

1-4-2019

## Domain Decomposition Based SPH Parallel Computing Method Study and Its Application

Chen Hong

*Hypervelocity Impact Research Center, China Aerodynamics Research and Development Center, Mianyang 621000, China;*

Huang Jie

*Hypervelocity Impact Research Center, China Aerodynamics Research and Development Center, Mianyang 621000, China;*

Li Yi

*Hypervelocity Impact Research Center, China Aerodynamics Research and Development Center, Mianyang 621000, China;*

Liu Sen

*Hypervelocity Impact Research Center, China Aerodynamics Research and Development Center, Mianyang 621000, China;*

Follow this and additional works at: <https://dc-china-simulation.researchcommons.org/journal>



Part of the Artificial Intelligence and Robotics Commons, Computer Engineering Commons, Numerical Analysis and Scientific Computing Commons, Operations Research, Systems Engineering and Industrial Engineering Commons, and the Systems Science Commons

---

This Paper is brought to you for free and open access by Journal of System Simulation. It has been accepted for inclusion in Journal of System Simulation by an authorized editor of Journal of System Simulation.

---

# Domain Decomposition Based SPH Parallel Computing Method Study and Its Application

## Abstract

**Abstract:** To improve the speed and the capability of large scale simulation of SPH (Smoothed particle hydrodynamics), a parallel SPH method based on dynamic domain decomposition is developed. *The decomposition of arbitrary numbers of subdomains is achieved by adopting dimension based factorable method. Rapid parallel cut plane search is achieved by adopting bisection method. The data communication between compute nodes is reduced by improving the subdomain assignment. A parallel SPH program was developed based on this method, with which the process of projectile hypervelocity impacting thin plate producing debris cloud was simulated. The result shows that: the method is suitable for the simulation of large scale shock dynamic problem, the load balance is well kept during the simulation, and the method shows a good linear acceleration feature.*

## Keywords

SPH, parallel computing, dynamic domain decomposition, hypervelocity impact, parallel compute efficiency

## Recommended Citation

Chen Hong, Huang Jie, Li Yi, Liu Sen. Domain Decomposition Based SPH Parallel Computing Method Study and Its Application[J]. Journal of System Simulation, 2018, 30(10): 3717-3723.

## 一种基于动态区域分解的 SPH 并行算法研究及应用

陈鸿, 黄洁, 李毅, 柳森

(中国空气动力研究与发展中心超高速碰撞中心, 四川 绵阳 621000)

**摘要:** 为提高 SPH(光滑粒子流体动力学)方法的计算速度和大规模仿真能力, 发展了一种基于动态区域分解的 SPH 并行算法。采用基于维度的因数分解法实现任意数目的并行计算区域分解。采用二分法实现快速并行切割面搜索。通过改进子区域分配方法, 有效减少计算节点间通信数据量。基于该算法开发了并行 SPH 计算软件 PTS, 并对弹丸超高速斜撞击平板形成碎片云过程进行了并行 SPH 仿真。仿真结果表明: 该算法适应大规模冲击动力学仿真, 可有效保证仿真过程中的负载平衡, 并具有较好的线性加速特性。

**关键词:** SPH; 并行计算; 动态区域分解; 超高速碰撞; 并行计算效率

中图分类号: O246

文献标识码: A

文章编号: 1004-731X (2018) 10-3717-07

DOI: 10.16182/j.issn1004731x.joss.201810015

## Domain Decomposition Based SPH Parallel Computing Method Study and Its Application

Chen Hong, Huang Jie, Li Yi, Liu Sen

(Hypervelocity Impact Research Center, China Aerodynamics Research and Development Center, Mianyang 621000, China)

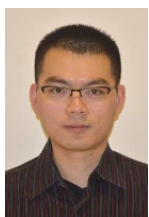
**Abstract:** To improve the speed and the capability of large scale simulation of SPH (Smoothed particle hydrodynamics), a parallel SPH method based on dynamic domain decomposition is developed. The decomposition of arbitrary numbers of subdomains is achieved by adopting dimension based factorable method. Rapid parallel cut plane search is achieved by adopting bisection method. The data communication between compute nodes is reduced by improving the subdomain assignment. A parallel SPH program was developed based on this method, with which the process of projectile hypervelocity impacting thin plate producing debris cloud was simulated. The result shows that: the method is suitable for the simulation of large scale shock dynamic problem, the load balance is well kept during the simulation, and the method shows a good linear acceleration feature.

