

目录

Using Nesterov's method to
accelerate multibody dynamics
with friction and contact

1. INTRODUCTION

2. BACKGROUND

2.1 Modeling Aspects

2.2 Numerical Solution Methodology

2.3 Convexification Artifacts

2.4 Embedding a Simple Cohesion Model

3. THE ACCELERATED PROJECTED GRADIENT DESCENT METHOD

3.1 Preamble: Nesterov's Method

3.2 The Overall Algorithm

4. NUMERICAL EXPERIMENTS

4.1 APGD Performance Analysis

4.2 Other APGD Simulations

5. DISCUSSION

6. CONCLUSIONS AND DIRECTIONS OF FUTURE WORK

APGD论文

Using Nesterov's method to accelerate multibody dynamics with friction and contact

1. INTRODUCTION

对于一个接触行为：接触前没力，接触时突然有力。

如果我们用传统的微分方程（ODE）工具（如 Runge-Kutta）模拟，就无法处理这个不连续的“跳变”。

以下是我的解释关于为什么ODE不可以使用：

系统的右端函数 $f(x)$ 不连续，违反ODE方法的基本前提条件。

常规ODE为例：

$$\frac{dt}{dx} = f(x, t)$$

小时间步 Δt 内，系统行为是**平滑可预测的**，但是接触力是**非连续的**，在 $y=0$ ：接触力突然变成 $f_c > 0$ ，所以是一种跳跃，传统积分器没办法解决。所以使用**非光滑动力学（non-smooth dynamics）建模方式**，也就是下面的两种方法

摩擦接触建模的两大类方法

在**多体动力学中处理摩擦接触问题**的方法很多，但可以粗略分为两大类：

- **基于罚函数法（Penalty Method）或正则化（Regularization）的方法**
 - 核心思想是：通过允许刚体在接触点“轻微穿透”，再通过某种“力函数”或“弹性系数”来恢复该穿透
 - 这种方法简单、易实现

e.g:

$$F_{contact} = -k \cdot \delta$$

你把这个力项当作**外力**，加进动力学微分方程：

$$my'' = -mg + F_{contact}$$

就可以直接用 Runge-Kutta 等传统 ODE 工具求解了。

罚函数法的原理与挑战

罚函数法（PM）的基本假设：

- 刚体在接触点处**可以轻微变形**（例如压入一点）
- 通过这种“局部变形”计算出作用力

现实中要精确模拟这类局部变形必须使用有限元法（FEA），代价很高，所以一般采取一种近似方式：

- 在碰撞检测阶段，**根据刚体相交的深度来估算变形**
- 通过一些经典力学模型（如 Hertz 接触理论）来代替真实变形，来计算接触力

FEA通过**将复杂的物体划分成许多小而简单的单元（elements）**，并在每个单元上建立局部方程，最终拼成一个大系统来近似整个物体的行为。这样会出现大量的计算，所以使用惩罚函数来近似这个值。

虽然罚函数方法简单易用，但在实际应用中面临三大挑战：

(i) 模型参数难以确定

- 如刚度、阻尼系数、摩擦系数等，这些参数很难准确获得

(ii) 数值稳定性受限

- 罚函数法引入的“刚性力”可能让仿真器在时间步长上受限，甚至数值发散

(iii) 力的随机性

- 同样的问题参数设置导致了接触力的不一致和不稳定，带来不可预期的模拟误差

“虽然罚函数法使用广泛，但由于它依赖近似变形模型、数值参数敏感以及物理真实性受限，因此许多研究者转向了更数学化的DVI建模方式。”

2. 基于“非光滑动力学”的方法

- 以 Filippov (1967) 的理论为基础，引入“微分变分不等式”(DVI, Differential Variational Inequalities)
- 最终形成一个 **带平衡约束与互补条件** 的数值优化问题

（也就是我们之前所学的多体动力学MBD）

DVI（Differential Variational Inequalities）方法介绍：

DVI 方法是处理刚体摩擦接触问题的一种现代数值方法。它通过构建 **非线性互补问题（NCP）** 来求解接触力，核心思想是：

- 用互补条件（Complementarity）来**限制法向力和速度不能同时为正**
- 采用牛顿法或 Fischer-Burmeister 函数来逼近求解器

非线性互补问题（Nonlinear Complementarity Problem, NCP）

很难求解的 DVI 中的 **非线性互补问题（NCP）**，通过“凸松弛”处理成更容易数值求解的 **凸互补问题（CCP）**

CCP 是对原始物理建模（NCP）的数值松弛与结构优化。

它保留了物理意义，但转换成了一个**优化器更易处理、大规模可扩展**的形式，是现代接触模拟系统（如 Chrono）背后的核心算法思想。

APGD (Accelerated Projected Gradient Descent) 加速投影梯度下降法。

基于 Nesterov 提出的加速梯度方法，解决的是一种带圆锥约束的二次优化问题。
具体来说，在每个仿真时间步，它都要解一个优化问题来求解系统中的法向力与摩擦力。

与传统的Gauss-Seidel方法相比，它大大加快了模拟速度，可以将计算时间缩短 1到2个数量级（即10到100倍更快）。

QP (Quadratic Programming, 二次规划) 问题是一类优化问题，其目标函数是二次函数，约束条件是线性函数。

DVI 中求解摩擦接触的互补问题，其实等价于一个带约束的优化问题 (Quadratic Program)。APGD 就是为了高效地解决这个优化问题。

2. BACKGROUND

??? : the set of generalized coordinates used to position and orient 为什么这里说的是广义坐标系，而这种表达方法不应该是最大坐标系吗??? 存疑!

- $\mathbf{r}_j$: 提供刚体质心在**全局坐标系**中的绝对位置;
- ϵ_j : 作为欧拉参数，四元数，描述刚体在**全局坐标系**中的朝向 (方向)。

$$\mathbf{q} = [\mathbf{r}_1^T, \epsilon_1^T, \dots, \mathbf{r}_{n_b}^T, \epsilon_{n_b}^T]^T \in \mathbb{R}^{7n_b}$$

从广义的速度向量到笛卡尔速度向量

$$\dot{\mathbf{q}} = [\dot{\mathbf{r}}_1^T \quad \dot{\epsilon}_1^T \quad \dots \quad \dot{\mathbf{r}}_{n_b}^T \quad \dot{\epsilon}_{n_b}^T]^T$$

广义速度

$$\mathbf{v} = [\dot{\mathbf{r}}_1^T \quad \boldsymbol{\omega}_1^T \quad \dots \quad \dot{\mathbf{r}}_{n_b}^T \quad \boldsymbol{\omega}_{n_b}^T]^T$$

新的速度变量

用V相比Q可以从7维降到6维度，并且四元数和角速度有着对应的关系，四元数求导计算公式里面是存在w这项的，可以通过这个推导出来。

$$\begin{aligned}\dot{\epsilon} &= \frac{1}{2} \mathbf{G}^T(\epsilon) \bar{\omega} \\ \bar{\omega} &= 2 \mathbf{G}(\epsilon) \dot{\epsilon}\end{aligned}$$

$G(\epsilon)$: 是一个从四元数导数到角速度的转换矩阵;

定义一个变换矩阵 $L(q)$, 它把新速度变量 v 映射回传统的广义速度:

$$\mathbf{L}(\mathbf{q}) = \text{diag} \left[\mathbf{I}_{3 \times 3}, \frac{1}{2} \mathbf{G}^T(\epsilon_1), \dots, \mathbf{I}_{3 \times 3}, \frac{1}{2} \mathbf{G}^T(\epsilon_{n_b}) \right] \in \mathbb{R}^{7n_b \times 6n_b}$$

$$\dot{\mathbf{q}} = \mathbf{L}(\mathbf{q})\mathbf{v}$$

2.1 Modeling Aspects

聚焦于两个刚体之间接触点的局部几何建模, 是后续构建接触力、摩擦力、约束条件等的基础。

考虑两个刚体 A 和 B 之间的接触:

