

宇宙大尺度结构数值模拟中的短程力

宋国玄 陈兴明

(中国科学院上海天文台, 上海 200030)

摘 要

本文提出了宇宙大尺度结构 P³M 法数值模拟中短程力计算的改进方法。其改进主要在于对质点分块重新编号, 从而大大减少检索所需时间。为了检验该算法的有效性, 我们和国际上著名的 Efstathiou 等人的算法进行了比较, 结果表明, 在相同的初始条件下, 在同类型机器上运算, 我们的计算短程力算法要比他们的快一倍到五倍。

关键词: 宇宙学, 星系成团, 数值方法, 短程力

、 引 言

宇宙大尺度结构的数值模拟研究是 N 体问题在天体物理中应用的一个重要方面。但它又不同于星系的数值模拟研究。星系数值模拟主要研究星系的总体性质, 如三轴椭圆星系的塌缩形成, 盘状星系中大尺度棒状不稳定性, 旋臂在盘状星系中的展开与维持等等。而在宇宙大尺度结构研究中, 主要讨论纤维状结构、空洞、星系团等等的内部局部结构。

在 N 体问题中, 关键的一步是计算模拟质点的自引力。到目前为止, 自引力的计算可以分为三大类: 1. PP 法; 2. PM 法; 3. P³M 法。一般来说, PP 法的自引力计算是直接的, 它可以达到很高的分辨率, 但计算所费时间却与计算质点数的平方成正比。在大质点数时, 所耗时是惊人的, 因此对 PP 法也有各种改进^[1,2]。PM 法引入了快速 Fourier 变换, 把微分方程化成差分方程后可以大大节约计算机时。但 PM 法的一个致命弱点是分辨率较低, 利用该算法计算的自引力在 2—3 个网格内被抹平。这在研究星系的总体性质时是基本满足要求的, 并被广泛应用。但宇宙大尺度结构研究着眼于其内部结构, 而使用分辨率较差的 PM 法往往使相邻质点间的引力减小, 因此数值模拟的成团等聚集现象并没有真正反映宇宙中的实际过程。P³M 法就是为了弥补这一缺点而引入的, 这是在 PM 法基础上的一个修正。因为 PM 法主要在 2—3 个网格距离内把自引力抹平, 所以 Hockney 等建议^[3]: 首先对每一质点的受力用 PM 法来求解, 然后把每一质点由邻近质点所产生的力部分除去, 代之以用更精确的直接方法求得的近距离力。这就是 P³M 法的由来: PP 法加上 PM 法。1985 年 Efstathiou 等^[4]在把 P³M 法应用到宇宙大尺度结构研究时发现, 在 PM 法中对 Green 函数作适当修正后, 对长程力部分的算计结果不会产生任何影响, 而对短程力的数值大小可以大大降低。Green 函数

是在用 PM 法计算自引力前已一次预先算好的,其后不需消耗任何机时.这样在 P³M 法中可以不扣除局部力的效应,只需简单地把短程力的修正加进由 PM 法得到的结果上就可.当然,在一定范围内短程力的加入会在截止长度 r_s 处(即短程力的影响仅在 r_s 之内)产生一定的间断跳跃.文献[4]指出,在 PM 法中,若设一维方向上有 32 个网格, $r_s = 2.7$ 个网格长度,则在 r_s 处力的跳跃仅为 2.4%,这是一个小量.在 P³M 法中由对 PM 校正而引入的 PP 法是极费机时的. Efstathiou 等^[4]根据文献[3]的思想对短程力的计算编出了程序,并用于解决宇宙大尺度结构的许多问题.因此他们的算法国际上公认可靠而又省时.本文对 P³M 法中短程力的计算提出了一个新的方法.这一方法在分辨率上可达到文献[4]中的精度,但在计算机时要比他们的节省.

二、短程力算法的描述

正如前所述, P³M 法是对 PM 法在分辨率上的一个改进.这一改进主要在于计算每一质点附近距离 r_s 内的其它质点对它产生的近距离力.这样的近距离力若用 PP 法直接计算,虽然对每一质点仅在距离小于 r_s 内的其它质点需要考虑,但距离的判断也是极费机时的.这一直接 PP 算法的方框图如图 1 所示.