**Keywords:** SPH; parallel computing; dynamic domain decomposition; hypervelocity impact; parallel compute efficiency

## 引言

光滑粒子流体动力学方法(SPH)是一种拉格朗日型的无网格算法, 早期主要应用于天体物理的模

拟<sup>[1-2]</sup>, 现已广泛应用于流体仿真和碰撞动力学仿真领域<sup>[3]</sup>。在 SPH 计算中物质被离散成粒子, 计算过程中不会出现网格畸变现象, 相对于传统的基于网格的数值仿真方法, SPH 方法在处理大变形问题、材料断裂问题、多物质交界面等问题有独特的优势。然而, 和传统的有网格算法相比, SPH 方法计算复杂度高、占用内存大, 导致计算速度慢、规模小, 在大规模复杂问题的仿真上仍有较多局限性, 并行计算技术是提高 SPH 方法的计算速度和



收稿日期: 2016-09-15 修回日期: 2017-01-07;  
作者简介: 陈鸿(1982-), 男, 湖北黄梅, 硕士, 副研究员, 研究方向为超高速碰撞数值仿真技术; 黄洁(1968-), 女, 四川渠县, 硕士, 研究员, 研究方向为超高速自由飞弹道靶和超高速碰撞; 李毅(1974-), 男, 湖南望城, 硕士, 研究员, 研究方向为超高速碰撞数值仿真技术。

<http://www.china-simulation.com>

• 3717 •

计算规模的主要手段。

SPH 计算有三种并行方式,一是基于运算过程的并行;二是基于粒子分解并行;三是基于计算区域分解并行。其中基于运算过程的并行是指将无数数据依赖关系的计算过程分配给多个处理器计算,刘桂荣、刘谋斌<sup>[3]</sup>开发的并行 SPH 程序即采用了这种并行方式。基于粒子分解的并行是指将计算域内所有粒子平均分配到每个处理器上,粒子分配与粒子空间位置无关。周浩等人<sup>[4-5]</sup>开发了基于粒子分解的并行 SPH 程序, David 等人<sup>[6]</sup>提出了基于大粒度粒子分解的并行 SPH 计算框架。基于计算区域的分解是指将计算区域划分成若干子空间,每个处理器负责计算一个子空间。这种并行方式又可以分为静态区域分解和动态区域分解,前者只在计算开始时进行区域划分,后者则是在计算过程中不断调整区域划分,确保计算过程中各个处理器的负载均衡,从而保证并行计算效率。

在上述三种 SPH 并行计算方式中,基于运算过程的并行与程序结构严格相关,扩展性较差,并行效率受制于程序结构;由于 SPH 算法中的粒子相互依赖特性,基于粒子分解并行会导致不同处理器间的通信量过大从而降低并行计算效率;基于区域分解的并行则不受问题规模和程序结构影响,具有扩展性好、并行计算效率高的特点。因此在三种并行计算方式中,基于区域分解并行方式得到了广泛应用,如并行 SPH 程序 JOSEPHINE<sup>[7]</sup>和开源程序 parallelSPHYSICS<sup>[8]</sup>等。王裴、洪滔<sup>[9]</sup>开发了基于静态空间区域分解的并行 SPH 程序 CSPH3D,使用三维微喷射和斜侵彻过程仿真检验并行计算效率,结果表明三维微喷射模拟并行计算效率较高,但斜侵彻过程仿真并行计算效率相对较低。周浩等人<sup>[5]</sup>的研究结果同样表明,对于粒子位置变化较大的计算,静态区域分解并行计算效率严重偏低。Ferrari 等人<sup>[10]</sup>针对自由表面流动问题,提出了一种基于动态区域分解的三维 SPH 并行方案,并应用在溃坝分析和流体撞墙等典型问题上,取得了较好的加速比。

