

购买协议:

1. 购买caDesign设计《面向设计师的编程设计知识系统PADKS(Programming Aided Design Knowledge System)》任何一本电子书需要遵循版权声明，任何侵权的行为我们将采取必要的法律措施；
2. 购买者从购买之后起，电子书仅供自身阅读，任何传播侵害作者和被授权方使用的行为，我们均将采取必要的法律措施；
3. 对于购买者caDesign提供的服务请参看cadesign.cn官网，caDesign设计有权对服务项目做出修改，最终解释权归caDesign设计。
4. pdf非加密版仅在购买者购买加密版电子书之后，由于系统升级等原因无法打开原购买的加密版电子书，caDesign设计所采取的替代措施，不用于任何销售行为。

-caDesign设计

作品登记证书

登记号：国作登字-2014-L-00133782

No. 00133782



作品名称：面向设计师的编程设计知识系统 作品类别：其他
PADKS

作 者：包瑞清

著作权人：包瑞清

创作完成时间：2014年01月28日

首次发表时间：2014年01月28日

以上事项，由包瑞清申请，经中国版权保护中心审核，根据《作品自愿登记试行办法》规定，予以登记。

登记日期：2014年04月03日

登记机构签章



版权声明:

为了保护《面向设计师的编程设计知识系统PADKS(Programming Aided Design Knowledge System)》系列课程/研究成果作者以及购买者的权益，向中国版权保护中心申请作者著作权保护，任何未经作者或者北京沐本科技有限责任公司的允许不得进行任何形式的再利用，任何侵权的行为我们将采取必要的法律措施。任何发现侵权PADKS系列课程/研究成果的行为，请联系caDesign.cn设计（沐本）。

E-mail: cadesign@vip.163.com;

Telephone number:010-64848493, 13811960916;

《面向设计师的编程设计知识系统PADKS》作品说明书:

《面向设计师的编程设计知识系统PADKS(Programming Aided Design Knowledge System)》旨在建立面向设计师（建筑、风景园林、城乡规划）编程辅助设计方法的知识体系，使之能够有效的协助设计者步入基于编程的逻辑构建过程方法领域，实现设计方法的创造性改变和设计的创造性。

《面向设计师的编程设计知识系统PADKS》包括十本书三个方向的内容，其十本书分别为：

1. 《基于编程的逻辑构建过程》；
2. 《Grasshopper参数化逻辑构建过程》；
3. 《Grasshopper参数模型构建》；
4. 《Grasshopper参数设计方法》；
5. 《学习Python_做个有编程能力的设计师》；
6. 《编程景观》；
7. 《折叠的程序》；
8. 《地理信息系统在风景园林和城市规划中的应用》；
9. 《ArcGIS下的Python编程》；
10. 《生态辅助设计技术》。

《基于编程的逻辑构建过程》，该研究成果架构了“面向设计师的编程设计知识系统PADKS”的基本框架，分别为基于Grasshopper节点式编程语言和Python语言的参数化设计研究，基于ArcGIS以及Python语言的地理信息系统辅助规划设计的研究和计算机辅助生态分析设计技术三个主要方面。

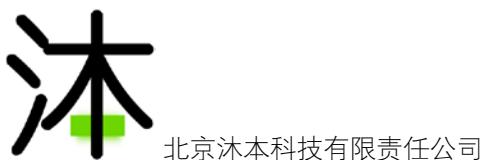
《面向设计师的编程设计知识系统PADKS》为作者提出面向设计师（建筑、风景园林、城乡规划）编程辅助设计方法的知识体系，梳理了目前计算机辅助规划设计的主要方法，并互相构建联系使繁复而庞杂的知识从编程辅助设计的方向成为一个能够有效辅助规划设计的知识体系。

Everybody should learn how to program a computer,
because it teaches you how to think, and allows
designers more creative!

“编程让设计更具有创造力！”

-caDesign.cn设计

caDesign.cn设计网络服务平台:<http://www.cadesign.cn/>



北京沐本科技有限责任公司



微信公众平台服务号

@caDesign.cn

本书以电子文档形式制作和出版，未经作者
允许，本书的任何部分不得进行任何形式的
再利用。

新浪微博网页版:<http://weibo.com/cadesign>
手机版:<http://weibo.cn/cadesign>

售价:¥97

版权所有,侵权必究

Grasshopper

参数设计方法

Algorithmic Design Methodology

包瑞清

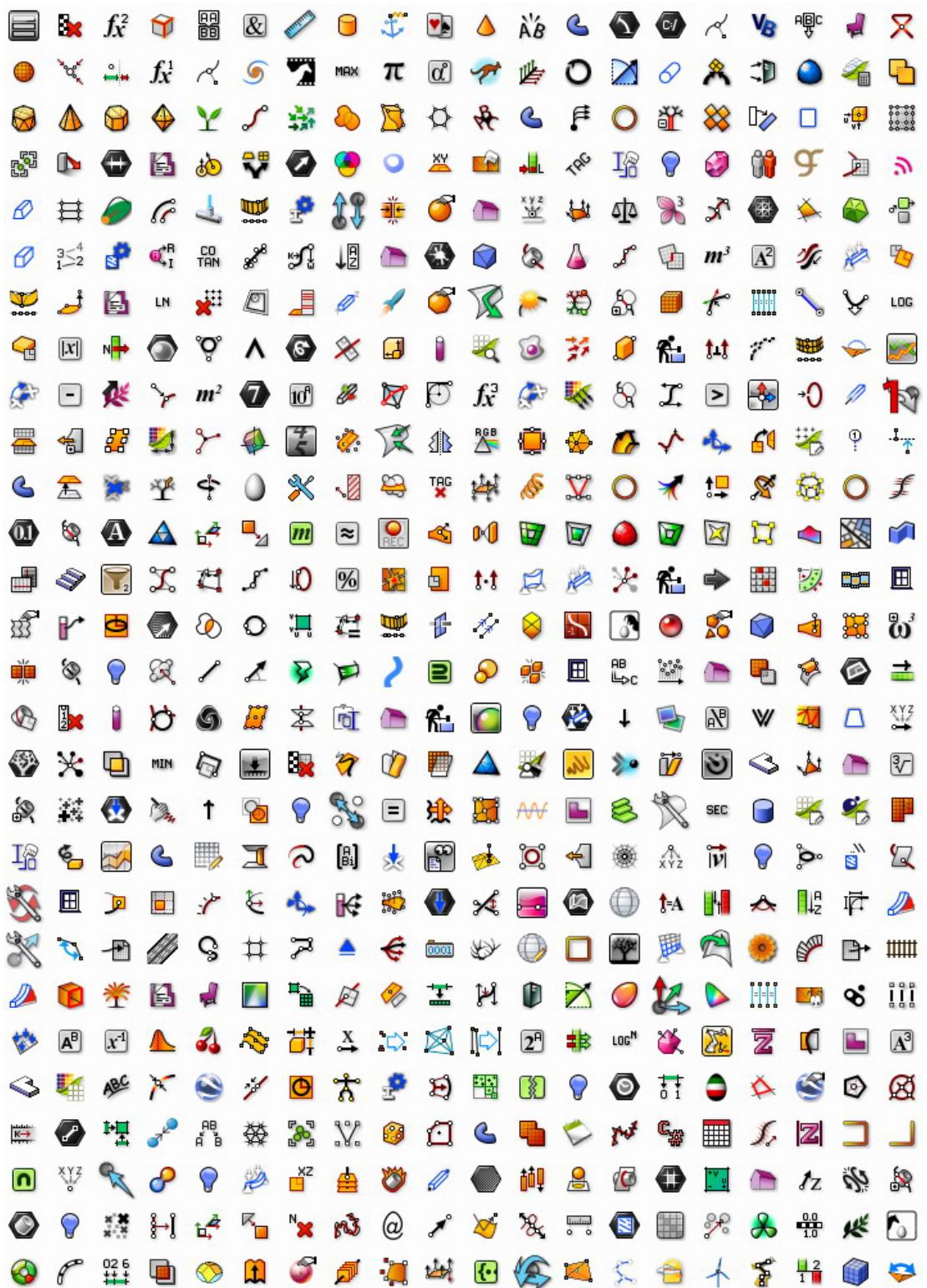
caDesign.cn设计

PD_C

2013-10-20







任何事情的发生发展似乎都存在一个契机，就如这本书的完成一样。2009年的时候，设计团队希望能够应用目前世界上参数化设计方法来完成一个设计项目，而这个研究领域刚刚开始也没有几年，绝大部分的技术信息也都是欧美国家在不断地拓展中。在项目进行的时候，最开始我们试图使用3DMAX等影视三维软件来处理模型构建问题，但是很快被击退下来，那时对脚本语言编写程序，建立模型还认为是很高深的领域，只有专业的程序员才可以做到。在非常迷茫的时候，就不断的在网上搜索信息，记得之前有个项目是使用了RHINO，于是从此开始着手，但是，对于繁琐的相似性体构建，如果仅仅依靠RHINO本身处理就有些难度，同事提到了Grasshopper（GH），也就是从那时开始真正接触以节点化组件处理参数进行模型构建。

接触GH时，它带给了我们两个方面的重要信息，一个就是不使用脚本语言而进行参数化模型的构建，另一个可能是大家最容易忽略的节点模式的操作方法。到现在，对于参数模型构建的认识也不应该停留在纯粹的几何建立上，例如参数化控制下的几何变化，参数所打开的领域不仅是几何上的问题，应该说是一个多专业联合，参数化平台关键控制的是数据处理，而地理信息系统，生态分析，动力学，结构分析等等设计相关学科基本都可以转化为一系列的数据，例如地形的坡度可以转化为栅格数据，每个栅格存储一个坡度信息，或者太阳辐射强度分析，以及结构中风载荷可以为有方向的矢量，正是所有的一切似乎都可以转化为数据，而数据都是数字的表现，只是不同环境下所代表的意义不同，因此基于数据处理的参数化平台就可以把所有这些都统一起来，协同进行设计。可以在设计模型构建的同时，与结构优化同时处理，与生态日照分析协同处理，这样的一个设计必然更加合理。从传统中的定向走向定量，从各项分析的分离处理，到一个平台下的协同完成，可以说是这个世纪，设计发展的一个重要趋势。

而节点模式的方法，则将设计师从面板中解脱出来，通过各个单独的组件不同的连接方式，极大的增加了模型构建的灵活性，更为重要的是，软件的发展也在不断地朝着这个方向努力，例如Houdini，ARCGis的地理处理，Vue的材质设置，都具有节点模式，因此

在软件学习中，操作模式大同小异，很快能够掌握，而是将更多的时间用于具体组件的功能上。

GH的确为设计师发展设计带来了无穷的潜力，但是欲望的无止境，总是试图探索和解决新的问题，这些问题在目前的GH组件中又找不到答案，如果等待GH未来开发此种组件，一是不确定是否会有，二是时间的问题，因此就需要去接触一开始认为那设计师不可能涉及到的领域—程序语言（脚本）。

目前语言的种类很多，C++，C#，Java，VB，Python，如果只有一种就好说了，但是这么多，对于设计师来说适合于哪一种是很关键的，把精力花费在无用的或者对未来帮助不大的语言学习上，就有些划不来，毕竟设计师的专业与程序员是有很大差异的，更何况我们也并不是要成为程序员。最为直接的判断方法就是看主流模型构建软件共同支持的语言种类，那就是Python，而且大多数软件支持Python的另一个原因就是它可以被认为是所有程序语言中最易掌握的，少则几天，多则一个月就可以掌握用于协助设计模型构建部分的主要语言部分。同时，如果你学的是VB，但是支持该类的软件平台就有几个，如果使用其它的软件，恐怕就不得不不再学一门语言了，那简直就是时间的极度浪费。

基于Grasshopper的参数模型构建手册有些不同于大家市面上所看到的各类教程书籍，这也不是本书的目的，对于Grasshopper等教程完全可以从官方网站上获得并加以学习，本书最希望的是有一定参数构建基础的设计师处理各类模型建构时能够从本书中获得有益的信息，例如查找主要结构术语，结构优化方法；辅助对该领域了解甚少的设计师能够从本书中获得学习的途径，例如参数能做什么，可以查找哪些网站，可以阅读哪些书籍。同时书中配套光盘包含了书中所有的程序，可以直接使用来分析构建模型。参数模型的构建也不再仅仅是几何推敲的形象思维，其中包含的还有逻辑数理思维，也就是设计师要在这两种思维中不断的转换。

SketchUp的普及是迅速的，因为学会它只需要两天的时间，参数领域的学习却需要成倍的时间，因为这

不仅仅是一个学科的问题。所以在无止境的加班，追求直接利益的条件下，设计师早已身心疲惫，而选择能够快速掌握的软件，无暇顾及更深的领域。但是，是否想过，一旦掌握了参数的方法，工作效率能够成倍的增长，如果不想一直无止境的在模型构建上较劲的话，是否可以考虑学习一下参数模型构建呢。

对于参数领域的探索是无止境的，它会让你从一个想法又蹦出一个想法，这也是基于参数的最大魅力，一个充满冒险却让设计师乐于探索的领域。一本本书是无法涵盖该领域的所有内容，因为这不再是一门学科的问题，而是多专业的联合，随着时间的推进，研究的深入，设计师的实践，以设计（生态规划，城市规划，风景园林规划，建筑设计）为主体协同地理信息系统，生态分析设计，参数模型构建，计算机语言的多学科发展必然是未来设计发展的方向。

对于一个多学科综合的设计领域，涉及到的知识牵连甚广，本书的编写和程序必然会有疏漏甚至错误之处，请读者不吝指正。

A handwritten signature in black ink, reading "Richie", centered below the author's note.

程序下载地址:caDesign.cn网络服务平台:<http://www.cadesign.cn/>

M-RHINO模型.3dm

Rhinoceros 5.0

G-Grasshopper文件.gh

Grasshopper Version January-12, 2012, 0.8.0065

P-Python文件.py

Python 3.2.2

T-文本文件.txt

H-Houdini文件.hip

Houdini Master Version 11.0.581

V-VUE文件.vue

VUE 9.5

A-ANSYSWorkbench文件.wbpj

ANSYS Workbench 2.0 Framework Version:12.0.1

CONTENTS: 目录

序	
011 第一部分	基于Grasshopper的基本构建
	1.在RH平台上构建基本的几何模型 013
	2.展览建筑部分参数模型的构建 015
	3.棚架部分参数模型的构建 023
	4.空间点的定位，二维DWG文件输出 与V-Ray渲染表达 029
037 第二部分	Python与Grasshopper的联合
	1.Python Script 039
	2.GH Python 053
	3.建筑的表皮结构 058
	4.建筑结构 063
068 第三部分	协同设计方法
	1.GEKO 070
	2.地理信息与GH 083
	3.基于动力学的模型构建 102
	4.结构协同 139
	5.扩展组件 210
220 第四部分	进化计算与迭代
	1.Galapagos 222
	2.HoopSnake 233
238 第五部分	表达
	1.图解 240
	2.渲染 242
246 附录	





第一部分 基于 Grasshopper的基本构建

<http://www.grasshopper3d.com/> GH官方网站

M-01_01

在计算机三维制图技术发展之前，设计最基本的工具就是图版制图了，同时结合实体模型的制作来推敲和表达设计。在AutoCAD出现之后，部分制图的工作就移植到计算机中去处理，使制图工作比尺规作图更加容易，修改起来也不是很费劲。现在，AutoCAD仍旧是，以后很大程度上也是设计最基本的制图软件。但是，使用AutoCAD时，虽然这个计算机辅助软件具有一定的三维制图功能，但是不是很强大，在三维中推敲方案还是不大现实。这个世纪初，对设计界设计方式影响最大的就是SketchUp，下面都简称为SU。它使设计师在三维中便捷的推敲设计方案成为可能，这个时候，像3DMAX和MAYA虽然已经存在很久，但是SU能够快速在设计师中普及的一个重要原因应该是其易学性，大概2-3天时间，一个不懂三维软件操作的人，基本都可以学会。同时，SU后来被Google收购后，与Google Earth结合，网络组件的支持，使该款三维设计软件功能日益变得强大。使用SU的三维推拉技术，一般的设计模型构建基本都可以实现，也能够较真实的表达材质。目前，可以使用Lumion，一款后期处理软件与SU结合，将SU模型导出为Dae格式的文件再在Lumion中导入处理环境和材质，进行渲染，可以使后期的渲染效果更加的真实。

现今，基本所有的建筑，景观设计师都在使用SU处理方案设计，但是，不可否认的是，设计师在使用SU辅助设计模型构建还不是那么得心应手，操作简单易学是其最大的优点，同时也丧失了更多的几何构建方式，尤其对于重复性构件，非线性的设计，SU基本会花费更多的时间在重复性的劳作上，甚至对于较复杂的设计，恐怕无能为力，后期的修改更是件不容易的事情。如图所示的设计在SU中构建的话，即费时又费力。这时，设计师应该去寻找更加合适的三维模型构建软件，我们在该项目中借助了Rhinoceros与Grasshopper来处理相对复杂的三维模型构建工作。

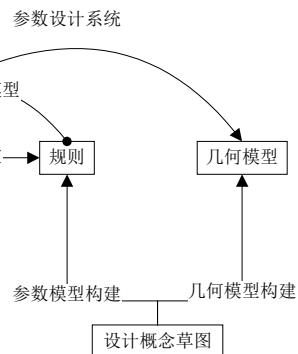


Rhinoceros (RH) 中文名称是犀牛，由美国Robert McNeel & Assoc.开发的PC上强大的专业3D造型软件，可以广泛应用于三维动画制作，工业制造，科学研究以及机械设计等领域。从设计稿，手绘到实际的产品，或是一个简单的构思，提供的曲面工具可以精确地制作所有用来作为渲染表现、动画、工程图、分析评估以及生产用的模型。不受约束的自由造型3D建模能力，为设计师提供更广阔的设计创造领域，这正是我们采用Rhino作为三维建模基本软件的原因之一。



Grasshopper (GH) 是基于Rhino环境下运行，采用程序算法生成模型的插件。与Python 和Rhino-Scrip(Rhino的脚本语言)不同，不需要太多程序语言的知识就可以通过一些简单的流程方法达到设计师所想要的模型。把方案逻辑与建模过程联系起来，并通过调整参数直接改变设计模型的形态，可以实时观察参数和模型之间的变化关系，更具有直观性，是极具有参数化设计的软件。

在详细的介绍使用RH与GH构建模型之前，需要对参数控制下的三维模型构建有个基本的认识。最初可以表达设计师设计意图的设计概念草图是三维模型构建的前提条件，有了基本的想法后，再使用SU构建设计模型，是一种直观的构建方式，比如在设计构思时，希望有面景墙，几步台阶，那么就如在搭建实体模型那样，用SU拉出一个矩形几何，赋予材质当做景墙，构建台阶时，可以拉伸出一个踏步，再复制就可以了。可以说这种构建模型的方式是直观的操作模式。如图右侧部分的流程。直接使用RH本身的平台来构建同样的事物时，也是一种直观的构建模式，可以用推拉的方式构建，也可以用RH中其它的建模命令，例如单轨扫描，或者放样工具实现，达到同一个目的可以使用的方法比SU来说，就多出很多，具有更多的灵活性。但是，构建顶部的廊架时，只是在SU或者RH中以上述直观构建模型的方式，就会相当繁琐。这时，需要以RH为基本软件平台，利用它的插件GH来处理，构建的时候需要思考的不仅是设计构思本身，还要考虑将设计构思转换为三维模型的参数模型构建规则，不可否认的是，绝大部分的设计构思过程，例如搭建棚架顶部的构筑，需要有什么样的梁柱体系，椽构件又是如何布置的，都可以直接作为参数模型构建的规则，而梁柱椽的尺寸，间距或者其它的约束关系可以作为可调整的参数，约束规则的构建，形成参数模型，GH的参数模



型可以与RH平台的几何模型构建直接的联系，也就是说RH中的梁柱椽等几何构件对应着GH中某一个参数，如果在参数模型中调节椽构建截面的大小，在RH中对应的椽构件几何模型也会实时的变化。这样，参数设计模型不仅可以完成繁复的模型构建任务，与几何模型实时的对应变化关系，方案的推敲和调整也相当直观。

1 在RH平台上构建基本的几何模型

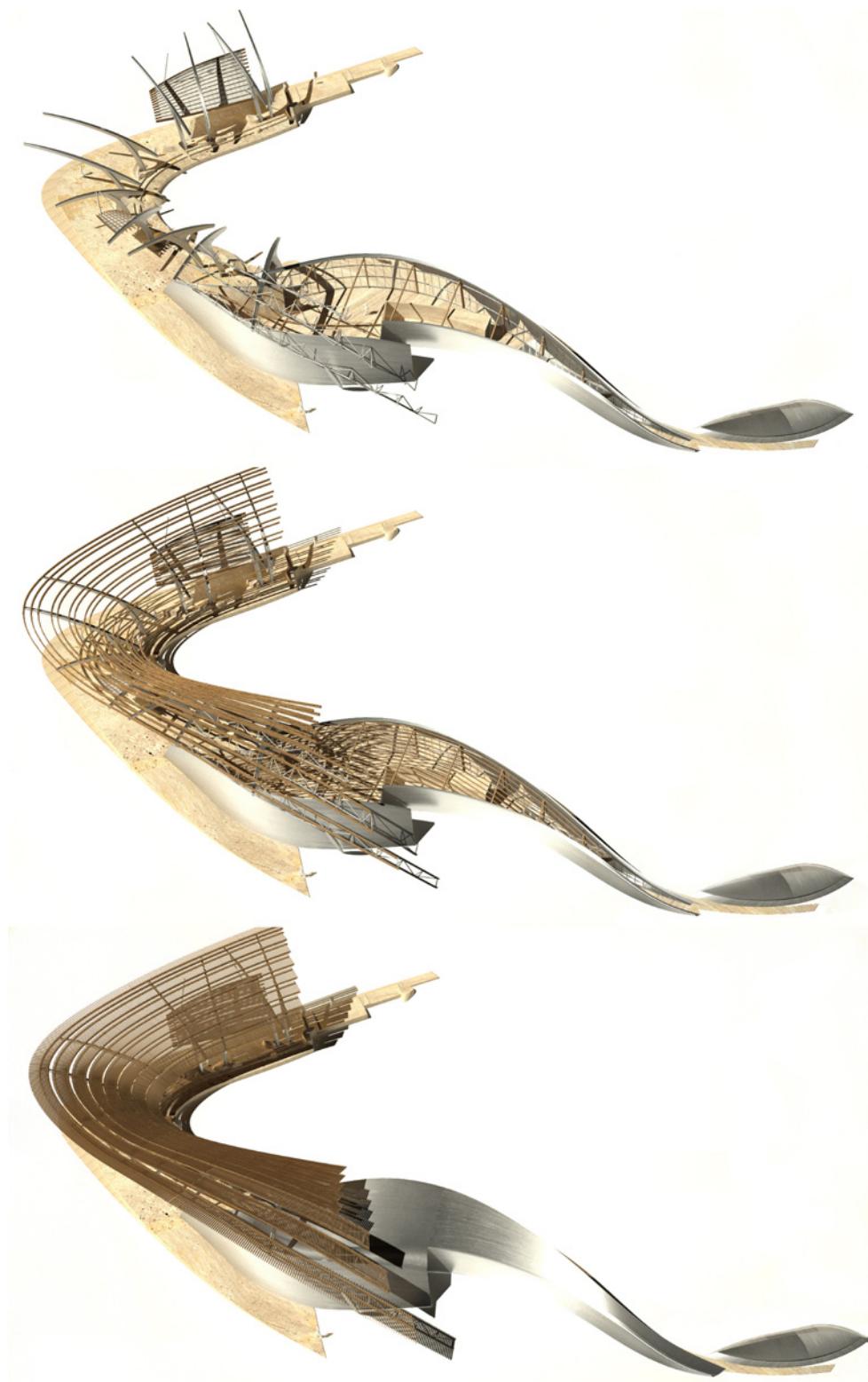
根据参数模型规则建立的不同，参数调整的方式也会有很大差异，在相应的一些参数模型研究中，可以将日照，热辐射，风流动，人群集聚的特点作为规则的参数输入，与几何模型的变化构建联系建立参数模型。对于例举的这个设计，参数模型构建的规则与设计构思过程或者建筑建造的过程基本一致。在使用GH构建参数模型之前，需要输入RH中基本的几何模型作为参数模型构建的初始条件。

参数模型的初始条件是由几个可以控制设计几何模型的基本面组成，展览建筑部分为几个面组成的几何体，棚架部分的基本几何模型，为一个曲面。

从图中的基本几何体我们能够看出设计的大概形态，一个大鹏飞翔和一个水中鱼儿游动，一沉一浮，就是设计最初的设计概念。其它详细构架部分，例如具有不同尺度但形式一致的梁架，吊棚的格栅，玻璃幕墙都是可以使用参数模型来完成。建筑部分的墙体则是在RH中以直接建立几何模型的方式完成。



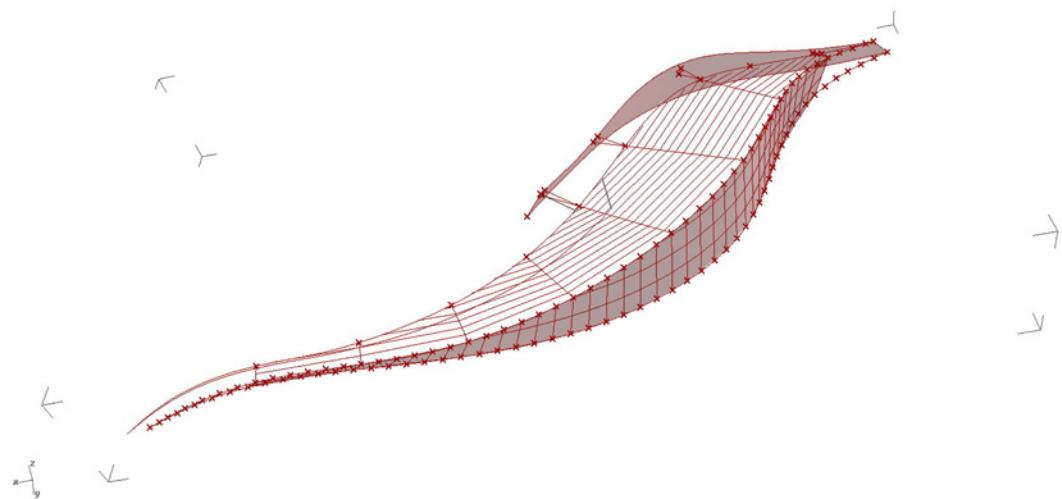
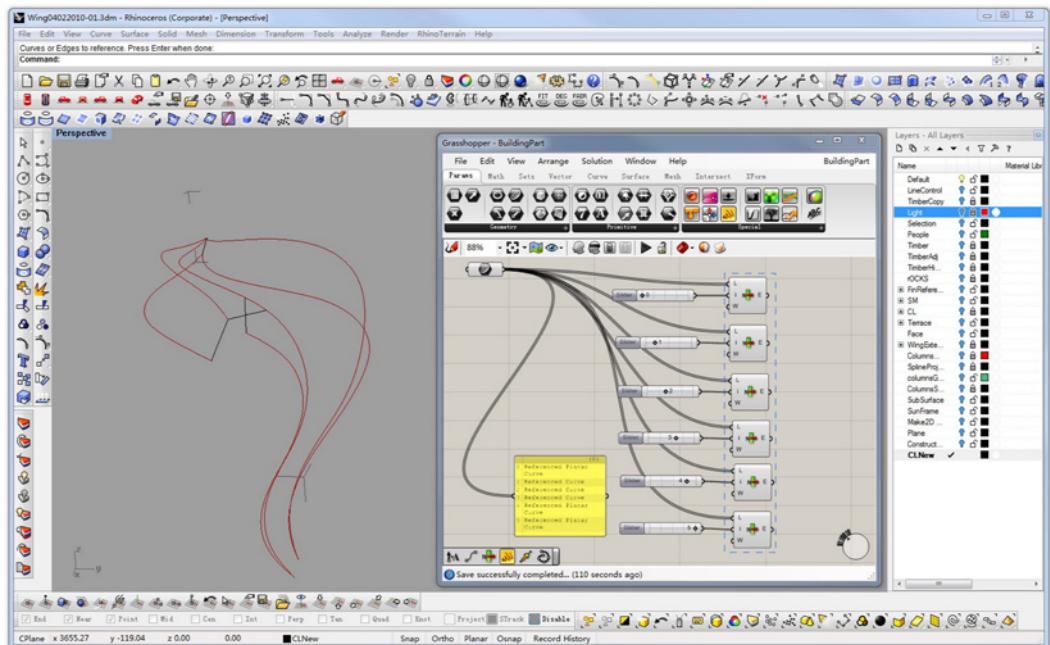
注：Rhinoceros的学习可以挑选任意一本市面上符合自己学习习惯的教程即可，也可以通过网络资源寻找，带有视频教学的教程可以提高学习的效率；



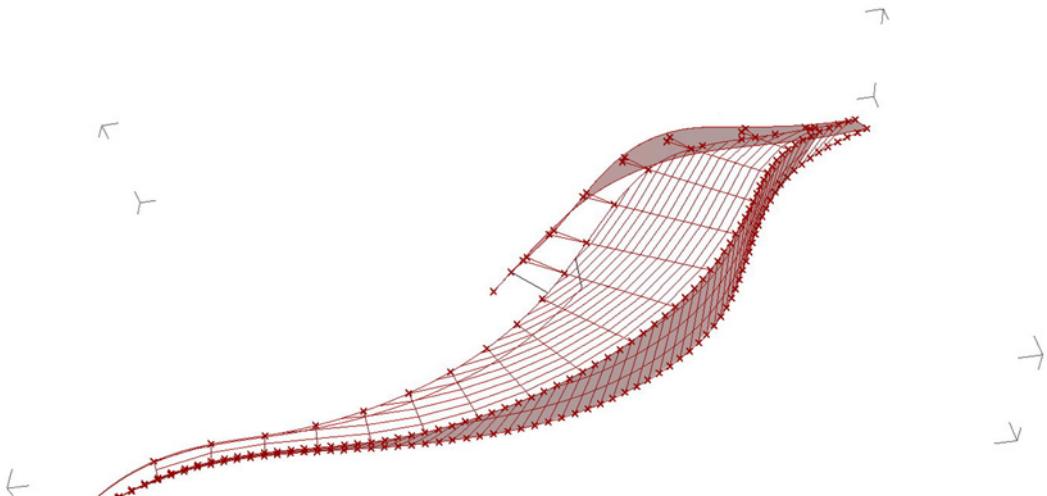
2 展览建筑部分参数模型的构建

在使用GH建立参数模型的时候，总是希望能利用最简单的初始几何条件完成整个建筑的构建，例如该案例中建筑部分的5个主要曲面或者构成这五个曲面的边线来完成所有的建筑元素。不可否认，利用参数建模可以实现，但是规则会变得较为复杂，参数模型构建的一个主要目的，是能够协助设计师最快的建立几何模型和方便进行后期修改。如果构建的参数模型过于复杂，反而在后期的调整过程中不宜修改，这个时候可以将能够使用参数模型建构的部分分成几个程序处理，相对会变得容易一些。建筑部分的构建，使用基本几何体的边线作为初始条件，输入到GH中。在RH空间中红色显示的边线说明已经输入到GH空间里，在GH空间可以看到很多小滑块，图中虚线蓝色边框圈住的部分就是对应RH空间中的6条边线，接下来的参数模型构建就是以这六条边线为初始条件。

注：Grasshopper的学习可以从官方网站上下载其所提供的教程或《基于Grasshopper的参数模型构建手册-I》加以学习；



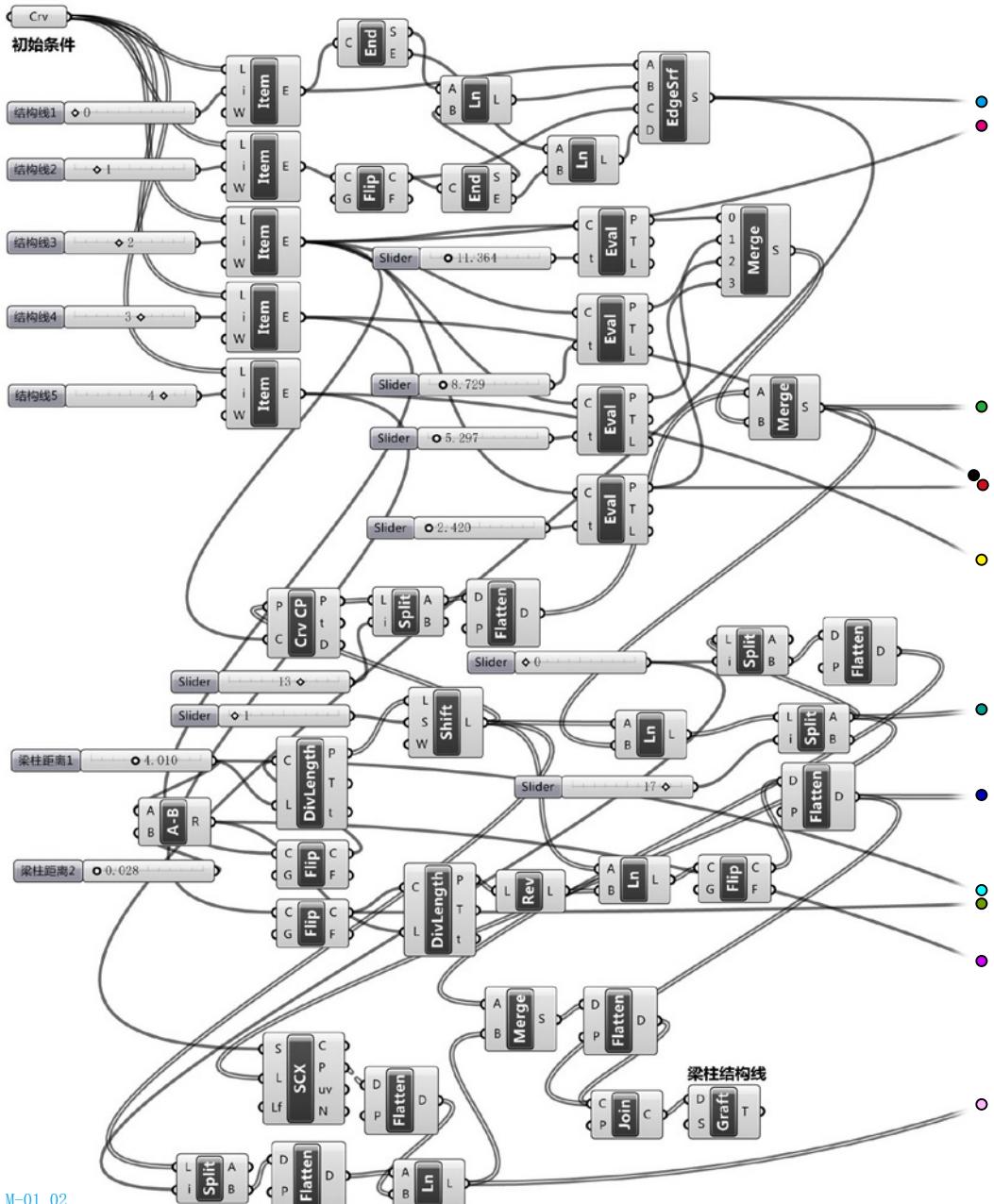
在参数模型构建的过程中，首先要找到各种几何构件的结构线，结构线间的逻辑关系就是参数模型的主要规则，如图所示的红线，包括了主要梁柱、椽构件、玻璃幕墙格栅的结构线部分，然后再在结构线基础上构建几何模型形成体积部分的参数规则。在结构线部分对应GH文件的参数模型，我们用不同的颜色圈出形成各结构线的规则部分，并标明对应哪个构件。各结构线间的逻辑关系不是独立的，梁柱的结构线影响椽的长度和玻璃幕墙的栅格，因此在处理参数模型的规则时，先建立对整个架构有主要影响的几何构件，本例中是梁柱部分的规则，然后受主要结构影响的其它构件部分以主要规则为基础，进一步建立即可，从而形成一个联动的关系。当希望梁柱间的间距小一些时，由基于主要规则的其它次要规则所构建的结构线也会实时的进行调整，不需要像手绘制图或者SU建模那样，不得不重新绘制以使椽，幕墙栅格部分适应梁柱的变化。



梁柱数量由原来的8个修正为17个，所做的工作只是修改GH文件的参数，工作效率能够成倍的增加。不过，达到一个目的，例如建立梁柱结构，根据GH插件所提供的不同工具和不同的组合方式，可以有多种解决方案，也就是可以构建不同的规则达到同一个目的，规则构建方式的不同，在进行几何架构数量调整时，参数调整的难易也会有差异。

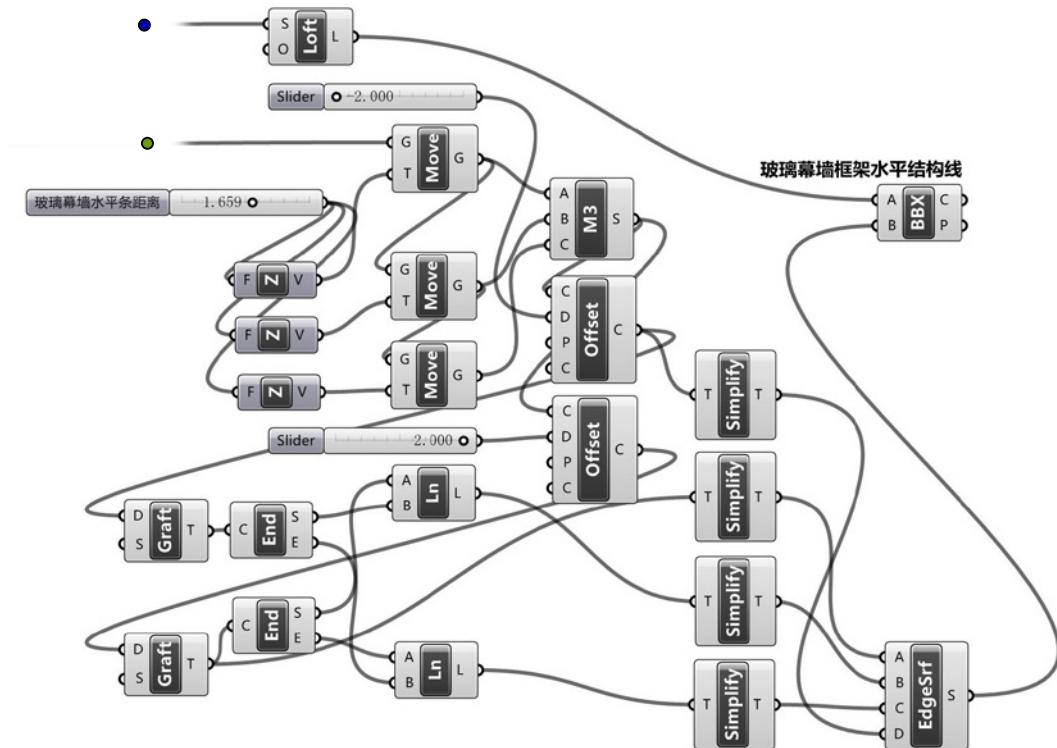
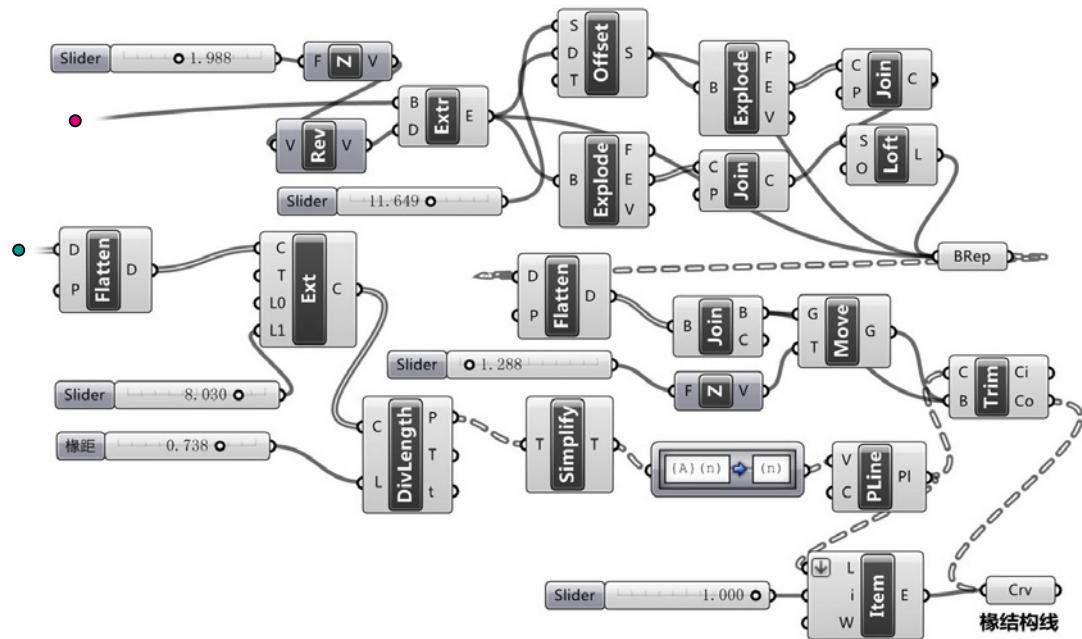
图中有部分椽并没有在GH中构建，当然GH有能力处理，但是可能会使程序变得有些复杂，在后期调整上也会变得不那么容易。因此，对于该部分的椽构件可以在RH中直接构建即可。

