

碰撞检测方法	2
第一节 综述	2
第二节 基于数值优化的碰撞检测	2
第三节 平面三角形间的最短距离求解	3
第四节 平面椭圆间的最短距离求解	4
第五节 总结	7
附录 1: KKT 方程求解的条件(避坑 1)	8
附录 2: 例子 2 的 Jacobi 矩阵	9
附录 3: 例子 2 中的错误表达方式(避坑 2)	11
附录 4: 椭圆间最短距离求解的其他方法	11
附录 5: KKT 矩阵的不定性说明	11
参考文献	12

碰撞检测方法

第一节 综述

碰撞检测综述。碰撞检测几乎是基于物理模拟中的一个必须环节，有碰撞检测才能计算后续物体的相互作用。碰撞检测的综述性文献请参看[1]。就检测方法说，可以通过基于几何的方法(通过检测空间是否有相交区域来判断碰撞)，也可以基于代数的方法(即计算物体之间的最小距离)。目前尚未见到碰撞检测方法在速度和所需计算资源对比的文章(话也说回来，关注此方向的人少，文章自然少)。

基于几何方法的碰撞检测简述。使用空间图形方式完成碰撞检测，已经有书籍[2]，算法有 OBB, GJK 等。在三维空间中实现 OBB 的碰撞检测的 C++代码见[3]。在二维空间中，OBB 算法是将两个检测物体的最大位置坐标和最小位置坐标投影在每个轴上，检查是否有重叠，有重叠，则相交；在三维空间中，则需要检查 15 个轴向(具体的算法见[8])，检测的内容与二维情况下相同。

而使用最优化方法进行检测的方法和文章不多，其中[13]提供了多面体之间距离求解的方法和实例代码，但讲解过于简略；部分文章，见[4]，思维跳跃太大；本章就对基于最优化的方法进行介绍。

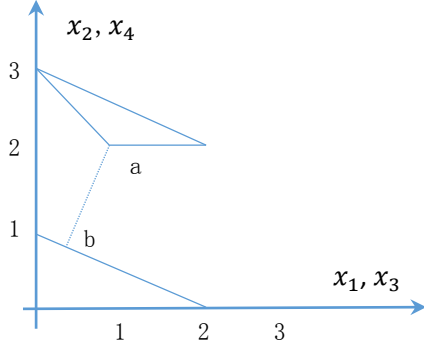
一个需要说明的特例。目前基于物理的仿真中，物质点方法(模拟雪，高速物体的碰撞)在图形学中的优势，是其他方法所不具备的，而计算物质之间的碰撞就不进行检测(就当前娱乐用计算机的算力来说，不可能对物质点进行碰撞检测并计算质点的响应)，其核心思想是在两套坐标系(欧拉视角和拉格朗日视角)中交换两个碰撞体之间的动量。

第二节 基于数值优化的碰撞检测

- 一、碰撞检测的要求。(1)检测是否碰撞，(2)碰撞点坐标，(3)分离方向(对刚体而言)。
- 二、(1)对于检测是否碰撞，根据物体之间的最小距离判断是否碰撞。使用数值方法能直观提供依据。(2)求解出的决策变量也就是碰撞点的坐标。(3)分离方向也就是碰撞点产生作用力的方向，大小相等，分别作用在不同的物体上，具体的方向与最短距离线段的方向一致。在实际物理引擎中，一般不计算作用力，而是计算冲量(即速度与时间差，这个时间差，就是显示两帧之间的时间量)。此外碰撞中产生的摩擦力也是一个较大的话题，关于碰撞响应则和摩擦力(由于碰撞可能导致物体变形)，将另行讨论。
- 二、本小节讨论两个例子。(1)求解 2 维平面中两个三角形的最短距离，(2)求解 2 维平面上两个椭圆的最短距离。注意，使用最优化方法的碰撞检测必须保证物体为凸，以保证优化方法计算的可行性；如果物体非凸，可以将物体分解，保证每个碰撞物体为凸。
- 三、问题转换成数学模型。两个实例中，第一个实例在数学上是个二次规划问题(QP)，第二个问题则是个二次约束二次规划问题(QPQP)。当然可以使用二阶锥规划(SOCP)来解决 QCQP 问题。关于二次规划的综述性文献，请参看[12]
- 四、本章节的数学符号。大写黑体表示矩阵，如： \mathbf{Q} ；小写黑体表示向量，如： \mathbf{x} ；正常斜体表示标量，如： r 。所有向量以列方式表示，矩阵的转置或逆在变量的右上角增加 T 或 -1，向量的转置类似。
- 五、阅读本节所需的知识。高等数学，线性代数，数值方法，最优化。
- 六、由于两个实例在求解过程中，都要求解线性方程组，由于方程组的矩阵对称不定，在编写代码时请参看求解线性方程组的 GMRES 部分。

第三节 平面三角形间的最短距离求解

一、实例一，求解平面上两个三角形的最短距离，如图所示



- (1) 三角形如图所示，两个三角形的最短距离在点 a, b 处。
- (2) 最短距离为 1.341640786
- (3) 注意，在实际计算中，仅有一个顶点和一个线段起约束作用，其他未起。
- (4) 这也是不能使用等式约束的原因之一，详细见附录 1。

