

同时发生的颗粒凝并和沉积现象 的 Monte Carlo 模拟*

赵海波 郑楚光

(华中科技大学煤燃烧国家重点实验室, 武汉 430074)

摘要 同时发生的颗粒凝并和沉积所导致的离散系统动力学演变过程数学上可由通用动力学方程描述. Monte Carlo 算法是求解通用动力学方程的一类重要方法. 然而, 常体积法由于模拟颗粒数目的波动而存在着计算代价和计算精度无法协调的矛盾, 常数目法由于计算区域的不断收缩或扩展而难以工程实际应用和科学定量分析, 且这些 Monte Carlo 算法均依赖于子系统概念而大大限制了其扩展性和应用范围. 发展一种多重 Monte Carlo 算法求解同时考虑凝并和沉积的通用动力学方程. 该算法引入加权虚拟颗粒的概念, 基于时间驱动, 在颗粒尺度分布时间演变过程中同步保持恒定的虚拟颗粒数目和稳定的计算区域体积, 这使得 MMC 算法具有考虑边界条件、颗粒尺度分布空间扩散、甚至颗粒-流体动力学的可扩展性. 利用多重 Monte Carlo 算法对几种特殊工况进行数值模拟, 结果与理论分析解符合很好, 表明该算法具备较高且稳定的计算精度和较低的计算代价, 这是由于虚拟颗粒数目稳定且较少的缘故. 最后分析了该算法的误差源以及相对误差.

关键词 多重 Monte Carlo 算法 通用动力学方程 凝并 沉积 颗粒尺度分布 离散系统

颗粒的凝并和沉积现象在自然界和工程领域普遍存在, 如雨雪雾冰的沉降过程、火电厂静电除尘器中飞灰颗粒的脱除过程、燃煤锅炉烟道内烟气的迁移过程、核电站安全分析中模拟核电站爆炸过程中气溶胶颗粒的扩散和沉积、化工中乳胶体和絮凝物的沉积、室内空气污染物的迁移变化、纳米材料的制备等等过程,

收稿日期: 2004-11-18; 接受日期: 2005-12-06

* 国家重点基础研究专项经费(批准号: 2002CB211602)和国家自然科学基金重点项目(批准号: 90410017)资助

SCIENCE IN CHINA Ser. E Technological Sciences

其中颗粒物的凝并和沉积是核心问题之一。两颗颗粒碰撞粘结在一起而形成一颗较大的颗粒, 以及某些颗粒的沉积均会使得颗粒尺度分布发生变化, 鉴于诸多颗粒的物化属性如毒性、荷电性、放射性、光散射性等均与颗粒尺度分布紧密相关, 描述颗粒尺度分布的时间演变过程是一项重要的研究任务。已有成熟的通用动力学方程(general dynamic equation, 简称 GDE)来描述由于凝并和沉积的发生而导致的颗粒尺度分布时间演变过程^[1]:

$$\left\{ \frac{dn_p(v,t)}{dt} \right\}_{\text{时间变化率}} = \left\{ -R(v)n_p(v,t) \right\}_{\text{沉积}} + \left\{ \frac{1}{2} \int_0^v \beta(v-u,u)n_p(v-u,t)n_p(u,t)du - n_p(v,t) \int_0^\infty \beta(v,u)n_p(u,t)du \right\}_{\text{凝并}}, \quad (1)$$

该方程基于颗粒稀疏和分子混沌假设, 只考虑二元凝并。其中, $n_p(v,t)$ 为颗粒尺度分布函数(量纲为 $m^{-3} \cdot m^{-3}$), 表示时刻 t , 体积为 v 的颗粒的颗粒数目浓度; 方程(1)左边表示体积为 v 的颗粒的尺度分布函数的时间演变; $R(v)$ 称为沉积核或沉积系数, 描述单位时间内体积为 v 的颗粒可能沉积的概率(量纲为 s^{-1}), 沉积核可以集中描述颗粒沉积的各种机理, 如重力沉降、Brown 扩散、湍流作用、热泳力作用、静电吸附、湿去除等; 方程(1)右边第一部分表示由于体积为 v 的颗粒的沉积对其尺度分布函数的时间变化率的贡献; $\beta(v,u)$ 是体积分别为 v 和 u 的两颗粒的凝并核(量纲为 $m^3 \cdot s^{-1}$), 表示单位时间内两颗颗粒发生一次凝并事件的概率, 颗粒凝并的各种机理如 Brown 扩散、湍流输运效应、局部富集效应、重力作用等都可集中在凝并核中; 方程(1)右边第二部分描述了凝并事件的发生而导致体积为 v 的颗粒的尺度分布函数随时间变化, 其中第一项为生成项, 表示体积为 $(v-u)$ 的颗粒和体积为 u 的颗粒凝并而生成体积为 v 的颗粒, 第二项为消亡项, 表示体积为 v 的颗粒与其他颗粒发生凝并产生更大颗粒而使得自身消亡。需要说明的是, 对于不同粒径的颗粒的凝并核和沉积核在不同时刻可能发生变化。

通用动力学方程(1)是一个典型的非线性局部积分微分方程, 不仅仅一般不存在理论分析解, 而且对于多分散性的颗粒群和非线性的凝并和沉积机理而言, 普通的数值方法如有限差分法、有限体积法等都难以对其求解。目前, 主要 GDE 数值求解方法有矩方法^[2]、分区法^[3]、离散法^[4]、离散分区法^[5]和 Monte Carlo (MC)方法^[6-9]等。MC 法具备与 GDE 同样的物理基础和物理假设^[10], 它天生的离散特性与离散系统内离散的动力学事件不谋而合。MC 方法的优点在于能得到颗粒的轨道经历效应和历史效应, 可以方便处理多组分、多分散性颗粒群, 对于复杂的颗粒演变如结构重建、包覆、不规则形状颗粒甚至沉积之后的重新携带等均能处理, 算法相对较简单, 易于编程实现^[6], 因此得到了广泛的发展; 但是, MC 法本身计算代价过大, 而且凝并和沉积事件使得实际颗粒总数目不断减少, 对于

普通的基于常计算区域体积的 MC^[7]而言, 其模拟颗粒总数目 N 也会随之不断减少. 由于这种 MC 算法的统计精度与 \sqrt{N} 成反比, 所以普通 MC 算法为了保证相当的统计精度而不得不增多初始模拟颗粒的数目, 这进一步导致了计算代价的增大, 因此存在无法协调计算精度和计算代价的矛盾. Lin 等人^[8]发展了一种常数目法, 在计算过程中保持模拟颗粒数目不变, 从而保证了 MC 算法稳定的计算精度. 但是, 该算法以不断调整计算区域的体积为代价来保证计算区域内恒定的模拟颗粒数目, 不便于具体的工程应用和科学定量分析, 难以考虑颗粒尺度分布的空间扩散、边界条件和流体-颗粒场的动力学演变等.