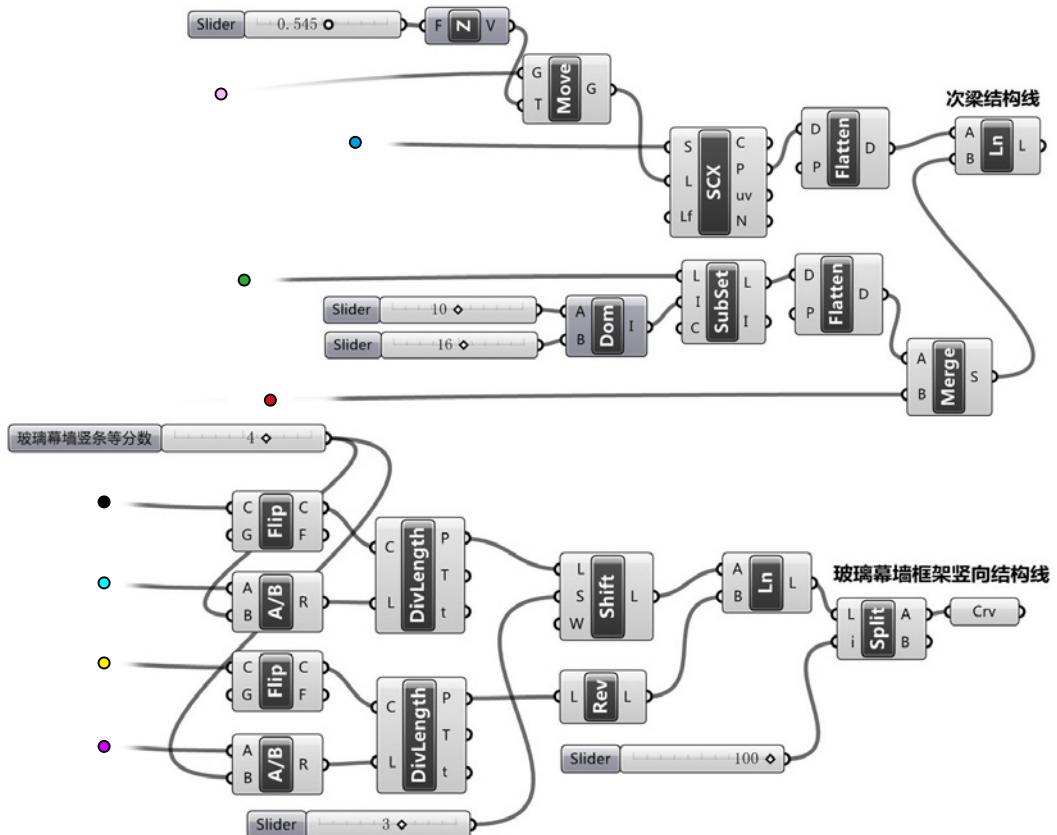




GH组件界面
GH Component Interface

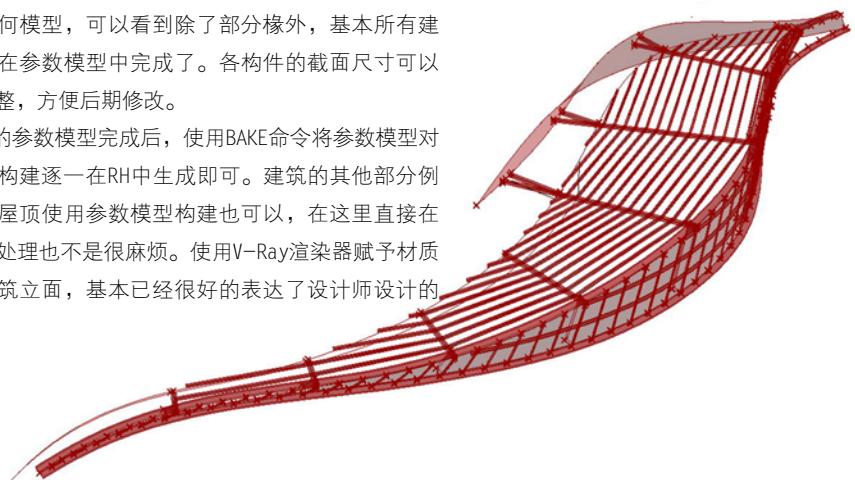






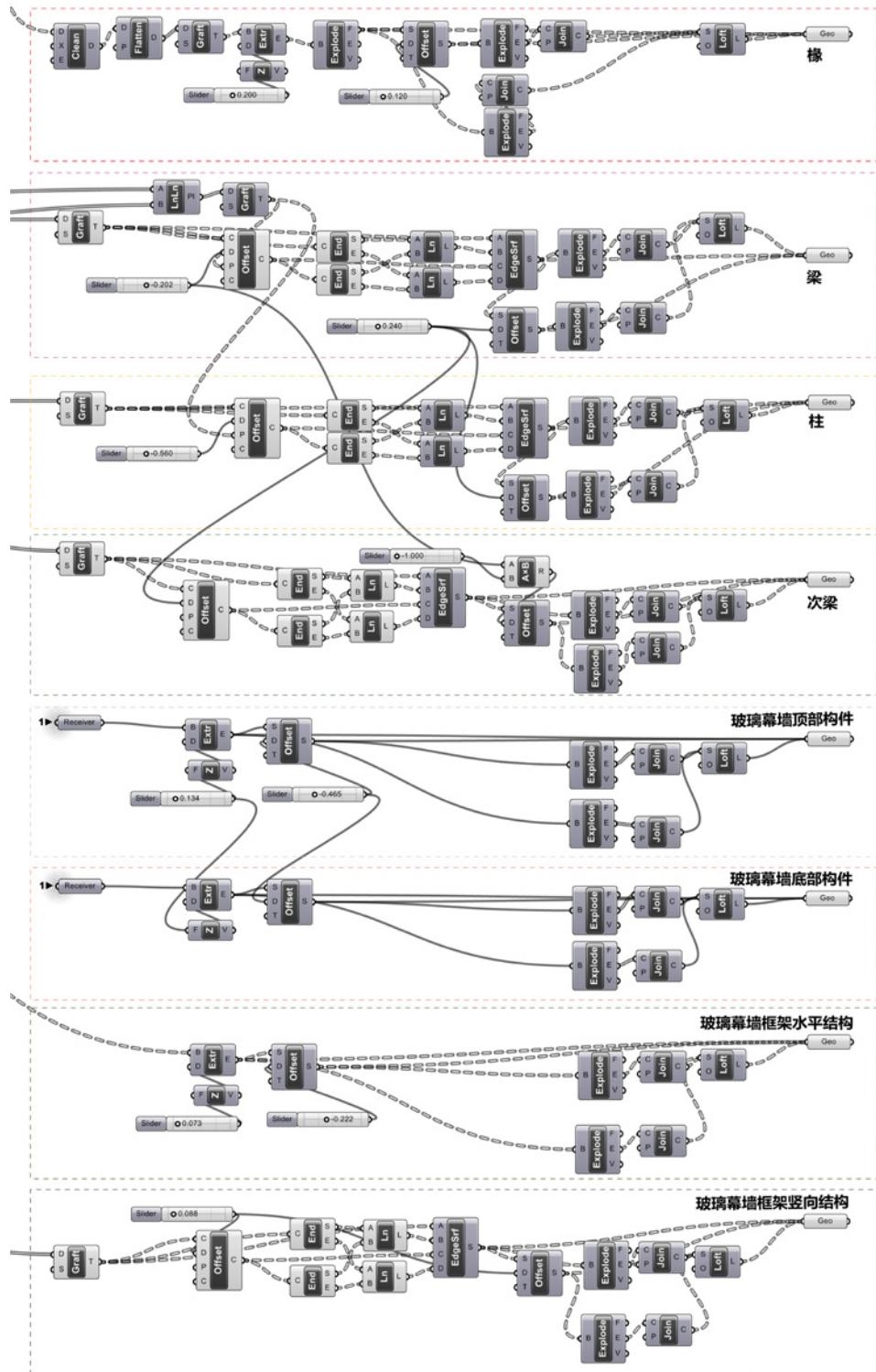
有了基本的结构线之后就可以构建几何体积，GH提供了很多构建面和体积的方法，基本都可以在Surface工具栏中找到。构建几何体参数的GH文件，参数规则的思路基本一致，例如构建玻璃幕墙的顶部与底部构件的方式是一样的，建立完顶部的规则后，直接拷贝连接底部的结构线即可生成底部幕墙构件。在最后完成的参数模型对应的几何模型，可以看到除了部分椽外，基本所有建筑构架都在参数模型中完成了。各构件的截面尺寸可以实时的调整，方便后期修改。

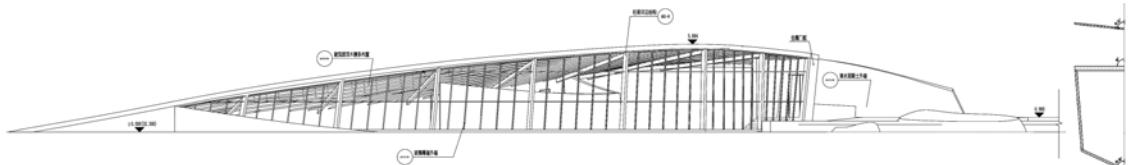
GH中的参数模型完成后，使用BAKE命令将参数模型对应的几何构建逐一在RH中生成即可。建筑的其他部分例如墙体，屋顶使用参数模型构建也可以，在这里直接在RH中直接处理也不是很麻烦。使用V-Ray渲染器赋予材质渲染的建筑立面，基本已经很好的表达了设计师设计的意图。



M-01_02

G-01_02



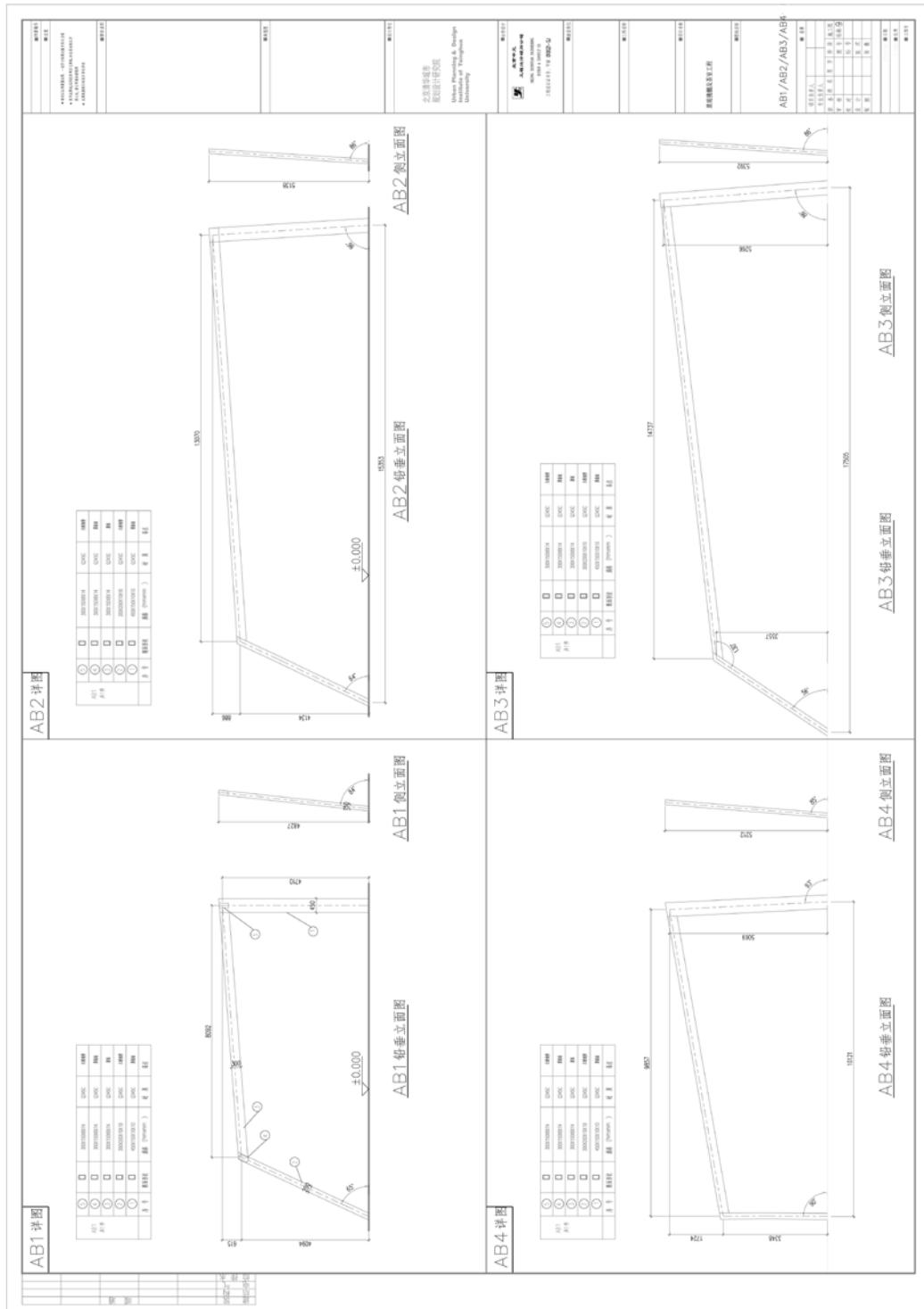


利用RH/Dimension/Make-2D Drawing工具导出的DWG格式立面文件，可以在AutoCAD加载，进一步标注，利于扩出阶段和施工图阶段的衔接。

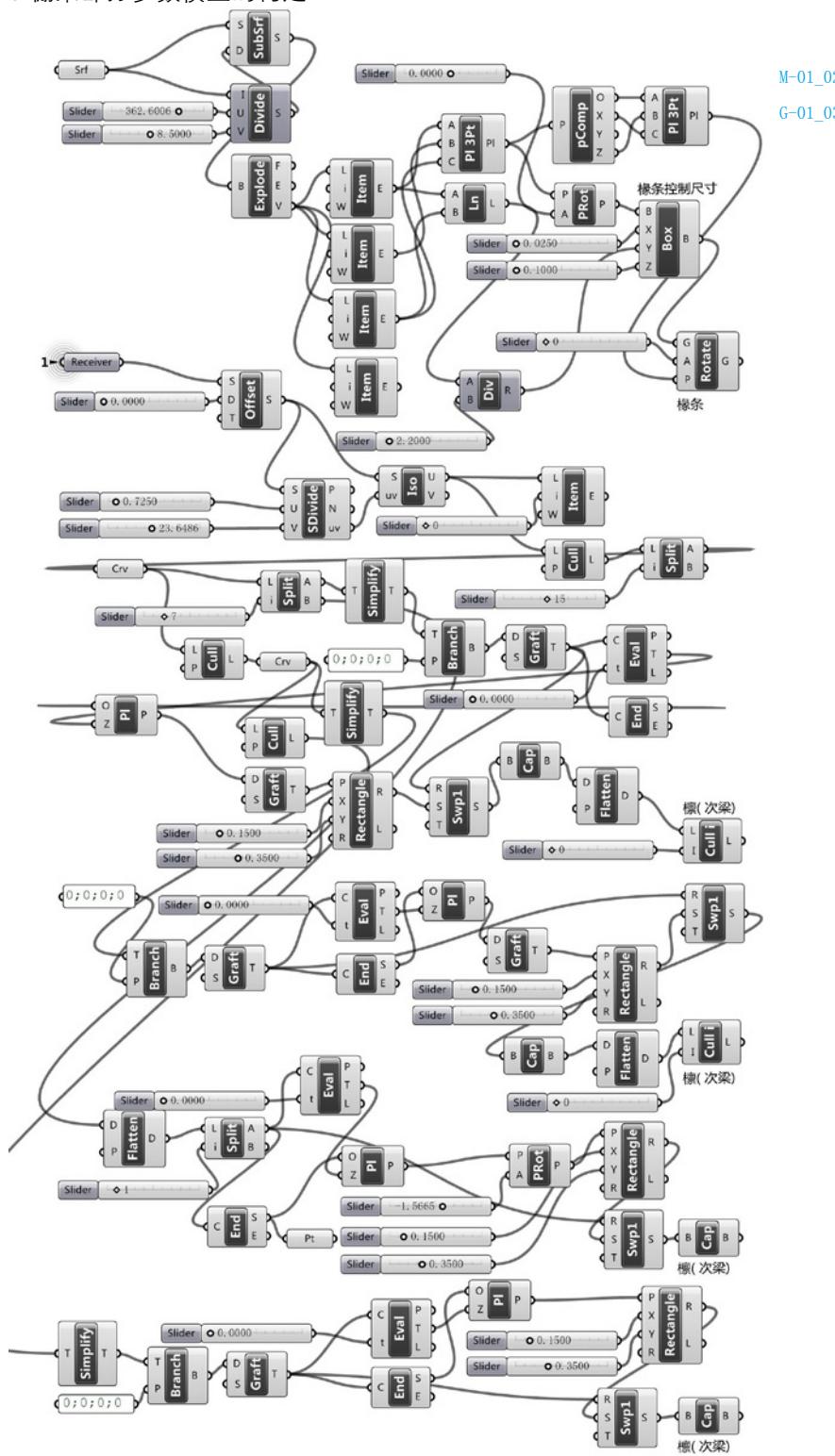


参数模型建立完之后，BAKE到RH中形成几何模型，图为使用V-Ray渲染器赋予材质渲染的建筑内部场景。在这里也提一下BIM (Building Information Modeling) 建筑信息模型技术，BIM集成了建筑工程项目各种相关信息的工程数据模型，为设计和施工提供相协调的信息，内部保持一致，并可以进行运算的信息，这些信息可以根据模型自动生成，并且与模型实时关联。BIM技术为设计师提供了从方案到施工的全套解决策略，节省了后期施工图制作的工作量，例如Revit软件，以尺寸驱动的参数化软件Digital Project (DP)，都是具有BIM技术的建筑三维建模工具，但是相对于RH+GH的组合，设计师更倾向于使用后者。RH+GH的方式在方案设计时使用的频率高于Revit和DP，很大的原因在于其操作的灵活性，虽然Revit可以处理工程数据信息，但是在方案构思建立模型时，更倾向于现代主义的设计形式，对于非线性设计部分就不是那么灵活了；DP是基于工业软件CATIA发展而来的，尺寸驱动方式的参数化设计非常强大，对于非线性的设计足以胜任，只是在易用性上，尤其自由的方案创作阶段，过于精细的模型建立方式，相对于GH来说，还是有些不顺手。

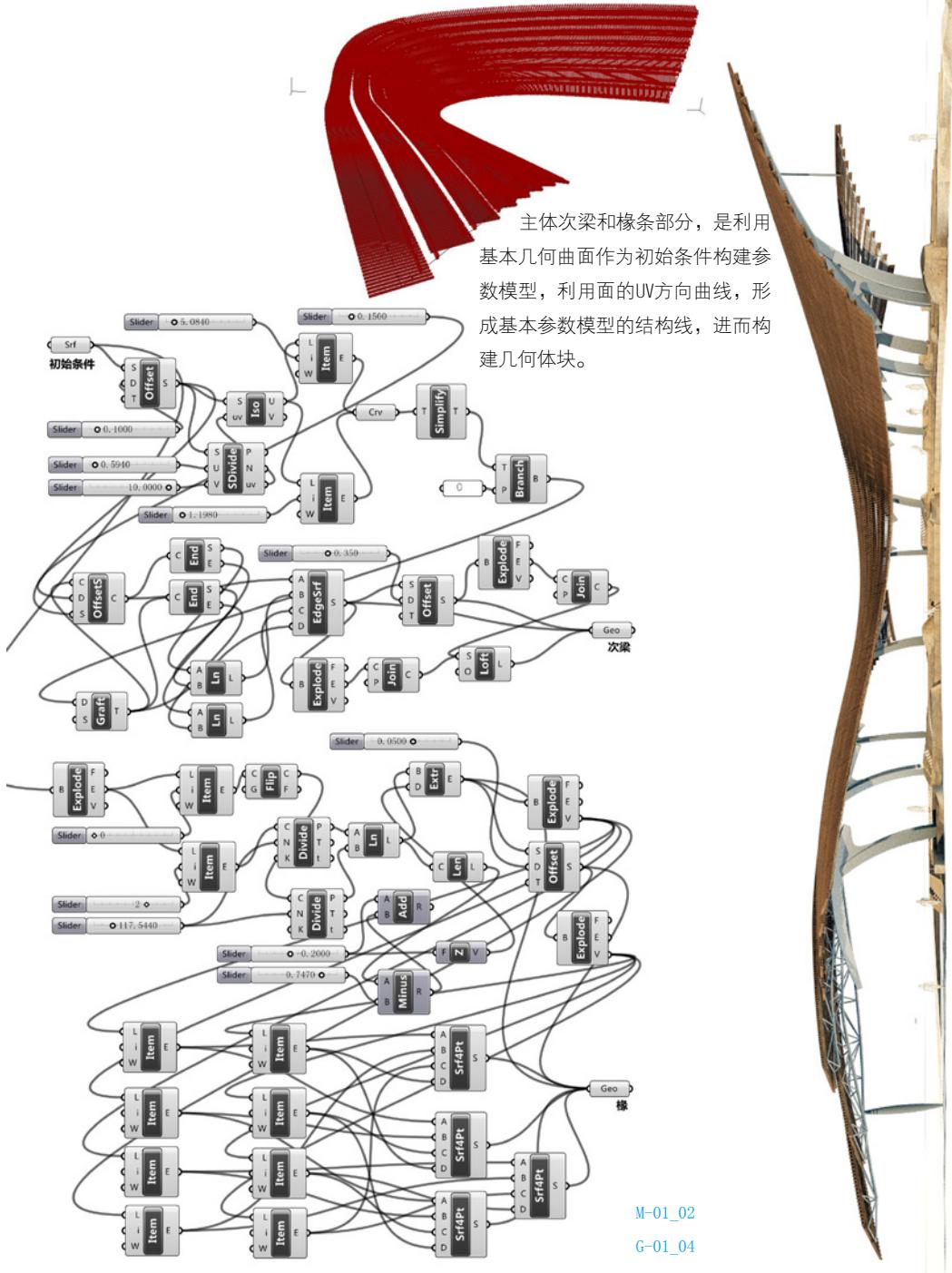
建筑梁架导出的DWG文件，给出了各个梁架的截面和角度，角度的标注是在RH中完成的。



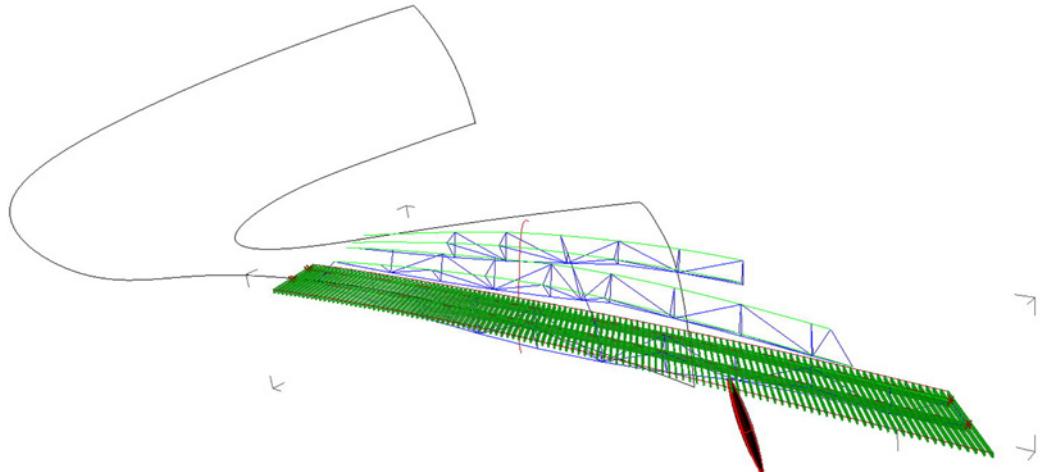
3 棚架部分参数模型的构建



棚架部分参数模型构建的时候，主要分为4个部分，顶部主体次梁和椽条部分，顶部棚架由桁架和棱柱支撑部分，吊棚部分，以及梁柱部分。为了使棚架更加延展、自由，棚架的一侧继续向外延展，并且像鸟儿的翅膀一样在端处分开。这样设计调整之后，如果将端处由桁架和棱柱支撑部分与主体次梁和椽条部分作为一个参数模型处理就会有些费力，所以单独分出来。

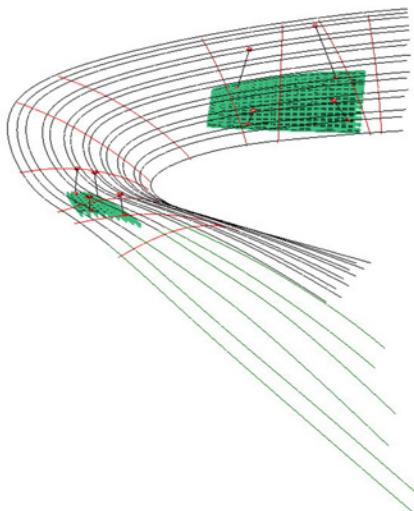


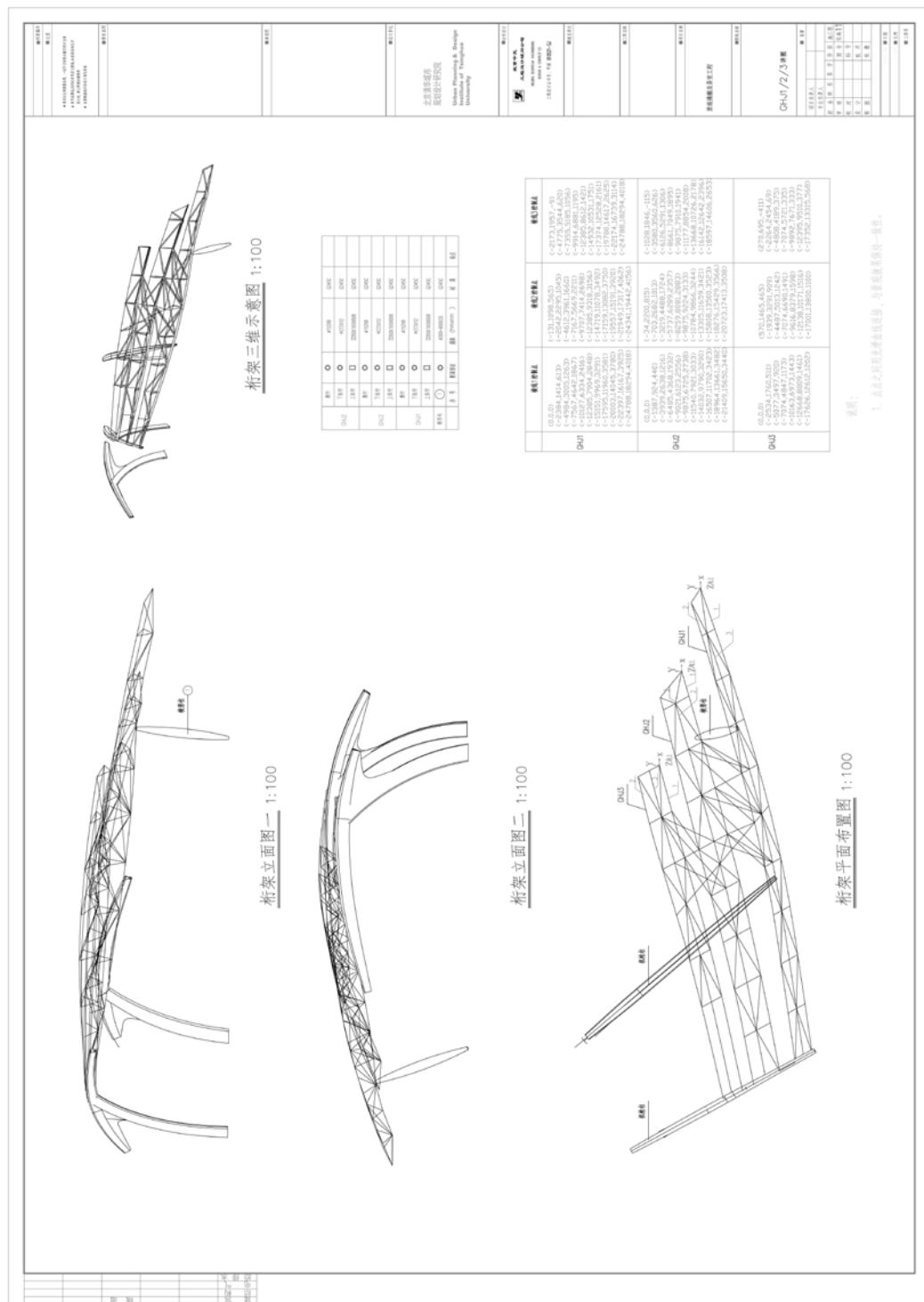
顶部棚架由桁架和梭柱支撑部分根部与主体部分衔接，先在RH中根据设计的目的，单独建立三个长条形的曲面作为GH参数模型的基本初始条件。三个曲面只要构建其中一个参数模型就可以了，而剩余两个直接使用这个参数模型。在最开始设计的时候，并没有考虑到由于端部近20m的延伸造成结构支撑上的问题，在与结构师配合的时候，他们为我们提供了桁架的支撑策略，如图所看到桁架结构线部分。在进一步设计修改时，直接使用结构师提供的桁架结构线在RH中建立几何模型。该桁架结构线部分也可以在GH中以参数模型的方式处理。



吊棚部分处理方式与上述参数模型类似，这里不再赘述。梁柱部分也是只需要建立一个模型就可以了，其余的梁柱重新输入下初始条件，再根据设计的要求适当调整参数。梁柱部分需要与棚架顶部次梁相接，梁上皮要与次梁吻合，红色线就是其主要的控制线之一。

梁柱的设计有意识的突破传统拘束的设计形式，大小不一，间距富于变化，希望能够创造一种自由的感觉。如《异规》(Informal [英]赛西尔·巴尔蒙得 Cecil Balmond) 中所说“现在计算机为我们打开了大门，它赋予我们前所未有的自由去探索———其结构是令人迷惑并改变思维，且万物皆可。而新造型如果是以通常的梁柱结构来支撑的话，它也就仅仅是一个立面。我们需要一种全新的方式，以便在建造自由造型时形成一种完整型，其构成的出发点是灵活的。以面代替线，散布代替等分支撑，运动轨迹代替固定中心，区域代替点”。我们在这里尝试新的方法，但是为了使梁柱的支撑更加的合理，梁柱本身并不发生倾斜，位于一个铅锤面上，只是围绕底座中心根据设计目的进行旋转，梁上皮则适宜顶部曲面的变化。

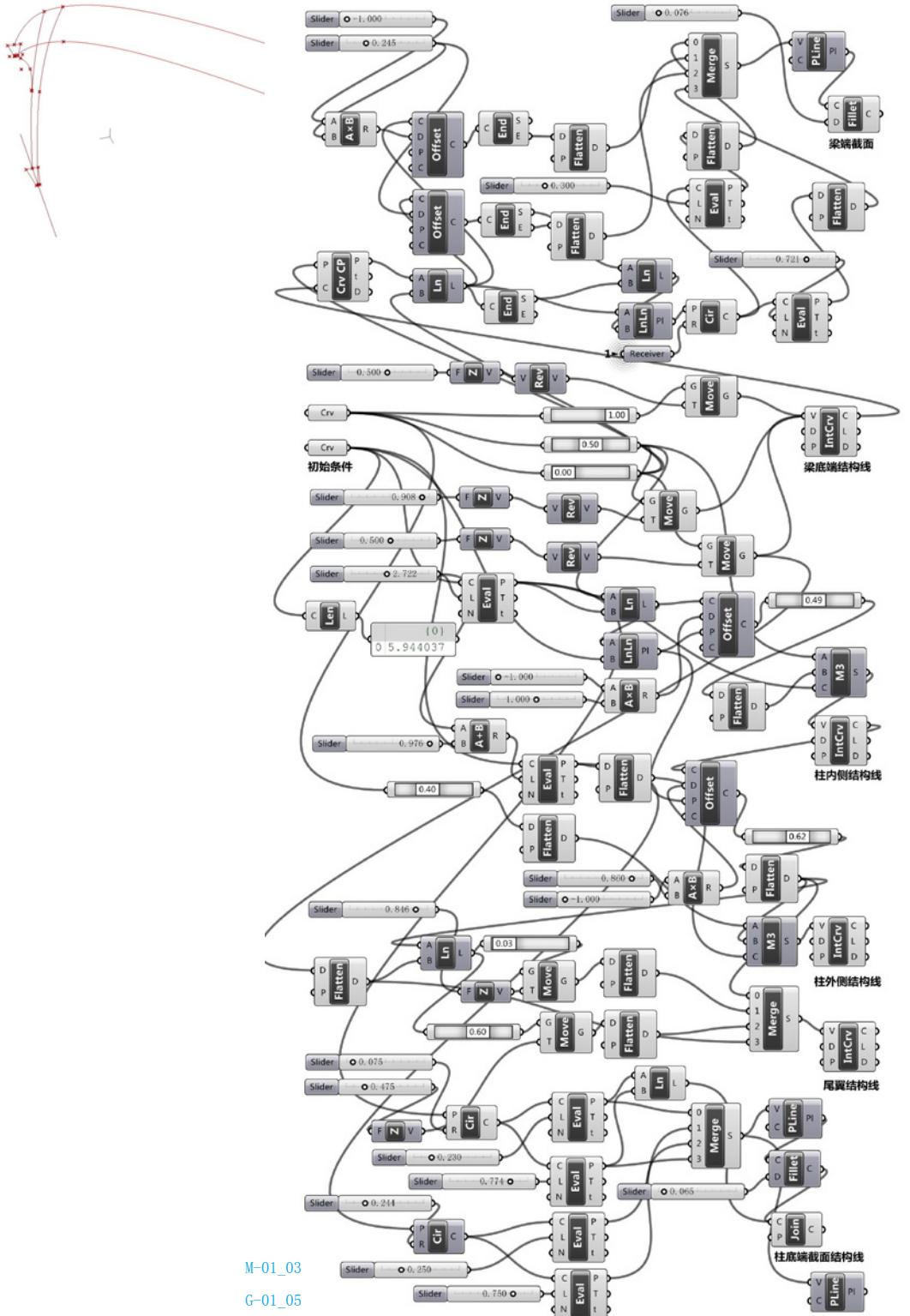


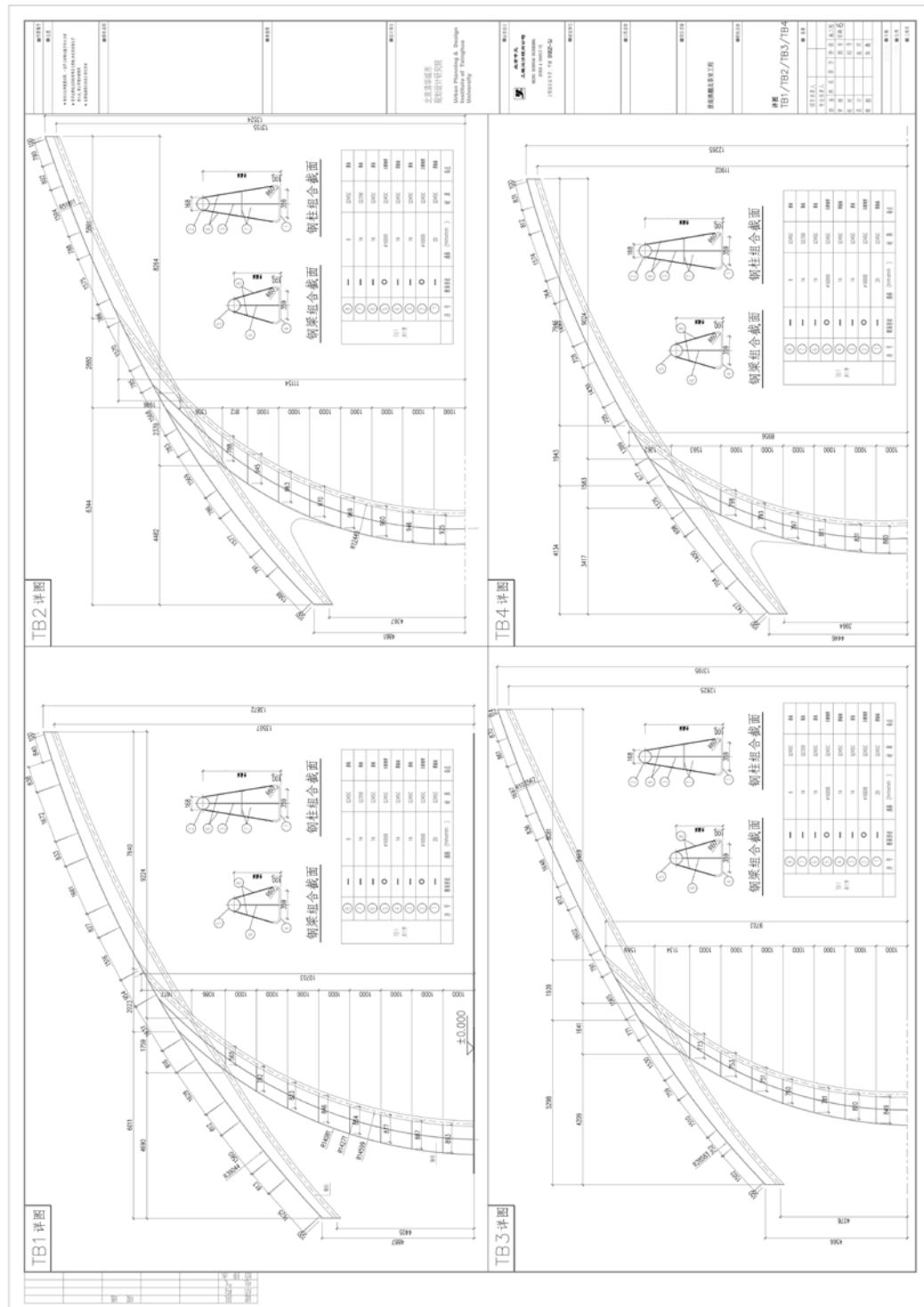


桁架立面图—1:100

桁架立面图二 1:100

桁架平面布置图 1:100

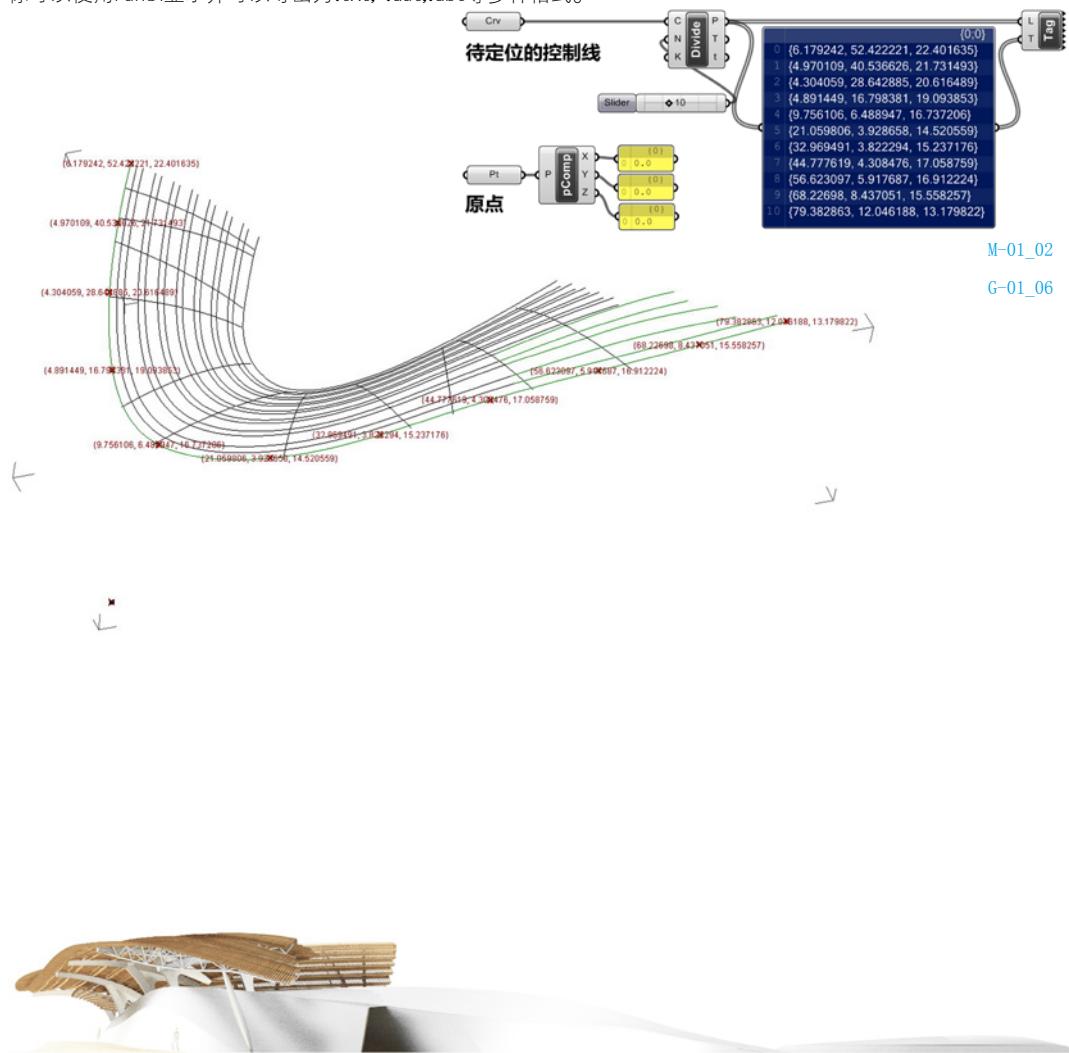


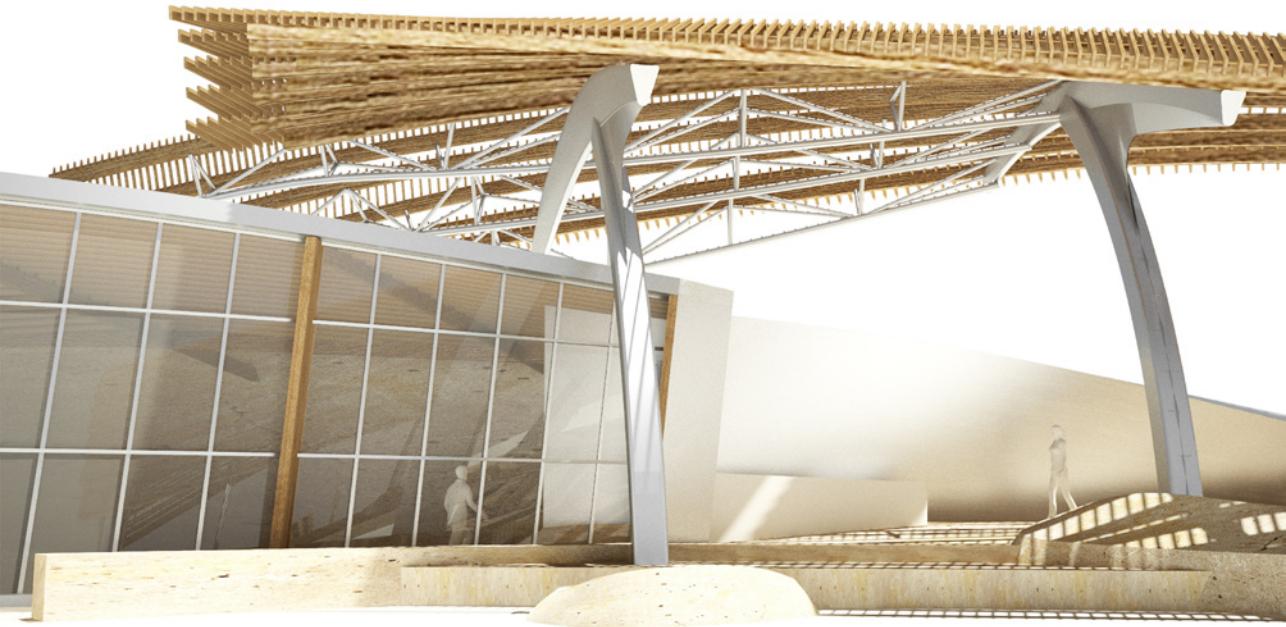


梁柱的参数模型依然是先建立结构部分的参数模型，GH参数模型的初始条件是梁上皮中心结构线和穿过底部截面的轴线。梁柱截面的宽高需要和结构师配合，结构师一般会给我们一个基本最小截面的值，其它变截面的变化不能小于该值。梁柱的体积参数模型部分，只要按照图示中的轨道，按截面放样即可。至此建筑和棚架部分的参数模型构建就基本完成了。在使用该种方法构建模型时，并没有将全部几何模型在GH中以参数的方式构建，而是根据设计的几何结构，结合参数模型构件规则的难易，分成多个部分相结合的方式。对于建筑结构部分是要和结构师配合的，而且在方案基本成型时，就需要结构师的基本意见，然后融入到设计中再进一步设计，以利于方案的顺利发展。

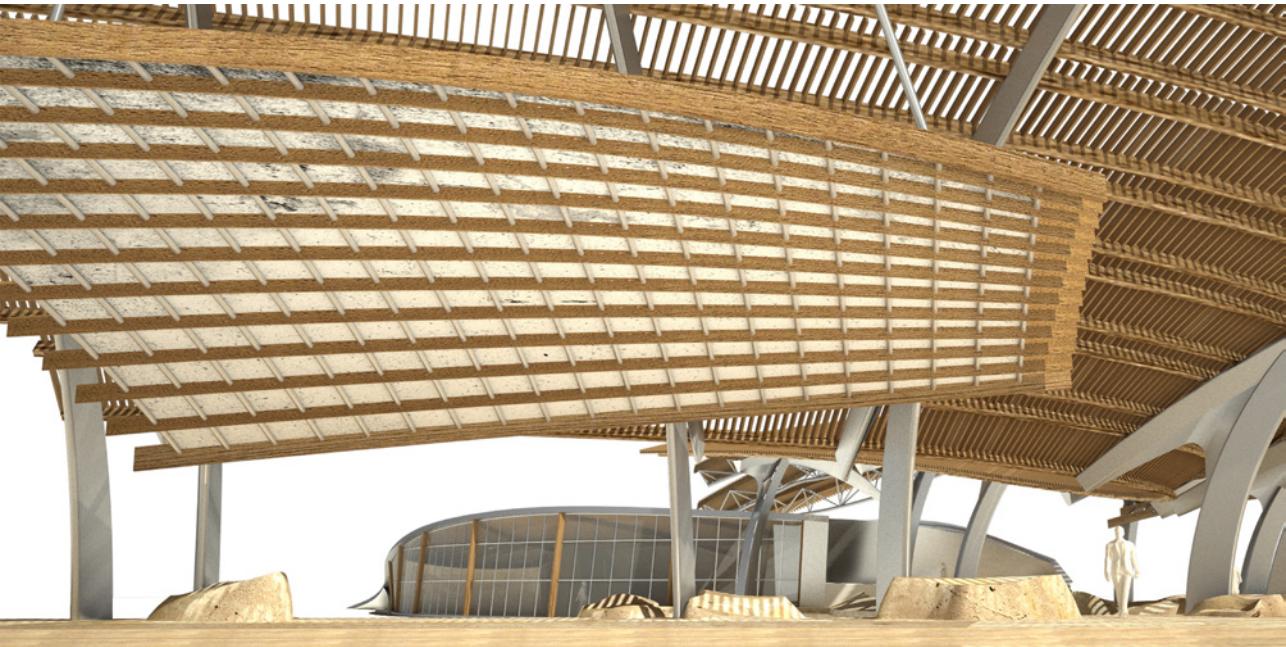
4 空间点的定位，二维DWG文件输出与V-Ray渲染表达

具有非线性特征的设计，在未来施工定位也与传统设计有所不同，对于方盒子式的建筑，完全可以仅通过长宽高的几何尺寸确定，但是本例中的建筑棚架仅以尺寸定位恐怕还不够，需要主要空间控制点的坐标辅助施工定位。我们以RH空间的原点(0, 0, 0)为参照点，在GH参数建模空间里根据施工需要划分主要控制线，各点的坐标可以使用Panel显示并可以导出为.txt, .dat, .dbt等多种格式。





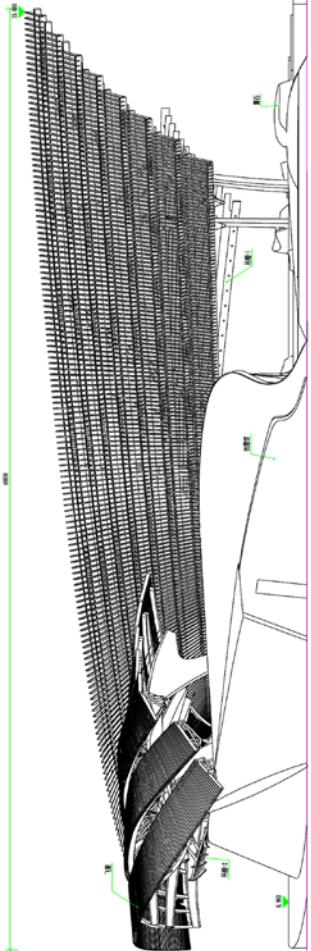
在处理方案的时候，是要实时的赋予材质进行观察以获得真实的尺度感，RH与V-Ray的结合可以帮助在表达上获得这个理想的效果，同时，可以作为最后的表达与业主的交流文件，能够节省后期表达处理的时间。在使用V-Ray渲染几何模型时，植物部分不是它的擅长，因此对于景观设计师而言，不为一个遗憾。一般增加了植物种植的表达需要导出几何模型在LUMION或者VUE里处理。