图 7.1

二、基本原理。

(1) 2 维平面上两个点的距离可以写成为 $f(\mathbf{x}) = (x_1 - x_3)^2 + (x_2 - x_4)^2$ 。其中 $[x_1 \ x_2]^T$ 属于第一个三角形， $[x_3 \ x_4]^T$ 属于第二个三角形，4 个点必须在两个三角形上，即满足约束方程式 7.1。

$$\begin{aligned} \min f(\mathbf{x}) &= (x_1 - x_3)^2 + (x_2 - x_4)^2 \\ \text{s. t. } &x_1 \geq 0 \\ &x_2 \geq 0 \\ &x_1 + x_2 \leq 2 \\ &x_4 \geq 0 \\ &x_3 + x_4 \geq 3 \\ &x_3 + 2x_4 \leq 6 \end{aligned} \quad \text{式 7.1}$$

(2) 化为标准型，即约束为大于等于符号，使用矩阵描述，见式 7.2。

$$\begin{aligned} \min f(\mathbf{x}) &= \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}^T \begin{bmatrix} 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \\ -1 & 0 & 1 & 0 \\ 0 & -1 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} \\ \text{s. t. } \mathbf{Ax} &\geq \mathbf{b} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ -1 & -2 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & -1 & -2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} \geq \begin{bmatrix} 0 \\ 0 \\ -2 \\ 2 \\ 3 \\ -6 \end{bmatrix} \end{aligned} \quad \text{式 7.2}$$

(3) 使用路径跟踪法求解，即求解下式方程(具体推导见附录 1)：

$$\begin{aligned} \mathbf{Hx} - \mathbf{A}^T \mathbf{y} + \mathbf{c} &= \mathbf{0} \\ \mathbf{Ax} - \mathbf{w} - \mathbf{b} &= \mathbf{0} \\ \mathbf{YWe} &= \mu \mathbf{e} \\ \mathbf{y}, \mathbf{w} &\geq \mathbf{0} \end{aligned} \quad \text{式 7.3}$$

(4) 其中涉及 3 个向量 \mathbf{y} , \mathbf{w} , \mathbf{e} , 和 2 个矩阵 \mathbf{Y} , \mathbf{W} 。其中 \mathbf{y} 是拉格朗日乘子， \mathbf{w} 是松弛变量， \mathbf{e} 是单位向量， \mathbf{Y} 是 \mathbf{y} 向量对角化的矩阵， \mathbf{W} 是 \mathbf{w} 向量对角化的矩阵。即 $\mathbf{Y} = \text{diag}(\mathbf{y})$, $\mathbf{W} = \text{diag}(\mathbf{w})$ ，最终求解式 7.4 线性方程组。例子中，需要计算拉氏乘子，因此方程维数为 10 维。即 4 个待求变量，和 6 个约束方程和 6 个拉氏乘子。

$$\begin{bmatrix} -\mathbf{H} & \mathbf{A}^T \\ \mathbf{A} & \mathbf{Y}^{-1}\mathbf{W} \end{bmatrix} \begin{bmatrix} \Delta \mathbf{x} \\ \Delta \mathbf{y} \end{bmatrix} = \begin{bmatrix} \mathbf{c} + \mathbf{H}\mathbf{x} - \mathbf{A}^T\mathbf{y} \\ \mathbf{b} - \mathbf{A}\mathbf{x} + \mu\mathbf{Y}^{-1}\mathbf{e} \end{bmatrix} \quad \text{式 7.4}$$

(5) 方程组的矩阵的特征值为(第一次迭代), 3.1694, 2.0935, 1.4959, 1.1056, 1.0, 1.0, -0.4655, -2.1457, -4.1369, -5.1164, 不能使用 CG 等方法。需要注意; 而 $\Delta \mathbf{w}$ 的计算由式 7.5 得到

$$\Delta \mathbf{w} = \mathbf{Y}^{-1}(\mu \mathbf{e} - \mathbf{Y}\mathbf{W}\mathbf{e} - \mathbf{W}\Delta \mathbf{y}) \quad \text{式 7.5}$$

(6) 三个变量 ($\Delta \mathbf{x}$, $\Delta \mathbf{y}$, $\Delta \mathbf{w}$) 的步长由式 7.6 决定,

$$\lambda = \min\left(p \left[\max\left(\frac{\Delta \mathbf{y}}{\mathbf{y}}, \frac{\Delta \mathbf{w}}{\mathbf{w}}\right) \right]^{-1}, 1\right) \quad \text{式 7.6}$$

(6) 三个变量 ($\Delta \mathbf{x}$, $\Delta \mathbf{y}$, $\Delta \mathbf{w}$) 的下一个位置, 按式 7.7 计算

$$\begin{aligned} \mathbf{x}^{(n+1)} &= \lambda \Delta \mathbf{x} + \mathbf{x}^{(n)} \\ \mathbf{y}^{(n+1)} &= \lambda \Delta \mathbf{y} + \mathbf{y}^{(n)} \\ \mathbf{w}^{(n+1)} &= \lambda \Delta \mathbf{w} + \mathbf{w}^{(n)} \end{aligned} \quad \text{式 7.7}$$

(6) 迭代过程, 误差小于设定值即收敛到解。设置初始值, 2.0, 1.0, 2.0, 1.0, 不在可行域内; 本实例收敛到最优值, 0.4, 0.8, 1.0, 2.0, 最短距离为 1.341640786。