本文发展一种全新的多重 Monte Carlo (multi-Monte Carlo, 简称 MMC) 算法来求解同时考虑凝并和沉积的通用动力学方程, 同时具备常体积法和常数目法的特点, 能够克服计算代价和计算精度的矛盾, 并具有考虑颗粒尺度分布的空间演变、边界条件甚至流体-颗粒场的动力学演变等扩展性.

1 多重 Monte Carlo 算法描述

1.1 加权虚拟颗粒的引入

普通尺度的计算区域内实际颗粒数目一般可以达到 10^{10} 以上. 由于目前计算机计算能力的限制, Monte Carlo 算法均只能模拟 $10^3 \sim 10^7$ 个模拟颗粒的时空演变, 所以目前所有求解 GDE 的 Monte Carlo 算法^[6-9]均引入“子系统”的概念. 子系统内包含 $10^3 \sim 10^7$ 个模拟颗粒, 每颗模拟颗粒均代表一定数目的实际颗粒, 即每颗模拟颗粒均有一定的数目权值, 认为整体系统是充分混合搅拌而空间各向同性, 子系统具备周期性边界条件, 子系统是整个系统的指示器. 常体积 MC^[7]中模拟颗粒的数目权值相同且始终不变. 在常数目 MC^[8]中, 如果由于破碎或成核等事件使得整体系统内颗粒数目浓度增加, 则缩小子系统的体积以保持其内模拟颗粒总数目不变, 如果由于凝并和沉积等事件使得整体系统内颗粒数目浓度减小, 则扩大子系统的体积使其内模拟颗粒数目不变. 实际上, 常数目 MC 中所有模拟颗粒的数目权值均随时间发生改变.

在 MMC 算法中, 为了在模拟过程中保持计算区域体积不变以及计算区域内模拟颗粒数目不变, 不同于以上思想, 引入加权虚拟颗粒的概念. 认为一组体积相同或者相近的实际颗粒具备同样的属性和行为, 这些实际颗粒由一颗或几颗虚拟颗粒代表, 虚拟颗粒是这些实际颗粒的指示器, 虚拟颗粒的演变过程可以代表计算区域内实际颗粒的演变过程.

生成虚拟颗粒的步骤如图 1 所示. 可见, 每类实际颗粒可以由若干个虚拟颗粒代表, 虚拟颗粒的体积就是所代表的实际颗粒的体积, 不同类的虚拟颗粒具有不同的初始数目权值 w , 但是同类的虚拟颗粒具有相同的初始 w , w 的值就是这颗

虚拟颗粒所代表的当地实际颗粒的个数. 一般来说, w 的值达到 $O(10^3) \sim O(10^4)$ 量级仍然可以保持 MMC 算法较高的计算精度.

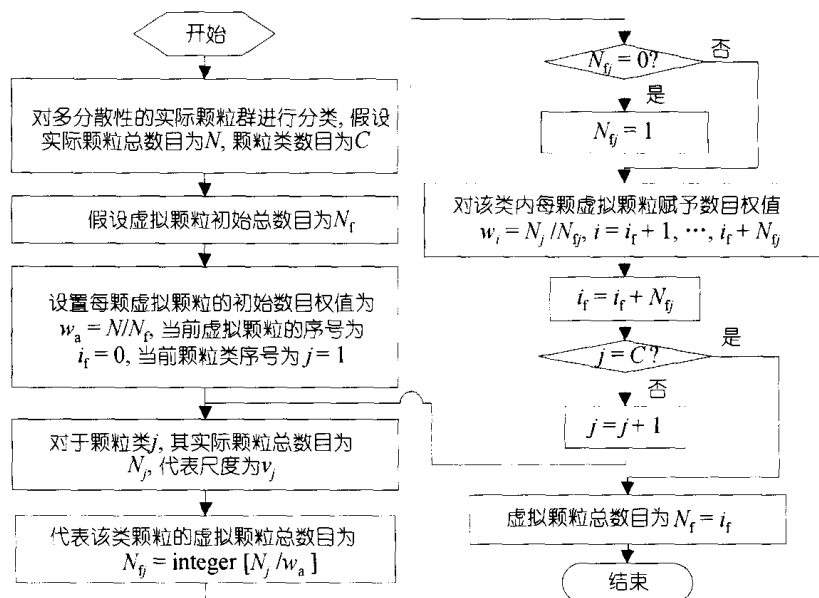


图1 虚拟颗粒数目权值 w 的设置步骤

在系统演变过程中, 按照各种事件的不同后果, 在每个时间步长之后改变对应的虚拟颗粒的数目权值和体积, 不仅保持计算区域总体积不变, 而且保持虚拟颗粒总数目不变, 下文将详述. 实际上, 在 MMC 算法中, 随着时间的演变, 只有发生了凝并和沉积等事件的相关虚拟颗粒的数目权值发生变化.

对于颗粒数目很多的整体系统, 虽然引入了加权虚拟颗粒的概念, 模拟颗粒的数目可能仍然具有很大的量级. 所以此时 MMC 算法仍然需要继续沿用子系统的假设. 如果整体系统内颗粒数目较少, 或者对计算精度要求不高而允许一颗虚拟颗粒代表更多的实际颗粒, 那么此时整体系统内虚拟颗粒数目可能在计算机的计算能力允许范围之内. 因此 MMC 算法可以抛弃子系统的假设, 从而可以考虑颗粒尺度分布的空间演变过程、边界条件、甚至与两相流模型耦合起来考虑气固两相流的动力学演变过程^[11].

1.2 同时考虑凝并和沉积的多重 Monte Carlo 算法计算流程

多重 Monte Carlo 算法设计为时间驱动(time-driven)的 Monte Carlo 方法^[9], 也就是说, 在设定的时间步长范围内, 考虑每一颗颗粒可能的各种事件. 在基于时间驱动的算法中, 存在如下假设: 在足够小的时间步长 Δt 内, 同一颗颗粒发生的凝并事件与沉积事件可以互相解耦, 不同颗粒之间发生的凝并事件和沉积事件

也可以互相解耦, 这样可以分别单独处理每一个凝并事件和沉积事件, 不管同一颗颗粒是否同时发生凝并事件和沉积事件, 也不管同一颗颗粒是否同时参与多个凝并事件或沉积事件. 同时考虑凝并和沉积的多重 Monte Carlo 算法的计算流程如图 2 所示. 需要注意的是, 在一个时间步长内, 颗粒即使发生了凝并和沉积事件, 也认为其不会在当前时间步长内影响这颗颗粒和相关颗粒的行为和属性, 而只会在下一个时间步长内影响这些颗粒的行为和属性, 所以直到当前时间步长内所有虚拟颗粒循环全部结束之后, 再进行凝并和沉积的后果处理, 参照图 2.

