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 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

2. 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

2.1. LCP/MLCP 求解

2.1.1. Gauss-Seidel (GS 迭代求解法)

矩阵

$$\mathrm{A} = \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nn} \end{bmatrix} = \begin{bmatrix} a_{11} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & a_{nn} \end{bmatrix} - \begin{bmatrix} 0 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ -a_{n1} & \cdots & 0 \end{bmatrix} - \begin{bmatrix} 0 & \cdots & -a_{1n} \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 0 \end{bmatrix}$$

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

$$x_{k+1} = [b - (L + U)x_k] / D$$

Algorithm 3 Approximately solve $Ax = b$ given x^0

$x = x^0$
for $iter = 1$ to iteration limit **do**
  for $i = 1$ to n **do**
    $\Delta x_i = \left[b_i - \sum_{j=1}^{n} A_{ij} x_j\right] / A_{ii}$
    $x_i = x_i + \Delta x_i$
  end for
end for

2.1.2. 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）

2.1.3. Lemke (互补枢轴算法)

$$Ax - b = 0$$
$$x^T(Ax - b) \geq 0$$
$$x^T Ax \geq x^T b$$

（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：

$$\min f(x) = \frac{1}{2}x^T H x + c^T x$$
$$\text{s.t. } Ax \leq b$$
$$x \geq 0$$

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）

$$(LCP)\begin{cases} w - Mz = q \\ w, z \geq 0 \\ w^T z = 0 \end{cases} . \qquad \begin{matrix} w - Mz - Iz_0 = q \\ w, z \geq 0, z_0 \geq 0 \\ w^T z = 0 \end{matrix} ,$$

引入人工变量z_0，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，z_0对应的列为主列，主元消去，令$y_s = z_s$

(2) 设现行表中变量y_s下面的列为d_s，若d_s <= 0，退出；否则按最小比值规则确定指标 r，使 $q_i/d_{rs} = min\{q_i/d_{rs}|\, d_{rs} > 0\}$,如果 r 行的基变量是z_0，转 4；否则转 3

(3) 设 r 行的基变量为w_l或z_l ($l \neq s$) ，y_s进基，以 r 行为主行，y_s对应的列为主列，主元消去。如果离基变量是w_l，则令$y_s = z_l$；如果离基变量是z_l，则令$y_s = w_l$，转 2

(4) y_s进基，z_0离基。以 r 行为主行，y_s对应的列为主列，主元消去。得可行解，退出

具体样例：

$$\begin{aligned} &\min f(x) = 2x_1^2 - 2x_1x_2 + 10x_2^2 - x_1 - 10x_2 \\ &s.t. \quad -3x_1 - 2x_2 \geq -6 \\ &\qquad\quad x_1, x_2 \geq 0 \end{aligned}$$

$$M = \begin{bmatrix} H & -A^T \\ A & 0 \end{bmatrix} = \begin{pmatrix} 2 & -1 & 3 \\ -1 & 10 & 2 \\ -3 & -2 & 0 \end{pmatrix}, \quad q = \begin{bmatrix} c \\ -b \end{bmatrix} = \begin{pmatrix} -1 \\ -10 \\ 6 \end{pmatrix}$$

$$\begin{pmatrix} w_1 \\ w_2 \\ w_3 \end{pmatrix} = \begin{pmatrix} 2 & -1 & 3 \\ -1 & 10 & 2 \\ -3 & -2 & 0 \end{pmatrix} \begin{pmatrix} z_1 \\ z_2 \\ z_3 \end{pmatrix} + \begin{pmatrix} -1 \\ -10 \\ 6 \end{pmatrix}$$

$$w^T z = 0, \quad w \geq 0, \quad z \geq 0, \quad w, z \in R^3$$

Tab.3-2 The initial table of classical Lemke algorithm

基向量	w_1	w_2	w_3	z_1	z_2	z_3	z_0	q
w_1	1	0	0	−2	1	−3	−1	−1
w_2	0	1	0	1	−10	−2	−1	−10
w_3	0	0	1	3	2	0	−1	6

表 3.3 经典的 Lemke 算法的迭代表 1

Tab.3-3 The first iterated table of classical Lemke algorithm

基向量	w_1	w_2	w_3	z_1	z_2^*	z_3	z_0	q
w_1	1	−1	0	−3	11^*	−1	0	9
w_2	0	−1	0	−1	10	2	1	10
w_3	0	−1	1	2	12	2	0	16