(7) 收敛性情况。按图 7.2 显示的精度, 上第 4 次迭代已经达到要求。误差在 1% 以下。

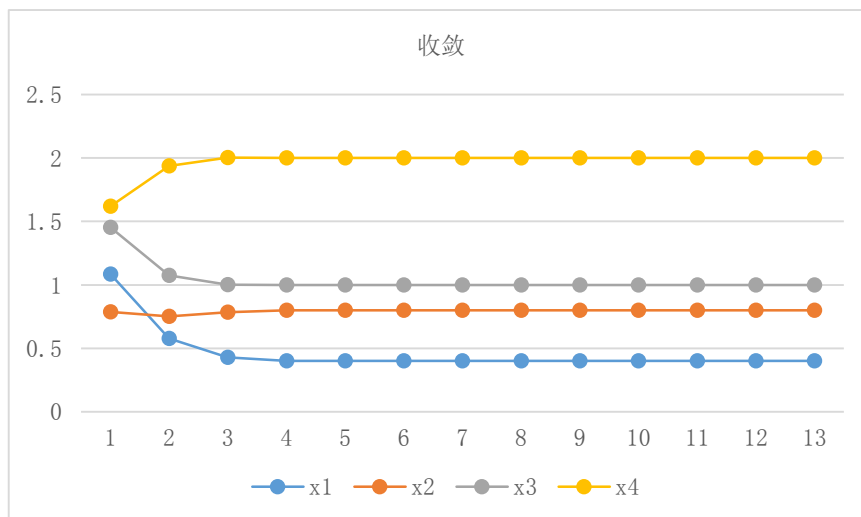


图 7.2

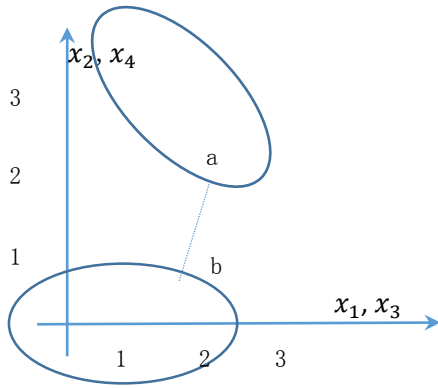
二、代码讲解

(1) 目前未完成 C 代码, 使用 Excel 完成计算, 请见[9]。

(2) 如果需要 C 代码, 可以参看[7], 算法使用 Lemke 方法(类似于单纯性法)。总体感觉 Lemke 方法类似线性方程组求解的高斯消元法(尚未认真理解过), 变量每稍微有变换, 需要重新开始计算, 而迭代法则持续迭代即可(上一个位置的解与当前位置不太远, 因此在有限步骤下能迭代求解到所需精度)。

第四节 平面椭圆间的最短距离求解

一、基本原理



(1) 两个椭圆位置图 7.3 所示，最短距离在点 a, b 处。

(2) 最短距离为 1.707800

图 7.3

即求两个椭圆之间的最短距离，可以使用数学描述表述，即受两个椭圆的约束。为模型简单起见，约束方式以等式考虑，不考虑不等式范畴，见式 7.8

$$\begin{aligned} \min f(\mathbf{x}) &= \mathbf{x}^T \mathbf{Q}_0 \mathbf{x} \\ \text{s. t. } \mathbf{x}_1^T \mathbf{Q}_1 \mathbf{x}_1 + \mathbf{p}_1 \mathbf{x}_1 + r_1 & \\ \mathbf{x}_2^T \mathbf{Q}_2 \mathbf{x}_2 + \mathbf{p}_2 \mathbf{x}_2 + r_2 & \end{aligned} \quad \text{式 7.8}$$

其中 \mathbf{x} 表示空间的 4 个位置， $\mathbf{x} = [x_1 \ x_2 \ x_3 \ x_4]^T$ ，其中 \mathbf{Q}_0 表示两个向量差的平方和的矩阵。矩阵 \mathbf{Q}_0 为：

$$\mathbf{Q}_0 = \begin{bmatrix} 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \\ -1 & 0 & 1 & 0 \\ 0 & -1 & 0 & 1 \end{bmatrix} \quad \text{式 7.9}$$

使用标量表示为 $f(\mathbf{x}) = (x_1 - x_3)^2 + (x_2 - x_4)^2$ 。

因约束条件是椭圆，因此 \mathbf{Q}_1 ， \mathbf{Q}_2 是对称正定矩阵，其中 \mathbf{x}_1 在椭圆 1 上， \mathbf{x}_2 在椭圆 2 上， $\mathbf{x}_1 = [x_1 \ x_2]^T$ ， $\mathbf{x}_2 = [x_3 \ x_4]^T$ ； $\mathbf{p}_1 = [p_1 \ p_2]^T$ ， $\mathbf{p}_2 = [p_3 \ p_4]^T$ 。为方便计算，使用 3 个标量描述约束条件，其中椭圆 \mathbf{Q}_1 ， \mathbf{Q}_2 矩阵中的各个元素，具体见式 7.10。

$$\begin{aligned} \mathbf{Q}_1 &= \begin{bmatrix} q_1 & q_2 \\ q_2 & q_3 \end{bmatrix} \\ \mathbf{Q}_2 &= \begin{bmatrix} q_4 & q_5 \\ q_5 & q_6 \end{bmatrix} \end{aligned} \quad \text{式 7.10}$$

可以使用标量表示两个约束，如式 7.11

$$\begin{aligned} q_1 x_1^2 + 2q_2 x_1 x_2 + q_3 x_2^2 + p_1 x_1 + p_2 x_2 + r_1 & \\ q_4 x_3^2 + 2q_5 x_3 x_4 + q_6 x_4^2 + p_3 x_3 + p_4 x_4 + r_2 & \end{aligned} \quad \text{式 7.11}$$

