

# ***Eulerian-on-Lagrangian Cloth Simulation***

The Final Report of  
Research Advances in Computer Graphics

文嘉豪 3160104626

程浩然 3160105088

杜林峰 3160105466

*Department:* 计算机科学与技术学院

*Teacher:* 周昆/任重

*Date:* 2019-06-24

## *Contents*

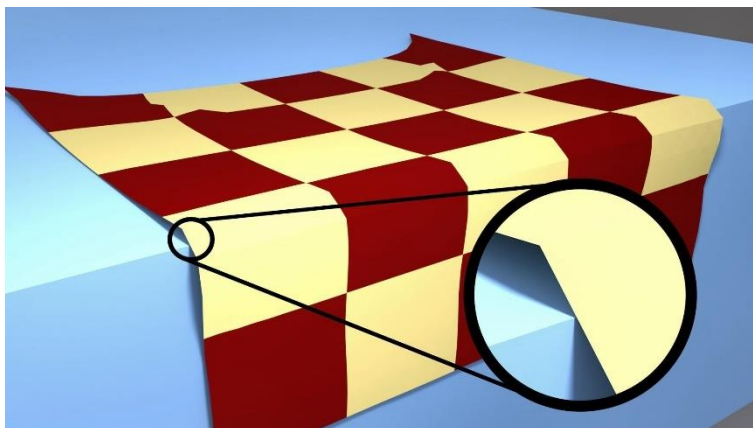
<b>1 Introduction .....</b>	<b>2</b>
1.1 Background .....	2
1.2 Pipeline .....	3
<b>2 Finite Element Model .....</b>	<b>4</b>
2.1 Barycentric Coordinates .....	4
2.2 Motion Equation .....	4
2.3 Inertia Matrix .....	5
2.4 Bending Force .....	5
2.5 Membrane Force.....	6
2.6 External Force.....	6
<b>3 Dynamic Evolution .....</b>	<b>7</b>
3.1 Collision Detection.....	7
3.2 Mesh Process .....	8
3.2.1 Mesh Clean .....	8
3.2.2 Dynamic Remesh.....	9
3.2.3 Velocity Transfer.....	9
3.3 Constraint .....	10
<b>4 Result .....</b>	<b>11</b>
<b>5 Summary.....</b>	<b>12</b>

# 1 Introduction

## 1.1 Background

基于物理的模拟、动画(physically based simulation / animation)一直是图形学中非常重要的方向，而布料模拟则是物理模拟中一个经久不衰的热门课题。传统的布料模拟大概可以分为两种模型，即弹簧质点模型(Mass-Spring simulation)和有限元模型(Finite Element Method)。弹簧质点模型将布料拆解为一系列离散的质点，质点和质点之间用弹簧相连接，通过改变弹簧的本构方程，可以模拟不同特征的布料。而有限元模型，则将布料模型分割为面网格，一般采用三角形单元，给定每个单元的属性、基函数、边界条件，将所有单元组装为整个离散域上的总体矩阵方程，迭代求解，即可得到模型上各节点的状态分布。

然而，无论是弹簧质点还是有限元，当布料与尖锐特征(sharp feature)发生碰撞和接触时，都会产生不准确的结果。因为以上两种离散模型都只能允许布料在节点处或边上产生弯曲，当没有节点和边恰好处在尖锐特征上或与尖锐特征对齐时，就会产生锯齿形的失真运动，如下图所示：



当布料沿着立方体边缘滑动时，由于有限元网格的边并不总能与立方体的边对齐，于是在滑动时会有网格边被“抬起”的锯齿感。

SIGGRAPH 2018 年的论文《Eulerian-on-Lagrangian Cloth Simulation》提出了一种解决上述问题的方法。文章在原来三维的拉格朗日坐标上新引入了节点的欧拉坐标（可以理解为纹理坐标），使得布料在与尖锐特征发生碰撞时仍然能够在节点处保证有足够的自由度，从而平滑地运动。欧拉坐标和拉格朗日坐标原先是流体模拟中两种不同的坐标空间，在流体模拟中，欧拉坐标固定在空间内，不随质点的运动而改变，只用来观测其物理属性的变化，而拉格朗日坐标则关联在流体质点上，随流体粒子的运动而运动，描述其运动的轨迹。