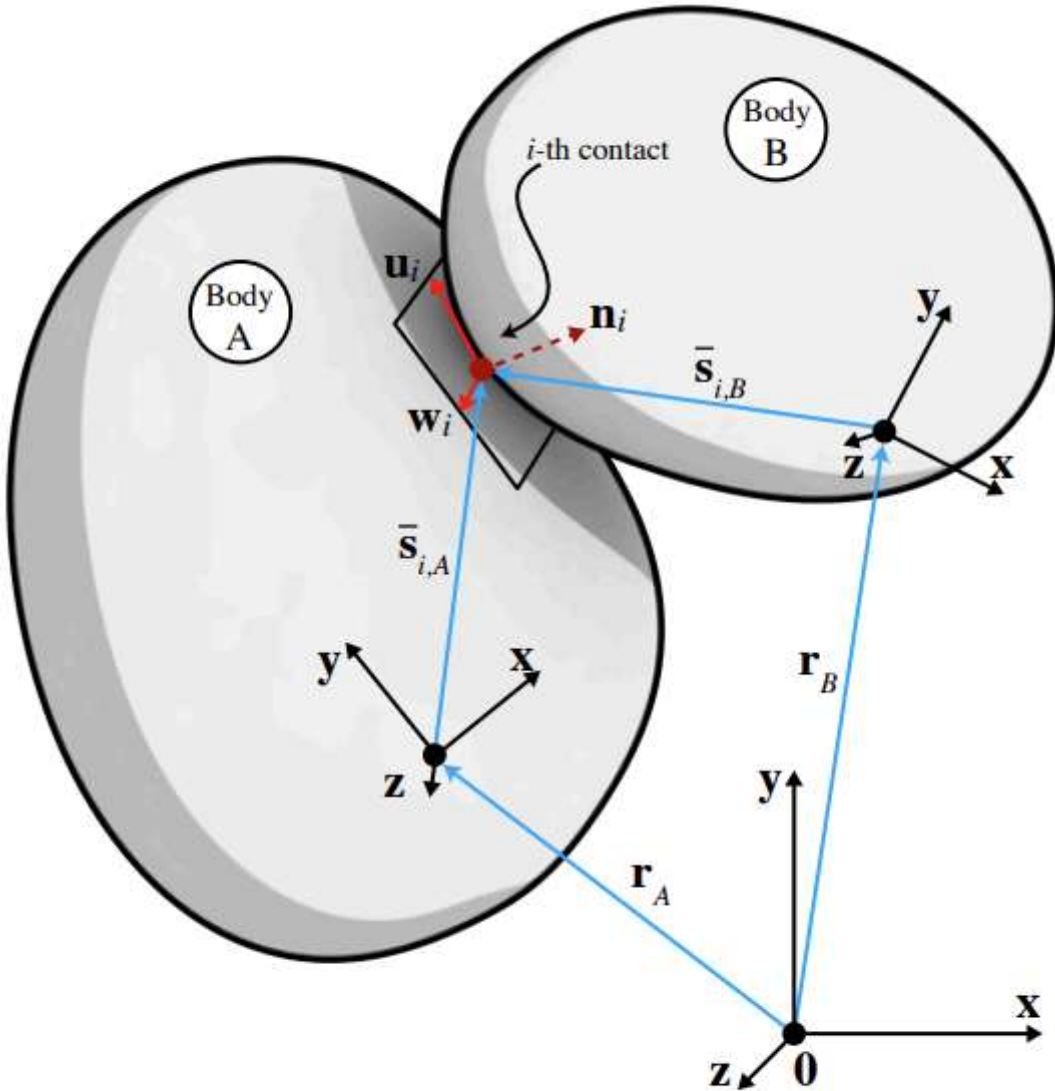


Fig. 1. Contact i between two bodies $A, B \in \{1, 2, \dots, n_b\}$.

- 假设接触点的几何光滑（无奇异点）；
- 使用接触点 + 共用切平面来为每个物体构造**局部坐标系**。

每个刚体都需要一个局部坐标系来表示接触点上的：

- 法向（normal）
- 切向（tangent1, tangent2）
- 摩擦锥也基于此构建

刚体 A 在接触点 i 上的法向 $\mathbf{n}_{i,A}$ 是垂直于切平面、指向物体外部的单位向量。这和我们在摩擦力建模中常说的“法向接触力”方向一致。

选择两个正交单位向量 $\mathbf{u}, \mathbf{w}$ ，分别作为切向基底；

与法向 $\mathbf{n}$ 组成一个**右手坐标系**（用于摩擦建模），即局部接触坐标系为

$$\mathbf{T}_i = [\mathbf{n}_{i,A}, \mathbf{u}_{i,A}, \mathbf{w}_{i,A}]$$

刚体 B 也按类似方法构造自己的接触坐标系，包含：

$$\mathbf{n}_{i,B}, \mathbf{u}_{i,B}, \mathbf{w}_{i,B} \in \mathbb{R}^3$$

接触力与互补性约束

物理量	意义
$\hat{\gamma}_{i,n}$	接触法向拉格朗日乘子 = 法向接触力
$\Phi_i(\mathbf{q})$	接触间隙函数（gap function）
$0 \leq \gamma \perp \Phi \leq 0$	接触力和间隙互补，即：

- 有间隙 $\Rightarrow$ 没力
- 有力 $\Rightarrow$ 无间隙

如果接触几何是光滑、凸形的（例如球面、平面），则可以使用我们之前所学过的关于接触到 MBD 刚体建模。

加入互补限制，可以使其成为一个互补问题，使摩擦力的解唯一。

库伦摩擦建模：

对于第 i 个接触点，接触力分为：

法向力：

$$\mathbf{F}_{i,N} = \hat{\gamma}_{i,n} \mathbf{n}_i$$

切向摩擦力：

$$\mathbf{F}_{i,T} = \hat{\gamma}_{i,u} \mathbf{u}_i + \hat{\gamma}_{i,w} \mathbf{w}_i$$

其中：

- $\hat{\gamma}_{i,n} > 0$ 表示接触点 i 的法向乘子，控制法向力大小；
- $\mathbf{n}_i$ 是接触点的单位法向量；
- $\mathbf{u}_i, \mathbf{w}_i$ 是切向单位基；
- $\hat{\gamma}_{i,u}$ 和 $\hat{\gamma}_{i,w}$ 是切向方向的摩擦力分量。

摩擦力建模：库仑摩擦模型的三条核心条件：

条件1：摩擦力大小不超过摩擦锥（friction cone）

$$\sqrt{\hat{\gamma}_{i,u}^2 + \hat{\gamma}_{i,w}^2} \leq \mu \hat{\gamma}_{i,n}$$

条件2：摩擦力与切向速度满足互补关系

$$\|\mathbf{v}_{i,T}\| \left(\sqrt{\hat{\gamma}_{i,u}^2 + \hat{\gamma}_{i,w}^2} - \mu \hat{\gamma}_{i,n} \right) = 0$$

条件3：滑动方向与摩擦力方向相反

$$\langle \mathbf{F}_{i,T}, \mathbf{v}_{i,T} \rangle = -\|\mathbf{F}_{i,T}\| \|\mathbf{v}_{i,T}\|$$

内积等于负模长乘积 $\Rightarrow$ 两个向量方向完全相反

$\mathbf{v}_{i,T}$ ：两个刚体在第 i 个接触点处的**切向相对速度**，滑动就意味着这个值不为零。

切向摩擦力的优化表达（KKT 条件）

$$(\hat{\gamma}_{i,u}, \hat{\gamma}_{i,w}) = \arg \min_{\sqrt{y^2 + z^2} \leq \mu_i \hat{\gamma}_{i,n}} \mathbf{v}_{i,T}^T (y \mathbf{u}_i + z \mathbf{w}_i)$$

让摩擦力在切向速度方向上“做的功”最小（即最大抵消滑动）；库仑摩擦条件的**KKT一阶条件的几何等价形式**。

（KKT 条件是对 **拉格朗日乘子法**的推广，用来判断一个约束优化问题的候选解是否为最优解（必要条件）。）

总接触力的向量表达：对于第 i 个接触点，我们可以把接触点的总作用力由两部分组成法向力 $F_{i,N}$ 和切向摩擦力 $F_{i,T}$

$$\mathbf{F}_i = F_{i,N} \mathbf{n}_i + \mathbf{F}_{i,T}$$

力分解表达式：

$$\mathbf{F}_{i,T} = \hat{\gamma}_{i,n} \mathbf{n}_i + \hat{\gamma}_{i,u} \mathbf{u}_i + \hat{\gamma}_{i,w} \mathbf{w}_i$$

摩擦锥定义：

接触力必须在一个以 $\mathbf{n}_i$ 为轴的圆锥形集合内（即库仑摩擦锥）；

$$\Upsilon_i = \left\{ [x, y, z]^T \in \mathbb{R}^3 \mid \sqrt{y^2 + z^2} \leq \mu_i x \right\}$$

非光滑动力学框架下的**刚体动力学建模核心公式**：

1.运动学关系（从广义速度计算广义坐标导数）：

$$\dot{\mathbf{q}} = \mathbf{L}(\mathbf{q}) \mathbf{v}$$

2.Newton-Euler 动力学方程（含接触力）：

$$\mathbf{M} \dot{\mathbf{v}} = \mathbf{f}(t, \mathbf{q}, \mathbf{v}) + \sum_{i \in \mathcal{A}(\mathbf{q}, \delta)} (\hat{\gamma}_{i,n} \mathbf{D}_{i,n} + \hat{\gamma}_{i,u} \mathbf{D}_{i,u} + \hat{\gamma}_{i,w} \mathbf{D}_{i,w})$$

$\mathcal{A}(\mathbf{q}, \delta)$ ：所有在 δ 距离内的潜在接触点集合

3.接触约束的互补性表达式（normal）：

$$0 \leq \hat{\gamma}_{i,n} \perp \Phi_i(\mathbf{q}) \geq 0$$

$\Phi_i(\mathbf{q})$ ：第 i 个接触点的“间隙函数”；表示法向接触力与间隙互补：有间隙 $\Rightarrow$ 无力，有力 $\Rightarrow$ 无间隙

4.切向摩擦力的投影最小化形式（Coulomb）：？

$$(\hat{\gamma}_{i,u}, \hat{\gamma}_{i,w}) = \arg \min_{\sqrt{y^2 + z^2} \leq \mu_i \hat{\gamma}_{i,n}} \mathbf{v}^T (y \mathbf{D}_{i,u} + z \mathbf{D}_{i,w})$$

接触 Jacobian ($\mathbf{D}_i$) 定义：

$$\mathbf{D}_i = [\mathbf{D}_{i,n}, \mathbf{D}_{i,u}, \mathbf{D}_{i,w}] \in \mathbb{R}^{6n_b \times 3}$$

维度：每个刚体有 3 个线速度分量 + 3 个角速度分量 $\Rightarrow$ 一共 6 个广义速度；

接触 Jacobian 的转置：

$$\mathbf{D}_i^T = [\cdots \quad -A_{i,p}^T \quad A_{i,p}^T A_A \tilde{\mathbf{s}}_{i,A} \quad \cdots \quad A_{i,p}^T \quad -A_{i,p}^T A_B \tilde{\mathbf{s}}_{i,B} \quad \cdots] \in \mathbb{R}^{3 \times 6n_b}$$