表 3.4 经典的 Lemke 算法的迭代表 2

Tab.3-4 The second iterated table of classical Lemke algorithm

基向量	w_1	w_2	w_3	z_1	z_2	z_3	z_0	q
z_2	1/11	−1/11	0	−3/11	1	−1/11	0	9/11
z_0	−10/11	−1/11	0	$19/11^*$	0	32/11	1	20/11
w_3	−12/11	1/11	1	58/11	0	34/11	0	68/11

表 3.5 经典的 Lemke 算法的迭代表 3

Tab.3-5 The third iterated table of classical Lemke algorithm

基向量	w_1	w_2	w_3	z_1	z_2	z_3	z_0	q
z_2	−1/19	−2/19	0	0	1	7/19	3/19	21/19
z_1	−10/19	−1/19	0	1	0	32/19	11/19	20/19
w_3	32/19	7/19	1	0	0	−110/19	−58/19	12/19

2.1.4. 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。

3. 物理约束式子的由来

3.1. 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)	m_i	标量	
惯性张量(转动惯量)	I_i	二阶矩阵	
力 (force)	$\overrightarrow{F_i}$	矢量	$\overrightarrow{F_i} = m_i \overrightarrow{a_i}$
力矩 (torque)	$\overrightarrow{\tau_i}$	矢量	$\overrightarrow{\tau_i} = r \times \overrightarrow{F_i}$，其中$r$为质心到指定点的矢量
线速度 (velocity)	$\overrightarrow{v_i}$	矢量	$\mathrm{d}\,\overrightarrow{v_i} = \overrightarrow{a_i}\ \mathrm{dt}$
角速度	$\overrightarrow{\omega_i}$	矢量	$\mathrm{d}\,\overrightarrow{\omega_i} = \overrightarrow{\alpha_i}\,dt$
线加速度 (acceleration)	$\overrightarrow{a_i}$	矢量	
角加速度	$\overrightarrow{\alpha_i}$	矢量	
动量 (momentum)	$\vec{P}$	矢量	$m_i\,\overrightarrow{v_i}$
角动量	$\vec{L}$	矢量	$I_i\,\overrightarrow{\omega_i}$
冲量 (impulse)	*I*		$\int \overrightarrow{F_i}\,dt = \vec{P} = m_i\,\overrightarrow{v_i}$
角冲量			$\int \overrightarrow{\tau_i}\,dt = \vec{L} = I_i\,\overrightarrow{\omega_i}$
动能	*E*	标量	$1/2\ m_i\,\overrightarrow{v_i}^2$
转动动能	*E*	标量	$1/2\ I_i\,\overrightarrow{\omega_i}^2$
质心位置 (position)	$\overrightarrow{r_i}$	矢量	$\mathrm{d}\,\overrightarrow{r_i} = \overrightarrow{v_i}\,dt$
位置朝向 (orientation)	$\overrightarrow{q_i}$	四元数	$\overrightarrow{q_i} = [s_i, x_i, y_i, z_i]$ $\mathrm{d}\,\overrightarrow{q_i} = 1/2\,[0, \overrightarrow{\omega_i}]\,\overrightarrow{q_i}\ \mathrm{dt}$
广义位置	$\overrightarrow{s_i}$	矩阵	$\overrightarrow{s_i} = [\overrightarrow{r_i}, \overrightarrow{q_i}]$
广义速度 (status)	$\overrightarrow{u_i}$	矩阵	$\overrightarrow{u_i} = [\overrightarrow{v_i}, \overrightarrow{\omega_i}]^T$
广义合外力	$\overrightarrow{f_{ext}}$	矩阵	$\overrightarrow{f_{ext}} = [\overrightarrow{F_i}, \overrightarrow{\tau_i} - \overrightarrow{\omega_i} \times (I_i\overrightarrow{\omega_i})]$
法线 (normal)	$\overrightarrow{n_i}$	矢量	
摩擦系数	μ_i	标量	