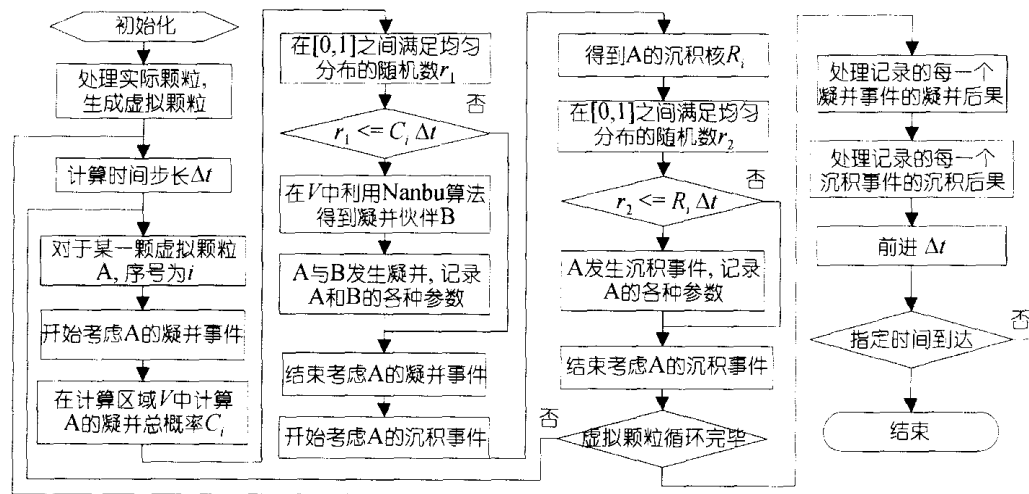


图 2 同时考虑凝并和沉积的多重 Monte Carlo 算法的计算方案

由于需要明确的时间窗口间隔, 相比于另外一种相对应的基于事件驱动的 Monte Carlo 方法^[7], 一般情况下该类 MC 法计算代价要更高一些; 但是, 目前发展非常成熟的两相湍流模型的数值解法基本上都是基于时间驱动的思想, 所以基于时间驱动的 MC 方法要比基于事件驱动的 MC 方法更容易与求解两相湍流模型的数值解法耦合, 所以其扩展性要更好一些.

1.3 时间步长的设置

时间步长的设置是 MMC 算法的关键之一. 对于基于时间驱动的 MC 而言, 为了不遗漏每一个颗粒可能的凝并事件和沉积事件, 需要保证在一个时间步长内, 任何虚拟颗粒都最多只可能发生一次凝并事件或沉积事件.

首先考察虚拟颗粒 i 的凝并概率. 在某个计算区域 V 内, 假设实际颗粒个数为 N , 虚拟颗粒个数为 N_v , 则单位时间内, 第 i 颗虚拟颗粒所代表的实际颗粒与不同于 i 的其他虚拟颗粒所代表的实际颗粒之间发生凝并事件的次数为

$$NC_{ij} = \beta_{ij} \times w_i \times w_j, \quad (2)$$

其中, β_{ij} 为颗粒 i 和 j 的凝并核. 单位时间内第 i 颗虚拟颗粒所代表的实际颗粒内部发生的凝并事件的次数为^[12]

$$NC_{ii} = \beta_{ii} \times w_i \times (w_i - 1) / 2. \quad (3)$$

所以单位时间内虚拟颗粒 i 代表的所有实际颗粒发生的凝并事件的总次数为

$$NC_i = NC_{ii} + \sum_{j=1, j \neq i}^{N_f} NC_{ij} = \beta_{ii} \times w_i \times \frac{(w_i - 1)}{2} + \sum_{j=1, j \neq i}^{N_f} (\beta_{ij} \times w_i \times w_j). \quad (4)$$

由于虚拟颗粒 i 代表了 w_i 个实际颗粒, 所以单位时间内每颗实际颗粒平均发生的凝并事件的次数为

$$C_i = \frac{NC_i}{w_i} = \frac{\beta_{ii} \times (w_i - 1)}{2} + \sum_{j=1, i \neq j}^{N_f} (\beta_{ij} \times w_j). \quad (5)$$

由于虚拟颗粒 i 是它所代表的那些实际颗粒的指示器, 虚拟颗粒 i 在单位时间内发生一次凝并事件(包括不同的虚拟颗粒间发生的凝并事件和同一颗虚拟颗粒内部发生的凝并事件)的次数就是 C_i , 也可称为虚拟颗粒 i 发生凝并事件的总概率. 所以虚拟颗粒 i 只发生一次凝并事件的时间尺度为

$$t_{i, \text{coag}} = 1 / C_i. \quad (6)$$

依据沉积核的定义, 虚拟颗粒 i 只发生一次沉积事件的时间尺度为

$$t_{i, \text{depo}} = 1 / R_i, \quad (7)$$

其中, R_i 为虚拟颗粒 i 的沉积核.

时间步长应该至少均小于等于最小的凝并时间尺度($t_{i, \text{coag}}$), 也必须至少小于等于最小的沉积时间尺度($t_{i, \text{depo}}$), 也就是

$$\Delta t = \min \left\{ 1 / \max_{i=1, \dots, N_f} (C_i), 1 / \max_{i=1, \dots, N_f} (R_i) \right\}, \quad (8)$$

一般来说为了增加 Monte Carlo 循环的次数以增加计算精度, 时间步长如下设置:

$$\Delta t = \alpha \times \min \left\{ 1 / \max_{i=1, \dots, N_f} (C_i), 1 / \max_{i=1, \dots, N_f} (R_i) \right\}, \quad (9)$$

这里, α 为乘积因子, 一般 $\alpha = 0.01$.

值得注意的是, 随着凝并和沉积事件的进行, 颗粒尺度分布会不断变化, 这会使得凝并时间尺度和沉积时间尺度发生改变, 故在进行每一次虚拟颗粒整体循环之前, 都需要重新计算时间步长, 参照图 2.

1.4 凝并事件的处理

对凝并事件的处理包括当前跟踪的虚拟颗粒是否发生凝并事件的判断、凝并伙伴的选择、该次凝并事件的后果处理.