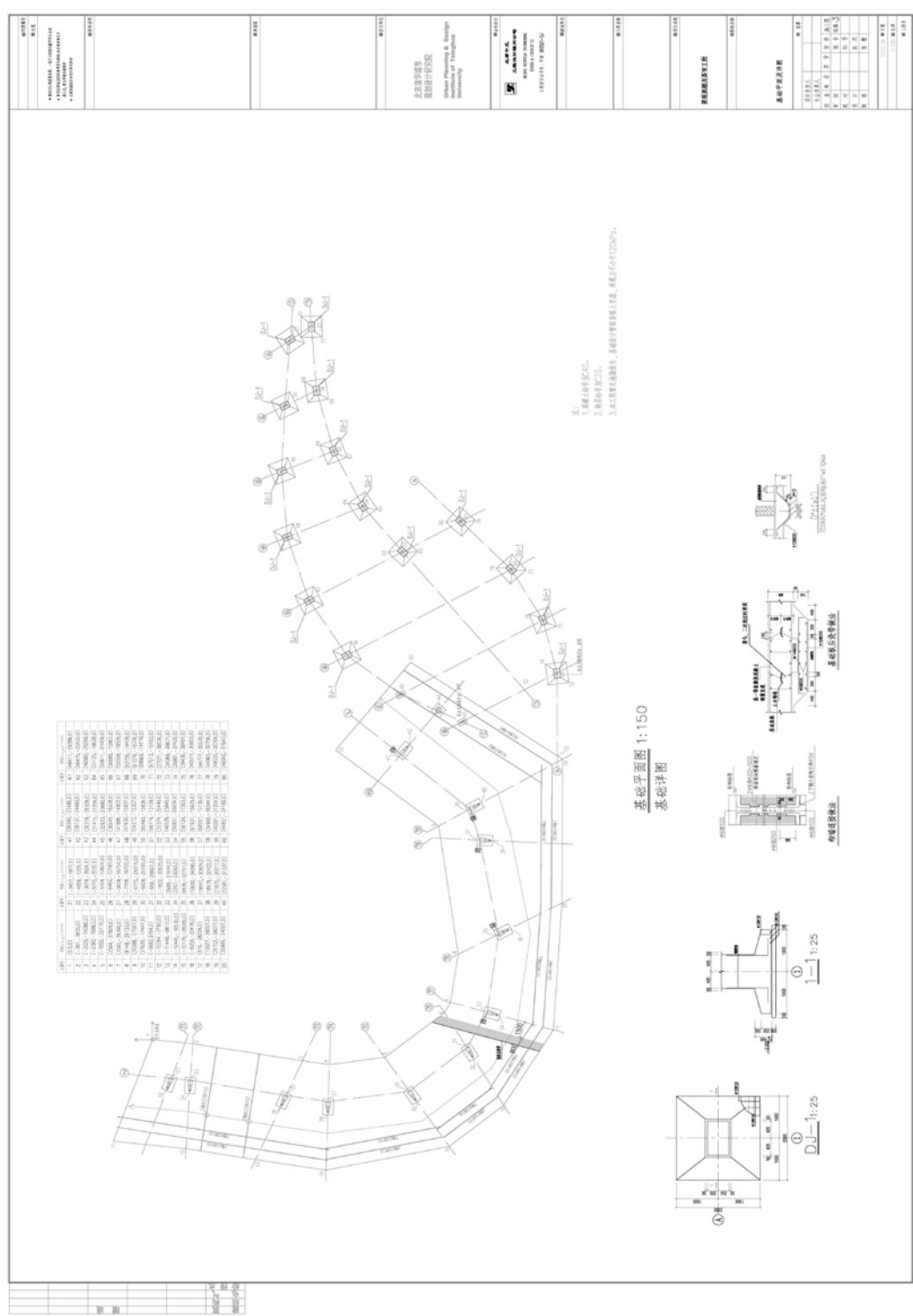


将RH中的几何模型
转为二维DWG文件可以使
用Make-2D Drawing工
具根据所需视图导出即
可。

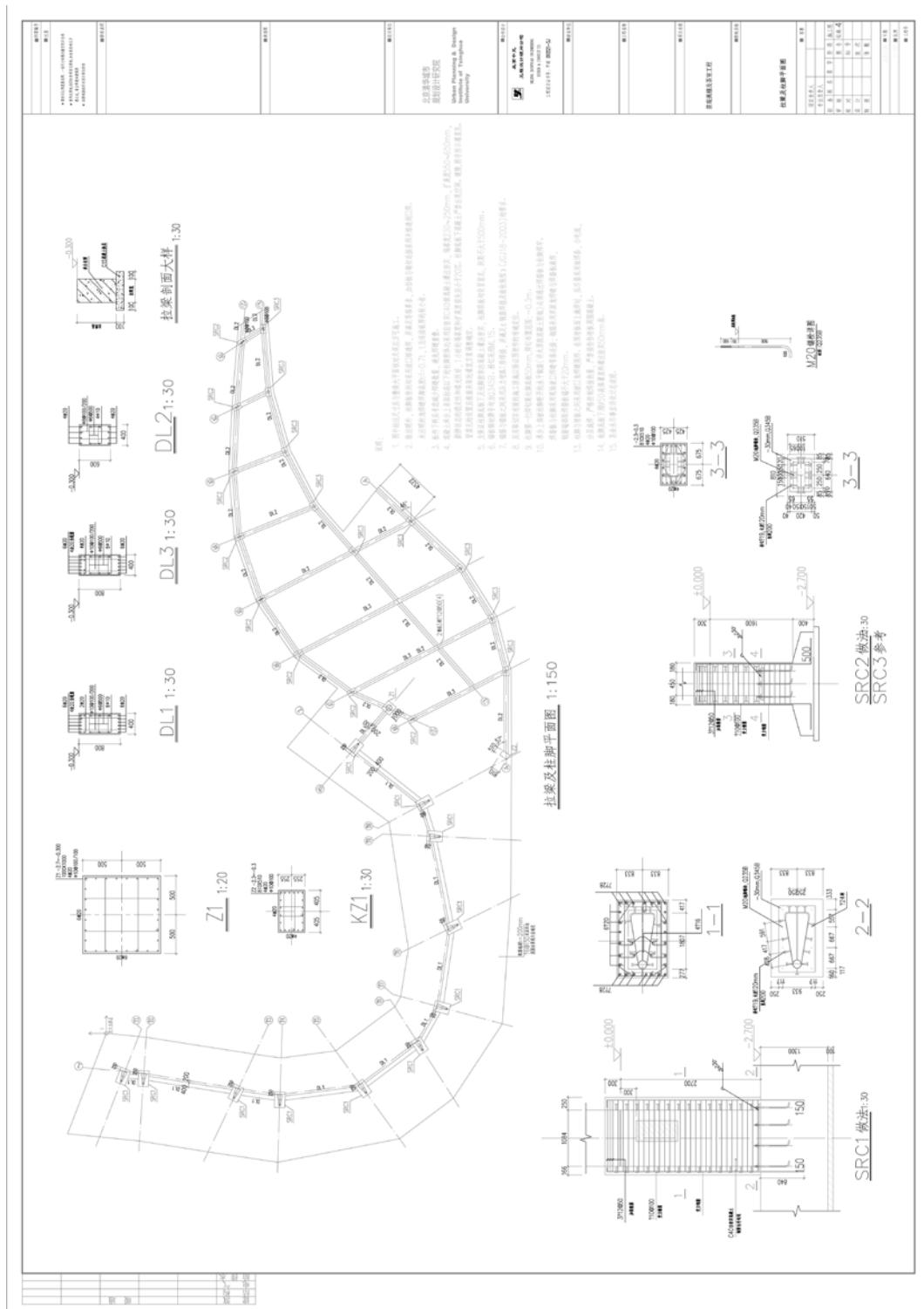
使用GH参数模型的方式协助设计师进行方案设计，与SU模型推敲方式相比，具有自身明显的优势。从方案的设计到扩出施工图的制图都有很好的表现。现在我们看到的很多非线性参数设计，基本都是基于GH参数模型处理的。本例中参数模型规则的构建基本是与设计结构搭建顺序一致，因此，对于传统的一些木结构等具有明显搭建顺序的设计，都可以使用参数模型协助设计，而并不是只有曲线形式的设计才使用参数。在构建规则时受到GH所提供的工具的限制，有些设计即使使用GH似乎也无法处理，所以GH提供了C#和VB语言以供设计师使用，但是对于以设计为主的设计师而言，这两种语言学起来还是不那么容易，现在GH的版本中能够嵌入Python语言。Python语言优美，设计师在短时间内就可以掌握，并进入实践；同时，MAYA自8.5之后支持Python语言。Rhinoceros自5.0之后嵌入Ironpython，Houdini使用的是HScript，自9.0后使用HOM（Houdini Object Mode），支持Python语言。地理信息软件，ArcGis8基于地理视图的脚本语言开始引入，9.0开始支持Python。VUE自然景观生成软件同样支持Python语言。Python程序语言逐渐的被更为广泛的三维分析设计软件所支持。所以以后，设计师应该不会被使用一种软件就得学习一门新的语言所困惑吧。

参数模型的构建现在已经体现出了明显的优势，正在形成这个世纪初设计界发展最为显著的特点，在以后很长一段时间里从规划师，建筑师到景观师，艺术家仍然会在参数模型技术方面与程序设计人员一起进一步发展参数的潜力，协助设计师更好的设计创作。





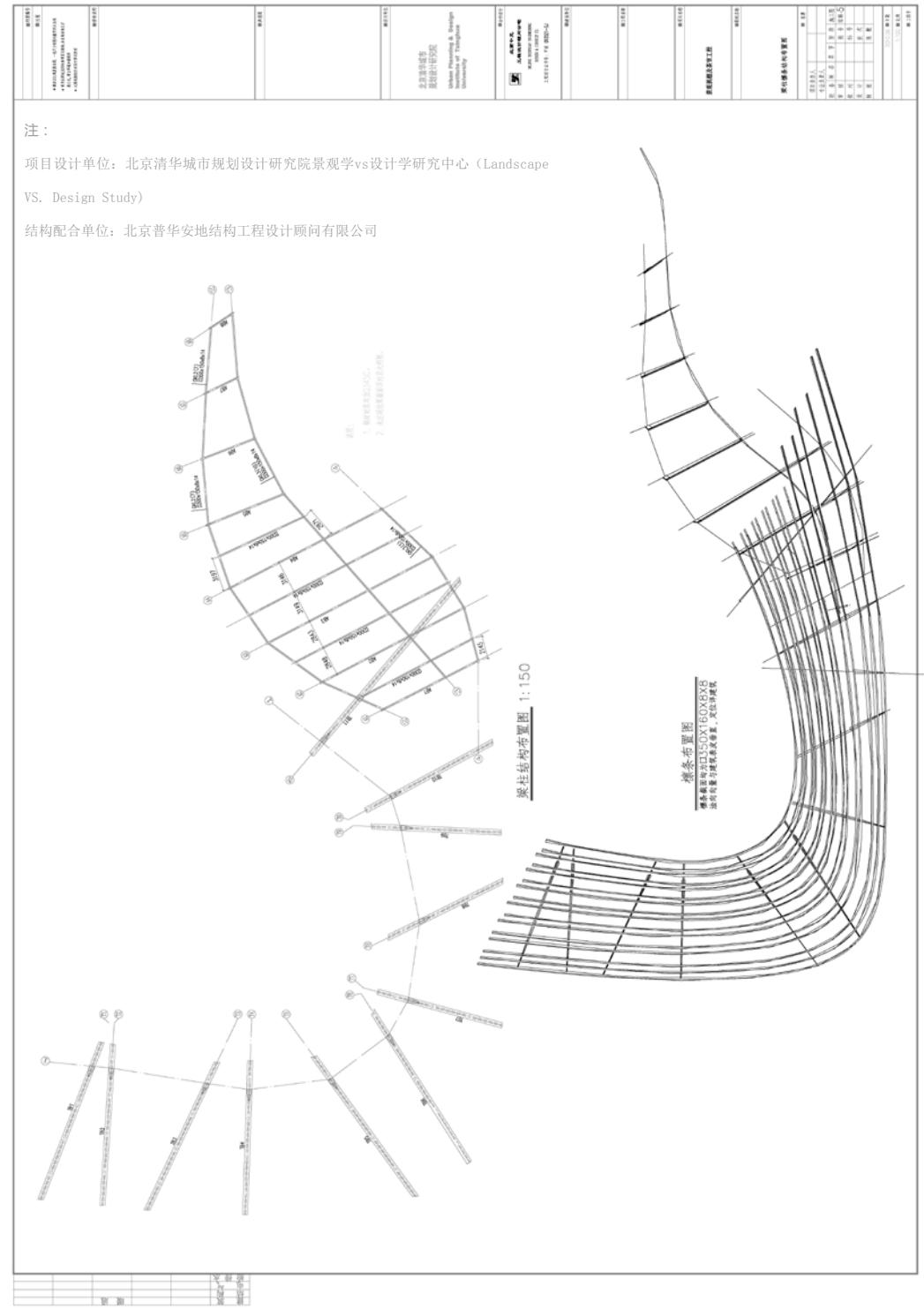
基础平面图 1:150



注：

项目设计单位：北京清华城市规划设计研究院景观学vs设计学研究中心（Landscape
VS. Design Study）

结构配合单位：北京普华安地结构工程设计顾问有限公司



第二部分

Python与Grasshopper的联合

[http://www.python.org./](http://www.python.org/)

Python官方网站



以程序语言为基础的参数驱动技术辅助设计逐渐的成为三维模型构建的主流，是未来发展的重要方向。现在在计算机语言发展已经构建出一个强大的阵营，包括C,C++,C#,Java,VB,Python,PHP,Perl等等。设计师似乎难以想象自己何时竟与程序语言构建了联系。而这种联系竟然是未来发展的一个重要方向。在规划、建筑、景观设计师的培训机制中，国内尚且很少有将其与设计挂钩的相关课程。如果作为设计师培训课程的一部分，设计行业的发展必然会出现另一种局面。

在众多的程序语言中，让设计师极为棘手的莫过于选择哪种语言。被认为只有计算机专业才应该掌握的程序语言，对于设计师来讲系统的学习要耗费非常大的精力。实际上也并不推荐学习难度大的C语言等耗费精力的语言。对于设计师，扎实的设计基本功和空间感悟、美学修养是其永不变的基础。语言是在这个基础之上，放大设计师可以触及到的形态领域。目前三维分析设计软件基本都有自己的脚本语言。MAYA是MEL，自8.5之后支持Python语言。Rhinoceros是RhinoScript，自5.0之后嵌入Ironpython,Houdini使用的是HScript,自9.0后使用HOM (Houdini Object Mode)，支持Python语言。地理信息软件，ArcGis8基于地理视图的脚本语言开始引入，9.0开始支持Python。VUE自然景观生成软件与FME地理数据转化平台同样支持Python语言。Python程序语言在逐渐的被更为广泛的三维分析设计软件所支持，这正是在于Python语言的优美，同时可以让设计师在短时间内即可掌握，并进入实践的语言。

在程序语言的辅助设计下，设计内容也日趋复杂，仅依靠传统纸笔的设计方式难以达到目的。而在三维模型构建程序中，仅仅依靠传统建模方式，会带来繁重的工作量。尤其在设计不断整合，来回修改的过程中，依靠程序语言（编码）的方式，可以大幅度减轻工作难度，将重点放置于设计推敲上，并进入和探索新的形态领域。

在未接触程序语言时，设计师看待语言的态度是神秘的。由无数代码产生的图形总比直观的形态推敲要抽象的多。实际上一旦读懂语言，所关注的重点自然是这种构建逻辑的合理性，不会被抽象的语言所迷惑。当然，程序越简单，可读性越强是最好的。不仅使模块易于操作修改，同时使其他设计师或程序员也更加易读。Python语言与GH的结合，极大的拓展了GH的模型构建能力。

使用Python语言协助设计，相对于其他编程语言，例如Rhino另一个支持的语言VB，和C等语言来说，python语言能够给予设计师更流畅的表达，“…但最重要的是，Python是一种使你在编程时能够保持自己风格的程序设计

语言。你不用费什么劲就可以实现你想要的功能，并且编写的程序清晰易懂（和当前流行的其他各种程序设计语言相比更是如此）。”—*Beginning Python from Novice to Professional*由Python语言编写的程序可以感觉到类似于一篇优雅的散文。在使用Python协助设计的过程中，设计师不用过多的关注类似其他语言那样严格的控制结构，例如给予VB语言的Rhinoscript，必须声明变量，语句块结构结束需要有End If, End Sub关键字标示结束。更重要的是Python语言与一般英语结构相似，例如：

```
name=input('What is your name?')  
if name.endswith('Gumby'):  
    print ('Hello, Mr. Gumby')  
else:  
    print('Hello, Stranger')
```

注：学习Python建议首先阅读（[挪]Mangus Lie Hetland. *Begining Python from novice to professional second edition python*基础教程，司维，曾军威，谭颖华译.北京，人民邮电出版社，2010.）在实际编程过程中需要（David M. Beazley. *Python Essential Reference fourth edition python* 参考手册.谢俊，杨越，高伟译.2011.北京，人民邮电出版社.）一书为辅助，可以帮助查询语法和命令。

针对RHINO平台的Python可以从官方网址下载教程（Skylar Tibbits, Arhur van der Harten, Steve Baer. 2011. *Python for rhinoceros 5.*），实际操作过程中需要参考RHINO提供的Rhino IronPython帮助文件；

1 Python Script

Python的IDE (Python GUI)交互式解释可以在<http://www.python.org>中下载。在解释器中例举了《Beginning Python From Novice to Professional》一书中关于鸟类定义的一个函数，使用Python解释器能够随时的运行函数和检查语句，这非常有助于非专业程序员的设计师进行程序的编写。虽然Rhino本身带有Ironpython解释器，但是建议设计师也要安装Python GUI交互式解释器，在这里可以查看和学习诸多的Python函数、方法、类以及模块。

P-02_01

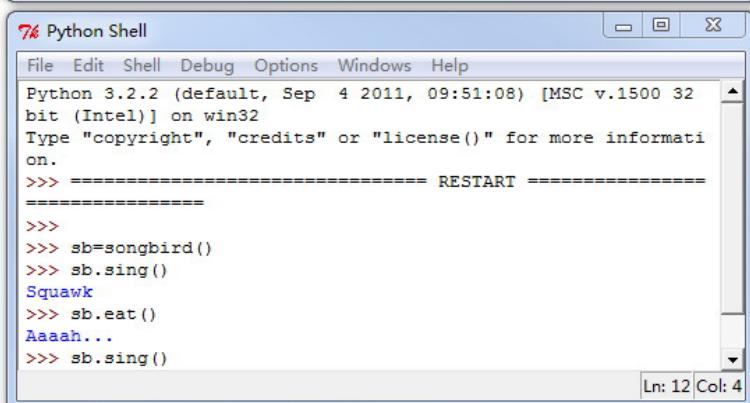


```

74 bird.py - E:/01-DissERTATIONPhase03/PythonCode/bird.py
File Edit Format Run Options Windows Help
#bird类定义所有鸟类具有一些基本能力，本例中为进食。
class bird:
    def __init__(self):
        self.hungry=True
    def eat(self):
        if self.hungry:
            print ('Aaaaah...')
            self.hungry=False
        else:
            print('No, thanks!')
    #bird类定义所有鸟类具有一些基本能力，本例中为进食。
class songbird(bird):
    def __init__(self):
        #使用super函数，调用songbird超类构造方法，确保进行基本的初始化。
        super(songbird,self).__init__()
        self.sound='Squawk'
    def sing(self):
        print(self.sound)

```

Ln: 7 Col: 30

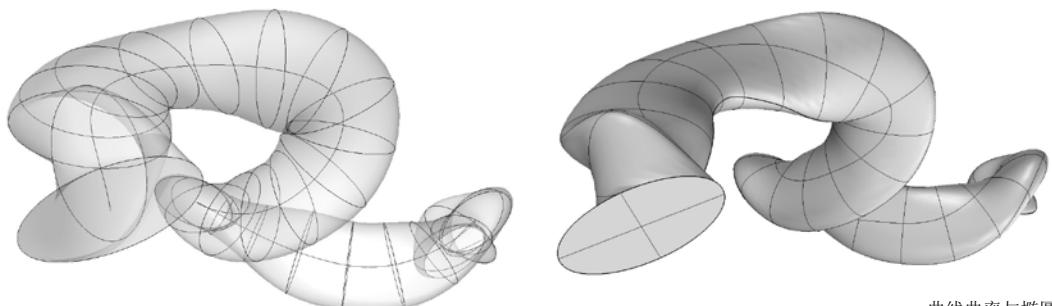


```

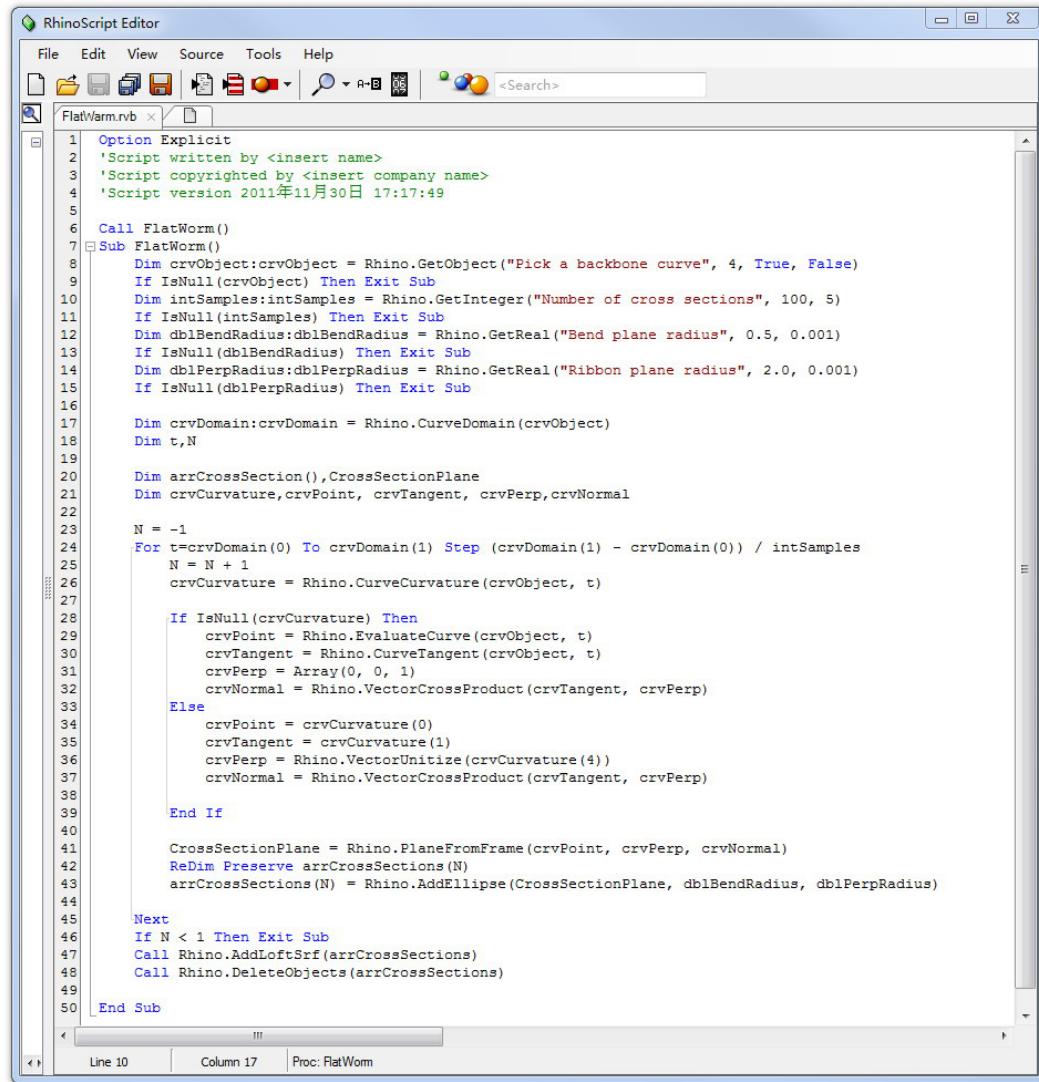
74 Python Shell
File Edit Shell Debug Options Windows Help
Python 3.2.2 (default, Sep 4 2011, 09:51:08) [MSC v.1500 32
bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> ===== RESTART =====
=====
>>>
>>> sb=songbird()
>>> sb.sing()
Squawk
>>> sb.eat()
Aaaaah...
>>> sb.sing()

```

Ln: 12 Col: 4



曲线曲率与椭圆



The screenshot shows the RhinoScript Editor window with the file "FlatWorm.rvb" open. The code is a VBA script for creating a flat worm shape from a backbone curve. It includes error handling, variable declarations, and a main loop for generating cross-sections.

```

1 Option Explicit
2 'Script written by <insert name>
3 'Script copyrighted by <insert company name>
4 'Script version 2011年11月30日 17:17:49
5
6 Call FlatWorm()
7 Sub FlatWorm()
8     Dim crvObject:crvObject = Rhino.GetObject("Pick a backbone curve", 4, True, False)
9     If IsNull(crvObject) Then Exit Sub
10    Dim intSamples:intSamples = Rhino.GetInteger("Number of cross sections", 100, 5)
11    If IsNull(intSamples) Then Exit Sub
12    Dim dblBendRadius:dblBendRadius = Rhino.GetReal("Bend plane radius", 0.5, 0.001)
13    If IsNull(dblBendRadius) Then Exit Sub
14    Dim dblPerpRadius:dblPerpRadius = Rhino.GetReal("Ribbon plane radius", 2.0, 0.001)
15    If IsNull(dblPerpRadius) Then Exit Sub
16
17    Dim crvDomain:crvDomain = Rhino.CurveDomain(crvObject)
18    Dim t,N
19
20    Dim arrCrossSection():CrossSectionPlane
21    Dim crvCurvature,crvPoint, crvTangent, crvPerp,crvNormal
22
23    N = -1
24    For t=crvDomain(0) To crvDomain(1) Step (crvDomain(1) - crvDomain(0)) / intSamples
25        N = N + 1
26        crvCurvature = Rhino.CurveCurvature(crvObject, t)
27
28        If IsNull(crvCurvature) Then
29            crvPoint = Rhino.EvaluateCurve(crvObject, t)
30            crvTangent = Rhino.CurveTangent(crvObject, t)
31            crvPerp = Array(0, 0, 1)
32            crvNormal = Rhino.VectorCrossProduct(crvTangent, crvPerp)
33        Else
34            crvPoint = crvCurvature(0)
35            crvTangent = crvCurvature(1)
36            crvPerp = Rhino.VectorUnitize(crvCurvature(4))
37            crvNormal = Rhino.VectorCrossProduct(crvTangent, crvPerp)
38        End If
39
40        CrossSectionPlane = Rhino.PlaneFromFrame(crvPoint, crvPerp, crvNormal)
41        ReDim Preserve arrCrossSections(N)
42        arrCrossSections(N) = Rhino.AddEllipse(CrossSectionPlane, dblBendRadius, dblPerpRadius)
43
44    Next
45    If N < 1 Then Exit Sub
46    Call Rhino.AddLoftSrf(arrCrossSections)
47    Call Rhino.DeleteObjects(arrCrossSections)
48
49
50 End Sub

```

V-U2_U1

```

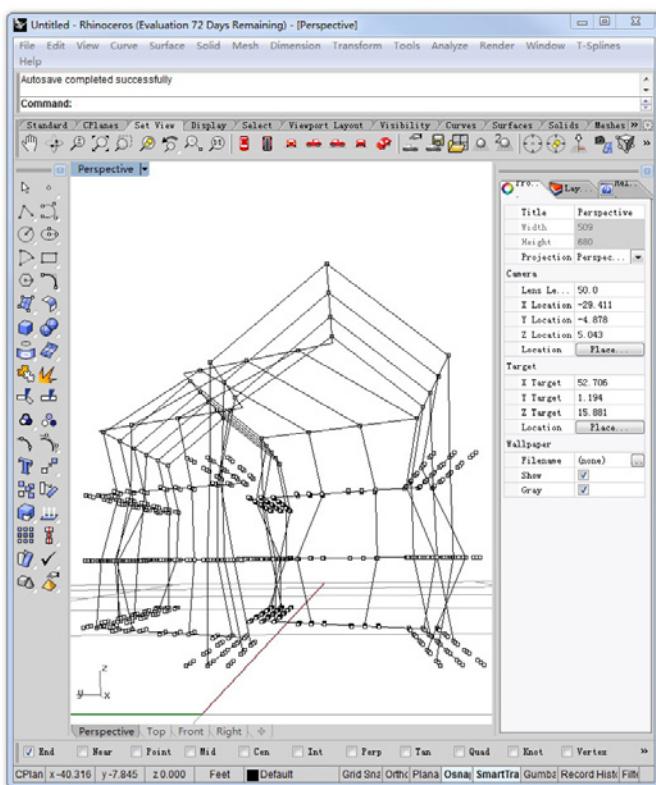
1  def FlatWorm():
2      import rhinoscriptsyntax as rs
3
4      crvObject=rs.GetObject('Select one curve:',4,True,True)
5      if not crvObject:return
6      ellipseradius1=rs.GetReal('input ellipse long radius:',2)
7      if not ellipseradius1:return
8      ellipseradius2=rs.GetReal('input ellipse short radius:',1)
9      if not ellipseradius2:return
10     sample=rs.GetInteger('Input interger:',30)
11     if not sample:return
12
13     domain=rs.CurveDomain(crvObject)
14     step=(domain[1]-domain[0])/sample
15
16     ellipses=[]
17     t=domain[0]
18     for t in rs.Frange(domain[0],domain[1],step):
19         ccurvature=rs.CurveCurvature(crvObject,t)
20         if not ccurvature:
21             curpoint=rs.EvaluateCurve(crvObject,t)
22             curtangent=rs.CurveTangent(crvObject,t)
23             curperp=(0,0,1)
24             curnormal=rs.VectorCrossProduct(curtangent,curperp)
25             crosssectionplane=rs.PlaneFromFrame(curpoint,curperp,curnormal)
26         else:
27             curpoint=ccurvature[0]
28             curtangent=ccurvature[1]
29             curperp=rs.VectorUnitize(ccurvature[4])
30             curnormal=rs.VectorCrossProduct(curtangent,curperp)
31             crosssectionplane=rs.PlaneFromFrame(curpoint,curperp,curnormal)
32
33             if crosssectionplane:
34                 elli=rs.AddEllipse(crosssectionplane,ellipseradius1,ellipseradius2)
35                 ellipses.append(elli)
36
37
38     if not ellipses: return
39     rs.AddLoftSrf(ellipses)
40     rs.DeleteObjects(ellipses)
41
42 FlatWorm()

```

P-02_02

对于rhino内置的python解释器可以通过在命令行中敲入Editpythonscript命令打开,或者Rhino/Tools/Python Script打开。在此例举了<Python for Rhinoceros 5.0> 和<RhinoScript for Rhinoceros 4.0>中关于曲线曲率,以及椭圆与基础参考面方向关系的程序。图的左侧为基于VB语言的Rhino Script,右侧为Python Script。并列在一起做比较,可以明显的感受到Python语言的流畅,这样设计师可以把精力主要集中在几何的设计上,而不是语言本身。

在这里用Python编写建筑的主要架构结构线，开始的概念来自于<Cecil Balmond>的S-Project的‘乌拉姆玫瑰’。(a+u.Cecil Balmond赛西尔·巴尔蒙得.北京：中国电力出版社，2008.)



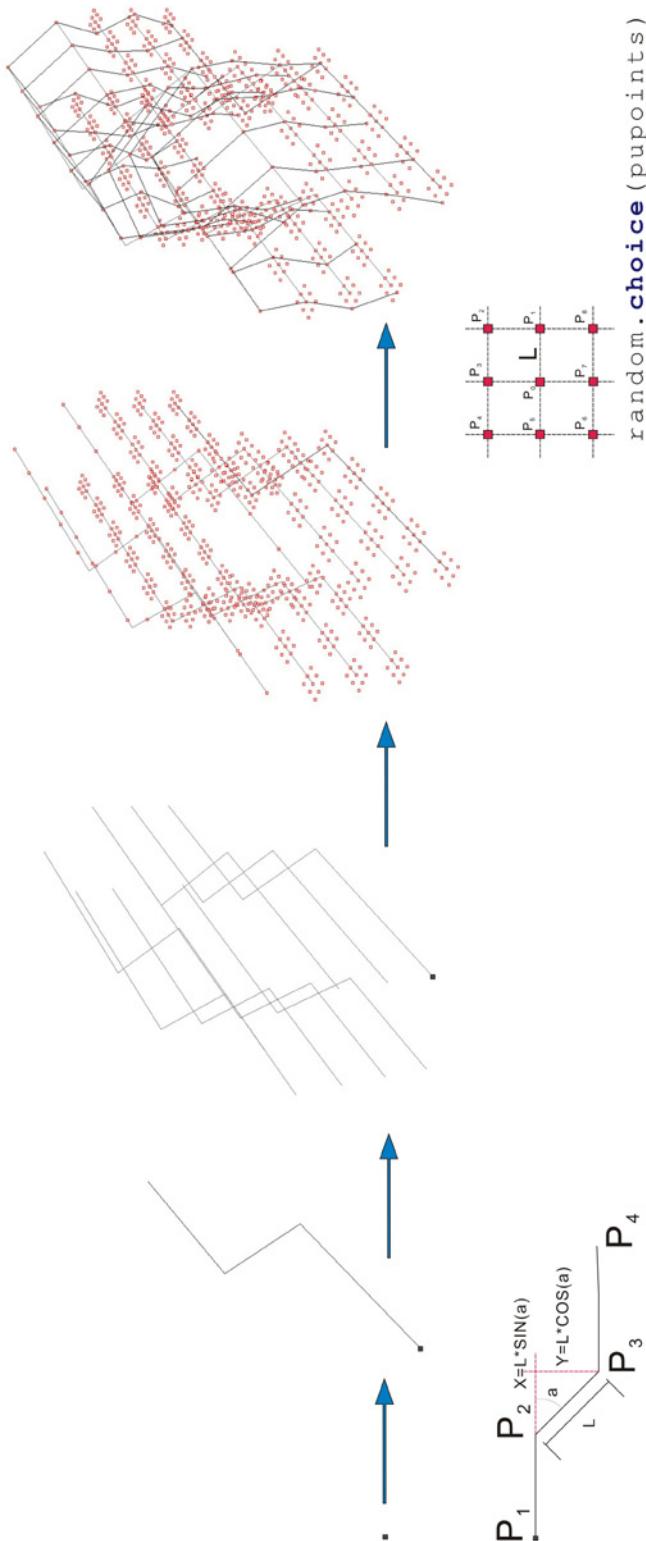
Rhino Python Editor - E:\01-DissERTATION\Phase03\03-Parameter\05-PythonGH\hyperbolicControl01.py

```

File Edit Debug Tools Help
untitled1 | newvertices.py | numberControl01.py | copy.py | 
1 import rhinoscriptsyntax as rs
2 import math
3 import random
4
5
6 def basiclines(basicpoint,lengthunit,angle,offsetvalue,topblineheight,
7 multiple1,multiple2,multiple3):
8     bpoint0=(basicpoint[0],basicpoint[1],basicpoint[2])
9     bpoints=[]
10    bpoints.append(bpoint0)
11    lengthunit=lengthunit
12    multiple1=multiple1
13    bpoint1=(bpoint0[0]+multiple1*lengthunit,bpoint0[1],bpoint0[2])
14    bpoints.append(bpoint1)
15
16    angle=angle
17    multiple2=multiple2
18    hypotenuse=multiple2*lengthunit
19    bpoint2=(bpoint1[0]+hypotenuse*math.sin(angle),\
20    bpoint1[1]+hypotenuse*math.cos(angle),bpoint1[2])
21    bpoints.append(bpoint2)
22
23    multiple3=multiple3
24    bpoint3=(bpoint2[0]+multiple3*lengthunit,bpoint2[1],bpoint2[2])
25    bpoints.append(bpoint3)
26
27    bline0=rs.AddPolyline(bpoints)
28    bplines=[]
29    bplines.append(bline0)
30
31    for i in range(1,4):
32        dividecurvelength=rs.CopyObject(bpline0,[0,0,5*i])
33        bplines.append(dividecurvelength)
34        offsetbplines=[]
35        offsetvalue=offsetvalue
36        for j in bplines:
37            offsetpline=rs.OffsetCurve(j,[0,0,0],offsetvalue)
38            offsetbplines.append(offsetpline)
39            topblineheight=topblineheight
40            topblinecenter=rs.OffsetCurve(bplines[-1],[0,0,0],offsetvalue/2)
41            topbpline=rs.CopyObject(topblinecenter,[0,0,topblineheight])

```

P-02_03



几何构建逻辑

首先要明确的是，写代码不是为了写代码而写代码，写代码的目的是协助设计师进行设计，并进行计算机模型的构建，因此，在开始写代码之前，最为重要的是缕清楚几何构建的逻辑(基本就是设计的逻辑)，同时需要搞清楚几何构建逻辑是否可以转化为代码，当这些都明确了之后，代码的书写自然就会流畅。

本例中几何构建逻辑，是由一个点开始的，通过控制单位长度构建折线，生成水平向的控制线，并增加9个控制点，从中随机的挑选一个，构建具有变化的截面，从而完成基本结构线的设计。

```

#导入需要用到的标准库
import rhinoscriptsyntax as rs
import math
import random

#basiclines函数定义建筑水平方向的结构线，输入条件为一个点
def basiclines(basicpoint,lengthunit,angle,offsetvalue,topbplineheight,
multiple1,multiple2,multiple3,floorheight):
    bpoint0=(basicpoint[0],basicpoint[1],basicpoint[2])
    bpoints=[]
    bpoints.append(bpoint0)
    lengthunit=lengthunit #lengthunit定义截面基本间距单元
    multiple1=multiple1 #定义第一段截面基本间距单元倍数
    bpoint1=(bpoint0[0]+multiple1*lengthunit,bpoint0[1],bpoint0[2])
    bpoints.append(bpoint1)

    angle=angle #angle定义建筑转折处的角度
    multiple2=multiple2 #定义第二段截面基本间距单元倍数
    hypotenuse=multiple2*lengthunit
    bpoint2=(bpoint1[0]+hypotenuse*math.sin(angle),\
    bpoint1[1]+hypotenuse*math.cos(angle),bpoint1[2])
    bpoints.append(bpoint2)

    multiple3=multiple3 #定义第三段截面基本间距单元倍数
    bpoint3=(bpoint2[0]+multiple3*lengthunit,bpoint2[1],bpoint2[2])
    bpoints.append(bpoint3)

    bpline0=rs.AddPolyline(bpoints) #底层水平结构线
    bplines=[]
    bplines.append(bpline0)
    fh=floorheight #层高
    #建立上部水平结构线
    for i in range(1,4):
        dividecurvelength=rs.CopyObject (bpline0,[0,0,fh*i])
        bplines.append(dividecurvelength)
        offsetbplines=[]
        offsetvalue=offsetvalue
        for j in bplines:
            offsetbpline=rs.OffsetCurve(j,[0,0,0],offsetvalue)
            offsetbplines.append(offsetbpline)
        topbplineheight=topbplineheight

```

```
topbplinecenter=rs.OffsetCurve(bplines[-1],[0,0,0],offsetvalue/2)
topbpline=rs.CopyObject(topbplinecenter,[0,0,topblineheight])
rs.DeleteObject(topbplinecenter)

return bplines,offsetbplines,topbpline

#basicpoints 定义用于建立截面结构线的点阵
def basicpoints(bplines,lengthunit):
    #定义位于水平结构线上的点和点阵
    basicplanepoints=[]
    for u in range(len(bplines)):
        basicplanepoint=rs.DivideCurveLength(bplines[u],lengthunit,True,True)
        basicplanepoints.append(basicplanepoint)

    lengthunit=lengthunit

    planeunit=1
    randomselectionp=[]
    pupoints=[]

    for o in range(len(basicplanepoints)-1):
        for p in range(len(basicplanepoints[0])):
            basicplanepointscor=[basicplanepoints[o][p][0],basicplanepoints[o][p][1],\ basicplanepoints[o][p][2]] #以
            列表的形式提取点的三维坐标

            pupoints.append(basicplanepointscor)

    #建立类似于九宫格的点阵
    pupoint1=[basicplanepointscor[0]+planeunit,\ basicplanepointscor[1],\ basicplanepointscor[2]]
    pupoints.append(pupoint1)
    rs.AddPoint(pupoint1)

    pupoint2=[basicplanepointscor[0]+planeunit,\ basicplanepointscor[1]+planeunit,\ basicplanepointscor[2]]
    pupoints.append(pupoint2)
    rs.AddPoint(pupoint2)

    pupoint3=[basicplanepointscor[0],\ basicplanepointscor[1]+planeunit,\ basicplanepointscor[2]]
    pupoints.append(pupoint3)
    rs.AddPoint(pupoint3)
```

```

pupoint4=[basicplanepoints[0]-planeunit,\n
          basicplanepoints[1]+planeunit,\n
          basicplanepoints[2]]\n
pupoints.append(pupoint4)\n
rs.AddPoint(pupoint4)\n\n
pupoint5=[basicplanepoints[0]-planeunit,\n
          basicplanepoints[1],\n
          basicplanepoints[2]]\n
pupoints.append(pupoint5)\n\n
rs.AddPoint(pupoint5)\n\n
pupoint6=[basicplanepoints[0]-planeunit,\n
          basicplanepoints[1]-planeunit,\n
          basicplanepoints[2]]\n
pupoints.append(pupoint6)\n
rs.AddPoint(pupoint6)\n\n
pupoint7=[basicplanepoints[0],\n
          basicplanepoints[1]-planeunit,\n
          basicplanepoints[2]]\n
pupoints.append(pupoint7)\n
rs.AddPoint(pupoint7)\n\n
pupoint8=[basicplanepoints[0]+planeunit,\n
          basicplanepoints[1]-planeunit,\n
          basicplanepoints[2]]\n
pupoints.append(pupoint8)\n
rs.AddPoint(pupoint8)\n\n
#使用random.choice函数随机在各点阵中选择一个点\n
randomselectionp.append(random.choice(pupoints))\n
pupoints=[]\n\n\n
pupoints0=randomselectionp[:len(basicplanepoints[0])]\n
pupoints1=randomselectionp[len(basicplanepoints[0]):-len(basicplanepoints[0])]\n
pupoints2=randomselectionp[-len(basicplanepoints[0]):]\n\n
pupoints4sub=basicplanepoints[-1]

```

```

        pupoints4=[]
for e in range(len(pupoints4sub)):
    pupoints4.append([pupoints4sub[e][0],pupoints4sub[e][1],pupoints4sub[e][2]])

sectionpolylinesparses=[]
for q in range(len(pupoints0)):
    #通过随机选择的点建立截面结构线
    sectionpolylinesparses.append(rs.AddPolyline((pupoints0[q],pupoints1[q],\
        pupoints2[q],pupoints4[q])))

return sectionpolylinesparses,pupoints4

#mainfunction函数时主要为interface code,建立于用户间的互动操作
def mainfunction():
    basicpoint=rs.GetPoint('Select one point:')
    if not basicpoint:return

    values=[5,120,12,5,4,3,4, 5]
    lengthunit=values[0]
    angle=values[1]
    offsetvalue=values[2]
    topbplineheight=values[3]
    multiple1=values[4]
    multiple2=values[5]
    multiple3=values[6]
    floorheight=values[7]

#与用户的互动程序，图为程序运行时Rhino命令行的提示，可以看到设计过程中主要用于控制建筑结构线的相关参数
    while True:
        prompt='Setting'
        result=rs.GetString(prompt,'Insert:','Lengthunit','Angle','Offsetvalue',\
            'Topbplineheight','Multiple1','Multiple2','Multiple3', 'Floorheight','Insert')
        if not result:return

        result=result.upper()
        if result=='LENGTHUNIT':
            f=rs.GetReal('Lengthunit:',values[0])
            if f is not None:lengthunit=f