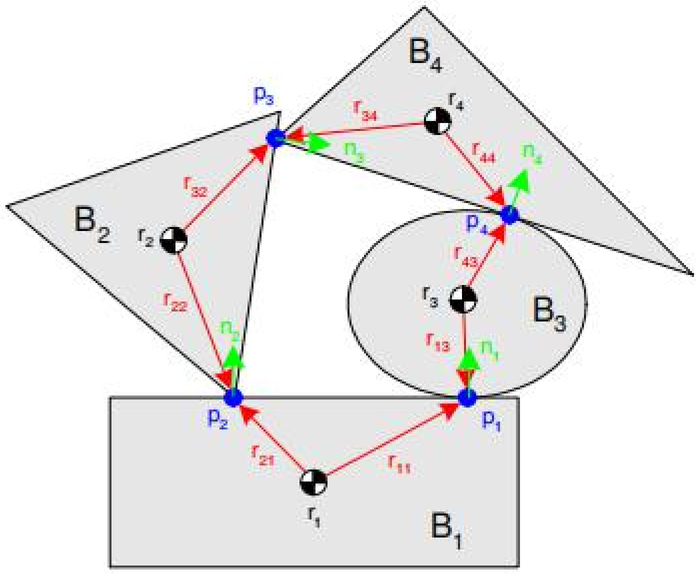
3.2. 根据动量守恒定律推导约束方程式

3.2.1. 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$\overrightarrow{u_i}$，求经过Δt 时刻后的广义速度。



整个场景中，对于第 k 个接触点$\overrightarrow{p_k}$，两边接触的物体记为i_k和j_k，记$\overrightarrow{f_k} = f_k\,\overrightarrow{n_k}$为第 k 个接触点的接触力，$\vec{f}_i^{ext}$为物体 i 所受的其它外力，如重力等；$\overrightarrow{r_{ki}} = \overrightarrow{p_k} - \overrightarrow{r_i}$为质心到接触点的矢量，$\overrightarrow{\tau}_i^{ext}$为物体 i 所受的其它外力矩。
则对第 i 个物体有

$$\dot{\vec{r}}_i = \vec{v}_i$$
$$\dot{q}_i = \frac{1}{2}\vec{\omega}_i q_i$$
$$\dot{v}_i = m_i^{-1}\sum_{j_k=i}\vec{f}_k - m_i^{-1}\sum_{i_k=i}\vec{f}_k + m_i^{-1}\vec{f}_i^{\ \mathrm{ext}}$$
$$\dot{\vec{\omega}}_i = I_i^{-1}\sum_{j_k=i}\vec{r}_{kj}\times\vec{f}_k - I_i^{-1}\sum_{i_k=i}\vec{r}_{ki}\times\vec{f}_k - I_i^{-1}\vec{\omega}_i\times I_i\vec{\omega}_i + I_i^{-1}\vec{\tau}_i^{\ \mathrm{ext}}$$

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：$q_1 = [a_1, \overrightarrow{v_1}]$，$q_2 = [a_2, \overrightarrow{v_2}]$，则
$q_1 \boxtimes q_2 = [a_1 a_2 - \overrightarrow{v_1}\cdot\overrightarrow{v_2}, a_1\overrightarrow{v_2} + a_2\overrightarrow{v_1} + \overrightarrow{v_1}\times\overrightarrow{v_2}]$

旋转角度 θ，旋转轴 $\vec{n}$，以不带矢量箭头的 ω 表示角速度的模长，则任一四元数都可以写成：$q = [\cos\left(\frac{\theta}{2}\right), \vec{n}\sin\left(\frac{\theta}{2}\right)] = [\cos\left(\frac{\omega t}{2}\right), \frac{\vec{\omega}}{\omega}\sin\left(\frac{\omega t}{2}\right)]$
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

$$\dot{q} = [-\frac{\omega}{2}\sin\left(\frac{\omega t}{2}\right), \frac{\vec{\omega}}{\omega}\frac{\omega}{2}\cos\left(\frac{\omega t}{2}\right)] = \frac{1}{2}[-\omega\sin\left(\frac{\omega t}{2}\right), \vec{\omega}\cos\left(\frac{\omega t}{2}\right)]$$
$$= \frac{1}{2}[0, \vec{\omega}] \boxtimes q$$

后两个式子的原型为:

$$\overrightarrow{F_i} = m_i\overrightarrow{a_i} = \sum_{j_k=i}\overrightarrow{f_k} - \sum_{i_k=i}\overrightarrow{f_k} + \vec{f}_i^{ext}$$

$$\overrightarrow{\tau_i} = I_i\overrightarrow{\alpha_i} = \sum_{j_k=i}\overrightarrow{r_{kj}}\times\overrightarrow{f_k} - \sum_{i_k=i}\overrightarrow{r_{ki}}\times\overrightarrow{f_k} - \overrightarrow{\omega_i}\times I_i\overrightarrow{\omega_i} + \vec{\tau}_i^{ext}$$