本文基于 RCB(Recursive Coordinate Bisection, 递归坐标二分法)分解方法提出了一种动态区域分解并行 SPH 算法,编写了相应的并行 SPH 计算软件 PTS,并在此基础上开展了弹丸超高速撞击薄板过程数值仿真。

## 1 控制方程及 SPH 离散

在冲击动力学仿真中,控制方程为连续介质力学守恒方程,形式如下:

$$\begin{cases} \frac{d\rho}{dt} = -\rho \frac{\partial v^\beta}{\partial x^\beta} \\ \frac{dv^\alpha}{dt} = \frac{1}{\rho} \frac{\partial \sigma^{\alpha\beta}}{\partial x^\beta} \\ \frac{de}{dt} = \frac{\sigma^{\alpha\beta}}{\rho} \frac{\partial v^\alpha}{\partial x^\beta} \\ \frac{dx^\alpha}{dt} = v^\alpha \end{cases} \quad (1)$$

式中,  $\rho$ 、 $e$ 、 $v$ 、 $\sigma$ 、 $x$ 、 $t$  分别为密度、内能、速度、应力张量、位移、时间。

在 SPH 方法中,物质被离散成粒子,每个粒子既是物质点又是积分点,在粒子  $i$  处的函数可以通过依赖域内的粒子进行近似<sup>[11-12]</sup>:

$$f(x_i) = \sum_{j=1}^N \frac{m_j}{\rho_j} f(x_j) W(x_i - x_j, h) \quad (2)$$

式中:  $m$  为粒子质量,  $W$  为核函数,  $h$  为光滑长度,下标  $i$  为粒子自身编号,下标  $j$  为粒子  $i$  依赖域内的其它粒子编号。采用 SPH 方法,方程(1)可近似为:

$$\begin{cases} \frac{d\rho_i}{dt} = \sum_{j=1}^N m_j (v_i^\alpha - v_j^\alpha) \frac{\partial W_{ij}}{\partial x_i^\alpha} \\ \frac{dv_i^\alpha}{dt} = -\sum_{j=1}^N m_j \left( \frac{\sigma_i^{\alpha\beta}}{\rho_i^2} + \frac{\sigma_j^{\alpha\beta}}{\rho_j^2} + \Pi_{ij} \right) \frac{\partial W_{ij}}{\partial x_i^\beta} \\ \frac{de_i}{dt} = -\frac{1}{2} \sum_{j=1}^N m_j \left( \frac{p_i}{\rho_i^2} + \frac{p_j}{\rho_j^2} + \Pi_{ij} \right) (v_i^\alpha - v_j^\alpha) \frac{\partial W_{ij}}{\partial x_i^\beta} + \frac{1}{\rho} \tau_i^{\alpha\beta} \varepsilon_i^{\alpha\beta} \\ \frac{dx_i^\alpha}{dt} = v_i^\alpha \end{cases} \quad (3)$$

式中:  $\Pi$  为人工粘性,  $p$  为压力,  $\varepsilon$  为应变率张量。

## 2 基于动态区域分解的 SPH 并行计算方案

并行程序有两种结构, 分别为主从式结构和对等式结构。前者由主进程负责任务分配、更新、汇总, 子进程负责计算, 通信方式为一对多; 后者则各个进程无主从之分, 每个进程执行基本相同的计算任务, 由各个进程协调完成任务分配、更新、汇总, 通信方式为多对多。对于大规模 SPH 并行程序, 若采用主从式结构, 则会导致通信过载和主进程处理器空闲, 从而影响计算效率。因此, 本文采用对等式结构, 通过 MPI 标准接口实现进程间通信。

基于动态区域分解的并行 SPH 程序的每一个并行进程计算步骤如下:

- a. 读入输入数据并完成初始化工作;
- b. 和其它进程协调完成计算区域分解和分配工作;
- c. 建立邻近粒子列表;
- d. 计算粒子间相互作用力;
- e. 积分;
- f. 判断是否达到终止条件, 若未达到则返回 b 进行下一时间步计算;
- g. 输出计算结果。

