了保证精度, 需要将 $r(t_0), r(t), v(t_0), t_0$ 等量用双精度表示, 这样每颗

星至少需要43个变量单元。

类似的单时步方法,早在六十年代由 Aarseth 和 Wielen 等人使用过。后来经过 Aarseth 的总结完善,形成现在的单时步方法,即标准多项式方法。

3. 改进的 Ahmad-Cohen方法

单时步方法对系统中的天体逐一积分,用它来研究系统的运动规律,需占用相当长的机时。为了缩短机时,同时又要降低精度,需要寻找新的途径。Ahmad 和 Cohen两人在 Aarseth 的单时步方法的基础上提出双时步方法。他们研究天体系统实际情况时发现,对于第 i 天体来说,其他天体对它的作用力 F 在不断地变化之中,靠得近的变化快,离得远的变化慢。于是他们提出邻域球 (neighbor sphere) 的概念。设想系统中每一个天体以半径 R_i 的球包围着。邻域内的天体对第 i 个天体的作用力变化快,是不规则力;邻域外部天体对第 i 天体的作用力是长程力,变化较缓慢,是规则力。规则力和不规则力用两个不同的多项式逼近,分别具有不同的时步,一是规则时步,一是不规则时步;为了减少力的计算时间,两个不同时步是完全需要的。Aarseth 将邻域球的外面又加了一个壳层,壳层的外半径为 $2^{1/3}R_i \approx 1.26R_i$,壳层起着—个缓冲带的作用,正如前面叙述的,用相对坐标和相对速度对壳层内的天体逐一判别,这可以使那些快速运动的天体,在穿进邻域球之前就能辨别出来,以便区别对待。

每个天体都有一个邻域球序列目录 $LIST(i, I)$, $I=1, 2, \dots, N$; $i=1, 2, \dots, n$, $n=N^{1/2}$ 。罗列邻域中的所有天体,其中 $LIST(1, I)$ 存放第 I 天体邻域球中天体的总数。积分开始时,每一个天体具有相同的邻域球半径 R_i ,随着系统内天体的运动,系统内各处密度是不等的。由于限制了邻域内允许最大的天体数目,必须自动调节邻域半径。结果使得邻域球中的天体数目既不会超过最大值,也不会使邻域处于空白状态。

邻域球半径自动调节的方式是,首先给出邻域球半径初值 R_i ,然后计算半质量半径 $\langle R \rangle$ ($\langle R \rangle = 0.5M^2/\text{势能}$) 和邻域中的星数 L_1 ,计算密度 C :

$$C = \frac{2L_1}{N} \left(\frac{\langle R \rangle}{R_i} \right)^3 \quad (14)$$

有了 C 值,可以用来预测邻域球中的星数 L_p ,

$$L_p = 0.75L_{\max}(C/20)^{1/3} \quad (15)$$

式中 L_{\max} 是邻域中允许拥有星数的最大值 ($=N^{1/2}$),要求 L_p 满足: $2 \leq L_p \leq 0.75L_{\max}$,超出范围的,以区域边值代替。要求 L_p/L_1 满足: $0.75 \leq L_p/L_1 \leq 1.25$,超出的以边值代替。于是新的邻域球半径 R_i^{new} 由下式算得:

$$R_i^{\text{new}} = R_i^{\text{old}}(L_p/L_1)^{1/3} \quad (16)$$

有了新的邻域球半径 R_i^{new} ,就能重新决定邻域的序列目录 $LIST(i, I)$,只要比较新老邻域球的序列目录,就能够决定进入和离开邻域球的星数。

邻域球的任何变化,都迫使两个力的多项式要作相应的改变。在均差的计算中,首先计算规则力 $F_d(t)$, $F_d(t_0)$,然后计算邻域中的不规则力 F_n^{new} 和 F_n^{old} ,给出新的规则力差分:

$$D^1 = [F_d(t) - (F_n^{\text{old}}(t) - F_n^{\text{new}}(t)) - F_d(t_0)] / (t - t_0) \quad (17)$$

式中 $F_n^{\text{old}}(t) - F_n^{\text{new}}(t)$ 表示邻域球中不规则力的净变化。力多项式的修正是在相应的 Taylor 级数导数(8)式上,加上或减去相应的改正项,而后再利用(11)式变换得到均差。

给定初值 m, r, v 后, 两个力的多项式需要置初值。进入积分前的主要步骤和单时步方法基本一致, 积分的主要轮廓也和单时步方法一样, Aarseth 将积分循环总结为 15 步:

- (1) 决定下一个被积分的天体, 即 $i = \min_j(t_j + \delta t_j)$, 将 $t = t_i + \delta t_i$ 定为当前的历元。
- (2) 如果历元 t 大于修改时步目录时间 TLIST, 修改被积分天体的时步目录 LIST, 即改进时步目录间隔 DTLIST。
- (3) 比较规则时间 $t_r + \delta t_r$ 和不规则时间 $t + \delta t$ 。如果 $t + \delta t$ 大于 $t_r + \delta t_r$ 为情况 2, 否则为情况 1。
- (4) 当为情况 1 时, 作邻域坐标预测; 当为情况 2 时, 用 (13) 式将所有天体的坐标计算到一阶 $F^{(1)}$ 。
- (5) 将第 i 颗星的坐标 $r(t)$ 和速度 $v(t)$ 预测到 $F^{(3)}$ 。
- (6) 得到邻域内不规则力 F_n^{old} , 修改不规则均差多项式的时间 $t_k, k=3, 2, 1, 0$, 即:

$$t_3 = t_2, t_2 = t_1, t_1 = t_0, t_0 = t$$
- (7) 形成新的不规则均差, 包括半迭代过程, 即将被积分天体的坐标和速度作四阶改正。
- (8) 仅当为情况 1 时, 外推规则力 $F_d(t)$ 和一阶导数 $F_d^{(1)}(t)$, 得到新的力 $F(t)$ 和一阶导数 $F^{(1)}(t)$ 。然后转到第 (14) 步。
- (9) 当为情况 2 时, 算新的规则力 F_d 和不规则力 F_n^{new} , 形成新的邻域目录, 包括密度的稀疏情况 $L_1 = 0$ 和 $L_1 > L_{max}$ 的情况。
- (10) 修改邻域半径 R_s , 且修改规则时间 $t_{rk} (k=3, 2, 1, 0)$, 即: $t_{r3} = t_{r2}, t_{r2} = t_{r1}, t_{r1} = t_{r0}, t_{r0} = t$ 。
- (11) 用 (17) 式计算新的规则力一阶均差及二阶、三阶均差。用 (11) 式得到规则力 $F(t)$ 和 $F^{(1)}(t)$, 加上半迭代过程将坐标和速度计算到 $F^{(4)}$ 。
- (12) 判别邻域中成员的得失, 计算导数改正, 然后用变换运算 (11) 式得到均差。
- (13) 用 (12) 式计算新的规则时步 δt_r 。
- (14) 用 (12) 式计算新的不规则时步 δt 。
- (15) 重复第 (1) 步循环。

改进的 Ahmad-Cohen 方法, 需要下列基本变量:

初始值: $m, r(t_0), r(t), v_0, v, F, F^{(1)}$ 。

两套多项式变量: F_n (或 F_d), $D^1, D^2, D^3, \delta t, t_0, t_1, t_2, t_3$,

邻域球半径: R_s ,

邻域球目录: LIST (数目为 $L_{max} + 1$),

总的贮存量, 每颗星需要 $55 + L_{max}$,

$L_{max} = N^{1/2}$, 对于多数情况是足够的, 通常取 $L_{max} = 30$, 那么一颗星需要 85 个变量单元。当将 $r(t_0), r(t), v(t_0), t_0$ 取为双精度时, 每颗星需要 95 个变量单元。

三、Aarseth 程序

Aarseth 用数值积分方法, 编制若干程序, 为世界各地天文学家引用。这些程序是逐步

深入,如同阶梯一样,一个高过一个。例如文件名分别为 NBODY1, NBODY2, NBODY3 的三个 N 体模拟程序:

NBODY1——标准多项式程序,单时步方法。580条。

NBODY2——改进的Ahmad-Cohen方法程序,双时步方法。1388条。

NBODY3——正规化的 N 体模拟程序(Regularized N -BODY Program),处理系统中天体相互碰撞的问题。2976条。

前两个程序的数值方法,本文已作系统介绍。NBODY1 和 NBODY2 两个程序不能做碰撞方面的研究,适合于研究软势问题、耗散效应(dissipative effects)[例如潮汐场(tidal fields)或者超新星质量损失(supernova mass loss)等]以及穿透时间较短的问题 (small number of crossing times)等。

程序NBODY1和 NBODY2 是用 FortranIV 编制而成。可以直接在VAX机或 IBM370/165 机上运行。这两个程序的结构轮廓为以下六个部分:

MAIN——主程序,控制子程序的调用顺序。

MYDUMP——从磁盘或磁带读写公用块变量,起到信息转贮作用。

DATA——为‘START’子程序提供初始条件。

START——为积分做准备工作。

MAINPT——包括输出和误差控制。

INTGRT——进行积分。

感谢S. J. Aarseth教授热情提供 N 体模拟程序!

参 考 文 献

- [1] Sebastian Von Hoerner, *Z. Astrophys.*, 50 (1960), 184.
- [2] Colloque sur le Probleme des N Corps, Proceedings of IAU Colloquium held in Paris, France, August 1967, *Bull. Astron.*, Tome, 3.
- [3] Gravitational N -body Problem, Proceedings of IAU Colloquium No.10 held in Cambridge, England, August 12—15, 1970.
- [4] Ahmad, A. and Cohen, L., *Journal of Computational Physics*, 12 (1973), 389—402.
- [5] Carlos Cruz Gonzalez and Myron Lecar, *Bulletin Astronomique*, 3 (1968), 209.

The Improved Ahmad-Cohen Scheme — A Numerical Integration Scheme for the Gravitation N -Body Problem

Liu Bulin

(Beijing Observatory, Academia Sinica)

Abstract

This paper describes Ahmad-Cohen Scheme, developed by S. J. Aarseth, as well as Aarseth's FORTRAN'S routine On computer VAX780.