1.4.1 凝并是否发生的判断

认为颗粒凝并所导致的动力学演变过程为 Markov 过程^[7], 此时, 对于虚拟颗粒 i , 其在 Δt 内发生凝并事件的概率 $Pr_{\text{coag},i}(\Delta t)$ 满足指数分布^[9], 即 $Pr_{\text{coag},i}(\Delta t) = 1 - \exp(-C_i\Delta t) \approx C_i\Delta t$. 这里, 利用 Nanbu 算法^[13]来考虑颗粒是否凝并以及寻找其凝并伙伴, 如图 3 所示. 首先生成满足均匀分布, 位于 $[0,1]$ 的随机数 r_1 , 则在某个时间步长 Δt 内, 对于当前跟踪的虚拟颗粒 i , 其发生凝并事件的总概率为乘积 $(C_i\Delta t)$, 如果随机数 r_1 小于等于 $(C_i\Delta t)$, 那么虚拟颗粒 i 将确定要发生一次凝并事件.

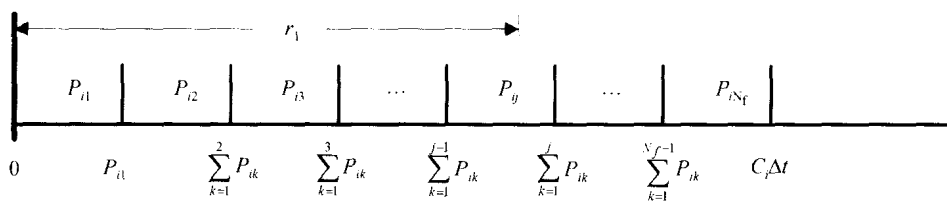


图 3 Nanbu 算法示意图

1.4.2 凝并伙伴的选择

如果确定虚拟颗粒 i 将发生凝并事件, 则继续利用 Nanbu 算法^[13]来寻找凝并伙伴. 首先计算当前 Δt 内, 跟踪颗粒 i 与不同于 i 的某颗虚拟颗粒 j 发生凝并的概率 $P_{ij} = w_j \times \beta_{ij} \times \Delta t$, 当前 Δt 内跟踪颗粒 i 内部发生凝并的概率为 $P_{ii} = [(w_i - 1)/2] \times \beta_{ii} \times \Delta t$. 可见,

$$C_i\Delta t = \left[\sum_{j=1, i \neq j}^{N_f} (\beta_{ij} \times w_j) + \frac{\beta_{ii} \times (w_i - 1)}{2} \right] \Delta t = \sum_{k=1}^{N_f} P_{ik}. \quad (10)$$

然后仍然通过随机数 r_1 来判断与虚拟颗粒 i 发生凝并的伙伴颗粒 j , 如果以下关系得到满足, 则认为虚拟颗粒 j 为 i 的凝并伙伴:

$$\sum_{k=1}^{j-1} P_{ik} < r_1 \leq \sum_{k=1}^j P_{ik}, \quad j \in [1, N_f]. \quad (11)$$

1.4.3 凝并事件的后果处理

对凝并后果的处理的一项重要内容是保持虚拟颗粒数目恒定以保持稳定的计算精度, 对凝并事件后果的处理集中体现了 MMC 算法常体积和常数目的思想, MMC 算法通过调整发生凝并的虚拟颗粒的数目权值 w 来实现.

当前跟踪的虚拟颗粒 i (数目权值为 w_i , 体积为 v_i) 与它的凝并伙伴 j (数目权值为 w_j , 体积为 v_j) 之间发生的二元凝并事件, 相当于 i 所代表的某些实际颗粒与 j

代表的某些实际颗粒发生了二元凝并事件. 由于两颗实际颗粒 A 和 B(体积分别为 v_A 和 v_B)的凝并结果是一颗新的体积为(v_A+v_B)的实际颗粒, 所以不管虚拟颗粒 i 和 j 凝并过程中实际颗粒间具体的凝并配对过程, 总的凝并后果是 i 和 j 所代表的旧的实际颗粒被新的体积为(v_i+v_j)的实际颗粒所取代, 并且新的实际颗粒数目较原有实际颗粒数目总和减少了一半, 即(w_i+w_j)/2. 为了保持虚拟颗粒总数目, 并不舍弃虚拟颗粒 A 和 B 中的任何一颗, 而是通过调整两颗虚拟颗粒的数目权值和体积来体现凝并后果. 可以认为虚拟颗粒 i 和 j 的数目权值均各自减少了一半, 并且其体积均变为(v_i+v_j). 由于每颗虚拟颗粒均需要判断其是否凝并和寻找其凝并伙伴, 所以同一凝并事件将被检测到两次. 如果当前跟踪的虚拟颗粒为 i , 则只处理虚拟颗粒 i 以体现凝并事件的后果, 而不对当前的凝并伙伴 j 作任何改变, 也就是

$$\begin{aligned} (w_i)_{\text{new}} &= (w_i)_{\text{old}}/2, & (v_i)_{\text{new}} &= (v_i)_{\text{old}} + (v_j)_{\text{old}}, \\ (w_j)_{\text{new}} &= (w_j)_{\text{old}}, & (v_j)_{\text{new}} &= (v_j)_{\text{old}}. \end{aligned} \quad (12)$$

如果当前跟踪颗粒轮到虚拟颗粒 j 的时候, 理论上会检测到同一凝并事件, 且其凝并伙伴为虚拟颗粒 i , 同样仅仅改变当前跟踪的虚拟颗粒 j 的数目权值和体积以体现凝并事件的后果, 也就是

$$\begin{aligned} (w_j)_{\text{new}} &= (w_j)_{\text{old}}/2, & (v_j)_{\text{new}} &= (v_i)_{\text{old}} + (v_j)_{\text{old}}, \\ (w_i)_{\text{new}} &= (w_i)_{\text{old}}, & (v_i)_{\text{new}} &= (v_i)_{\text{old}}. \end{aligned} \quad (13)$$

这样不仅保持了整体虚拟颗粒数目的不变, 而且不需要调整计算区域的体积.

