

Nucleus in Autodesk Maya

介绍

Nucleus 旨在帮助满足对通用 Autodesk® Maya® 动力学解算器的需求。通过使用通用解算器，不同的动态特效能够以使用独立解算器不可能实现的复杂方式进行交互。而且，该核心解算器是一个独立的组件，对 Maya 的剩余部分没有任何依赖。

1.0 介绍

1.1 什么是 Autodesk Maya Nucleus

Maya 具有几个充当 Nucleus 解算器接口的节点（例如 Nucleus、Maya nCloth 和 Maya nParticle 节点），以建立关系和至/自 Nucleus 转换数据。这不仅有助于支持不同实体之间的更好交互，而且还能利用 Nucleus 内在的稳定解算和碰撞功能。

本文介绍了内部解算器结构以及如何通过 Maya 节点和属性控制解算器的组件，接着详细介绍了如何通过 Maya nCloth 和 nCloth 缓存以及一些常用特效实现最优的结果。

本文的相当部分将介绍 Nucleus 的衣物应用；不过就 Nucleus 而言，nParticles 可以视为 nCloth 的一个子集。粒子可以视为仅没有弯曲或拉伸阻力以及顶点碰撞和约束的衣物物体。

1.2 它不只是衣物

虽然 nCloth 起初的开发目的是帮助对衣物建模，但是有许多自然现象可以使用 nCloth 以及现在的 nParticles 创造。

nCloth 特效

nCloth 不只用于衣物和标记之类的东西，它还可以用于帮助创造能够破裂/撕裂、弯曲和变形的刚性及半刚性物体。此外，还有一个用于空气动力学特效的上升 (Lift) 模型。示例特效如下：

- 衣物（带撕裂）
- 旗帜
- 飞机
- 气球（膨胀/收缩）
- 桥
- Slinkys

nParticle 特效

nParticles 可以用于典型的粒子特效（比如烟和烟火）以及更复杂的流体行为。示例特效如下：

- 烟、火
- 渐隐物体
- 倾泻流体
- 飞溅
- 粘性流体（粘液）

nParticles 上的 Nucleus 约束

Nucleus 约束可以与初始状态 nParticles 一起使用以获得以下特效：

- 沿复杂表面滚动的流体小滴
- 简化的刚体行为
- 悬浮粒子（卷须）

1.3 现在需要帮助吗？请首先阅读此部分内容

如果你遇到 nCloth 设置问题并在这里寻找答案，可首先尝试以下方法：

1. 根据你的单位设置，在 nucleus 节点上设置空间比例。如果你是以米为单位建模的，则将其设置为 1.0，但是如果是以默认的厘米为单位建模的，则将其设置为 0.01。主要特效是把重力设置为正确的数值，因为 Maya 中的动力学始终将单位解释为米，而不管单位设置如何。
2. 对于非弹性衣料，可增加拉伸阻力或应用预置。你可以在衣料节点的属性编辑器中查找预置。
3. 增加 nucleus 节点上的 Substeps 以解决碰撞问题或碰撞区域中的拉伸。
4. 相对于你的空间比例设置，调低 nCloth 上的 Lift 值。目前有一个问题，在大场景比例情况下，Lift 具有比例太大的特效。因此，如果你是以厘米为单位建模的，并将空间比例设置为 0.01，则将 nCloth Lift 值乘以 0.01。

2.0 内核解算器设计

2.1 构建块设计

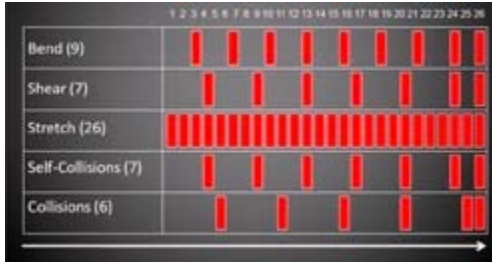
nucleus 解算器采用了一种多用途设计，通过结合更小的任务组来处理复杂的现象，比如衣料。这种构建块方法有助于使同一个解算器适合不同类型的模拟，比如水、衣料或刚体。这些组件部分的解算通过有助于确保大部分设置稳定性的暗示性方法来执行。基本原理是尝试针对大步长进行解算，而不需要小时间步长。这使得模拟可以针对更快的低质量设置相对温和地退化，而不是锁定或变得不稳定。