这个有点小疑问

$A_{i,p} = [\mathbf{n}_i, \mathbf{u}_i, \mathbf{w}_i]$ （由法向和两个正交切向组成的局部接触坐标系）

$A_A = \mathbf{A}(\boldsymbol{\epsilon}_A), \quad A_B = \mathbf{A}(\boldsymbol{\epsilon}_B)$ （由刚体 A/B 的姿态四元数生成的旋转矩阵）

$\tilde{\mathbf{s}}_{i,A}, \tilde{\mathbf{s}}_{i,B} \in \mathbb{R}^{3 \times 3}$ （接触点到刚体质心的位移向量的反对称矩阵形式，用于计算叉

$\mathcal{A}(\mathbf{q}, \delta)$ （系统状态 $\mathbf{q}$ 下，所有间距小于 δ 的潜在接触点索引集合）

2.2 Numerical Solution Methodology

DVI (Differential Variational Inequality) 方法的数值求解策略

联立求解速度、接触冲量与位置更新，以模拟多体刚体系统中的**摩擦接触动力学**。

整体思路概览：

给定当前位置 $\mathbf{q}(l)$ 和速度 $\mathbf{v}(l)$ ，我们要求出：

- 下一步速度 $\mathbf{v}(l+1)$
- 接触冲量 γ_i
- 下一步位置 $\mathbf{q}(l+1)$

满足：**动量守恒 + 接触互补约束 + 摩擦锥约束**

步长为 h

$$\underbrace{\mathbf{M} \left(\mathbf{v}^{(l+1)} - \mathbf{v}^{(l)} \right)}_{\text{动量变化}} = \underbrace{h \cdot \mathbf{f}^{(l)}}_{\text{广义外力作用}} + h \underbrace{\sum_{i \in \mathcal{A}(\mathbf{q}^{(l)}, \delta)} (\gamma_{i,n} \mathbf{D}_{i,n} + \gamma_{i,u} \mathbf{D}_{i,u} + \gamma_{i,w} \mathbf{D}_{i,w})}_{\text{接触冲量 (单位: } N \cdot s)} \quad (5)$$

冲量部分，也就是接触力和极短时间的乘积。

不穿透条件 (non-penetration constraint) 的离散化 + DVI (变分不等式) 建模：

公式6计算流程：

$$\text{gap函数 (距离): } \Phi_i(\mathbf{q}(t)) \geq 0$$

对其求导，得到**速度约束**：

$$\dot{\Phi}_i(\mathbf{q}) = \frac{d}{dt} \Phi_i(\mathbf{q}) = \nabla \Phi_i(\mathbf{q})^T \dot{\mathbf{q}} = \mathbf{D}_{i,n}^T \mathbf{v}(t) \geq 0$$

对两边对时间 t 求导（我们假设 Φ_i 对 $\mathbf{q}$ 可导）：

$$\frac{d}{dt}\Phi_i(\mathbf{q}(t)) = \nabla\Phi_i(\mathbf{q})^T \cdot \frac{d\mathbf{q}}{dt} = \nabla\Phi_i(\mathbf{q})^T \cdot \dot{\mathbf{q}}$$

注意：

- $\nabla\Phi_i(\mathbf{q})$ 是 Φ_i 关于广义坐标 $\mathbf{q}$ 的导数；
- $\dot{\mathbf{q}}$ 是广义速度；
- 而由于我们使用广义速度 $\mathbf{v}$ ，有：

$$\dot{\mathbf{q}} = \mathbf{L}(\mathbf{q}) \cdot \mathbf{v}$$

因此可写为：

$$\dot{\Phi}_i(\mathbf{q}) = \nabla\Phi_i(\mathbf{q})^T \mathbf{L}(\mathbf{q}) \cdot \mathbf{v}$$

我们定义：

$$\mathbf{D}_{i,n} := \mathbf{L}(\mathbf{q})^T \nabla\Phi_i(\mathbf{q}) \quad \Rightarrow \quad \dot{\Phi}_i = \mathbf{D}_{i,n}^T \mathbf{v}$$

这里其实和之前学的MBD中对约束函数求导，得到J雅可比矩阵，在这里是一样的，所以进行了替换。

距离和力互补

使用半隐式欧拉得出下一个时间的位置，也需要互补：

对位置约束 $\Phi_i(\mathbf{q}) \geq 0$ 进行时间离散，使用半隐式欧拉：

$$\Phi_i^{(l+1)} \approx \Phi_i^{(l)} + h \cdot \mathbf{D}_{i,n}^T \mathbf{v}^{(l+1)} \geq 0$$

变换一下得到：

$$\frac{1}{h}\Phi_i^{(l)} + \mathbf{D}_{i,n}^T \mathbf{v}^{(l+1)} \geq 0$$

得到了公式6：

$$0 \leq \frac{1}{h}\Phi_i(\mathbf{q}^{(l)}) + \mathbf{D}_{i,n}^T \mathbf{v}^{(l+1)} \perp \gamma_i^n \geq 0 \quad (6)$$

公式 (7) – 摩擦锥投影（变分不等式）：

$$(\gamma_{i,u}, \gamma_{i,w}) = \arg \min_{\sqrt{y^2+z^2} \leq \mu_i \gamma_{i,n}} \mathbf{v}^T (y\mathbf{D}_{i,u} + z\mathbf{D}_{i,w}) \quad (7)$$

公式 (8) – 位置更新（半隐式欧拉法）：

$$\text{位置更新: } \mathbf{q}^{(l+1)} = \mathbf{q}^{(l)} + h \cdot \mathbf{L}(\mathbf{q}^{(l)}) \cdot \mathbf{v}^{(l+1)} \quad (8)$$

$\mathbf{L}(\mathbf{q})$: 是将广义速度 $\mathbf{v}$ 映射为 $\dot{\mathbf{q}}$ 的变换矩阵;

冲量 $\gamma_{i,s}$ 的定义与单位换算

$$\gamma_{i,s} = h \cdot \hat{\gamma}_{i,s}, \quad s \in \{n, u, w\}$$

$$\hat{\gamma}_{i,s}: \text{单位是 } N \text{ (力)};$$

h 是步长。

上标 $(l+1)$ 被省略的约定, 为了简洁, 我们默认所有 γ 都是**下一步 $(l+1)$ 时刻**的值;

系统中其他力项统一记为 $\mathbf{f}$ 。

$$\mathbf{f} \equiv \mathbf{f}(t^{(l)}, \mathbf{q}^{(l)}, \mathbf{v}^{(l)})$$

稳定项的目的:

$$\frac{1}{h} \underbrace{\Phi_i(\mathbf{q}^{(l)})}_{\substack{\text{gap 函数:} \\ \text{第 } i \text{ 接触点的距离} \\ \text{(单位: m)}}} \quad \text{单位: m/s}$$

- 这是在互补约束中加入的**位置误差项**;
- 目的: **消除数值穿透** (eliminating any penetration);
 - 作用方式: 通过类似 Baumgarte 的稳定器, 在一个时间步内“强制恢复”非穿透。

大规模接触问题难以直接求解:

- 表明在工程仿真中, 直接求解带互补约束的系统 (方程 5-7) 是**不可扩展的**;
- 所以作者提出**第三个决策 (D3)**: 将其转化为更适合数值计算的形式 (比如 QP、LCP、APGD 等);

[Anitescu and Hart 2004]提出把公式 (6) 中的 **互补条件** 转换为一个**凸形式 (convexified NCP)**。

原始的接触互补条件 (非光滑动力学 DVI) 中, 引入摩擦速度项后的凸近似互补约束,

摩擦锥是非凸的, 数值求解中难以进行梯度下降或投影, 所以需要对他进行近似, 使其变成一个凸问题,

绕开冲量空间 → 转入速度空间

不如直接在“相对速度空间”中构造一个等价的约束，把摩擦的几何限制投影到法向速度上，用一个**速度相关的摩擦项**代替摩擦锥。

原始无摩擦形式是：

$$0 \leq g_n := \frac{1}{h} \Phi_i(q^{(l)}) + \mathbf{D}_{i,n}^T \mathbf{v}^{(l+1)} \quad \perp \quad \gamma^n \geq 0$$

这个 g_n 表示：**法向接触速度 + 渗透修正项**

当你有一个**很大的切向速度** $\mathbf{v}_T$ ，说明物体在滑动，**实际接触的有效性就变弱**。

 **但如果滑动速度 $\|\mathbf{v}_T\|$ 很大呢？**

- 表明两物体在接触点处正在显著滑动；
- 摩擦力已处于 **动摩擦极限状态**；
- 法向接触力可能已经不再是**纯粹的支撑力**，而变得“无效”或“脱离真实接触”。

这时我们在数值模型中就不能再简单地认为“只要有摩擦锥，法向力就能正常起作用”。

让法向速度“为摩擦所惩罚”，

即：**只有当切向速度够小，法向速度才能为 0，冲量才会生效**

于是就加了：

$$-\mu_i \|\mathbf{v}_T\| = -\mu_i \sqrt{(\mathbf{v}^T \mathbf{D}_{i,u})^2 + (\mathbf{v}^T \mathbf{D}_{i,w})^2}$$

于是法向互补变为：

$$0 \leq \mathbf{D}_{i,n}^T \mathbf{v}^{(l+1)} + \frac{1}{h} \Phi_i(q^{(l)}) - \mu_i \|\mathbf{v}_T\| \quad \perp \quad \gamma^n \geq 0$$

因为我们对 Eq. (6) 互补条件进行了“凸化”(relaxation)，得到了 Eq. (9)，所以整个系统最终变成了一个**锥互补问题 (CCP)**，即一个具有摩擦锥结构的变分不等式问题。

从一个非线性的互补问题NCP转化为了CCP锥互补问题，就可以使用APGD。

更明确地说，将 Eq. (5) 中速度表达式代入 Eq. (9)，就可以把整个系统**只用接触冲量 γ (拉格朗日乘子)**来表示，形成一个只在 γ 上优化的变分不等式 (即 CCP)。

