

分类号_____

UDC _____

密 级 _____

编 号 _____

清 华 大 学

博 士 后 研 究 报 告

物质点法几个相关问题的研究

马 志 涛

清 华 大 学 (北京)

2010年6月

1、物质点法面向对象设计

AN OBJECT-ORIENTED MPM FRAMEWORK

2、物质点接触算法改进与混凝土侵彻的数值模拟

IMPROVED CONTACT ALGORITHM OF MPM AND SIMULATION OF CONCRETE
PENETRATION

博 士 后 姓 名: 马 志 涛

流动站（一级学科）名称: 清华大学力学

专 业（二级学科）名称: 一般力学

合 作 导 师 : 张 雄 （教授）

研究工作起止日期 2008年7月——2010年6月

物质点法面向对象设计

摘 要

由于物质点法(MPM)需要同时用到网格和粒子的数据,所以MPM比其他的方法会耗费更多的内存。因此,面向工程应用(如:爆炸侵彻问题)开发一套有效的MPM框架是很必要的。这篇文章提出了一个MPM程序的新框架,采用面向对象设计,能够处理多物体接触、大变形、大应变、高应变率等复杂物理现象。对于不同的求解过程,该框架都具有较好的灵活性、可扩展性、可维护性。将接触算法与USF和MUSL算法统一起来,并提出了一种改进的接触检测方式来避免物体的提前接触。为了减少内存耗费量,提出了局部多重网格、动态分配历史变量、动态网格的方法;为了提升运行效率,对程序流程进行了优化。最后,对一些算例进行了模拟,验证了新框架的具有较高的执行效率和较低的内存耗费量。

关键词: 物质点法, 多重网格, 接触, 侵彻, 框架

Abstract

The Material Point Method (MPM) is more expensive in terms of storage than other methods, as MPM makes use of both mesh and particle data. Therefore, it is critical to develop an efficient MPM framework for engineering applications, such as impact and explosive simulations. This paper presents a new architecture for MPM computer code, developed using object-oriented design, which enables MPM analysis of a mass of grains, large deformation, high strain rates and complex material behavior. It is flexible, extendible, and easily modified for a variety of MPM analysis procedures. An MPM scheme combining contact algorithm with USF and MUSL formulation is presented, and an improved contact detection scheme is proposed to avoid contact occurring earlier than actual time, and several schemes are developed to reduce the memory requirement and computational cost, including the local multi-mesh contact algorithm, dynamic internal state variables for materials, dynamic grid and moving grid technique. Finally, some numerical examples are presented to demonstrate the computational efficiency and memory requirement of the framework.

Key words: Material point method, Multi-mesh, Contact, Impact, Framework.

目 录

摘 要.....	2
Abstract.....	3
目 录.....	4
1 概述	5
2 MPM流程	6
2.1 离散化	6
2.2 带有接触算法的时间积分	7
2.3 应力与应变计算	9
2.4 MPM3Dpp的流程图	9
3 整体框架	9
3.1 物质点类的结构	10
3.2 材料类的结构	11
3.3 局部多重网格	12
3.4 动态网格	14
3.5 移动网格	14
3.6 流程优化	15
4 算例	16
4.1 侵彻	16
4.2 平板压球	17
4.3 圆盘晃动	18
4.4 圆环碰撞	20
5 结论	21
参考文献.....	22

1 概述

物质点法 (MPM) ^[1,2] 作为 FLIP ^[3,4] 在固体力学中的扩展, 其核心思想是同时利用欧拉和拉格朗日两种方法的优点。它将物质区域离散化为物质点, 在预设的规则背景网格上进行动量积分, 这样就完全避免了网格畸变。在过去10余年中, MPM已经发展为模拟具有挑战性的工程问题的有力工具, 已被应用于大应变问题^[5,6]、动态能量释放率^[7]、断裂力学^[8]、动态失效^[9,10]、超高速撞击^[11]、爆炸^[12]、薄膜^[13]、颗粒材料^[14-18]和泡沫材料^[19]等的模拟。

接触是工程领域非常广泛的一种现象。因为物质点法更新位置时采用单一速度场映射, 所以无滑移接触在传统物质点法中自动处理, 而不需要查找接触面。Bardenhagen等人^[17,18]扩展了传统物质点法, 将滑动及带有摩擦的接触引入到物质点法中, 在接触节点处使用Coulomb摩擦。物体间的接触力通过节点相对速度来计算。为了实现接触算法, Hu和Chen^[22]提出了多重全局网格映射的想法, 让每个物体都有其独立的背景网格, 而不是都在一个共同的背景网格中。但如果计算的物体比较多, 这种方法就会浪费非常多的内存。Pan等^[23]基于[22]提出了三维全局多重网格接触算法, 其物体间的接触力通过接触面上的加速度连续条件来点对于物质点法, 有几种可选的节点速度、应力更新方式。1994年, Sulsky等^[2]解决这个问题。一种名为MUSL (Modified Update Stress Last) 由Sulsky等^[2]提出, 在更新节点速度之前物质点动量会映射到网格两次。另一种修正是在更新动量之前更新应变^[20], 这种方法被命名为USF (Update Stress First)。在许多模拟中, 这两种方法相比USL, 都能够很大程度提升计算的稳定性。Nairn^[21]将这几种方法都统一实现在他们编制的裂纹模拟程序中。

还有一些其他的关于物质点法改进的研究工作。当物质点跨越背景网格时, 由于形函数导数不连续, 会引入人工噪声。Bardenhagen^[24]发展了广义插值物质点法 (GIMP) 解决了这个问题。随后, Ma^[25] 提出了一种用于GIMP的改进的结构化网格。Steffen^[26] 分析并提出了GIMP中的光滑长度对计算误差和稳定性的影响, 以及形函数和边界处理对空间收敛性的影响。Chen^[27]将浸没边界法在物质点法的框架下进行了实现。Wallstedt^[28]提出了一种改进的映射方式, 依然使用原有的速度梯度信息, 提升了线性形函数的映射精度并降低了映射时物质点的位置和密度对非线性函数的依赖。Chen^[29]在物质点法框架下实现了热力模型, 使得不同微分控制方程中的梯度和散度算符可以在单一的计算域中离散。

为了解决各种各样的工程问题, 越来越多的本构模型、状态方程和失效模型被集成到数值模拟软件中。这导致了許多材料的状态量需要存储下来, 材料之间的关系也变得更加复杂。而不同的算法和材料模型又需要被集成到同一个程序中。随着物质点法程序复杂度的增加, 其灵活性、可扩展性、可维护性、重用性等变得愈加重要。面向过程开发的代码包含了很多复杂的数据结构, 它们在整个程序中都可以访问。这

种全局访问降低了系统的灵活性，使得难以修改和扩展已有的代码来适应新的需求。这种不灵活性主要表现在几个方面：(1) 即使是修改程序中的一小部分，也需要了解整个程序；(2) 代码重用很困难；(3) 在数据结构上的很小的修改也可能会引起整个系统的调整；(4) 组件间众多的相互依赖被隐藏而难以确定；(5) 数据结构的完整性难以保证。

由于这些原因，一个合适的物质点法计算环境的需求是明显的。面向对象程序(OOP)技术已被表明能够显著提升软件的可扩展性和重用性。一个精心设计的框架或者架构可以明显降低程序在维护和扩展方面的难度。在过去几十年中，许多结构工程研究人员追求面向对象设计。1990年，Fenves^[30]指出了将OOP应用于工程软件开发的好处。同年，Forde等^[31]详细说明了如何将面相对应的设计方式用于有限元法(FEM)，他们将必要的组件从FEM中抽象成为类（比如：单元、节点、材料、边界条件、载荷等），使得其可以为后续开发者使用。从那以后，完全的面向对象FEM架构陆续提出^[32-38]。近年来，Luo等^[39]提出了面向对象无网格Galerkin法。Atluri^[40-41]提出了MLPG(Meshless Local Petrov-Galerkin)的通用框架，使得可以灵活选择局部弱形式、试函数和检验函数来求解偏微分方程组。Zhang等^[42]提出了使用Java开发的集成CAD的设计分析框架jNURBS，其使用无网格法进行材料行为分析和优化设计。

C++作为面相对象的语言，其运行效率高于基于虚拟机或通用语言运行时得语言（例如：Java和C#），与传统语言（例如：FORTRAN和C）。这篇论文提出了新的面向对象物质点法框架MPM3Dpp，采用C++开发，设计用于大变形、高应变率和复杂材料行为的模拟。MPM3Dpp源于MPM3D^[43]（支持OpenMP 并行，使用FORTRAN开发），现在采用面向对象的方法被重新设计。考虑到执行效率和内存耗费量对于计算程序是很重要的，我们利用C++面向对象和内存分配机制进行了程序优化。为了体现新框架具有高效率 and 低内存耗费量的优点，还进行了一些算例的模拟。

2 MPM流程

2.1 离散化

物质点法的离散化起始于将物质区域划分为一组物质点。固体力学物质点法弱形式由下述方程给出^[1,2]。动量方程

$$\rho \frac{d\mathbf{v}}{dt} = \nabla \cdot \boldsymbol{\sigma} + \rho \mathbf{b} \quad (1)$$

在背景网格以拉格朗日格式求解。式中， $\mathbf{b}$ 为体力， ρ 为密度， $\mathbf{v}$ 为速度， $\boldsymbol{\sigma}$ 为应力张量。

物质点法中，连续体离散化为 N_p 个物质点。每个物质点记录着位置、速度、质量、密度、应力、应变和其他所有的本构模型的状态量。在每个时间步，物质点的质量和速度都映射到背景计算网格上。映射的节点质量 m_i 和动量 $\mathbf{p}_i$ 分别通过下列方程获

得

$$m_i = \sum_{p=1}^{n_p} m_p N_{ip} \quad (2)$$

$$\mathbf{p}_i = \sum_{p=1}^{n_p} m_p \mathbf{v}_p N_{ip} \quad (3)$$

这里， m_p 是质点质量， $\mathbf{v}_p$ 是质点速度， N_{ip} 是节点*i* 处质点 *p* 的形函数值。

2.2 带有接触算法的时间积分

当两个物体相互接近时，其运动收到约束是必要的。当两个物体的动量映射在同一节点处时，接触就可能会发生，物体*r*和*s*的接触可通过比较节点速度 $\mathbf{v}_i^r$ 和 $\mathbf{v}_i^s$ 来判断^[23]，

$$(\mathbf{v}_i^r - \mathbf{v}_i^s) \cdot \mathbf{n}_i^r > 0 \quad (4)$$

这里， $\mathbf{n}_i^r$ 是物体*r*在边界节点*i*处的外法线方向。

Bardenhagen^[18] 提出了一种根据法向外力 $\mathbf{t}_n^\alpha$ 的判断接触方法。该外力通过积分物质点对接触网格节点的贡献来计算(即： $\mathbf{t}_n^\alpha = \mathbf{n}^\alpha \cdot \boldsymbol{\sigma}^\alpha \cdot \mathbf{n}^\alpha$)，该方法更适合判断物体的分离。

上述两种接触检测方式都会出现提前接触的情况（如果两个物体的间距小于两倍的网格间距，它们就可能会被认为发生接触）。这里提出了一种改进的接触检测方式来避免物体的提前接触（见图1）。假设两个物体在节点*i*处相互接近，当它们之间的距离 D_i^{rs} 可通过下式来计算

$$D_i^{rs} = D_i^r + D_i^s \quad (5)$$

$$D_i^g = -\max(\mathbf{X}_{ip}^g \cdot \mathbf{n}_i^g + l_p^g)$$

$$\mathbf{X}_{ip}^g = \mathbf{X}_p^g - \mathbf{X}_i$$

这里 $\mathbf{X}_i$ 是节点*i*的位置， l_p^g 和 $\mathbf{X}_p^g$ 分别是某个时间步时物体*g*中某个质点的长度和位置。如果 D_i^{rs} 小于指定的距离*D*，那么才表明接触发生。

考虑到法向动量的守恒，接触节点动量由下列方程更新^[23]

$$\bar{\mathbf{p}}_i^r = \mathbf{p}_i^r - \frac{m_i^s \mathbf{p}_i^r - m_i^r \mathbf{p}_i^s}{m_i^r + m_i^s} \cdot \mathbf{n}_i^r \mathbf{n}_i^r \quad (6)$$

$$\bar{\mathbf{p}}_i^s = \mathbf{p}_i^s + \frac{m_i^s \mathbf{p}_i^r - m_i^r \mathbf{p}_i^s}{m_i^r + m_i^s} \cdot \mathbf{n}_i^r \mathbf{n}_i^r \quad (7)$$

网格加速度可以由下式计算

$$m_i \frac{d\mathbf{v}_i}{dt} = \mathbf{f}_i = \mathbf{f}_i^{\text{int}} + \mathbf{f}_i^{\text{ext}} \quad (8)$$