```

```

elif result=='ANGLE':
    f=rs.GetReal('Angle(110-120):',values[1],110,120)
    if f is not None:angle=f
elif result=='OFFSETVALUE':
    f=rs.GetReal('Offsetvalue:',values[2])
    if f is not None:offsetvalue=f
elif result=='TOPBPLINEHEIGHT':
    f=rs.GetReal('Topbplineheight:',values[3])
    if f is not None:topbplineheight=f
elif result=='MULTIPLE1':
    f=rs.GetReal('Multiple1:',values[4])
    if f is not None:multiple1=f
elif result=='MULTIPLE2':
    f=rs.GetReal('Multiple2:',values[5])
    if f is not None:multiple2=f
elif result=='MULTIPLE3':
    f=rs.GetReal('Multiple3:',values[5])
    if f is not None:multiple3=f
elif result=='FLOORHEIGHT':
    f=rs.GetReal('Floorheight:',values[5])

    if f is not None: floorheight=f
elif result=='INSERT':break
bplines,offsetbplines,topbpline=basiclines(basicpoint,lengthunit,angle,\n
offsetvalue,topbplineheight,multiple1,multiple2,multiple3, floorheight)

sectionpolylinepart0,pupoints40=basicpoints(bplines,lengthunit)
sectionpolylinepart1,pupoints41=basicpoints(offsetbplines,lengthunit)

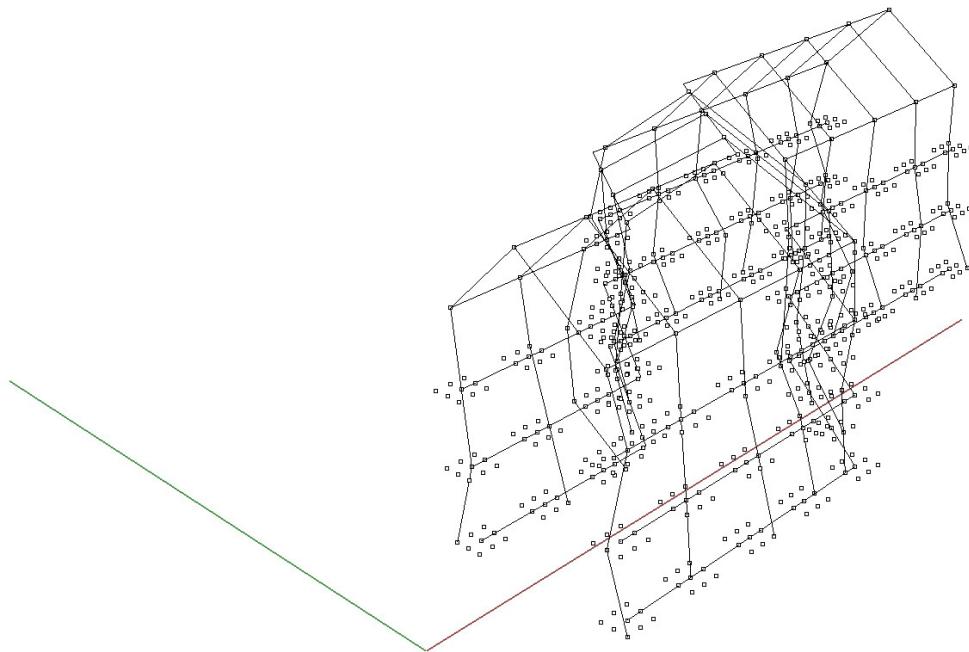
topdivide=rs.DivideCurveLength(topbpline,lengthunit,True,True)
topdividepoints=[]
for a in range(len(topdivide)):
    topdividepoints.append([topdivide[a][0],topdivide[a][1],topdivide[a][2]])

toppolyline=[]

for s in range(len(topdividepoints)):
    toppolyline.append(rs.AddPolyline([pupoints40[s],topdividepoints[s],pupoints41[s]]))

#执行函数
mainfunction()

```



Setting <Insert> (Lengthunit Angle Offsetvalue Topbplineheight Multiple1 Multiple2 Multiple3 Insert):

在与用户的交互中，提取了几个主要用于控制建筑结构线形态的参数：

Lengthunit: 截面间距单元控制距离；

Angle: 建筑转折角度；

Offsetvalue: 建筑进深；

Topbplineheight: 屋脊线相对高度；

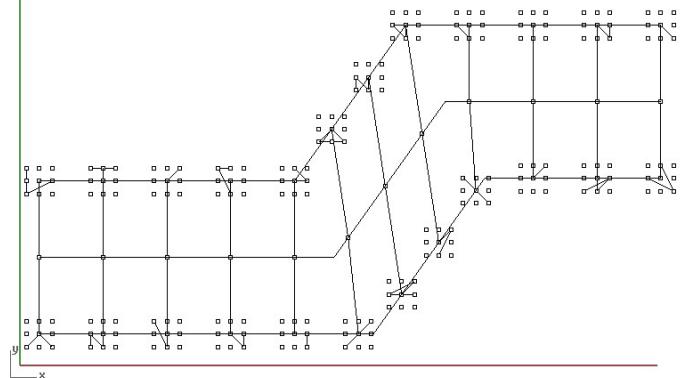
Multiple1: 建筑开始段截面间距单元控制距离倍数；

Multiple2: 建筑转折段截面间距单元控制距离倍数；

Multiple3: 建筑结束段截面间距单元控制距离倍数；

Floorheight: 建筑层高；

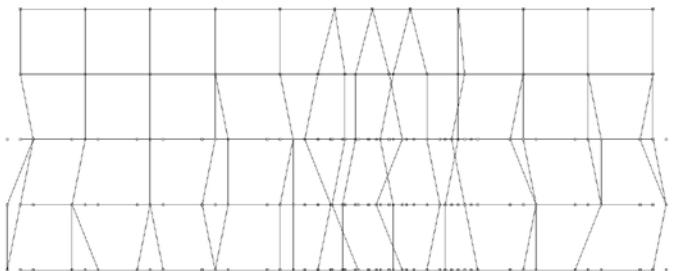
关于随机参数：



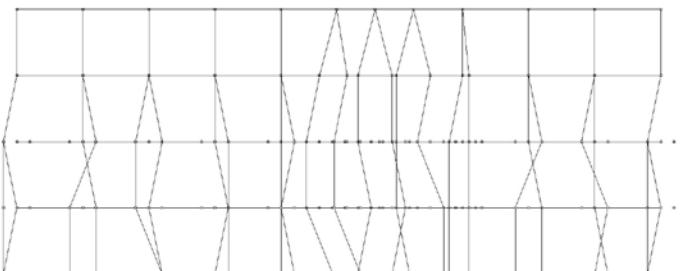
三个立面是在默认参数条件下，每次运行脚本后的结果，由于使用了随机函数choice (seq) 从非空序列seq中返回一个随机元素。

```
randomselectionp.append(random.choice(pupoints))
```

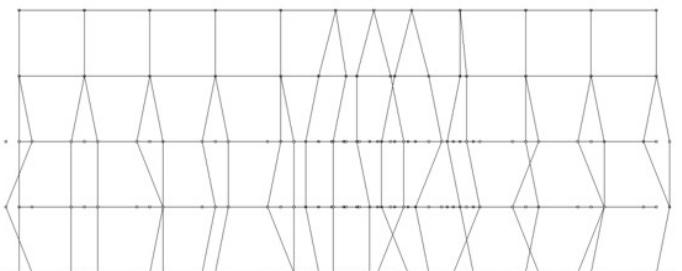
每次运行脚本就会产生不同的选择结果。



Random 1



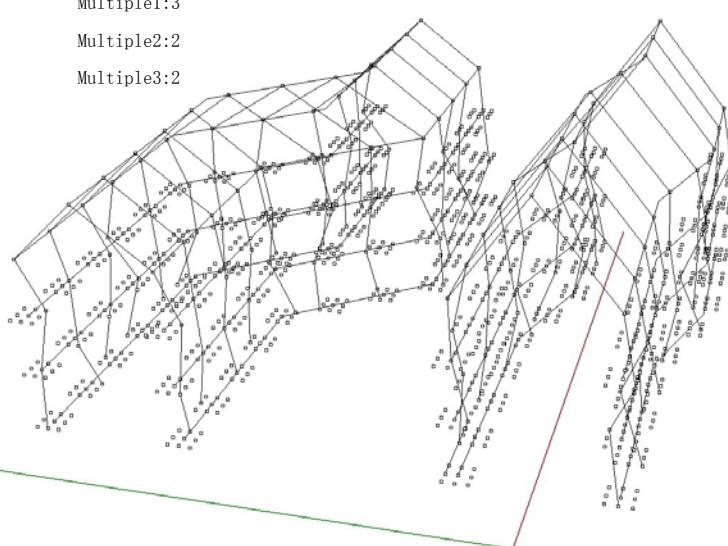
Random 2



Random 3

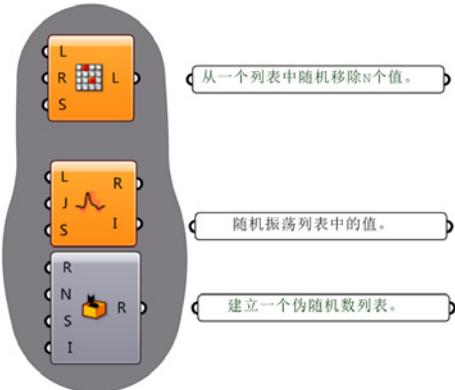
单位: m, °

左侧为默认参数： 右侧为部分参数调整：
Lengthunit:5 Lengthunit:8
Angle: 120 Angle: 115
Offsetvalue: 12 Offsetvalue: 12
Topbplineheight:5 Topbplineheight:5
Multiple1:4 Multiple1:3
Multiple2:3 Multiple2:2
Multiple3:3 Multiple3:2



Python脚本语言的编写，要与设计不断调整和修正的过程相协调，能够实时的观察脚本编写，参数变化所影响的几何图形变换，对于设计师来讲是很重要的。在脚本编写中，每完成一个基本步骤，就会运行一下程序，一方面来检查脚本语言的正确性并进行错误格式的调整，同时观察设计师最为关心的几何变化，确保几何沿着设计的目的进行，并时刻根据几何的修改来调整脚本。

那么对于GH来讲，GH所具有的代码与几何图形的实时显示调整过程比Python以及Rhinoscript脚本代码更加的具有交互性。GH一定程度上可以被看作是脚本方法，属性，函数和模块的节点化，将命令行和语言块转化为单一的可视化节点，例如：



几个GH中用于随机的组件（Component）。

Python的Random模块所提供生成随机函数的方法比GH多出很多，例举如下，资料来源于，《Python Essential Reference (Fourth Edition)》—David M. Beazley，也可以在 Python的官方网站获得更详细的资料。

在 Python中搜索Random模块，获得模块内的函数：

```
>>> import random
>>> dir(random)
['BPF', 'LOG4', 'NV_MAGICCONST', 'RECIP_BPF', 'Random', 'SG_MAGICCONST', 'SystemRandom', 'TWOPI', '_BuiltinMethodType', '_MethodType', '_Sequence', '_Set', '__all__', '__builtins__', '__cached__', '__doc__', '__file__', '__name__', '__package__', '_acos', '_ceil', '_cos', '_e', '_exp', '_inst', '_log', '_pi', '_random', '_sha512', '_sin', '_sqrt', '_test', '_test_generator', '_urandom', '_warn', 'betavariate', 'choice', 'division', 'expovariate', 'gammavariate', 'gauss', 'getrandbits', 'getstate', 'lognormvariate', 'normalvariate', 'paretovariate', 'randint', 'random', 'randrange', 'sample', 'seed', 'setstate', 'shuffle', 'triangular', 'uniform', 'vonmisesvariate', 'weibullvariate']
```

R_{andom}:

Random模块提供各种用于生成伪随机数的函数，以及根据不同的实数分布来随机生成值的函数。该模块中的多数函数取决于函数random()，该函数使用 Mersenne Twister生成器在[0.0, 1.0]范围内生成一致分布的数值。

种子和初始化

以下函数用于控制基础随机数生成器的状态。

Seed ([x])： 初始化随机数生成器。如果省略x或x为None，则使用系统时间来设置生成器。否则，如果x是整数或者长整数，则使用该值。如果x不是整数，那么它必须是可散列的对象并且将hash (x) 的值作为种子。

Getstate ()：返回表示当前生成器的对象。稍后将该对象传递到setstate()来恢复状态。

Setstate(state)：从 getstate()返回的对象中恢复随机数生成器的状态。

Jumpahead(n)：如果在一行中调用了n次random()，则快速将生成器状态改为其应有的状态。N必须是非负的整数。

随机整数

以下函数用于操作随机整数。

Getrandbits(k)：创建包含k个随机位的长整数。

Randint(a,b)：返回随机整数x，范围是a<=x<=b。

Randrange(start,stop,[step])：返回一个范围在 (start,stop,step) 之间的随机整数。不包括结束值。

随机序列

以下函数用于产生随机序列数据。

`Choice(seq)`:从非空序列`seq`中返回一个随机元素。

`Sample(s,len)`:返回序列长度`len`,它包含从序列`s`中随机选择的元素。结果序列中的元素按照选择它们的顺序排列。

`Shuffle(x,[random])`:随机选择列表`x`中的项, `random`是可选参数, 它指定随机生成函数。如果提供该参数, 则不能包含参数并且返回范围在[0.0, 1.0]内的浮点数的函数。

实数随机分布

以下函数生成实数的随机数。分布和参数名与概率和统计中使用的标准名称一致。需要查询其它相关内容来了解更多细节。

`Random()`:返回范围在[0.0, 1.0]之间的随机数。

`Uniform[a,b]`:返回范围在[\[a, b\]](#)之间的一致分布随机数。

`Betavariate(alpha,beta)`:从beta分布中返回一个在0和1之间的值, 其中`alpha>-1`而`beta>1`.

`Cunifvariate(mean,arc)`: 圆形一致分布, `mean`是平均角, 而`arc`是沿平均角周围居中的分布范围。这些值都必须设置在0到pi之间的弧度范围内。返回值范围是 (`mean-arc/2,mean+arc/2`) 。

`Expovariate(lambd)`:指数分布, `lambd`是由1.0除以预期均值。返回值范围是[0, +∞]。

`Gammavariate(alpha,beta)`:gamma分布, 其中 `alpha>-1,beta>0`.

`Gauss(mu,sigma)`:均值为`mu`且标准偏差为`sigma`的高斯分布。比`normalvariate()`稍快。

`Lognormvariate(mu,sigma)`:对数正态分布。取该分布的自然对数的结果是均值为`mu`且标准偏差为`sigma`的正态分布。

`Normvariate(mu,sigma)`:均值为`mu`且标准偏差为`sigma`的正态分布。

`Paretovariate(alpha)`:形态参数为`alpha`的pareto分布。

`Triangular([low[,high[,mode]]])`:三角分布。随机数n的范围是`low≤n≤high`, 且模式为`mode`。默认情况下, `low`是0, `high`是1.0, 而`mode`设为`low`和`high`的中点值。

`Vonmisesvariate(mu,kappa)`:von Mises分布, 其中`mu`是平均角, 弧度范围在0到2*pi之间, 而`kappa`是非负集中因子。如果`kappa`为零, 则该分布简化为统一随机角, 范围在0到2*pi之上。

`Weibullvariate(alpha,beta)`:weibull分布, 比例参数为`alpha`, 形状参数为`beta`。

注意:

该模块中的函数都不是线程安全的。如果要在不同线程中生成随机数, 就应当使用锁定以防止并发访问;

随机数生成器的区间(在数值开始重复之前)是`2**19937-1`;

该模块生成的随机数都是确定的, 不应用于密码;

通过实现`random.Random`的子类并实现`random()`, `seed()`, `getstate()`, `setstate()`和`jumpahead()`方法, 就可以创建新的随机数生成器类型。实际上, 该模块中的所有其它函数在内部都作为`random`的方法实现的。因此, 可以将他们作为新的随机数生成器实例的方法来访问;

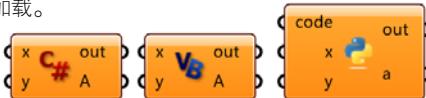
该模块提供了两种供选择的随机数生成器类——`WichmannHill`和`SystemRandom`, 通过实例化正确的类并将前面的函数作为方法进行调用就可以使用。`WichmannHill`类实现了早期Python发布版中所使用的Wichmann-Hill生成器。

`SystemRandom`类使用系统随机数生成器`os.urandom()`生成随机数;

如果需要真的随机性, 应该使用`os`模块的`urandom`函数。

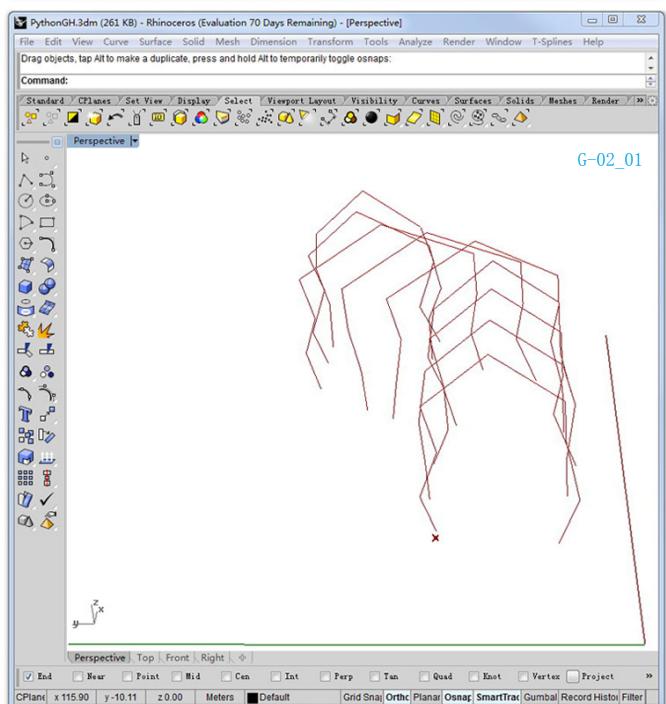
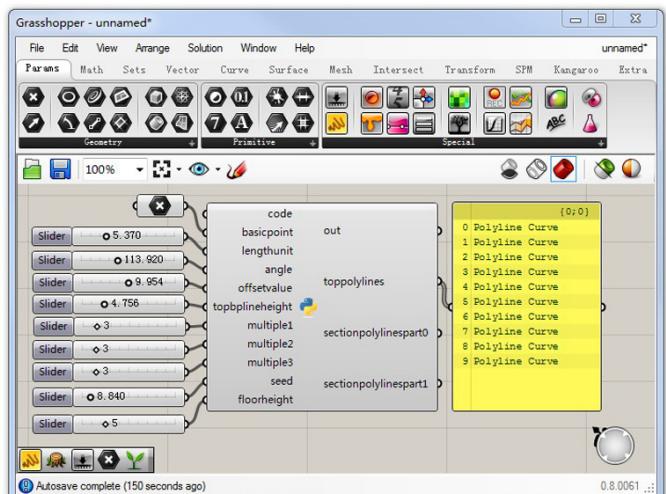
2 GH Python

比较GH与Python脚本，Python命令行形式的编程环境是基础的语言，可以理解为GH中所有的命令都可以通过Python命令行形式的语言得以实现，甚至可以扩展GH的组件，也就是为什么在了解了GH之后，还要学习脚本语言。使用GH，在设计的过程中不免受制于GH现有组件，有些设计逻辑不得不求助于Python等脚本语言才可以实现。将GH与Python结合可以使GH功能无限扩展。实际上GH内嵌了c#,VB语言，目前GH版本Version November-24,2011,Build0.8.0031的Python需要额外加载。



使用GH中的Python组件把RhinoPython中编写的程序调入进来，在调入的过程中，需要根据GHPython的输入，输出的方式调整代码。

这里把主要的控制参数作为输入条件，包括basicpoint,lengthunit,angle,fsetvalue,topbpplineheight,multiple1,multiple2,multiple3,floorheight。同时，针对random.choice函数，增加了初始化随机数生成器，seed([x]),在GH中可以实时的获得随机几何形式，增加设计调整的便利。下面列出主要针对GHPython组件调整的代码部分，与RhinoPython一致的部分以省略号替代。



```

import rhinoscriptsyntax as rs
import math
import random

#将输入的条件放置在函数的外面
values=[5,120,12,5,4,3,4]
if lengthunit is None:lengthunit=values[0]
if angle is None:angle=values[1]
if offsetvalue is None:offsetvalue=values[2]
if topbplineheight is None:topbplineheight=values[3]
if multiple1 is None:multiple1=values[4]
if multiple2 is None:multiple2=values[5]
if multiple3 is None:multiple3=values[6]
if floorheight is None:floorheight=values[7]
if basicpoint is None:basicpoint=(0,0,0)
basicpoint=rs.PointAdd (basicpoint,(0,0,0))
random.seed(seed) #增加的初始化随机数生成器， seed([x])

#调整交换函数interface code
def basiclines(basicpoint,lengthunit,angle,offsetvalue,topbplineheight,
multiple1,multiple2,multiple3,floorheight):
...
def basicpoints(bplines,lengthunit):
...
Def mainfunction(basicpoint,lengthunit,angle,offsetvalue,topbplineheight,multiple1,\
multiple2,multiple3, floorheight):
    lengthunit=lengthunit
    angle=angle
    offsetvalue=offsetvalue
    topbplineheight=topbplineheight
    multiple1=multiple1

    multiple2=multiple2
    multiple3=multiple3
    basicpoint=basicpoint
    floorheight=floorheight

...
    bplines,offsetbplines,topbpline=basiclines(basicpoint,lengthunit,angle,\ 
    offsetvalue,topbplineheight,multiple1,multiple2,multiple3,floorheight)
#主函数返回值，可以在外部调用

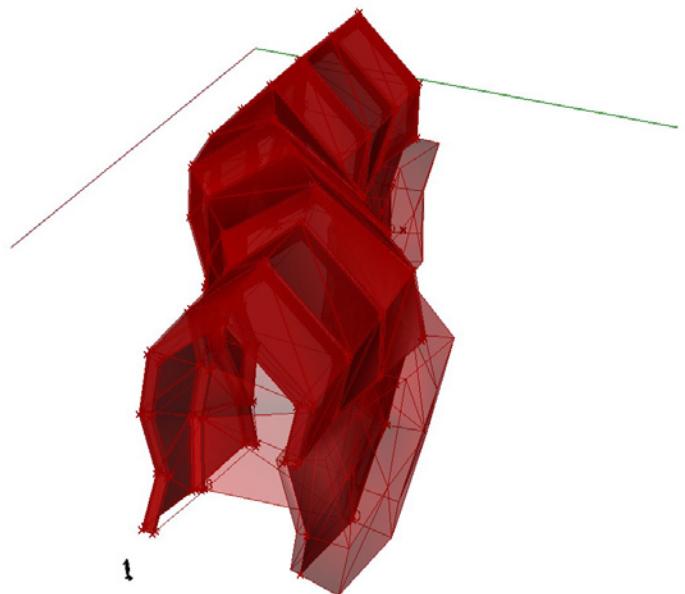
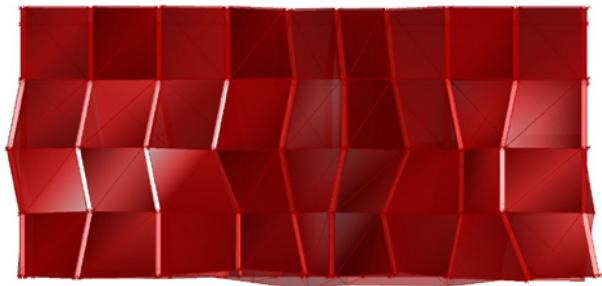
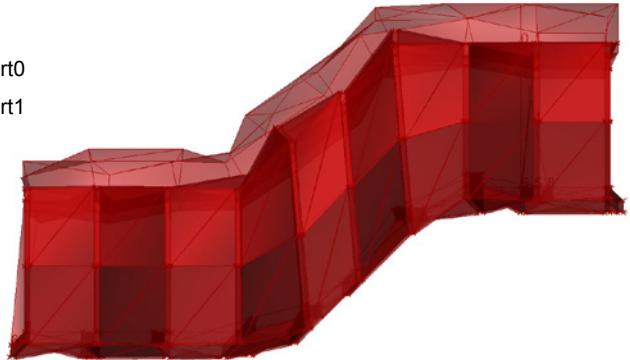
```

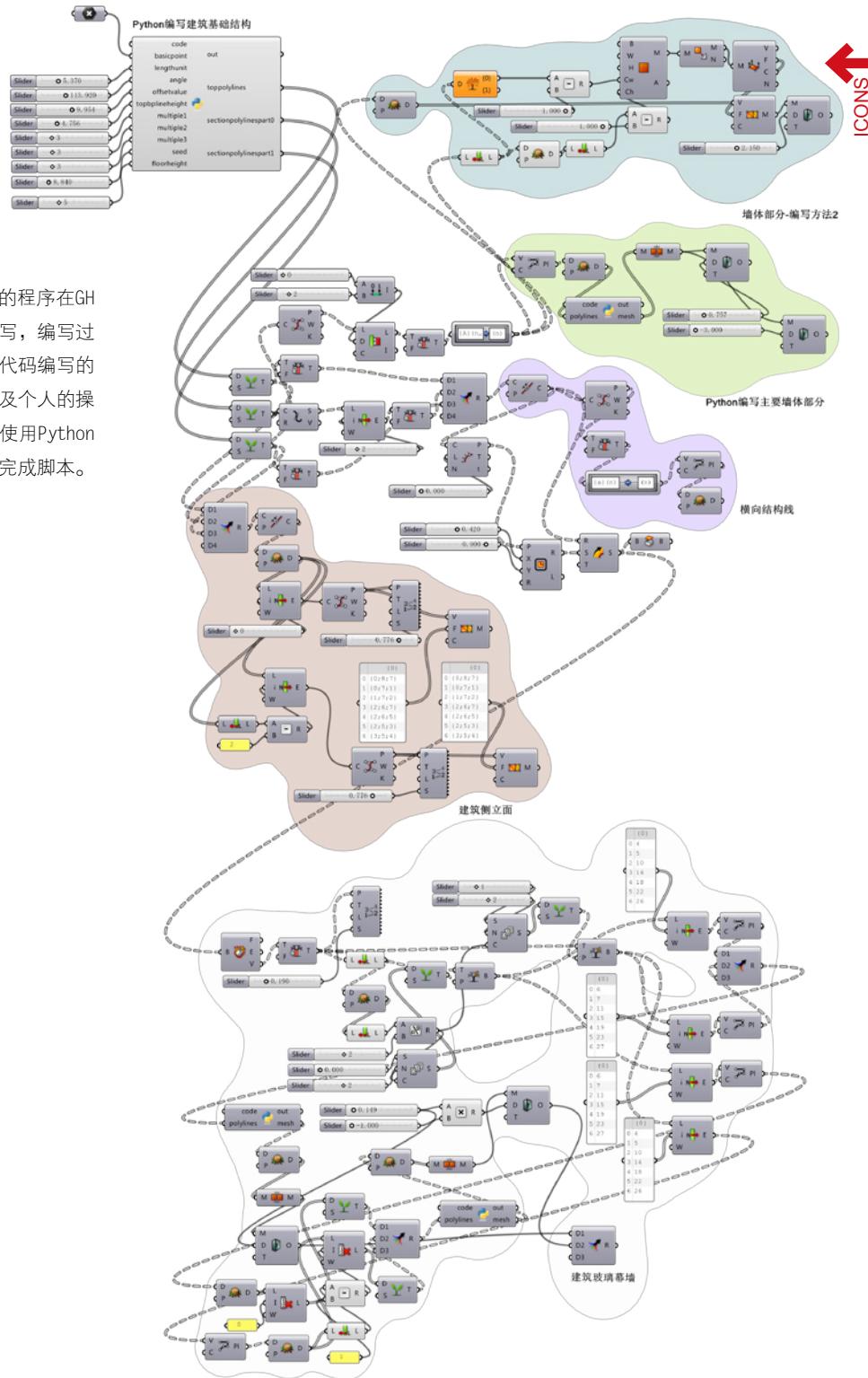
```
return toppolylines,sectionpolylinepart0,sectionpolylinepart1
```

```
toppolylines,sectionpolylinepart0,sectionpolylinepart1=mainfunction(basicpoint,lengthunit,angle,offsetvalue,to  
pbplineheight,multiple1,\  
multiple2,multiple3,floorheight)
```

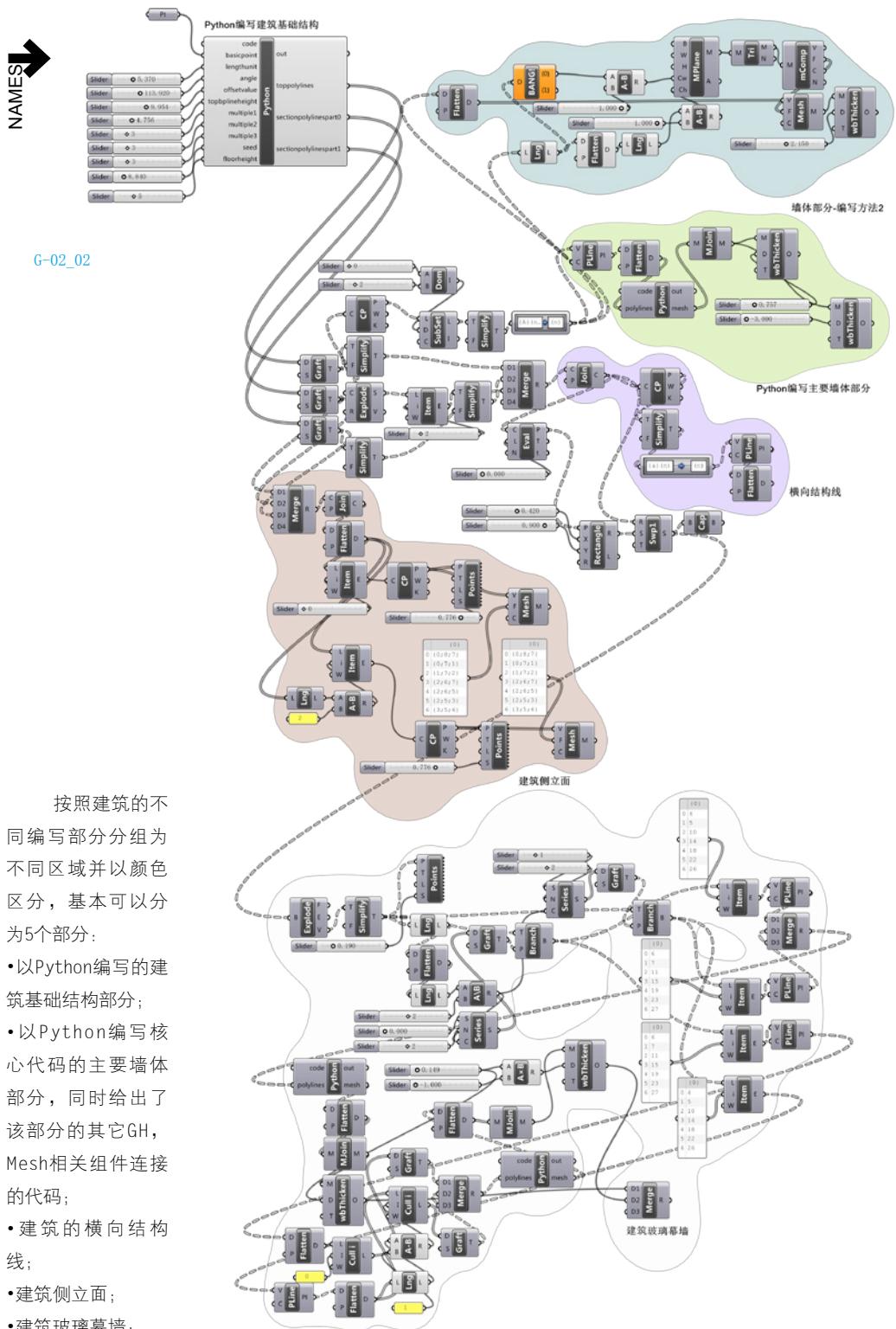
#GHPython的输出端值

```
toppolylines=toppolylines  
sectionpolylinepart0=sectionpolylinepart0  
sectionpolylinepart1=sectionpolylinepart1
```



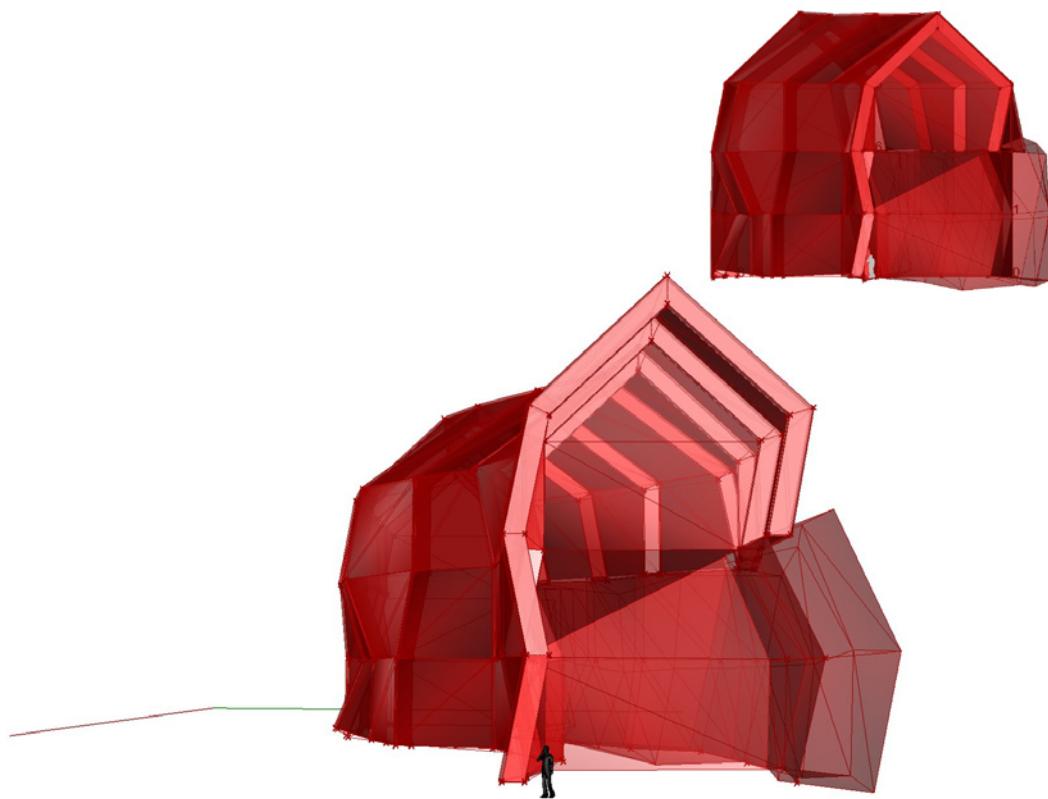


余下的程序在GH中继续编写，编写过程中根据代码编写的逻辑，以及个人的操作习惯，使用Python语言协助完成脚本。



按照建筑的不同编写部分分组为不同区域并以颜色区分，基本可以分为5个部分：

- 以Python编写的建筑基础结构部分；
- 以Python编写核心代码的主要墙体部分，同时给出了该部分的其它GH，Mesh相关组件连接的代码；
- 建筑的横向结构线；
- 建筑侧立面；
- 建筑玻璃幕墙；

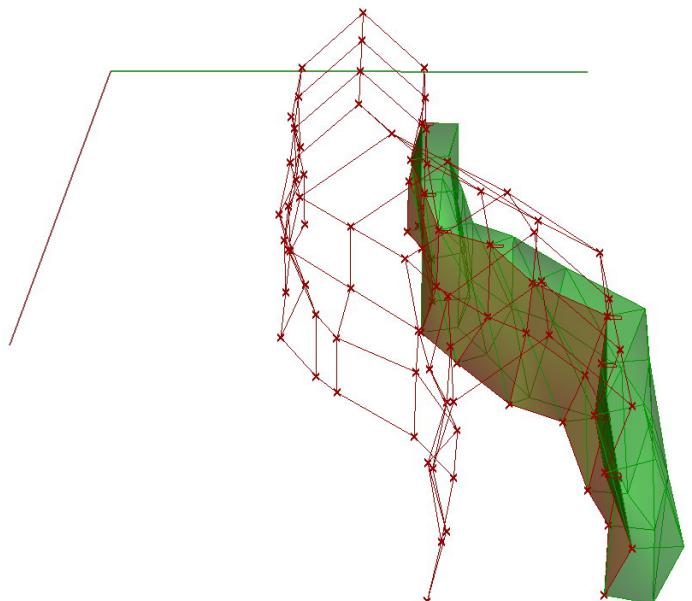
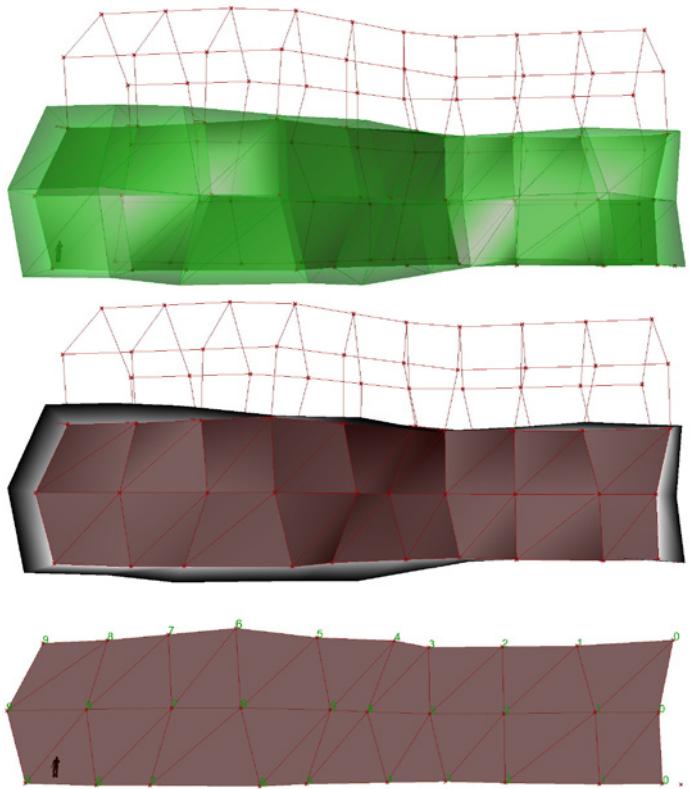


3 建筑的表皮结构

这里所涉及到的建筑表皮结构，是基于概念设计层面的几何探讨，还没有深入到具体的构造层面，而是将更多的重点放在计算机算法辅助下的几何形体。在面的计算机语言表达下，主要有两种类型的面，NURBS以及MESH (Polygon)。

NURBS全名为Non-uniform rational B-spline(非均匀有理样条曲线)。NURBS曲线是在B-样条曲线和贝塞尔曲线基础上发展而来的，目前在计算机图形学中已经成为描述曲线和曲面的一种方式。NURBS建模技术可以精确描述一条空间曲线，并通过曲线精确描述3D曲面和实体。NURBS对于计算机辅助设计、制造和工程 (CAD、CAM、CAE) 是几乎无法回避的，并且是很多业界广泛采用的标准的一部分，例如IGES、STEP和PHIGS。

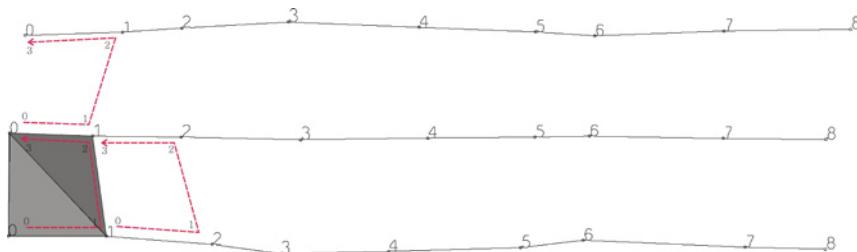
Polygon (Mesh) 是二维多边形，通常由三角和四角的Polygon多边形平面组成。单个Polygon通常可以进行着色 Colored，产生阴影 Shaded，赋予材质 Textured，Polygon的精确位置可以通过定点坐标进行确定。Mesh的方式，不仅是几何的构建，GIS、Ecotect等地理，生态分析也都是基于Mesh。因此，在Rhino中，Mesh的作用不仅是构建模型，几何Mesh的分析为设计师同样带来不小的帮助，未来这也是mesh的重点发展方向，从而将更多方面的协同设计整合进来。在建筑的表皮结构构建中，使用的是Mesh面的处理方式，主要以墙体部分的构建进行说明，玻璃幕墙部分与之类似，只要使用同样的方法就可以构建。



编写的Python组件，用于Mesh面的构建



构建Mesh面的关键是调整点的顺序，每四个围合的点形成一个Mesh面



```

import rhinoscriptsyntax as rs
curves=polylines #向GHPython组件中输入Polylines
print(curves)

knots=[]
for i in curves:
    knots.append(rs.PolylineVertices(i)) #提取Polylines的顶点
print(knots)

#使用for循环语句重新组织顶点顺序
mesh=[]
for m in range(len(knots)-1):
    for n in range(len(knots[m])-1):
        meshivertice=[]
        meshivertice.append(((knots[m][n][0],knots[m][n][1],knots[m][n][2]),\
        (knots[m][n+1][0],knots[m][n+1][1],knots[m][n+1][2]),\
        (knots[m+1][n+1][0],knots[m+1][n+1][1],knots[m+1][n+1][2]),\
        (knots[m+1][n][0],knots[m+1][n][1],knots[m+1][n][2])))
        print(meshivertice[0])
        fv=[(0,1,3,3),(1,2,3,3)] #构建face_vertices，组织生产面的方式
        mesh.append(rs.AddMesh(meshivertice[0],fv)) #使用调整后的顶点构建mesh面
        meshivertice=[]
        print(meshivertice)

mesh=mesh #mesh面的输出

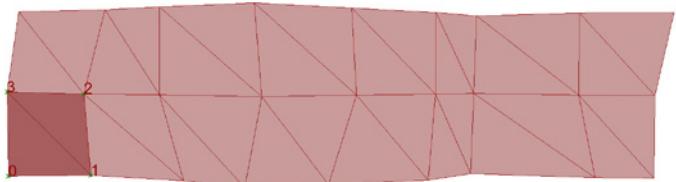
```

几何构建逻辑与Python代码

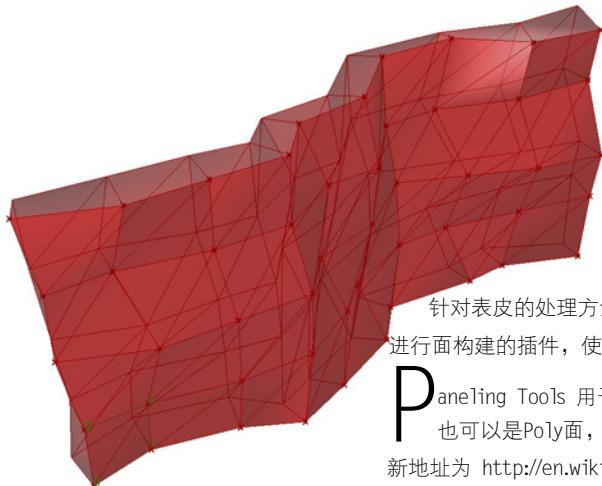
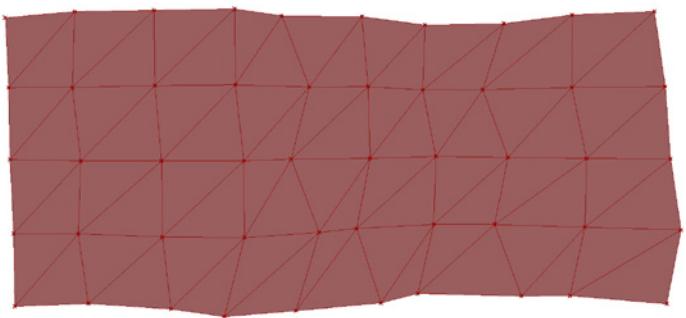
PanelingTools的Grasshopper版本：



`fv=[(0,1,3,3),(1,2,3,3)]`语句是构建面的顶点`face_vertices`,可以根据最终设计的目的,调整顶点组织方式,将顺序调整为`fv=[(0,1,3,3),(1,2,3,3)]`, Mesh面的对角线划分三角面的方向发生了改变,结果如下:



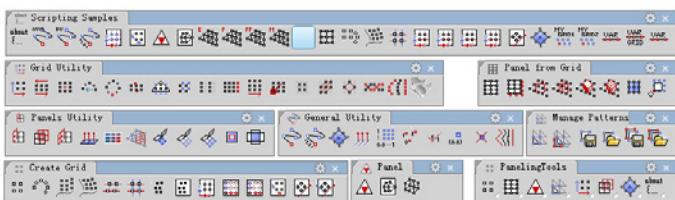
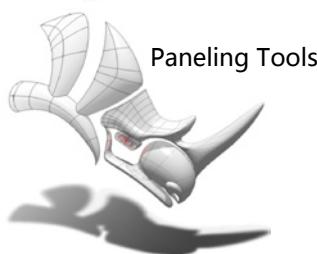
编写了Python的程序之后,可以根据几何构建逻辑获得更丰富的几何形式:



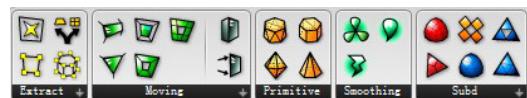
关于表皮构建的工具

针对表皮的处理方法,Rhino和GH都相应的开发了一些有助于设计师进行面构建的插件,使用这些插件可以增加几何构建的效率。

Paneling Tools 用于模型的表皮构建,构建的表皮可以是NURBS表皮也可以是Poly面,该工具为免费插件可以在其官方网站上下载,更新地址为 <http://en.wiki.mcneel.com/default.aspx/McNeel/PanelingTools.html>.