## 1.2 Pipeline

整个模拟的流程如下：

---

**Algorithm 1** EOL Cloth Time Stepper

---

```
1: while simulating do  
2:   Detect collisions  
3:   Preprocess & Remesh  
4:   Compute EOL constraints on new mesh  
5:   Integrate velocities and positions  
6: end while
```

---

首先进行碰撞检测，检测并记录布料与障碍物上尖锐特征的碰撞接触，然后调用第三方 remesh 库 Arcsim，在这些碰撞点上引入新的节点，并标记为欧拉点，之后添加边界条件，构造并求解总体矩阵，更新各个节点的速度和位置。

EoL 方法的核心在于，每一个时间步长内都会在碰撞点处引入新的欧拉-拉格朗日点，这些点的存在保证了网格可以在尖锐特征上有充分的自由度弯曲而不会造成失真，同时将滑动的速度“储存”在欧拉坐标内。在下一个时间步长内，旧的欧拉-拉格朗日点会重新变为普通的拉格朗日点，欧拉速度会被“释放”转换为三维世界中的拉格朗日速度，使得这些点滑动远离障碍物的尖锐特征，而在原来 sharp feature 的位置又会有新的欧拉-拉格朗日点产生。整个系统就像是一个定滑轮，将原本由于各种边界条件限制而无法运动的拉格朗日速度变换方向，使其可以流畅连续的滑动。

## 2 Finite Element Model

论文中所采用的基础物理模型基于有限元方法，即用一张三角面网格来对布料进行建模。所考虑的力有 elastic force, external force 两种，其中 elastic force 描述了布料本身的弹性特征，分为 bending force 和 membrane force，分别刻画了布料的弯曲和拉伸行为，而 external force 则表示布料所受到的外力，比如重力、拉力等。通过基础的物理模型，就可以模拟出布料一些简单的形变行为。

### 2.1 Barycentric Coordinates

本小节首先讲述欧拉坐标空间与拉格朗日坐标空间下的转换关系。设一个三角形片元的三个顶点为  $(a, b, c)$ ,  $\mathbf{X} \in R^2$  为该三角形中任意一个质点，通过其欧拉坐标，可以计算它的 barycentric 坐标为  $(\alpha, \beta, \gamma)$ 。则拉格朗日空间下，它的坐标可以写作  $\mathbf{x}(\mathbf{X}) = \alpha \mathbf{x}_a + \beta \mathbf{x}_b + \gamma \mathbf{x}_c$ ,  $(x_a, x_b, x_c)$  表示三角形三个顶点的拉格朗日坐标。那么可以证明，该点最终在三维世界中的速度为：

$$\dot{\mathbf{x}} = (\alpha \dot{\mathbf{x}}_a + \beta \dot{\mathbf{x}}_b + \gamma \dot{\mathbf{x}}_c) - F(\alpha \dot{\mathbf{X}}_a + \beta \dot{\mathbf{X}}_b + \gamma \dot{\mathbf{X}}_c)$$

$$F^{3 \times 2} = D_x D_X^{-1}$$

$$D_x^{3 \times 2} = (\mathbf{x}_b - \mathbf{x}_a, \mathbf{x}_c - \mathbf{x}_a)$$

$$D_X^{2 \times 2} = (\mathbf{X}_b - \mathbf{X}_a, \mathbf{X}_c - \mathbf{X}_a)$$

其中  $F$  表示了该三角单元的形变梯度。通过  $F$ ，就可以将欧拉坐标下的速度、形变、力转换为拉格朗日坐标。写成更广义的形式如下：

$$J = \alpha I \quad \beta I \quad \gamma I \quad -\alpha F \quad -\beta F \quad -\gamma F$$

$$\dot{\mathbf{x}} = J \dot{\mathbf{q}}, \text{ where } \dot{\mathbf{q}} = (\dot{x}_a, \dot{x}_b, \dot{x}_c, \dot{X}_a, \dot{X}_b, \dot{X}_c)$$

### 2.2 Motion Equation

论文中采用隐式欧拉法对整个系统的状态做更新，其动力学方程为：

$$(M - h^2 K) \dot{\mathbf{q}}^{(k+1)} = M \dot{\mathbf{q}}^{(k)} + h \mathbf{f}$$