这里，节点处的内力和外力可由下式计算

$$\mathbf{f}_i^{\text{int}} = - \sum_p^{N_p} \frac{m_p}{\rho_p} \boldsymbol{\sigma}_p \cdot \nabla N_{ip} \quad (9)$$

$$\mathbf{f}_i^{\text{ext}} = \sum_p^{N_p} N_{ip} \mathbf{b}_p \quad (10)$$

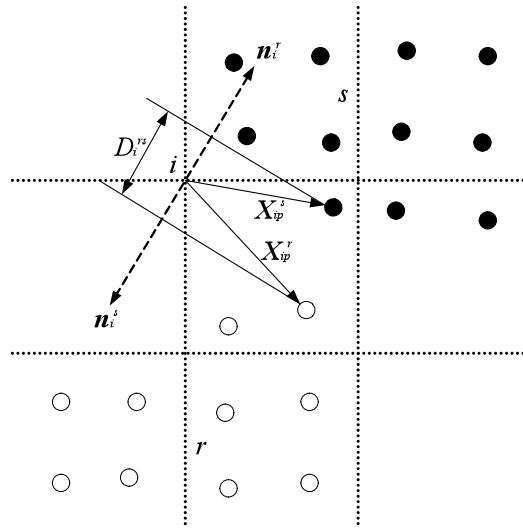


图1 改进的接触检测方式

一旦物体 r 和 s 发生接触，它们沿法向的加速度必须相等，这样就需要更新它们的节点力^[23]

$$\bar{\mathbf{f}}_i^r = \mathbf{f}_i^r - f_i^{\text{nor}} \mathbf{n}_i^r - f_i^{\text{fric}} \mathbf{s}_i^r \quad (11)$$

$$\bar{\mathbf{f}}_i^s = \mathbf{f}_i^s + f_i^{\text{nor}} \mathbf{n}_i^r + f_i^{\text{fric}} \mathbf{s}_i^r \quad (12)$$

$$f_i^{\text{nor}} = \frac{m_i^s \mathbf{f}_i^{r,\text{int}} - m_i^r \mathbf{f}_i^{s,\text{int}}}{m_i^r + m_i^s} \cdot \mathbf{n}_i^r \quad (13)$$

$$f_i^{\text{fric}} = \min(\mu f_i^{\text{nor}}, f_i^{\text{tan}}) \quad (14)$$

$$f_i^{\text{tan}} = \frac{(m_i^s \mathbf{p}_i^r - m_i^r \mathbf{p}_i^s + (m_i^s \mathbf{f}_i^{r,\text{int}} - m_i^r \mathbf{f}_i^{s,\text{int}}) \Delta t) \cdot \mathbf{s}_i^r}{(m_i^r + m_i^s) \Delta t} \quad (15)$$

这里， $\mathbf{s}_i^r$ 是节点 i 沿边界的单位切向矢量， μ 是摩擦系数， Δt 是时间步长。

使用显式积分求解方程(8)，从而获得节点加速度，其时间步长满足稳定性条件。临界时间步长是最小的单元尺寸与声速及粒子速度之和的商。

更新粒子位置和速度的方程分别为^[2],

$$\bar{\mathbf{x}}_p = \mathbf{x}_p + \sum_{i=1}^{n_i} \frac{\mathbf{p}_i}{m_i} N_{ip} \Delta t \quad (16)$$

$$\bar{\mathbf{v}}_p = \mathbf{v}_p + \sum_{i=1}^{n_i} \frac{\mathbf{f}_i}{m_i} N_{ip} \Delta t \quad (17)$$

2.3 应力与应变计算

应变和旋量增量通过分别计算下列方程获得

$$\Delta \varepsilon_{p\alpha\beta} = \frac{1}{2} \sum_{i=1}^{n_i} (G_{ip\beta} v_{ia} + G_{ip\alpha} v_{i\beta}) \Delta t \quad (18)$$

$$\Delta \Omega_{p\alpha\beta} = \frac{1}{2} \sum_{i=1}^{n_i} (G_{ip\beta} v_{ia} - G_{ip\alpha} v_{i\beta}) \Delta t \quad (19)$$

这里 G_{ip} 定义为

$$\mathbf{G}_{ip} = \nabla N_{ip} \quad (20)$$

将应变与旋量增量输入给材料本构来更新粒子的应力，任何本构都可以使用。

2.4 MPM3Dpp的流程图

MPM3Dpp流程图如图2所示，其包含了接触算法和3种不同的粒子应力应变更新方式（USF、MUSL）。

3 整体框架

系统架构设计的核心思想是将任务分解为若干个不同的对象。整体设计的主要的类显示在图3中，采用UML (Unified modeling language)静态模型来显示架构。类名显示在方块中，类之间的引用关系用带有菱形的实线连接。

CMPM3DPP是顶层类，用于控制整体计算过程，它创建类CSolution、CWriteResult和CDomain来完成整个的任务。这个类在计算开始前解析输入文件，完成后即开始求解。CSolution专门负责整个的求解过程。它获得CDomain对象的属性并进行求解，再通过CWriteResult对象输出计算结果。计算时间通过Clock类来进行记录。CDomain类负责创建和初始化背景网格(CGrid类)、材料 (CMatGeneric类)和物体(CBody类)。其他的类会在下面子章节中描述。

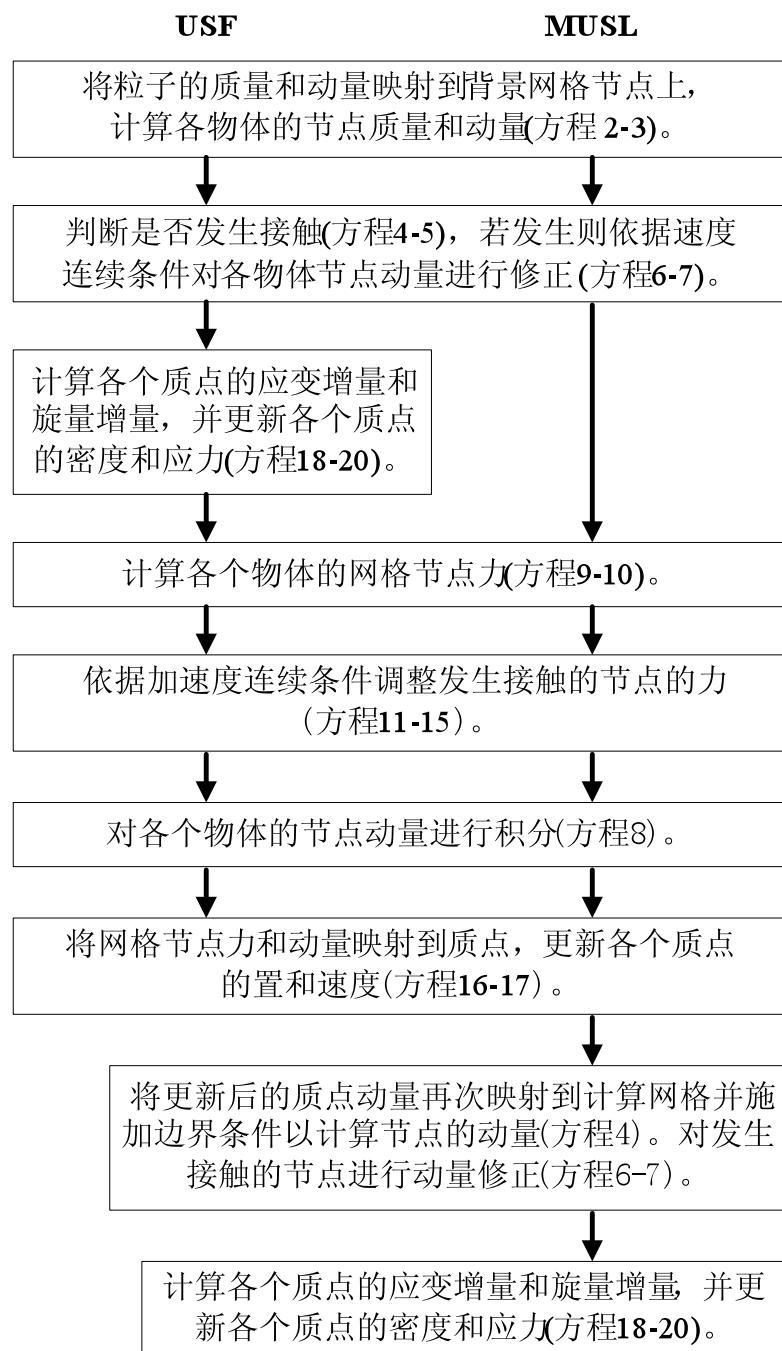


图2 MPM3Dpp计算流程

3.1 物质点类的结构

MPM粒子通过类CParticle(由类CBody来创建，见图3)来描述，该类包含了坐标、速度、体积、人工体积粘、平均应力、失效、内能、质量、声速和其他的属性。为了尽可能的节省内存，只将材料和算法中需要的状态量保存在一个名为ExtraProp的动态分配的额外变量数组中。额外变量可由CBody、CDomain和所有的材料类添加。

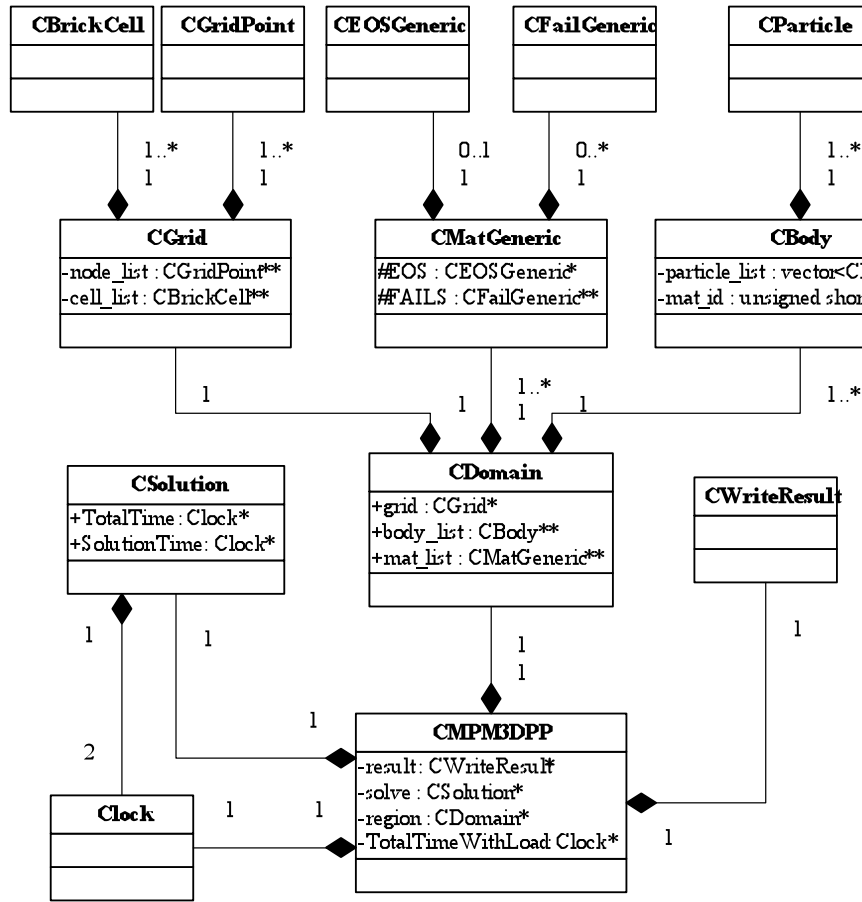


图3 MPM3Dpp的UML静态模型图

一个名为MPM的命名空间被使用，用来定义程序版本、算法类型、粒子属性、全局设置等。在MPM命名空间中，可选的额外粒子属性ExtraParticleProperty的类型定义为ENUM数据类型。为了便于快速存取ExtraProp额外变量，对下标运算符进行了重载，即：

```

inline double & operator[] (MPM::ExtraParticleProperty seq) {
    return ExtraProp[ExtraParticlePropPos[seq]]
}

```

这里，ExtraParticlePropPos是一个定义在类CBody中指向存储额外变量位置的指针，seq是某种额外属性的类型。

3.2 材料类的结构

材料类之间的引用和继承关系显示在图4中。类CMatGeneric（强度模型）、CEOSGeneric（状态方程）和CFailGeneric（失效模型）都是抽象类，它们需要派生类的实现来完成材料的定义。派生类允许重载基类来反映特殊的需求。一个材料可以最多有一个状态方程和多个失效模型。变量material_id用来指定每个CBody对象中的材料编号。列表1给出了CMatGeneric类的部分变量和函数的定义声明。一些材料的状态量可

以通过ApplyParticleProperties函数添加到额外变量中。

3.3 局部多重网格

在MPM中，背景网格负责动量方程的求解，粒子的状态根据求解后的值来更新。Bardenhagen、Brackbill和Sulsky提出了MPM的接触算法，用来模拟多个颗粒的相互作用^[13]。为了在物质点法中引入接触算法，Hu and Chen^[21]提出了全局多重映射网格的方法。在该方法中，每个颗粒独立占有一套背景网格，而不是在统一的一个网格中，如图5所示。对于二维情况，这些独立的网格都规则的分布在整個计算区域，与使用一套背景网格时的情形相同。这种方法会耗费很多的内存资源，提别是当颗粒很多的时候会更加显著。故在此提出局部多重网格的方法，用于解决耗费大量内存的问题。局部多重网格也就是说仅在交叠的节点处创建多重，其他区域还是只有一重，如图6所示。