除计算步骤 b 外, 其余计算步骤与串行 SPH 计算程序一致。步骤 b 通常被称为动态负载平衡, 本文提出的并行方法的具体实施流程如图 1 所示。由于动态负载平衡检测需要耗费一定的计算时间, 因此可采用间隔若干时间步检测一次的方法来提高效率。未进行负载平衡检测或检测到负载平衡时, 只需考虑各个子区域内粒子的迁入和迁出即可。负载不平衡时, 则需进行动态区域分解。本节将对其中的重要概念以及本文提出的动态区域分解实施方法进行详细介绍。

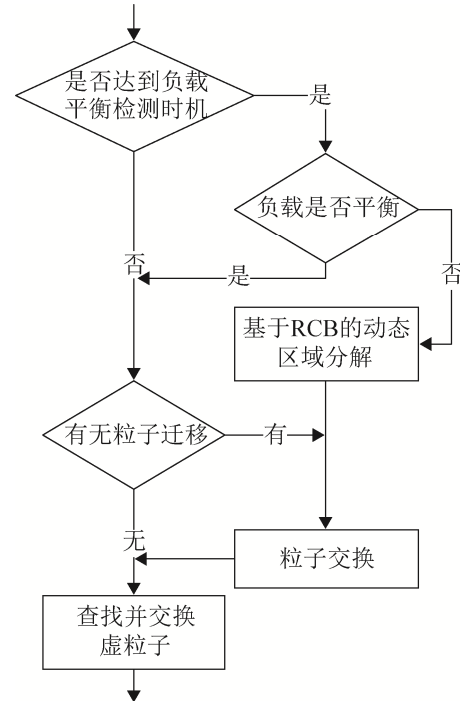


图 1 动态负载平衡实施流程

Fig. 1 Implement procedure of load balance

### 2.1 RCB 方法简介

RCB 是一种空间分解算法, 最初用于静态空间分解, 后被扩展到动态算法中。RCB 空间分解步骤为: 首先采用垂直于某个坐标轴的平面按照负载(由负载评估函数给出)将计算域平分为两部分, 沿哪个坐标轴划分, 由计算域的形状决定, 一般选择计算域投影最长的坐标轴。此后, 递归的应用此方法对已划分的区域进行划分, 直到区域总数与处理器数相等。通过改变分区大小, RCB 方法可以进行任意数目的区域划分。RCB 算法分解速度快, 分区分界面简单, 并且是一种“增量算法”, 即当计算区域发生较小的变化时, RCB 方法分解结果也只产生较小的变化。

### 2.2 虚粒子

由式(2)可知, 每个粒子对应物理量的计算需要利用依赖域内的粒子信息, 在基于区域分解的 SPH 并行计算中, 对于区域边界附近的粒子, 其积分依赖域将覆盖到相邻子区域, 这样在进行粒子积分近似时, 需要应用相邻子区域边界附近的粒子的

物理量, 为避免多次通信降低计算效率, 通常是一次性将相邻边界附近的粒子拷贝过来, 这些粒子便称之为虚粒子, 如图2所示。

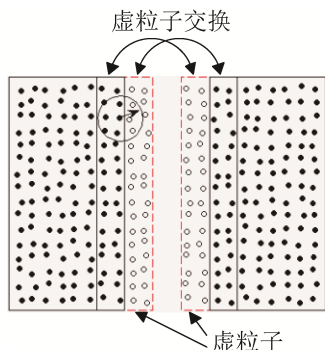


图2 虚粒子示意图

Fig. 2 Scheme of ghost particles

在计算过程中, 粒子的位置和物理属性均可能发生变化, 因此每一个计算步均需要进行虚粒子查找和交换。虚粒子从两个方面影响计算速度, 一是虚粒子数量, 虚粒子数量越多, 需要的通信时间越多; 二是虚粒子查找时间, 子区域间的边界越复杂, 则虚粒子查找时间越长。RCB算法的分区边界为平面, 因而有利于虚粒子快速查找。