记无法直接获得的接触力 $\vec{f} = [\overrightarrow{f_1}, \overrightarrow{f_2}, \ldots, \overrightarrow{f_n}]$，则上述两个式子可以表示成:

$$
\begin{bmatrix} \dot{\vec{r_1}} \\ \dot{\vec{q_1}} \\ \vdots \\ \dot{\vec{r_n}} \\ \dot{\vec{q_n}} \end{bmatrix} = \begin{bmatrix} 1 & & & \\ & \frac{1}{2}\begin{bmatrix} -x_1 & -y_1 & -z_1 \\ s_1 & z_1 & -y_1 \\ -z_1 & s_1 & x_1 \\ y_1 & -x_1 & s_1 \end{bmatrix} & \cdots & \\ & \vdots & \ddots & \vdots \\ & & 1 & \\ & & \cdots & \frac{1}{2}\begin{bmatrix} -x_n & -y_n & -z_n \\ s_n & z_n & -y_n \\ -z_n & s_n & x_n \\ y_n & -x_n & s_n \end{bmatrix} \end{bmatrix} \begin{bmatrix} \vec{v_1} \\ \vec{\omega_1} \\ \vdots \\ \vec{v_n} \\ \vec{\omega_n} \end{bmatrix}
$$

$$
\begin{bmatrix} \dot{\vec{v_1}} \\ \dot{\vec{\omega_1}} \\ \vdots \\ \dot{\vec{v_n}} \\ \dot{\vec{\omega_n}} \end{bmatrix} = \begin{bmatrix} m_1 & & & & \\ & I_1 & & & \\ & & \ddots & & \\ & & & m_n & \\ & & & & I_n \end{bmatrix}^{-1} \left([C] \begin{bmatrix} \vec{n_1} & & & & \\ & \vec{n_2} & & & \\ & & \ddots & & \\ & & & \overrightarrow{n_{n-1}} & \\ & & & & \vec{n_n} \end{bmatrix} \begin{bmatrix} \vec{f_1} \\ \vec{f_2} \\ \vdots \\ \overrightarrow{f_{n-1}} \\ \vec{f_n} \end{bmatrix} + \overrightarrow{f_{ext}} \right)
$$

简记为

$$
\begin{aligned}
\dot{\vec{s}} &= \mathbf{S}\vec{u} \\
\dot{\vec{u}} &= \mathbf{M}^{-1}\left(\mathbf{CN}\vec{f} + \vec{f}_{ext}\right).
\end{aligned}
$$

其中位置朝向$\vec{q_i} = [s_i, x_i, y_i, z_i]$，$\overrightarrow{f_{ext}} = \left[\vec{f}_1^{ext}, \overrightarrow{\tau_1} - \overrightarrow{\omega_1} \times (I_1\overrightarrow{\omega_1}), \ldots, \vec{f}_n^{ext}, \overrightarrow{\tau_n} - \overrightarrow{\omega_n} \times (I_n\overrightarrow{\omega_n})\right]$

$$
\mathbf{Q}_i = \frac{1}{2}\begin{bmatrix} -x_i & -y_i & -z_i \\ s_i & z_i & -y_i \\ -z_i & s_i & x_i \\ y_i & -x_i & s_i \end{bmatrix}.
$$

满足 $\frac{1}{2}\overrightarrow{\omega_i}\vec{q_i} = Q_i\overrightarrow{\omega_i}$

$$
\mathbf{S} = \begin{bmatrix} \mathbf{1} & & & & \mathbf{0} \\ & \mathbf{Q}_1 & & & \\ & & \ddots & & \\ & & & \mathbf{1} & \\ \mathbf{0} & & & & \mathbf{Q}_n \end{bmatrix};
$$

广义质量矩阵

$$\mathbf{M}=\begin{bmatrix} m_1\mathbf{1} & & & & \mathbf{0} \\ & \mathbf{I}_1 & & & \\ & & \vdots & & \\ & & & m_n\mathbf{1} & \\ \mathbf{0} & & & & \mathbf{I}_n \end{bmatrix}$$