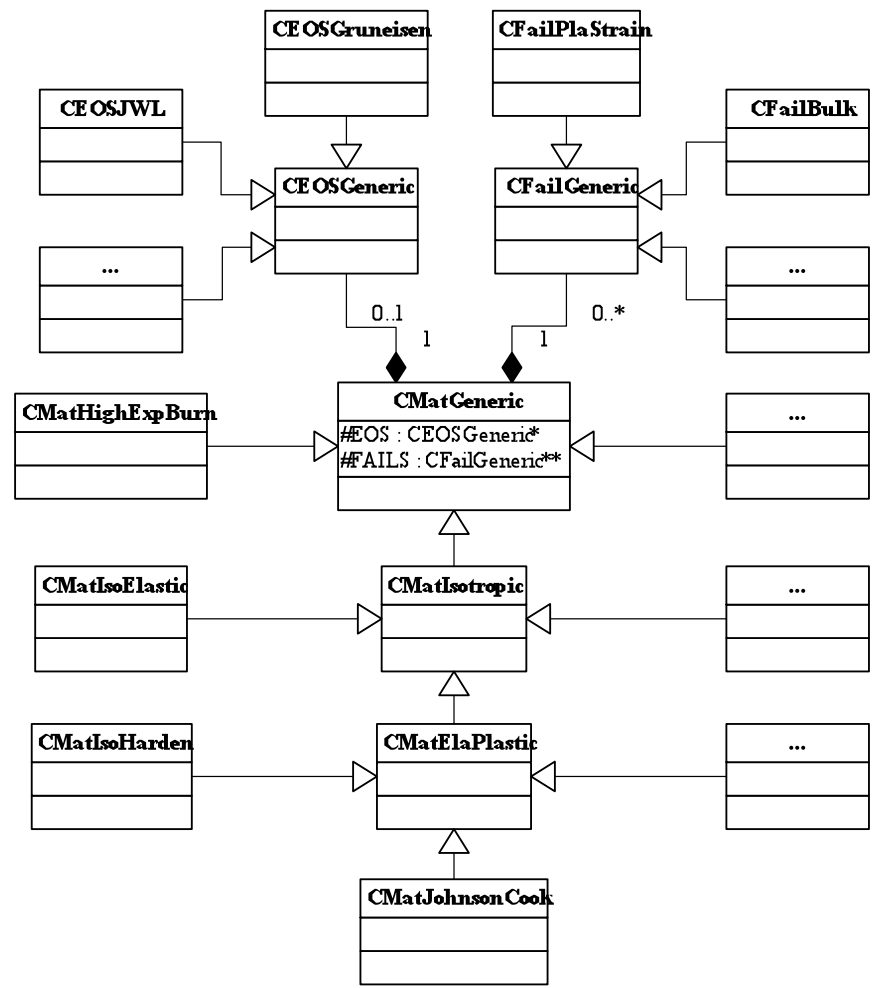


图4 材料类的引用和继承关系

protected:

```

CEOSGeneric *EOS;
CFailGeneric ** FAILS;
void ApplyParticleProperties(CBody *body);
public:
    virtual void UpdateStress(double(&de)[6], double (&vort)[6], CParticle *p, double
    &vold, double &dt);
    virtual double SoundSpeed(double rho, CParticle *p);

```

列表1 类CMatGeneric的部分定义

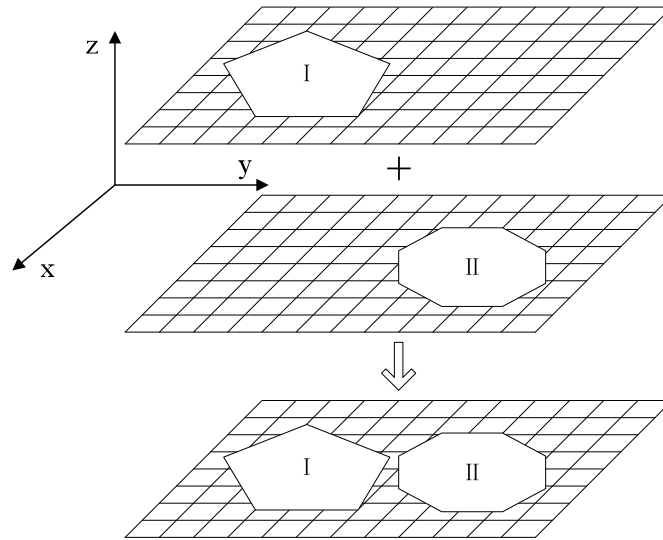


图5 全局多重映射网格^[22]

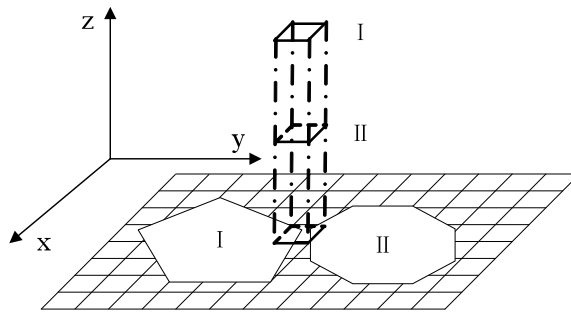


图6 局部多重映射网格

对于接触和非接触两种情况，节点间的连接关系定义在CBrickCell类中（见图3, cell_list）；CGridNodeForContact和CGridNodeForNoContact继承于CGridPoint抽象类（包含坐标、边界条件，以及用于被继承类调用的一些函数），根据该节点是否发生接触分别进行创建。继承类允许重载基类的函数来实现其特殊的需要。这允许在抽象的层面上使用继承类，隐藏特殊的实现细节，仅使用基类中声明的接口服务。列表2列出了节点基础类CGridPoint的部分定义。类之间的引用和继承关系分别使用带有实心菱形的实线和带有空心箭头的实线连接，如图7所示。

protected:

```

double Xg [3];
void _ApplyBoundaryConditions (CGridNode *gd, unsigned char fixed);
void _InitiateMomentum (CGridNode *gd);
void _IntegrateMomentum(CGridNode *gd, unsigned char fixed, double DTx);
public:
virtual bool Clear (bool check=false);
virtual CGridNode * GetGridNodeByCom (unsigned short comid);
virtual void ApplyBoundaryConditions (unsigned char fixed);
virtual void InitiateMomentum ();
virtual void IntegrateMomentum(unsigned char fixed, double DTx);

```

列表2 CGridPoint类的部分定义

对于非接触情况，CGridNodeForNoContact类继承于CGridPoint和CNoContactProperty（包含质量、动量、力等）两个类。

对于接触情况，CGridNodeForContact类继承于CGridPoint类，包含CNoContactProperty和CContactProperty（包含法向量、是否接触、物体数量、每个物体的属性等）两个类中的一个实例。一个节点的接触和非接触状态可根据其物体的数量来进行转换。如果节点处仅有一个物体，那么它会保存在CNoContactProperty类的实例中，否则会保存在CContactProperty类中的TinyMap类型的变量中。TinyMap是一个类似C++标准库中map的模板类容器。TinyMap用于从包含少量键值对的集合中进行快速存取，它比map类具有更高的效率和更少的内存耗费用量。即使模拟中有很多物体，MPM3Dpp也能胜任，且具有较高的效率和很少的耗费用量。

3.4 动态网格

通常，粒子只占有部分计算区域，以至于剩余的区域就没有用处又会浪费内存。为了解决这个问题，发展了动态网格方法。在计算开始时，创建节点和网格的指针数组，并初始化为NULL，节点和单元类实例的内存不立即进行分配，直到有粒子在节点的周围或在单元内。在每个时间步的开始，对于没有粒子的节点和单元的实例，将其变量初始化或每隔指定步进行删除。如果节点没有被实例化，那么它不会参与计算。这个方法可以降低内存耗费用量并提升计算效率。

3.5 移动网格

随着粒子的移动，原来的网格可能不能覆盖计算区域，这样会导致某些粒子在计算中被忽略掉。为了解决这个问题，这里提出了移动网格的方法。当那种情况发生时，网格会被移动或变大。这种方法的实现步骤列举如下：

- i. 对所有的粒子循环，如果有粒子接近边界（除固定、对称等边界之外），记录网格需要移动的方向（即x、y或z）和位置。

- ii. 沿着记录的方向放大或移动网格。
- iii. 根据新的网格大小调整单元尺寸。

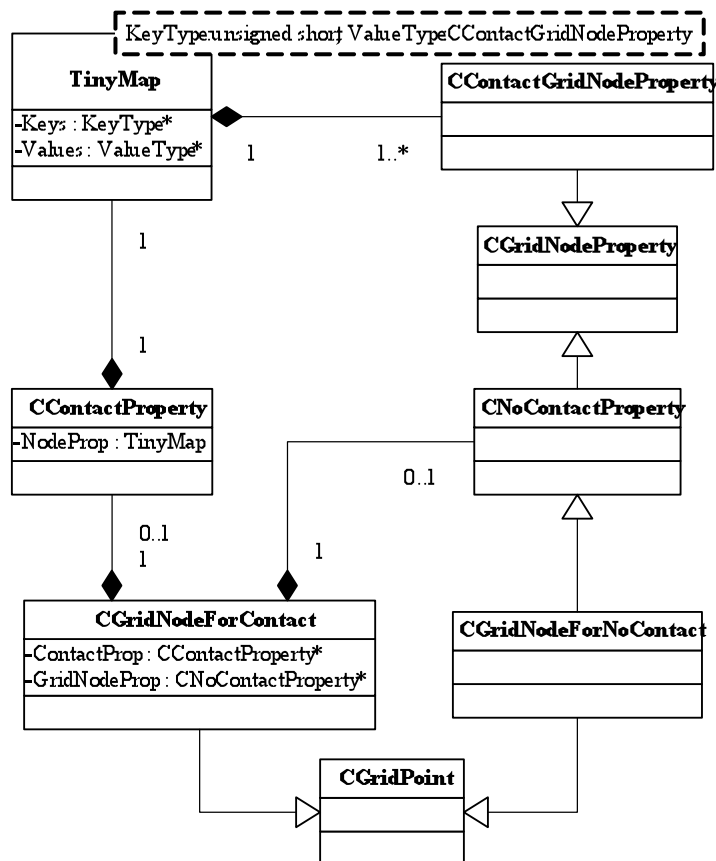


图7 节点类的引用和继承关系

3.6 流程优化

这一节中将对MPM3Dpp计算流程进行了优化，包含了对不同用于应力更新的节点速度更新的方式。优化的局部多重网格物质点法也将在下述中列出：

- i. 初始化所有节点和单元的指针为NULL。
- ii. 初始化已存在的节点或每隔指定的步数（在MPM3Dpp中缺省值为10）析构无用的节点和单元。
- iii. 如果单元中存在粒子，才创建它及其相应的节点。形成集中质量阵和节点动量；对于开启接触的情况，标记有粒子的节点和单元。
- iv. 这一步仅用于接触的情况。循环所有的网格节点，如果该节点存在并包含多个物体（即：`TinyMap`的大小大于1），创建并初始化物体的法向量数组，将该节点的序号记入到`ContactNodeList`数组中。循环所有的网格单元，如果有多余一个物体在网格中，则对它进行标记，记入`CellMayContact`数组。循环所有在可能发生接触的网格中的粒子，计算每个物体在节点处的法向。循环所有可能发生接触的节点，通过解除条件判断，如果某个节点处发生接触，则调整接触物

体的节点动量。

- v. 这一步仅用于USF方法。计算每个粒子的应变增量和旋量增量，求解本构方程、更新应力。
- vi. 计算已存在的节点的內力。
- vii. 这一步仅用于接触的情况。调整每个接触物体的节点力。
- viii. 更新已存在节点的动量。
- ix. 映射节点力和动量到粒子，以更新它们的位置和速度。
- x. 这一步仅用于MUSL方法。将新计算得到的粒子速度映射到网格节点更新节点动量。重新调整接触的节点的动量。
- xi. 这一步仅用于MUSL 和USL方法。基于在 $t^{n-1/2}$ 时间步的节点速度更新粒子应力(即第v步)。
- xii. 重新定义新的网格，如果必要，返回第ii步开始新的循环。

4 算例

在这一部分，用MPM3Dpp来模拟不同的问题来验证其框架的性能。面向对象软件架构将灵活性提升的同时也会影响执行效率和内存耗费量。为了研究新架构的影响，在4.1节中与MPM3D进行了对比。由于MPM3D只能处理两个物体，因此其他的算例只使用MPM3Dpp来计算。

所有的模拟都在相同的计算机上计算，其配置如下：

CPU: Intel CPU Q6600 2.4GHz

内存: 4G

操作系统: Microsoft windows XP

4.1 侵彻

为了比较执行效率和内存耗费量，对卵形杆弹斜侵彻铝靶进行了模拟^[44]。其中，弹体材料为钢4340，靶体材料为铝6061-T651，侵彻速度为575m/s，角度为30°，初始物质点离散如图8所示。弹体材料弹性材料来模拟，靶体采用John-Cook强度模型和Mie-Gruneisen状态方程来模拟。侵彻后的结果如图9所示。模拟的剩余速度为441m/s，实验结果为455m/s，结果一致。

为了比较MPM3Dpp和MPM3D的计算效率和内存耗费量，都使用相同的时间步长和求解算法(USF)，移动网格被禁用。模拟中总网格数为170688，总粒子数为200864。模拟中有粒子的网格占总网格的比例如图10所示，在整个计算过程中，最大的有用网格数量占总网格数量比值仅为13.13%。为了表明优化的效果，将MPM3Dpp和MPM3D的平均每步计算时间和最大内存耗费量及其比值都列在表1中。可以看出，与MPM3D