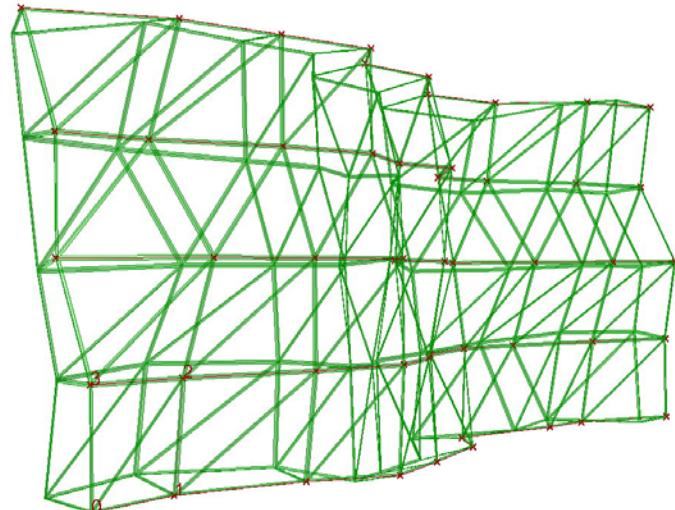


学习Paneling Tools工具手册《Paneling Tools Manual》,可以在其官网上下载

Weaver Bird官方网站 <http://www.giuliopiacentino.com/portfolio/>

- Catmull–Clark smoothing (wbCatmullClark):根据1978年Edwin Catmull 与 Jim Clark首次提出的网格递归细分方法。根据递归的次数，结果网格将由不同数量和形式的四边形构成；
- ▲ Split mesh into Quads (wbSplitQuad):用于计算产生一个新的网格细分，获得的新网格与原有网格近似，可以认为是Catmull–Clark smoothing方法的一种拓补；
- Loop smoothing (wbLoop): 根据1987年，Charles Loop首次在他的论文中所描述的网面递归细分类型进行计算，获得的结果网格由多个三角面组成；
- ▲ Split mesh with inner face (wbSplitPolygons): 从每一个原始面边缘的中心分离出一个新面，并使用Sierpinski三角形封面；
- ▲ Sierpinsky Triangles subdivision (wbSierpinskyTriangle): 在每一个面里产生一个或多个三角面，而每次居中的面为开敞的三角形；
- Frame (wbFrame): 在每一个面的中心开洞，洞口的形式与原有面近似，产生一个类似于画框的几何形式，边缘的方式为内外四角点连接；
- Carpet (wbCarpet): 在每一个面的中心开洞，洞口的形式与原有面近似，边缘的方式沿洞口边缘线延伸；
- Window (wbWindow): 在内部产生一个新面替代原有面，组成面的边数与原有面一致；

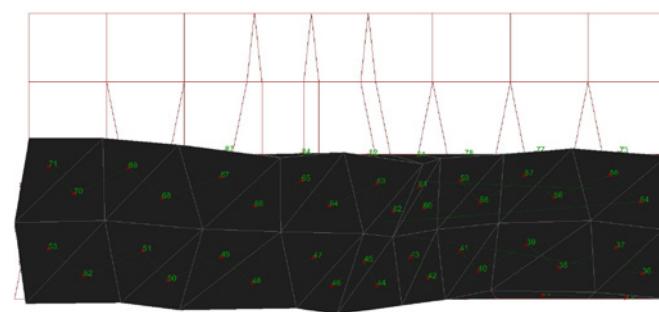
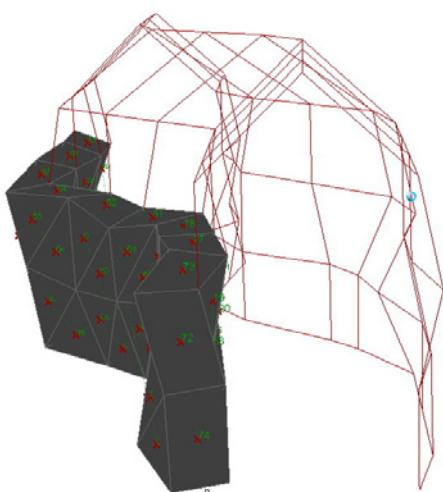
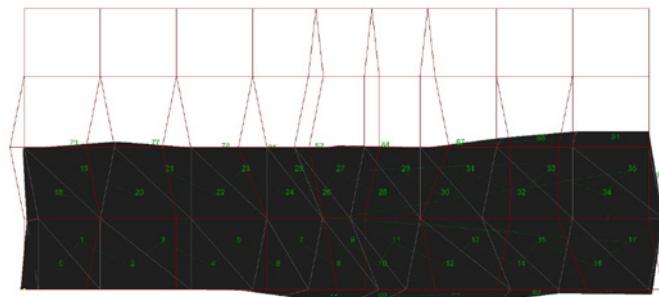
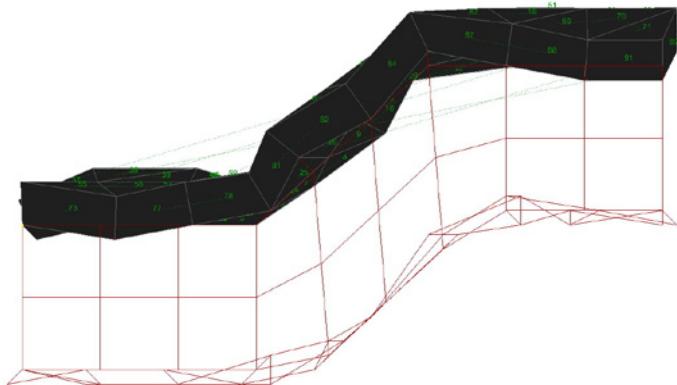
使用WB能够轻松处理一些基本的网格化分的问题，不用再单独的编写程序，这也是GH节点式编程的优势，将各种程序的命令封装成一个个具有特殊功能的节点使用。当然方便的组件的方式，也带来一定的局限性，例如不宜拓展更多地设计形式，同时都使用同样逻辑关系的几何，设计结果不免有相似。对于组件的使用需要根据设计师的目的权衡与使用，尤其在使用产生相似几何逻辑关系的组件时。



4 结构建构

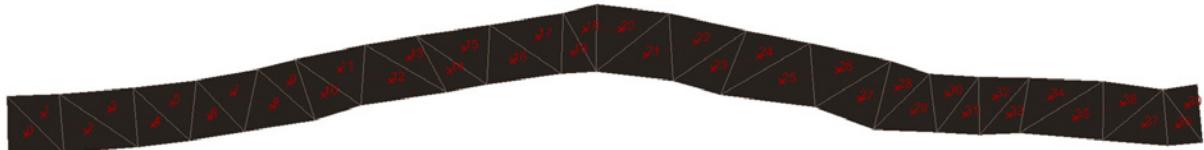
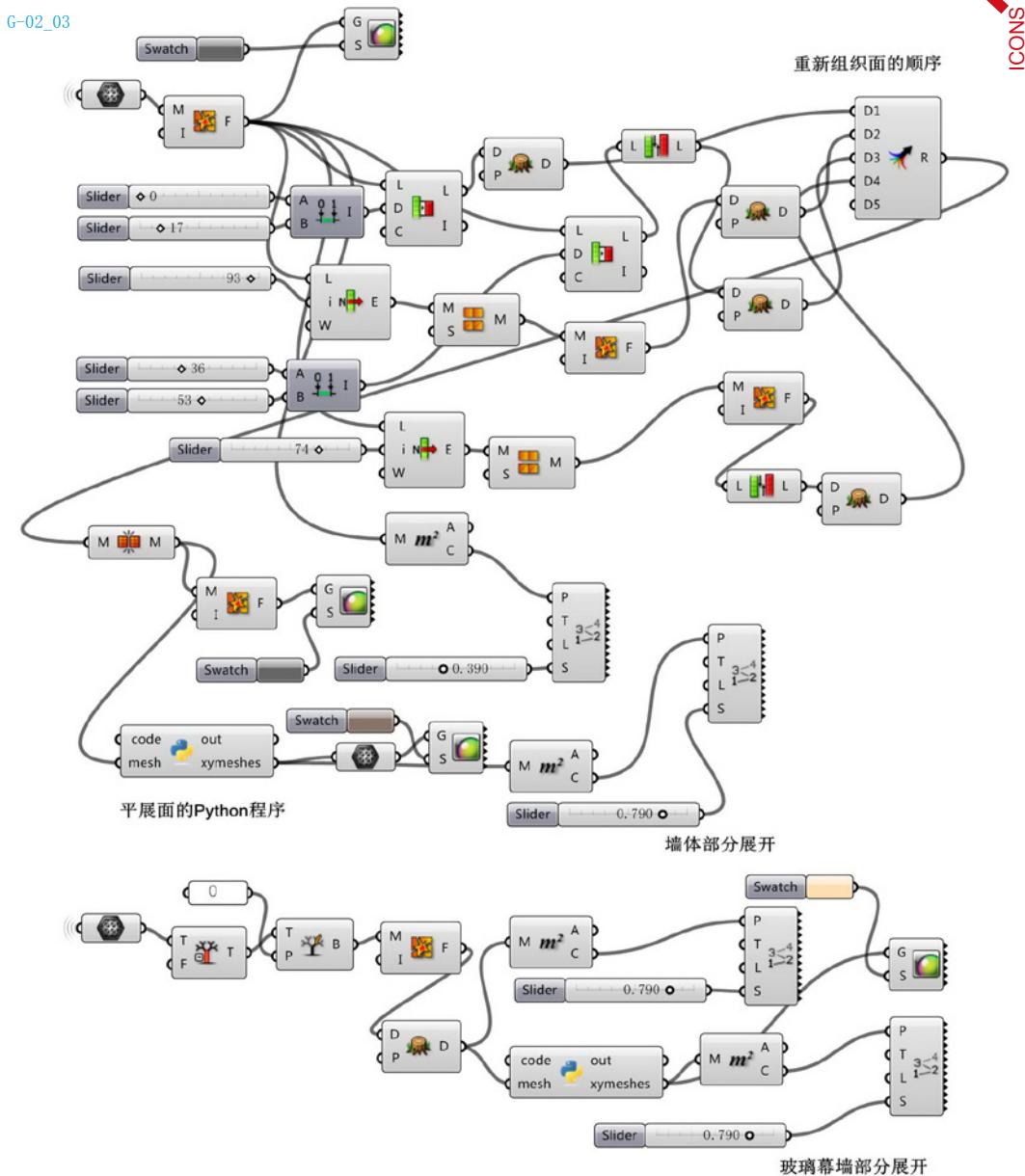
几何构建完成后，在向施工图转换的过程中，仍然会有很多需要计算机技术解决的问题，例如建筑表皮面的延展，建筑构建的几何体块的切割，很多问题往往是几何构建完成后，在思考如何构建时出现的，因此很多问题的解决仅依靠GH现有的组件处理起来会很繁琐，这时候结合Python语言协同处理往往能够化繁为简。

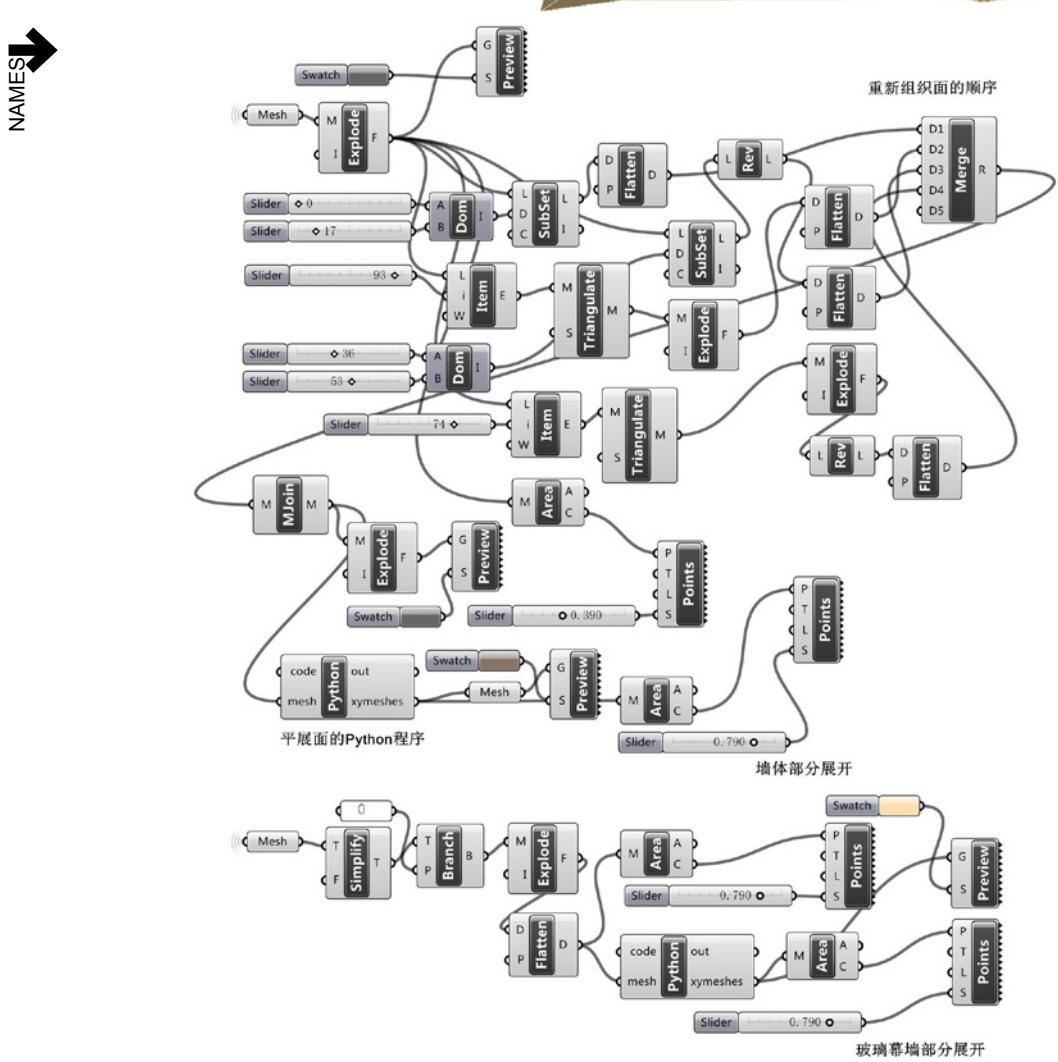
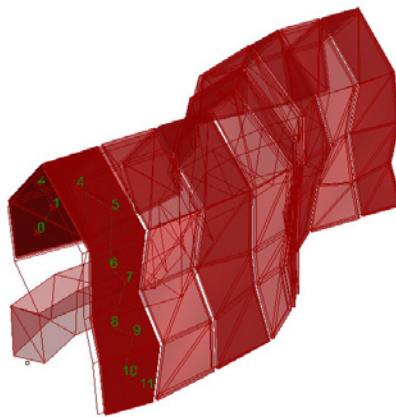
对于前述的建筑，需要把建筑表皮延展开来，如果将一块块三角面单独在XY平面内展平，RH与GH都可以解决，不过不会形成一个连续的面。这里处理的目的，是希望能够在展平的同时，各个面根据本来的衔接顺序在XY平面内连续排列，并互相的衔接。



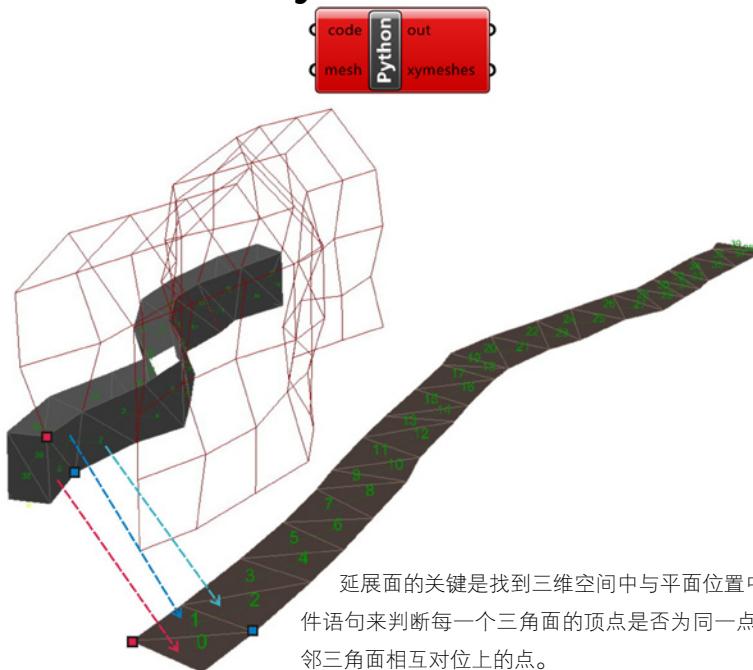
首先对墙体部分的Mesh面做处理，为了获得展开后Mesh面的连续性，分开为上下两部分。在处理的过程中需要调整面的顺序，使其保持连续。

G-02_03





几何构建逻辑与Python代码



```

import rhinoscriptsyntax as rs
mesh=mesh #导入外部程序的Mesh面

meshes=rs.ExplodeMeshes(mesh) #分解单一Mesh面为多个

xymeshes=[]
for i in range(len(meshes)):
    if i ==0: #处理分解后第一个Mesh面的展平位置
        mesh0point=rs.MeshVertices(meshes[i])
        mesh0points=[]
        for r in mesh0point:
            mesh0points.append([r[0],r[1],r[2]])

        xymesh0=rs.OrientObject(meshes[i],mesh0points,\n[[0,10,0],[10,0,0],[0,0,0]],1)
        xymeshes.append(xymesh0)

    else: #余下Mesh面的循环遍历
        vertices2=rs.MeshVertices(meshes[i])
        vertices1=rs.MeshVertices(meshes[i-1])

```

```

vertices2lst=[]
vertices1lst=[]
for q in vertices2:
    vertices2lst.append([q[0],q[1],q[2]])
for p in vertices1:
    vertices1lst.append([p[0],p[1],p[2]])

#找到相邻两个面的共同顶点
ver=[m for m in vertices1lst for n in vertices2lst if m==n]
a=ver[0]
b=ver[1]

#找到相邻两个面共同顶点的索引
indexa=vertices1lst.index(a)
indexb=vertices1lst.index(b)
#找到相邻两个面不共边的顶点
cref=[m for m in vertices1lst if m not in ver][0]
cv=[m for m in vertices2lst if m not in ver][0]

#定义面的延展方向
refvertice=rs.MeshVertices(xymeshes[i-1])
refvertices=[]
for x in refvertice:
    refvertices.append([x[0],x[1],x[2]])
indexc=[c for c in range(0,3) if c !=indexa and c!=indexb]
print(indexc)
refverticespoint=rs.MirrorObject(rs.AddPoint(refvertices[indexc[0]]),refvertices[indexa],refvertices[indexb])
mirrorpoint=[rs.PointCoordinates(refverticespoint)]

for z in mirrorpoint:
    mirrorpoint=[z[0],z[1],z[2]]

#获得面的延展
xymesh=rs.OrientObject(meshes[i],[a,b,cv],[refvertices[indexa],refvertices[indexb],mirrorpoint],1)
xymeshes.append(xymesh)
print(xymesh)

vertices2lst=[]
vertices1lst=[]
ver=[]
print(xymeshes) #可以用Print函数检查结果

```

处理设计的过程中，往往涉及到两个方面的问题，一个是设计几何，例如规划、建筑、景观最后的图式，另外就是分析几何，例如规划中城市结构，功能部件的合理性，建筑的热工，景观地形水体的分析对设计的影响等。在分析设计的过程中，往往是在设计前有个对场地的基本分析，可以使用相关的分析程序，例如Arcgis的空间分析，三维分析，统计分析，水文分析，基于流体软件的风分析，河流等水体的分析，Energyplus的热环境分析，Ecotect对建筑的生态分析等等，其所涉及的领域和相关支持的软件很多。同时，在设计的各个阶段都会结合分析几何，例如建筑的设计概念阶段，将模型导入到相关的其他分析软件中进一步处理相关的分析，如上述所提到的Ecotect中分析热环境、光环境、日照分析等，观察分析的结果再对设计进行修正。

现在以及未来的发展方向将是分析和设计进一步融

合，不用单独在不同的软件平台上进行操作，而是在一个界面下实现，这样就能够更加方便的对设计做出快速的反应。这种发展的趋势在Rhino以及GH中体现的很明显。我们把这种设计的方式可以认为是协同设计方法，即在设计几何的同时，不断地进行各种分析，协助设计师判断阶段性的设计几何对结果所产生的影响。

需要强调的是，各种类型的分析并不仅是一个可以使用的程序。每一类分析都是设计知识架构的一个分支，而这个分支仍旧在不断地拓展中。如果想进行水文分析，就得了解水文的知识，汇流累积量，水流长度，河网分级，流域分割等概念，所以设计师需要能够不断地跟进知识的发展，来拓展设计的潜力，从而有的放矢的使用不同的分析策略协同设计。

第三部分 协同设计方法

1 GECO



<http://www.food4rhino.com/project/geco>

在生态策略部分，已经对Ecotect有了深入的阐述，Ecotect主要用于辅助设计的早期设计阶段对环境问题的分析，并以可视化的方式表达，从而更加直观的阐述问题。但是Ecotect与Rhino以及其Grasshopper并没有直接的联系，一般是在RH中处理完部分设计后，导出模型在Ecotect中再进行分析几何部分，获得数据结果，调整设计。Geco为设计师提供了一个便捷的处理该类问题的方法，可以直接在GH中调入与调出几何模型以及分析数据，从而高效的协同设计。

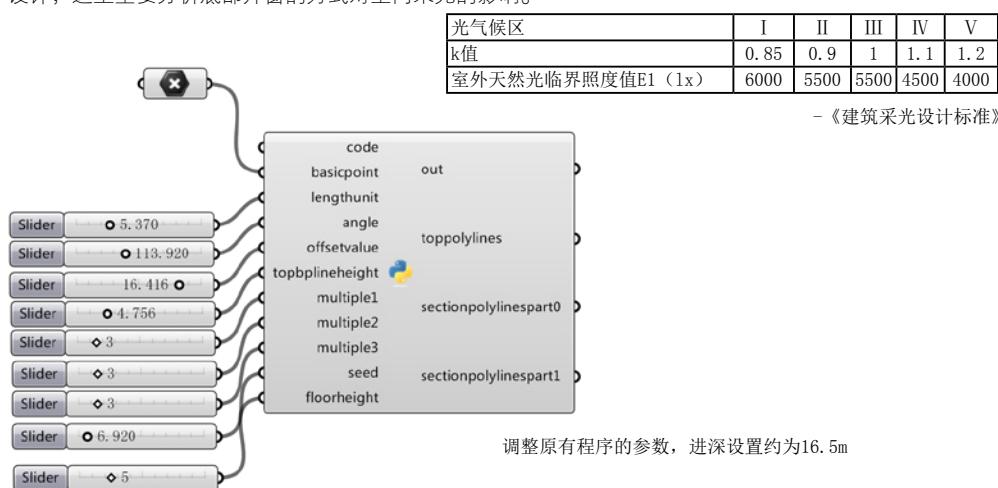
注：学习协同部分涉及到的知识，最好的方法就是对该部分阐述的内容，尤其包含的专业知识（部分内容在另外两个分册）有所了解之后，逐步的分析各个GH或者Python程序：

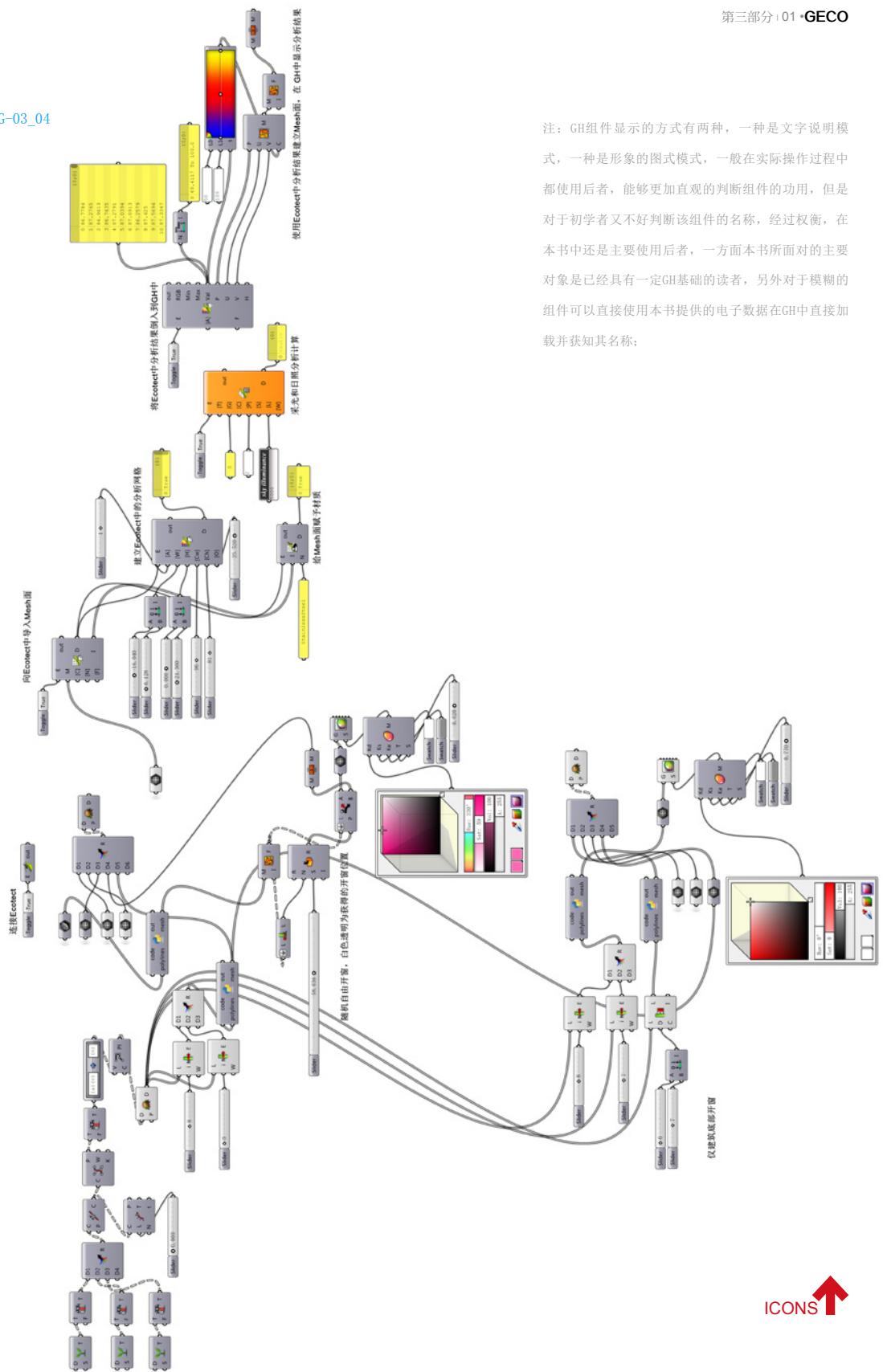


采光和照明分析：

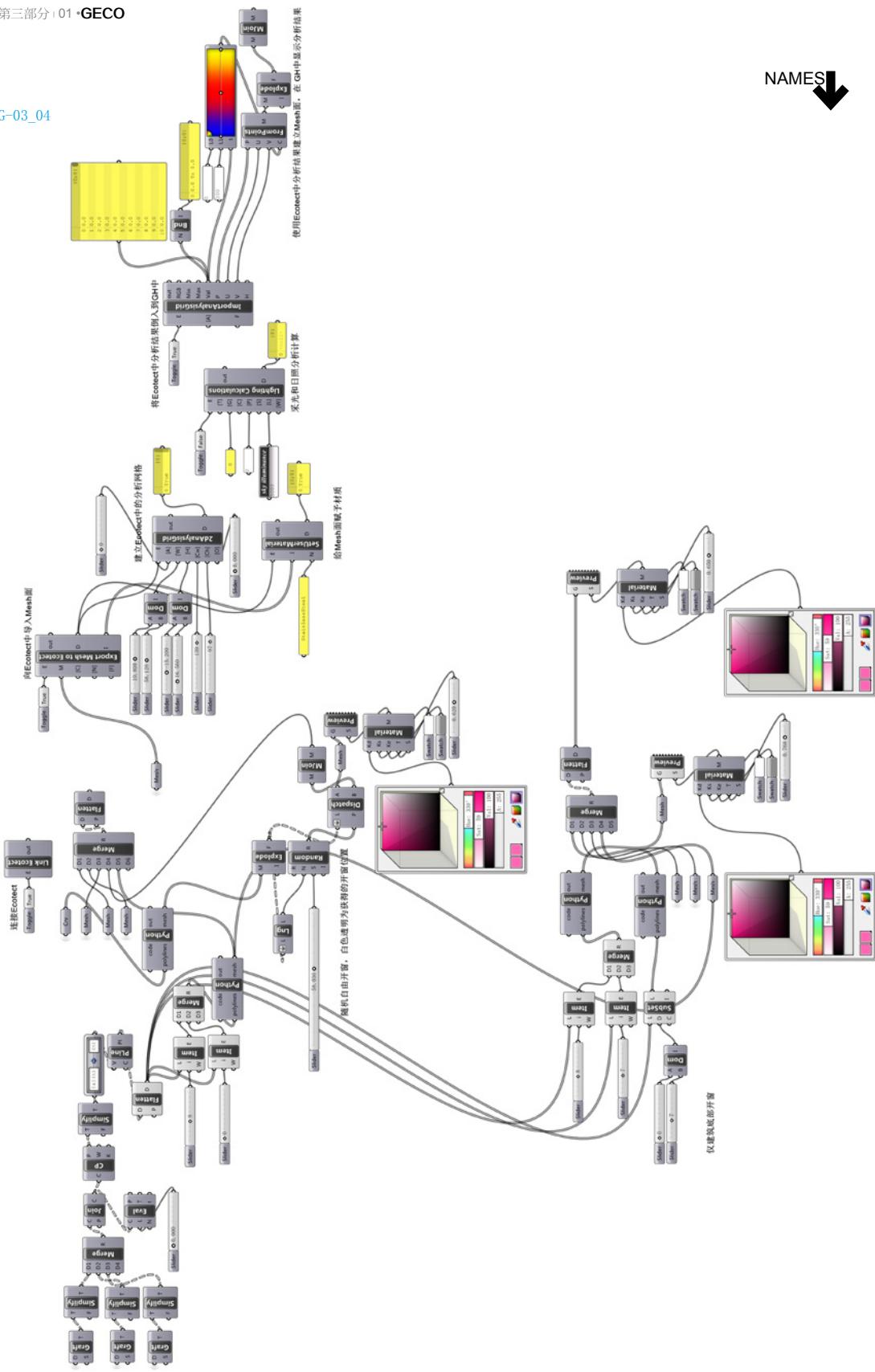
采光和照明分析，首先根据《建筑采光设计标准》查找分析所需要的相关值，这里使用北京的室外天然光临界照度值，北京位于中国光气候分区的三区，值为5000。

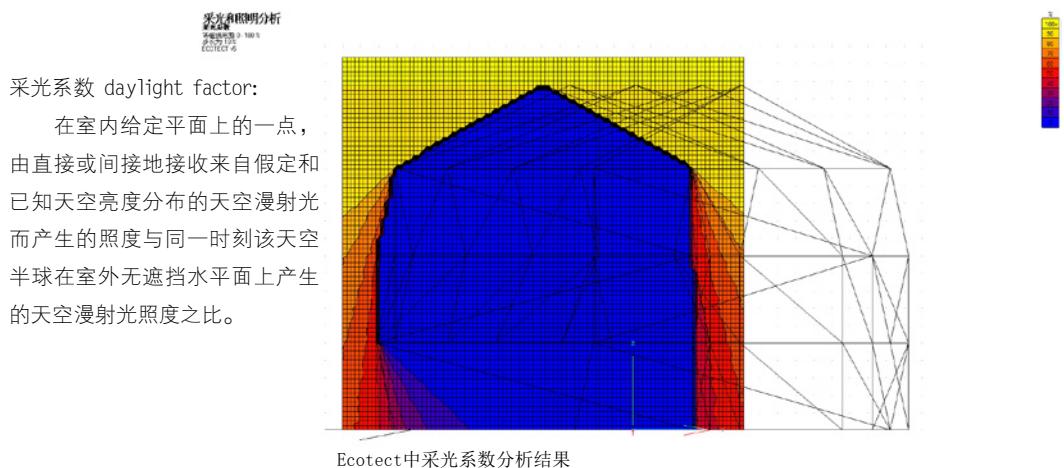
在GH中，使用Geco组件将Ecotect与GH关联，并将Eco中的分析结果调入到GH中从而直观的协助与实时的调整设计，这里主要分析底部开窗的方式对室内采光的影响。



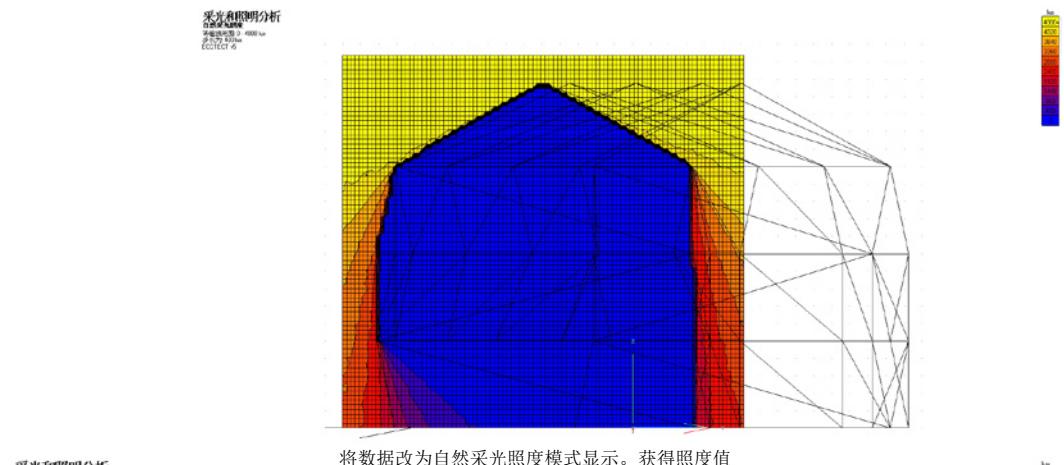


G-03_04

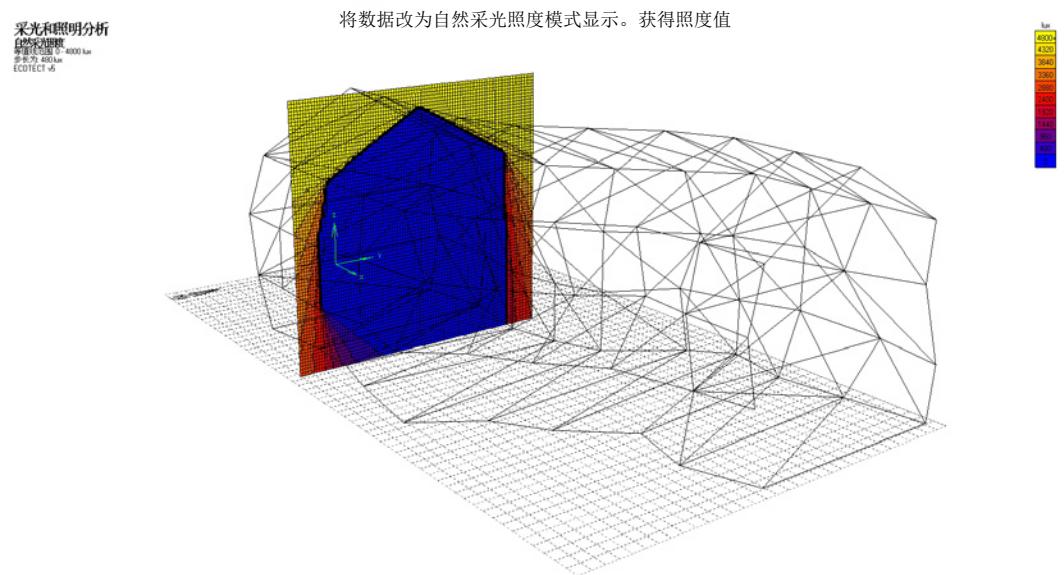




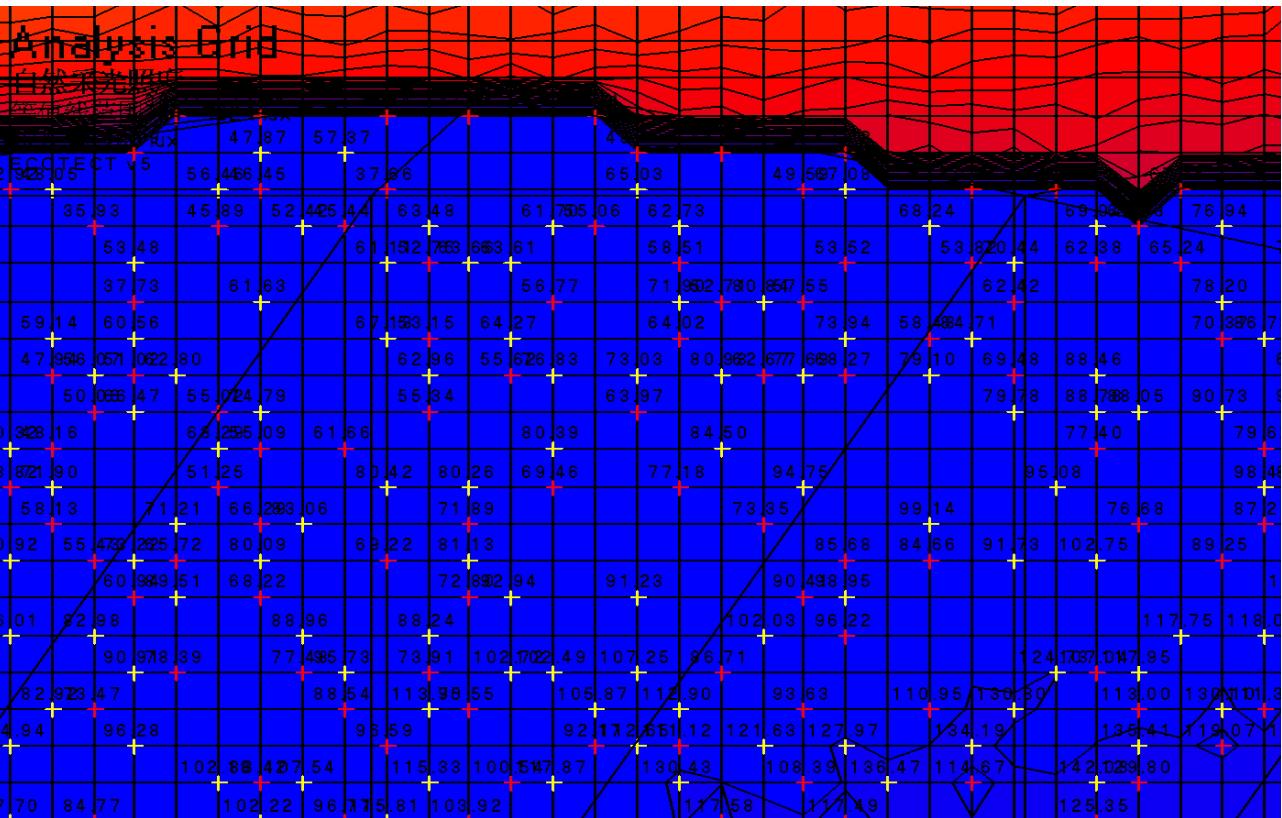
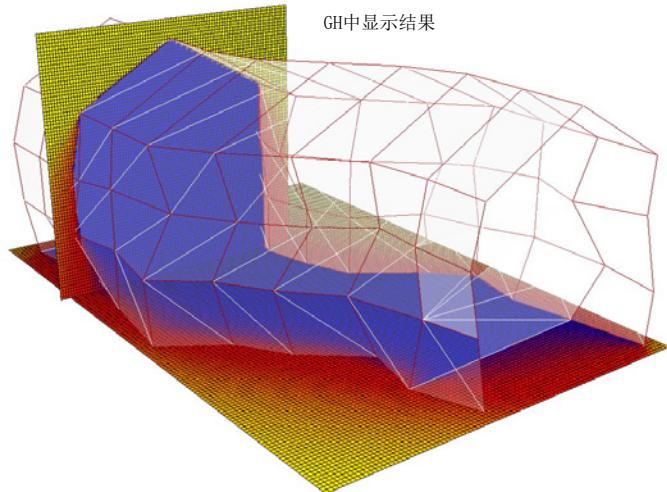
Ecotect中采光系数分析结果

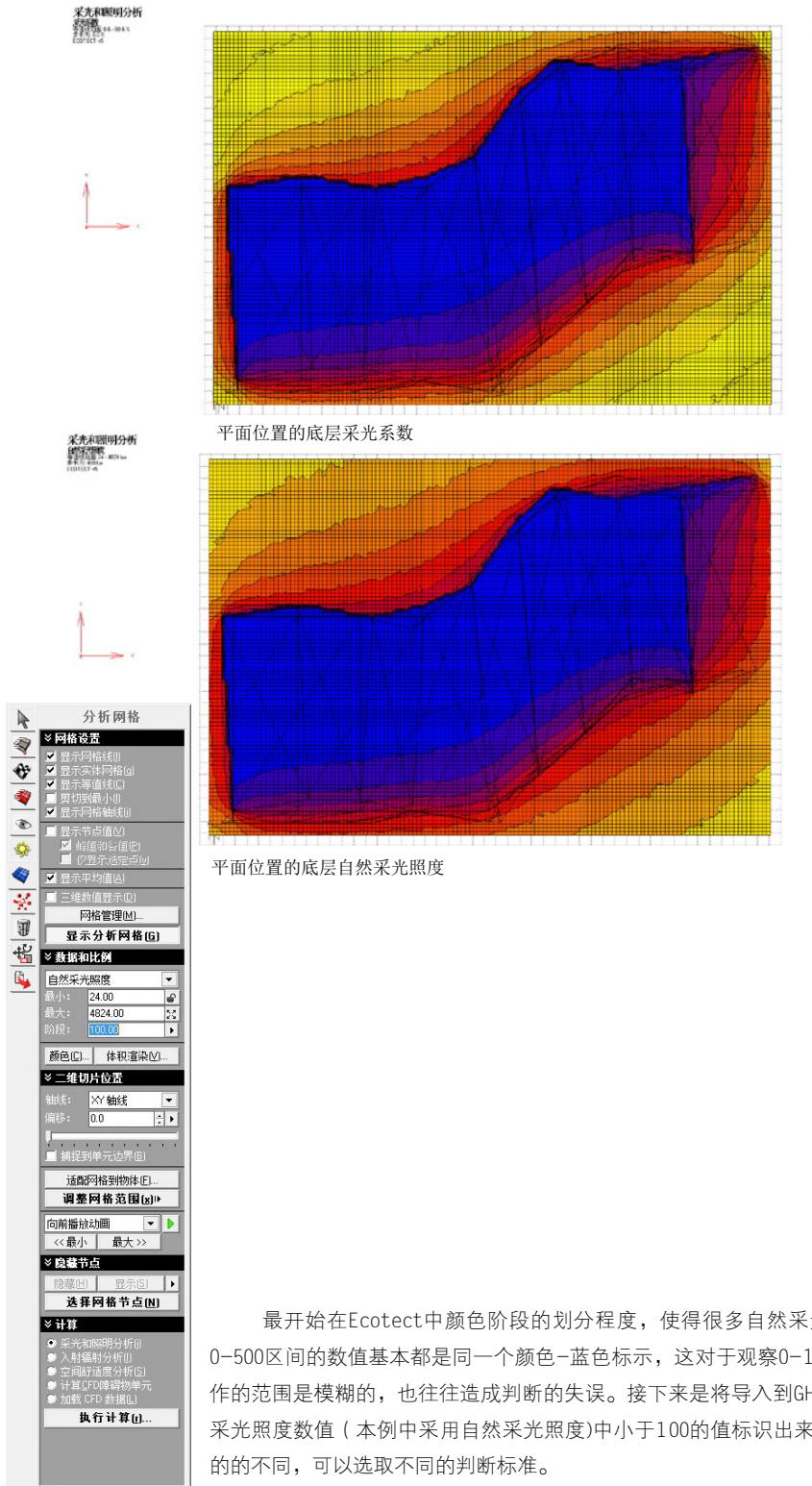


将数据改为自然采光照度模式显示。获得照度值



将Ecotect中的分析结果导入到GH中，协助设计师观察建筑室内的采光状况，结合分析结果，进一步修正设计。





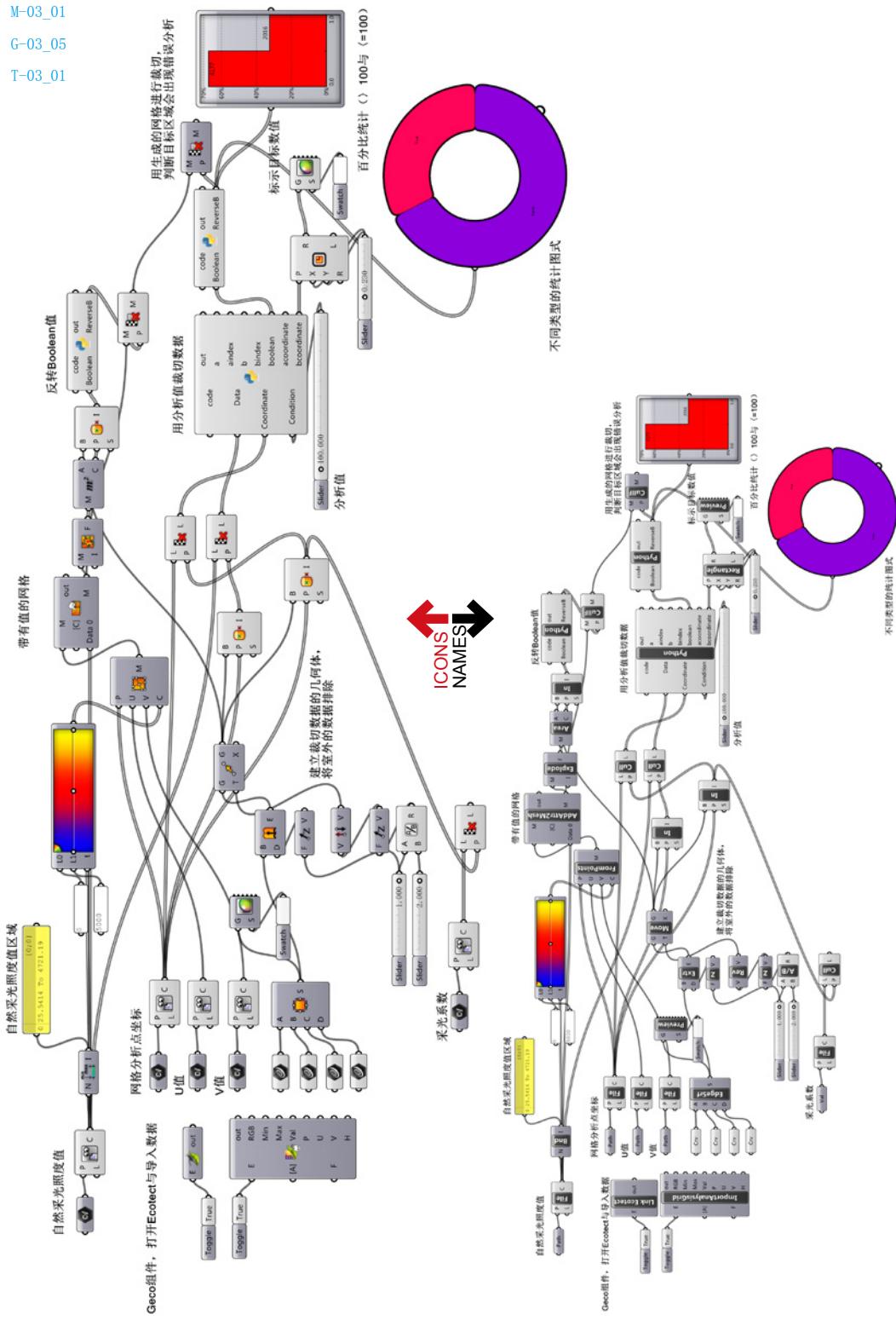
完成上述分析后，基本对分析结果如何影响设计有了初步的认知，就本例而言，仅底层一侧开窗的方式使得大部分区域采光受到限制，但是，这并不意味着上述图中暗蓝色的区域就是完全受影响的区域，实际上，需要根据视觉作业工作面上的采光系数标准值对网格数据进行有目的细分，按照标准中视觉工作等级Ⅲ，最低照度为100lx，在GH中将所有小于改值的网格标示出来，并进行相关的统计。