这一算法费机时的原因主要在于要判断每对质点其相互距离是否大于截止长度 r_s . 这样的算法并没有充分利用截止长度 r_s 的引入. 因此对短程力算法的改进也就在于如何利用 r_s .

为 Hockney 等提出,又为 Efstathiou 等实现的短程力算法对此作了些大改进.

本文对文献[3, 4]的算法再作一改进,相应的与文献[4]之算法作一比较.

在宇宙大尺度结构数值模拟中,总是对有限质点进行研究.这些有限质点被限制在一立方体(也可以是长方体)中.设此立方体每边长为 L . PM 法就在边长为 L 的立方体网络内进行,短程力的计算限制在 r_s 内.

再作一网络,其每边长度为 $N_L r_s$,其中 $N_L = \text{Int}(L/r_s) + 1$,并将其分成每边长为 r_s 的网格,即此网格数为 $N_L \times N_L \times N_L$. 该网络的一个顶点与原来立方体的一顶点重合,显然,后者包含在前者之中.

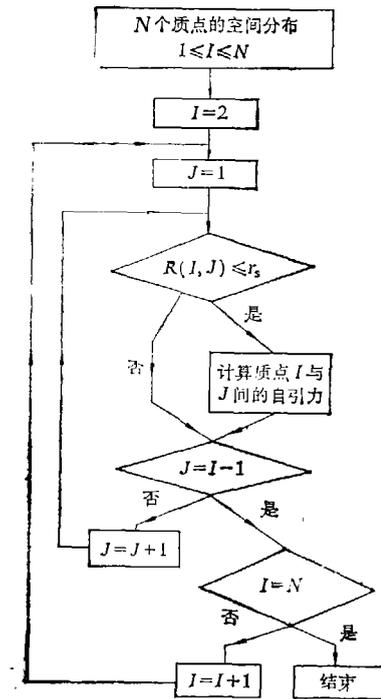


图 1 PP 法方框图

这一网络中每一网格内的质点,由于截止长度 r_s 的限制,其短程力的影响只可能是在同一网格内的质点及其邻接的其它 26 个网格内的质点产生的力.这样的分割使距离的判断减少了很多.文献[3, 4]在这一步上也是这样做的.

原先的质点是编号的,并据此得到质点的位置和速度,它们的值已由前面的计算存储起来.但顺序编号的质点可以分布在完全不同的网格内,因此问题是如何给出在某一网格内所存在的质点.

为此, 我们引入 3 个一维数组: $MM(N_L \times N_L \times N_L)$, $NUR(N_L \times N_L \times N_L)$ 和 $NTT(N)$. 然后进行下列运算:

1. 对任一网格 $(I, J, K; I, J, K = 1, 2, \dots, N_L)$ 予以一维排列:

$$LL = K + N_L \times (I - 1) + N_L \times N_L \times (J - 1);$$

2. 将第 LL 个网格内包含的质点数目记入 $MM(LL)$;
3. 将第 LL 个网格以前的所有网格中的质点数记入 $NUR(LL)$, 即

$$NUR(LL) = \sum_{KT=1}^{LL-1} MM(KT);$$

4. 利用 $MM(LL)$ 及 $NUR(LL)$ 将质点的编号依网格的顺序分别记入数组 $NTT(LLT)$, 其中 LLT 是新编号, NTT 是原编号.

在进行这样处理后, 若要找第 LL 个网格中所有质点的编号, 只要由 $NTT(LLT)$ 中取:

$$NUR(LL) + 1 \leq LLT \leq NUR(LL) + MM(LL) \text{ 的值即可.}$$

文献[3,4]的算法中, 在把质点依网格分类时把相应质点的坐标与速度值输入一维存储. 实际上, 在整个程序中, 坐标与速度值已由前述过程算出而存储起来. 本文算法与其区别可以由实际计算而得到比较.

把质点编号重新排列后, 短程力就可依网格顺序对每个网格内的质点分别计算了. 为要计算第 LL 个网格内所有质点所受的短程力, 除要计算此网格内距离小于 r_s 的两两质点间的相互作用力外, 还要计算与此网格相邻接的 13 个网格内的质点对它们的引力(这儿由于作用力反作用力相等, 可以避免计算邻接的另 13 个网格内质点的作用). 如前所述, 第 LL 个网格内的质点编号在 LL 给定以后可以很快地由 $NUR(LL)$ 与 $MM(LL)$ 算出, 这样一个过程可以大大减少为判别两质点间距离是否大于 r_s 所耗的机时.