相比，效率得到了很大的提升，内存耗费用量降低了很多，这得益于对计算流程的优化、动态分配网格内存和粒子的额外属性。

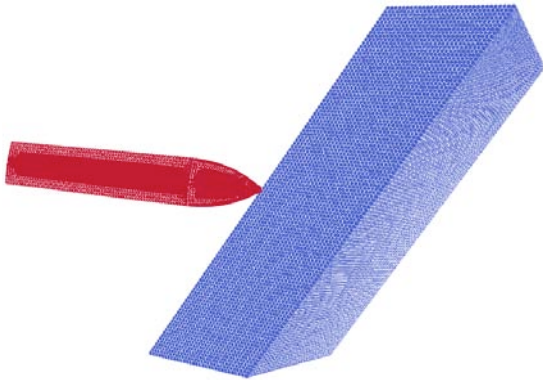


图8 斜侵彻物质点模型

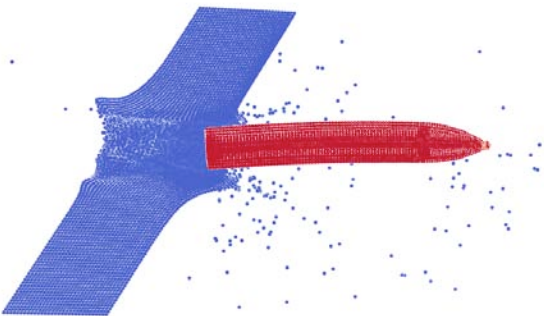


图9 侵彻后的结果

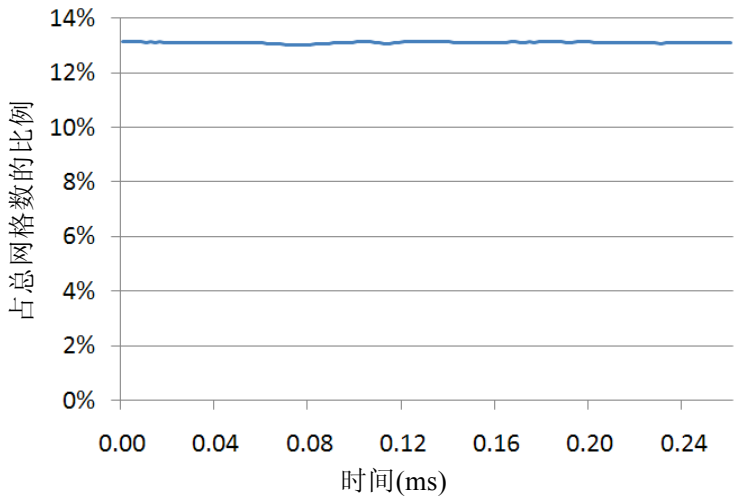


图10 有粒子的网格占总网格的比例随时间的变化

表1 侵彻模拟算例计算效率和内存耗费用量不同程序的比较

接触	平均每步时间(s)			最大内存耗费用量(M)		
	MPM3D	MPM3Dpp	比值	MPM3D	MPM3Dpp	比值
关闭	0.5455	0.4064	1.342	73.18	54.25	1.349
开启	0.8146	0.4313	1.888	109.36	55.02	1.988

4.2 平板压球

为了说明接触算法的模拟多个物体的能力，1000个铝球（以10×10×10在三维中排列）放置在一个容器中，一块平板置于这些球之上并受到均布载荷的作用（见图11）。在这个算例中，共有810个接触面，求解方法使用MUSL。

为了进行计算效率和内存耗费用量的考察，与关闭接触的结果进行了比较，见表2。

每步平均计算时间和最大内存耗用量都列在了这个表中。可以看出，打开接触后，耗费的额外CPU时间和内存并不多，虽然接触的物体很多。图12为打开接触情况下应力波传递、反射和耗散的过程。

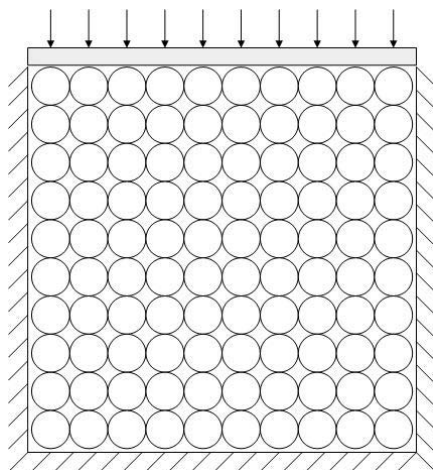


图11 平板压球平面示意图

表2 平板压球算例中的计算效率和内存耗用量

接触	粒子数	网格数	平均每步时间(s)	最大内存耗用量(M)
关闭	652000	137500	1.4263	288.60
打开			2.2187	305.19

4.3 圆盘晃动

为了验证MPM3Dpp模拟长时间加载的能力，对在周期性作用力和重力作用下的3个大圆盘和207个小圆盘的运动进行了模拟。这些圆盘放在一个固定不动的容器中。这个算例采用2维模拟，求解算法采用MUSL，开启接触算法。每个小圆盘和大圆盘分别离散为205和1821粒子，容器离散为4832个粒子。模拟进行了接近1,700,000步。在图13中，显示了这些在周期性作用力和重力作用下的圆盘的晃动，以及大圆盘被筛选出来。

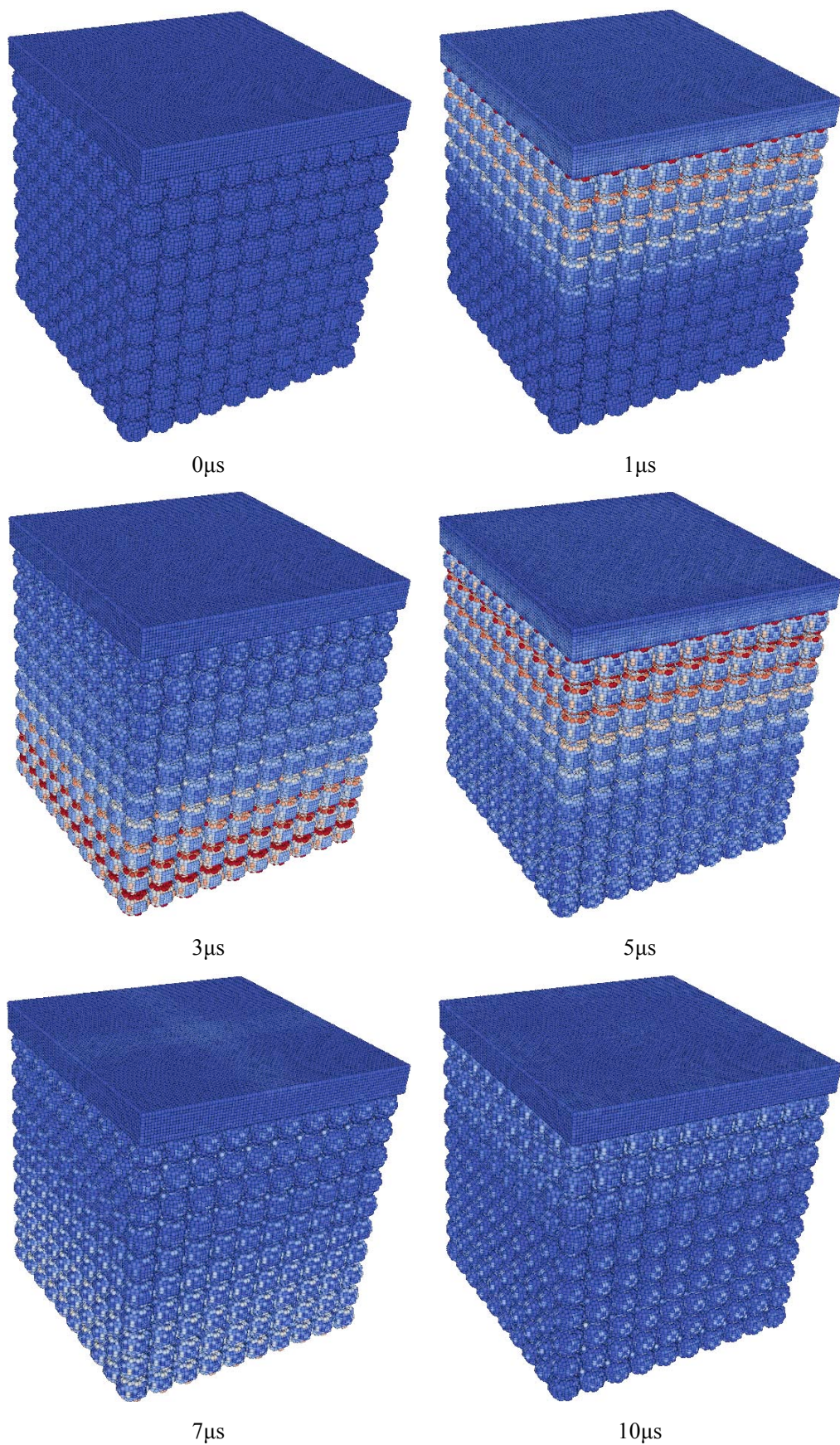


图12 开启接触情况下1000个球受压的时间历程

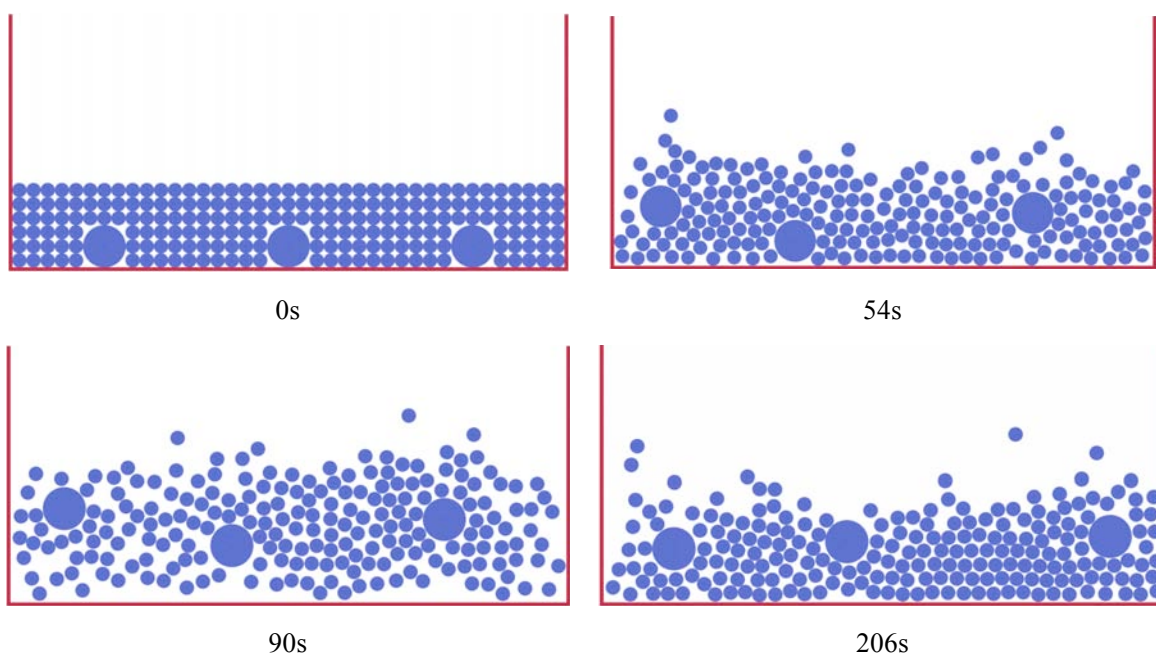


图13. 圆盘晃动模拟结果

4.4 圆环碰撞

这个算例用来验证提出的改进的接触检测算法。两个弹性圆环以60m/s的相对速度相撞，然后再弹开。每个圆环的内径、外径和厚度分别为30mm、40mm和10mm，都离散成29760个粒子。接触检测算法改进前和改进后的模拟结果分别如图14和图15所示。比较图14和图15，可以看出改进的接触检测算法能够避免圆环碰撞时产生的缝隙。

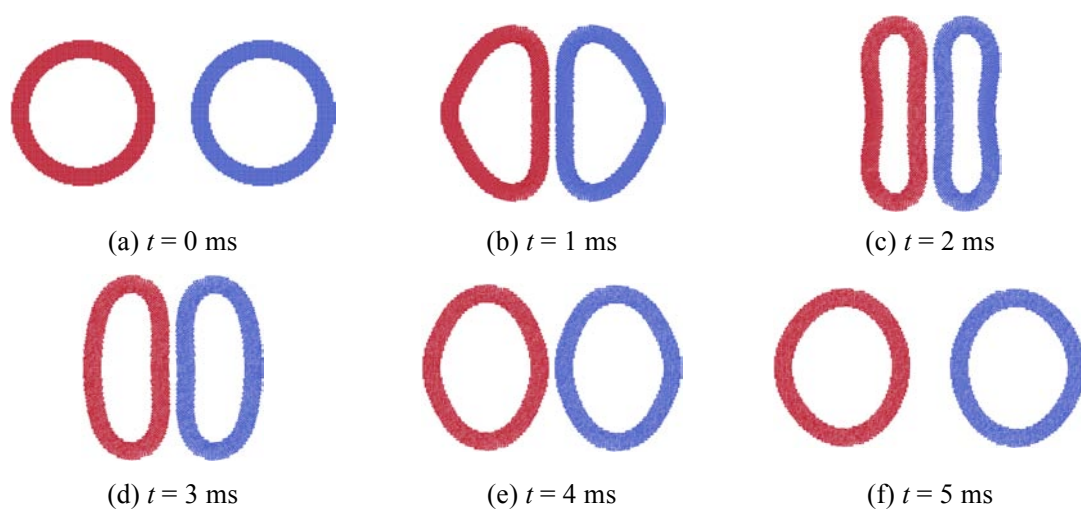


图14 采用改进前的接触检测方式的圆环碰撞结果

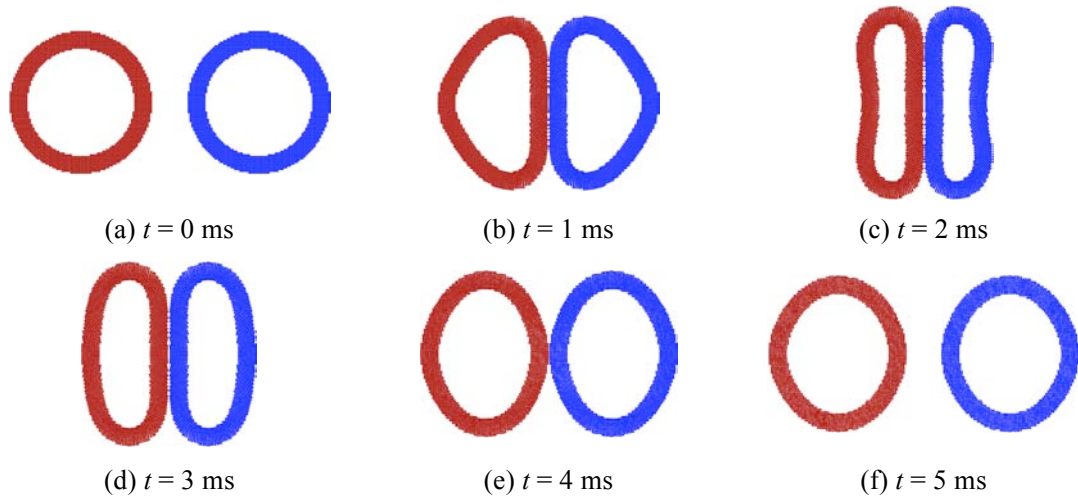


图15 采用改进后的接触检测方式的圆环碰撞结果

5 结论

提出了面向对象设计的MPM程序。其对于不同的MPM求解过程都具有较好的灵活性、可扩展性、可维护性，克服了面向过程程序的困难。对于修改或扩展程序，只需要了解需要修改的部分或基类的接口。通过使用继承，代码的重用性得到了提升。