二、算法步骤

(1) 算法的具体步骤。构造 Lagrange 函数，求导，令其为 0，即最优值满足此方程；要求解此非线性方程，可以使用牛顿法，则必须构造多元函数的 Jacobi 矩阵。下面分别列示入下，使用 Lagrange 构造函数如式 7.12

$$\begin{aligned} L(\mathbf{x}, \lambda) &= \mathbf{x}^T \mathbf{Q}_0 \mathbf{x} + \lambda_1 (\mathbf{x}_1^T \mathbf{Q}_1 \mathbf{x}_1 + \mathbf{p}_1 \mathbf{x}_1 + r_1) \\ &+ \lambda_2 (\mathbf{x}_2^T \mathbf{Q}_2 \mathbf{x}_2 + \mathbf{p}_2 \mathbf{x}_2 + r_2) \end{aligned} \quad \text{式 7.12}$$

使用标量表示为式 7.13

$$\begin{aligned} L(x_1, x_2, x_3, x_4, \lambda_1, \lambda_2) &= (x_1 - x_3)^2 + (x_2 - x_4)^2 + \lambda_1 (q_1 x_1^2 + \\ &2q_2 x_1 x_2 + q_3 x_2^2 + p_1 x_1 + p_2 x_2 + r_1) + \lambda_2 (q_4 x_3^2 + 2q_5 x_3 x_4 + \\ &q_6 x_4^2 + p_3 x_3 + p_4 x_4 + r_2) \end{aligned} \quad \text{式 7.13}$$

(2) 对 Lagrange 求各阶偏导数，令其为 0，求解得到式 7.13。

$$\begin{aligned}
 \frac{\partial L}{\partial x_1} &= 2x_1 - 2x_3 + 2\lambda_1 q_1 x_1 + 2\lambda_1 q_2 x_2 + \lambda_1 p_1 \\
 \frac{\partial L}{\partial x_2} &= 2x_2 - 2x_4 + 2\lambda_1 q_3 x_2 + 2\lambda_1 q_2 x_1 + \lambda_1 p_2 \\
 \frac{\partial L}{\partial x_3} &= 2x_3 - 2x_1 + 2\lambda_2 q_4 x_3 + 2\lambda_2 q_5 x_4 + \lambda_2 p_3 \\
 \frac{\partial L}{\partial x_4} &= 2x_4 - 2x_2 + 2\lambda_2 q_5 x_3 + 2\lambda_2 q_6 x_4 + \lambda_2 p_4 \\
 \frac{\partial L}{\partial \lambda_1} &= q_1 x_1^2 + 2q_2 x_1 x_2 + q_3 x_2^2 + p_1 x_1 + p_2 x_2 + r_1 \\
 \frac{\partial L}{\partial \lambda_2} &= q_4 x_3^2 + 2q_5 x_3 x_4 + q_6 x_4^2 + p_3 x_3 + p_4 x_4 + r_2
 \end{aligned}
 \tag{式 7.14}$$

由于式 7.14 中有 6 个方程组中，最后两个是非线性方程组，考虑使用牛顿法求解。由于函数和求解区域为凸，因此保证在任意初值情况下函数能收敛到最优解。

(3) 使用牛顿迭代法求解。牛顿法基本思想，在解附近，可以使用线性展开表示，即式 7.15：

$$\mathbf{f}(\mathbf{x}^{(n+1)}) = \mathbf{f}(\mathbf{x}^{(n)}) + \frac{\partial \mathbf{f}}{\partial \mathbf{x}}(\mathbf{x}^{(n+1)} - \mathbf{x}^{(n)}) + o(\Delta \mathbf{x}) \tag{式 7.15}$$

忽略多维高阶无穷小 $o(\Delta \mathbf{x})$ ，令 $\mathbf{f}(\mathbf{x}^{(n+1)}) = \mathbf{0}$ ，其中 $\partial \mathbf{f} / \partial \mathbf{x}$ 就是 Jacobi 矩阵，表示成式 7.16：

$$\mathbf{f}(\mathbf{x}^{(n)}) + \frac{\partial \mathbf{f}}{\partial \mathbf{x}}(\mathbf{x}^{(n+1)} - \mathbf{x}^{(n)}) = \mathbf{0} \tag{式 7.16}$$

再求增量，即 $\Delta \mathbf{x} = \mathbf{x}^{(n+1)} - \mathbf{x}^{(n)}$ ，解线性方程组，可以得到 $\mathbf{x}^{(n+1)}$ ，如式子 7.17

$$\frac{\partial \mathbf{f}}{\partial \mathbf{x}} \Delta \mathbf{x} = -\mathbf{f}(\mathbf{x}^{(n)}) \tag{式 7.17}$$