符号定义 & 数学符号表：

★ 1. 接触数量

$$n_c := \text{number of contacts in } \mathcal{A}(q, \delta)$$

★ 2. 接触 Jacobian 总矩阵:

$$\mathbf{D} = [\mathbf{D}_1, \dots, \mathbf{D}_{n_c}] \in \mathbb{R}^{6n_b \times 3n_c}$$

每个接触点有一个 $\mathbf{D}_i \in \mathbb{R}^{6n_b \times 3}$, 表示它的 3 个方向 (法向、切向 $\mathbf{u}$ 和 $\mathbf{w}$) 的影响。

★ 3. 每个接触点的 Jacobian:

$$\mathbf{D}_i = [\mathbf{D}_{i,n}, \mathbf{D}_{i,u}, \mathbf{D}_{i,w}] \in \mathbb{R}^{6n_b \times 3}$$

它把这个接触点的冲量 $\gamma_i \in \mathbb{R}^3$ 映射为系统广义力。

★ 4. 接触点的速度项:

$$\mathbf{r}_i = \mathbf{b}_i + \mathbf{D}_i^T \mathbf{M}^{-1} \mathbf{f} \in \mathbb{R}^3$$

表示如果没有冲量, 系统当前对该接触点的速度趋势 (速度预测项)。

★ 5. 右侧常数项 $\mathbf{b}$:

$$\mathbf{b}_i = \begin{bmatrix} \frac{1}{h} \Phi_i(q^{(l)}) \\ 0 \\ 0 \end{bmatrix} \in \mathbb{R}^3$$

- 第一项是穿透深度的修正;
- 后两项是摩擦方向上的零约束项 (因为没有切向渗透)。

★ 6. Schur 补矩阵 (约束系统矩阵) :

$$\mathbf{N} := \mathbf{D}^T \mathbf{M}^{-1} \mathbf{D} \in \mathbb{R}^{3n_c \times 3n_c}$$

这是系统中用于描述冲量与速度之间关系的“压缩矩阵”, 是对系统质量矩阵 $\mathbf{M}$ 的一个投影, 通常非常稀疏 (sparse)。

上面符号是在定义, 如何使用公式5去带入公式9, 完成速度变量 $v(l+1)$ 的替换, 从而使问题变成一个只含接触冲量 γ 的问题。

$\gamma_i(l+1)$: 第 i 个接触点的拉格朗日乘子 (即冲量向量), 它有三个分量: 法向、切向 $\mathbf{u}$ 、切向 $\mathbf{w}$ 。

公式 (10) 和 (11) 是对摩擦接触问题的最终 **数学建模形式**, 它将多体系统中所有的接触与摩擦, 转化为了:

1. 一个 锥互补问题 (Cone Complementarity Problem, CCP)
2. 一个 等价的二次优化问题 (Quadratic Program, QP)

将 $\mathbf{v}^{(l+1)}$ 表达式代入上面的表达式:

$$\mathbf{D}_{i,n}^T \mathbf{v}^{(l+1)} = \mathbf{D}_{i,n}^T \left(\mathbf{v}^{(l)} + h\mathbf{M}^{-1}\mathbf{f} + \mathbf{M}^{-1} \sum_j \mathbf{D}_j \gamma_j \right)$$

我们定义:

$$\mathbf{r}_i = \mathbf{D}_{i,n}^T (\mathbf{v}^{(l)} + h\mathbf{M}^{-1}\mathbf{f})$$

$$\mathbf{N}_{ij} = \mathbf{D}_{i,n}^T \mathbf{M}^{-1} \mathbf{D}_j$$

则有:

$$\mathbf{D}_{i,n}^T \mathbf{v}^{(l+1)} = \mathbf{r}_i + \sum_j \mathbf{N}_{ij} \gamma_j$$

将其代入接触条件:

$$0 \leq \frac{1}{h} \Phi_i(q^{(l)}) + \mathbf{r}_i + \sum_j \mathbf{N}_{ij} \gamma_j - \mu_i \|\mathbf{v}_{i,T}\| \perp \gamma_i^n \geq 0$$

使用零段类型化的 $\|\mathbf{v}_{i,T}\|$ 的保存问题, 就能表达成关于全局出力形式:

接触约束+互补条件

将这些合并到一个整体系统后, 我们得到如下的互补条件组合:

将这些合并到一个整体系统后，我们得到如下的互补条件组合：

$$\gamma \in \Upsilon, \quad -(N\gamma + r) \in \Upsilon^\circ, \quad \gamma^T(N\gamma + r) = 0$$

逐项解释：

- $\gamma \in \Upsilon$ ：接触力必须在可行集合内（例如正的法向力、摩擦圆锥内）。
- $-(N\gamma + r) \in \Upsilon^\circ$ ：系统中产生的相对速度应落入对偶锥中。
- $\gamma^T(N\gamma + r) = 0$ ：互补条件：当接触力非零时，相对速度为零；当接触力为零时，相对速度允许非零。

这组条件就是所谓的**互补系统**（complementarity system）。

得到了公式10

$$\text{such that } \Upsilon_i \ni \gamma_i^{(l+1)} \perp -(N\gamma^{(l+1)} + r)_i \in \Upsilon_i^\circ \quad (10)$$

它等价于前面三项写法的紧凑表达形式，是一个标准的 **锥互补问题**（Cone Complementarity Problem，简称CCP）。

Cone Complementarity Problem (CCP)的最终形式

??

我觉得是个二阶锥问题SOCCP

构造一个优化问题，它的一阶最优性条件正好就是这个 CCP

我们尝试如下自然的二次函数作为目标：

$$f(\gamma) = \frac{1}{2}\gamma^T N\gamma + r^T \gamma$$

动机：

- 如果你接触过“Gauss 最小加速度原理”或“最小势能原理”，这类结构在动力学优化中很常见。
- 此函数是**严格凸函数**（前提： $N \succ 0$ ）——所以最优解唯一。

一阶求导

总体方法与推导流程：

1. 问题建模起点：DVI（Differential Variational Inequality）
多体系统 + 摩擦接触建模，首先以 DVI 形式写出。
2. 决策步骤（D1, D2）使得它变为 NCP（非线性互补问题）

表示法向力与摩擦的非线性约束

3. 通过 D3 再次线性化或凸化为 CCP

这一步将其写为带**锥约束**的 QP 问题

4. 最终形式是一个 QP

5. 求解后可得接触力 γ ，再求出速度 $v(t+1)$ 、广义坐标 $q(t+1)$

两个重要问题（核心）：

1. 使用 CCP 是通过放松 DVI 得到的

- 即将原本精确的不穿透条件（通常是带有非光滑函数）**放宽成了线性或凸形式**
- 这种近似在 $h \rightarrow 0$ 时是有效的，但在 **大时间步长** 时会带来伪影（artifacts）
- 这叫做**时间积分误差 + 近似误差耦合问题**

✦ 这强调了**时间步长选择**对数值精度的重要影响

2. N 仅为半正定，CCP 解可能不唯一

- 意味着目标函数是凸的，但可能有**不唯一的极小值**
- 特别是在摩擦接触问题中：
 - 存在多个接触力解都能满足约束
 - 摩擦力的方向性、接触冗余，造成**力/速度分布不唯一**

唯一性缺失的解释：

案例说明：**四条腿的桌子**

- 完全刚性、对称结构，接触冗余（4个接触点满足3个方程）
- 无法唯一分配腿上的反力（即使无摩擦）
- 如果有摩擦情况，唯一性更差

✓ 对无摩擦问题而言：

- **速度解**仍是唯一的（即系统下一步状态是确定的）
- 但力的分配（接触力 γ ）是多解的

✗ 有摩擦时：

- 即使是速度解，也**可能不唯一**
- 因为摩擦引入非线性锥约束，耦合更复杂

“四条腿的桌子”这个例子告诉我们：**冗余接触 + 非线性摩擦**的组合可能导致**力和速度都不唯一**，这就是摩擦刚体接触问题的数值挑战所在。

2.3 Convexification Artifacts

凸化伪影，从原始“非凸、非光滑”的 DVI 问题放松为一个凸优化问题（CCP），这种放松带来的“非真实物理现象”或“数值伪影”

举例场景：3D 球体滑动 + 摩擦 → 滚动

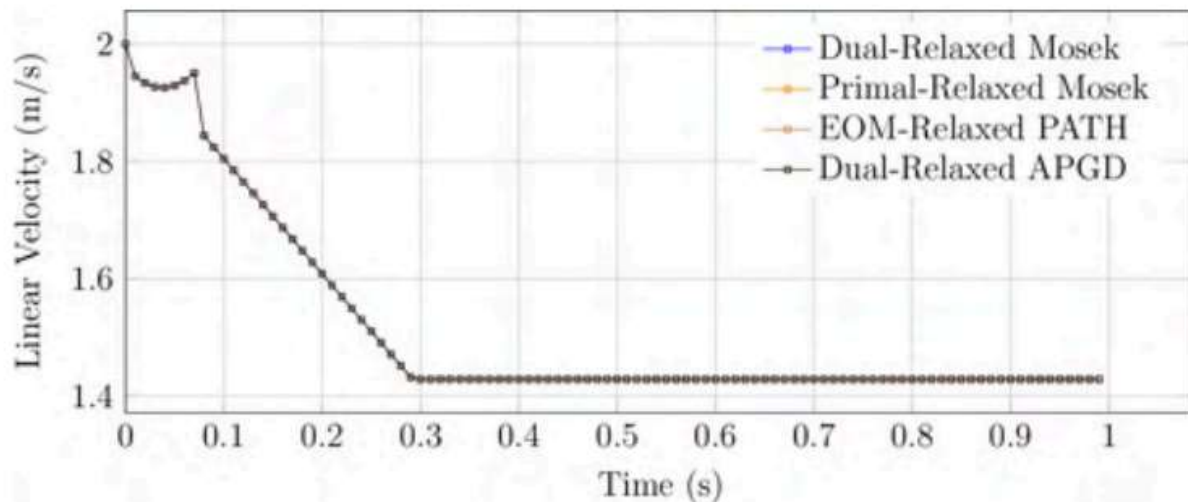
- 若你用**原始 DVI 模型（未放松）**去仿真，图像中可以看到**真实的滚动时间 trolling**，没有问题。
- 但如果你用 CCP 方法（放松了 DVI），球可能**提早进入滚动状态**，或者**转得过猛**，这就是伪影。