其中  $M$  为质量矩阵， $h$  为时间步长， $K$  为劲度矩阵， $\dot{\mathbf{q}}$  是广义坐标下的速度。

## 2.3 Inertia Matrix

本小节中介绍如何计算系统的质量矩阵。对于一个三角形单元来说，其动能可以写作：

$$T = \frac{1}{2} \int_A \rho \dot{x}^T \dot{x} dA$$

其中 $\rho$ 为布料密度，积分域是整个三角形的面积，将上文所描述的 barycentric 坐标作为积分变量，可以得到以下形式：

$$T = \frac{1}{2} \int_{\alpha=0}^1 \int_{\beta=0}^{1-\alpha} \rho \dot{x}^T \dot{x} A d\beta d\alpha$$

其中 $A$ 是欧拉坐标下三角形的面积。将 $\alpha$ 和 $\beta$ 积出来，可以写成矩阵的形式：

$$T = \frac{1}{2} \dot{q}^T M \dot{q}$$

其中  $M$  即为所求的质量矩阵，由上面的公式 $\dot{x} = J\dot{q}$ ，可以推出 $M$ 为：

$$M = \frac{\rho A}{12} \begin{pmatrix} 2I & I & I & -2F & -F & -F \\ \cdot & 2I & I & -F & -2F & -F \\ \cdot & \cdot & 2I & -F & -F & -2F \\ \cdot & \cdot & \cdot & 2F^T F & F^T F & F^T F \\ \cdot & \cdot & \cdot & \cdot & 2F^T F & F^T F \\ \cdot & \cdot & \cdot & \cdot & \cdot & 2F^T F \end{pmatrix}$$

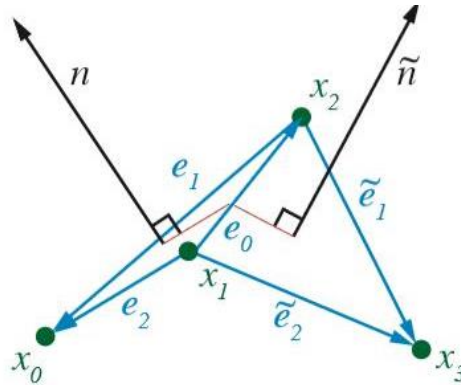
## 2.4 Bending Force

关于弯曲力的计算，论文中并没有直接给出显式的计算公式，而是给出了弯曲所产生的弹性势能的能量公式，通过计算能量对空间坐标的偏导，就可以得到布料在各个方向上的弯曲应力。

计算能量的 hessian 矩阵，即为动力学方程中的刚度矩阵。相邻两个三角单元的弯曲势能如下：

$$V_b = \frac{K(3\|e_i\|^2)}{A_i} \sin^2\left(\frac{\theta_i}{2}\right)$$

示意图如下：



其中 $e_i$ 为公共边（欧拉空间下）， $A_i$ 为两相邻三角形的面积和（欧拉空间下）， $\theta_i$ 为两相邻三角形法向量的夹角， $K$ 为劲度系数。可以发现 $V_b$ 与拉格朗日坐标和欧拉坐标均相关。

## 2.5 Membrane Force

关于拉伸力，同上，论文中也没有显式地给出力的计算公式，而是给出了能量的定义。对于一个三角单元，论文给出 membrane 能量形式为：

$$V_m = A(\mu||\bar{F} - R||_F^2 + \frac{\lambda}{2}tr^2(R^T\bar{F} - I))$$

其中  $\bar{F} \in R^{2 \times 2}$ ，为三角单元形变梯度的投影， $R$  是对  $\bar{F}$  做极分解得到的旋转矩阵， $||\cdot||_F$  表示 Frobenius 范数， $tr(\cdot)$  表示矩阵的迹。下面介绍如何计算  $\bar{F}$ ：

$$\mathbf{n} = (\mathbf{x}_b - \mathbf{x}_a) \times (\mathbf{x}_c - \mathbf{x}_a)$$