(4) Jacobi 矩阵如下式子 7.18，矩阵中各个元素见附录 2 推导过程

$$\begin{bmatrix}
 \frac{\partial^2 L}{\partial x_1^2} & \frac{\partial^2 L}{\partial x_1 \partial x_2} & -2 & 0 & \frac{\partial^2 L}{\partial x_1 \partial \lambda_1} & 0 \\
 \frac{\partial^2 L}{\partial x_2 \partial x_1} & \frac{\partial^2 L}{\partial x_2^2} & 0 & -2 & \frac{\partial^2 L}{\partial x_2 \partial \lambda_1} & 0 \\
 -2 & 0 & \frac{\partial^2 L}{\partial x_3^2} & \frac{\partial^2 L}{\partial x_3 \partial x_4} & 0 & \frac{\partial^2 L}{\partial x_3 \partial \lambda_2} \\
 0 & -2 & \frac{\partial^2 L}{\partial x_4 \partial x_3} & \frac{\partial^2 L}{\partial x_4^2} & 0 & \frac{\partial^2 L}{\partial x_4 \partial \lambda_2} \\
 \frac{\partial^2 L}{\partial \lambda_1 \partial x_1} & \frac{\partial^2 L}{\partial \lambda_1 \partial x_2} & 0 & 0 & 0 & 0 \\
 0 & 0 & \frac{\partial^2 L}{\partial \lambda_2 \partial x_3} & \frac{\partial^2 L}{\partial \lambda_2 \partial x_4} & 0 & 0
 \end{bmatrix} \tag{式 7.18}$$

(5) 其中右端项，由 Lagrange 方程式 7.14 计算得到。

三、线性方程组的求解

(1) 方程组特性。由于线性方程组属于鞍点矩阵，不定，可以使用 MinRes 或 GMRES 求解。具体算法参见数值方法章节。由于矩阵条件数较好，无须做预处理。代入求解得到。

(2) 本实例没有编写代码，使用 Excel 求解。参见[6]，注意在 Excel 中，直接使用 minverse() 求矩阵逆。

四、数值实例。

(1) 问题的具体数值，初值设置为 2.00, 3.00, 5.00, 2.00, 2.00, 2.00。注意，初始值可以不满足约束条件，如式 7.19

$$\min f(\mathbf{x}) = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}^T \begin{bmatrix} 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \\ -1 & 0 & 1 & 0 \\ 0 & -1 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}$$

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}^T \begin{bmatrix} 0.25 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}^T \begin{bmatrix} -0.5 \\ 0 \end{bmatrix} - 0.75 = 0 \quad \text{式 7.19}$$

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}^T \begin{bmatrix} 0.625 & 0.375 \\ 0.375 & 0.625 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}^T \begin{bmatrix} -5.5 \\ -6.5 \end{bmatrix} + 17.5 = 0$$

(2) Jacobi 矩阵, 迭代第一次时的矩阵, 如式 7.20

$$\begin{bmatrix} 3 & 0 & -2 & 0 & 0.5 & 0 \\ 0 & 6 & 0 & -2 & 6 & 0 \\ -2 & 0 & 4.5 & 1.5 & 0 & 2.25 \\ 0 & -2 & 1.5 & 4.5 & 0 & -0.25 \\ 0.5 & 6 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2.25 & 0.25 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ \lambda_1 \\ \lambda_2 \end{bmatrix} = \begin{bmatrix} 5 \\ -14 \\ -10.5 \\ 2.5 \\ -8.25 \\ -2.625 \end{bmatrix} \quad \text{式 7.20}$$

(3) 经过求解, 在第 4 次得到 2.044750, 0.852716, 2.544913, 2.485633, 坐标值满足约束, 距离的最小值为 1.707800.

(4) 收敛的数值情况, 将图 7.4

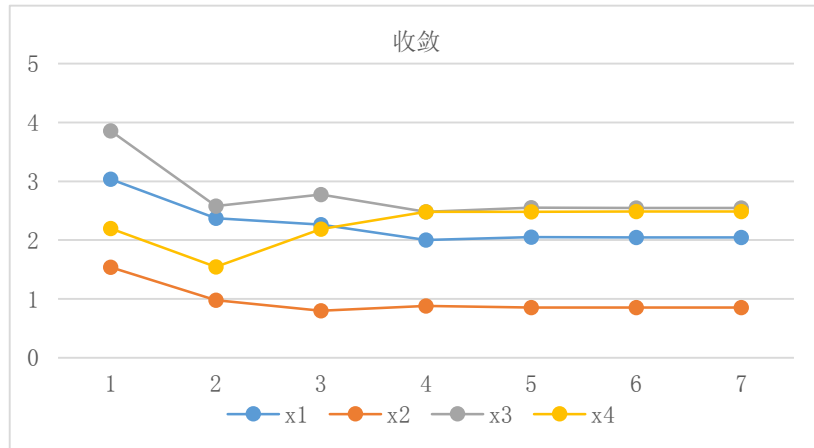


图 7.4

第五节 总结

本章节通过最优化方法来求解物体之间的碰撞检测, 并实现了两个例子, 第一个例子是不等式约束下的二次规划问题(QP), 第二个是二次约束下的二次规划问题(QCQP)。

通过最优化求解凸体之间最短距离的方法还有很多, 特别是在不等式约束的情况下, 还有起作用集, Lemke 方法等, 特别是内点法(IPM)中的原对偶路径跟踪算法并未涉及(后续考虑)。目前尚未查询到各种方法比较的文章, 因此未做深入的论述。

备注说明 1. 当前流行的交替方向乘子法(ADMM)的简单例子可以参看[10], Excel 的求解二次规划的简单实现, 在图形学中的应用可参看[11]。

备注说明 2. 两个实例具体算法使用 Excel 完成(桌面机器都装 Excel), 看了计算过程都懂; 并未使用 Matlab(庞大, 部分算法封装后无法应用在实际项目)。

备注说明 3. 实时显示的代码将在后期完成。

备注说明 4. 修改多次, 依然有理解错误, 叙述不清, 别字等, 请反馈。

附录 1: KKT 方程求解的条件(避坑 1)