改进的接触检测方式能够避免物体的提前接触。优化后，MPM3Dpp具有更高得效率和更低的内存耗费量。更新应力和应变的不同方式和接触算法都被整合到MPM3Dpp中。提出的局部多重网格只需要在重叠处创建，这样可以比全局多重网格需要更少的内存。动态创建材料内变量和动态创建网格的方法能够减少内存的耗费量。提出的移动网格的方法可以防止粒子飞出计算域外。

通过一些算例说明了程序架构的高效率和低内存耗费量，无论打开接触与否。为了验证改进的接触检测方式，对圆环碰撞的算例进行了模拟，改进后的算法能够避免圆环提前接触。通过这些算例验证了本文C++语言设计框架的成功实现，同时也说明给出了一个健壮的物质点法模拟程序。

参考文献

- [1] D. Sulsky, Z. Chen, H. Schreyer. A particle method for history-dependent materials. *Comput. Methods Appl. Mech. Engrg.* 1994; 118:179-196.
- [2] D. Sulsky, S. Zhou, H. Schreyer. Application of a particle-in-cell method to solid mechanics. *Computer Physics Communications* 1995; 87:236-252.
- [3] J.U. Brackbill and H.M. Ruppel, FLIP: A method for adaptively zoned particle-in-cell calculations in two dimensions. *J. Comput. Phys.* 65 (1986) 314-343.
- [4] J.U. Brackbill, D.B. Kothe, and H.M. Ruppel, FLIP: A low-dissipation, particle method for fluid flow. *Comput. Phys. Commun.* 48 (1988) 25-38.
- [5] Z. Wiechowski. The material point method in large strain engineering problems. *Comput. Methods Appl. Mech. Engrg.* 193 (2004) 4417-4438.
- [6] C.J. Coetzee, P.A. Vermeer, and A.H. Basson. The modelling of anchors using the material point method. *Int. J. Numer. Anal. Meth. Geomech.* 29 (2005) 879-895.
- [7] Hong-Lai Tan and John A. Nairn. Hierarchical, adaptive, material point method for dynamic energy release rate calculations. *Comput. Methods Appl. Mech. Engrg.* 191 (2002) 2095-2109.
- [8] Joris J. C. Remmers, Rene de Borst, and Alan Needleman. Simulation of Fracture in heterogeneous materials with the cohesive segments method. *Proceedings of VIII International Conference on Computational Plasticity, Barcelona* (2005).
- [9] Z. Chen, W. Hu, L. Shen, X. Xin, and R. Brannon. An evaluation of the MPM for simulating dynamic failure with damage diffusion. *Eng. Fract. Mech.* 69 (2002) 1873-1890.
- [10] Z. Chen, R. Feng, X. Xin, and L. Shen. A Computational Model for Impact Failure with Shear-Induced Dilatancy. *Int. J. Numer. Meth. Engrg.* 56 (2003) 1979-1997.
- [11] X. Zhang, K.Y. Sze, and S. Ma. An explicit material point finite element method for hyper-velocity impact. *Int. J. Numer. Meth. Engrg.* 66 (2006) 689-706.
- [12] S Ma, X Zhang, YP Lian et al. Simulation of high explosive explosion using adaptive material point method, *CMES: Computer Modeling in Engineering & Sciences*, 39(2): 101-123, 2009
- [13] A. R. York, D. Sulsky, and H. L. Schreyer. The material point method for simulation of thin membranes. *Int. J. Numer. Meth. Engrg.* 44 (1999) 1429-1456.
- [14] S.G. Bardenhagen, J.U. Brackbill, and D. Sulsky. The material-point method for granular materials. *Comput. Methods Appl. Mech. Engrg.* 187 (2000) 529-541.
- [15] S.J. Cummins and J.U. Brackbill. An implicit particle-in-cell method for granular materials. *J. Comput. Phys.* 180 (2002) 506-548.
- [16] S.G. Bardenhagen and J.U. Brackbill. Dynamic stress bridging in granular material. *J. Appl. Phys.* 83 (1998) 5732-5740.
- [17] S.G. Bardenhagen and J.U. Brackbill. Numerical study of stress distribution in sheared granular material in two dimensions. *Phys. Rev. E* 62 (2000) 3882-3890.
- [18] S.G. Bardenhagen, J.E. Guilkey. An improved contact algorithm for the material point method and application to stress propagation in granular material. *CMES* 2 (4) (2001) 509-522.
- [19] H.W. Zhang, K.P. Wang, Z. Chen. Material point method for dynamic analysis of saturated porous media under external contact/impact of solid bodies. *Comput. Methods Appl. Mech. Engrg.* 198 (2009) 1456-1472

- [20] S.G. Bardenhagen. Energy Conservation Error in the Material Point Method. *J. Comp. Phys.*, 180(2002) 383–403.
- [21] J. A. Nairn. Material Point Method Calculations with Explicit Cracks. *CMES*. 4 (6) (2003) 649-663.
- [22] W. Hu and Z. Chen. A multi-mesh MPM for simulating the meshing process of spur gears. *Computers and Structures* 81 (2003) 1991-2002.
- [23] X. F. Pan, A. G. Xu, G. C. Zhang, et al. Three-dimensional multi-mesh material point method for solving collision problems. *Commun. Theor. Phys.* 49 (2008) 1129-1138.
- [24] S. G. Bardenhagen and E. M. Kober. The Generalized Interpolation Material Point Method. *CMES: Computer Modeling in Engineering & Sciences*, 5(6): 477-495, 2004
- [25] Jin Ma, Hongbing Lu, and Ranga Komanduri. StructuredMesh Refinement in Generalized InterpolationMaterial Point (GIMP) Method for Simulation of Dynamic Problems. *CMES: Computer Modeling in Engineering & Sciences*, 12(3): 213-227, 2006
- [26] M. Steffen, P.C. Wallstedt, J.E. Guilkey, et al. Examination and Analysis of Implementation Choices within the Material Point Method (MPM). *CMES: Computer Modeling in Engineering & Sciences*, 31(2): 107-127, 2008
- [27] Luming Shen and Zhen Chen. A Silent Boundary Scheme with the Material Point Method for Dynamic Analyses. *CMES: Computer Modeling in Engineering & Sciences*. 7(3): 305-320, 2005
- [28] P. C.Wallstedt and J. E. Guilkey. Improved Velocity Projection for the Material Point Method, *CMES: Computer Modeling in Engineering & Sciences*. 19(3): 223-232, 2007
- [29] Z. Chen, Y. Gan and J.K. Chen. A Coupled Thermo-MechanicalModel for Simulating the Material Failure Evolution Due to Localized Heating, *CMES: Computer Modeling in Engineering & Sciences*. 26(2): 123-137, 2008
- [30] G.L. Fenves. Object-oriented programming for engineering software development. *Engineering with computers* 6 (1990) 1-15.
- [31] B.W.R. Forde, R.O. Foschi, S.F. Stierner. Object-oriented finite element analysis. *Computers and structures*. 34(1990) 355-374.
- [32] G.R.Miller, M.D. Rucki. A program architecture for interactive nonlinear dynamic analysis of structures. In: *Proc 5 Int Conf Comput Civ Build Engng V ICCCB*. New York: ASCE, 1993. p. 529-536.
- [33] Y. Duboisplerin, T. Zimmermann. Object-oriented finite element programming 3 an efficient implementation in C++. *Computer Methods in Applied Mechanics And Engineering*. 108 (1993) 165-183.
- [34] J.W. Baugh, D.R. Rehak. Data abstraction in engineering software development. *Journal of Computing in Civil Engineering* 6 (1992) 282-301
- [35] J. Lu, D.W. White, W.F. Chen, H.E. Dunsmore. A matrix class library in C++ for structural engineering computing. *Computers and Structures* 55 (1995) 95-111.
- [36] R. Chudoba, Z. Bittnar, P. Krysl. Explicit finite element computation: an object-oriented approach. In: Pahl, Werner, editors. *Computing in Civil and Building Engineering*. Rotterdam: Balkerna, 1995. p. 139-145.
- [37] G.C. Archer, G. Fenves, C. Thewalt. A new object-oriented finite element analysis program architecture. *Computers and Structures*. 70 (1999) 63-75.
- [38] L.C. Yu, A.V. Kumar. An object-oriented modular framework for implementing the finite element method. *Computers and Structures*. 79 (2001) 919-928.

- [39] S.M. Luo, Y.C. Cai, X.G. Zhang. Object-oriented element free Galerkin method. Chinese journal of mechanical engineering. 36 (2000) 23-26.
- [40] S.N. Atluri, S.P. Shen. The meshless local Petrov-Galerkin (MLPG) method. A simple & lesscostly alternative to the finite element and boundary element methods. CMES: Computer Modeling in Engineering and Sciences, 3 (1) 11-51, 2002
- [41] S.N. Atluri, H. T. Liu, Z. D. Han. Meshless Local Petrov-Galerkin (MLPG) Mixed Finite Difference Method for Solid Mechanics. CMES: Computer Modeling in Engineering & Sciences, 15(1): 1-16, 2006
- [42] X.F. Zhang, G. Subbarayan. jNURBS: An object-oriented, symbolic framework for integrated, meshless analysis and optimal design. Advances in engineering software. 37 (2006) 287-311
- [43] H. Peng, X. Zhang, S. Ma, H.K. Wang. Shared Memory OpenMP Parallelization of Explicit MPM and Its Application to Hypervelocity Impact. CMES: Computer Modelling in Engineering & Sciences. 38(2): 119-148, 2008
- [44] A. J. Piekutowski, M. J. Forrestal, K. L. Poormon, T. L. Warren. Perforation of aluminum plates with ogive-nose steel rods at normal and oblique impacts. Int. J. Impact Engng. 18 (1996) 877-887.

物质点接触算法改进与混凝土侵彻的数值模拟

摘 要

对物质点法改进的接触检测方式进行了稳定性和效果验证，对圆柱滚动和圆环碰撞两个算例进行了对比分析。发现改进的接触检测方式能够更稳定更好的模拟接触问题。采用改进的接触检测方式对钢弹丸正侵彻混凝土靶进行了模拟（采用HJC模型），弹体剩余速度与实验吻合较好。

关键词：物质点法，接触，混凝土，侵彻，HJC

Abstract

Validation and effect of improved contact detection method for MPM is implemented. Two examples of rolling simulation on an inclined plane and collision of two elastic rings are simulated and analyzed. The improved contact detection method is more stable, and the results with it are more accurate. A serial of concrete penetrations are simulated with the method (HJC concrete model is used), and residual velocity of projectile is agreed with experiment.

Key words: Material point method, Concrete, HJC, Impact.

