**物理引擎求解器中的部分数学理论及公式推导**

**目录**

[1. 概述 1](#_Toc27448)

[2. 求解器 1](#_Toc30444)

[2.1. LCP/MLCP 求解 1](#_Toc536)

[2.1.1. Gauss-Seidel (GS迭代求解法) 1](#_Toc14764)

[2.1.2. Projected Gauss-Seidel (PGS迭代求解法) 2](#_Toc7131)

[2.1.3. Lemke (互补枢轴算法) 2](#_Toc24063)

[2.1.4. Dantzig-Wolfe算法 4](#_Toc4910)

[3. 物理约束式子的由来 5](#_Toc8127)

[3.1. 基本物理系数符号 5](#_Toc22752)

[3.2. 根据动量守恒定律推导约束方程式 6](#_Toc10099)

[3.2.1. 接触约束方程式 6](#_Toc19075)

[3.2.2. 摩擦力约束方程式 10](#_Toc32487)

[3.2.3. 非接触约束方程式 11](#_Toc27444)

[参考文献 13](#_Toc9045)

[附录一：单纯形法样例 13](#_Toc30599)

# 概述

1. 梳理物理引擎中求解器的几个 (M)LCP求解算法
2. 根据基本的物理知识，从数学角度推导输入到求解器中的方程式

# 求解器

问题描述：求解方程组 Ax = b, 其中A为n阶方阵，x和b均为n维列向量

* 1. LCP/MLCP 求解

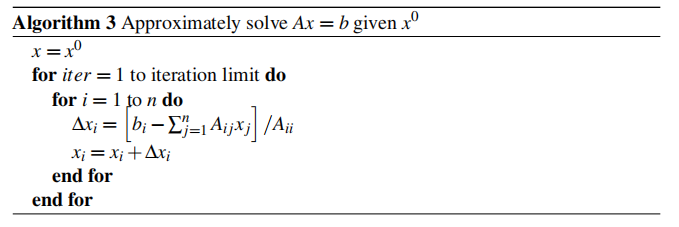
### Gauss-Seidel (GS迭代求解法)

矩阵

可以被分割为 A = D - L - U, 其中D是对角矩阵，L是左下部分矩阵，U是右上部分矩阵，得：**(L + U + D) x = b**

即 **Dx = b - (L + U)x**

根据该式设计



### Projected Gauss-Seidel (PGS迭代求解法)

PGS方法和GS方法的区别仅仅在于，PGS在每次迭代后会将 结果 截断到一个给定范围内，即所求x有上下界。

当矩阵 A 是正定矩阵的时候，PGS可得严格被证明为收敛（证明可参考任意数值分析（又名数值计算）书籍）

### Lemke (互补枢轴算法)

Ax - b = 0

b

（可参考大学教材《运筹学》第一章单纯形法相关内容）

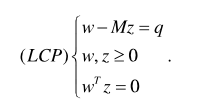
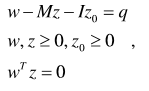
基本概念：

基向量：一个矩阵中的一组（若干）行（列）向量，组内任一向量都不能被这组基向量的其它向量线性表示，而其它非基行（列）向量都可以用这组基向量线性表示。

Lemke算法目标：

s.t.

上式可改写为：（证明可参考其它资料对Lemke算法的描述，此处略）

引入人工变量，有

查找主元，然后使用Gauss-Jordan消元法。

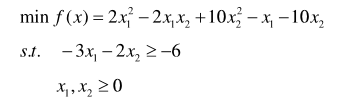
由原始单纯形法改进而来，主元消去法，由一个准互补基本可行解出发，通过主元消去法（转轴方法）求出一个新的准互补基本可行解，不断迭代。可能求解失败。

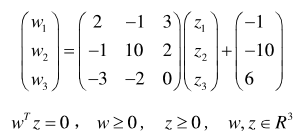
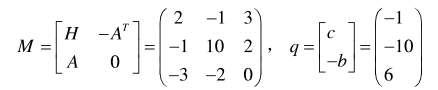
* 按照最小比值规则确定离基变量
* 保持准互补性，若w\_i(z\_i)是离基变量，则z\_i(w\_i)是进基变量