(1) 例子一，求距离两个三角形最近的点。将方程表示如下，其中 H 矩阵和不等式约束为已知问题；具体表示

$$\begin{aligned} \min f(\mathbf{x}) &= \mathbf{x}^T \mathbf{Q}_0 \mathbf{x} \\ \text{s. t. } \mathbf{A} \mathbf{x} + \mathbf{b} &\geq \mathbf{0} \end{aligned} \quad \text{附 1.1}$$

使用等式 Lagrange 方法求解，有

$$L(\mathbf{x}, \lambda) = \mathbf{x}^T \mathbf{Q}_0 \mathbf{x} + \lambda^T (\mathbf{A} \mathbf{x} + \mathbf{b}) \quad \text{附 1.2}$$

最优点满足

$$\begin{aligned} \frac{\partial L}{\partial \mathbf{x}} &= 2\mathbf{Q}_0 \mathbf{x} + \mathbf{A}^T \\ \frac{\partial L}{\partial \lambda} &= \mathbf{A} \mathbf{x} + \mathbf{b} \end{aligned} \quad \text{附 1.3}$$

方程的求解。

$$\begin{bmatrix} 2\mathbf{Q}_0 & \mathbf{A}^T \\ \mathbf{A} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \lambda \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{b} \end{bmatrix} \quad \text{附 1.4}$$

求解中，发现 KKT 方程的秩不满足求解条件，即矩阵 \mathbf{A} 的行秩小于 4。结合图形看，就是此最优点不能满足所有等式，只能满足不等式，因此使用等式 Lagrange 方法求解有误。

(2) 结论解释。如果矩阵 \mathbf{A} 的行数大于列数，说明约束条件太多，更本就没有解，数值上表现为 KKT 矩阵奇异；当然约束组合到 Lagrange 方程里面也同样没有解。这也就是 KKT 方程有解的条件。只有当矩阵 \mathbf{A} 的行数小于列数是，约束条件的解才有无数多个，才能从其中选择最优值。

注意：使用等式拉格朗日乘子计算最优值时，KKT 矩阵中分块矩阵为 0，即矩阵奇异；而使用不等式求解是，求解的线性方程组的右下角矩阵非 0，这也就保证了在矩阵的秩不满足求解情况下，整个矩阵非奇异。

此外，KKT 方程有唯一解的条件是目标函数和约束条件为凸。

(2) 求解方法。求解带不等式约束条件的方法很多，此实例使用原路径跟踪方法求解。具体简单推导如下

$$\begin{aligned} \min f(\mathbf{x}) &= \mathbf{x}^T \mathbf{H} \mathbf{x} \\ \text{s. t. } \mathbf{A} \mathbf{x} &\geq \mathbf{b} \end{aligned} \quad \text{附 1.5}$$

满足 Lagrange 乘子，使得 KKT 条件成立，有附 1.6

$$\begin{aligned} \mathbf{H} \mathbf{x} - \mathbf{A}^T \mathbf{y} + \mathbf{c} &= \mathbf{0} \\ \mathbf{A} \mathbf{x} - \mathbf{w} - \mathbf{b} &= \mathbf{0} \\ \mathbf{Y} \mathbf{W} \mathbf{e} &= \mu \mathbf{e} \\ \mathbf{y}, \mathbf{w} &\geq \mathbf{0} \end{aligned} \quad \text{附 1.6}$$

有增量满足，得到附 1.6 式

$$\begin{aligned} \mathbf{H}(\mathbf{x} + \Delta \mathbf{x}) - \mathbf{A}^T(\mathbf{y} + \Delta \mathbf{y}) + \mathbf{c} &= \mathbf{0} \\ \mathbf{A}(\mathbf{x} + \Delta \mathbf{x}) - (\mathbf{w} + \Delta \mathbf{w}) - \mathbf{b} &= \mathbf{0} \\ (\mathbf{Y} + \Delta \mathbf{Y})(\mathbf{W} + \Delta \mathbf{W}) \mathbf{e} &= \mu \mathbf{e} \end{aligned} \quad \text{附 1.6}$$

其中， $\mathbf{Y} = \text{diag}(\mathbf{y})$ ， $\mathbf{W} = \text{diag}(\mathbf{w})$ 展开有附 1.6

$$\begin{aligned} -\mathbf{H} \Delta \mathbf{x} + \mathbf{A}^T \Delta \mathbf{y} &= \mathbf{c} + \mathbf{H} \mathbf{x} - \mathbf{A}^T \mathbf{y} \\ \mathbf{A} \Delta \mathbf{x} - \Delta \mathbf{w} &= \mathbf{b} - \mathbf{A} \mathbf{x} + \mathbf{w} \\ \mathbf{W} \Delta \mathbf{y} + \mathbf{Y} \Delta \mathbf{w} &= \mu \mathbf{e} - \mathbf{Y} \mathbf{W} \mathbf{e} \end{aligned} \quad \text{附 1.6}$$

将附 1.6 的第三式，代入第二式，有式 7.4

$$\begin{bmatrix} -\mathbf{H} & \mathbf{A}^T \\ \mathbf{A} & \mathbf{Y}^{-1}\mathbf{W} \end{bmatrix} \begin{bmatrix} \Delta \mathbf{x} \\ \Delta \mathbf{y} \end{bmatrix} = \begin{bmatrix} \mathbf{c} + \mathbf{H}\mathbf{x} - \mathbf{A}^T\mathbf{y} \\ \mathbf{b} - \mathbf{A}\mathbf{x} + \mu\mathbf{Y}^{-1}\mathbf{e} \end{bmatrix} \quad \text{式 7.4}$$