2.2 解算乘约束的系统

动态实体（比如衣料）一般要求解算器计算许多不同的特效，而且这些特效可能相互冲突。例如，垂盖在某个物体上的衣料的顶点受重力、与该物体的碰撞、衣料的拉伸阻力以及空气摩擦力的影响。重力下推顶点，而碰撞、空气摩擦力和拉伸阻力全都抵制这种向下移动。其中每个特效都是顶点的一个约束，并且必须随时间推移进行迭代以解算动画，虽然在某些特效上进行更多迭代可能会有帮助，比如拉伸阻力。根据属性设置，解算器能以模拟的顺序帮助确定每个特效需要使用多少迭代。下表显示了多个特效的迭代的交叉方式。箭头表示连续或交叉解算的评估顺序。

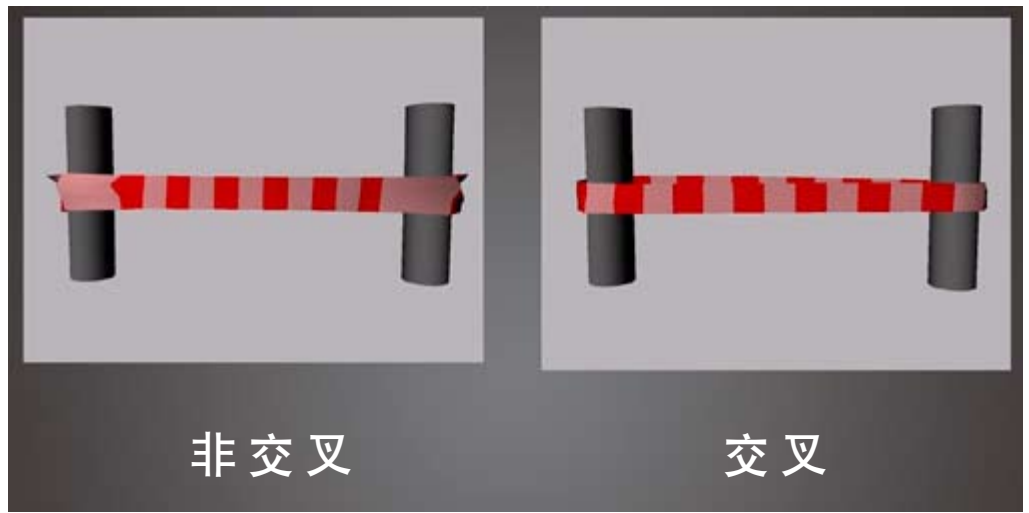


非交叉或连续评估。



交叉（Nucleus 使用的方法）。

碰撞评估的计算一般比拉伸评估更为昂贵，因此进行拉伸评估通常比进行碰撞评估更为高效，特别是在材质高度抗拉伸的情况下。但是，如果没有与拉伸评估交叉的足够碰撞评估，你可能会在碰撞区域看到无法接受的衣料拉伸。



上图显示了拉伸和碰撞的充分交叉如何造成沿碰撞面的拉伸减少。

拉伸和重力的应用有助于使衣料渗透某个物体。在应用碰撞迭代时，它有助于推出碰撞区域，这会在处于碰撞的面中造成拉伸。拉伸阻力的进一步迭代可以修复这个拉伸问题，但是会使衣料回到碰撞状态。一个步骤中的最后迭代始终是碰撞评估，因此如果拉伸迭代中没有足够的交叉碰撞迭代，特效将会是最后一个碰撞步骤有助于显著拉伸衣料，而不管拉伸阻力有多高。增加 Substeps 或最大碰撞迭代数应当可以解决此问题。极端运动或加速将需要解算更多的总体迭代。