$$\mathbf{p}_x = \frac{\mathbf{x}_b - \mathbf{x}_a}{||\mathbf{x}_b - \mathbf{x}_a||}, \quad \mathbf{p}_y = \frac{(\mathbf{n} \times \mathbf{p}_x)}{||\mathbf{n} \times \mathbf{p}_x||}$$

$$P = (\mathbf{p}_x, \mathbf{p}_y)^T \in R^{2 \times 3}$$

$$\bar{F} = PF$$

同理，根据  $V_m$  的能量定义，分别对三角形三个顶点的拉格朗日坐标和欧拉坐标求偏导，就可以得到其对应的 membrane force，求二阶偏导计算 hessian 矩阵，即可得到 membrane force 的刚度矩阵。Membrane force 反应了布料的拉伸性质。

## 2.6 External Force

布料所受的外力包括重力、拉力、风力等，对于布料上的每个节点，首先能够计算出其在世界坐标下的受力  $f$ ，之后使用上文中提到的 Jacobian 矩阵，将外力分解为拉格朗日坐标分量和欧拉坐标分量。

$$\begin{pmatrix} f^L \\ f^E \end{pmatrix} = \begin{pmatrix} f \\ -F^T f \end{pmatrix}$$

其相关的劲度矩阵为

$$\begin{pmatrix} K^{LL} & K^{LE} \\ K^{EL} & K^{EE} \end{pmatrix} = \begin{pmatrix} K & -KF \\ -F^T K & F^T K F \end{pmatrix}$$

在实际的模拟过程中，外力常常是预先给定一个常值或者有显式的解析形式，所以计算相对比较简单。

模拟过程中，在每个单元上分别计算其对应力的值和刚度矩阵、质量矩阵，并填充到总体动力学方程的系数矩阵中，使用 mosek 求解总体的动力学方程，就可以得到各个节点的拉格朗日速度和欧拉速度，依次更新，即可得到下一个状态。

## 3 Dynamic Evolution

第2章中介绍了论文中布料模拟的基本模型,该模型已经可以描述诸如拉伸、弯曲等布料基本的弹性特征。在本章中介绍布料如何与障碍物做碰撞检测,以及如何动态地根据碰撞检测添加和删除欧拉-拉格朗日点、如何动态地重构网格、如何对不同类型的节点做速度更新。

### 3.1 Collision Detection

在预处理和 Remesh 之前,我们需要进行碰撞检测环节来构造一个碰撞列表,为后面进行约束提供基础。

此处我们只关注障碍物与布料的二元碰撞,将其进行分类如下:

**第一种:** 布料顶点与障碍物的面

此类碰撞只发生运动约束,不需要与传递数据给 remesher。

**第二种:** 布料的边与障碍物的边

这种碰撞情形是最常见的,这类碰撞的列表传入 remesher 为折线形成的基础。

**第三种:** 布料的面与障碍物的点

这种碰撞因障碍物的性质而有所不同:障碍物本身可能是一个箱型物体上的角点,也可能是单独的一个点,这些情况在我们的测试中都会有所体现。

**下面讲解碰撞检测的具体实现:**

1. 最先进行的是检测障碍物的点与布的点、面之间的碰撞。对整个布料计算 AABB 包围盒,当经过扰动的障碍物上的一点(如果是方盒,则只有八个顶点可能符合此情形)落在布料的 AABB 包围盒内时,通过布料和障碍物的法向量方向判断是否该点与布发生碰撞,当投影向量的差在一定阈值内的时候,则判断碰撞发生。由于障碍物上的同一点可能与布料上多个面满足上述条件,通过循环取得碰撞距离最近的记录在碰撞列表中。

2. 检测障碍物的面与布料上的点的碰撞。如果布料上的某一点经过扰动可以落在障碍物的 AABB 包围盒内,则根据障碍物面法向量和投影向量判断点与面的距离是否在阈值之内,去最近的一组情形加入碰撞列表。

3. 边与边的碰撞较为复杂,每一组边对(一条来自障碍物、另一条在布上)都要计算。对于每条障碍物上的边,取其相邻的两个三角形构成的 AABB 盒,分别与布料上的某条边的 AABB 盒进行包含判断,若后者不包含前者中的任一个,则没有发生碰撞。否则继续进行两边是否平行的检测与法向量方向的判断。特别的,在这种情况下还要排除由于顶点碰撞而满足条件的情况。