具体步骤：

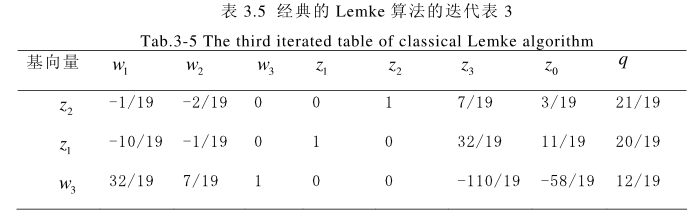
1. 若q >= 0, 则(w, z) = (q, 0)为解，否则,取max{-q\_i}所在的行s行为主行，对应的列为主列，主元消去，令
2. 设现行表中变量下面的列为，若 <= 0，退出；否则按最小比值规则确定指标r，使 ,如果r行的基变量是，转4；否则转3
3. 设r行的基变量为或 ，进基，以r行为主行，对应的列为主列，主元消去。如果离基变量是，则令；如果离基变量是，则令 = ，转2
4. 进基，离基。以r行为主行，对应的列为主列，主元消去。得可行解，退出

具体样例：









* + 1. Dantzig-Wolfe算法

只知道同样基于换基消元法，暂未找到文献描述具体选基与换基原理。

# 物理约束式子的由来

## 基本物理系数符号

|  |  |  |  |
| --- | --- | --- | --- |
| 物理系数 | 符号表示 | 类型 | 相关公式 |
| 质量  (mass) |  | 标量 |  |
| 惯性张量(转动惯量) |  | 二阶矩阵 |  |
| 力  (force) |  | 矢量 |  |
| 力矩  (torque) |  | 矢量 | , 其中为质心到指定点的矢量 |
| 线速度  (velocity) |  | 矢量 | dt |
| 角速度 |  | 矢量 |  |
| 线加速度  (acceleration) |  | 矢量 |  |
| 角加速度 |  | 矢量 |  |
| 动量(momentum) |  | 矢量 |  |
| 角动量 |  | 矢量 |  |
| 冲量(impulse) | *I* |  |  |
| 角冲量 |  |  |  |
| 动能 | *E* | 标量 |  |
| 转动动能 | *E* | 标量 |  |
|  |  |  |  |
| 质心位置  (position) |  | 矢量 |  |
| 位置朝向  (orientation) |  | 四元数 | dt |
| 广义位置 |  | 矩阵 |  |
| 广义速度  (status) |  | 矩阵 |  |
| 广义合外力 |  | 矩阵 |  |
| 法线  (normal) |  | 矢量 |  |
| 摩擦系数 |  | 标量 |  |

## 根据动量守恒定律推导约束方程式

### 接触约束方程式

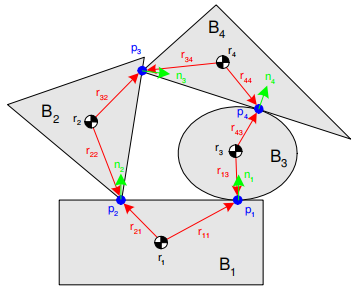
目标：把约束方程式写成 Ax + b = 0 的样式，以使用上述求解LCP问题的算法。

（参考《Physics-Based Animation》）

一个封闭的系统一定遵循动量守恒/能量守恒定律。

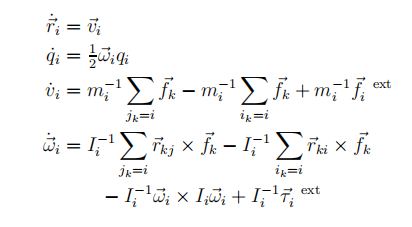
使用动量守恒比使用能量守恒更容易列方程式求解。物理系统的能量中，动能容易和其它势能、热能等相互转化，如两个非光滑刚体进行碰撞，由于摩擦力的存在，会使得一部分动能转化为热能，相比之下，动量的表达式比能量的表达式相对简单，更适合用来做物理引擎求解。