在计算两距离小于 r_s 的质点间的作用力时, 我们如文献[3, 4]那样, 一方面采用软化引力势, 另一方面采用由预先给定的引力与距离的关系进行内插的方法. 这样需要另外两个数组 $RR(SS)$ 及 $RG(SS)$ 来表示不同距离处的引力值及其导数值. 而 SS 视分割的情形而定, 一般取 $SS = 2026$ 已经足够精确了.

这样, 在本文短程力计算中, 引入的数值分别为: $NTT(N)$, $MM(N_L \times N_L \times N_L)$, $NUR(N_L \times N_L \times N_L)$, $RR(SS)$ 及 $RG(SS)$. 其中 $NTT(N)$ 是这儿最大的一个数组, 其大小由质点数决定. 由于此处计算是继 PM 法计算后进行的, 而在 PM 法中有一些数组到此步骤已无用, 因此这一数值就可放在等价数组的内存中, 例如用来存储节点上质量的数组就是一个. 因此, 在这一短程力的计算中, 实际所需的工作单元是不大的.

三、实例计算

我们已经知道, PM 法的计算结果使自引力在 2—3 个网格距离内被抹平. PM 中短程力的引入正是为了克服这一缺陷, 提高分辨率而作的. 因此截止长度 r_s 一般也就是 2—3 个网格长度. 为了检验我们的算法比较节约机时, 必须与他人比较. 到目前为止, 国际上有这方面报道的也就是 Efstathiou 等人^[4]在 1985 年所作的结果, 因此我们尽量把所有的条件取得与他们一样. 下面的所有结果与文献[4]一样是在具有 4M 内存的 VAX-11/780 上进行的.

文献[4]对两种初始条件的宇宙大尺度结构进行了计算。第一个模型取质点数 4096, 初始分布为 Poisson 随机分布, 演化时间为从 $a = 1$ 至 $a = 5.9$ 。第二个模型取质点数 32768, 初始分布为 Zeldovich 形式^[5], 扰动振幅的初始值取为 $\left\langle \left(\frac{\delta\rho}{\rho} \right)^2 \right\rangle^{1/2} = 0.22$ 。第一个模型的计算在

PM 法上取 $32 \times 32 \times 32$ 网格, 而第二个模型的计算在 PM 法上取 $64 \times 64 \times 64$ 网格。若在两个模型的 PM 法上, 网络一维方向长度均为 32 的话, 则在第一个模型中 $r_s = 2.7$, 而在第二个模型中 $r_s = 1.35$ 。

表 1 给出了用我们上面给出的算法及文献[4]的算法对两个模型进行计算时短程力部分所耗的 CPU 时间对照。

表 1 两个模型不同算法短程力计算所需的 CPU 比较

CPU 时间 (s)	Poisson $N = 4096$	Zeldovich $N = 32768$	
	$r_s = 2.7$	$r_s = 1.35$	$r_s = 2.7$
本文算法	6.73($a = 1$)	59.33($a = 1$)	278.15($a = 1$)
	7.81($a = 5.9$)	130.66($a = 4.5$)	436.19($a = 4.5$)
Efstathiou 等 ^[4]	16.7($a = 1$)	99.8($a = 1$)	
	18.1($a = 5.9$)	782.2($a = 4.5$)	

从表 1 看出, 在两个模型下, 由本文算法计算短程力在相同截止长度下所耗的 CPU 要比 Efstathiou 等人的算法少得多。在其中的 Zeldovich 分布情形, $r_s = 1.35$ 是对 $32 \times 32 \times 32$ 网络来说的, 如对 $64 \times 64 \times 64$ 网络^[4], 即是 $r_s = 2.7$ 。上表充分体现了本文提出的算法的优越性。

我们知道, 上面给出的是计算一个时间步长短程力所需的 CPU, 而通常在宇宙大尺度结构数值模拟中一般要计算几百个时间步长, 因此, CPU 时间的多少是一个突出的问题。为了更加体现出改进算法的快速性, 表 2 给出了用直接 PP 算法来计算初始 Poisson 分布下短程力所需的时间。