在内部，该解算器适合简单的实体，比如点之间的拉伸链接。有一个初始化阶段，在此期间，更复杂的项目（例如约束节点和衣料）会被分解，以便该解算器能够更轻易地领会它们。例如，当某个衣料网格被传递到该解算器时，将会建立点对之间的一套长度约束。约束节点也可以建立相同的点对列表。通常会为诸如碰撞和弯曲阻力之类的特效建立几个列表。Maya 中对应属性的数值通常有助于定义该特效在 1/24 秒（默认情况下对应于单一帧）期间的迭代次数。注意，如果 Scaling Relation 未被设置为 Per Link，则每帧的迭代数也随网格中的多边形数扩展。Per Link 是默认的 Scaling Relation 值，但是 nCloth 属性预置通常使用 Object Space 设置。

2.3 Nucleus 物体的层次结构

该解算器了解如何根据简单的维度层次结构将元素联系起来：

- 点（零维）
- 边（一维）
- 面（二维）

（你可以进一步对体积物体（比如四面体）执行此操作，虽然目前不使用它。）



该解算器知道如何在这些元素之间进行碰撞，并且会考虑厚度（如果进行了设置）。在内部，完整的表面碰撞（面对面）通过结合更简单碰撞的子集来处理：点对面和边对边。碰撞计算可以视为计算加厚组件的空间-时间交叉。

另外，每个元素也可以进行约束或与其它元素联系起来。例如，你可以约束点对点、点对边和边对边。

2.4 解算器初始化和步进

在开始帧处，该解算器可以帮助建立各种结构，比如分界树，以支持高效的计算。而且，在开始帧处，输入物体的初始状态进行了缓存，这有助于解算器（以及其它特效）防止相互渗透分解网格。在模拟开始时，你所定义的约束被分解成可以更快解算的链接列表和低级关系。对于 Slide on Surface 约束，初始化包括计算表面上沿某个法向的最近点，这是一种相对昂贵的计算。注意，在使用约束时，如果你对 dynamicConstraint 节点上的 Enable 属性进行关键帧处理，该约束将在 Enable 打开的帧上重新进行初始化。这有助于让手抓住动态物体。

Maya 动画（即物体的关键帧处理和变形）每帧（或步骤）进行一次计算，而且该解算器所处理的内部 nucleus 结构从这个动画更新。此步骤将被 Nucleus 系统进一步分解为子步骤，即帧内更小的时间细分。动画曲线和变形器在这些子步骤期间不进行评估，虽然它们的效果以线性方式内插在子步骤内。

即使只有一个子步骤，被动物体也有一个在用于碰撞的帧中捕捉形状变化的每顶点速度（你可以把时间拉伸的表面考虑为四维物体）。注意，在某些情况下，一个步骤中的简单动画线性插值可能是不够的。为了帮助处理这个问题，你可以在“缓存选项”中为每个 X 帧设置 Evaluate 的值，以使得基本步长更精细。

在一个子步骤内，根据设置的不同，可能有更精细的特效迭代，比如拉伸阻力和碰撞。这些更精细的迭代在子步骤定义的时间步中执行。例如，如果一个子步骤内有 10 个碰撞迭代，其中每个迭代都将为该子步骤的时间增量执行碰撞计算，而不是以更小的增量进行时间步进。因此，该子步骤代表了解算器使用的最小时间单位。在一个子步骤内，你可以进行更多的迭代，但是每个迭代都会尝试跨该子步骤定义的时间间隔进行解算。值（比如拉伸阻力和弯曲阻力）是相对于时间而不是子步骤定义的，虽然改变子步骤会影响这些属性的解决方案的质量，并因此改变效果。但是，解算器会尝试以独立于子步骤的方式保护定义的拉伸级别。默认情况下，如果你将拉伸阻力设置为 10，则每帧有 10 个拉伸阻力迭代。如果 Substeps 设置为 2，则每个子步骤有 5 个拉伸阻力迭代。如果 Substeps 增加到 10，则每个子步骤有一个拉伸阻力迭代。但是，如果 Substeps 增加到 20，现在每个子步骤将有一个拉伸阻力迭代或每帧有 20 个拉伸阻力迭代。拉伸阻力评估的强度会以总体拉伸阻力保持大致相同的方式每步骤递减，即使有两倍数量的评估。