则问题转化成：已知一个物理系统中n个物体的固有属性（如质量、摩擦系数等），及初始广义位置、广义速度，求经过时刻后的广义速度。



整个场景中，对于第k个接触点，两边接触的物体记为和，记为第k个接触点的接触力，为物体i所受的其它外力，如重力等；为质心到接触点的矢量，为物体i所受的其它外力矩。

则对第i个物体有



其中前两个式子是根据位置对时间的导数即为速度列的，后两个式子是根据牛顿第二定律列的。

第二个式子推导过程：

四元数乘法：，，则

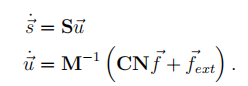
旋转角度，旋转轴，以不带矢量箭头的表示角速度的模长，则任一四元数都可以写成：

瞬时旋转中,角速度可以看成不随时间变化，则上式对时间求导，得

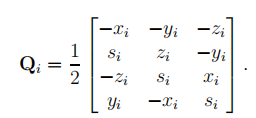
后两个式子的原型为：

记无法直接获得的接触力 ]，则上述两个式子可以表示成：

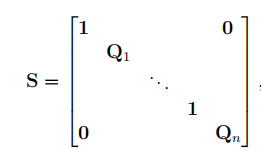
简记为



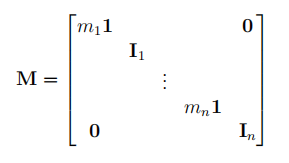
其中位置朝向，



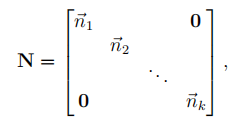
满足

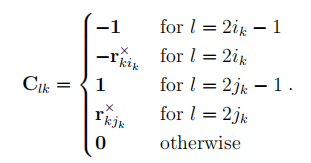


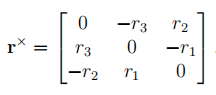
广义质量矩阵



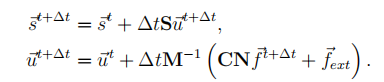
接触点法线矩阵







采用欧拉积分的方式，在时刻，有



转化成LCP形式

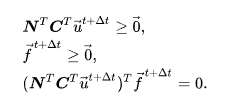
接触点，两边接触的物体和，为质心到接触点的矢量，则物体在接触点的速度为：和。

则两个物体在接触点处法线方向上的相对速度（记为接触速率）为

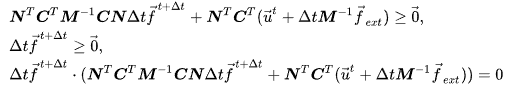
为的第k个分量，当该值大于0时，表示两个物体有相互分离的趋势。

当两个物体在时刻t接触，那么在时刻，要么接触完成，接触力为0，接触点速率大于0；要么正在接触，接触力大于等于0，接触点速率为0。

即有



即



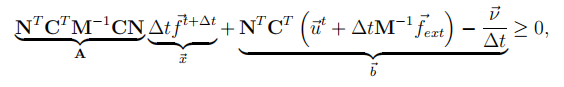


而书上多了





最终可以表示成的方程组形式。



### 摩擦力约束方程式

约束力不做功, 也就是说, 约束力的方向与速度的方向垂直.

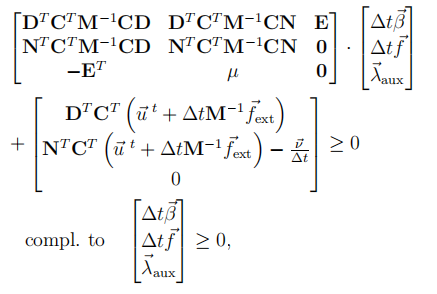
摩擦力方向

摩擦力大小

则接触力



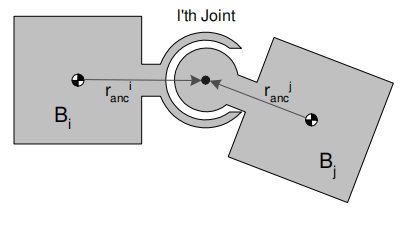


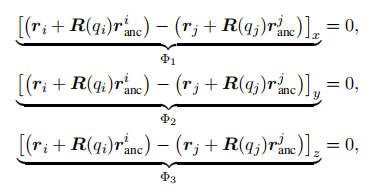


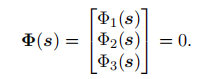
最终同样可以表示成的方程组形式。

### 非接触约束方程式

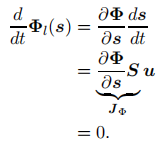
以铰链约束为样例

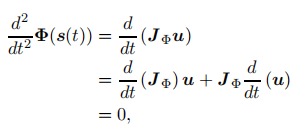






其中s是上面出现过的广义位置。上式同样对时间求导，得



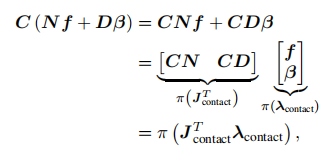


可得

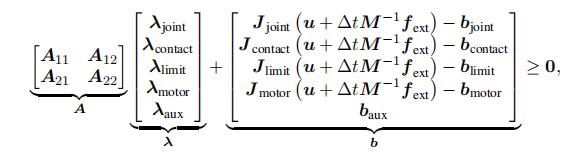
根据这里的约束力不做功，两边对时间求导得，则约束力可以表示成Jocabian矩阵的广义倍数

可以证明上述的接触约束力也可以表示成Jocabian矩阵的广义倍数





各种约束组合方程式



Bullet中只分了三种：接触约束、摩擦力约束和非接触约束。

# 参考文献

[Physics-Based Animation]( <https://www.researchgate.net/profile/Kenny-Erleben/publication/247181209_Physics-Based_Animation/links/5e1b2ed04585159aa4cb43d8/Physics-Based-Animation.pdf> )