CCP 方法虽然好算，但它是 DVI 的一个放松近似，它牺牲了真实性换来了数值稳定性与可解性。

在高滑动速度下出现的数值伪影，其根源是**接触点法向相对速度的表达发生了改变**，因为放松方法中人为**减去了一个特定的项**。

减去这个项的结果是：**整个广义接触相对速度表达式中的最后一项也被“顺带”取消了**。

Relaxation（放松）通过省略接触速度中的摩擦依赖项简化了求解问题，但代价是**摩擦力方向与大小发生了数值偏差**，这在高速滑动情况下尤其明显。



现象：Artifact 出现

- 在 0.05~0.2 秒之间，球体线速度出现反常振荡
- 有的甚至先升后降，说明摩擦反作用被误判

这种非物理行为叫做 **jitter (颤抖/抖动)**，是伪影的典型表现

- 不管你是用 APGD 还是其他 QP 解法 (Mosek、PATH 等)，**只要你用了 convexified (放松) 模型，就会有 artifact。**
- 问题出在建模上，而不是求解器上。
- 想真实模拟 → 用原始 NCP (带非凸非光滑模型)
- 想快速模拟 → 用 CCP，但要接受 artifact 存在，尤其在高速滑动下
- 想折中 → 可以研究 **Hybrid 方法** (半解析补偿、摩擦项修正)

2.4 Embedding a Simple Cohesion Model

引入一个简单的黏结模型

黏结现象 = 两个物体即使没有正压力，也可以“粘在一起”

这个模型允许我们在 DVI 接触建模中引入黏结力 (cohesion) 机制，以支持更多物理现象 (例如颗粒吸附、胶水、磁吸等)，同时又不破坏原本的动力学结构。

3. THE ACCELERATED PROJECTED GRADIENT DESCENT METHOD

APGD 方法 (Accelerated Projected Gradient Descent)

为了解决前面提到的带锥约束的 QP 问题：

$$\min_{\gamma \in \Upsilon} \left(\frac{1}{2} \gamma^T N \gamma + r^T \gamma \right)$$

我们引入 **Nesterov 加速梯度法 (AGD)**，目的是：

- 快（比普通梯度下降快）
- 支持投影（Projected Gradient）
- 能适应数以百万计变量/约束的大规模系统

对于一个含 **100 万个刚体** 的系统：

- 变量维度约 12 million
- 约束条数约 4 million
- 必须用**高效优化算法**来解（APGD 是合适选择）

3.1 Preamble: Nesterov's Method

1.原始梯度下降法的收敛性

- 若 $f(x)$ 是**光滑函数（连续可微）**，且**Lipschitz 连续**（即梯度变化不太剧烈）
- 且 $f(x)$ 是**凸函数（convex）**
- 那么：梯度下降法（式12-13）可以**保证收敛到全局最优解**

公式 (12)：梯度下降的基本更新规则

✦ 公式 (12)：梯度下降的基本更新规则

$$x_{k+1} = x_k - \alpha_k \nabla f(x_k) \quad (12)$$

✅ 含义：

- x_k ：当前迭代步的变量（我们要优化的变量）
- $\nabla f(x_k)$ ：在点 x_k 处的梯度（即函数上升最快的方向）
- $-\nabla f(x_k)$ ：函数下降最快的方向
- α_k ：步长（**step size**），控制我们“往下降方向走多远”

■ 总结一句话：

用梯度下降法时，我们从当前点 x_k 出发，向着使 $f(x)$ 变小的方向走一步，步长是 α_k 。

✦ 公式 (13)：如何选择步长

$$\alpha_k = \arg \min_{\alpha} f(x_k - \alpha \nabla f(x_k)) \quad (13)$$

✅ 含义：

- 这不是任意选一个步长，而是通过 **一维最小化（line search）** 找到使函数值最小的那个 α_k 。
- 即在当前的下降方向上（ $-\nabla f(x_k)$ 方向）找一段距离，使得新的点 $x_k - \alpha \nabla f(x_k)$ 对应的函数值 $f(\cdot)$ 最小。

这种策略叫做 **精确线搜索（Exact Line Search）**。

2.加入投影 —— Projected Gradient Descent (PGD)

当你有约束怎么办？你就不能简单地往梯度方向走了，要**先走一步，再投影回约束集合**。

于是有了公式 (14)：

$$x_{k+1} = \Pi_C(x_k - \alpha_k \nabla f(x_k)) \quad (14)$$

$\Pi_C(\cdot)$ ：是把点投影回集合 C

- 即你走了一步之后，不一定还在集合内，需要“拉回来”

✦ 这就是 **PGD（投影梯度法）**。

$$f(x_k) - f(x^*) \simeq \mathcal{O}(1/k) . \quad (15)$$

符号	含义
$f(x_k)$	第 k 次迭代时的函数值（当前解的目标函数）
$f(x^*)$	最优解 x^* 对应的函数最小值（理论极小）
$f(x_k) - f(x^*)$	当前函数值距离最优函数值的差距，叫 函数误差
$\mathcal{O}\left(\frac{1}{k}\right)$	表示“以 $1/k$ 的速度”逐渐接近最优解

使用标准投影梯度下降法优化一个凸函数时，函数值收敛到最优值的速度是 $\mathcal{O}(1/k)$ ，也就是说误差大致随着 $1/k$ 减小。

Nesterov 方法核心思想：

“引入动量 (momentum) 来加速梯度下降”

加入了**动量 (momentum)**：当前方向不只看现在，还结合前面几步的趋势
绕过“之字形振荡”，直接朝最优点冲刺

我们希望将梯度下降变得更快，通过引入**动量项**和**步长更新机制**，来加速收敛速度到：

$$f(x_k) - f(x^*) = \mathcal{O}\left(\frac{1}{k^2}\right)$$

✅ 公式 (16)

$$x_{k+1} = y_k - t_k \nabla f(y_k)$$

- 在点 y_k 上计算梯度下降
- t_k : 当前步长
- 得到更新后的 x_{k+1}

★ 这是核心梯度更新步骤，但不是直接从 x_k 出发，而是从加速点 y_k

✅ 公式 (17)

$$\theta_{k+1}^2 = (1 - \theta_{k+1})\theta_k^2 + q\theta_{k+1}$$

- 控制“动量”的参数更新方式
- q 是调参值，控制收敛速率、加速力度
 - $q = 0$ 表示不使用强凸信息
 - $q = \mu/L$ 表示使用强凸信息（推荐）
- 这是一个非线性方程，通常需数值解（或特殊情形直接给定）

✅ 公式 (18)

$$\beta_k = \frac{\theta_k(1 - \theta_k)}{\theta_k^2 + \theta_{k+1}}$$

- 动量因子
 - 用来构建新搜索点 y_{k+1} ，它将“结合之前的速度信息”
-

✅ 公式 (19)

$$y_{k+1} = x_{k+1} + \beta_{k+1}(x_{k+1} - x_k)$$

- 构造下一轮的“前瞻点” y_{k+1}
- 这是 **Nesterov 动量核心公式**：不仅依赖当前位置，还“借力”于过去更新的趋势

只要函数 $f(x)$ 是凸函数且梯度 Lipschitz 连续（即光滑）

就可以保证这个算法在 $t_k \leq 1/L$ 下收敛

若进一步强凸（strongly convex），可以通过 $q = \mu/L$ 来调整加速效果

传统梯度下降：沿当前梯度方向一步步走。

Nesterov AGD：提前“预测下一步”，结合“惯性”，提前滑得更远。

加速投影梯度下降法（APGD, Accelerated Projected Gradient Descent） 的正式定义，基于前面讲过的 Nesterov AGD 方法扩展到了带约束优化问题中，例如：

- 多体动力学中的摩擦锥约束
- QP 问题的可行域投影

$f(x)$: 目标函数, 必须是**凸函数**且其梯度是 Lipschitz 连续的 (“平滑”)

常数 L : Lipschitz 常数, 即:

$$\|\nabla f(x) - \nabla f(y)\| \leq L\|x - y\|$$

意思是函数的梯度变化不会太剧烈

✅ 公式 (20): 投影步

$$x_{k+1} = \Pi_C(y_k - t_k \nabla f(y_k)) \quad (20)$$

- 从当前前瞻点 y_k 出发, 做一次梯度下降
- 然后把结果投影回可行集合 C (比如摩擦锥、非负约束等)
- 这是 **Projected Gradient Descent** 的核心步骤
- $\Pi_C(\cdot)$ 表示对集合 C 的投影操作

✅ 公式 (21): 更新动量参数 θ_{k+1}

$$\theta_{k+1}^2 = (1 - \theta_{k+1})\theta_k^2 \quad (21)$$

- 这个是解一个关于 θ_{k+1} 的非线性方程
- 初始时 $\theta_0 = 1$, 之后每次迭代根据上一轮的 θ_k 来更新

✅ 公式 (22): 计算动量因子 β_k

$$\beta_k = \frac{\theta_k(1 - \theta_k)}{\theta_k^2 + \theta_{k+1}} \quad (22)$$