注意，除了拉伸阻力、弯曲阻力和刚性等特效之外，碰撞和自我碰撞迭代也是相对于时间定义的。因此，如果碰撞迭代数（最大碰撞迭代数）设置为 10，Substeps 设置为 2，则每个子步骤有 5 个碰撞迭代。在某些情况下，增加子步骤实际上可以帮助加快模拟解算。这是因为总体碰撞迭代数可能保持不变，但是每个迭代均在更小的时间步上进行，由于每个步骤的碰撞对更少，因此更小的时间步更容易计算。（碰撞对是在空间-时间中具有重叠边界的元素的列表。对于自我碰撞，当厚度大于面大小时，该列表会变大，特别是完整表面自我碰撞，这会增加计算时间。）子步骤包含重新计算所有物体的内部分界树的计算价格，十分昂贵。这会造成具有略高于 Substeps 值的碰撞迭代数有助于实现最大效率的情形。

另一个有用的经验法则与碰撞迭代和拉伸有关。一般来说（假设物体的 Scaling Relation 位于 Link 的默认设置），碰撞迭代至少应为 Stretch Resistance 值的 $1/10$ ，以便每 10 个拉伸阻力评估至少发生一次碰撞评估。

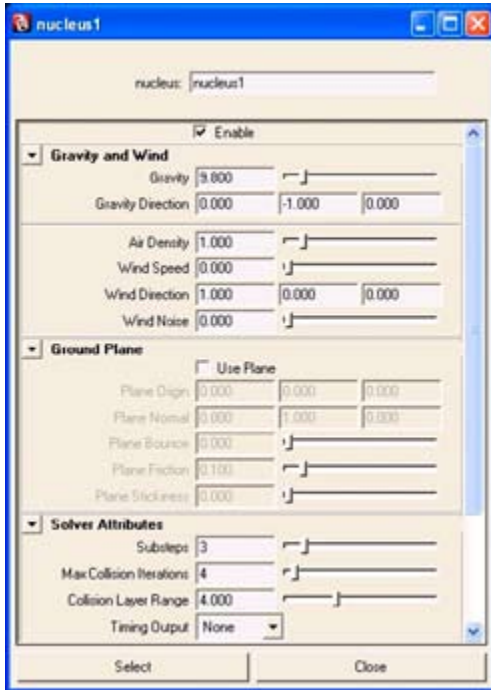
在模拟具有最小约束的 nParticles 时，这意味着只有与物体的碰撞，没有自我碰撞，Substep 值为 1.0 和 Max Collision Iterations 值为 1 可能是足够的。默认的 Substeps 值为 3，因此，对于简单的粒子模拟，它通常有助于降低 Max Collision Iterations 和 Substeps 的值。当你具有需要在网络中传播的约束时，例如同时碰撞中的多个元素，比如堆叠球或在网格上处理拉伸阻力的互连链接组，通常需要更高的迭代数。一般来说，鉴于此类网络的复杂性，迭代要求会急剧增长。