接触点法线矩阵

$$\mathbf{N}=\begin{bmatrix} \vec{n}_1 & & & \mathbf{0} \\ & \vec{n}_2 & & \\ & & \ddots & \\ \mathbf{0} & & & \vec{n}_k \end{bmatrix},$$

$$\mathbf{C}_{lk}=\begin{cases} -\mathbf{1} & \text{for } l=2i_k-1 \\ -\mathbf{r}_{ki_k}^{\times} & \text{for } l=2i_k \\ \mathbf{1} & \text{for } l=2j_k-1 \\ \mathbf{r}_{kj_k}^{\times} & \text{for } l=2j_k \\ \mathbf{0} & \text{otherwise} \end{cases}.$$

$$\mathbf{r}^{\times}=\begin{bmatrix} 0 & -r_3 & r_2 \\ r_3 & 0 & -r_1 \\ -r_2 & r_1 & 0 \end{bmatrix}$$

采用欧拉积分的方式，在 $t+\Delta t$ 时刻，有

$$\vec{s}^{t+\Delta t}=\vec{s}^{t}+\Delta t\mathbf{S}\vec{u}^{t+\Delta t},$$
$$\vec{u}^{t+\Delta t}=\vec{u}^{t}+\Delta t\mathbf{M}^{-1}\left(\mathbf{CN}\vec{f}^{t+\Delta t}+\vec{f}_{ext}\right).$$

转化成 LCP 形式

接触点$\overrightarrow{p_k}$，两边接触的物体i_k和j_k，$\overrightarrow{r_{ki}}=\overrightarrow{p_k}-\overrightarrow{r_i}$为质心到接触点的矢量，则物体在接触点的速度为：$\overrightarrow{v_{i_k}}+\overrightarrow{\omega_{i_k}}\times\overrightarrow{r_{ki_k}}$和$\overrightarrow{v_{j_k}}+\overrightarrow{\omega_{j_k}}\times\overrightarrow{r_{kj_k}}$。

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

$$\overrightarrow{n_k}(\overrightarrow{v_{i_k}}+\overrightarrow{\omega_{i_k}}\times\overrightarrow{r_{ki_k}}-\overrightarrow{v_{j_k}}+\overrightarrow{\omega_{j_k}}\times\overrightarrow{r_{kj_k}})$$

为$N^TC^T\overrightarrow{u}$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 $t+\Delta t$ 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

$$
\begin{aligned}
&\boldsymbol{N}^T\boldsymbol{C}^T\vec{u}^{t+\Delta t} \geq \vec{0},\\
&\vec{f}^{t+\Delta t} \geq \vec{0},\\
&(\boldsymbol{N}^T\boldsymbol{C}^T\vec{u}^{t+\Delta t})^T\vec{f}^{t+\Delta t} = 0.
\end{aligned}
$$

即

$$
\begin{aligned}
&\boldsymbol{N}^T\boldsymbol{C}^T\boldsymbol{M}^{-1}\boldsymbol{C}\boldsymbol{N}\Delta t\vec{f}^{t+\Delta t} + \boldsymbol{N}^T\boldsymbol{C}^T(\vec{u}^t + \Delta t\boldsymbol{M}^{-1}\vec{f}_{ext}) \geq \vec{0},\\
&\Delta t\vec{f}^{t+\Delta t} \geq \vec{0},\\
&\Delta t\vec{f}^{t+\Delta t} \cdot (\boldsymbol{N}^T\boldsymbol{C}^T\boldsymbol{M}^{-1}\boldsymbol{C}\boldsymbol{N}\Delta t\vec{f}^{t+\Delta t} + \boldsymbol{N}^T\boldsymbol{C}^T(\vec{u}^t + \Delta t\boldsymbol{M}^{-1}\vec{f}_{ext})) = 0
\end{aligned}
$$

$$
\boldsymbol{A} = \boldsymbol{N}^T\boldsymbol{C}^T\boldsymbol{M}^{-1}\boldsymbol{C}\boldsymbol{N}\,,\quad \vec{x} = \Delta t\vec{f}^{t+\Delta t}\,,\quad \vec{b} = \boldsymbol{N}^T\boldsymbol{C}^T(\vec{u}^t + \Delta t\boldsymbol{M}^{-1}\vec{f}_{ext})
$$