### 2.3 区域分解方法

传统的RCB算法采用递归方式划分空间, 这种方式存在两个缺点, 一是处理器数目必须为2的N次方, 二是递归方式计算效率较低。为避免上述问题, 这里采用维度分割法来划分计算区域, 即先沿X坐标轴划分 $N_x$ 个子区域, 再将每个生成区域沿Y坐标轴划分 $N_y$ 个子区域, 最后将每个生成的区域沿Z坐标轴划分 $N_z$ 个子区域,  $N_x$ ,  $N_y$ ,  $N_z$ 和处理器数目 $NP$ 满足以下关系:

$$N_x \times N_y \times N_z = NP \quad (4)$$

这样计算区域划分问题就由递归算法转变为因数分解。对于一定数目的处理器, 存在多种因数分解方式, 因此还需寻找最优分解方法。在三维问题中, 最优的分解方式是尽量确保子区域接近立方体。采用立方体子区域有两个方面的好处, 一是各个子分区相邻面面积总和较小, 保证了全局虚粒子

数目较少, 从而减小通信量; 二是各个相邻面的面积一致, 保证了各个相邻面的虚粒子数基本一致, 从而使各个处理器间的通信量基本一致, 保持通信均衡。定义边长差异如下:

$$r_{diff} = \max \left( \left| \frac{dx}{N_x} - \frac{dy}{N_y} \right|, \left| \frac{dy}{N_y} - \frac{dz}{N_z} \right|, \left| \frac{dz}{N_z} - \frac{dx}{N_x} \right| \right) \quad (5)$$

式中:  $dx$ 、 $dy$ 、 $dz$  分别为计算域在 X、Y、Z 三个维度上的长度。

遍历所有分解方式, 最小的  $r_{diff}$  对应的  $N_x$ ,  $N_y$  和  $N_z$  即为最优区域分解方案。

### 2.4 切割面搜索方法

在区域分解中, 确定  $N_x$ ,  $N_y$  和  $N_z$  后, 还需找出每个坐标轴上的切割面, 保证每个子区域的粒子数基本相等。为提高切割面搜索速度, 发展了一种基于二分法的并行切割面搜索算法。

为便于描述算法, 以 X 轴为例, 将切割面从左至右依次编号为 0, 1, 2, ...,  $N_x-2$ , 每个切割面应满足以下关系:

$$\left| \frac{n_l(i) \times (N_x - i - 1) - n_r(i) \times (i + 1)}{(n_l(i) \times (N_x - i - 1) + n_r(i) \times (i + 1)) \times 0.5} \right| \leq d_i \quad (6)$$

$$i = 0, 1, 2, \dots, N_x - 2$$

式中  $i$  为切割面编号,  $n_l$  是切割面左边的粒子数,  $n_r$  是切割面右边的粒子数,  $d_i$  是切割面间粒子数差异容忍值。

二分法的查找过程为: 以 X 坐标最大值和最小值的中点为切割面将空间分为两个子空间, 此时若上式成立则搜索完成, 若不成立则根据两个子空间内的粒子数目找出切割面所属子空间, 对该子空间重复上述搜索过程, 直到上式成立, 即完成切割面搜索。在二分法的每一次迭代过程中, 每个进程统计本计算区域内每个切割面两边粒子数, 再通过进程间通信进行归约操作, 这样每个进程就获得了全局数据。当迭代搜索过程完成时, 每个进程都获得了所有切割面位置。

在区域分解时, 既可采用粒子数作为负载依据, 也可以采用邻近粒子对数作为负载依据, 同时



还可加入 CPU 速度和通信时间作为权值, 只需在式(6)中加入相应的权值即可。

## 2.5 计算区域分配方案

并行计算机一般由大量计算节点组成, 计算节点间通过网络进行数据交换。每个计算节点内又有若干个计算核, 计算核之间可通过内部总线的方式实现数据共享。因此, 节点内的数据通信速度远高于节点间的数据通信速度。并行计算任务可能运行在多个计算节点上, 这时应尽量减少节点间的通信量。在基于区域分解的并行 SPH 计算中, 相邻子区域之间需进行虚粒子交换和实粒子迁移, 即相邻区域之间均需进行数据通信。对于确定的区域划分, 相邻区域对数是固定的, 因此, 保证同一节点内的所有子区域相邻, 则可有效减少节点间的相邻区域数目, 从而减少节点间的数据通信量。本文采用各子区域中心之间的距离作为判据, 来实现同一节点内子区域相邻。