附录 2：例子 2 的 Jacobi 矩阵

牛顿法中需要 Jacobi 矩阵，下面是推导表达式，是个痛苦活，容易出错。因为矩阵能分小块，但计算量也较大，为验证计算的正确性，按矩阵的每个元素进行计算。注意，由于次 Jacobi 矩阵对称，因此可以在推导时检查推导的正确性。

[1]矩阵第一行第一个元素

$$\frac{\partial^2 L}{\partial x_1^2} = 2 + 2\lambda_1 q_1 \quad 2.1$$

[2]矩阵第一行第二个元素

$$\frac{\partial^2 L}{\partial x_1 \partial x_2} = 2\lambda_1 q_2 \quad 2.2$$

[3]矩阵第一行第三个元素

$$\frac{\partial^2 L}{\partial x_1 \partial x_3} = -2 \quad 2.2$$

[4]矩阵第一行第五个元素

$$\frac{\partial^2 L}{\partial x_1 \partial \lambda_1} = 2q_1 x_1 + 2q_2 x_2 + p_1 \quad 2.4$$

[5]矩阵第一行其他元素

$$\frac{\partial^2 L}{\partial x_1 \partial x_4} = \frac{\partial^2 L}{\partial x_1 \partial \lambda_2} = 0 \quad 2.5$$

[6]矩阵第二行第二个元素

$$\frac{\partial^2 L}{\partial x_2^2} = 2 + 2\lambda_1 q_3 \quad 2.6$$

[7]矩阵第二行第一个元素

$$\frac{\partial^2 L}{\partial x_2 \partial x_1} = 2\lambda_1 q_2 \quad 2.7$$

[8]矩阵第二行第四个元素

$$\frac{\partial^2 L}{\partial x_2 \partial x_4} = -2 \quad 2.8$$

[9]矩阵第二行第五个元素

$$\frac{\partial^2 L}{\partial x_2 \partial \lambda_1} = 2q_3 x_2 + 2q_2 x_1 + p_2 \quad 2.9$$

[10]矩阵第二行其他元素

$$\frac{\partial^2 L}{\partial x_2 \partial x_3} = \frac{\partial^2 L}{\partial x_2 \partial \lambda_2} = 0 \quad 2.10$$

[11]矩阵第三行第一个元素

$$\frac{\partial^2 L}{\partial x_3 \partial x_1} = -2 \quad 2.11$$

[12]矩阵第三行第三个元素

$$\frac{\partial^2 L}{\partial x_3^2} = 2 + 2\lambda_2 q_4 \quad 2.12$$

[13]矩阵第三行第四个元素

$$\frac{\partial^2 L}{\partial x_3 \partial x_4} = 2\lambda_2 q_5 \quad 2.13$$

[14]矩阵第三行第六个元素

$$\frac{\partial^2 L}{\partial x_3 \partial \lambda_2} = 2q_4 x_3 + 2q_5 x_4 + p_3 \quad 2.14$$

[15]矩阵第三行其他元素

$$\frac{\partial^2 L}{\partial x_3 \partial x_2} = \frac{\partial^2 L}{\partial x_3 \partial \lambda_1} = 0 \quad 2.15$$

[16]矩阵第四行第四个元素

$$\frac{\partial^2 L}{\partial x_4^2} = 2 + 2\lambda_2 q_6 \quad 2.16$$

[17]矩阵第四行第二个元素

$$\frac{\partial^2 L}{\partial x_4 \partial x_2} = -2 \quad 2.17$$

[18]矩阵第四行第三个元素

$$\frac{\partial^2 L}{\partial x_4 \partial x_3} = 2\lambda_2 q_5 \quad 2.18$$

[19]矩阵第四行第六个元素

$$\frac{\partial^2 L}{\partial x_4 \partial \lambda_2} = 2q_5 x_3 + 2q_6 x_4 + p_4 \quad 2.19$$

[20]矩阵第四行其他元素

$$\frac{\partial^2 L}{\partial x_4 \partial \lambda_1} = \frac{\partial^2 L}{\partial x_4 \partial x_1} = 0 \quad 2.20$$

[21]矩阵第五行第一个元素

$$\frac{\partial^2 L}{\partial \lambda_1 \partial x_1} = 2q_1 x_1 + 2q_2 x_2 + p_1 \quad 2.21$$

[22]矩阵第五行第二个元素

$$\frac{\partial^2 L}{\partial \lambda_1 \partial x_2} = 2q_3 x_2 + 2q_2 x_1 + p_2 \quad 2.22$$

[23]矩阵第五行其他元素

$$\frac{\partial^2 L}{\partial \lambda_1 \partial x_3} = \frac{\partial^2 L}{\partial \lambda_1 \partial x_4} = \frac{\partial^2 L}{\partial \lambda_1 \partial \lambda_2} = \frac{\partial^2 L}{\partial \lambda_1^2} = 0 \quad 2.23$$

[24]矩阵第六行第三个元素

$$\frac{\partial^2 L}{\partial \lambda_2 \partial x_3} = 2q_4 x_3 + 2q_5 x_4 + p_3 \quad 2.24$$

[25]矩阵第六行第四个元素

$$\frac{\partial^2 L}{\partial \lambda_2 \partial x_4} = 2q_6 x_4 + 2q_5 x_3 + p_4 \quad 2.25$$