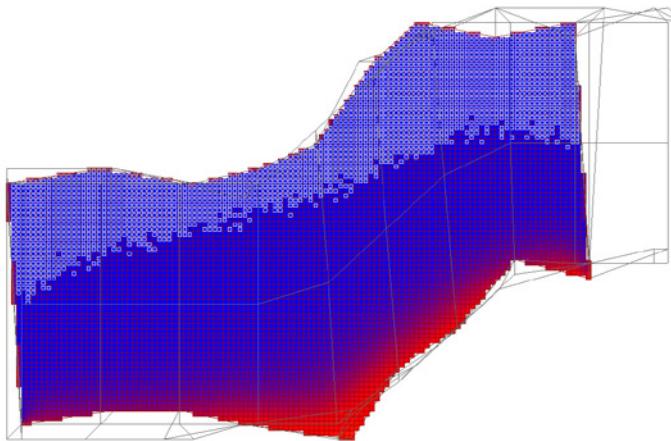
最开始在Ecotect中颜色阶段的划分程度，使得很多自然采光照度值，尤其分布在0~500区间的数值基本都是同一个颜色—蓝色标示，这对于观察0~100区域内不适宜精细工作的范围是模糊的，也往往造成判断的失误。接下来是将导入到GH中的采光系数或者自然采光照度数值（本例中采用自然采光照度）中小于100的值标识出来，当然根据分析设计目的的不同，可以选取不同的判断标准。

M-03_01

G-03_05

T-03_01





图中白色方块标示出的即为在GH中处理的结果，其值为小于100的自然采光照度值。

程序中的Python组件



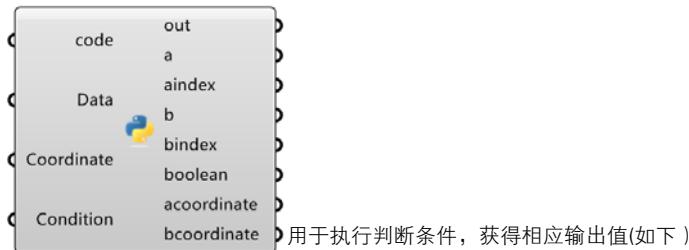
该组件反转Boolean值，将True转为False，将False转为True。

```
# sample script to show how to use this component and rhinoscriptsyntax
import math
import rhinoscriptsyntax as rs
```

```
p=Boolean

reverse=[]
for i in range(len(p)):
    if p[i]==False:
        p[i]=True
        reverse.append(p[i])
    else:
        p[i]=False
        reverse.append(p[i])
```

```
ReverseB=reverse
```



输入条件:

Date: 自然采光照度值或者采光系数;

Coordiante: 网格点坐标;

Condition: 判断条件 (数值);

输出值:

a, b: 为执行判断条件后的值 (本例为) 100和<=100值的序列);

aindex, bindex: 分别为a, b的索引值;

boolean: 标示a, b分别为True和False或者相反;

acoordinate, bcoordinate: 为a, b值所对应的网格点坐标;

sample script to show how to use this component and rhinoscriptsyntax

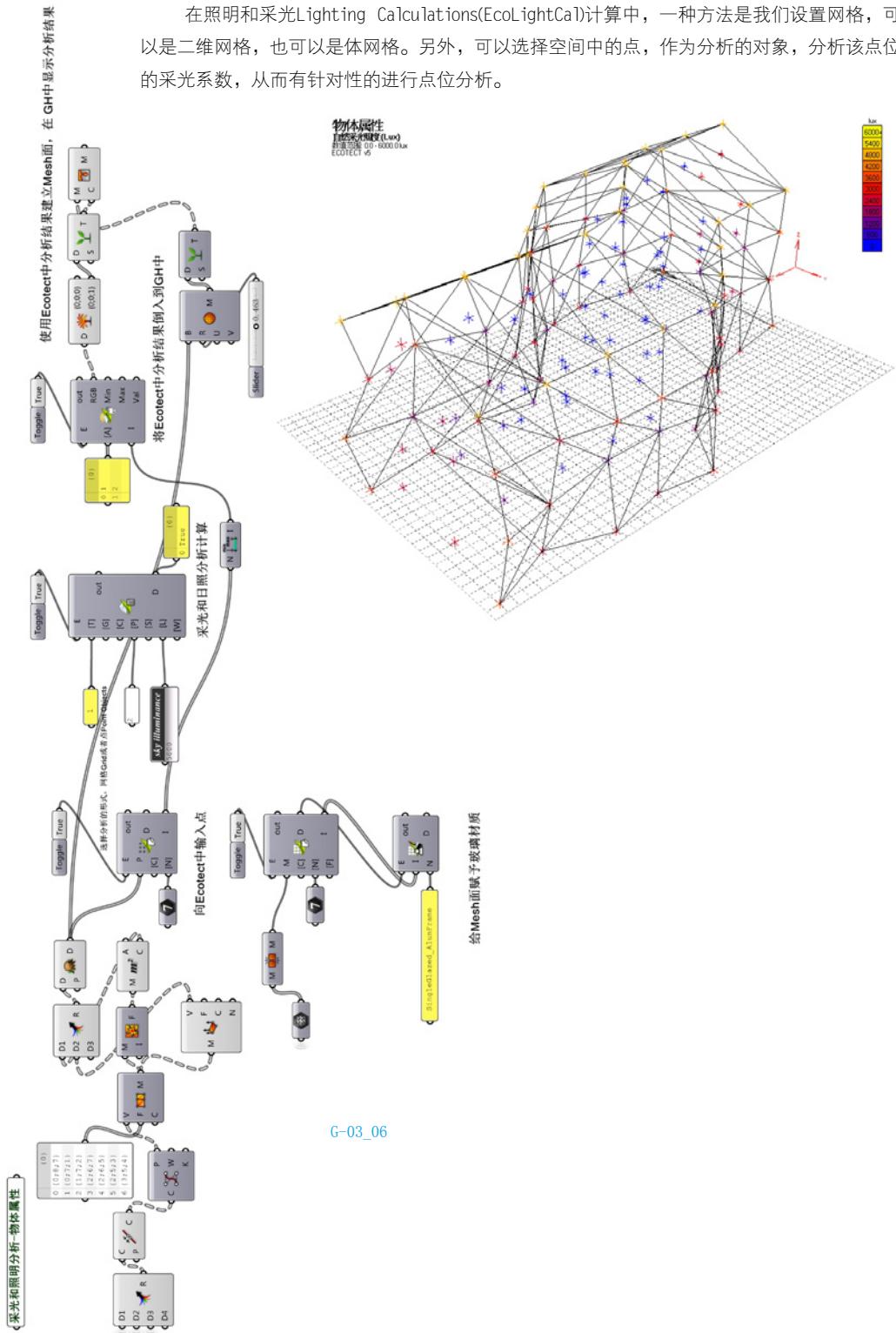
```
import math
```

```
import rhinoscriptsyntax as rs
```

```
inputdata=Data
coordi=Coordinate
condition=Condition
data=[]
acoordinate=[]
bcoordinate=[]
for n in range(len(inputdata)):
    data.append(float(inputdata[n]))

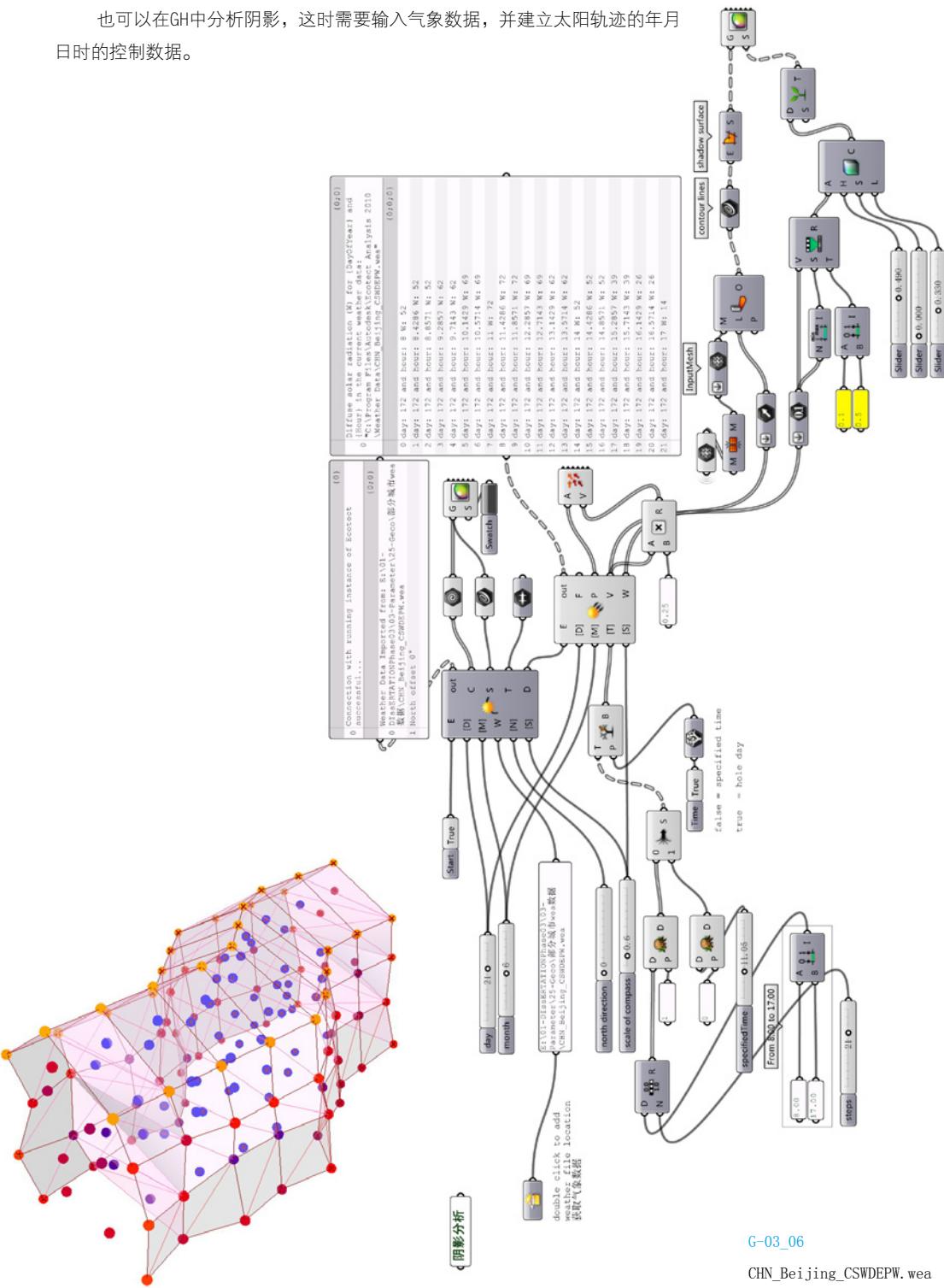
print(data)
a=[]
aindex=[]
b=[]
bindex=[]
boolean=[]
for i in range(len(data)):
    if data[i]>condition:
        a.append(data[i])
        aindex.append(i)
        boolean.append(' True ')
        acoordinate.append(coordi[i])
    else:
        b.append(data[i])
        bindex.append(i)
        boolean.append(' False ')
        bcoordinate.append(coordi[i])
```

在照明和采光 Lighting Calculations(EcoLightCal)计算中，一种方法是我们设置网格，可以是二维网格，也可以是体网格。另外，可以选择空间中的点，作为分析的对象，分析该点位的采光系数，从而有针对性的进行点位分析。



GH中点位的采光系数表示

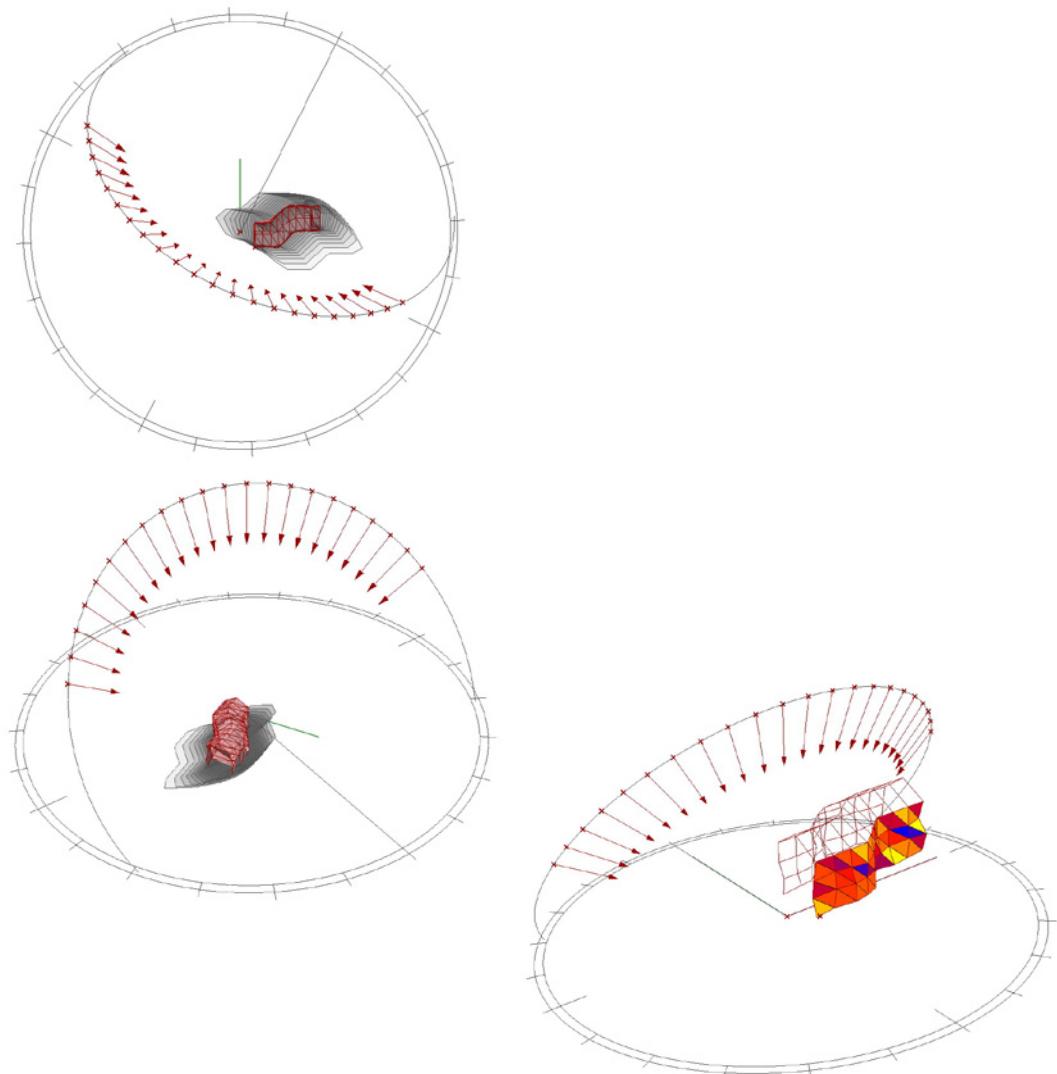
也可以在GH中分析阴影，这时需要输入气象数据，并建立太阳轨迹的年月日时的控制数据。



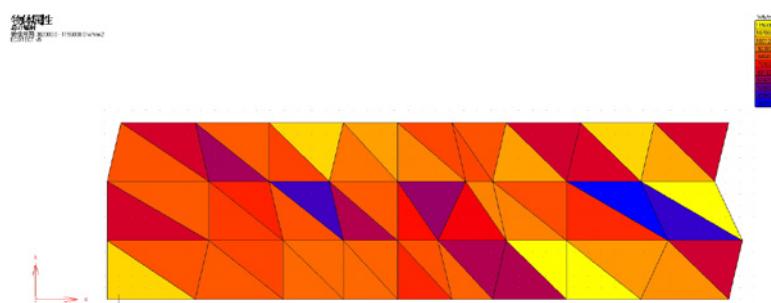
G-03 06

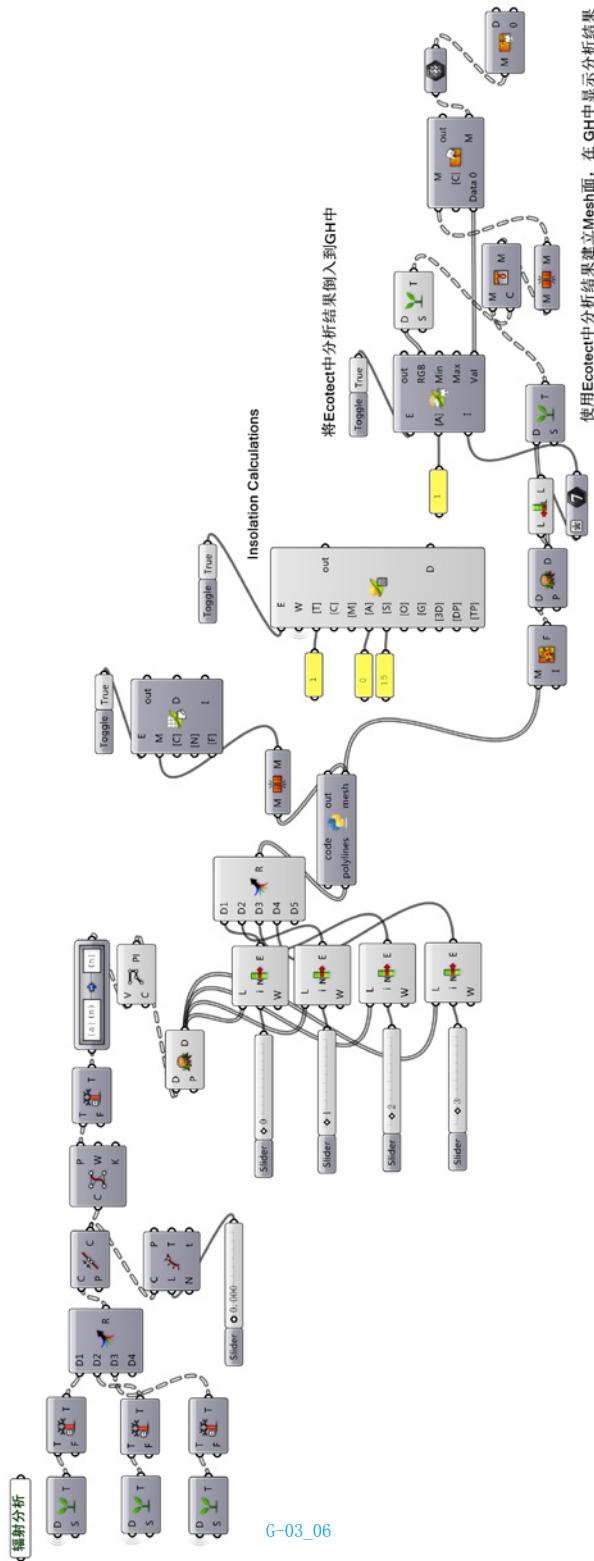
CHN_Beijing_CSWDEPW.wea

图中为夏至日6.21时的阴影，建筑的朝向的分析，只需要调整太阳轨迹北向即可。使用Weather Tool工具分析北京太阳辐射强度的最佳朝向为142.5°（具体阐述在生态部分）。



可以使用Geco结合Ecotect进行太阳辐射的计算，这里选择了墙面作为分析研究的对象。





将Ecotect中分析结果倒入到GH中
使用Ecotect中分析结果建立Mesh面，在GH中显示分析结果

2 地理信息与GH

另外两个分册主要叙述的是有助于设计顺利发展的各项数据的获得与分析，上一章节GECO部分则从生态分析出发，在模型构建的时候能够将生态辅助设计技术中涉及到的分析调入到GH的模型构建空间中，达到协同设计的目的。地理信息系统的设计相关部分同样可以将其分析数据导入到GH模型空间中，达到更直接的辅助设计目的。

不管是生态辅助设计技术还是地理信息系统的设计相关，绝大部分分析的结果都是基于栅格数据Mesh面的一种数据表达，甚至分析过程本身就是从Mesh面出发的。它是将空间划分为一定数量的网格，将各种信息的数据，例如高程、坡度、坡向、水文、人口、容积率等等以划分的单元网格为定位进行存储，并以颜色或者网格拉伸等方式直观的显示数据的结果，从而协助设计师对场地的属性有更直观的认知。

同时，模型构建所使用两种面的方式，基于NURBS的曲面和Mesh (Polygon) 面，其中Mesh面和上述分析所使用的栅格数据Mesh面基本是同一的，这就为分析数据和模型构建建立了联系，也就是说在GH中可以使用其Mesh面来显示各项分析结果，从而可以与模型构建同在一个平台进行处理。因此，有必要对Mesh面与其对数据信息的管理和分析进行叙述。

栅格数据（Mesh面Quad类型）

栅格数据模型是一个由大小相同像元组成的矩阵，这个矩阵作为整体用来描述专题数据、光谱数据和照片数据。栅格数据可以描述任何事物，从地表属性数据如高程和植被数据，到卫星影像数据、扫描地图和照片数据等。栅格数据通常有两种类型的栅格数据：专题数据和影像数据。专题数据可以被用于土地利用分析；影像数据可以用作其它地理数据的基准地图（Basemap），或用于生成专题数据。

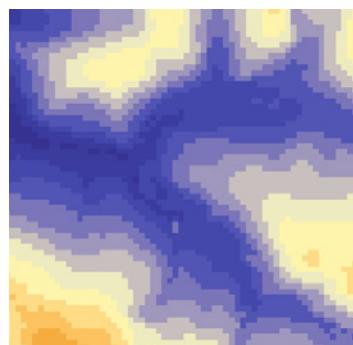
专题数据：

栅格数据的每个栅格像元（或象素）的值可以是一个测量值或分类值。把它绘制出来，这些栅格地图就是专题地图。

- 连续数据

这种栅格像元的值可以表示一个测量值如高程、污染物浓度或降雨量。从一个栅格像元到另一个栅格像元的值是逐渐变化的，从整体上说，这些值可以模拟一些类型的表面。

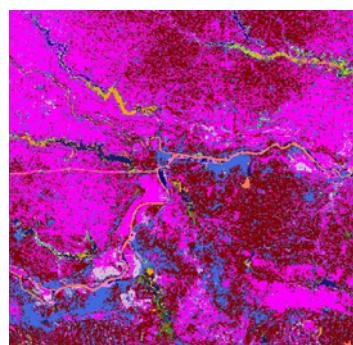
空间连续数据的栅格像元值表示的是在栅格像元中心的采样值。



- 离散数据

栅格像元值可以表示分类数据，比如土地的所有权类型，或植被类型。一个栅格像元与另一个栅格像元的值之间可以是一样的，也可以是有显著变化的。这类数据表现了具有相同值的区域，比如土地利用图或森林分布。

空间离散栅格数据像元描述的是整个区域像元的分类特征。



影像数据

影像栅格数据的特点是它是由卫星和飞机的成像系统获得的。

- 光谱和照片数据

成像系统记录的栅格数据是根据光在一个或多个电磁光谱波段上的反射率而获得的数据。

照片数据通常捕捉的是光谱中的红、绿、蓝部分，卫星影像会捕捉更多地波段，可以用于分析表面的地质和植被情况。

栅格数据模型



栅格数据是由栅格像元组成。一个栅格像元描述的是地球的某个特定区域，每个栅格像元大小都是一样的，比如一平方米或一平方米英里。每一栅格像元都有一个值表示的是某一位置的光谱反射信息或其它特征信息如土壤类型、人口普查值或植被类型。栅格像元的其它信息可以储存在属性表中。一个研究区域的栅格像元的大小取决于分析所需的数据分辨率。为了获得更详细的信息，要求栅格像元尽可能小，为了使计算机存储和分析更加高效要求栅格像元尽可能大。对于关键变量相似性越大的区域，比如地形地貌和土地利用，满足一定精度所需的栅格像元的尺寸可以大些。

栅格像元属性

一个栅格像元的值可以是在栅格像元位置处的类别、组别、类型或测量值。栅格像元的值是数值型：整型或浮点型。

具有相同值的栅格像元属于相同的Zone，Zone的栅格像元不一定是相连的。

当栅格像元的值是整数时，它可能是很复杂标识系统的一个编码。例如，在一个土地利用栅格上的一个家庭居住处的代码值是4，与这个数值4相关联的可能是一系列的属性，比如平均的商业价值、平均的居住人数，或者是人口普查代码等。

栅格像元的值和具有这种值的像元数量之间通常是一对多的关系。比如，在表示土地利用的栅格数据中可能有400个像元的值是4（4表示单个家庭的居住地），而有150个像元的值是5（5表示的是商业区）。

在栅格像元中，代码值会出现很多次，但在属性表中只出现一次，属性表中存储的是这个代码的其它属性值。这种设计降低了存储量并简化了更新过程。一个属性的单个变化可被应用到几百个具有相同代码值的像元。

数据类型

每个栅格像元都有一个值。栅格像元的值可以表示以下四种数据类型中的一种。

- 名词性数据 (Nominal data)

名词性数据的值能把一个实体同另一个实体区分开来。这些值能确定栅格的组别、类别、成员或分类，用它们将栅格像元的位置与地理实体联系起来。这些值是定性的，而不是定量的，与一个特定点或一个线性尺度没有关系。为土地利用、土壤类型等编码时应该用名词性数据。

- 序列数据 (Ordinal data)

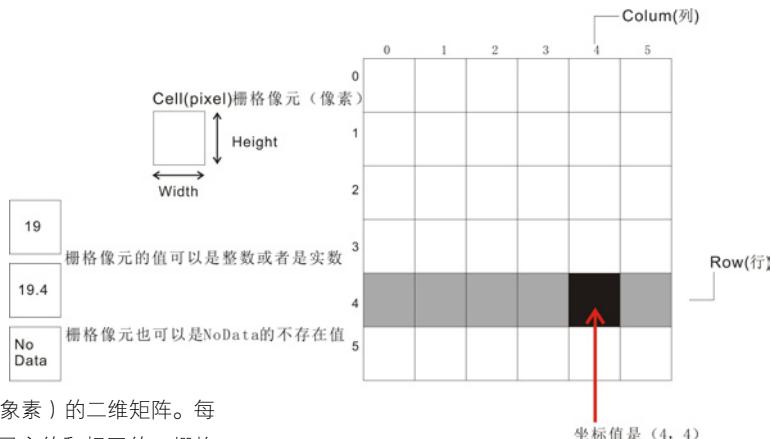
序列数据能确定一个实体相对于其它实体的等级。这种度量方法表示的是次序，如第一、第二或第三，但是它们不能确定数量或相对比例。你无法从中发现数量差别，比如一个实体相对于另一个实体有多大，多高或多密？

- 间隔数据 (interval data)

间隔数据的值表示的是对尺度的一种度量，如一天中的时间、华氏温度以及PH 值等。这些值都是从标定的刻度尺上读出来的，而不是相对于一个真实的零值点。你可以对间隔数据进行相对比较，但当与该尺度的零点进行比较时是没有意义的。

- 比值数据 (ratio data)

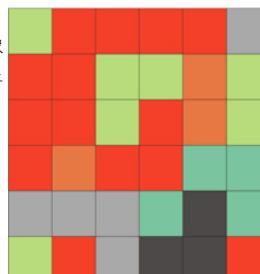
比值数据的值表示的是从一个有固定和有意义的零点的标尺上获得的数据。可对这些数据进行数学运算，能获得可预测的和有意义的结果。比值数据的例子有年龄、距离、重量和体积等。



栅格数据是栅格像元（或象素）的二维矩阵。每一个栅格像元的高度和宽度都是固定的和相同的。栅格数据覆盖的是一个矩形区域。

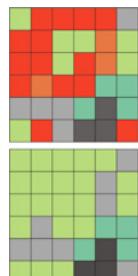
每个栅格元都有一个值，这个值可以表示一个区域的很多特征，包括反射率、颜色、降雨量和高程。栅格数据有一个整数坐标空间，可以确定一个栅格像元的坐标，从左边开始计算列数，从上边开始计算行数。行和列的开始值都是零。

具有整数值的栅格数据可以用一个附加属性表来定义，它记录每一个特定像元值的属性，并可以为属性表增加属性字段。



属性表			
Value	Count	Type	Code
23	5	樱花	400
29	3	国槐	410
31	3	海棠	420
37	4	木兰	500
41	14	木槿	510
43	7	白杨	600

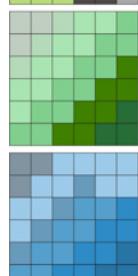
栅格数据分类



樱花
国槐
海棠
木兰
木槿
白杨
非常好
好
适度
贫瘠

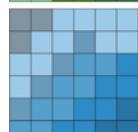
名词性数据是包含名称的分类数据。数据值是一个任意的类型代码。如土壤类型和土地利用类型；

序数数据是分类数据，有名称而且其值是数字等级。比如土地适宜性分类和土壤排水分类；



700-709
710-719
720-729
730-739
740-749
750-759

间隔数据是有数字顺序的，其间隔差别是有意义的。比如电压或浓度差。



0.0-10.0
10.1-20.0
20.1-30.0
30.1-40.0
40.1-50.0
50.1-60.0

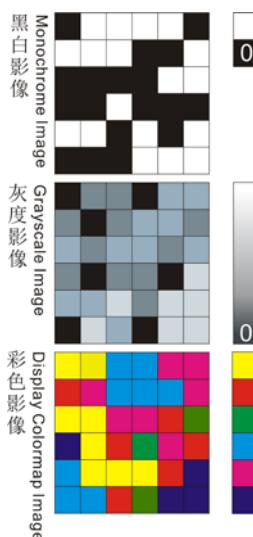
比值数据度量的是一个连续性现象，有自然的零值，比如降水和人口。

名词性和序数数据描述的是离散分类。它们最好用整数来描述栅格单元值。

21	17	18	22
18	16	17	19
21	18	22	29
14	16	19	20

间隔数据和比值数据表示是连续现象，通常用实数栅格单元值来度量。

21.1	17.3	17.2	18.1
18.5	16.2	17.3	19.4
21.0	19.1	19.4	19.2
26.3	21.6	20.5	16.5



在黑白影像中，每个栅格像元的值是0或1，它们通常被用于由简单线组成的扫描地图，如地界（Parcel）地图。

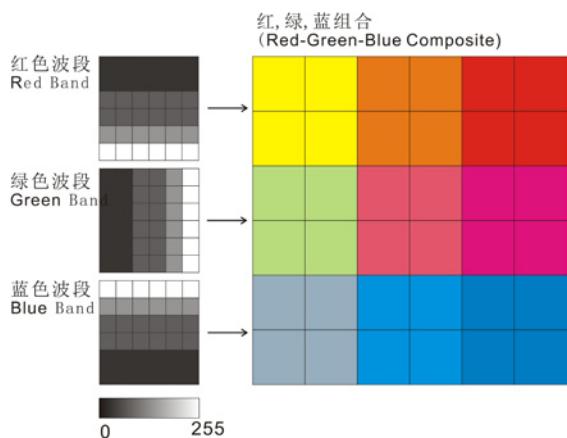
在灰度影像中，每个栅格像元的值范围为0到255，它们通常被用于黑白的航空相片。

颜色图 Colormap

Red Green Blue

255	245	0
218	37	29
0	146	63
0	147	221
221	19	123
40	22	111

描述影像颜色的一种方法是具有色彩映射表的图象。栅格像元值被编码并使之与相应的RGB值对应。



每个波段的属性值范围是0 到255

栅格数据集有一个或多个波段。在一个多波段的数据中，每一个波段描述的是由传感器收集的一段电磁光谱的信息；

多波段栅格数据通常用红绿蓝合成色表示，这种波段设置是由于它可以直接在计算机上显示，这种显示采用的是红绿蓝彩色模型。

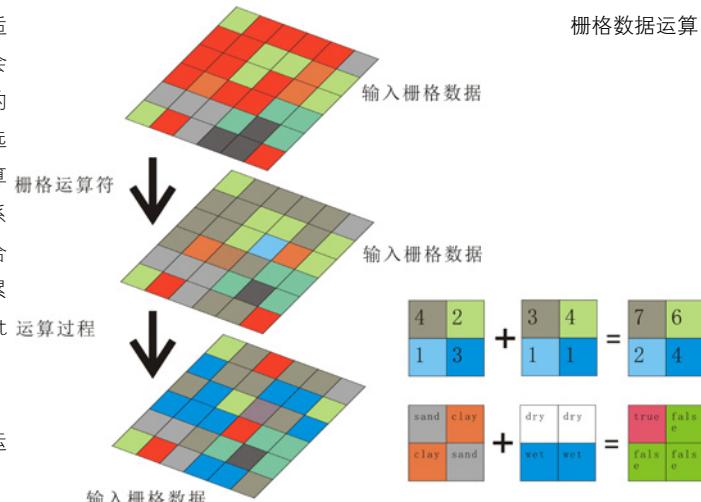
在研究一个区域时，可能要进行适宣性分析。为了做到这一点，可能会选择带有降水、土壤碱度、日照值的栅格数据，并根据研究使用的方程选择一系列的运算符。运算符可以是算术型arithmetic、布尔型boolean、关系型relational、位逻辑型biwise、组合型combinatorial、逻辑型logical、累积型accumulative和赋值型assignment 运算过程等。

• 地图运算

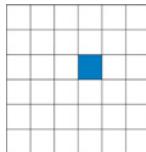
可以对两幅栅格图象进行数学运算，并将结果输出在另一幅栅格图中。方程包括加、减、乘、除、对数、指数、正弦、余弦和平方根。

• 地图查询

可以对两幅图象应用布尔和逻辑运算符创建一幅输出结果为true/false 值的栅格图。操作幅包括And, Or, XOr, Not, >, >=, =, <, <, 和=.。

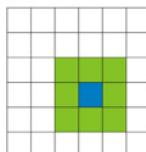


栅格函数



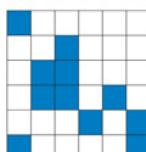
局域函数Local Function

局域函数一次只对一个栅格像元进行计算，邻近的栅格像元不会影响结果。该函数可以被应用到一个或几个叠加的栅格图象上。局域函数包括三角函数、指数、对数、重新分类、选择和统计函数；



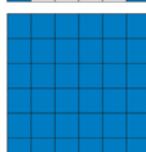
邻域函数Focal Function

邻域函数进行的是单个栅格像元与它的相邻像元之间的计算。相邻的范围可以是矩形、圆形、环面或楔形。这些函数的结果可以是均值、标准差、总和或在相邻范围内的栅格值的变化范围；



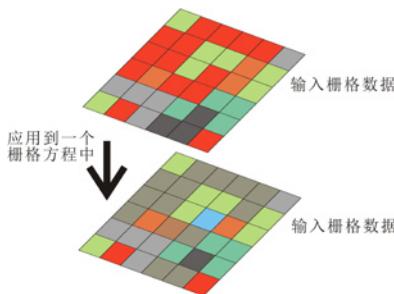
区域函数Zonal Function

区域函数是对Zone进行计算，Zone是指具有相同值得栅格像元。这些像元组成的Zone可以是不连续的。有两种类型的区域函数：统计和几何。这些函数包括面积，质心，直径，范围和总和；



全局函数Global Function

全局函数对这个栅格图象整体进行计算。例如计算欧氏距离，加权距离以及流域面积计算等；



有很多栅格函数。每个栅格函数可以有一个或多个栅格图像作为输入图象，并生成一个或多个输出栅格图象；

TIN表面模型 (Mesh面Triangle类型)

表面模拟的方法有两种，一为栅格，一为TIN。栅格是应用样点值或插值获得的z值采用规则的格网来模拟表面。TIN 采用不规则三角网上的一系列位置分布不规则的点来模拟表面，在每个点上有一个Z 值。

栅格和 TIN 在表面模拟方面都有各自的优点；可利用的源数据情况、分析的范围和支持的绘图方法等将决定采用哪种方法更为合适。

• 用栅格方法模拟表面

栅格方法是用带有 Z 值的具有统一步长的规则的格网来模拟表面。可以用相邻点之间插值的方法估计表面任一位置的值。格网的分辨率（栅格单元的高度和宽度）决定了栅格表面的精度。

栅格是最常用的模拟表面的方法，因此高程值广泛采用的是这种方法，并且代价较低。栅格表面的一个典型例子是由美国地理测绘局生产的数字高程模型 (DEM, digital elevationmodel)。

栅格模型支持大量丰富的空间分析，比如空间一致性分析、邻近分析、离散度分析以及最低成本路径分析等，这些分析执行速度都很快。

栅格表面模型的缺点是表面的不连续性，比如无法很好地表示山脊，以及不能精确地表示一些要素如山峰等。

栅格模型适合于那些位置精度要求不高以及表面特征不需要表示得太精确的小比例尺的制图应用。

• 用 TIN 方法模拟表面

TIN 表示的是彼此相邻不重叠的三角形面组成的表面。通过在一个三角形表面对高程数据进行简单或多项式插值，可以估计任何位置的表面值。因为 TIN 中高程数据是通过不规则采样获得的，可以在地形变化剧烈的区域应用可变的点密度，来生成一个高效精确的表面模型。

TIN 保存着表面要素的精确位置和形状。面状要素比如湖泊和岛屿用一个封闭的三角形边组成的多边形表示。线性要素比如山脊用一组相连的三角形边表示。山峰用三角形的顶点表示。

TIN 支持很多的表面分析，如计算高程、坡度、坡向、进行体积计算、创建剖面图等。

TIN 的缺点是经常不容易获得所需的数据集。

TIN 很适合那些对表面要素的位置和形状有很高的精度要求的大比例尺的制图应用。

TIN数据结构能够精确地表示任何类型的表面。在TIN表面中，不仅能够通过插值获得任何位置的高程值，而且能模拟在表面坡度中形成突变(Breaks)的自然要素，比如山脊、河流。

TIN 的定义：

术语不规则三角网 (Triangulated Irregular Network) 是对 TIN 特征简练的描述。

“三角形化”指的是用一些点形成一系列优化的三角形。三角形能很好地描述一个表面局部区域，因为含有Z值的三个点能在三维空间中唯一确定一个表面。

“不规则”表现了TIN进行表面模拟的一个主要优点，即模型在模拟那些表面变化起

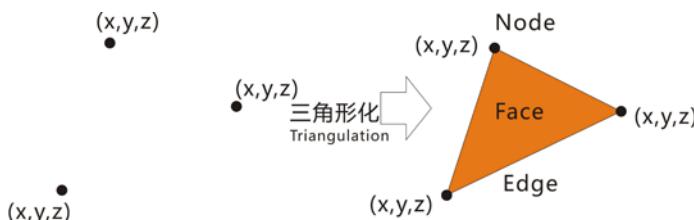
伏较大的区域时，样点的采集密度可以是变化的。

“网络”是指隐含在 TIN 中的拓扑结构。这种结构能进行复杂的表面分析，也能以压缩的方式来表示表面。

创建 TIN

TIN 由大量的点组成，这些点含有从不同来源采集而来的高程值。TIN 通常与摄影测量设备有密切关系，通过这些设备，能精确获得在一个立体模型中来自一对航空相片的同一样点值。TIN 也可以用下列数据生成，如调查数据，数字化的等高线，带有Z 值的栅格数据，数据文件或数据库中的点集，或者由其它TIN 运算而来。

从这些输入点中，便可以进行三角形化的工作。在TIN中，这些三角形被称为面（Face），形成三角形的点被称为结点（Node），三角形的边被称为边（Edge）。

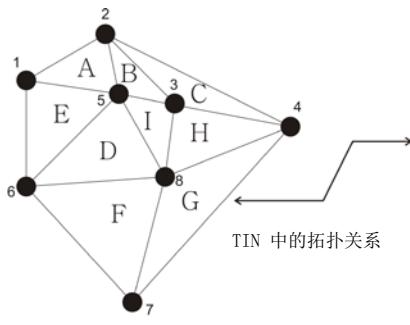


TIN 中的每个面都是三维空间表面的一部分。TIN 中的所有这些面通过每个结点和每个边精确地与相邻的面相连。面和面之间彼此不会交叉。

栅格表面模型

TIN表面模型

表面模型的精度	栅格表面的精度由栅格像元的大小决定。为了增加栅格表面的精度，整个栅格表面必须在高分辨率下重采样；	TIN 表面模型具有随坡度变化而变化的点密度。为了使 TIN 模型更精确可以增加点的数量，也可以增加断线以及多边形；
表面要素的逼真度	栅格模型在规则的格网上对表面要素的 Z 值进行采样。要素如山峰、山脊不能被定位到某一点上，它们的精度不会高于格网分辨率；	TIN 可以描述表面要素如河流、山脊和山峰。这些要素用精确的坐标存储，坡度不连续的地方如山脊，用断线（breakline）来模拟；
表面分析	空间一致性分析、邻近分析、离散度分析以及最低成本路径分析；	高程、坡度、坡向计算从表面生成等高线体积计算，垂直剖面分析，视线分析；
模型应用	小比例尺的表面分析和模拟。污染物扩散的模拟，流域的确定，洪水灾害的水文分析；	道路设计的体积计算。土地发展的排水系统研究，高质量等高线的生成，对某建筑物的视场模拟；



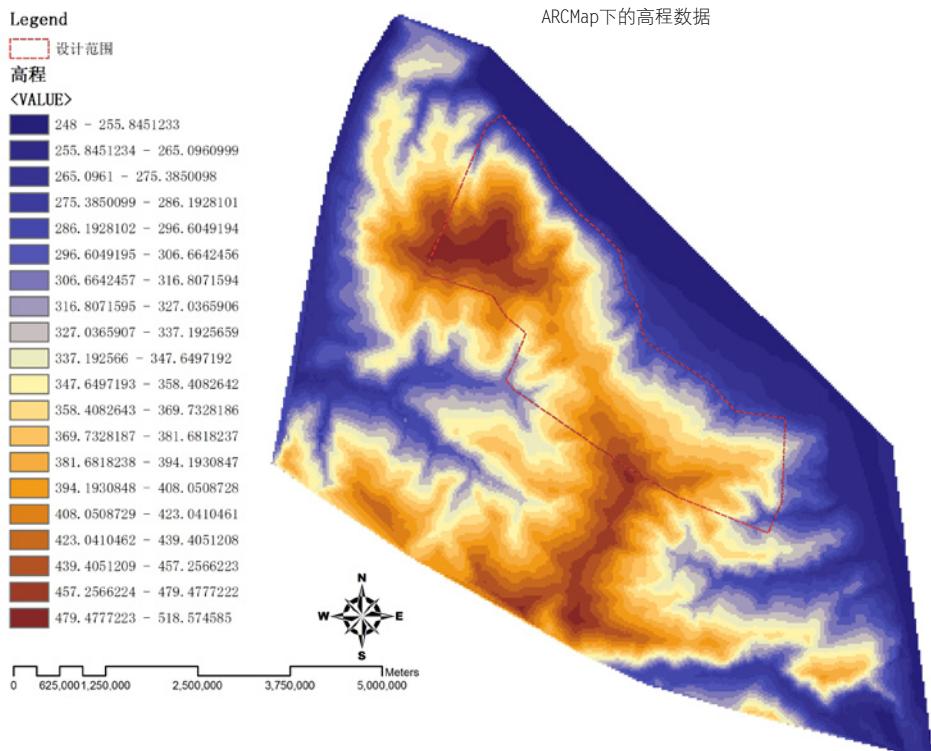
TIN 是一个具有拓扑关系的数据结构，因为它管理的是包含每个三角形节点信息以及与每个三角形相邻的三角形的节点信息。