表 2 Poisson 分布下利用直接 PP 法计算短程力的 CPU 时间 ($a = 1$)

质点数 N	4096		10000	
	1.35	2.7	1.35	2.7
截止长度 r_s				
CPU 时间 (s)	162.14	165.27	966.97	990.28

由此知道, 用直接 PP 法来计算短程力所需的机时实在太多了。这里有两点需要说明: 1. 在同样质点数下, r_s 不同但所耗机时却差得不多。这说明该方法中没有充分利用 r_s , 而主

要机时花在两两质点间的距离计算上; 2. 计算机时约与质点数的平方成正比。

前面我们已经说过, PM 法的主要缺点在于分辨率比较低, 短程力计算的引入可以提高其分辨率。图 2 给出了初始为 Poisson 分布, $N = 20000$ 演化时间为 $a = 9.9$ 时质点分布在一个方向上的投影。其中 (a) 为用 PM 法 $32 \times 32 \times 32$ 网格分割下的结果; (b) 为包括了 $r_s = 2.7$ 内的短程力下的结果。从两图的比较可知, 在包括了短程力后, 在相同的膨胀

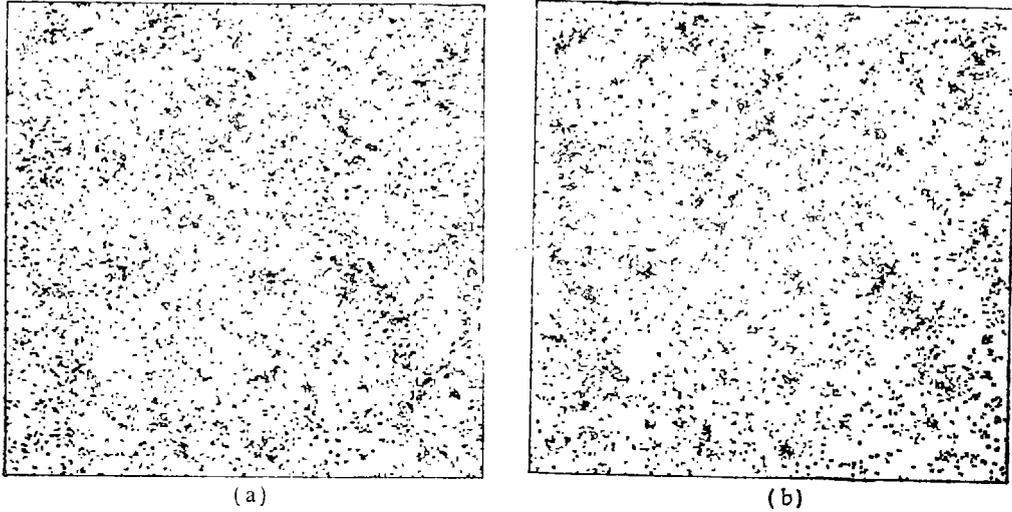


图 2 $N = 20000$, 初始 Poisson 分布, 演化到 $a = 9.9$ 时质点分布的二维投影图
((a) PM 法, (b) P³M 法)