目 录

摘 要.....	26
Abstract.....	27
目 录.....	28
1 引 言	29
2 MPM方法	29
3 算例与分析	30
3.1 圆柱滚动	30
3.2 圆环碰撞	31
4 HJC混凝土模型	36
5 混凝土侵彻算例	37
6 结 论	39
参考文献.....	40

1 引言

混凝土在民用建筑及军事设施和防护工程中都得到了较为广泛的应用。数值模拟由于及经济性与高效性成为了混凝土侵彻问题的一种重要研究手段,通过模拟可以对侵彻过程进行更深入的了解。

有限元法是模拟混凝土侵彻问题的常用方法,但在许多大变形问题中会将发生扭曲畸变的单元侵蚀或进行网格重划分,从而导致计算精度严重降低。近些年来,无网格法备受关注,其将连续体离散为质点,可以避免单元侵蚀和网格重划分的问题^[1,2]。PIC^[3]法采用拉格朗日质点和欧拉网格的双重描述,这种思想对计算方法的发展产生了深远的影响。FLIP^[4,5]法在PIC法基础上将流体的所有属性都赋予质点,发展成了一种完全的拉格朗日质点格式。Sulsky^[6,7]等人将扩展后的FLIP法改称为物质点法(Material Point Method, MPM)。物质点法通过形函数构成质点和网格的映射关系,在处理爆炸冲击等大变形问题时具有较好的效果。它避免了欧拉法因非线性对流项产生的数值困难,并易于跟踪物质界面。物质点携带了连续体的所有物质信息,因此在每个时间步结束时抛弃变形后的网格,从而避免了拉格朗日因网格畸变产生的数值困难。

接触是工程领域非常广泛的一种现象。因为物质点法更新位置时采用单一速度场映射,所以无滑移接触在传统物质点法中自动处理,而不需要查找接触面。Bardenhagen等人^[8,9]扩展了传统物质点法,将滑动及带有摩擦的接触引入到物质点法中,在接触节点处使用Coulomb摩擦。物体间的接触力通过节点相对速度来计算。

本文采用Holmquist-Johnson-Cook(HJC)模型来模拟混凝土的行为。该模型考虑了静水压力效应、应变率效应和材料损伤软化,是一种常用的混凝土模型。当不同物体发生滑动接触时,MPM在界面上会发生携带效应,因此在计算侵彻问题时需要采用接触算法。

2 MPM方法

MPM法采用拉格朗日质点和欧拉网格双重描述,将连续体离散为一组质点。物质信息均由质点携带,动量方程通过网格求解。在每一个时间步中,质点和网格完全固连。网格节点的运动方程可以通过将质点的运动量映射到网格得到,求解后再将网格节点的运动量映射回质点,从而得到下一时刻的运动量。求解动量方程时,物质点和网格一起运动,因此可以通过建立在网格节点上的有限元形函数来实现质点和网格之间的信息映射。

当两个物体相互接近时,其运动收到约束是必要的。当两个物体的动量映射在同一节点处时,接触就可能会发生。Bardenhagen^[8,9]提出了两种接触检测,但都会引

起提前接触，本报告中《物质点法面向对象设计》一文对改进的接触检测方式进行了描述。

3 算例与分析

为了验证改进的的接触检测方式，进行了两组算例分析。

3.1 圆柱滚动

采用3维来模拟一圆柱在斜面上滚动，模型如图1所示。斜面倾斜角度与水平面成 θ 角，重力加速度 g 竖直向下取值为 10m/s^2 。圆柱的半径和长度分别是 3.2m 和 4m ，斜面长度为 20m 。初始粒子间距为 0.2m ，网格宽度为 0.4m 。选择适当的材料参数使得时间步长较大。圆柱的弹性模量和密度分别取为 42MPa 和 1000kg/m^3 。斜面的弹性模量和密度分别取为 420MPa 和 $10,000\text{kg/m}^3$ 。

当 $\tan\theta > 3\mu$ 时，圆柱在斜面上既滚又滑，否则为纯滚动。对于刚性圆柱在斜面上滚动存在解析解，其中心位置 $x_{\text{cm}}(t)$ 作为时间的函数，形式为

$$x_{\text{cm}}(t) = \begin{cases} x_0 + \frac{1}{2}|g|t^2(\sin\theta - \mu\cos\theta), & \tan\theta > 3\mu \quad (\text{slip}) \\ x_0 + \frac{1}{3}|g|t^2\sin\theta, & \tan\theta \leq 3\mu \quad (\text{stick}) \end{cases} \quad (1)$$

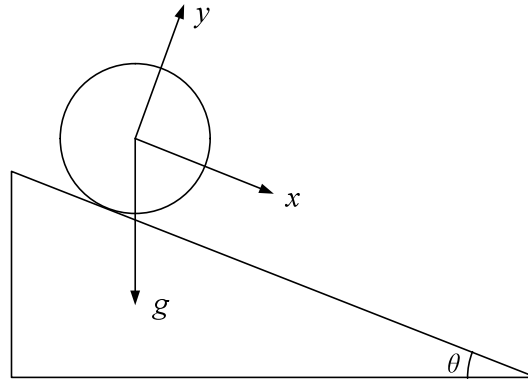
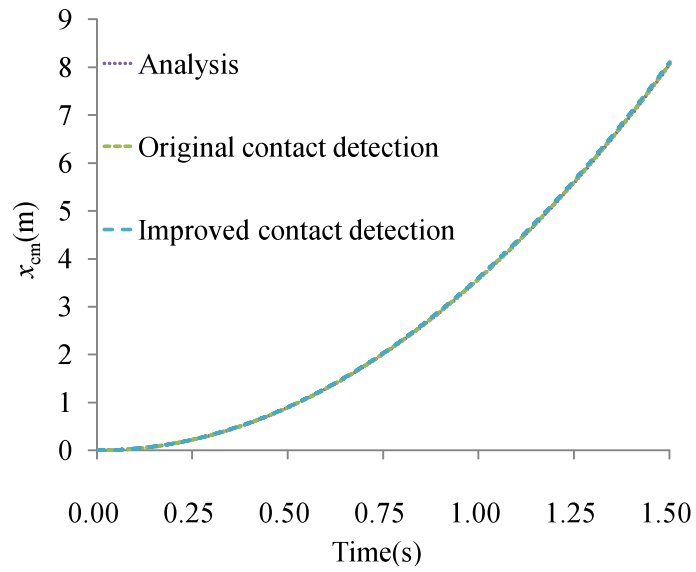
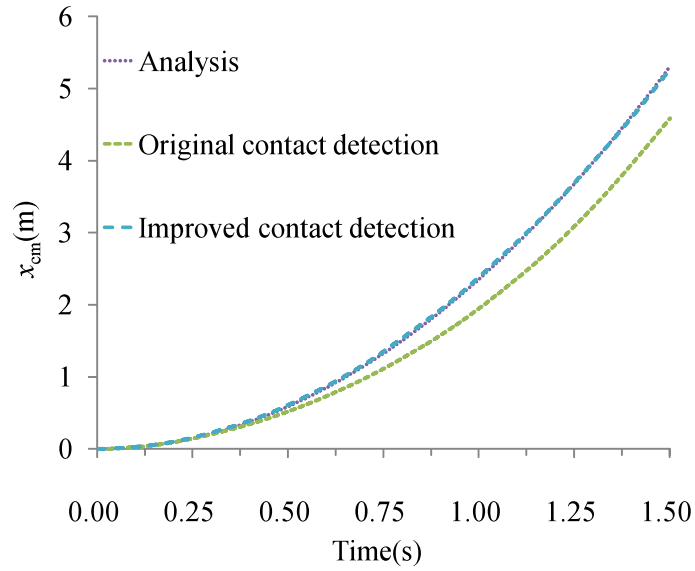


图1 圆柱沿斜面滚动示意图

计算结果如图2所示，图2(a)为既滚又滑的情况，解析解、接触检测方式改进前和改进后的曲线都重合。图2(b)为纯滚动的情况，改进后的接触检测方式得到的结果与解析解吻合，但改进前的差别较大。图3为不同时刻的接触检测方式改进前和改进后的模拟结果的对比，从中可以看出纯滚动时，如采用原先的接触检测方式圆柱会与斜面有缝隙，这可能由于圆柱部分粒子与斜面提前接触会产生力矩，使得圆柱离开斜面。当粒子和网格尺寸变小时，采用原有的接触检测方式的结果会趋近于解析解，改进的接触检测方式具有更小的网格粒子尺寸依赖性。



(a) 既滚又滑



(b) 纯滚动

图2 圆柱中心的位移随时间的变化

3.2 圆环碰撞

为了检验物质点法不同流程的收敛性，对弹性圆环碰撞进行了模拟。圆环碰撞的相对速度为60m/s，它们的内径、外径和厚度分别为30mm、40mm和10mm，分别都采用21991个粒子模拟。圆环的体积模量、剪切模量和密度分别为121.7MPa、26.1MPa和1.01g/cm³。摩擦系数设置为0.1，初始粒子间距为1mm，网格宽度为2mm。

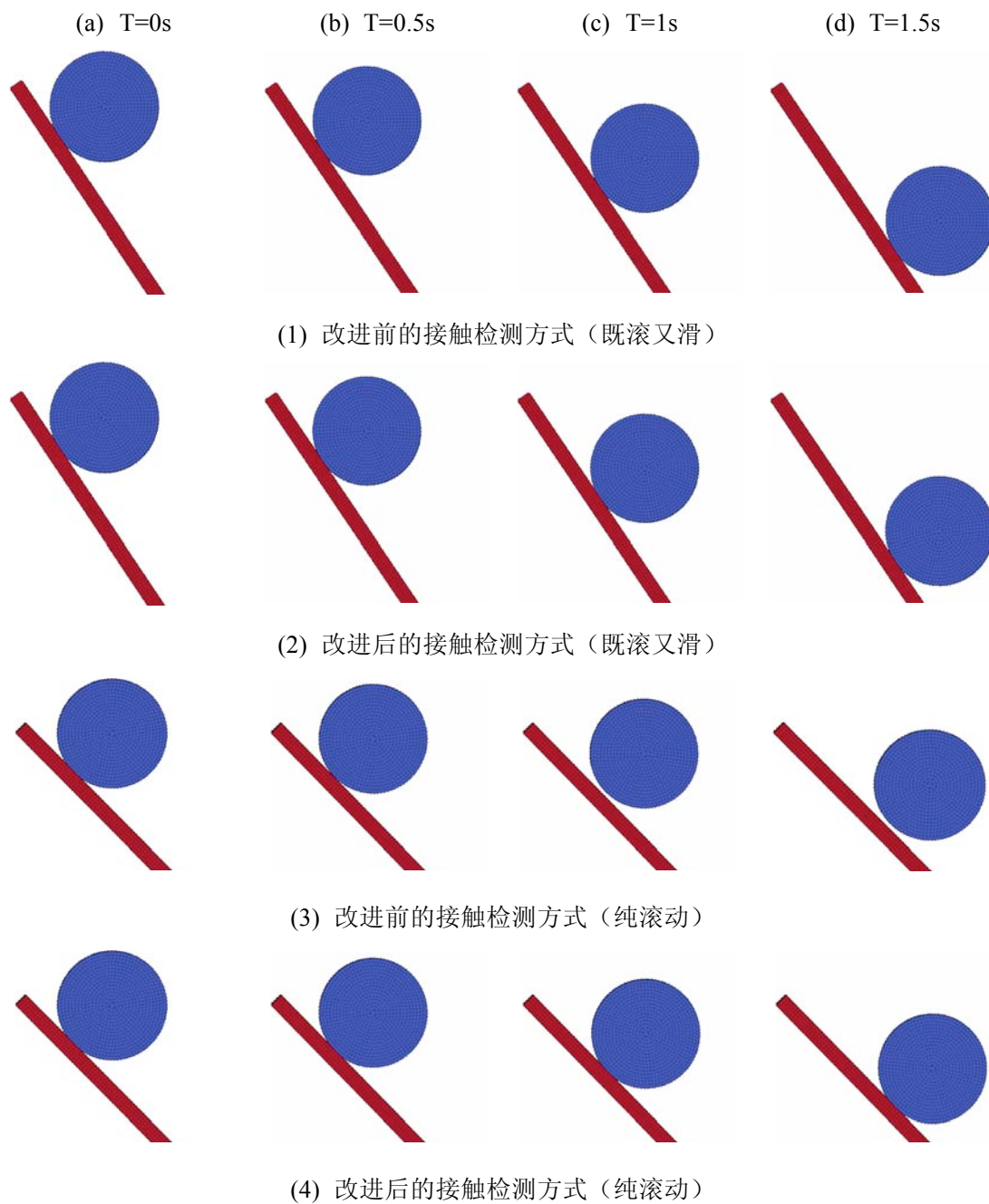
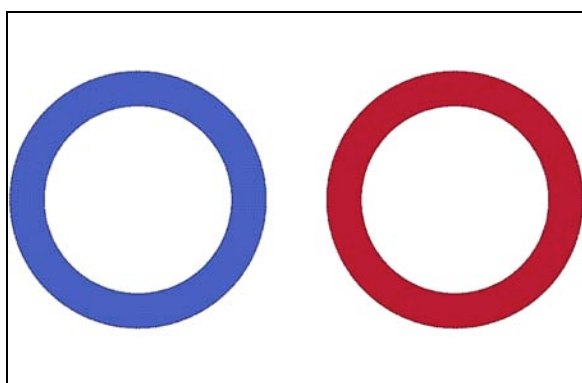
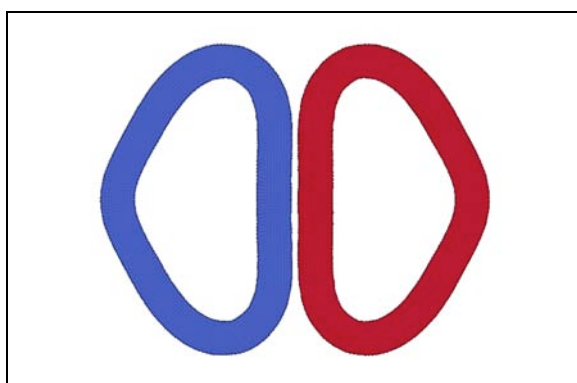


图3 球滚动模拟(MUSL, $\alpha=0.6$)

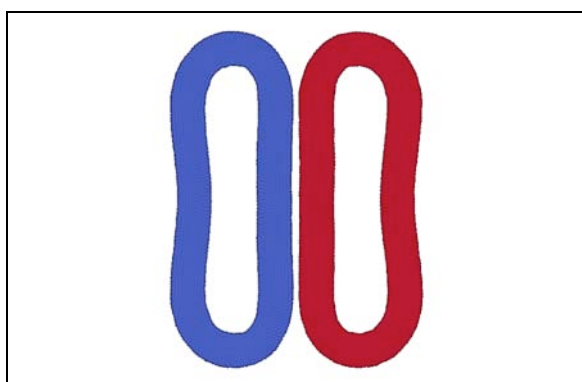
计算结果如图4和图5所示,可以看出改进后的接触检测方式避免了圆环碰撞时存在的缝隙。为了检验时间因子对误差的影响,进行了一系列数值模拟,结果曲线显示在图6中,可以看出,无论是MUSL还是USF,改进的接触算法都具有更小的误差,具有更好的收敛性。



