首先简单讲一下录这个视频的原因，Houdini因为它完备的节点属性

接下来我想要开一期可能能够

它会足够简单，可以让你省去写UI，写迭代，写几何处理等过程。但它也足够困难，因为复杂的节点堆积很容易就会产生混乱。但是完成后就可以迅速完成产品应用转化，而且我最近发现用microsolver搭建算法的会非常高效的帮助你理解算法过程，所以这些视频也算是我自己的一种学习记录。

好的，我们先简单的复习一下PBD和XPBD的算法框架。因为相关算法已经有非常丰富的资料了，所以我们一带而过，等到后面具体计算特定约束的时候我们再来详细讲解一下。首先我们应该认识到，PBD与XPBD其实本质上是一种数值解法，通过多次迭代最终求得运动方程的近似解。

对于一个动力学方程而言，我们可以写成这种隐式欧拉的方法，并同样可以表示成最优化的形式。此时我们将弹性能量项定义为一种约束，其中C是约束，K是stiffness matrix，并将K表示成一种alpha逆的形式，也即compliance matrix。通过这种表达形式，我们最终可以将最优化形式变成一种带约束的求解。并定义一个lambda为alpha inverse乘上约束C，作为拉格朗日乘子。最终我们可以得到XPBD的约束求解形式。此外，当stiffness无限大时，我们能够发现，alpha趋近于0，导致约束项会由XPBD退化成PBD形式，也即失去了物体的部分弹性表达。

通常我们可以直接使用Newton法求得精确解，但是求解这种约束本身是一种很痛苦的过程。所以我们还可以采用另一种迭代求解的方式，也即Position based dynamics方法。对于PBD的形式而言，我们可以简单粗暴的直接对约束C做一届泰勒展开，然后使用Schur Complement，最终得到delta lambda和delta x。我们可以采用Gauss Seidel对每个约束j进行迭代，每次找到一个约束并且让他满足约束条件，最终得到一个x的近似解。

通过这种求解框架我们就能看出来，大家通常会诟病PBD不够精确的很大一部分原因是当迭代次数不够时，x的解会更趋近于x initial的状态，也即不考虑弹性内力的初始状态。所以迭代的次数会严重影响到物体模型的形态，也就是解的不够干净。

接下来我们看到对于XPBD而言，多考虑了一个compliance matrix alpha，同时这个时候我们更加小心的认为lambda是一个逐渐变化到最终拉格朗日乘子的变量。因此我们可以将XPBD拆分成这种形式，然后同样的使用舒尔补和GS迭代，可以得到一个比PBD更具有一定物理准确性的计算框架，不过两者并没有差距太大，只需要为数不多的改进就可以了。

此外，GS迭代不同于Jacobi迭代，为了获得更好的并行性，我们可以采用graph coloring方法，每次并行迭代不相干的约束。更多的内容可以参考添添老师的视频和相关论文，下面我们进入Houdini开始搭建Solver。

在Houdini的SOP界面新建一个双层布料的模型，然后在DOP network节点中，创建简单的vellum solver，我们调整若干参数来来更好地模拟布料行为。

进入vellum solver节点内部，我们可以看到里面microsolver组织的非常非常复杂。不过我们从0开始用不到如此完备的功能。

现在把所有的节点都删除掉，然后我们准备开始自己使用microsolver自己新建