而书上多了$\frac{\vec{\nu}}{\Delta t}$

$$
\mathbf{N}^T\mathbf{C}^T\vec{u}^{t+\Delta t} \geq \frac{\vec{\nu}}{\Delta t} \quad \text{compl. to} \quad \vec{f} \geq 0,
$$

$$
\mathbf{N}^T\mathbf{C}^T\left(\vec{u}^t + \Delta t\mathbf{M}^{-1}\left(\mathbf{C}\mathbf{N}\vec{f}^{t+\Delta t} + \vec{f}_{ext}\right)\right) - \frac{\vec{\nu}}{\Delta t} \geq 0.
$$

最终可以表示成 $Ax + b \geq 0, x \geq 0$ 的方程组形式。

$$
\underbrace{\mathbf{N}^T\mathbf{C}^T\mathbf{M}^{-1}\mathbf{C}\mathbf{N}}_{\mathbf{A}}\underbrace{\Delta t\vec{f}^{t+\Delta t}}_{\vec{x}} + \underbrace{\mathbf{N}^T\mathbf{C}^T\left(\vec{u}^t + \Delta t\mathbf{M}^{-1}\vec{f}_{ext}\right) - \frac{\vec{\nu}}{\Delta t}}_{\vec{b}} \geq 0,
$$

3.2.2. 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 $D_k = [\overrightarrow{d_{1_k}}, \overrightarrow{d_{2_k}}, \ldots, \overrightarrow{d_{n_k}}]$

摩擦力大小 $\overrightarrow{\beta_k} = [\beta_{1_k}, \ldots, \beta_{n_k}]$

则接触力$\overrightarrow{f_k} = f_k\,\overrightarrow{n_k} + D_k\,\overrightarrow{\beta_k}$

$$
\dot{\vec{u}} = \mathbf{M}^{-1}\left(\mathbf{C}\left(\mathbf{N}\vec{f} + \mathbf{D}\vec{\beta}\right) + \vec{f}_{ext}\right).
$$

$$
\vec{u}^{\,t+\Delta t} = \vec{u}^{\,t} + \Delta t\mathbf{M}^{-1}\left(\mathbf{C}\left(\mathbf{N}\vec{f} + \mathbf{D}\vec{\beta}\right) + \vec{f}_{ext}\right).
$$

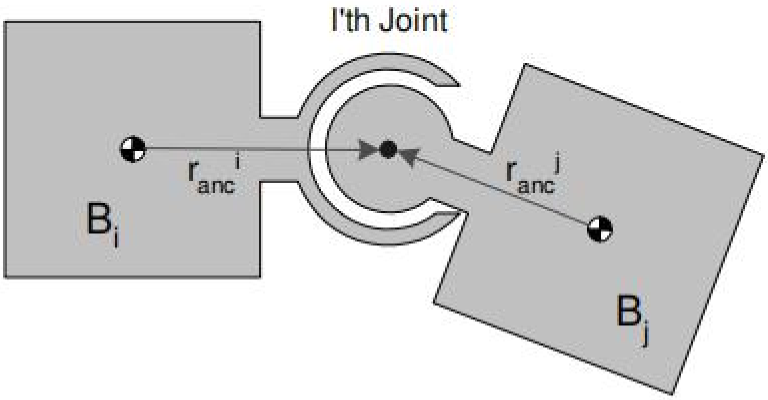
$$
\begin{bmatrix} \mathbf{D}^T\mathbf{C}^T\mathbf{M}^{-1}\mathbf{C}\mathbf{D} & \mathbf{D}^T\mathbf{C}^T\mathbf{M}^{-1}\mathbf{C}\mathbf{N} & \mathbf{E} \\ \mathbf{N}^T\mathbf{C}^T\mathbf{M}^{-1}\mathbf{C}\mathbf{D} & \mathbf{N}^T\mathbf{C}^T\mathbf{M}^{-1}\mathbf{C}\mathbf{N} & \mathbf{0} \\ -\mathbf{E}^T & \mu & \mathbf{0} \end{bmatrix} \cdot \begin{bmatrix} \Delta t\vec{\beta} \\ \Delta t\vec{f} \\ \vec{\lambda}_{\text{aux}} \end{bmatrix}
+ \begin{bmatrix} \mathbf{D}^T\mathbf{C}^T\left(\vec{u}^{\,t} + \Delta t\mathbf{M}^{-1}\vec{f}_{\text{ext}}\right) \\ \mathbf{N}^T\mathbf{C}^T\left(\vec{u}^{\,t} + \Delta t\mathbf{M}^{-1}\vec{f}_{\text{ext}}\right) - \frac{\vec{\nu}}{\Delta t} \\ 0 \end{bmatrix} \geq 0
$$

$$
\text{compl. to} \quad \begin{bmatrix} \Delta t\vec{\beta} \\ \Delta t\vec{f} \\ \vec{\lambda}_{\text{aux}} \end{bmatrix} \geq 0,
$$