- β_k 是构造下一步搜索方向的加速系数
- 它结合了过去的两次变量变化趋势 (相当于“惯性”)

✅ 公式 (23): 生成新的预测点 y_{k+1}

$$y_{k+1} = x_{k+1} + \beta_{k+1}(x_{k+1} - x_k) \quad (23)$$

- 使用当前点与上一步差值的线性组合
- 实现“动量推进”, 比单纯从当前点出发更快

APGD 是在约束优化问题中使用的加速梯度法，通过“**梯度下降 + 可行域投影 + 动量加速**”三合一的机制，实现比普通 PGD 更快的收敛速度，尤其适用于摩擦锥类接触力求解问题。

3.1.1 Adaptive Step Size

APGD 在 Lipschitz 连续的凸函数下，若使用固定步长 $t_k \leq 1/L$ ，可以获得 $O(1/k^2)$ 的最优收敛速率。

但在现实中，有两个问题：

1. 全局 Lipschitz 常数 L 往往不知道；
2. 固定步长 $1/L$ 太小，导致收敛慢、不敏感。

所以我们想让步长 $t_k = 1/L_k$ 自适应变化，动态估计一个当前迭代中“有效的 L_k ”即可。

■ 什么是 Lipschitz 常数？

函数 $f(x)$ 的梯度是 Lipschitz 连续的意思是：

$$\|\nabla f(x) - \nabla f(y)\| \leq L\|x - y\|$$

L 控制了函数的“变化陡峭程度”。在优化中我们通常用 $1/L$ 来做步长估计。

自适应策略 (Backtracking Line Search)：

我们不直接用全局的 L ，而是动态估计当前局部的 L_k ，再用它计算步长：

$$t_k = \frac{1}{L_k}$$

▢ 核心判断公式：条件 (24)

$$f(x_{k+1}) \leq f(y_k) + \nabla f(y_k)^T(x_{k+1} - y_k) + \frac{L_k}{2}\|x_{k+1} - y_k\|^2 \quad (24)$$

这个式子是目标函数的一阶泰勒展开上界。

★ 含义解释：

- $f(y_k) + \nabla f(y_k)^T(x_{k+1} - y_k)$ ：是从 y_k 出发的线性预测
- 加上 $\frac{L_k}{2}\|x_{k+1} - y_k\|^2$ ：是允许的“非线性误差上限”
- 整体右边是上界估计，只要 $f(x_{k+1})$ 小于这个估计，就说明当前步长足够小

✅ 如果不满足，就加大 L_k ，重新尝试

- 初始设 $L_k = L_0$ ，若不满足 (24)，就让 $L_k \leftarrow 2L_k$
- 重复直到满足为止 (这叫 backtracking)
- 最终得到一个局部满足 Lipschitz 条件的 L_k ，再用 $t_k = 1/L_k$ 做更新

To start off, L_0 can be estimated as...

$$L_0 = \frac{\|\nabla f(z_0) - \nabla f(z_1)\|}{\|z_0 - z_1\|}$$

这个就是两点之间的梯度变化除以距离，等于一个局部的 Lipschitz 常数估计。

方程24推导过程：

■ 背后原理 (Taylor 上界)

这个不等式其实就是一阶 Taylor 展开加上一个保守上界项：

$$f(x) \approx f(y) + \nabla f(y)^T(x - y) \quad (\text{线性化})$$

但因为 $f(x)$ 可能比这个线性近似大，所以加入一个“曲率上界”项：

$$+ \frac{L}{2} \|x - y\|^2$$

表示最多比线性项多这么多。

假设 $f(x)$ 是：

- 凸函数
- 梯度 L-Lipschitz 连续：

$$\|\nabla f(x) - \nabla f(y)\| \leq L\|x - y\|, \quad \forall x, y$$

这个条件也等价于下面这个不等式成立（优化中常用）：

$$f(x) \leq f(y) + \nabla f(y)^T(x - y) + \frac{L}{2} \|x - y\|^2 \quad (\text{对任意 } x, y) \quad (\star)$$

这就是我们要推的式子 (24) 的母式。

把值代入，就可以得到24

$$f(x_{k+1}) \leq f(y_k) + \nabla f(y_k)^T(x_{k+1} - y_k) + \frac{L_k}{2} \|x_{k+1} - y_k\|^2 \quad (24)$$

APGD 中的第二种自适应步长策略 (Adaptive Step Size Strategy #2)。

它是对前一节“backtracking 只增不减”的改进：允许步长既能变小也能变大，以获得更好的性能。

与前一种方法一样，如果不满足条件 (24)，还是用 backtracking 的方式让 $L_k \leftarrow 2L_k$ 来增加 Lipschitz 估计值，从而减小步长。

如果某次迭代中使用的 L_k 实际上过大、而条件 (24) 一直都满足，那么可以适当降低 L_k 来尝试更大的步长。

$$L_{k+1} = 0.9 L_k$$

因为你尝试更大的步长，有时条件 (24) 会失败，所以可能需要多次回退。但这是可以接受的。

这正是关键点：**更大步长意味着更快的进展**，只要系统保持稳定性，就可以**整体加速收敛过程**。

3.1.2 Adaptive Restart. 重启机制

在某些问题中，Nesterov 方法的动量项 (β_k) 可能导致“过冲 (overshoot)”或者“振荡 (rippling)”，从而减慢甚至干扰收敛。

问题本质：动量方向反了

- AGD 使用动量项：

$$y_k = x_k + \beta_k(x_k - x_{k-1})$$

- 如果 $x_k - x_{k-1}$ 这个方向不再指向下降方向 (即与 $-\nabla f$ 相反)，就会造成优化过程反复横跳、震荡不收敛

✅ 重启策略 (Adaptive Restart)

📌 条件判据：

$$\nabla f(y_{k-1})^T (x_k - x_{k-1}) > 0$$

如果这个内积大于 0，说明：

- 动量方向 $x_k - x_{k-1}$ 与当前梯度方向 $\nabla f(y_{k-1})$ 是同向
- 即动量在“拉着你往错误方向走”❌

✳️ 这时就触发重启机制：

🌟 操作：

- 设置：

$$\theta_k = 1, \quad y_k = x_k$$

- 即：动量归零，重置迭代点为当前位置，下一步重新开始

假设当前你正在“下坡”，但动量项却把你“往山上推”：

- 那么你走的方向不是下降方向 → 浪费步数，甚至走反了
- 所以需要判断动量方向与梯度方向之间的夹角（通过内积判断）
- 一旦判断失效 → 重启！

$$\nabla f(y_{k-1})^T (x_k - x_{k-1}) > 0 \Rightarrow \text{重启动量}$$

3.1.3 Fallback.

弥补非单调优化方法（如 AGD）在提前终止时可能“停在次优点”的问题

- 像 Nesterov 加速法这类方法，**目标函数值 $f(x_k)$ 可能不是逐步下降的**（即不是单调优化）：
 - 它可能先降后升再降（由于动量波动）
- 所以，即使你跑了 k 步，**也不能保证最后一个点 x_k 是最好的**

Fallback 回退机制：

- 在优化过程结束后，我们回顾历史上所有迭代点 $\{x_0, x_1, \dots, x_k\}$
- 从中选出“**最好**”的那个点，作为最终返回解

如何选“最好”？

通过一个选定的度量函数（metric）：

$$r(x_i) \in \mathbb{R}$$

最常见的就是目标函数值本身：

$$r(x_i) = f(x_i)$$

✅ 最终解为：

$$\hat{x} = x_i, \quad \text{其中} \quad i = \arg \min_{j \in [0, k]} r(x_j)$$

也就是说：

- 返回的是在所有迭代中使度量 $r(\cdot)$ **最小的那个点**
- 即便最后几步可能表现不佳，也不会影响整体最优性

3.2 The Overall Algorithm

公式 (25) 定义的 metric

一、公式 (25) 定义的 metric

$$\psi = \frac{1}{3n_c g_d} (\gamma - \Pi_C(\gamma - g_d(N\gamma + r))), \quad r = \|\psi\|_2 \quad (25)$$

含义:

- γ : 当前候选解
- $N\gamma + r$: 梯度 (实际上是 CCP 的 KKT 残差)
- $\gamma - g_d(N\gamma + r)$: 沿负梯度方向走一步
- $\Pi_C(\cdot)$: 将新点投影回可行域
- 所以:
 - $\gamma - \Pi_C(\cdot)$: 描述“走一步被拉回了多少”
 - 除以步长 g_d : 相当于归一化步长方向差
- 最终:

$$\psi \approx \text{沿约束边界的梯度方向} \Rightarrow \|\psi\| = r(\gamma) \text{ 是度量}$$

二、为什么这样定义 fallback metric?