严格来说, 只有 $(w_i)_{\text{old}}=(w_j)_{\text{old}}$ 或者 $(v_i)_{\text{old}}=(v_j)_{\text{old}}$ 时, 以上对凝并事件后果的处理方案才能保证凝并前后颗粒总体积不变, 即

$$\begin{aligned} & (v_i)_{\text{new}} \times (w_i)_{\text{new}} + (v_j)_{\text{new}} \times (w_j)_{\text{new}} \\ &= \left[(v_i)_{\text{old}} + (v_j)_{\text{old}} \right] \times (w_i)_{\text{old}}/2 + \left[(v_i)_{\text{old}} + (v_j)_{\text{old}} \right] \times (w_j)_{\text{old}}/2 \\ &= \left[(v_i)_{\text{old}} + (v_j)_{\text{old}} \right] \times \left[(w_i)_{\text{old}} + (w_j)_{\text{old}} \right] / 2 \\ &= \begin{cases} (v_i)_{\text{old}} \times \left[(w_i)_{\text{old}} + (w_j)_{\text{old}} \right] = (v_i)_{\text{old}} \times (w_i)_{\text{old}} + (v_j)_{\text{old}} \times (w_j)_{\text{old}}, & \text{若 } (v_i)_{\text{old}} = (v_j)_{\text{old}}, \\ \left[(v_i)_{\text{old}} + (v_j)_{\text{old}} \right] \times (w_i)_{\text{old}} = (v_i)_{\text{old}} \times (w_i)_{\text{old}} + (v_j)_{\text{old}} \times (w_j)_{\text{old}}, & \text{若 } (w_i)_{\text{old}} = (w_j)_{\text{old}}. \end{cases} \end{aligned}$$

显然绝大多数凝并工况不可能满足这个条件, 所以 MMC 算法的这种处理方式存在一定的数值误差, 这是为了同时保持常体积和常数目所必须付出的代价. 如果参与凝并事件的两虚拟颗粒的数目权值和尺度之间的差异较小, 通过多次 MC 循环并取平均值的办法可以有效抑制这种误差的传播和繁殖.

1.5 沉积事件的处理

MMC 算法对沉积事件的处理包括颗粒是否沉积的判断以及沉积后果的处理.

1.5.1 沉积事件是否发生的判断

同样认为颗粒沉积事件的发生为 Markov 过程, 虚拟颗粒 i 在 Δt 内发生沉积事件的概率 $Pr_{\text{depo},i}(\Delta t) = 1 - \exp(-R_i \Delta t) \approx R_i \Delta t$, 所以, 在一个标准的 Monte Carlo 模拟过程中, 确定虚拟颗粒 i 发生沉积事件的判断表达式如下:

$$r_2 < R_i \Delta t. \quad (14)$$

否则, 认为虚拟颗粒 i 不会发生沉积事件, 将在下一个时间步长内继续跟踪它. 其中, r_2 为满足均匀分布, 位于 $[0,1]$ 的随机数.

1.5.2 沉积事件的后果处理

颗粒沉积的直接后果是实际颗粒总数目的减少, 如果不进行处理, 虚拟颗粒数目也会相应减少. 假设虚拟颗粒 A 确定将要沉积下来, 则首先通过一个随机过程从虚拟颗粒数列中随机选取一颗虚拟颗粒 B, 其序号为 $\text{integer}[r_3 \times N_i]$, $\text{integer}[]$ 表示取整, r_3 为满足均匀分布, 位于 $[0,1]$ 的随机数. 如果选择的虚拟颗粒就是已经判断出来将要沉积的虚拟颗粒 A, 则需要重新选择. 此时, 把虚拟颗粒 B (数目权值为 w_B , 体积为 v_B) 分裂成两颗虚拟颗粒 C (数目权值为 w_C , 体积为 v_C) 和 D (数目权值为 w_D , 体积为 v_D), $w_B = 2w_C = 2w_D$, 且 $v_B = v_C = v_D$; 把虚拟颗粒 C 和 D 分别取代虚拟颗粒 A 和 B. 这些处理手段实际上相当于用一个随机选择的虚拟颗粒的一半来填补确定沉积的虚拟颗粒留下的空白, 这样的处理手段保持了虚拟颗粒总数目不变, 也符合颗粒沉积实际过程.

2 同时考虑凝并和沉积的多重 Monte Carlo 算法的验证

对于某些特殊的凝并核、沉积核以及满足特殊函数形式的初始颗粒尺度分布, 颗粒尺度分布的时间演变存在理论分析解, 本文利用理论分析解来验证同时考虑凝并和沉积的 MMC 算法.

假设颗粒初始尺度分布满足指数分布, 即

$$n_p(v, 0) = (N_0/v_{g0}) e^{-v/v_{g0}}, \quad (15)$$

其中, N_0 和 v_{g0} 分别为初始颗粒总数目浓度和初始颗粒几何平均体积, 选择 $N_0 = 10^6 \text{ cm}^{-3}$, $v_{g0} = 0.027 \mu\text{m}^3$. 两种存在理论分析解的特殊工况为

工况 1: 常凝并核和常沉积核, $\beta(u, v) = K_C$, $R(v) = K_D$,

工况 2: 常凝并核和重力沉积核, $\beta(u, v) = K_C$, $R(v) = K_D v^{2/3}$,

其中, K_C 和 K_D 均为常数, 常凝并核可近似描述连续区 Brown 凝并过程, 常沉积核

描述了密封容器内颗粒的泄漏过程, 重力沉积核描述了重力作用下颗粒的沉积. 两种特殊工况的计算条件如表 1 所示, 理论分析解参照文献[14]. 其中, τ_{coag} 为凝并特征时间尺度, τ_{depo} 为沉积特征时间尺度, 两时间尺度的比例 $\Omega = \tau_{\text{coag}}/\tau_{\text{depo}}$. 为了使凝并和沉积事件均将对颗粒尺度分布的演变起同等程度的主要作用, 对于每种工况均选择合适的 K_C 和 K_B , 使得 $\Omega \approx 1$.

表 1 两种特殊工况的计算条件

工况	$K_C/\text{cm}^3 \cdot \text{s}^{-1}$	K_B	$\tau_{\text{coag}}/\text{s}$	$\tau_{\text{depo}}/\text{s}$	$\tau_{\text{coag}}/\text{s}$	$\tau_{\text{depo}}/\text{s}$
1	6.405×10^{-10}	$3.2025 \times 10^{-4} \text{ s}^{-1}$	$2/(K_0 N_0)$	$1/K_B$	≈ 3122.6	≈ 3122.6
2	6.405×10^{-10}	$3.3928 \times 10^{-3} \text{ s}^{-1} \cdot \mu\text{m}^{-2}$	$2/(K_0 N_0)$	$1/(K_B v_{g0}^{2/3})$	≈ 3122.6	≈ 3122.6

两种工况中, 初始虚拟颗粒数目取为 3000, 实际虚拟颗粒数目为 3081, 平均每颗虚拟颗粒代表约 325 颗实际颗粒, 指定的演变时间长度均为 $\tau_{\text{coag}}/2$, 即 1561.3 s. 为了进行颗粒尺度分布函数的初始化和统计得到指定时刻的颗粒尺度分布信息, 对实际颗粒群离散为 200 类颗粒, 虽然如此, 实际上 MC 算法在模拟过程中并不需要任何颗粒分类信息, 这可减少数值误差及数值扩散. 另外, 颗粒尺度分布的一阶矩定义为 $M_1(t) = \int n_p(v, t) v dv$, 代表颗粒总体积.