最终同样可以表示成 $Ax + b \geq 0, x \geq 0$ 的方程组形式。

3.2.3. 非接触约束方程式

以铰链约束为样例



$$
\underbrace{\left[\left(\boldsymbol{r}_i + \boldsymbol{R}(q_i)\boldsymbol{r}^i_{\text{anc}}\right) - \left(\boldsymbol{r}_j + \boldsymbol{R}(q_j)\boldsymbol{r}^j_{\text{anc}}\right)\right]_x}_{\Phi_1} = 0,
$$

$$
\underbrace{\left[\left(\boldsymbol{r}_i + \boldsymbol{R}(q_i)\boldsymbol{r}^i_{\text{anc}}\right) - \left(\boldsymbol{r}_j + \boldsymbol{R}(q_j)\boldsymbol{r}^j_{\text{anc}}\right)\right]_y}_{\Phi_2} = 0,
$$

$$
\underbrace{\left[\left(\boldsymbol{r}_i + \boldsymbol{R}(q_i)\boldsymbol{r}^i_{\text{anc}}\right) - \left(\boldsymbol{r}_j + \boldsymbol{R}(q_j)\boldsymbol{r}^j_{\text{anc}}\right)\right]_z}_{\Phi_3} = 0,
$$

$$\mathbf{\Phi}(\boldsymbol{s}) = \begin{bmatrix} \Phi_1(\boldsymbol{s}) \\ \Phi_2(\boldsymbol{s}) \\ \Phi_3(\boldsymbol{s}) \end{bmatrix} = 0.$$

其中 s 是上面出现过的广义位置$\vec{s_i} = [\vec{r_i}, \vec{q_i}]$。上式同样对时间求导，得

$$\begin{aligned} \frac{d}{dt}\mathbf{\Phi}_l(\boldsymbol{s}) &= \frac{\partial \mathbf{\Phi}}{\partial \boldsymbol{s}} \frac{d\boldsymbol{s}}{dt} \\ &= \underbrace{\frac{\partial \mathbf{\Phi}}{\partial \boldsymbol{s}} \boldsymbol{S}}_{\boldsymbol{J}_\Phi} \boldsymbol{u} \\ &= 0. \end{aligned}$$

$$\begin{aligned} \frac{d^2}{dt^2}\mathbf{\Phi}(\boldsymbol{s}(t)) &= \frac{d}{dt}(\boldsymbol{J}_\Phi \boldsymbol{u}) \\ &= \frac{d}{dt}(\boldsymbol{J}_\Phi)\, \boldsymbol{u} + \boldsymbol{J}_\Phi \frac{d}{dt}(\boldsymbol{u}) \\ &= 0, \end{aligned}$$

可得 $$\boldsymbol{J}_\Phi \dot{\boldsymbol{u}} = -\dot{\boldsymbol{J}}_\Phi \boldsymbol{u}.$$

根据这里的约束力不做功 $\vec{\mathrm{f}} \cdot \vec{s_i} = 0$，两边对时间求导得 $\vec{\mathrm{f}} \cdot \vec{u_i} = 0$，则约束力

可以表示成 Jocabian 矩阵的广义倍数 $$\boldsymbol{F}_\Phi = \boldsymbol{J}_\Phi^T \boldsymbol{\lambda}_\Phi.$$