## 3 典型问题 SPH 并行仿真

### 3.1 并行仿真设置

碎片云是超高速碰撞特有的现象之一, 碎片云从产生到扩展过程是一个动态变化过程, 在此过程中, 碎片扩散范围较大, 可以有效的考察动态并行算法效率。因此采用弹丸超高速撞击薄板产生碎片云作为典型算例来考察 SPH 并行算法的计算效率。由于对称性问题通常具有较高的并行计算效率, 为体现一般性, 这里采用斜撞击过程作为仿真对象。

参照文献[13]中的试验数据设置计算条件。弹丸为铝球, 直径 4.02mm。靶板材料为铝, 厚度为 1.92mm。弹丸速度 4.47km/s, 撞击角度 45°。

弹丸和靶板均采用 Mie-Gruneisen 状态方程和 Johnson-Cook 本构模型, 并采用自由边界条件。SPH 粒子总数为 800 万, 采用本文开发 SPH 并行仿真软件 PTS 分别开展处理器数目为 1、2、4、8、16、32、48、96 的并行计算。

## 3.2 仿真结果分析

### 3.2.1 碎片云及靶板穿孔

超高速斜碰撞碎片云由弹丸碎片和靶板碎片组成。和超高速正碰撞相比, 弹丸碎片分布的存在明显的方向性。以弹丸撞击方向为“入射线”, 靶板为界面, 则弹丸飞溅碎片主要集中在“反射线”方向, 穿透靶板的弹丸碎片主要集中“折射线”方向附近。图 3 是 PTS 软件仿真结果与试验照片<sup>[13]</sup>的对比, 仿真结果在碎片云形状和碎片分布两个方面与试验结果吻合较好。靶板在超高速撞击斜撞击下, 会出现椭圆形穿孔, 试验测得的穿孔尺寸为 1.06cm×0.85cm, PTS 软件仿真获得的穿孔尺寸为 1.16cm×0.94cm, 在相同计算条件下采用 LS-DYNA 软件进行 SPH 仿真获得的穿孔尺寸为: 1.15cm×0.96cm, 两者结果较为接近, 和试验相比误差范围在 13%以内。由此证明, 本文提出的算法是有效、可信的。

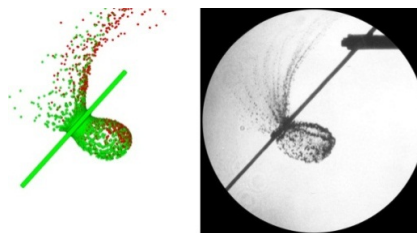


图 3 PTS 软件仿真结果与试验照片[13]的对比  
Fig. 3 Comparison between PTS simulation and test

### 3.2.2 并行计算过程及并行计算效率

对于基于区域分解的并行 SPH 计算程序, 其并行加速效率主要受两个因素影响: 并行处理器间通信时间和负载平衡时间。通信时间包括虚粒子交换时间、粒子迁移时间和粒子重新分配时间。负载平衡时间包括负载平衡检测时间和区域分解时间, 不包括通信时间。这里以处理器数目为 8 的算例为例来分析上述因素, 在计算过程中每一时间步统计一次虚粒子数、通信时间和负载平衡时间。图 4 是计算过程粒子数占总粒子数百分比变化曲线, 为便于观察规律, 这里只显示了 0~5 μs 时间段内的变化曲线。虚粒子数显周期变化, 每次回落均对应一次区域分解, 在整个模拟时间内(0~40 μs)虚粒子数占总粒子数百分比为平均为 1.9%。