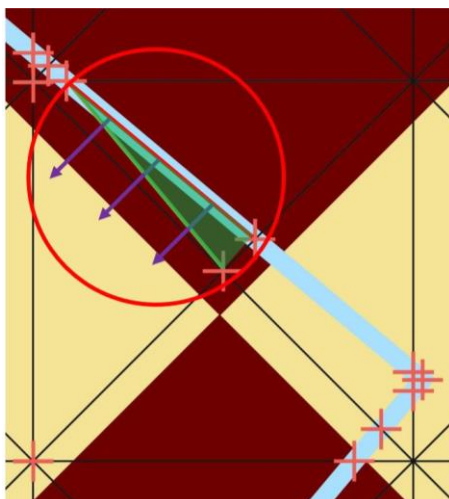
碰撞时，布料上的边与两个三角形的相对关系可能有四种，可能穿过两个三角形，可能穿过其一，也可能不与两个三角形的 AABB 盒相交。当两边碰撞情形符合且距离在阈值内，则将其加入碰撞列表。

### 3.2 Mesh Process

文章中对网格的处理主要分为：清理病态的三角形片元、依据碰撞检测添加新的欧拉-拉格朗日点、对不同类型的节点做速度更新这三个部分。实现代码中有些操作可能难以理解，其目的是为了适配第三方 remesh 库 Arcsim。

#### 3.2.1 Mesh Clean

在布料相对障碍物运动时，由于障碍物的 sharp feature 上始终有欧拉-拉格朗日点和拉格朗日点之间所构成的边，相当于发生碰撞时，在纹理空间上始终有运动的点和边，这些点和边的运动会造成一些三角形过于“狭长”，点和点之间距离过近即会产生一些病态的网格结构。如图所示：



为了 Arcsim 能够正常工作，以及运动方程能够准确求解，需要在每一次 remesh 时对网格结构进行清理。论文中给出了在不同情况下网格清理的策略。

1. 对于没有产生碰撞的边，如果其长度小于给定阈值，则直接合并删除。
2. 对于产生碰撞的边，如果其长度小于给定阈值，则直接合并删除，但删除的点不应该处在布料的边界和角点上。
3. 对于所有的病态三角形，分割其中的一条边并向其中插入新的节点，并删除该三角形。
  - (a) 如果该三角形有一个欧拉-拉格朗日点，则分割其对边。（例如：sharp feature 为单个点，且向着对边运动）
  - (b) 如果该三角形有两个欧拉-拉格朗日点，则分割发生碰撞的边。（例如：sharp feature

为一条边，且向着对点运动)

(c) 如果该三角形有三个欧拉-拉格朗日点，分割未发生碰撞的边。(例如：sharp feature 为立方体的某一个棱角，则分割两条棱之外的那条边)

值得注意的是，以上的网格清理策略并没有给出一个明确的收敛证明，论文中只说了这些策略在给定的一些情况下没有问题。在我们的实现过程中，我们发现上述策略在一些比较复杂的碰撞过程中会出现问题，因此我们做了一些比较 trick 的修改，代码中的算法与文章并不完全相同。

### 3.2.2 Dynamic Remesh

模拟过程中，动态的网格重建主要是交由第三方库 Arcsim 完成，而网格的处理并不是这篇模拟文章的重点，因此我们在本节中只介绍如何根据碰撞检测添加新的欧拉-拉格朗日点。

1. 障碍物的点与布料的面发生碰撞，首先依据碰撞检测获得碰撞点的位置，即碰撞点在欧拉空间下的坐标。如果该碰撞点位于三角形的边上，则分割该边，并新加入欧拉点，如果该碰撞点位于三角形的内部，则分割该三角形。新加入的点各项属性依据 barycentric 坐标对周围的点做加权平均。

2. 障碍物的边与布料的边发生碰撞：依据碰撞检测获得碰撞点在该边上的一维坐标，分割该边以引入新的欧拉点，新加入的点的各项属性等于该边两端点的加权平均。

3. 障碍物的面与布料的点发生碰撞，不引入新的点。

### 3.2.3 Velocity Transfer

在网格结构发生改变之后，需要对不同类型的点做不同的速度更新，才能代入到运动方程中求得新的速度。在模拟中的每个循环中，都会储存上一次循环的网格结构，以判断当前网格的每一个点是新加入得到的还是原先就有的。在做 velocity transfer 时，遍历当前网格的每个顶点，然后按照如下策略更新速度。