图 4 和 5 分别为 MMC 算法对工况 1 和 2 的数值模拟结果与相对应的理论分析解的比较, 其中, 图 4(a)和 5(a)为工况 1 和 2 中颗粒总数目、几何平均体积、颗粒尺度分布的一阶矩的时间演变过程, 图 4(b)和 5(b)分别为工况 1 和 2 中, 在 $t = 750$ 和 1500 s 两个特定时刻处, 颗粒尺度分布曲线的 MMC 数值模拟结果与对应的理论分析值的比较. 对于工况 1, 在 $t = 750$ 和 1500 s 处, 实际颗粒数目仅仅分别为初始颗粒数目的 65.43%和 44.57%; 对于工况 2, 在 $t = 750$ 和 1500 s 处, 实际颗粒数目仅仅分别为初始颗粒数目的 66.62%和 47.10%. 但是虚拟颗粒数目仍然能够稳定在 3081, MMC 算法的各种数值模拟结果与对应的理论分析解均符合非常好, 这表明该算法具备非常高且稳定的计算精度.

值得注意的是, MMC 算法的计算代价非常低, 对于工况 1 和 2, 其所消耗的 CPU 时间分别为 1564.03 和 350.55 s (计算环境为 Athlon Xp2500+, 512M), 这主要是因为 MMC 算法跟踪比实际颗粒数目小得多的虚拟颗粒所致. MMC 算法的计算代价主要消耗在时间步长的计算上, 即使对于工程和自然界中非常复杂的凝并核和沉积核, 计算代价也不会有明显的增加.

3 讨论

3.1 同时考虑凝并和沉积的 MMC 算法的误差分析

由于无法把 MMC 算法的每个子过程单独剥离开来, 所以很难精确定量地分析每一个子过程对其计算精度的影响, 但是仍然可以分析出同时考虑凝并和沉

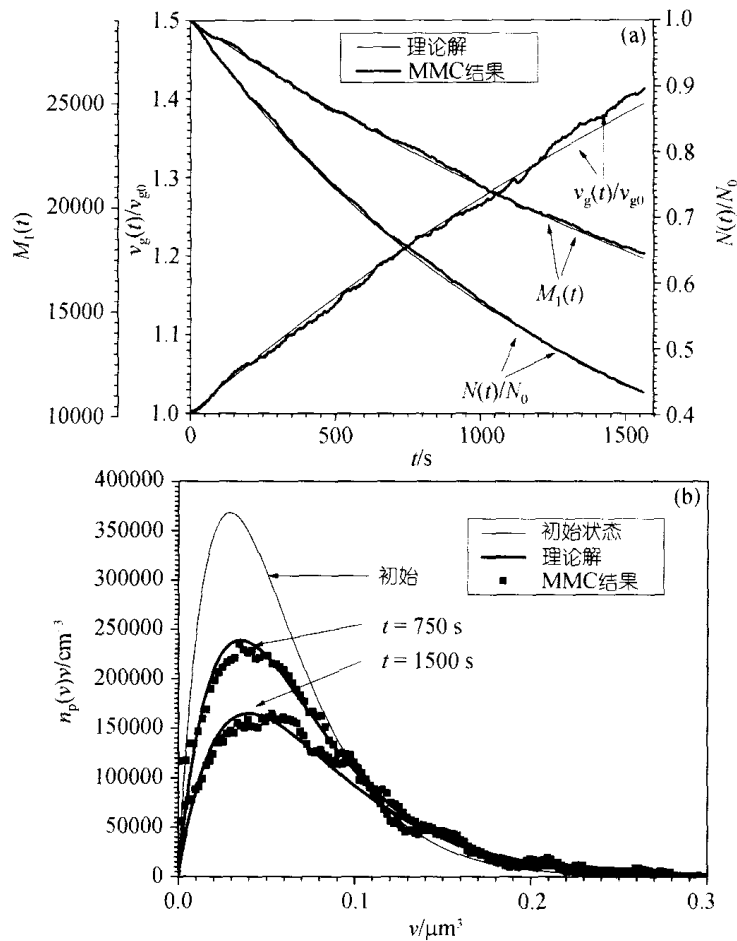


图 4 常凝并核和常沉积核的颗粒尺度分布时间演变

积的 MMC 算法的误差源. MMC 算法基于时间驱动技术, 假设足够小的 Δt 内离散系统的动力学事件(凝并和沉积等)之间互相解耦. 实际上在一个 Δt 内, 可能同时发生若干个凝并事件或/和沉积事件, 同一颗颗粒可能同时发生凝并事件和沉积事件, 同一颗颗粒也可能同时参与几个凝并事件, 这些事件并不完全相互独立而不能互相解耦, 所以基于时间驱动的 MC 算法必然存在这种所谓的“解耦误差”. 通过减小乘积因子 α (参见公式(9)), 可以有效减小 Δt , 从而减少该 Δt 内发生的动力学事件数量, 进而减小解耦误差, 但是计算代价将随之增加. 另外一个误差来源于动力学事件的后果处理过程. 在对颗粒凝并事件的后果进行处理时, MMC 算法采取的措施是把参与凝并的两颗虚拟颗粒的数目权值各自减半, 且体积均变为两颗虚拟颗粒凝并之前的体积之和. 这些措施不一定完全符合实际的凝并过程, 但是可以强制保持计算区域体积和计算区域内虚拟颗粒数目不变, 不妨称这种误差为所谓的“常体积和常数目误差”, 采用多次 MC 循环取平均值的办法可以降低这种误差. 对颗粒沉积事件的后果进行处理时也存在“常体积和常数目误