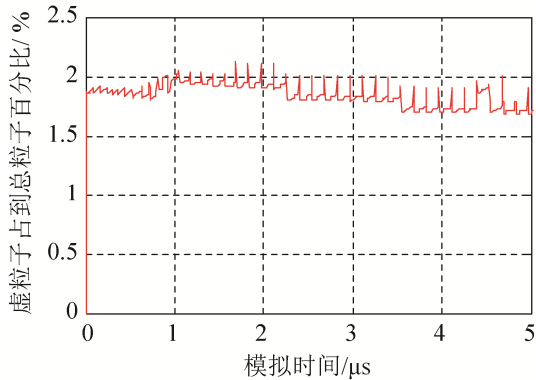


图4 计算过程中虚粒子百分比曲线

Fig. 4 Percentage curve of ghost particles during simulation

通信时间同样呈周期性变化,这是因为每次进行区域分解后需将实粒子重新分配,因而较为耗时。在整个模拟时间内(0~40 μs)通信时间占总时间百分比为7.7%。

图5是负载平衡时间占总时间百分比曲线,为便于观察这里只显示了0μs~5μs时间段内的变化曲线。当无需进行区域分解时,负载平衡耗时极少,当需要进行区域分解时,负载平衡耗时增多,每次区域分解耗时与上一次区域分解耗时基本一致。从整个模拟时间内(0~40 μs)来看,负载平衡平均时间平均百分比为1.7%。在整个模拟时间内,每个处理器的计算区域相对位置保持一致,每次区域重分,只是进行区域切割面微调,未发生区域分配大范围变动情况,上述情况表明该算法是一种“增量算法”,区域分解结果随问题域变化而变化,可有效避免单次大规模数据通信。

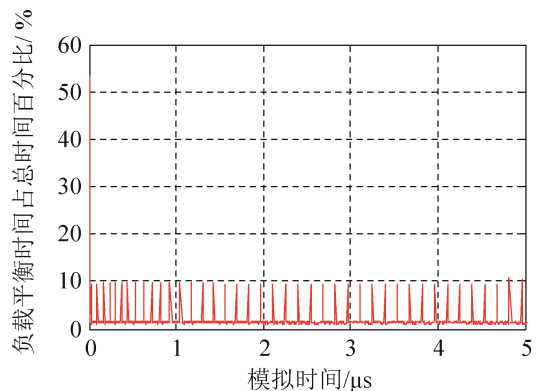


图5 负载平衡时间占总时间百分比曲线

Fig. 5 Percentage curves of load balancing

衡量程序的并行计算性能有两个指标,分别为并行加速比和并行加速效率。其中并行加速比定义为串行运行时间与并行运行时间的比值。并行加速曲线越接近线性,程序的并行扩展能力越好。并行加速效率定义为并行加速比与处理器数目的比值。并行加速效率越高,处理器利用率越高。图6是针对本文所选算例分别采用PTS软件和LS-DYNA软件获得的并行加速比曲线。PTS软件具有明显的线性加速特征,当处理器数目为96时,并行加速比为33.4。而LS-DYNA软件线性加速特征不明显,当处理器数目为96时,加速比出现了下降。由于通信开销以及其它由并行导致的额外开销,并行计算效率一般随处理器数目增加而降低,在本算例中当处理器数目在16个以内时,PTS软件并行计算效率最低为0.54,LS-DYNA软件并行计算效率最低为0.43;当处理器数目为96时,PTS软件并行计算效率为0.34,LS-DYNA软件并行计算效率为0.07,PTS软件并行算法明显优于LS-DYNA软件的SPH并行算法。

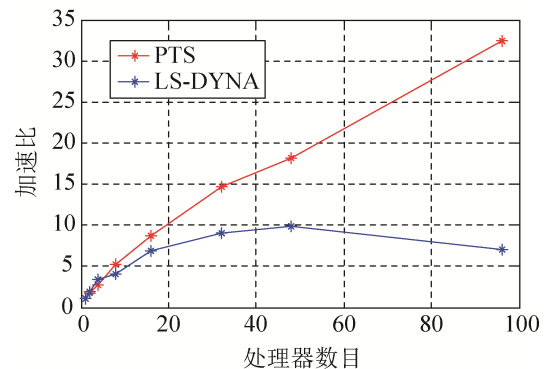


图6 并行加速比曲线

Fig. 6 Parallel accelerate ration curve

## 4 结论

本文提出了一种基于动态区域分解的SPH并行算法。基于SPH仿真的特点,改进了传统的递归区域分解方法;提出了可并行的快速切割面搜索算法;改进计算子区域分配方法,有效减少节点间通信量。编写了并行SPH计算软件PTS,开展了球形弹丸超高速斜撞击薄板过程数值仿真,实现了