1. 如果当前的点在之前的网格中存在。

(a) 如果当前的点在上一次循环中是欧拉-拉格朗日点，则该点的状态在当前循环退化为拉格朗日点，并将欧拉速度转化为拉格朗日速度，该点逐渐远离 sharp feature。

(b) 如果当前的点再上一次循环中是拉格朗日点，这是最普通的情况，不做任何变动。

2. 如果当前的点在之前的网格中不存在。

(a) 如果当前的点不是欧拉-拉格朗日点，则说明该点是 Arcsim 调整网格结构时新引入的拉格朗日点，则其速度直接等于所在三角形三个顶点的加权平均。

(b) 如果当前的点是欧拉-拉格朗日点，则判断该点的状态，如果该点是由分割两个欧拉-拉格朗日点构成的边得到的，则其拉格朗日速度和欧拉速度直接等于两端点的平均；如果该点是碰撞造成的新加入的点，则首先对其周围的点速度作加权平均，得到该点的世界速度。然后解一个最优化方程，使得其速度的欧拉分量尽可能大，得到欧拉速度分量和拉格朗日速度分量，即为该点的状态。

### 3.3 Constraint

在完成上述一系列复杂的过程之后，要得到下一时刻状态的速度分布，不能仅仅只解一个线性方程组就可以了，还要加上一系列的约束条件，解一个带约束的优化问题。在实现中，解带约束的优化问题所使用的是第三方 mosek 数值计算库。