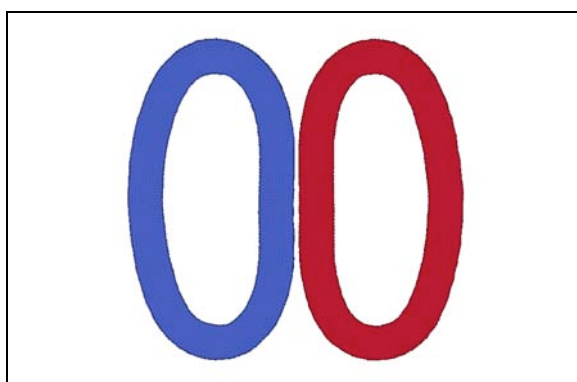
(a) $T=0\text{ms}$



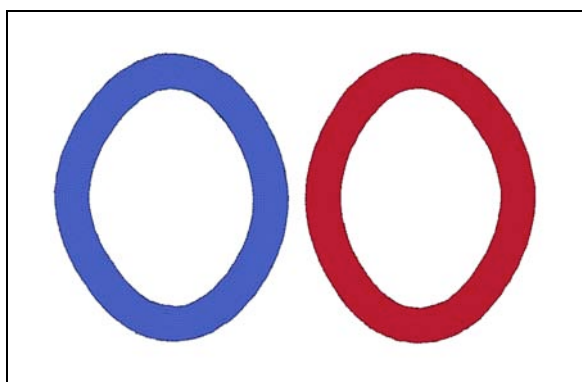
(b) $T=1\text{ms}$



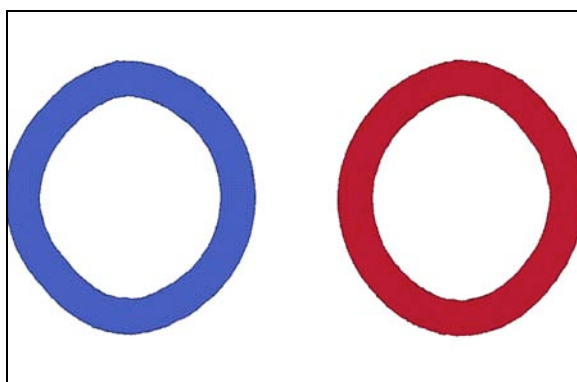
(c) $T=2\text{ms}$



(d) $T=3\text{ms}$

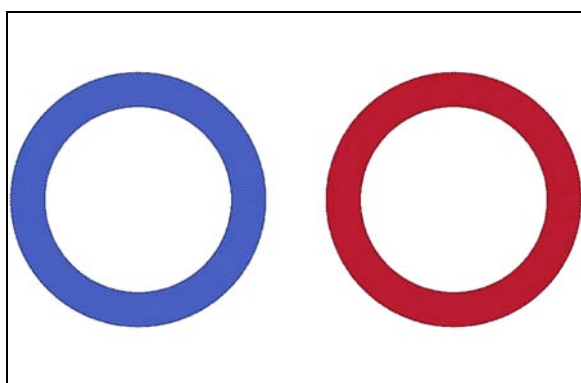


(e) $T=4\text{ms}$

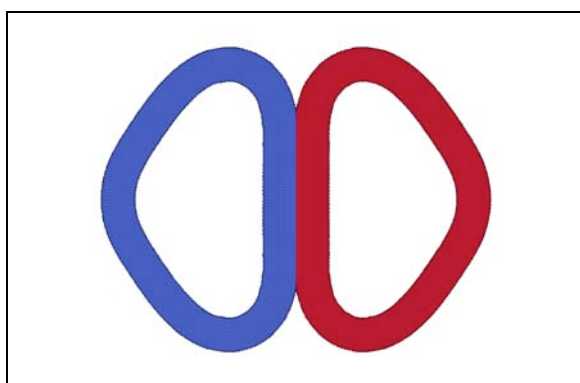


(f) $T=5\text{ms}$

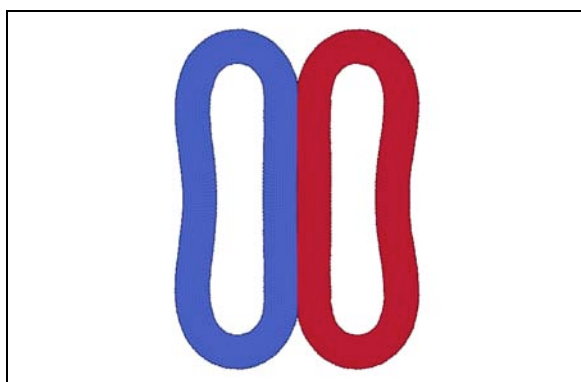
图4 采用原有的接触检测方式弹性圆环碰撞(MUSL, $\alpha=0.3$)



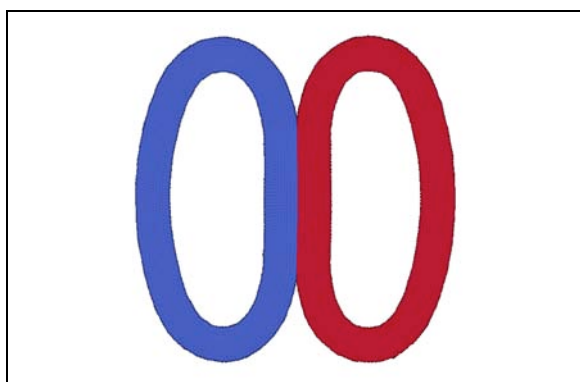
(a) $T=0\text{ms}$



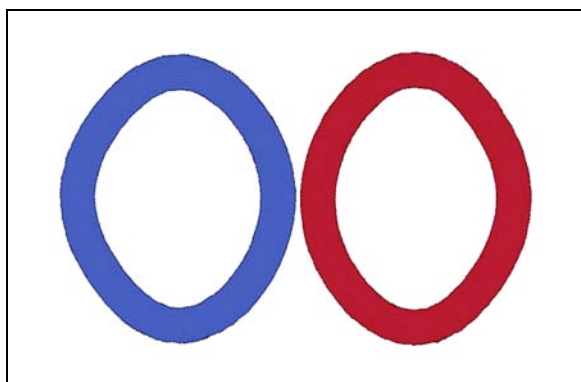
(b) $T=1\text{ms}$



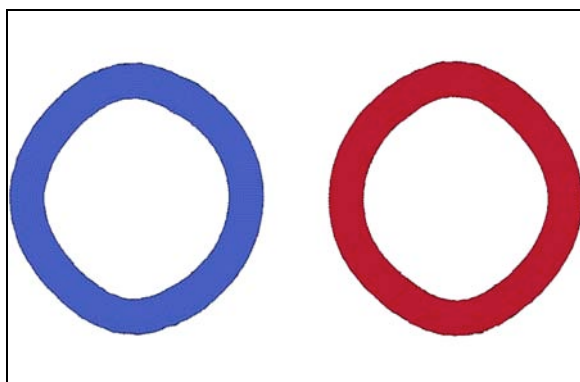
(c) $T=2\text{ms}$



(d) $T=3\text{ms}$

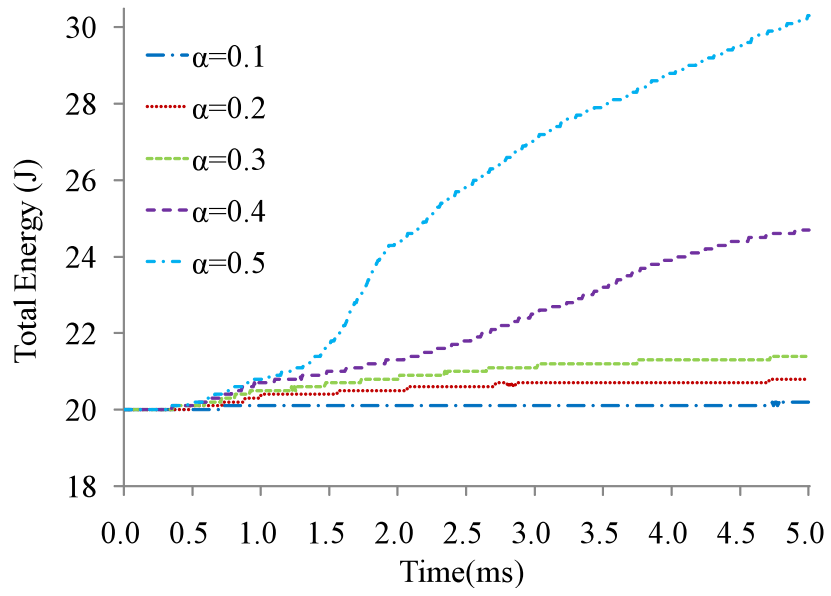


(e) $T=4\text{ms}$

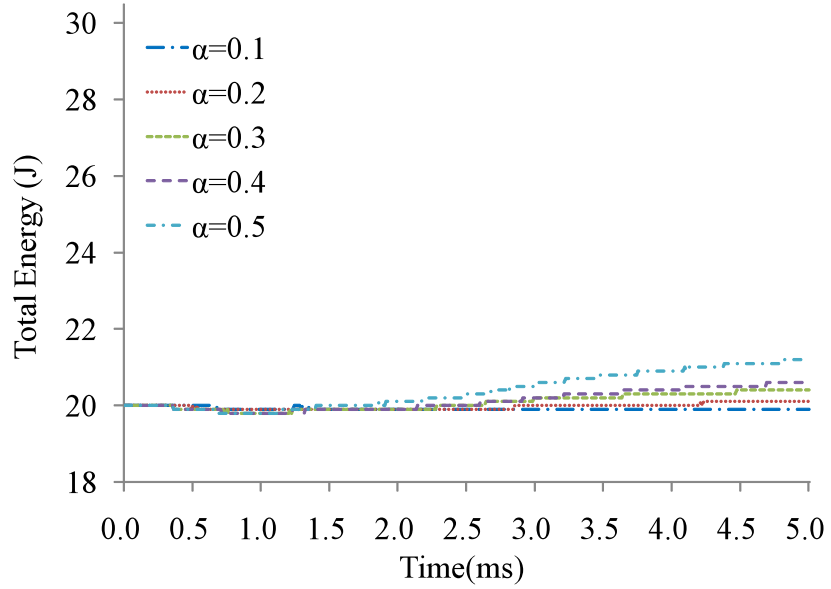


(f) $T=5\text{ms}$

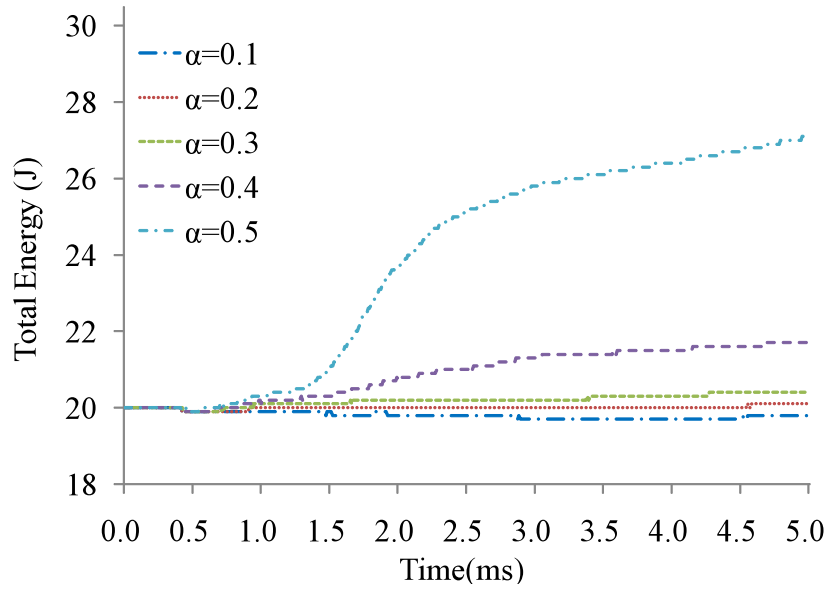
图5 采用改进的接触检测方式弹性圆环碰撞(MUSL, $\alpha=0.3$)