“First note that if $\gamma = \gamma^*$ is optimal, then $\Pi_C(\gamma - g_d(N\gamma + r)) = \gamma^*$ ”

✅ 说明:

- 若 γ 是最优点, 沿梯度方向走一步也**不会离开可行域** (最优点在临界边界)
- 所以:

$$\Pi_C(\gamma - g_d(N\gamma + r)) = \gamma \Rightarrow \psi = 0$$

这表明:

- $\psi = 0 \Leftrightarrow \gamma$ 是最优点

三、若 γ 不是最优点时

| "Then it can be verified that $\Pi_C(\gamma - g_d(N\gamma + r)) = \gamma - g_d\psi$ "

这是关键点，它告诉我们：

(26)

- 如果你从 γ 沿负梯度方向走一步再投影回来，等价于：
 - 沿 $-\psi$ 方向走步长 g_d
- 所以 ψ 是一个近似的投影梯度方向

算法伪代码讲解

Algorithm 1 APGD ($N, r, \tau, N_{max}, \gamma_0$)

```
1:  $\hat{\gamma}_0 = \mathbf{1}_{n_c}$ 
2:  $\mathbf{y}_0 = \gamma_0$ 
3:  $\theta_0 = 1$ 
4:  $L_k = \frac{\|N(\gamma_0 - \hat{\gamma}_0)\|_2}{\|\gamma_0 - \hat{\gamma}_0\|_2}$ 
5:  $t_k = \frac{1}{L_k}$ 
6: for  $k := 0$  to  $N_{max}$  do
7:    $\mathbf{g} = N\mathbf{y}_k + \mathbf{r}$ 
8:    $\gamma_{k+1} = \Pi_{\mathcal{C}}(\mathbf{y}_k - t_k \mathbf{g})$ 
9:   while  $\frac{1}{2}\gamma_{k+1}^T N \gamma_{k+1} + \gamma_{k+1}^T \mathbf{r} \geq \frac{1}{2}\mathbf{y}_k^T N \mathbf{y}_k + \mathbf{y}_k^T \mathbf{r} +$   

 $\mathbf{g}^T (\gamma_{k+1} - \mathbf{y}_k) + \frac{1}{2}L_k \|\gamma_{k+1} - \mathbf{y}_k\|_2^2$  do
10:     $L_k = 2L_k$ 
11:     $t_k = \frac{1}{L_k}$ 
12:     $\gamma_{k+1} = \Pi_{\mathcal{C}}(\mathbf{y}_k - t_k \mathbf{g})$ 
13:  end while
14:   $\theta_{k+1} = \frac{-\theta_k^2 + \theta_k \sqrt{\theta_k^2 + 4}}{2}$ 
15:   $\beta_{k+1} = \theta_k \frac{1 - \theta_k}{\theta_k^2 + \theta_{k+1}}$ 
16:   $\mathbf{y}_{k+1} = \gamma_{k+1} + \beta_{k+1} (\gamma_{k+1} - \gamma_k)$ 
17:   $r = r(\gamma_{k+1})$ 
18:  if  $r < r_{min}$  then
19:     $r_{min} = r$ 
20:     $\hat{\gamma} = \gamma_{k+1}$ 
21:  end if
22:  if  $r < \tau$  then
23:    break
24:  end if
25:  if  $\mathbf{g}^T (\gamma_{k+1} - \gamma_k) > 0$  then
26:     $\mathbf{y}_{k+1} = \gamma_{k+1}$ 
27:     $\theta_{k+1} = 1$ 
28:  end if
29:   $L_k = 0.9L_k$ 
30:   $t_k = \frac{1}{L_k}$ 
31: end for
32: return Value at time step  $t^{(l+1)}, \gamma^{(l+1)} := \hat{\gamma}$ .
```

APGD($N, r, \tau, N_{max}, \gamma_0$)

- N : CCP 中的系统矩阵 (如 $N = M^{-1}D^T D$)
- r : 线性项 (如 $r = M^{-1}D^T b$)
- τ : 收敛容差阈值 (如 $\tau = 10^{-6}$)
- $N_{\max}$: 最大迭代次数
- γ_0 : 初始猜测 (接触力)

4. NUMERICAL EXPERIMENTS

使用 **Chrono** 框架进行多体动力学模拟；
在 Chrono 中集成了多种解 CCP 的方法；
本文实现并测试了 **APGD 方法**，并与 **Jacobi** 和 **GS (Gauss-Seidel)** 进行对比

Chrono 提供的默认解接触力的 CCP 方法有：

方法名	特点
Projected Gauss-Seidel (PGS)	常用，简单但慢，不适合并行
Jacobi	支持并行化，但收敛慢
Krylov-based	更高级，适合大型稀疏系统

- 实验目的：
1. 验证 **APGD 的有效性**（效率、稳定性）
 2. 与 **Jacobi、GS 方法**对比，看看是否更快/更稳定

比较项	意义
精度	哪种方法收敛到更优解
收敛速度	哪种方法达到目标误差更快
鲁棒性	哪种对问题初始值、系统规模更不敏感
并行能力	哪些能在 CPU/GPU 上高效并行

- Chrono 默认使用 **Jacobi 法** 作为顺序求解器（即单核）
- 并行版本也是基于 Jacobi 法
- 所以：**APGD 被认为是高性能替代候选**

本节数值实验基于 Chrono 平台，测试了新实现的 APGD 方法解 CCP 的性能，并与默认的 Gauss-Seidel 和 Jacobi 方法进行比较，关注收敛速度、稳定性和并行潜力，以探索更高效的多体接触求解策略。

4.1 APGD Performance Analysis

Algorithm 2: Projected Gauss-Seidel (PGS) 方法的完整伪代码以及其背景说明。

这是 Chrono 默认支持的接触力求解方法之一，用于解多体接触问题中的 **CCP (Cone Complementarity Problem)**。

PGS 是最常用的 DVI-based 接触求解算法 (Discrete Variational Inequality)

Gauss-Seidel 方法通过对每个接触点逐一迭代、松弛和投影，构建了简单而稳定的接触力迭代求解器，广泛用于 DVI-based 模拟中。尽管其收敛速度有限且难以并行，但因实现简洁，依然是业界基础方案。

4.1.1 Speed of Convergence Study: APGD vs. Gauss Seidel vs. Jacobi.

实验场景设置：

- 容器中装有 4000 个球体，半径 0.15 m，质量 1 kg
- 顶部再放一个重物（重板，slab），质量 1000 kg

这个场景很典型：**颗粒堆叠 + 上压物体 + 静摩擦稳定性**

模拟挑战点：

- 多层颗粒之间存在**复杂的接触网络** (force chain)
- 接触分布深、复杂，带来显著的**非线性摩擦耦合**
- 是一个公认的**难解问题场景**

实验流程说明：

- 使用 **GS 解法** 先运行系统到参考状态（静力学平衡）
- 然后将 $\gamma_0=0$ 作为初始猜测（cold start）
- 对比 3 种方法（APGD、GS、Jacobi）各自的 1000 次迭代效果

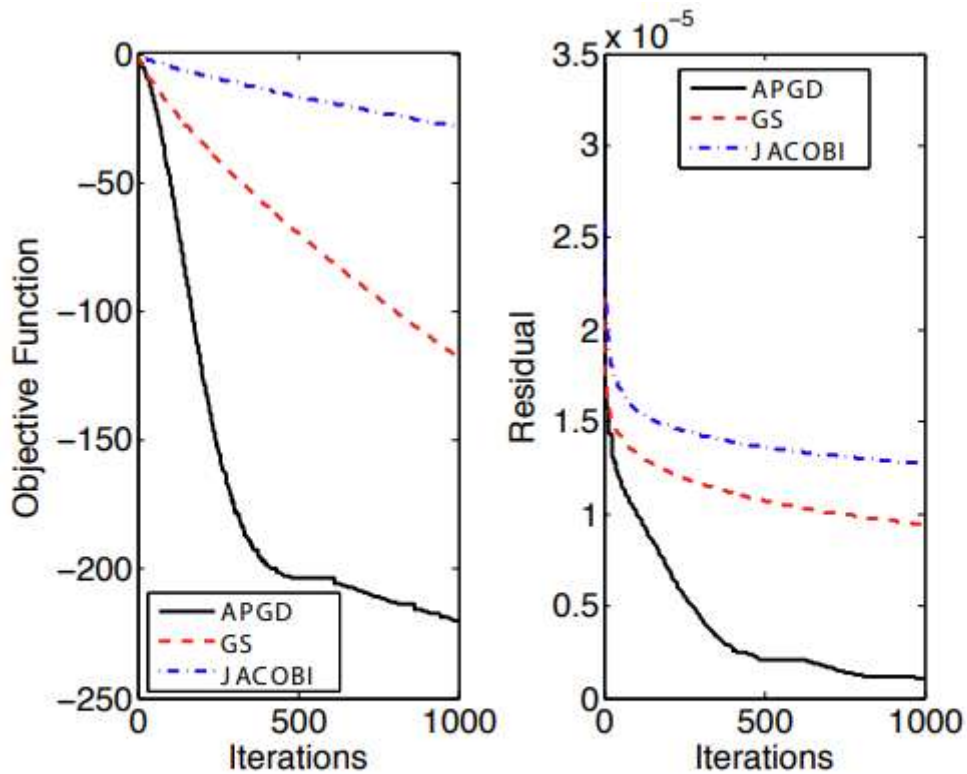


Fig. 5. APGD vs. GS vs. Jacobi: Residual r and Objective Function $f(\gamma)$ after 1000 iterations for the pressure test with a slab of 1000 kg, see Fig 4.