Triangle	Node list	Neighbors
A	1,2,5	-,E,B
B	2,3,5	A,C,D
C	2,3,4	-,B,H
D	5,6,8	I,E,F
E	1,5,6	-,A,D
F	6,7,8	-,D,G
G	4,7,8	-,H,F
H	3,4,8	I,C,G
I	3,5,8	B,D,H

三角形总是有三个结点，通常情况下有三个相邻三角形。在 TIN 外围的三角形可能只含有一个或两个相邻三角形。

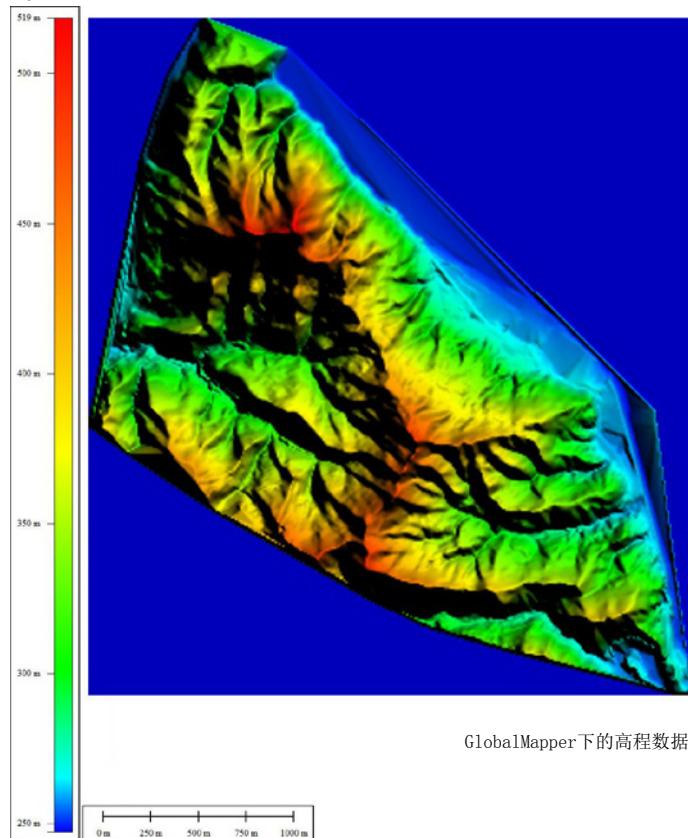
地理信息数据的调入

在 GIS 平台可以处理空间分析，例如栅格数据的统计分析、重分类、栅格计算等，三维分析例如坡度、坡向、可视性，以及水文分析例如汇流累积量、河网的分级，这些在地理信息系统相关部分已经有所涉及，在模型构建部分，主要的目的是将分析的结果调入到 GH 模型构建空间中协同设计。



在ARCMAP平台下处理的高程数据，并没有根据实际研究的需要进行高程的重分类，是按默认情况下划分的高程等级。

可以在GIS中以脚本的方式输出数据为某种特定的文本格式，例如本例中所要使用的XYZ方式，即将各个栅格以xy坐标的方式定位，将高程值存储在z值中。这里使用GlobalMapper(GM)，作为数据转换的中间平台，先将ARCMAP中的高程数据右键/Data/ExportData 输出为TIFF格式文件，当然也可以选择其它在GE中可以打开的格式。在GE中以File/Export Elevation Grid Format/XYZGrid方式输出，为了减小数据量，这里将分辨率Resolution调整为X、Y方向均为50m。转换后的数据文件后缀名为xyz，这里将后缀名改为txt，可以使用文本打开。



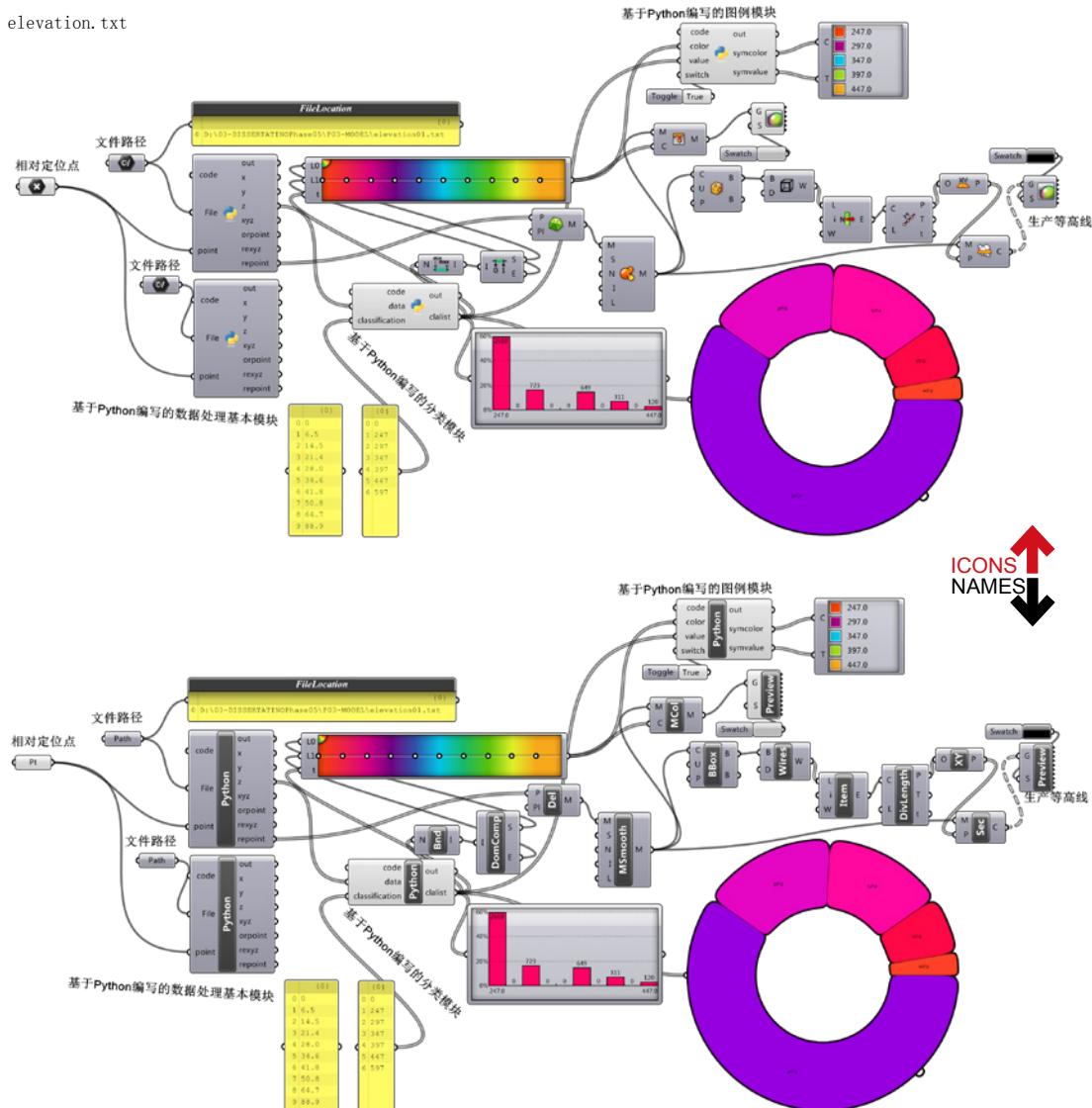
```

-93.7975895648, 32.2277380368, 247
-93.7975895648, 32.2276472463, 247.309
-93.7975895648, 32.2275564559, 280.73
-93.7975895648, 32.2274656654, 279.842
-93.7975895648, 32.2273748749, 277.584
-93.7975895648, 32.2272840844, 274.884
-93.7975895648, 32.2271932939, 275.211
-93.7975895648, 32.2271025034, 305.808
-93.7975895648, 32.2270117129, 306.095
.....
```

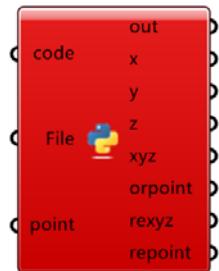
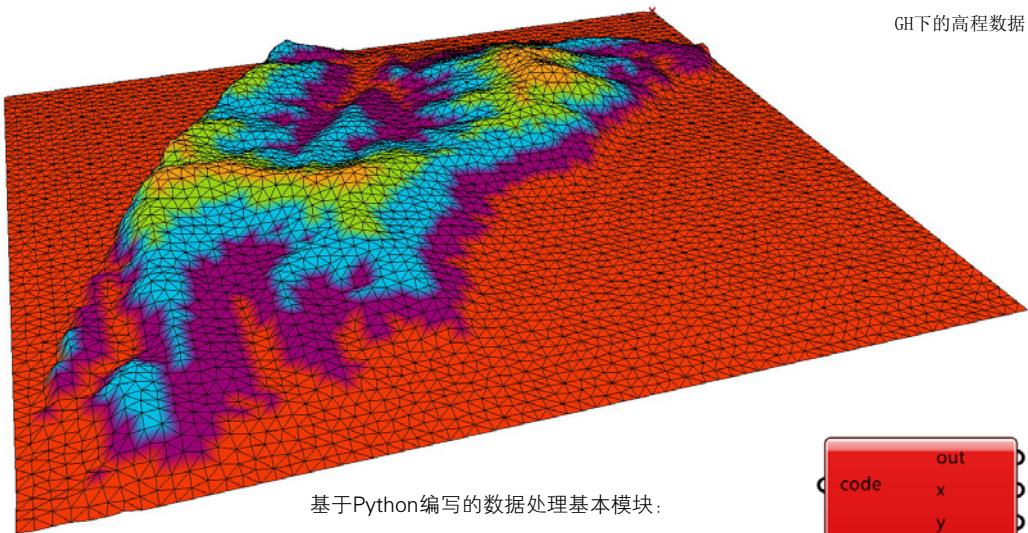
.txt格式的文本文件：可以看到数据格式每行以逗点隔开为x、y、z三个值，最后数字为z值，即(x, y)坐标点的高程。

G-03_07

elevation.txt



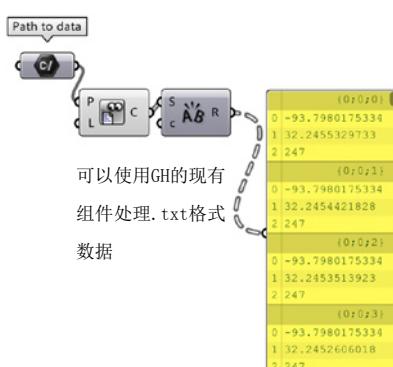
将.txt文本文件调入到GH中，使用Python编写文本数据处理模块，将.txt文件中的x, y, z数值分别以单独列表的方式输出，并同时编写了(x, y, z)坐标点的列表输出`orpoint`，以及为了控制图形的位置，编写了相对输出坐标点的列表输出`repoin`，将输出的数据在GH中生成Mesh面，并以z值即高程值为参考，以颜色标识高程重分类结果，最后编写图例部分，完成地理信息数据在GH中的调入与重分类。



```

import rhinoscriptsyntax as rs
import re
point=point#调入相对控制点
print(point) #编写过程中通过Print查看运行结果
pcoordi=rs.PointAdd (point,(0,0,0))#通过点相加的方式改变数据格式
print(pcoordi)
file=File
print(file)
f=open(file,'r')#以只读的方式打开.txt文件
x=[]
y=[]
z=[]
xyz=[]
rexyz=[]
firstpoint=f.readline()#以读取单独行的方式获得第一个点的坐标
repoin= []
orpoint= []
print(firstpoint)
subdistance=rs.PointSubtract (rs.AddPoint (firstpoint),rs.AddPoint
(pcoordi))
print(subdistance[0])
for line in f.readlines():#循环遍历.txt文件数据
    lst=re.split('[,]+',line)
    x.append(float(lst[0]))#X值输出
    y.append(float(lst[1]))#Y值输出
    z.append(float(lst[2]))#Z值输出

```



```

xyz.append((float(lst[0]),float(lst[1]),float(lst[2])))
#xyz坐标值输出
orpoint.append(rs.AddPoint(float(lst[0]),float(lst[1]),float(lst[2])))#xyz坐标点输出
rexz.append(((float(lst[0])-subdistance[0]),(float(lst[1])-subdistance[1]),\#相对xyz坐标值输出
(float(lst[2])-subdistance[2])))
repoint.append(rs.AddPoint((float(lst[0])-subdistance[0]),(float(lst[1])-subdistance[1]),\
(float(lst[2])-subdistance[2])))#相对xyz坐标点输出
print(orpoint)

```

基于Python编写的分类模块：



```

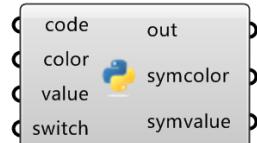
import math
import rhinoscriptsyntax as rs
data=data
cal=classification
print(cal)
for i in cal:
    cal[cal.index(i)]=float(i)#将文本值改为浮点数值模式
print(cal)
for k in range(len(cal)-1):#遍历输入列表，将同一分类区间值赋予相同值
    a=cal[k]
    b=cal[k+1]
    for q in range(len(data)):
        if a<=data[q]<b:
            data[q]=a
for w in range(len(data)):
    if data[w]>=cal[-1]:
        data[w]=0
clalist=data#输出分类列表数据

import math
import rhinoscriptsyntax as rs
color=color#输入颜色值
value=value#输入分类值
selectvalue=value
boolean=switch#设置开关，当数据量较大时可以先
输入False值，禁止运算
print(boolean)

def classification(color,value,selectvalue):
    color=color
    value=value

```

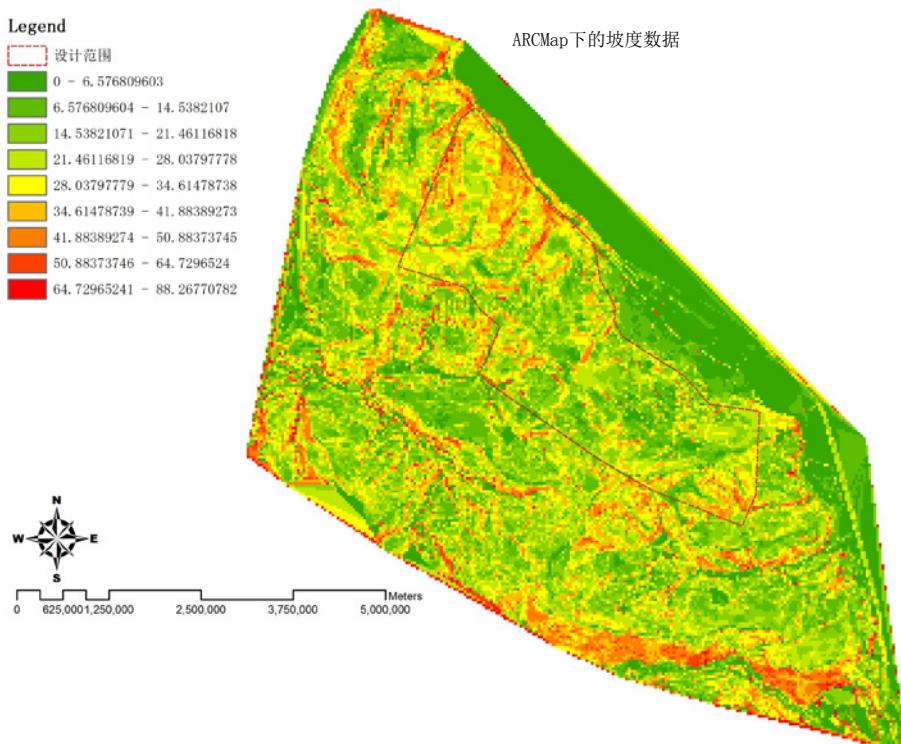
基于Python编写的图例模块：

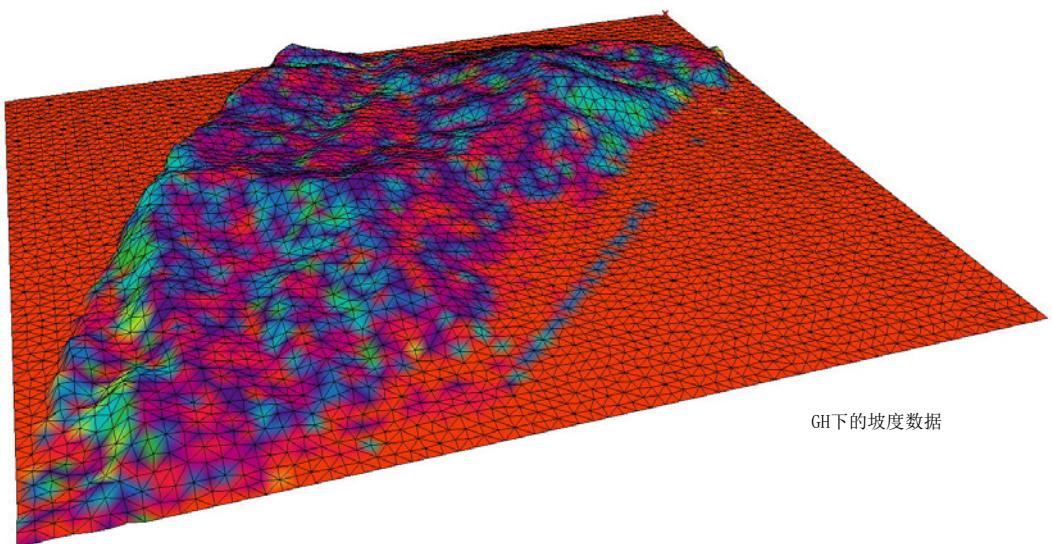


```

selectvalue=selectvalue
symcolor=[]
symvalue=[]
sym=[]
for m in value:
    for n in range(len(selectvalue)):
        if selectvalue[n]==m:
            selectvalue[n]=None#将相同的值赋予None，占据列表
            sym.append(m)
    for i in range(len(sym)):
        if sym[i]!=None:#排除None值
            symvalue.append(sym[i])
            symcolor.append(color[i])
return symvalue,symcolor
if boolean==True:#True, False判断，为True时则运行函数classification()
    symvalue,symcolor=classification(color,value,selectvalue)

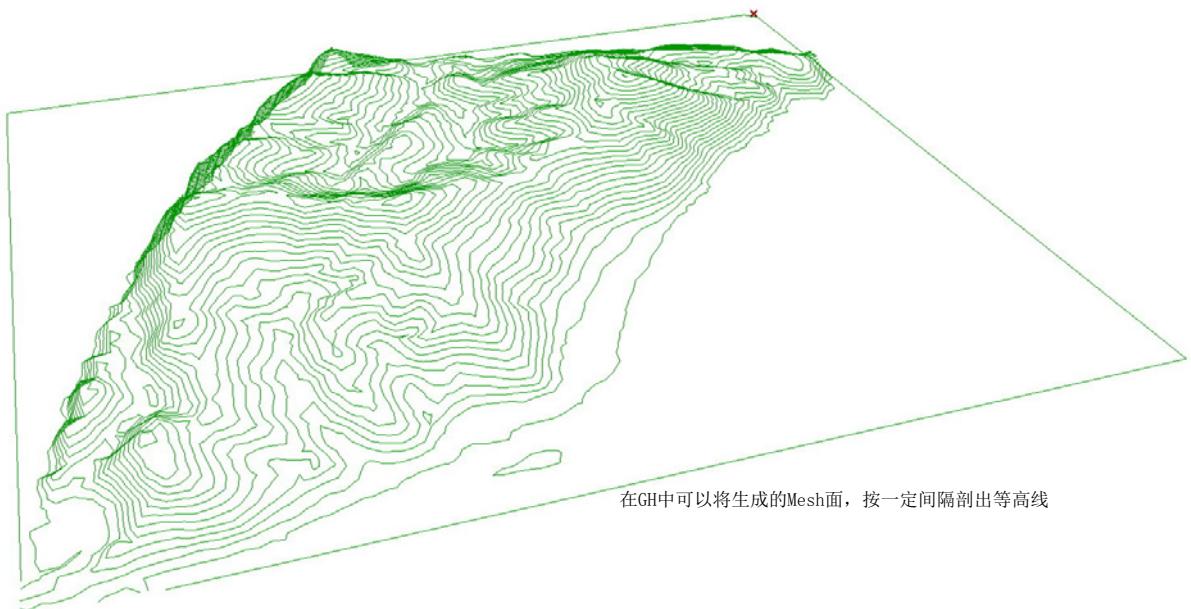
```





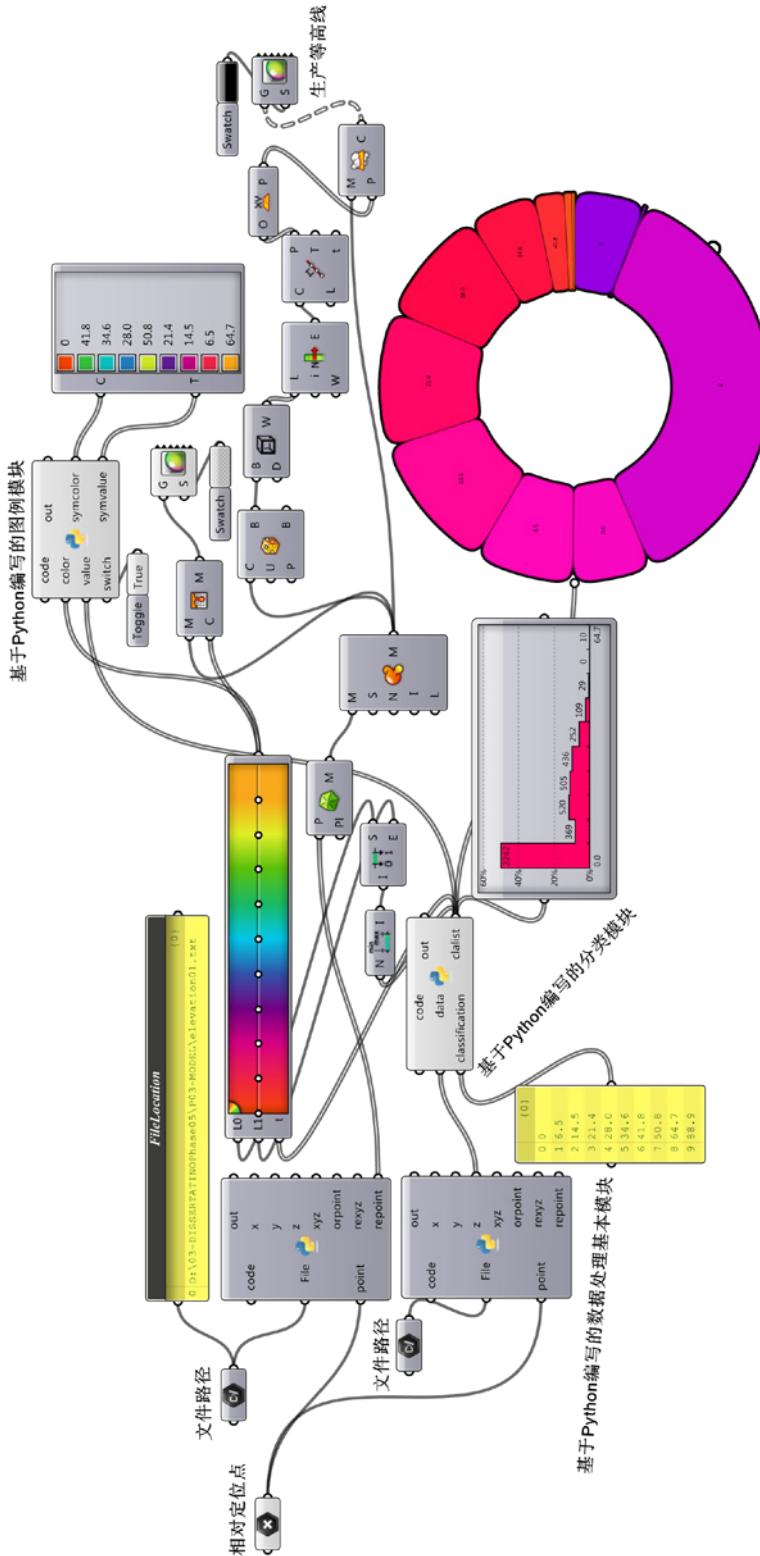
坡度值的调入：

在GIS平台处理的坡度等信息数据调入到GH中的方式与高程调入方式是一样的，之前编写的程序同样可以在这里使用。



G-03_07

slope.txt



gHowl



<http://www.grasshopper3d.com/group/ghowl>

gHowl包含一系列的组件，用于扩展GH与其它信息平台以及设备之间的信息交互。



UDP :

UDP允许多点传送（发送和接收）数据，可以使用的IP地址范围在224.0.0.0–239.255.255.255之间。

- Network Source: 测试网络链接;
- UDP Send: 可以通过网络发送UDP信息数据，或者发送OSC信息到支持OSC信息数据的软件平台或设备;
- UDP Receive: 可以通过网络发送和接收UDP信息，也可以接受OSC信息数据;
- OSC Channel: 该组件可以存储单独的OSC通道，改变组件的昵称可以存储地址信息;

- OSC Dispatch: 该组件可以用来存储多个OSC地址数据;

Spreadsheet:

电子数据表Spreadsheet只有在安装有OpenXMLDKV2.msi后才可以使用，该部分组件用于读取和写入*.xlsx文件。

- Spreadsheet Reader: 可以读取存储在硬盘空间的电子数据表;
- Spreadsheet Writer: 可以写入存储在硬盘空间的电子数据表;

XML:

该部分组件可以检索本地或者网络上的资源，例如RSS提供的数据。

- Pachube: 接收Pachube发送的数据（必须有可用的Pachube的API接
- OSC Channel: 该组件可以存储单独的OSC通道，改变组建的昵称可以存储地址信息;
- XML Parser: 该组件用于从语法上剖析存储于硬盘空间或者网络上的XML文件;
- XML Writer: 向XML文档写入数据;

GEO:

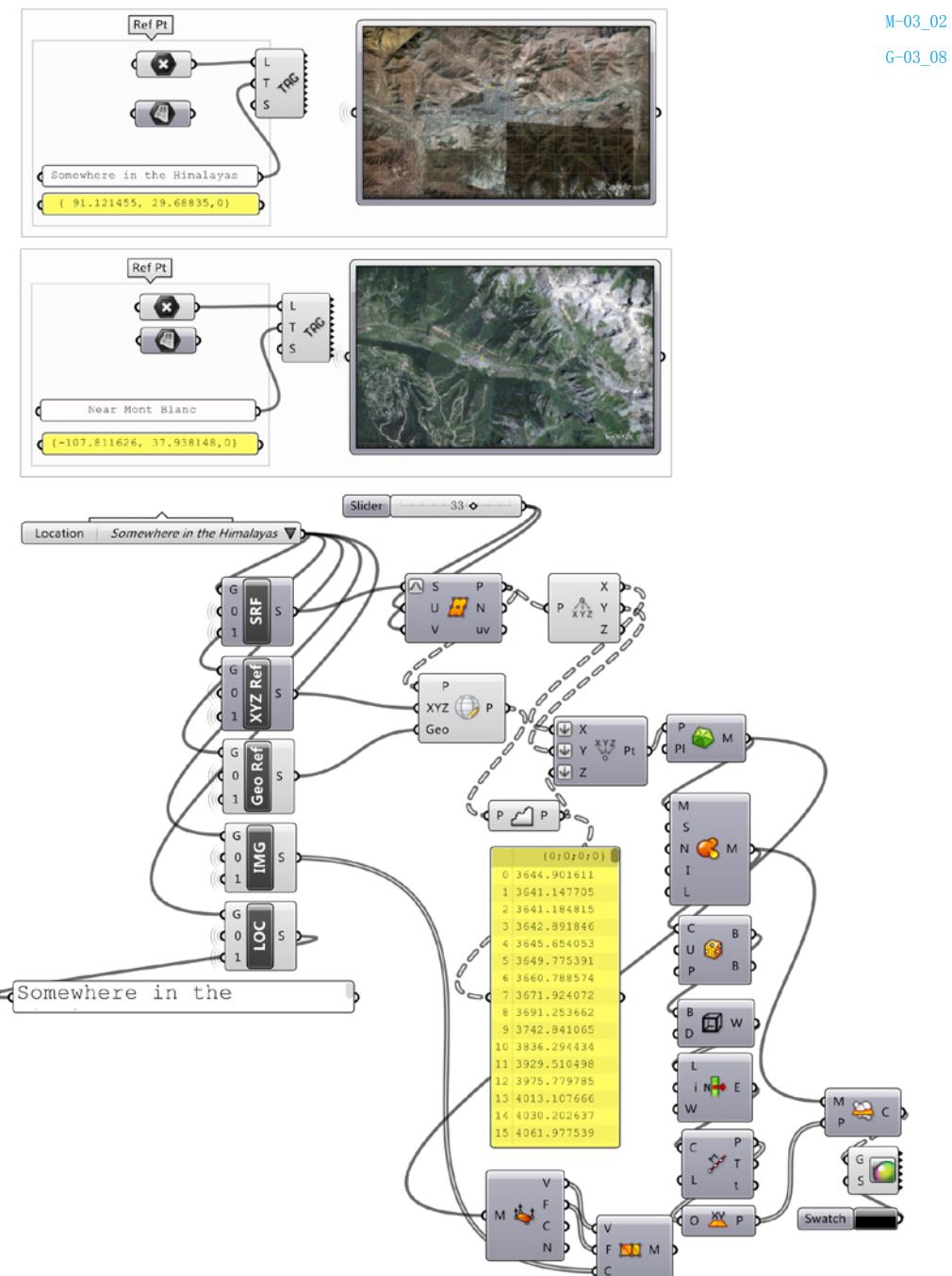
该部分组件可以转换GPS数据格式为XYZ坐标形式。

- Format GEO: 可以将NMEA（美海洋电子协会）格式的GPS数据转换为十进制格式;
- XYZ->GEO: 将GH中的XYZ格式坐标转换为经纬度和高程;
- GPS->GEO: 将经纬度和高程转换为XYZ数据格式;
- Elevation Service: 可以从GoogleEarth上下载指定坐标位置的高程数据;

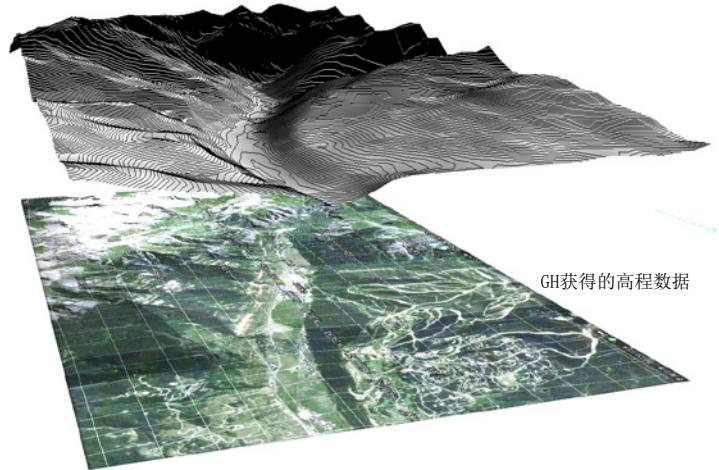
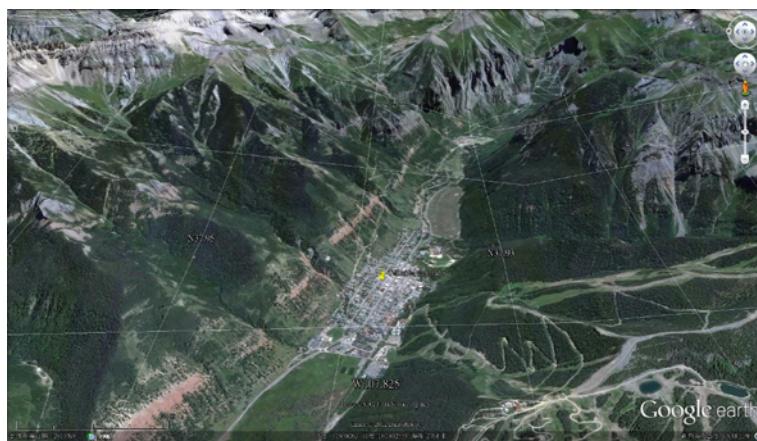
KML:

- KML Exporter: 可以输出点、线、面为KML文件，用于在GoogleEarth上加载。Breps输出前应转化为Mesh面。物体属性，例如颜色、透明度、线宽、线颜色等也可以同时被输出;
- KML Attributes: 可以增加填充颜色，线颜色和线宽等属性输出为KML格式文件;

从Google Earth上获得目标区域的高程

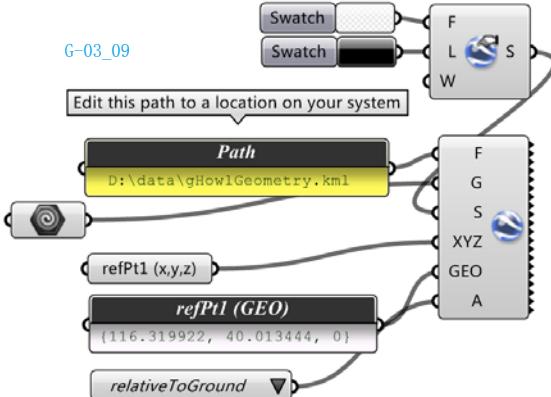


使用Elevation Service组件可以从Google Earth上下载目标区域的高程，并生成Mesh面以及获得等高线。

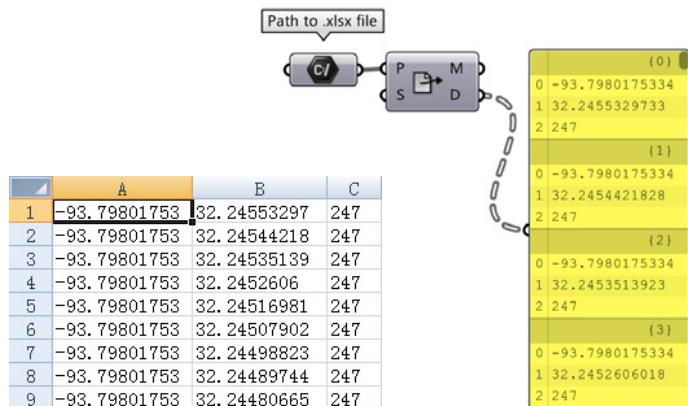


将GH模型输出为KML格式在GoogleEarth中加载

在GH中进行模型构建，设计推敲，有时并不能满足设计师对设计结果是否在实际场地上适宜的判断，因此可以将模型存储为KML格式的文件在GoogleEarth中加载，能够提供更加直观的体验，从而有利于判断设计各个阶段成果与场地因素的互动关系，推动设计发展。



gHowl中提供了SpreadSheet组件，可以读取.xlsx格式文件，并初步处理数据，与GH提供的ReadFile+StringSplit组合类似。对于.txt文件可以通过‘自文本’的方式，以逗号等分隔加载到Excel文档中。



3 基于动力学的模型构建

设计的几何形式丰富多样，一方面要发展出与设计场地精神及设计目的相一致的几何形式，同时该几何形式在建造时应符合力学基本要求，在发展几何形式时，可以通过各种方法来获得想要的几何，例如手绘草图的各种构思，想象下水滴融合的情景，击打水面的情景，翅膀的挥动，鱼儿的游动，甚至立方体的破碎，扭曲等各种变形，所能够列举的例子会很多，而上述所例举的往往与动力学相关，水滴在分子力作用下逐渐融合，水面受到外力干扰而出现波动等等，在没有计算机协助设计师进行模拟的时候，这需要收集足够多的相关资料，尤其运动图片用于研究，从而找出运动发生的特点来刺激设计师的构思。目前，已经可以依靠计算机来完成各种形态的模拟，而且不仅仅是自然现实中所发生的动力学形态，可以通过不同动力的使用，获得更加丰富的几何形态来协助设计师的构思。

另外，基于合理的动力学模拟，可以协助设计过程中结构的塑造趋于合理，很简单的一个例子就是吊桥。使用动力学组件来模拟各种内外力的作用，使几何形态、绳索、拉模等更接近真实，协助设计的发展。

Houdini

动力学模拟三维构建软件种类很多，3DMAX,MAYA等都具有动力学模块，这里使用更专业的动力学模拟软件平台Houdini来处理基于动力学几何形体的模拟，寻找丰富的几何形态。

<http://www.sidefx.com/> 官方网址

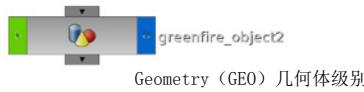
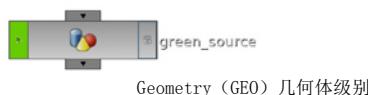


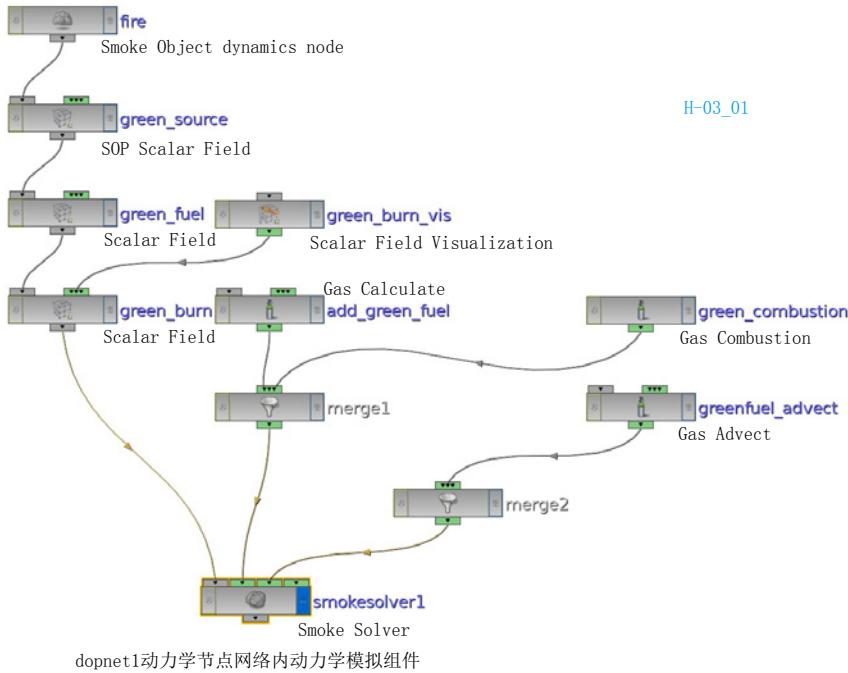
Houdini是由Side Effects Software Inc. (SEI) 好莱坞顶级视觉特效研发软件公司开发的主导产品。SEI公司由Kim Davidson和Greg Hermanovic创建于1987年。Houdini是在Prisms基础上重新开发而来，可运行于Linux, Windows, Mac OS等操作系统，是完全基于节点模式设计的产物，其结构、操作方式等和其它的三维软件有很大的差异，与GH (Grasshopper) 属于类似的节点操作模式，对底层组件的理解能够带来设计上极大的灵活性，和创造力。

Houdini有多种许可方式，其高清学习版Houdini Apprentice HD可以使用所有内置模块功能，对于设计师来说是可以选择的一种不错方式。

绿色火焰Green Flame:

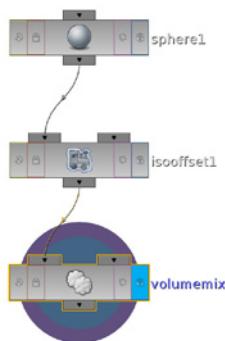
与GH相似在Houdini中主要以操作节点的方式完成模型的构建，这里构建了一个DOP Network动力学节点网络 (dopnet1)，用于模拟气体燃烧，两个Geometry (GEO) 几何组件分别用于构建原始的气体形态 (green_source) 和导出几何体 (greenfire_object2)。



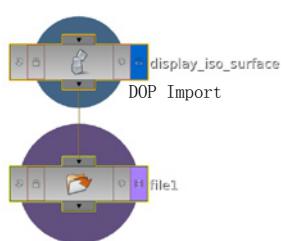


注：目前对于Houdini的市面教程微乎其微，可以参考。（张宝荣，吕新欣等编著。Houdini学习总动员-基础卷。北京：清华大学出版社，2010.）（张宝荣，吕新欣等编著。Houdini学习总动员-动力学卷。北京：清华大学出版社，2010.）两本书，但是更主要的应该是参考Houdini自带的帮助文件，其中包含多数案例，是学习的最佳资源；