其施加约束条件的流程如下：

---

**Algorithm 2** EOL Constraint Generation

---

```

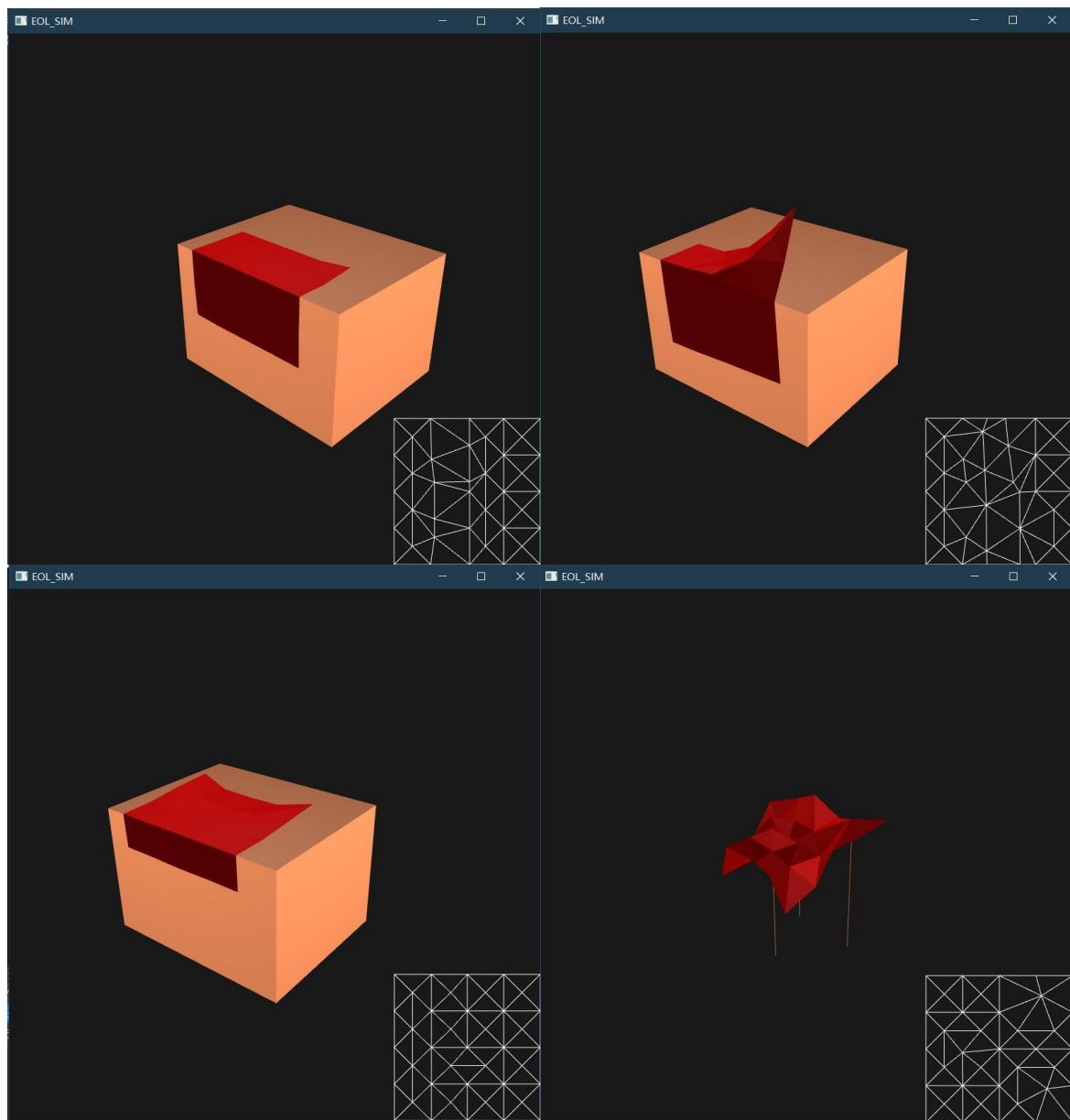
1: for each EOL vertex  $v$  do
2:   if  $v$  colliding with box corner then
3:     Lagrangian Constraint:  $\mathbf{n}_1^T \dot{\mathbf{x}} \geq 0, \mathbf{n}_2^T \dot{\mathbf{x}} \geq 0, \mathbf{n}_3^T \dot{\mathbf{x}} \geq 0$ 
4:     Eulerian Constraint: none
5:   else if  $v$  colliding with box edge then
6:     Lagrangian Constraint:  $\mathbf{n}_1^T \dot{\mathbf{x}} \geq 0, \mathbf{n}_2^T \dot{\mathbf{x}} \geq 0$ 
7:     if  $v$  on cloth border then
8:       Eulerian Constraint:  $\mathbf{b}^T \dot{\mathbf{x}} = 0$ 
9:     else
10:      Eulerian Constraint:  $\mathbf{i}^T \dot{\mathbf{x}} = 0$ 
11:     end if
12:   end if
13: end for

```

---

对于普通的拉格朗日点，还是按照传统的布料模拟方式，根据障碍物和碰撞检测为其加上约束条件，对于欧拉-拉格朗日点来说，如果该点与障碍物的角点发生碰撞，则仍然按普通拉格朗日点处理，在角点相邻面的法向量方向上速度为正；如果该点与障碍物的边发生碰撞，其拉格朗日分量要求在边相邻两个面上的法方向上速度为正，且欧拉速度要满足不能滑出布料打的边界，且欧拉速度沿着边方向的分量为 0（这是要求欧拉-拉格朗日点在沿着障碍物的边方向上滑动这一行为是完全依靠拉格朗日速度来实现的，而垂直于障碍物的边方向上的滑动则完全依靠欧拉速度来做转换）。

## 4 Result



## 5 Summary

实现本篇文章的难度很大，因为物理模拟中涉及到大量的数值计算，以及存在对应的收敛性问题。同时，虽然网格处理并不是本篇文章的重点，但这一部分会有大量的 bug 和 trick 性的东西，调试起来非常麻烦。然而，我们在这个痛苦的过程中仍然收获颇丰，我们仔细分析了作者放在 GitHub 上的 reference code，虽然论文作者的代码中充满了各种 bug、magic number，可读性很差，且在很多 case 下都会 crash，但分析其代码还是让我们有种亲身经历如何写 SIGGRAPH 工程的感觉，让我们去想如果自己是这篇论文的作者，自己会怎么去设计架构，怎么去解决问题。我们在无数次重读 paper 中，清晰地感受到了文章的思路和逻辑顺序，这对我们之后思考在科研中遇到的问题是大有裨益的。