实验结果总结：

方法	最终残差 r	目标函数值 $f(\gamma)$	特性
APGD	1.1×10^{-6} ✓	最小 ✓	收敛最快，精度最高
GS	1.2×10^{-5}	中等	收敛较慢
Jacobi	1.2×10^{-5}	最大 ✗	最慢

图 5 清晰显示了在复杂颗粒接触问题中，**APGD 显著优于 GS 和 Jacobi**：不仅目标函数值更优，残差更小，而且收敛速度快出一个数量级，表现出色，是大规模 DVI 问题中更现代更高效的首选方法。

4.1.2 APGD Scaling Analysis

APGD 并行扩展与收敛性能的实验数据分析部分

表 I: 不同 slab 质量下 Jacobi / GS / APGD 方法的残差和目标函数值

表 II: 收敛到特定容差时, 三种方法所需迭代次数与时间

表 III: 并行 APGD 的缩放测试结果 (对象数量 vs. 单步耗时)

APGD 已在 Chrono 中实现 OpenMP 多核并行化;

碰撞检测阶段也通过并行加速 (参考 Mazhar et al. 2011);

本节通过“搅拌器模型”验证并行扩展性:

实验模型:

- 圆柱容器 + 四叶桨叶 (以 $\frac{\pi}{2}$ rad/s 逆时针旋转)
- 容器中有球体、椭球、立方体和圆柱体, 平均半径 0.05m
- 密度 1000 kg/m^3 , 摩擦系数 $\mu = 0.5$, 内聚力 $\hat{c} = 200\text{N}$

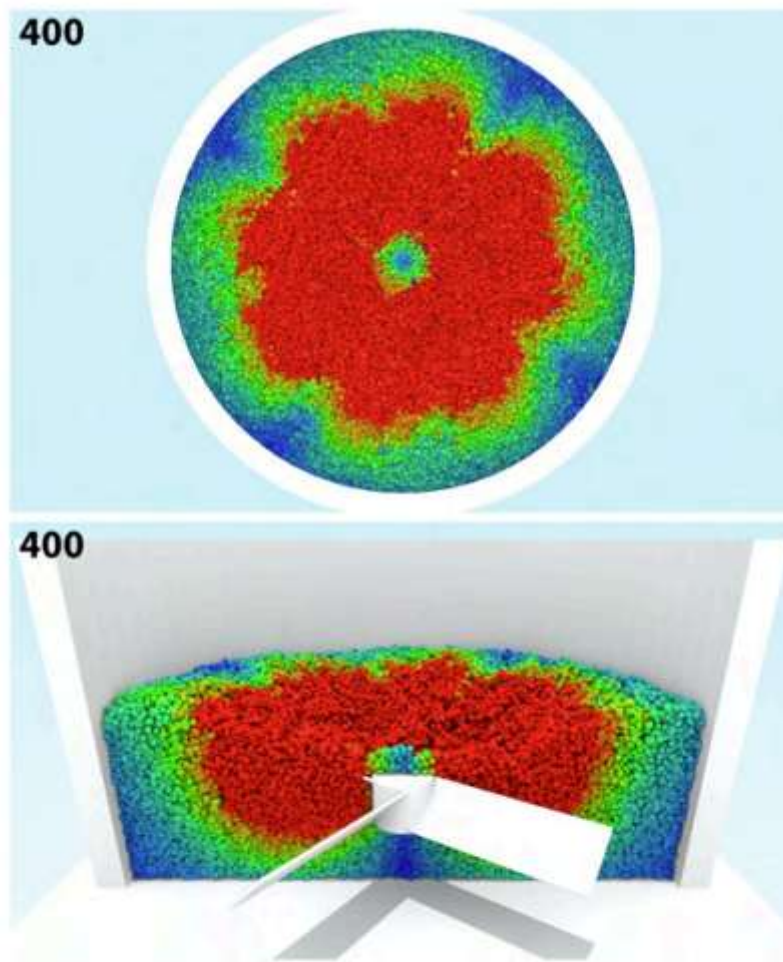


Fig. 6. Mixer simulation with cohesive material made up of several different geometries. The mixer blade rotates counter-clockwise pulling material down.

与二阶方法（如 Mosek、PATH）相比，APGD 方法迭代次数多但每步便宜，在中大规模接触问题中更快、更易并行实现，尤其在物理仿真和粒子系统中更适合。

4.2 Other APGD Simulations

APGD 方法如何处理“带内聚力的 CCP 问题”

该机制如何嵌入实际物理仿真系统中

同时验证其对不同材料行为的模拟能力

APGD 在模拟高接触密度、强非线性行为（如内聚断裂）的颗粒材料时表现出的数值稳定性和可扩展性，并通过改变 cohesion 强度控制材料从“软雪”到“硬冰”的物理行为。

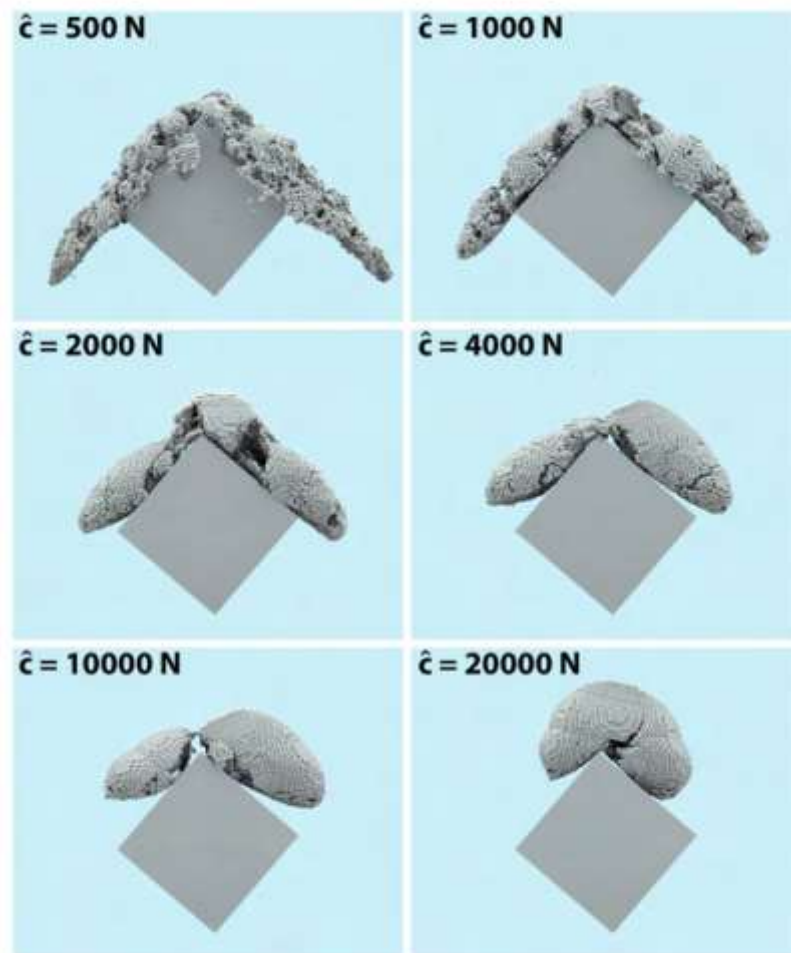
两个目标：

1. 验证 **内聚力模型** (cohesion) 是否能成功加入 DVI-CCP 框架；
2. 检验 **APGD 方法** 对修改后 CCP（含 cohesion 约束）的求解能力

场景构建

✅ 设置：

- 75000 个均匀球体，速度为 10 m/s，自由落体
- 落在一个以 45° 倾斜的立方体上
- 球与球之间摩擦系数 $\mu = 1.0$
- 球与墙体摩擦系数 $\mu = 0.1$



越高的 cohesion 值，代表颗粒之间越**不容易分离**，越能抵抗拉伸。

4.2.2 Bulldozer Dynamics.

这一节构建了一个带高质量刚体、复杂铰接结构、驱动约束与非线性接触的大规模多体系统，用以验证 APGD 在处理病态 N 矩阵与复杂运动约束下的稳定性和扩展能力，是工程仿真中典型的 benchmark 场景。

4.2.3 Modeling Deformation and Brittleness Using Cohesion

不同内聚力下的物体断裂演化

4.2.4 Granular Flow.

土壤流动模拟 Granular Flow

4.2.5 Impact Problem

冲击问题 Impact Problem

APGD 在多个复杂材料力学场景中的应用能力，包括脆性断裂、颗粒流、非弹性冲击等，验证其不仅适用于基础接触力计算，还能有效扩展到软物质、材料失效及连续介质类问题建模中。

5. DISCUSSION

首次将 cohesion 模型引入刚体多体动力学中

提出 APGD 方法高效求解 CCP

替代 GS / Jacobi：并行接触建模中作用

关于 Relaxation 带来的问题与价值

6. CONCLUSIONS AND DIRECTIONS OF FUTURE WORK

APGD 核心贡献：

- 在每个时间步使用 APGD 解摩擦接触中的锥互补问题（CCP）；
- 针对大规模刚体系统（上百万个物体、千万个接触）：
 - 提供 **数值稳定**
 - 支持 **并行扩展**
 - 可与 Chrono 仿真平台集成

实验结果表明：

- **几乎所有测试中**，APGD 的求解速度比 Chrono 默认 Gauss-Seidel 快一个数量级
- 即使在不同摩擦、接触密度、约束组合下，APGD 仍表现出极高鲁棒性

- 指出当前 Chrono 框架对冲击类（impulse-based）建模支持较弱
- 暗示未来工作需增强“瞬时冲击+刚体反弹”建模能力

作者展望性能优化可通过：

- 更好利用：
 - SIMD 向量化指令（如 AVX）
 - 多核并行
 - GPU 协处理

- 结构层面：
 - **Interior Point Method**（用于 DVI 问题的替代解法）
 - 非矩阵实现（进一步降低内存带宽压力）

LCP AND CCP

（有没有把摩擦锥进行多边体化，就是将基向量运用在摩擦锥的底部）

关于具体这篇文章是怎么写的还需要去chrono里的关于APGD的代码查看一下，是不是LCP问题

CCP能否转化QP

CCP 可以等价转化为 QP 问题，前提是锥为凸锥、矩阵为正定或半正定，解出的最优解满足 KKT 条件，从而满足锥互补性。Chrono 中的 APGD、SPG、P-SPG-FB 解法本质上就是这样“优化-互补性”转化策略的实现。

需要去考证

单从代码来看，是直接从CCP问题转换为了QP问题，并没有涉及新的基向量的出现

所以APGD是CCP问题的solver，通过投影实现，适合物理真实性模拟。

CCP → QP → APGD