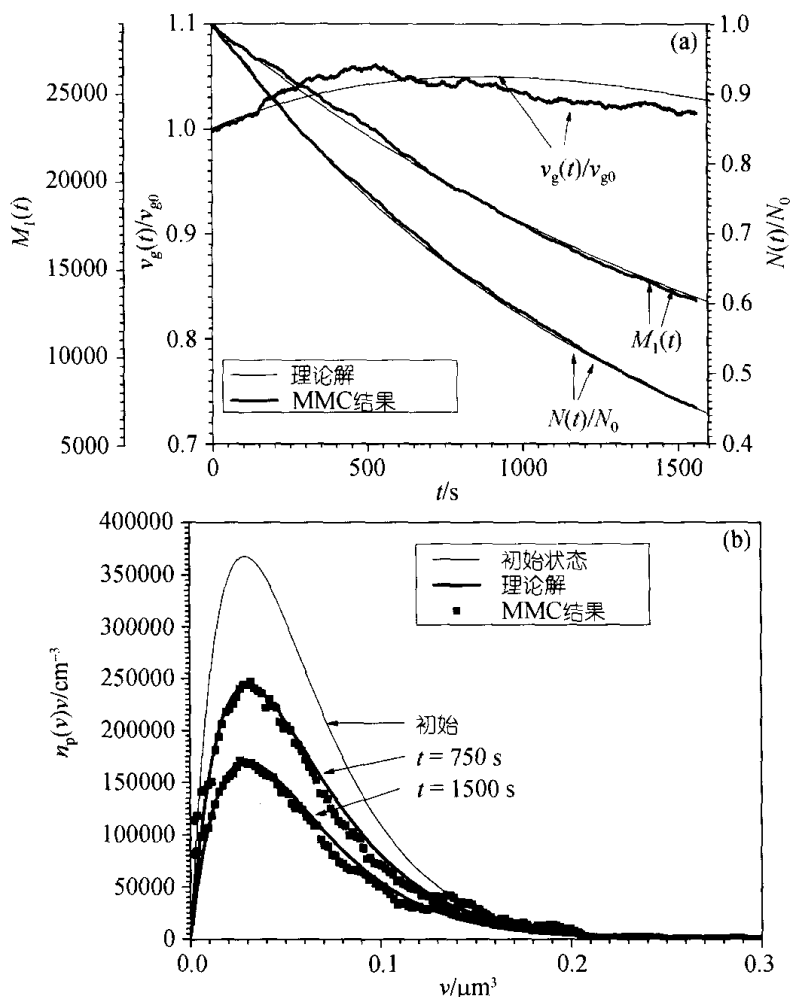


图5 常凝并核和重力沉积核的颗粒尺度分布时间演变

差”, 此时它直接关联于强制保持常数目的和常体积过程中对虚拟颗粒数目权值的扰动. 第三个误差是随机误差. MMC 算法采用了大量的随机过程, 如判断虚拟颗粒是否发生凝并事件和沉积事件时采用的随机数等, 这些随机过程必然存在一定量级的误差, 可称之为随机误差. 最后, 所有 MC 算法均存在统计误差, 其误差大小与模拟颗粒数目的 $1/2$ 次方成反比. 对于颗粒总数目 N , 平均体积 v_g , 颗粒总体积 M_1 这些宏观量或整体量而言, 只要模拟颗粒数目达到 $O(10^3)$ 以上, 该误差对其的计算精度的影响并不大; 而对于特定时刻的颗粒尺度分布这种瞬时量或细节量而言, 如果是颗粒尺度谱范围较广的多分散性颗粒群, 模拟颗粒数目对颗粒尺度分布的精度影响较大. 因为对于离散系统而言, 必须保证每个颗粒类内模拟颗粒数目达到一定量级(如 $O(10^2) \sim O(10^3)$ 以上), 才可能保证颗粒尺度分布的统计精度, 通过增加虚拟颗粒数目可以减少这种误差.

以工况 1 为例来分析同时考虑凝并和沉积的 MMC 算法的相对误差, 选择初

始时刻颗粒数目浓度与演变过程中颗粒数目浓度的比值 $N_0/N(t)$ 作为参考, $N_0/N(t)$ 的相对误差如下定义:

$$\delta = \left| \frac{[N_0/N(t)]_{\text{MMC}} - [N_0/N(t)]_{\text{理论}}}{[N_0/N(t)]_{\text{理论}}} \right| = \left| \frac{[N(t)]_{\text{理论}}}{[N(t)]_{\text{MMC}}} - 1 \right|. \quad (16)$$

Liffman^[9]分析了其发展的直接模拟 Monte Carlo 算法(标记为 Liffman 的 DSMC)中 $N_0/N(t)$ 的相对误差, 近似的拟合公式为 $\delta < 2/\sqrt{N_c(t)}$, 其中 $N_c(t)$ 为随时间变化的子系统内模拟颗粒数目, 由于 Liffman 的 DSMC 采用阶梯式倍增子系统的办法, $N_c(t)$ 总是在 $N_{c0}/2 \sim N_{c0}$ 之中变化, 这里 N_{c0} 是初始设置的模拟颗粒数目, 所以 $\delta < 2\sqrt{2}/\sqrt{N_{c0}}$. Smith 等人^[15]分析了其发展的常数目法中 $N_0/N(t)$ 的相对误差, 近似的拟合公式为 $\delta = \{\ln[N_0/N(t)]\}^{0.8}/\sqrt{2N_c}$, 常数目法中子系统内模拟颗粒数目 N_c 始终保持不变. 图 6 为 δ 随 $N_0/N(t)$ 变化的曲线图, 显然, MMC 算法具有相当高的计算精度, 相对误差基本限制在 $\pm 1.5\%$ 范围之内. MMC 算法的精度整体上要高于 Liffman 的 DSMC, 且与常数目法基本上在同一级别. 类似于常数目法, 可以认为 MMC 算法的相对误差 $\delta = \{\ln[N_0/N(t)]\}^{0.8}/\sqrt{2N_c}$.

一般来说, 随着离散系统动力学演变过程往前推进, MMC 算法的计算精度将越来越差, 当 MMC 算法的相对误差大于 10% 时, 认为 MMC 算法不适合继续准确地描述离散系统动力学演变过程, 所以 MMC 算法的收敛或终止判据是 $\delta = \{\ln[N_0/N(t)]\}^{0.8}/\sqrt{2N_c} < 10\%$.