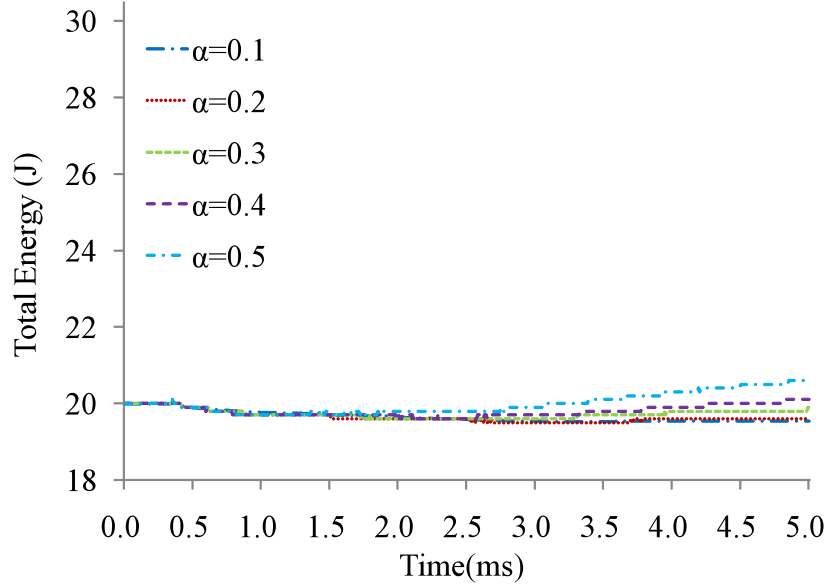
(a) USF, 原有的接触检测方式



(b) MUSL, 原有的接触检测方式



(c) USF, 改进的接触检测方式



(d) MUSL, 改进的接触检测方式

图6 不同时间步长因子情况下总能量随时间的变化

4 HJC混凝土模型^[10]

HJC模型中, 规范化屈服应力 (如图7a所示) 定义为

$$\sigma^* = [A(1-D) + BP^{*N}][1 + C \ln \dot{\epsilon}^*] \quad (4)$$

其中, A 、 B 、 C 、 N 为材料常数, $\sigma^* = \sigma / f_c$ 为无量纲应力, $P^* = P / f_c$ 为无量纲静水压力, f_c 为静态单轴抗拉强度, $\dot{\epsilon}^* = \dot{\epsilon} / \dot{\epsilon}_0$ 为无量纲应变率, D 为损伤因子 ($0 \leq D \leq 1$)。

由上式可知, σ^* 随着 P^* 的增长而无穷增长, 为此引入一个无量纲强度 $S_{\max}$, $\sigma^* \leq S_{\max}$ 。

损伤因子 D (如图7b所示) 为等效塑性应变引起的损伤和塑性体积应变引起的损伤之和

$$D = \sum \frac{\Delta \varepsilon_p + \Delta \mu_p}{\varepsilon_p^f + \mu_p^f} \quad (5)$$

其中， $\Delta \varepsilon_p$ 和 $\Delta \mu_p$ 为一个计算循环内的等效塑性应变增量和塑性体积应变增量， $\varepsilon_p^f + \mu_p^f$ 是在压力 P 下破碎的塑性应变

$$\varepsilon_p^f + \mu_p^f = D_1 (P^* + T^*)^2 \quad (6)$$

其中， D_1 和 D_2 为材料常数， $T^* = T/f_c$ ， T 为混凝土的抗拉强度。

HJC模型的状态方程分为三段：线性段、过渡段和压实段（如图7c所示）。混凝土在各阶段受拉时服从的规律和该阶段的卸载规律类似，但须满足 $p \leq p_{\max} = T(1-D)$ 。

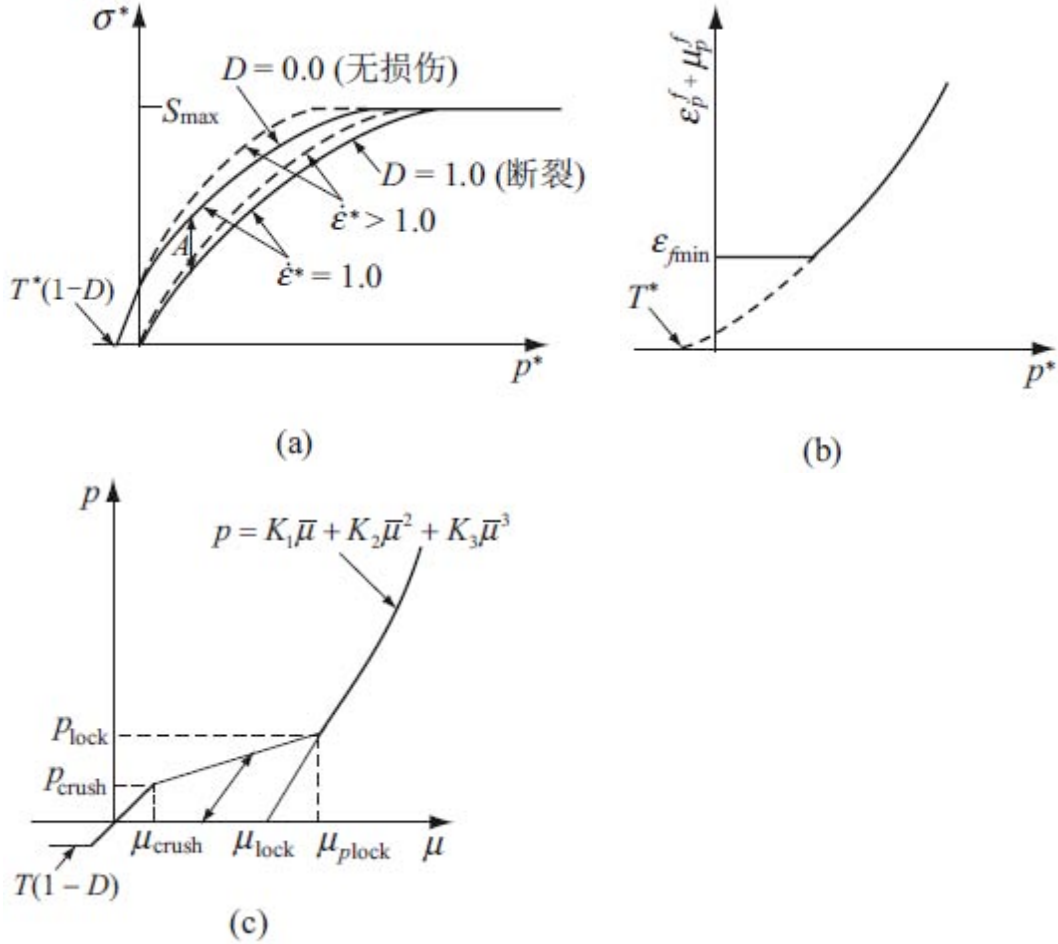


图7 HJC混凝土模型

5 混凝土侵彻算例

Hanchak^[11]等人对卵形杆弹侵彻混凝土靶进行了一系列的实验，下面将对其进行数值模拟并进行比较。弹体结构如图8所示，弹头曲径比为3，混凝土靶厚度为178mm。混凝土采用HJC模型，其材料参数取自文献[10]，弹体采用弹性材料，杨氏模量取为212.42GPa。粒子间距为1.78mm，网格宽度为3.56mm，总数量约为293万。

弹体初速度为1058m/s时的侵彻如图9所示，从中可以清晰的看到混凝土的辐射状裂纹。图10给出了不同初速度的弹体侵彻混凝土后的剩余速度的模拟与实验结果。从图中可以看出，模拟结果与实验结果吻合较好，高速时略有差别。

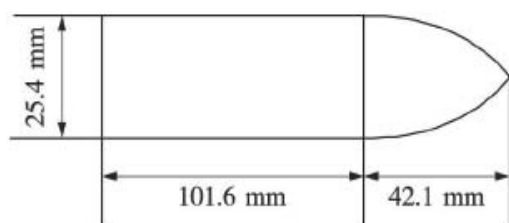


图8 弹体结构

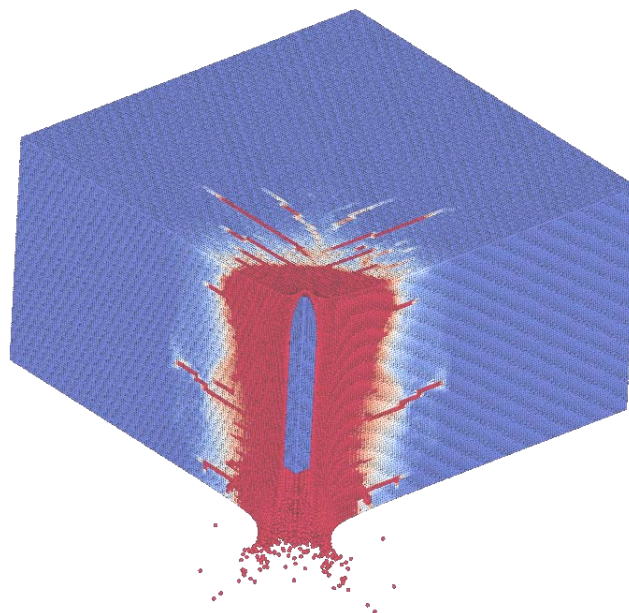


图9 弹体初速度为1058m/s时的混凝土侵彻

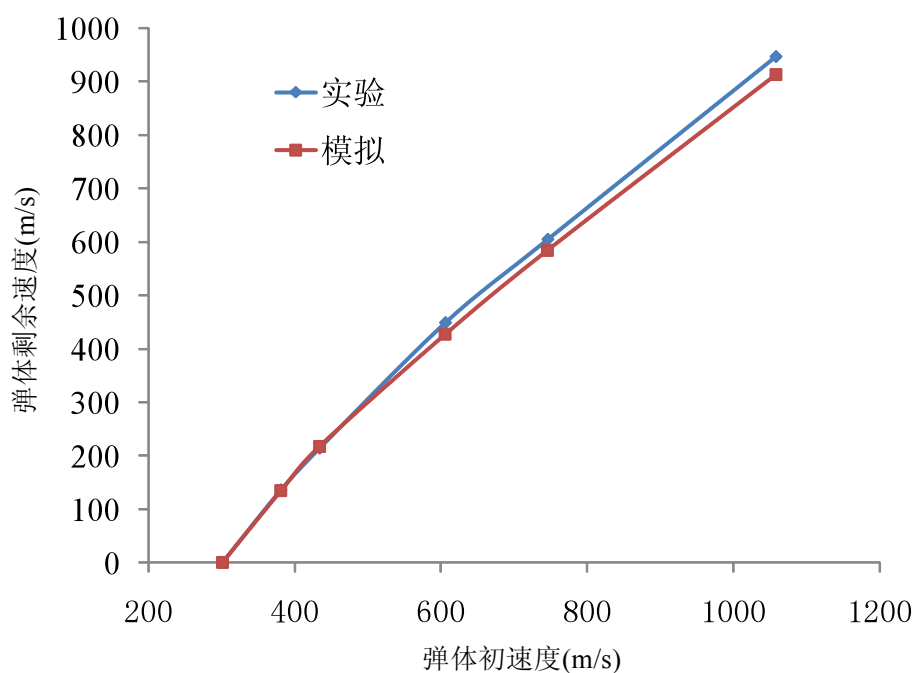


图10 剩余速度模拟值与实验值的比较

6 结论

通过圆柱滚动和圆环碰撞两个算例的模拟，说明了改进的接触检测方式具有更好的模拟效果和收敛性。将其应用于混凝土的模拟取得了较好的效果，剩余弹速与实验结果吻合较好。

参考文献

- [1] D. Sulsky, Z. Chen, H. L. Schreyer. A particle method for history-dependent materials [J]. Comput. Meth. Appl. Mech. Eng. 1994, 118: 179-186
- [2] D. Sulsky, H. K. Schreyer. Axisymmetric form of the material point method with applications to upsetting and Taylor impact problems [J]. Comput. Meth. Appl. Mech. Eng., 1996, 139: 409-429
- [3] F. H. Harlow. The particle-in-cell computing method for fluid dynamics. Methods in Computational Physics, 1964, 3:319-345
- [4] J. U. Brackbill, H. M. Ruppel. FLIP: a method for adaptively zoned, particle-in-cell calculations in two dimensions. Journal of Computational Physics, 1986, 65:314-343
- [5] J. U. Brackbill, D. B. Kothe, H. M. Ruppel. FLIP: a low-dissipation, particle-in-cell method for fluid flow. Computer Physics Communications, 1988, 48:25-38
- [6] D. Sulsky, Z. Chen, H. L. Schreyer. A particle method for history-dependent materials. Computer Methods in Applied Mechanics and Engineering, 1994, 118(1-2):179-196
- [7] D. Sulsky, S. J. Zhou, H. L. Schreyer. Application of a particle-in-cell method to solid mechanics. Computer Physics Communications, 1995, 87(1-2):236-252
- [8] S.G. Bardenhagen and J.U. Brackbill. Numerical study of stress distribution in sheared granular material in two dimensions. Phys. Rev. E 62 (2000) 3882-3890.
- [9] S.G. Bardenhagen, J.E. Guilkey. An improved contact algorithm for the material point method and application to stress propagation in granular material. CMES 2 (4) (2001) 509–522.
- [10] T. J. Holmquist, G.. R. Johnson. A computational constitutive model for concrete subjected to large strains, high strain rates and high pressure. 14th International Symposium on Ballistics, 1993.
- [11] S. J. Hanchak, M. J. Forrestal, E. R. Young, et al. Perforation of concrete slabs with 48 MPa and 140 MPa unconfined compressive strengths. Int. J. Impact Eng, 1992, 12(1):127

个人简历

1979年11月生于哈尔滨，1998年考入哈尔滨工业大学航天科学与力学系本硕连读，2002-2008年硕博连读。硕博期间编写了光滑质点流体动力学代码，并进行了改进，用于模拟超高速空间碎片防护的问题。对不同泡沫铝防护结构进行了实验和模拟，进行了防护结构优化分析。2008年7月开始在清华大学做博士后，从事物质点法在侵彻冲击领域的研究。

博士后期间发表的学术论文，专著

1. Z.T. Ma, B.J. Pang, B. Jia, X. Zhang. A fast and stable SPH method. ICCES Special Symposium on Meshless Methods, October 2008
2. 马志涛, 张雄. 混凝土侵彻MPM数值模拟. 中国力学大会2009
3. Z.T. Ma, X. Zhang, P. Huang. An object-oriented MPM framework for simulation of large deformation and contact of numerous grains. CMES, 2010, 55:61-87