3.0 Nucleus 物体的 Maya 相关性图形结构

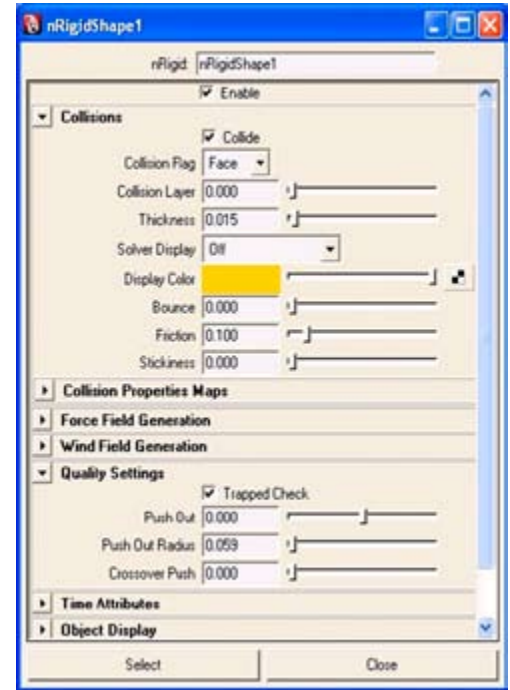
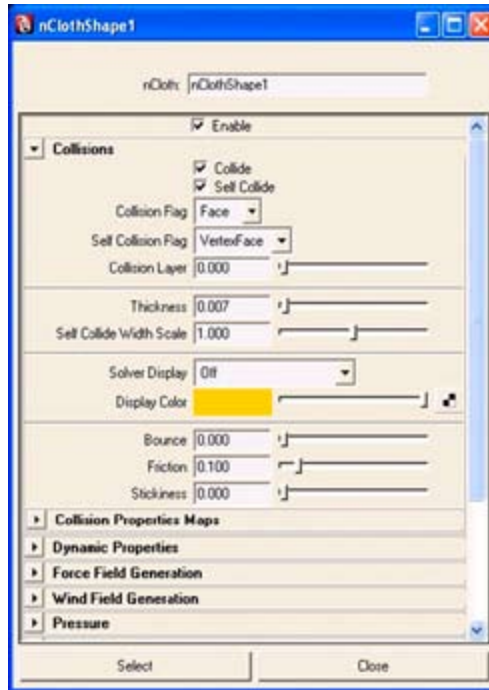
3.1 节点类型和继承性

Maya 中基本上有四类 nucleus 节点：Nucleus 解算器节点、物体节点、动态约束节点和 Nucleus 组件节点。

3.1.1 Nucleus 解算器节点



3.1.2 Nucleus 物体节点



目前，只有一种风格的解算器，而且它的节点类型是 nucleus。nucleus 节点连接到参与解算的实体，比如 nCloth 和 nParticle 物体，并且包含影响其连接实体评估的属性。nucleus 节点可以视为一个包容世界或宇宙。只有在连接到同一个 nucleus 节点时，物体才能进行交互。nucleus 节点还包含影响连接到它的物体的全局值，比如风和重力。

在单个物体上，风和重力可以忽略，而且与 Maya Fields 相似，它们可以应用到单个 Nucleus 物体。但是，Nucleus 解算器中的风模型了解 Maya 风场并不知道的表面方向。这允许让衣料在面对风时具有比边更多的空气阻力，这对实现逼真的衣料和风交互十分重要。而且，Nucleus 风和重力是在子步骤内计算的，而外部场每帧只计算一次。因此，Nucleus Wind 可以比风场更好地与特效（比如拉伸）集成。（通过 nucleus 的自定义和扩展场，这种情况将来可能会改变。）Nucleus 节点是在时间改变时帮助为每个新步骤调用解算器的节点。

Nucleus 物体是由 Nucleus 操作的实体。目前有 nParticle (nParticleShape)、nCloth (nClothShape) 和 nRigid (nRigidShape) 节点。nRigid 节点目前只能帮助定义被动碰撞物体，但是如果刚体模拟被添加到 Nucleus，它可能会在某个时候具有一个活动开关。nCloth、nRigid 和 nParticle 节点从一个共享基类继承了属性和行为，而后者是从 Maya 经典粒子继承而来。这允许 Maya 在适当时共享现有的功能，比如经典粒子发射器、表达式、事件、nParticles 渲染以及 Maya nCloth 和 nParticles 场。目前，该结构确实造成了节点具有一些不使用的继承属性。这些属性在 Maya 用户界面 (UI) 中受到了抑制，比如属性编辑器，但它们在其它地方可能仍然是可见的，比如 Channel Box。一些继承的功能目前仍然未得到完全支持，比如 nCloth 上的粒子表达式。