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

$$\boldsymbol{C}(\boldsymbol{N}\boldsymbol{f} + \boldsymbol{D}\boldsymbol{\beta}) \equiv \boldsymbol{J}_{\text{contact}}^T \boldsymbol{\lambda}_{\text{contact}}.$$

$$\begin{aligned} \boldsymbol{C}(\boldsymbol{N}\boldsymbol{f} + \boldsymbol{D}\boldsymbol{\beta}) &= \boldsymbol{C}\boldsymbol{N}\boldsymbol{f} + \boldsymbol{C}\boldsymbol{D}\boldsymbol{\beta} \\ &= \underbrace{\begin{bmatrix} \boldsymbol{C}\boldsymbol{N} & \boldsymbol{C}\boldsymbol{D} \end{bmatrix}}_{\pi\left(\boldsymbol{J}_{\text{contact}}^T\right)} \underbrace{\begin{bmatrix} \boldsymbol{f} \\ \boldsymbol{\beta} \end{bmatrix}}_{\pi(\boldsymbol{\lambda}_{\text{contact}})} \\ &= \pi\left(\boldsymbol{J}_{\text{contact}}^T \boldsymbol{\lambda}_{\text{contact}}\right), \end{aligned}$$

各种约束组合方程式

$$\underbrace{\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} \lambda_{\text{joint}} \\ \lambda_{\text{contact}} \\ \lambda_{\text{limit}} \\ \lambda_{\text{motor}} \\ \lambda_{\text{aux}} \end{bmatrix}}_{\lambda} + \underbrace{\begin{bmatrix} J_{\text{joint}} \left(u + \Delta t M^{-1} f_{\text{ext}} \right) - b_{\text{joint}} \\ J_{\text{contact}} \left(u + \Delta t M^{-1} f_{\text{ext}} \right) - b_{\text{contact}} \\ J_{\text{limit}} \left(u + \Delta t M^{-1} f_{\text{ext}} \right) - b_{\text{limit}} \\ J_{\text{motor}} \left(u + \Delta t M^{-1} f_{\text{ext}} \right) - b_{\text{motor}} \\ b_{\text{aux}} \end{bmatrix}}_{b} \geq 0,$$

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

$$\begin{aligned} \max \quad & 2x_1 + x_2 \\ \text{s.t.} \quad & 5x_2 \leq 15 \\ & 6x_1 + 2x_2 \leq 24 \\ & x_1 + x_2 \leq 5 \\ & x_1 \geq 0, x_2 \geq 0 \end{aligned}$$

先化为标准型，即增加人工变量把不等式转化为等式

$$\begin{aligned} \max\ & 2x_1 + x_2 \\ \text{s.t.}\ & 5x_2 + x_3 = 15 \\ & 6x_1 + 2x_2 + x_4 = 24 \\ & x_1 + x_2 + x_5 = 5 \\ & x_i \ge 0,\ i = 1,2,\cdots,5 \end{aligned} \quad \Leftrightarrow \quad \begin{aligned} \max\ & C^T X \\ \text{s.t.}\ & AX = \vec{b} \\ & X \ge 0 \end{aligned}$$

$$C = \begin{pmatrix} 2 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix},\ A = \begin{pmatrix} 0 & 5 & 1 & 0 & 0 \\ 6 & 2 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 \end{pmatrix},\ \vec{b} = \begin{pmatrix} 15 \\ 24 \\ 5 \end{pmatrix}$$

得单纯形表

BV	x_1	x_2	x_3	x_4	x_5	RHS
x_2	0	1	0.2	0	0	3
x_4	6	0	-0.4	1	0	18
x_5	1	0	-0.2	0	1	2
	2	1	0	0	0	z

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

BV	x_1	x_2	x_3	x_4	x_5	RHS
x_2	0	1	0.2	0	0	3
x_4	6	0	-0.4	1	0	18
x_5	1	0	-0.2	0	1	2
	2	0	-0.2	0	0	$z-3$

选基和换基

$$\min\left\{\frac{3}{0},\frac{18}{6},\frac{2}{1}\right\} \Rightarrow$$

BV	x_1	x_2	x_3	x_4	x_5	RHS
x_2	0	1	0.2	0	0	3
x_4	6	0	-0.4	1	0	18
x_5	1	0	-0.2	0	1	2
	2	0	-0.2	0	0	$z-3$

BV	x_1	x_2	x_3	x_4	x_5	RHS
x_2	0	1	0.2	0	0	3
x_4	0	0	0.8	1	-6	6
x_1	1	0	-0.2	0	1	2
	0	0	0.2	0	-2	$z-7$

BV	x_1	x_2	x_3	x_4	x_5	RHS
x_2	0	1	0	-0.25	1.5	1.5
x_3	0	0	1	1.25	-7.5	7.5
x_1	1	0	0	0.25	-0.5	3.5
	0	0	0	-0.25	-0.5	$z-8.5$

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