[26]矩阵第六行其他元素

$$\frac{\partial^2 L}{\partial \lambda_2 \partial x_1} = \frac{\partial^2 L}{\partial \lambda_2 \partial x_2} = \frac{\partial^2 L}{\partial \lambda_2 \partial \lambda_1} = \frac{\partial^2 L}{\partial \lambda_2^2} = 0 \quad 2.26$$

附录 3：例子 2 中的错误表达方式(避坑 2)

(1) 为计算和后续推导方便，可以将两个约束合并成一个表达式，具体如下，但这种方式有误。

$$\begin{aligned} \min f(\mathbf{x}) &= \mathbf{x}^T \mathbf{Q}_0 \mathbf{x} \\ \text{s. t. } &\mathbf{x}^T \mathbf{Q}_3 \mathbf{x} + \mathbf{p}^T \mathbf{x} + r \end{aligned} \quad \text{附 3.1}$$

其中 \mathbf{Q}_3 由 \mathbf{Q}_1 , \mathbf{Q}_2 对角方式组合而成

$$\begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix}^T \begin{bmatrix} \mathbf{Q}_1 & 0 \\ 0 & \mathbf{Q}_2 \end{bmatrix} \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix} + \begin{bmatrix} \mathbf{p}_1 \\ \mathbf{p}_2 \end{bmatrix}^T \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix} + r_1 + r_2 = 0 \quad 3.2$$

如果按此方式计算，计算结果同时满足式 3.1 的约束条件，但不能分别满足两个椭圆，数值结果与原问题比较，列表 1.1 如下，可以看出，求解得到的结果，更小的目标值，但不能分别满足约束，但满足约束的和。

	x1	x2	x3	x4	目标函数	约束条件 1	约束条件 2	两个约束条件的和
错误方法的解	2.0215	0.7894	2.5456	2.4096	1.7029	-0.1160	0.1160	0.0000
正确方法的解	2.0447	0.8527	2.5449	2.4856	1.7078	0.0000	0.0000	0.0000

表 1.1

(2) 对于仅有一个约束的 QCQP 问题，如附 3.1 可以附 3.3 方式的 Lagrange 函数，其基本思路是将 \mathbf{Q}_3 阵(对称正定)进行特征分解，具体参见文献[5]。

$$L(\hat{\mathbf{x}}, \lambda) = \hat{\mathbf{x}}^T (\mathbf{I} + \lambda \mathbf{A}) \hat{\mathbf{x}} + \lambda \hat{\mathbf{p}}^T \hat{\mathbf{x}} + \lambda r \quad \text{附 3.3}$$

其中 $\hat{\mathbf{x}}$ 和 $\hat{\mathbf{p}}^T$ 都经过特征变换，如附 3.3 为一个仅有对角矩阵的方程，可以通过求解 λ ，得到最优解，且作者证明了函数在特殊情况下仅求解一个一元高次方程即可。

附录 4：椭圆间最短距离求解的其他方法

由于椭圆间的最短距离求解也有其他应用(数据集的分类等)，是一个基本问题，有多种求解方式，经不完全查询，有以下方法。

(1) 方法 1。可以将一个椭圆通过坐标变换到原点(坐标平移和旋转，属于线性变换)，再原点处的椭圆缩放成圆(属于线性变换)，这时原问题变成一个圆到椭圆的距离，即原点到另一个椭圆的最短距离。基本上使用初等数学可以求出。未测试过是否正确(可能在变换坐标后，变换坐标后的最优解不是原坐标中的最优解)。

(3) 方法 2。未归类，链接如下，其中第二个是《Game Physics Second》的作者

<https://mpewsey.github.io/2021/11/12/minimum-distance-between-ellipses.html>

<https://www.geometrictools.com/Documentation/DistanceEllipse2Ellipse2.pdf>

附录 5：KKT 矩阵的不定性说明

KKT 方程如附件 5.1

$$\begin{bmatrix} \mathbf{A} & \mathbf{B}^T \\ \mathbf{B} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \lambda \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{b} \end{bmatrix} \quad \text{附 5.1}$$

可以将 KKT 矩阵做合同变换如附 5.2

$$\begin{bmatrix} \mathbf{A} & 0 \\ \mathbf{B}^T & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{A}^{-1} & 0 \\ 0 & -\mathbf{B}^T \mathbf{A}^{-1} \mathbf{B} \end{bmatrix} \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{I} & 0 \end{bmatrix} \quad \text{附 5.2}$$

附 5.2 中间的矩阵，其中 \mathbf{A} 正定， \mathbf{A}^{-1} 也正定， $-\mathbf{B}^T \mathbf{A}^{-1} \mathbf{B}$ 负定，则整个矩阵 KKT 矩阵不定。

参考文献

- [1] Collision detection: A survey
- [2] 《Real-Time Collision Detection》，有中译本
- [3] <https://github.com/ljb1672/DirectX-3D-HLSL-Senior-instance-succinctly> 中的文件。
- [4] GPU GEM3:LCP Algorithms for Collision Detection Using CUDA
- [5] General Heuristics for Nonconvex Quadratically Constrained Quadratic Programming
- [6] QCQP.xlsx
- [7] Game Physics Second
- [8] 《DirectX 3D HLSL 高级实例精讲》
- [9] QP.xlsx
- [10] ADMM.xlsx
- [11] Interactive Hair Simulation on the GPU using ADMM
- [12] Methods for quadratic programming: A survey
- [13] Game Physics Second