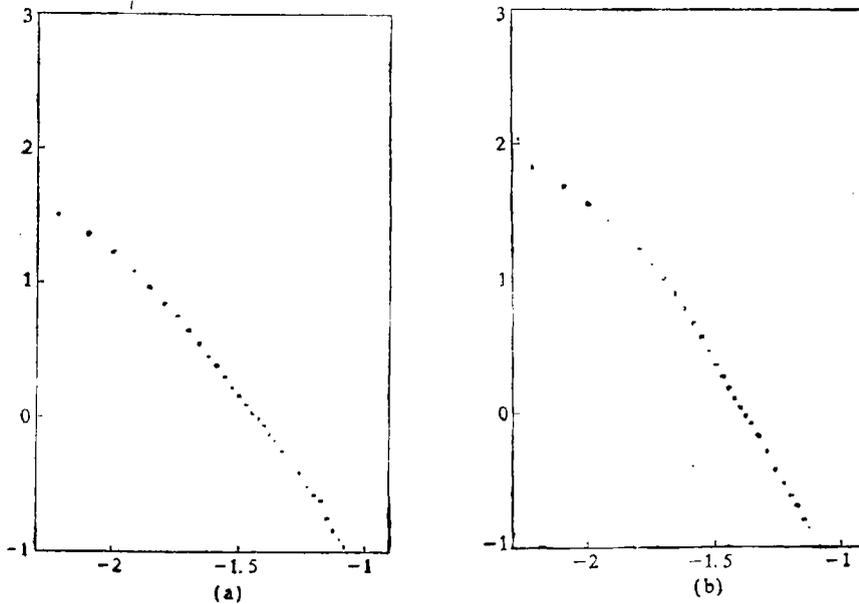


图 3 $N = 20000$, 初始 Poisson 分布, 演化到 $a = 9.9$ 时二点相关函数
((a) PM 法, (b) P³M 法)

因子下,质点的成团性要强得多.图3的(a)及(b)分别绘出了相应的两点相关函数,其差别主要在于小尺度范围内.图3(b)中的两点相关函数在较小的距离处要比图3(a)中的大.这主要是考虑了近距力的效应,提高了分辨率的缘故.

由本文所提出的改进短程力算法计算所得的质点演化的图形结构及两点相关函数可以与 Efstathiou 和 Eastwood^[6]的结果作一比较.从图形上看,在 Poisson 初始条件下,质点分布均显示出成团结构,而纤维状及空洞并不明显.更好的比较可通过两点相关函数.首先从两点相关函数的斜率来看,两文都接近于 -1.8 ,其次,关于其大小,本文在 $\lg \frac{r}{L} = -2.0$ 时,两点相关函数 $\lg \xi(r) = 1.8$,而文献[6]中图2显示,当 $\frac{x}{R_s} = 0.02$ 也即令本文的 $\lg \frac{r}{L} = -2.0$ 时,其 $\lg \xi(x)$ 也接近于本文的 1.8 .因此,两种结果是一致的.

四、讨 论

在牛顿力学框架内,应该说自引力的计算是简单直接的.两个质点间的引力势可以容易地写成 $\phi = GM/r$,其中 r 为两质点间的距离.可是在N体问题中,随着质点数的增加,相应的计算工作量却与质点数平方成正比.因此在数值计算中,突出的问题为极大的计算时间.以致N体问题的一个重要研究方向为如何用尽量少的计算机时来计算尽量多的质点间的自引力.在当前的计算机发展水平及算法条件下,在星系与宇宙大尺度结构的数值模拟中,最多的质点数也仅达到 10^9 量级.而对具有 $10^{14}M_{\odot}$ 的星系来说,每个质点的代表质量就在 10^6M_{\odot} 左右.这样大的点质量在某种程度上来说远不满足所研究的问题的需要.

在宇宙大尺度结构研究中,成团性的特征要求自引力的计算需充分反映其近距性.这样对近距力的计算就成了一个突出的问题.事实上,直接的PP法能够很好地把近距力包括进去.但正如表1和表2比较后可知,直接的PP法计算近距力太费机时了.

本文对近距力的新算法是对文献[4]的算法的改进.我们的目的也仅是如何减少计算所费的机时.计算的大框架还是在PM法上加进PP法.在这种P²M算法中,大量的机时还是耗费在PP法上.要对这一问题有所突破可能需要冲破这一大框架.当然在宇宙大尺度结构的数值模拟中,问题要比星系数值模拟多得多.另外一个重要问题是边界约束问题,这是由于我们所模拟的是宇宙中的一部分.这些问题的解决可能要与上面提出的大框架问题一起解决.

参 考 文 献

- [1] Aarseth, S. J., in *Methods of Computational Physics* (Eds. Brackbill, J. O. et al.), New York, Academic 1984, 1.
- [2] Barnes, J. & Hut, P., *Nature*, 324(1986), 446.
- [3] Hockney, R. W., Goel, S. P. & Eastwood, J. W., *J. Comput. Phys.*, 14(1974), 148.
- [4] Efstathiou, G. et al., *Ap. J. Supp. Series*, 57(1985), 241.
- [5] Zeldovich, Ya. B., *Astro. Ap.*, 5(1970), 84.
- [6] Efstathiou, G. & Eastwood, J. W., *MNRAS*, 194(1981), 503.