较好的并行加速效果。研究结果表明:

a. 提出的并行 SPH 计算方法适用于大规模动态变化问题;

b. 提出的并行 SPH 计算方法较好的满足了“增量算法要求”, 计算过程中两次区域分解结果只发生较小的变化, 计算过程中的并行额外开销保持较低水平;

c. 提出的 SPH 并行计算方法加速比曲线具有较好的线性特征, 扩展性较好;

d. 开发了 SPH 并行计算程序, 在非对称性的超高速碰撞问题中, 当仿真粒子数为 800 万、96 个处理器时, 并行计算效率达到 0.34。

### 参考文献:

- [1] Lucy L B. A numerical approach of the testing of the fission hypothesis [J]. *Astron. J.* (S0004-6256), 1977, 82(12): 1013-1024.
- [2] Gingold R A, Monaghan J J. Smoothed particle hydrodynamics: theory and application to non-spherical stars [J]. *Monthly Notices of The Royal Astronomical Society* (S0035-8711), 1977, 181(3): 375-389.
- [3] Liu G R, Liu M B. *Smoothed Particle Hydrodynamics: A Meshfree Particle Method* [M]. Singapore: World Scientific Publishing Co Pte Ltd, 2003.
- [4] 周浩, 汤文辉, 冉宪文, 等. 光滑粒子流体动力学的一种并行数值计算方案 [J]. *航天器环境工程*, 2012, 29(1): 23-26.  
Zhou Hao, Tang Wenhui, Ran Xianwen, et al. A scheme of parallelization of a smoothed particle hydrodynamics algorithm [J]. *Spacecraft Environment Engineering*, 2012, 29(1):23-26.
- [5] 周浩, 徐志宏, 邹顺, 等. SPH 并行计算方案及其在自由表面流中的应用 [J]. *国防科技大学学报*, 2015, 37(1): 71-76.  
Zhou Hao, Xu Zhihong, Zou Shun, et al. Smoothed particle hydrodynamics parallel scheme particle decomposition domain decomposition [J]. *Journal of National University of Defense Technology*, 2015, 37(1): 71-76.
- [6] Holmes D W, Williams J R, Tilke P. A framework for parallel computational physics algorithms on multi-core: SPH in parallel [J]. *Advances in Engineering Software* (S0965-9978), 2011, 42(11): 999-1008.
- [7] Cherfils J M, Pinon G, Rivoalen E. JOSEPHINE: a parallel SPH code for free-surface flows [J]. *Computer Physics Communication* (S0010-4655), 2012, 183(7): 1468-1480.
- [8] Gomez-Gesteira M, Crespo A J C, Rogers BD, et al. SPHysics- development of a free-surface fluid solver-Part2: Efficiency and test cases [J]. *Computers & Geosciences* (S0098-3004), 2012, 48: 300-307.
- [9] 王裴, 洪滔. 三维光滑粒子流体动力学并行计算程序 [J]. *计算物理*, 2006, 23(4): 431-435.  
Wang Pei, Hong Tao. *Parallel Computation of Three-Dimensional Smoothed Particle Hydrodynamics* [J]. *Chinese Journal of Computational Physics*, 2006, 23(4):431-435
- [10] Angela F, Michael D, Toro E F. A new 3D parallel SPH scheme for free surface flows [J]. *Computers and Fluids* (S0045-7930), 2009, 38(6): 1203-1217.
- [11] Monaghan J J. Particle methods for hydrodynamics [J]. *Computer Physics Report* (S0010-4655), 1985, 3(2): 71-124.
- [12] Monaghan J J. Smoothed particle hydrodynamics [J]. *Rep. Prog. Phys.* (S0034-4885), 2005, 68(8): 1703-1759.
- [13] 柳森, 李毅, 黄洁, 等. 用于验证数值仿真的 Whipple 屏超高速撞击试验结果 [J]. *宇航学报*, 2005, 26(4): 505-508.  
Liu Sen, Li Yi, Huang Jie, et al. Hypervelocity Impact Test Results of Whipple Shield for the Validation of Numerical Simulation [J]. *Journal of Astronautics*, 2005, 26(4): 505-508