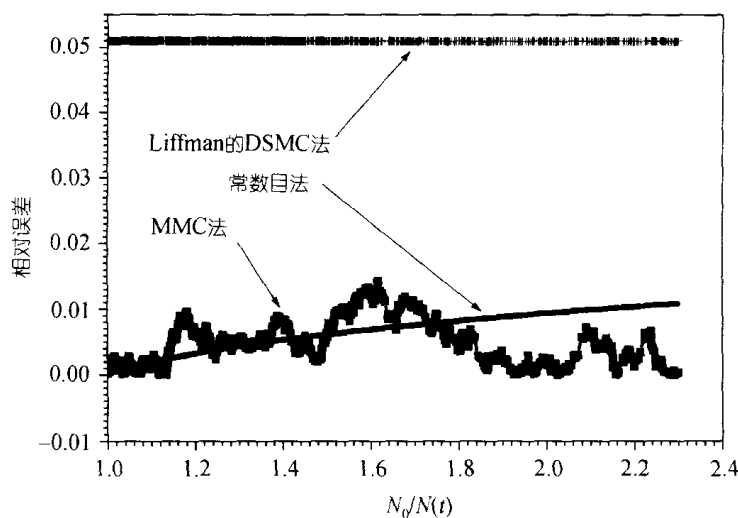


图 6 $N_0/N(t)$ 的相对误差

3.2 求解通用动力学方程的 MMC 算法的扩展能力

GDE 的建立把复杂的凝并机理和沉积机理分别转移到凝并核和沉积核, 对于任何存在颗粒凝并、沉积、破碎、成核、冷凝/蒸发、化学反应等现象的系统, GDE 方程均能够描述颗粒尺度分布的时空演变过程, 它具有坚固的物理基础和非常简单的物理假设, 可以描述从纳米材料到宇宙中的星系, 从气溶胶到可吸入颗粒物等. 当 MMC 算法把求解 GDE 的难题解决, 人们就可以把研究精力集中于发展各种领域、各种工况下的复杂的“核”模型. 值得注意的是, GDE 方程建立在 Euler 体系上, 而 Monte Carlo 算法建立在 Lagrange 体系上. 但是由于 MC 方法有着与 GDE 方程同样的物理基础和基本假设, 两者能够完美统一起来, 而且相比较于那些基于 Euler 体系的矩方法、分区法、离散法、离散-分区法等, MC 方法由于能够得到颗粒的轨道经历效应和历史效应而可以有更广泛的应用和更强的扩展能力.

目前, 所有算法均集中于求解描述颗粒尺度分布时间演变过程的 GDE, 这实际上只是描述了一个空间为零维的系统. 本文发展的 MMC 算法目前的计算能力也在这个层面上, 对于很多工业分析和科学研究, 得到这些信息已经足够, 但是对于某些场合, 如研究粉煤燃烧过程中易挥发的痕量重金属在炉膛和烟道甚至大气中的富集规律时^[16], 通过凝并、冷凝和成核等机理飞灰颗粒上吸附有重金属, 得到颗粒尺度分布的空间演变过程是非常重要的, 这有助于设计合理的捕捉策略和控制手段. 然而目前所有的数值算法对于颗粒尺度分布三维空间演变过程都难以涉足, 更令人失望的是, 由于种种限制和假设, 目前所有的 Monte Carlo 算法由于引入子系统概念甚至并不具备考虑颗粒尺度分布的空间扩散的扩展性. 而对于 MMC 算法而言, 由于引入加权虚拟颗粒的概念, 不仅具备较低的计算代价, 而且, 如果在某些情况下抛弃“子系统”概念, 就有更强的扩展能力来模拟整体系统的颗粒尺度分布函数随时间和空间的演变规律, 这部分的算法是作者下一阶段的研究目标之一.

MMC 算法具备非常强的扩展能力, 可以方便地把其他事件如冷凝/蒸发、成核和破碎等考虑进来; 甚至, MMC 算法可以与求解两相湍流模型的算法耦合起来, 不但描述颗粒尺度分布函数的时空变化, 而且描述颗粒相和流体相的各种参数; 本算法也可以进行多组分、多分散性颗粒的 GDE 求解. 这些也是我们下一阶段的研究目标之一.

4 结论

求解通用动力学方程的 MMC 算法引入加权虚拟颗粒的概念, 基于时间驱动, 通过调整相关虚拟颗粒的数目权值和尺度来强制保持虚拟颗粒数目和计算区域体积. 类似于常体积 MC, MMC 算法非常适合于工程实际计算和科学定量分析;

类似于常数目 MC, MMC 算法具有稳定和高的计算精度; 加权虚拟颗粒概念的引入使其可能抛弃子系统的概念, 从而具备考虑颗粒尺度分布的空间扩散、边界条件和两相流动力学等的扩展性.

MMC 算法的误差主要包括解耦误差、常数目和常体积误差、随机误差和统计误差, 其相对误差可近似为 $\delta = \left\{ \ln \left[N_0 / N(t) \right] \right\}^{0.8} / \sqrt{2N_c}$, 计算精度与常数目法在同一级别而优于 Liffman 发展的直接模拟 Monte Carlo 算法.

参 考 文 献

- 1 Kourti N, Schatz A. Solution of the general dynamic equation (GDE) for multicomponent aerosols. *Journal of Aerosol Science*, 1998, 29(1-2): 41~55
- 2 Barrett J C, Mills R. An approximate treatment of aerosol coagulation and removal. *Journal of Aerosol Science*, 2002, 33(9): 1327~1339
- 3 Gelbard F, Tambour Y, Seinfeld J H. Sectional representations for simulating aerosol dynamics. *J Colloid Interface Sci*, 1980, 76(2): 541~556
- 4 Landgrebe J D, Pratsinis S E. Gas-phase manufacture of particulate: interplay of chemical reaction and aerosol coagulation in the free-molecular regime. *Ind Eng Chem Res*, 1989, 28(10): 1474~1481
- 5 Wu J J, Flagan R C. A discrete-sectional solution to the aerosol dynamic equation. *J Colloid Interface Sci*, 1988, 123(2): 339~352
- 6 Kruis F E, Maisels A, Fissan H. Direct simulation Monte Carlo method for particle coagulation and aggregation. *AIChE Journal*, 2000, 46(9): 1735~1742
- 7 Garcia A L, van den Broek C, Aertsens M, et al. A Monte Carlo simulation of coagulation. *Physica A*, 1987, 143(3): 535~546
- 8 Lin Y, Lee K, Matsoukas T. Solution of the population balance equation using constant-number Monte Carlo. *Chemical Engineering Science*, 2002, 57(12): 2241~2252
- 9 Liffman K. A direct simulation Monte-Carlo method for cluster coagulation. *Journal of Computational Physics*, 1992, 100(1): 116~127
- 10 Fichthorn K A, Weinberg W H. Theoretical foundations of dynamical Monte Carlo simulations. *J Chem Phys*, 1991, 95(2): 1090~1096
- 11 赵海波, 郑楚光, 柳朝晖, 等. 考虑颗粒碰撞的多重 Monte Carlo 算法. *力学学报*, 2005, 37(5): 564~572
- 12 Hu K C, Mei R. Particle collision rate in fluid flows. *Physics of Fluids*, 1998, 10(4): 1028~1030
- 13 Nanbu K. Direct simulation scheme derived from the Boltzmann equation. I. Monocomponent gases. *J Phys Soc Jpn*, 1980, 49(11): 2042~2049
- 14 Williams M M R. Some exact and approximate solutions of the nonlinear Boltzmann equation with applicatios to aerosol coagulation. *Phys A: Math Gen*, 1981, 14(8): 2037~2089
- 15 Smith M, Matsoukas T. Constant-number Monte Carlo simulation of population balances. *Chemical Engineering Science*, 1998, 53(9): 1777~1786
- 16 Lockwood F C, Yousif S. A model for the particulate matter enrichment with toxic metals in solid fuel flames. *Fuel Processing Technology*, 2000, 65-66: 439~457