[Lemke-Howson Algorithm]( <https://web.stanford.edu/~saberi/lecture4.pdf> )

[Iterative Dynamics with Temporal Coherence]( <https://box2d.org/files/ErinCatto_IterativeDynamics_GDC2005.pdf> )

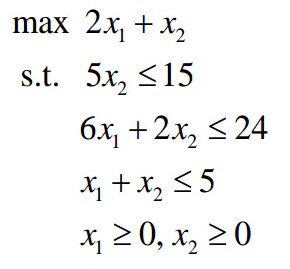
[Practical methods of optimization] by Fletcher R.

[二次规划的若干算法研究]( <https://www.docin.com/p-892892104.html>)

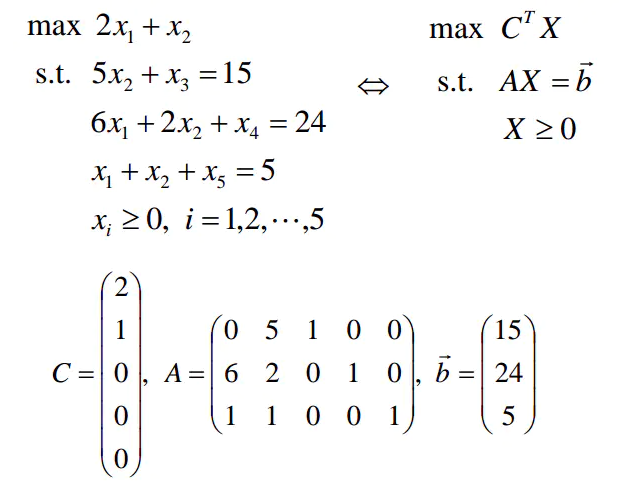
# 附录一：单纯形法样例

单纯形法具体例子：

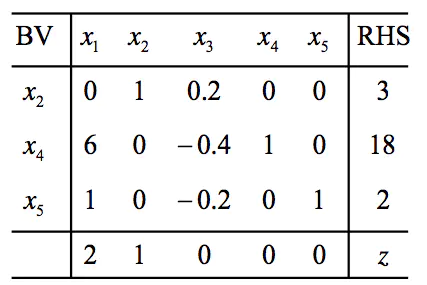
在一系列约束条件下求目标函数的最值



先化为标准型，即增加人工变量把不等式转化为等式

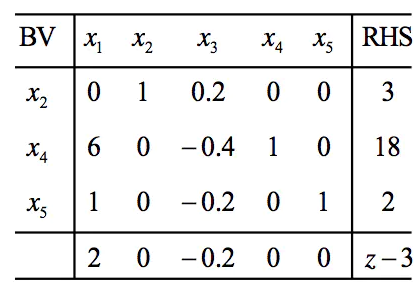


得单纯形表

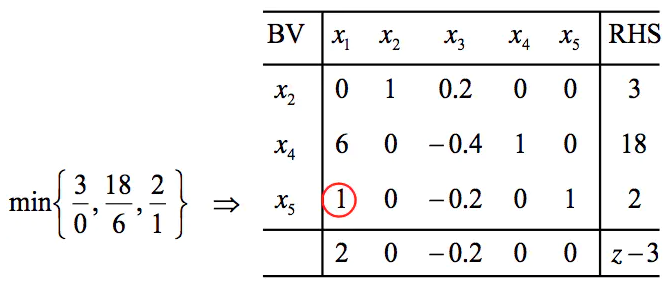


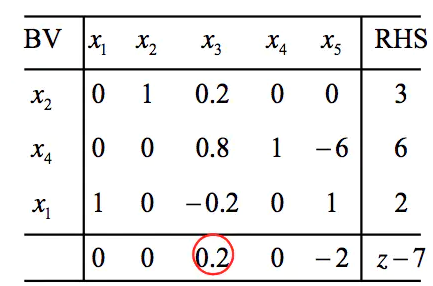
最左边一列是基变量，最右边一列是约束右边的常数项，中间是决策变量的系数。最下边一行是目标函数。

通过行变换将最后一行的基变量前面系数变成0，得



选基和换基







附录2：问题记录

1. 求解器的求解原理
2. 约束背后的数学原理
3. 求解器使用了哪些数学理论基础
4. 数学知识、公式推导
5. 物理引擎的设计框架
6. IK
7. 欧拉方法还是Verlet Integration