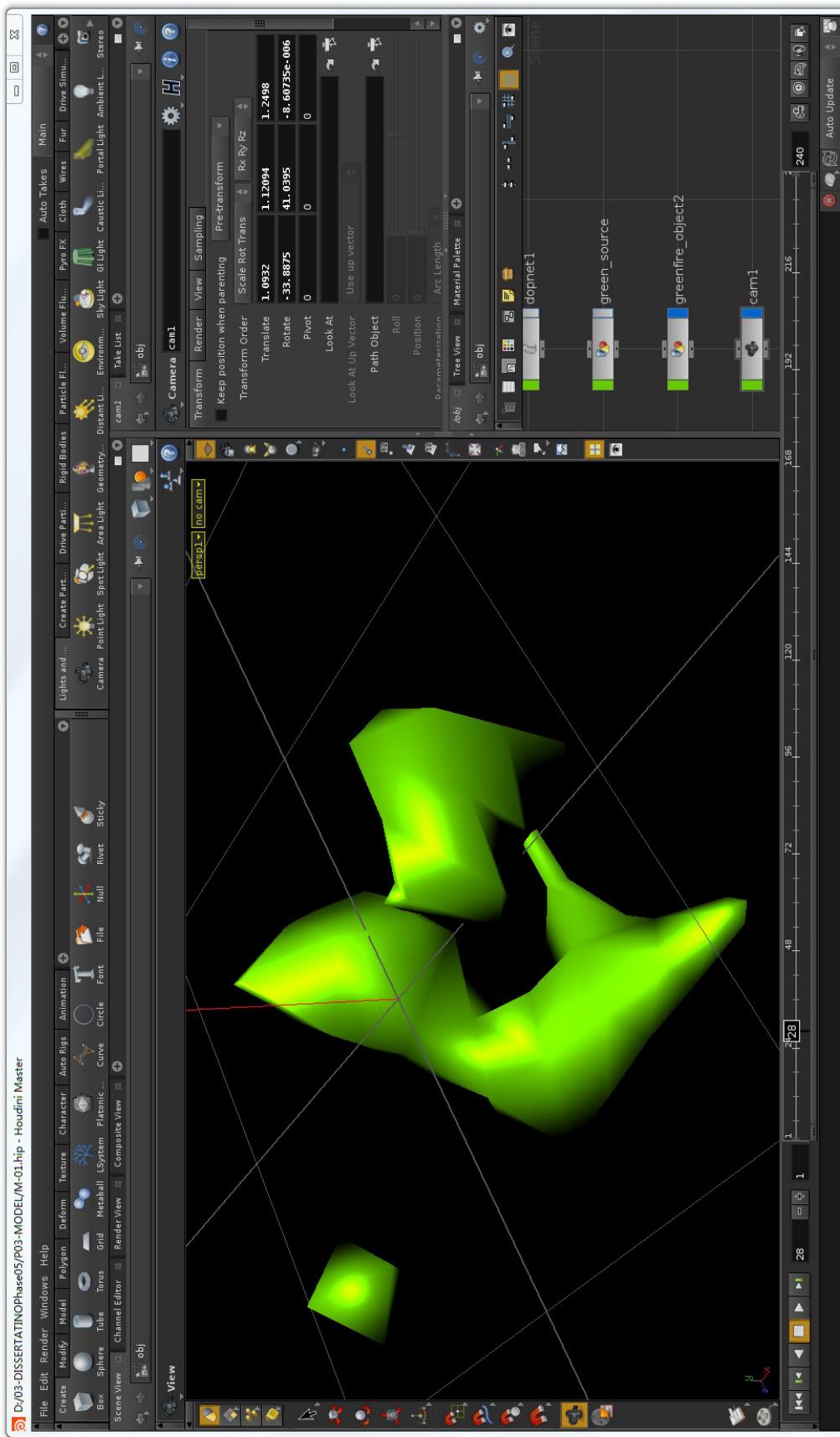
green_source 内组件连接

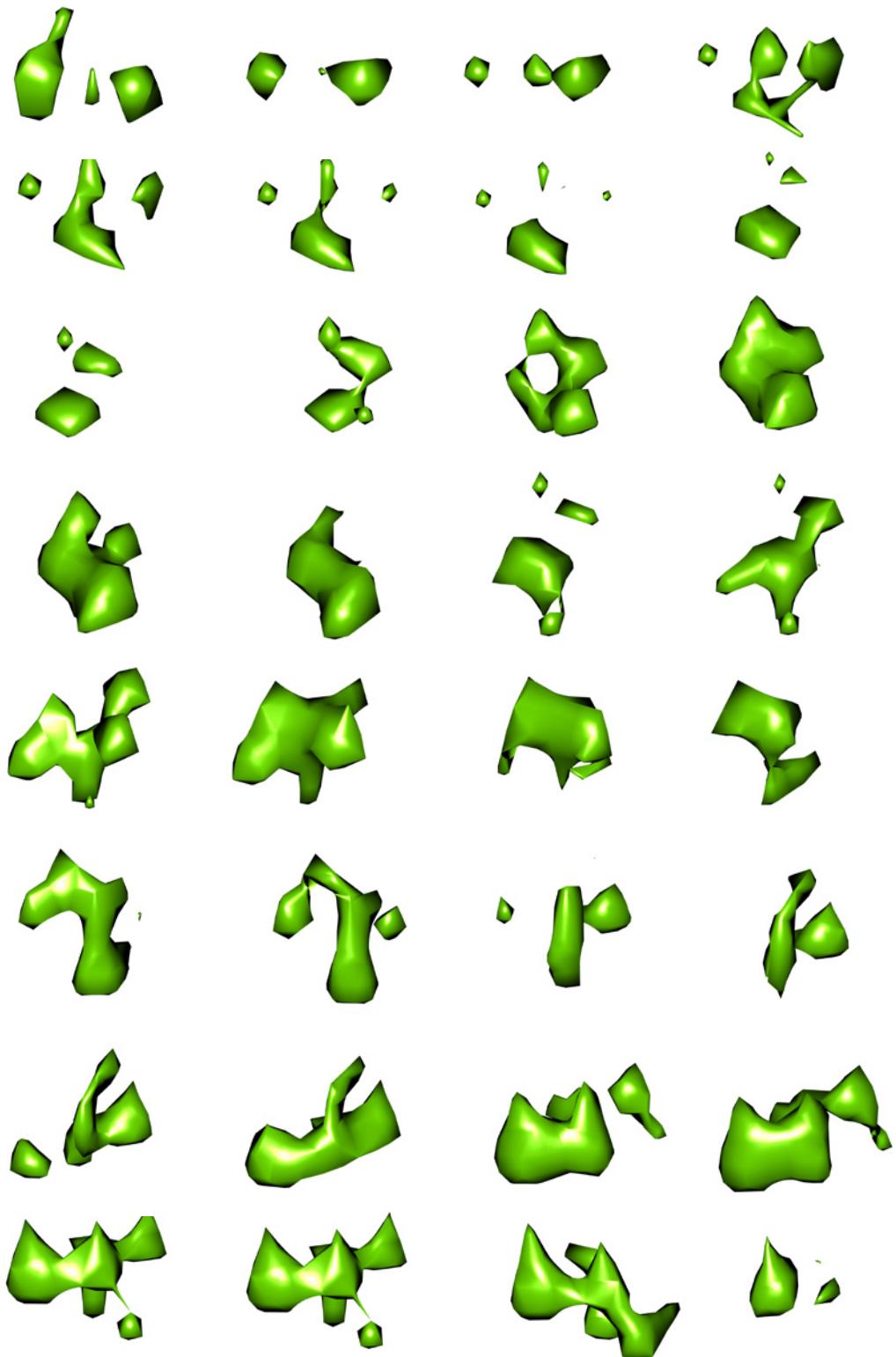


greenfire_object2 内组件连接

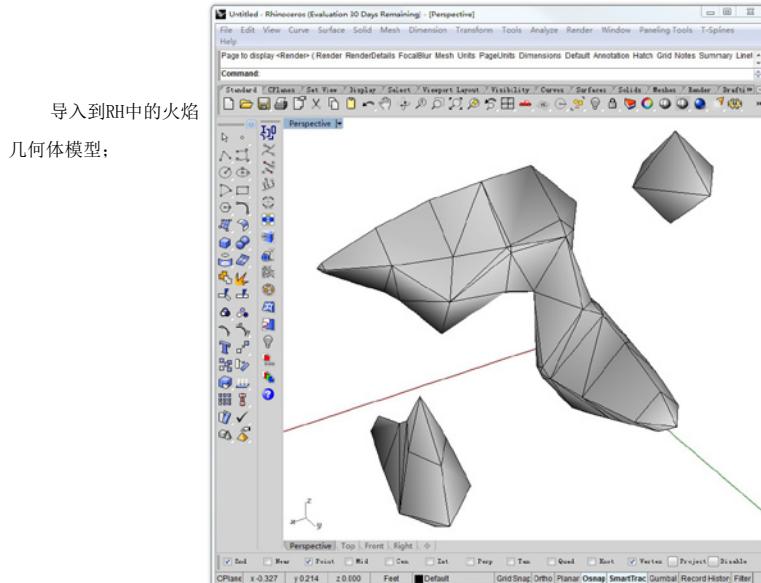


每个操作节点内往往包含丰富的设置内容，需要根据设计的目的进行调整。该例使用了Smoke Solver DOP烟雾动力学解算器进行计算。获得火焰的动态模拟。



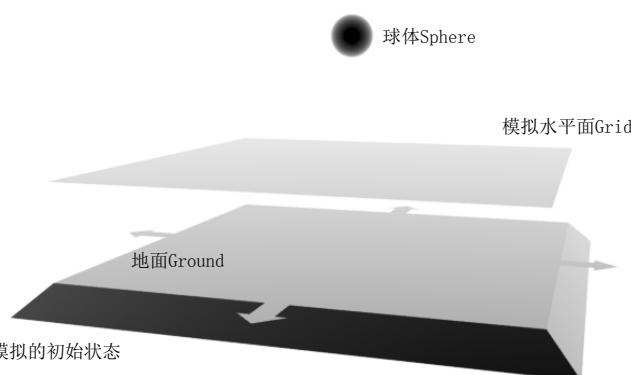


在Houdini中可以捕捉到任何时刻的火焰形态，这些形态可以提供设计上几何体的参考，这与过去观察并手绘草图，甚至实景拍摄照片研究火焰形态来推断设计形体是不同的，这里可以控制多种参数，火焰大小、密度、矢量方向，比起实物观察更具有可操作性，同时每种形态都是三维几何体面，可以直接与其它三维软件接口，例如Rhino，这里将某一时刻的几何体导出为Obj格式文件在Rhino中打开，进一步进行设计。

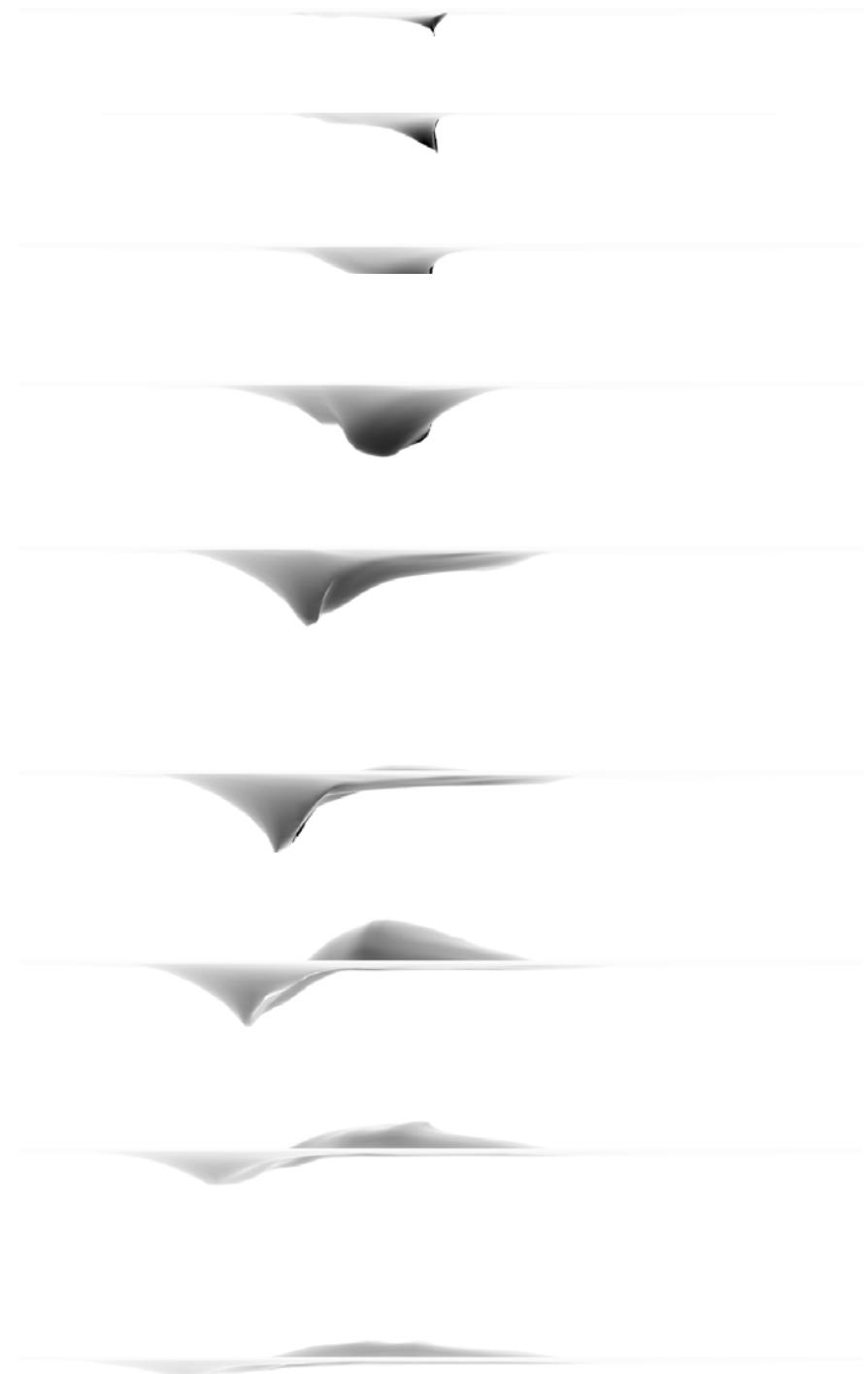


涟漪Ripple:

这里使用Ripple (波纹) Solver解算器模拟物体击打水面，产生波纹的形态。

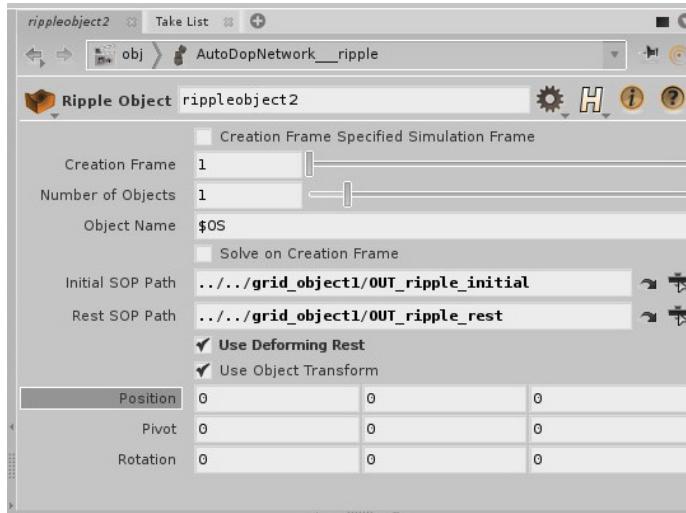


球体在重力作用下落入模拟水平面，使用Ripple Solver波纹解算器解算Ripple Object波纹对象，通过模拟波纹的推进对整个曲面进行变形。

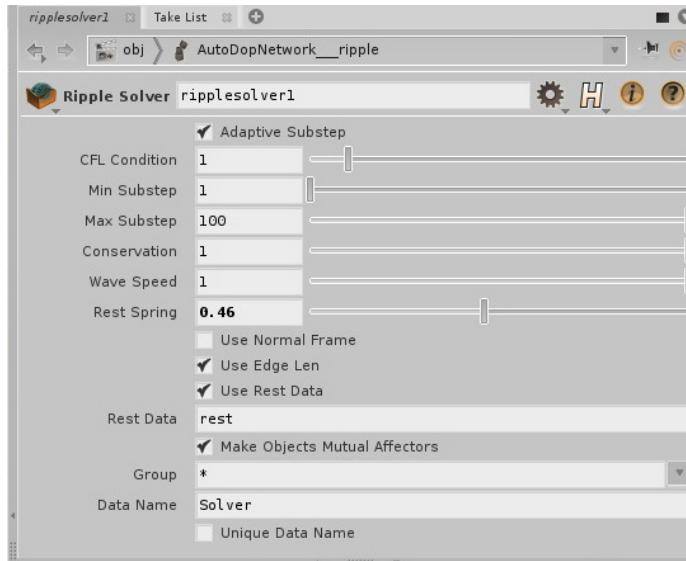


波纹对象与波纹解算器：

波纹对象Ripple Object dynamics node



波纹解算器Ripple Solver dynamics node



波纹对象通常从几何图形容器对象中提取已有的几何图形，然后用波纹解算器所需要的数据对其进行配置，波纹解算器通过模拟波纹的推进对整个曲面进行变形。

波纹对象包括两个曲面：

1—休止曲面The rest surface：曲面没有波纹时的普通形状，波浪会对曲面进行变形，产生上下波动。

2—初始曲面The initial surface：曲面的起始状态，它与休止曲面之间的差别会催生出曲面上的第一组波纹。

当前波纹状态和休止几何图形网络拓补（点数和接点数量）必须相同，以生产理想的结果；如果休止曲面和初始曲面拓扑不相同就会产生随机波纹形状。

将该解算器附加到带有波纹对象数据的对象（通常由波纹对象动力学节点创建或者由空对象和波纹配置对象共同创建出来）上，通过移动曲面上的点，在整个对象曲面范围内进行波纹推进。

有两种力可以影响波纹衍生过程：

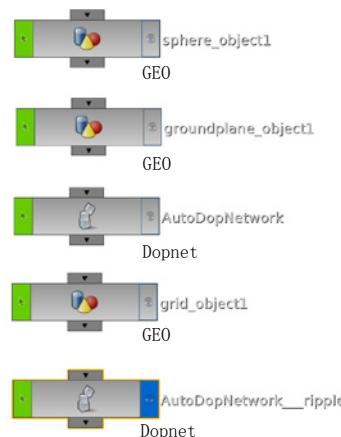
1-Wave speed参数控制波浪在曲面上的移动速度：

可以在每个点上绘制波浪的速度属性，使某一区域的波浪速度大于或小于其它区域，如果速度为零，当波浪碰到该区域时，会被弹回，而且该区域内的曲面都会保持静止。

2-Rest Spring在每个时间步骤上尽量保持曲面的休止形状。参数值越大，波纹消散越快，整个曲面损失能量就越大，最后恢复成休止状态。

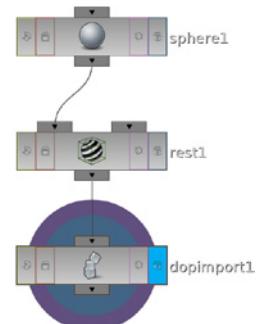
可以在每个点上绘制波浪的速度属性，使某一区域比其它区域弹性更好。同时，波纹推进不仅限于网格拓扑，还可以沿着曲线，3D点晶格甚至复杂的三角形网格推进波浪。

波纹模拟节点构成：

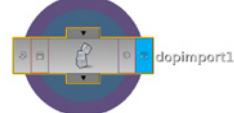


H-03_02

Sphere_object1内组件



groundplane_object1内组件

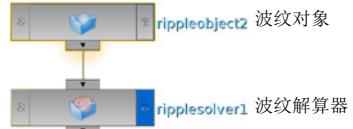


导入dopnet中的地面，方便显示

AutoDopNetwork内组件



AutoDopNetwork_ripple内组件



merge2



groundplane1



rigidbodysolver1



merge3



merge1



staticsolver1

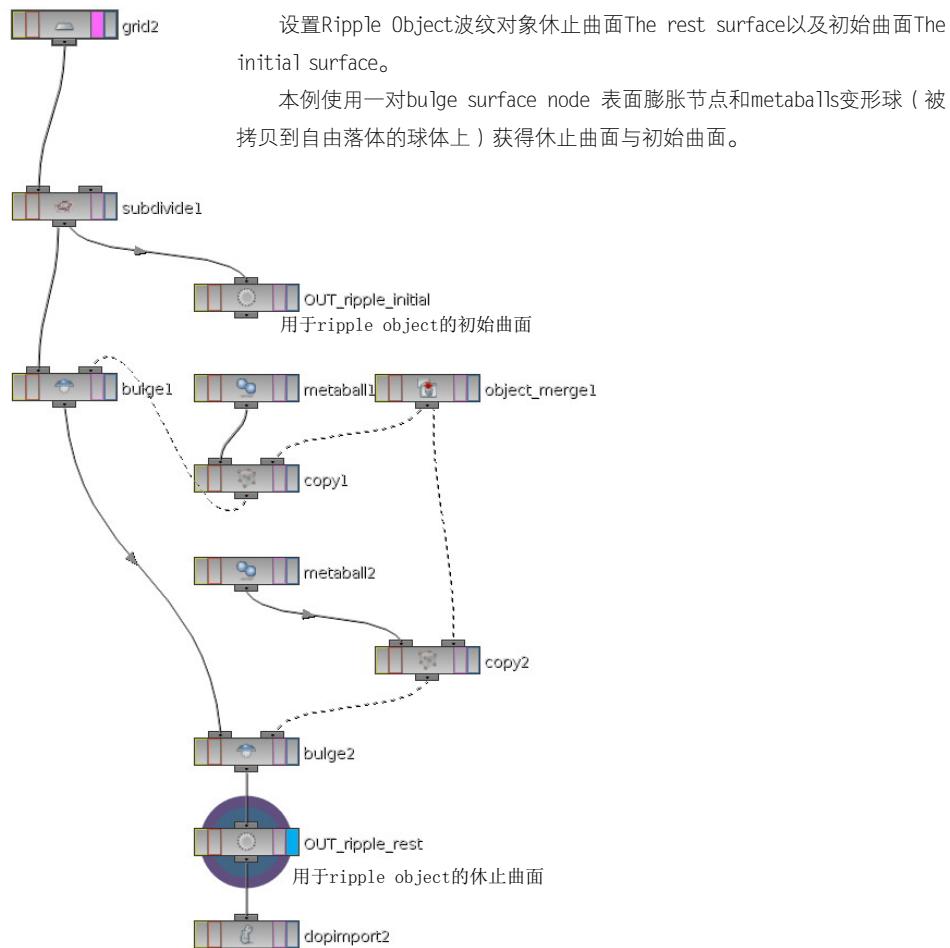


gravity1

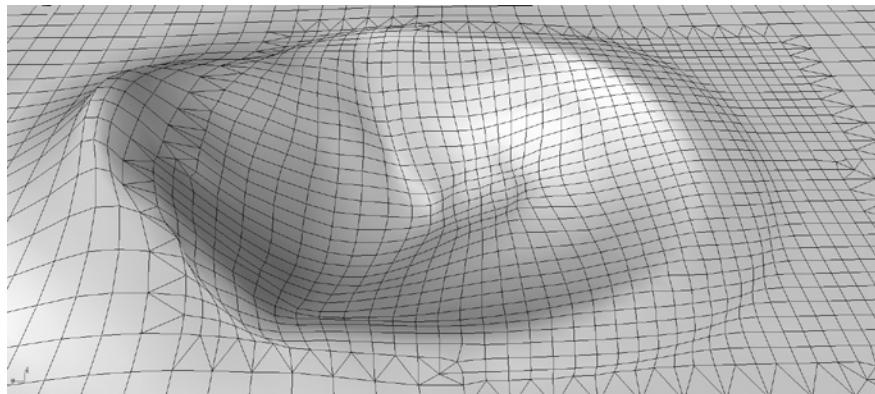


球体自由落下模拟

grid_object1内组件



导入到RHINO中模型曲面（某一时间点）





Kangaroo

<http://www.grasshopper3d.com/group/kangaroo> 官方网址

Kangaroo是可以直接在Grasshopper中交互式结构分析，动画模拟，优化，构型的动力学引擎。

设计者：Daniel Piker

研发小组：Robert Cervellione, Giulio Piacentino, Daniel Piker

Kangaroo目前可以免费在官网上下载和使用。

目前Kangaroo的内核算法并不能使模拟十分精确，在使用时建议作为设计师设计的协助工具，帮助构型与设计形态模拟。

什么是粒子系统Partical System?

"Particles粒子对象虽然不占据空间，但是具有质量、位置、速度，能够对各种力做出反应；即使粒子非常的简单，却表现出多样的运动行为，例如连接受到阻尼弹簧影响的粒子所建立起来的非刚体结构。

当然在现实的世界里，对象是由数不清的粒子所构成的，远远超出了模拟的数量，但是在模拟的过程中，将关注点放在点的布置，质量的分配上来获得接近于真实物理动力的模拟。对于设计师群体所开发的动力学模拟系统，尽量避免过于专业的技术知识，使模拟操作更加的直接，易于操控。



Kangaroo Physics Engine Kangaroo的主引擎组件

主引擎组件是在使用Kangaroo进行模拟时，一开始就应首先放置的。

Force Objects:

模拟中Kangaroo所提供的各种影响粒子的力都可以连接到Force Objects节点上。这些力矢量的形式具有共同的语法表述，因此力之间是可以相互作用的，在输入到Force Objects节点上时，应以展开的(Flatten)方式连入；



Anchor Points:

Anchor Points 定位点是在模拟过程中不受到Kangaroo各种力形式影响的点，它被固定在其位置上，只有在RHINO空间中可以移动，进行交互式的模拟；

Particale Consolidation-Joining objects together

粒子合并与对象连接

如果一开始输入到Force Objects与Anchor Points输入端的点不只一个，并具有相同的位置，那么在Kangaroo模拟中将被连在一起，视为一个单独的点存在。没有必要专门对它们事先进行合并。

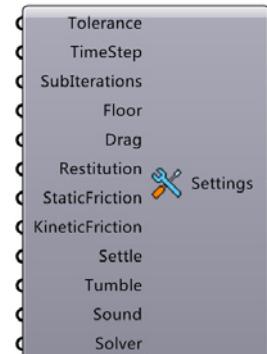
如果Force Objects力对象的点与Anchor Points的点重合，例如Spring弹簧的末端，在模拟的过程中会移动这些对象，在停止模拟，力对象的点不再被连接，在重新设置时，需要将模拟过程中移动的这些对象点在RHINO空间中移回。

Kangaroo Settings Kangaroo设置

Settings用于全局性的模拟设置，连接到kangaroo Physics组件的Settings输入端。

Tolerance:

不同点之间的最小距离，当点距小于该值，将被整合为一个点；



TimeStep:

迭代一次所占据的系统时间范围。较小的值能够使结果更加的稳定，但是会降低模拟速度。对于较强的力和不易弯曲的弹簧一般需要设置较小的TimeStep时间步；

SubIteration:

用于计算每次将动力模拟结果进行显示的迭代数；

Floor:

用于设置Floor地面的开启与关闭，限制粒子是否能够通过Z=0的水平面，模拟速度要比Brep碰撞快；

Drag:

Drag拖拽（粘力）作用于所有粒子限制其运动，使模拟的粒子运动趋于静力平衡。如果拖拽的力很小，系统将会摆动较长时间；较高，粒子运动则会较慢；

Restitution:

恢复指数可以帮助设置粒子之间以及与地面之间的弹性碰撞，值为1时，粒子将会被反弹回初始位置，为0时，不会发生反弹；

StaticFriction:

粒子与地面之间静力摩擦系数；

KineticFriction:

粒子与地面之间动力摩擦系数；

Settle:

使粒子停止反弹的速度值；

Tumble:

设置粒子与地面发生碰撞后保持粒子水平运动速度的数量值，值域为0~1；

Sound;

音响效果，在目前的版本中不可用；

Solver:

解算器，用于设置kangaroo计算粒子运动位置的积分方法；

Utilities工具

Kangaroo包含移出复制的工具：

RemoveDuplicatePts: 用于清理列表中的点。输入t项可以设置其容许值，小于改值的粒子将被合并为一个粒子点；



RemoveDuplicateLines: 用于清理列表中的线；



Running the simulation 运行模拟

运行模拟需要在Kangaroo主动力引擎组件的SimulationReset输入项连接Boolean Toggle组件使用True或者False控制模拟。在没有Timer组件连接到主引擎时，每次只计算一次迭代，同时Timer组件可以设置时间步幅，从而控制每次解算更新的时间长度。

Newton's Laws 万有引力定律

物体的动力变化与作用与物体上的冲量是成比例的，并且沿直线发生。

公式： $F=ma$ Force力=Mass质量 \times Acceleration加速度

$$a=F/m$$

Kangaroo通过计算所有作用于粒子不同类型的力找到力矢量F，并使用牛顿第二定律获得加速度，以数值积分法，通过设置的时间步幅计算运动微分方程，获得所有粒子运动轨迹。

注：在Kangaroo中Mass质量没有自动关联到weight重量上；

质量较大的对象需要较大的力才能改变其速度；

Weight重量是由于Gravity重力而作用与物体上的力；

在现实的世界中，重量与质量成比例，在Kangaroo中可以轻松地进行该情形的设置，或者完全各自独立的进行设置达到特别的模拟目的，例如测试单位荷载；

在Kangaroo中可以设置粒子重量为0，但是如果质量为0，将会引起计算错误；

并不总是需要指定质量值，如果没有特别值的输入，粒子默认的质量为1；

如果需要指定粒子的质量和速度，需要使用Particle 组件；



如果包含重力的影响，给粒子一个重量，需要使用Unary Force组件；



To every action there is always an equal and opposite reaction
总是存在相等与相对的反作用力

Kangaroo中使用线段来计算力，每对粒子群之间的相互作用力视为线段的端点之间的相互作用力，该作用力沿线段的方向，使两端的粒子速度增加或者减小。

Unary Force组件例外，只应用于单独的粒子群。可以假定为一个无限距离与质量的粒子作用与该粒子的结果。

Every body persists in its state of being at rest or of moving uniformly straight forward, except insofar as it is compelled to change its state by force impressed

一般每个物体都会保持直线运动或者静止，除非受到外力的作用迫使其改变本身的状态

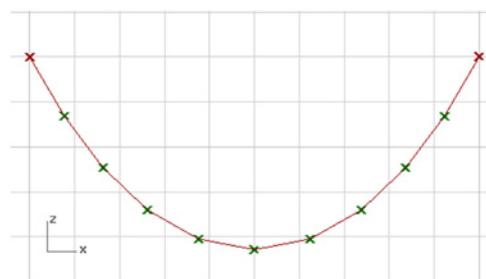
如果作用与粒子上的力和为零，将不会改变其加速度；相反，不为零，粒子的速度将会发生持续的改变。在模拟过程中设置Friction摩擦力和Drag拖拽力，将会阻止运动，最终使粒子处于在合力为零的平衡状态下。

Discretization 离散

在Kangaroo中Spring弹力由两个点决定，并保持为一个直线。为了模拟富有韧性的对象，需要将对象例如代表绳索的曲线打断为多个Segment部分，将弹力作用与每个单独的部分。

Cable 绳索：

可以在RHINO或者Grasshopper中将线段分段；

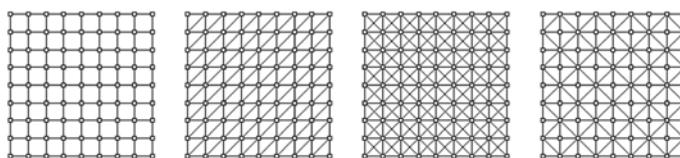


Sheet materials—members, fabrics, paper etc 板材—薄膜，织物，纸张：

在Kangaroo中建立板材最简单的方法就是使用富有弹力的栅格Grid。Weavebird插件（前文有述）能够很好的提取网格Mesh的边协助力学模拟。也可以使用UT0的Mesh工具。

对于Rest Length值为0的栅格，栅格面积将不断的缩小直至消失。如果想避免该种情况例如模拟富有张力的织物，需要设置每个线段部分Rest Length值在0~1之间变化。也可以增加Mesh面外的Stiffness弹簧刚度。对于悬链线的动力模拟，结果长度会比初始长度长，因此Rest Length的值应比初始长度大。

模拟现实中布料的运动，需要增加剪切力Shear springs（对角线），来阻止方格网变形为菱形。对于该类模拟有很多增加对角线的方法：



Rods 棒条体：

通过设置3个点可以模拟抗弯曲运动，该组件会使用于模拟的点尽量保持在一条直线上。对于棒条体可以固定其末端点为Anchor Points进行模拟。

目前Kangaroo还不能模拟1D元素的抗扭性。



Springs:

开始使用Springs组件进行整个结构的动力学模拟看起来有些奇怪—但是这并不是讨论车弹簧以及床垫的问题，任何具有弹性的对象都可以对其施加Springs来模拟拉伸和压缩。

关于Hooke's law胡克定律：是力学弹性理论中的一条基本定律，表述为：固体材料受力之后，材料中的应力与应变（单位变形量）之间成线性关系。满足胡克定律的材料称为线弹性或胡克型（英文Hookean）材料。

从物理的角度看，胡克定律源于多数固体（或孤立分子）内部的原子在无外载作用下处于稳定平衡的状态。

许多实际材料，如一根长度为L、横截面积A的棱柱形棒，在力学上都可以用胡克定律来模拟——其单位伸长（或缩减）量 ϵ （应变）在常系数E（称为弹性模量）下，与拉（或压）应力 σ 成正比例，即： $\sigma = E \epsilon$

或者： $\Delta L = \frac{1}{E} \times L \times \frac{F}{A} = \frac{1}{E} \times L \times \sigma$

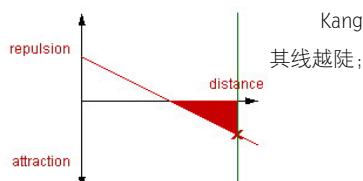
其中 ΔL 为总伸长（或缩减）量。胡克定律用17世纪英国物理学家罗伯特·胡克的名字命名。胡克定律仅适用于特定加载条件下的部分材料。钢材在多数工程应用中都可视为线弹性材料，在其弹性范围内（即应力低于屈服强度时）胡克定律都适用。另外一些材料（如铝材）则只在弹性范围内的一部分区域行为符合胡克定律。对于这些材料需要定义一个应力线性极限，在应力低于该极限时线性描述带来的误差可以忽略不计。

还有一些材料在任何情况下都不满足胡克定律（如橡胶），这种材料称为“非胡克型”（non-hookean）材料。橡胶的刚度不仅和应力水平相关，还对温度和加载速率十分敏感。

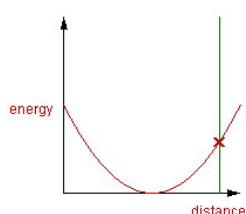
胡克定律在磅秤制造、应力分析和材料模拟等方面有广泛的应用。

在Kangaroo中，开始的长度由曲线的长度确定。Rest Length休止长度或者称之为Natural, slack length自然，松弛长度，由Rest Length输入端输入。如果不提供该输入值，默认值为零。如果想使该值与初始长度一样，则可以将曲线的长度连入，也可以提供一个乘积，使休止长度为初始长度的倍数。如果乘数在0~1之间，则类似于弹簧的预拉伸。

Spring

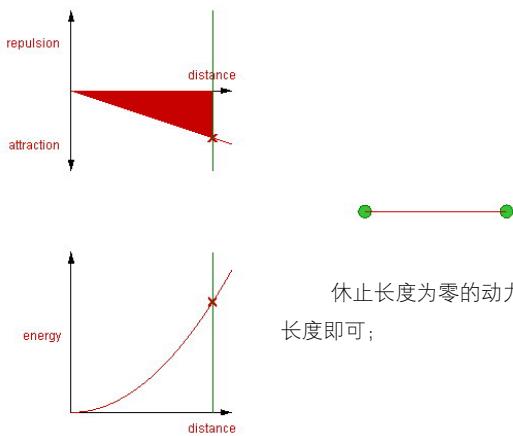


Kangaroo中的力计算图示，水平轴向表示弹力的自然长度；弹簧的弹力越大，



势能变化图示，动能Kinetic energy越大，其势Potential energy越小，反之亦然。当不存在拖拽力时，将在相对的同一高度滑动，动能和势能不断的转换；

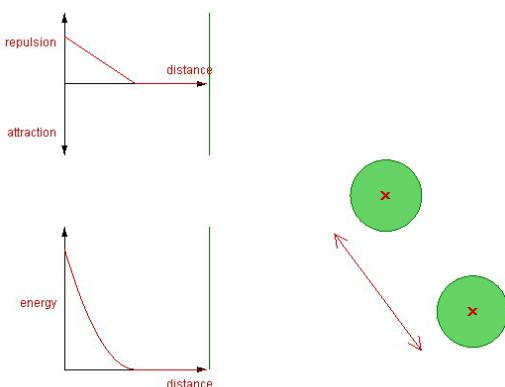
Zero rest-length spring 休止长度为零时



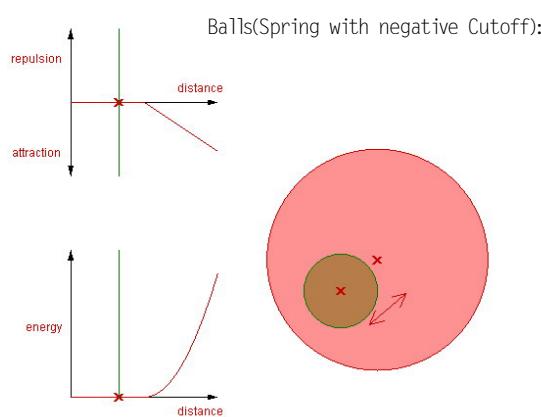
休止长度为零的动力模拟可以应用到最小曲面模拟中，将Mesh面的边设置为零长度即可；

Cut-offs 切断

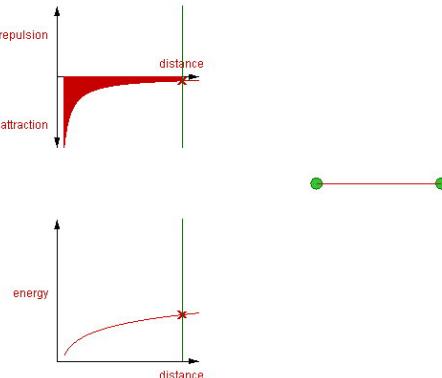
Balls(Spring with positive Cutoff):



SpringCutoffs输入项可以控制弹力在一个指定的距离上失效；如果该值与弹簧长度一致，那么粒子模拟类似于对一对未碰触的固体球不施加力的作用，为正值时，在该范围内弹簧力有效，为负值时，在该范围外弹簧力有效，为零时则在任何距离都有效；



Pover Laws 幂定律：



在模拟过程里使用幂函数能够获得较大的负指数。使力在一个较短的距离内急速增加，类似于靠近的物体爆炸式的分离。较小的指数值例如0.1可以减小这种效应。

图示为指数为-1时的情况，随着距离的增加，吸引力由急速降低到逐渐趋于零。在Kangaroo中指数值仅为整数值。当使用幂函数，为正值时，粒子间的斥力随距离而增大；为负值时，粒子间的引力随距离而减小。

其它力形式.....

The simulation oscillates wildly or explodes 模拟失真

Kangaroo是以不连续的时间步计算近似于连续的运动模拟，所以偶尔会发生错误，例如在处理非常硬的弹簧，粒子时容易引起计算失败，这是可以适量松弛弹力，或者减小时间步。例如从0.01到0.001。增加牵引力和阻尼值也会有所帮助。小的时间步意味着更多的计算，模拟运行速度会降低。抵消这种情况的方法可增加主引擎SubIterations输入项的值。

Kangaroo中涉及到各种力的形式：

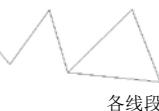


悬吊物

Kangaroo的图形部分条件输入可以在RHINO (RH) 中完成，再调入到Grasshopper (GH) 中，也可以直接在GH中直接处理。

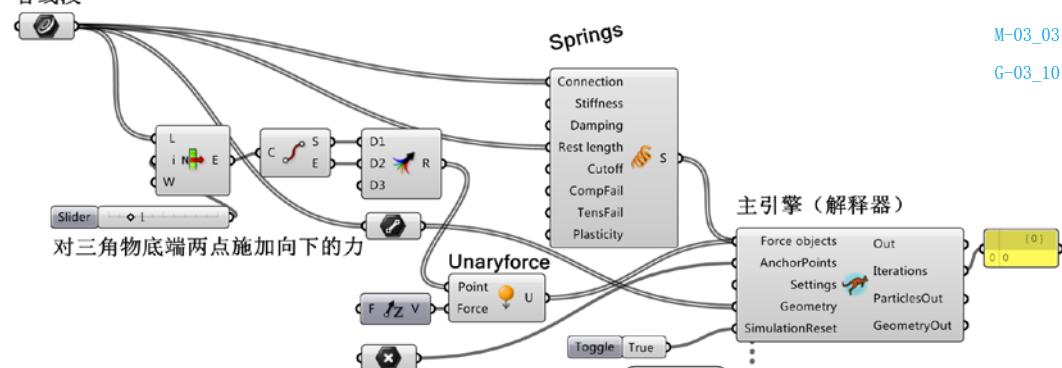
本例图形部分是在RH中处理，其中包含用于弹力计算的各线段，连入主引擎AnchorPoints输入端的固定点。

固定点



各线段

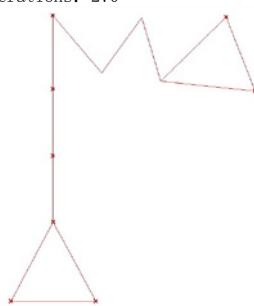
各线段



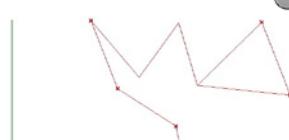
对三角物底端两点施加向下的力



Iterations: 270



Iterations: 5550

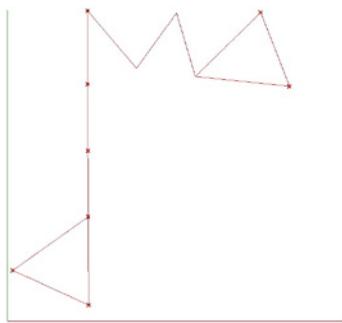


Iterations: 770



Iterations: 1320

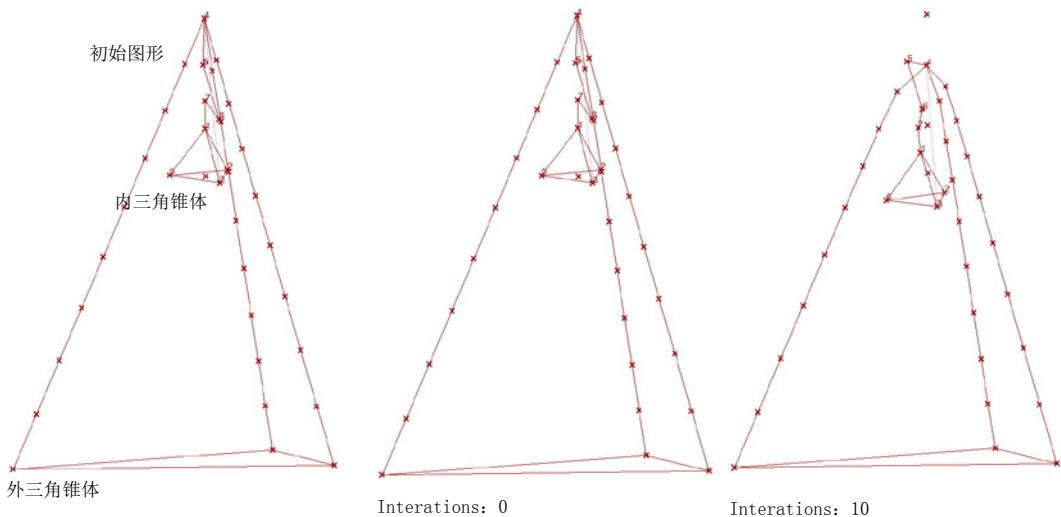
在Springs组件设置中，将Rest Length设定为与初始线段一样的长度，形成刚性对象，不发生弹性长度改变。三角物底边两点被施加于向下的力，带动整个链条坠落。如果只选择底边一点施加向下的力，最后达到力平衡时，三角物将向一边倾斜。



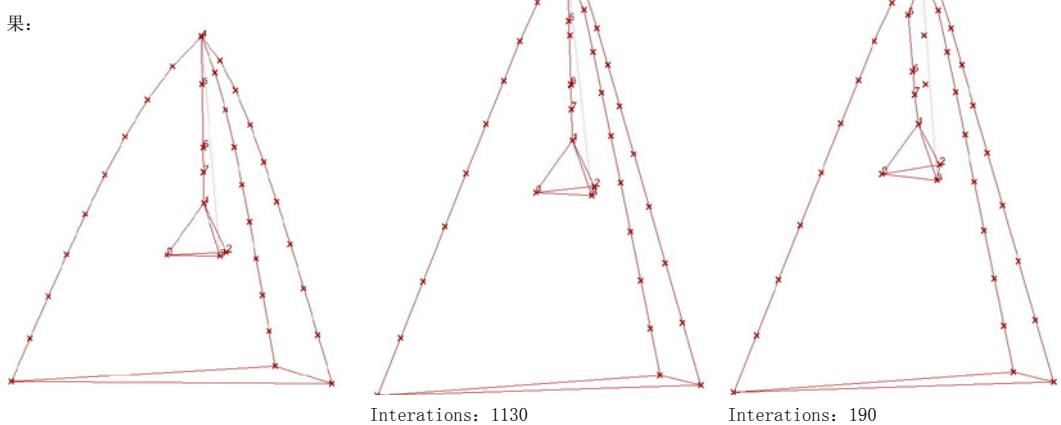
本例是一个比较简单的案例，以线段为弹簧力施加的对象。而在模型构建时可以对想要施加的对象采取该种处理方法，同时可以调整各样参数，例如休止长度等，动力模拟也并不局限于2D空间，可以在3D中进行模拟，达到设计师进行模型动力学部分的构建目的。

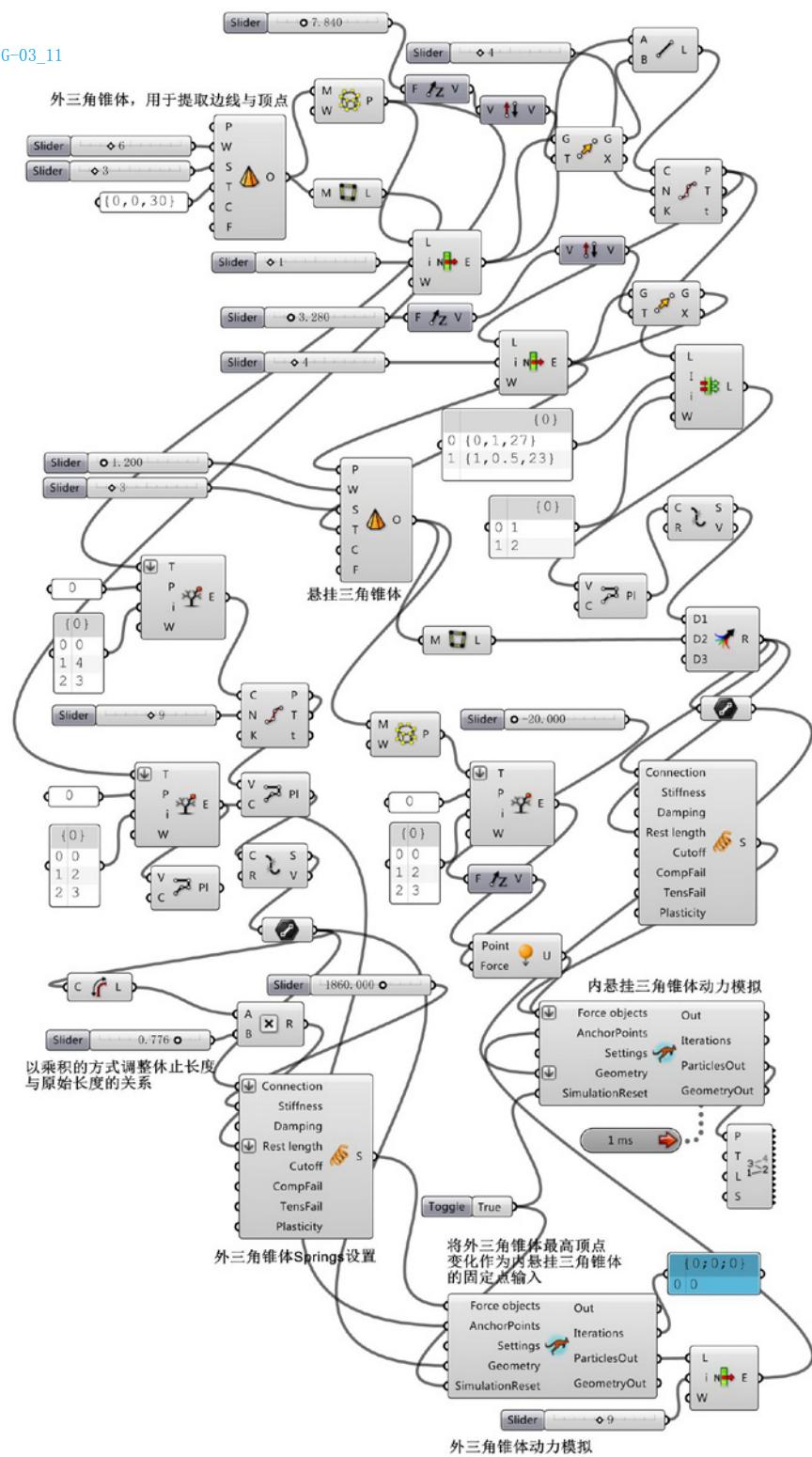
三角锥体建筑与悬吊物

这里试图推敲由三根支柱顶点相交构成三角锥体，在弹性力作用下的形变，同时将上例中2D的三角形变化为3D中的三角锥，所有条件都是在GH中设置完成。

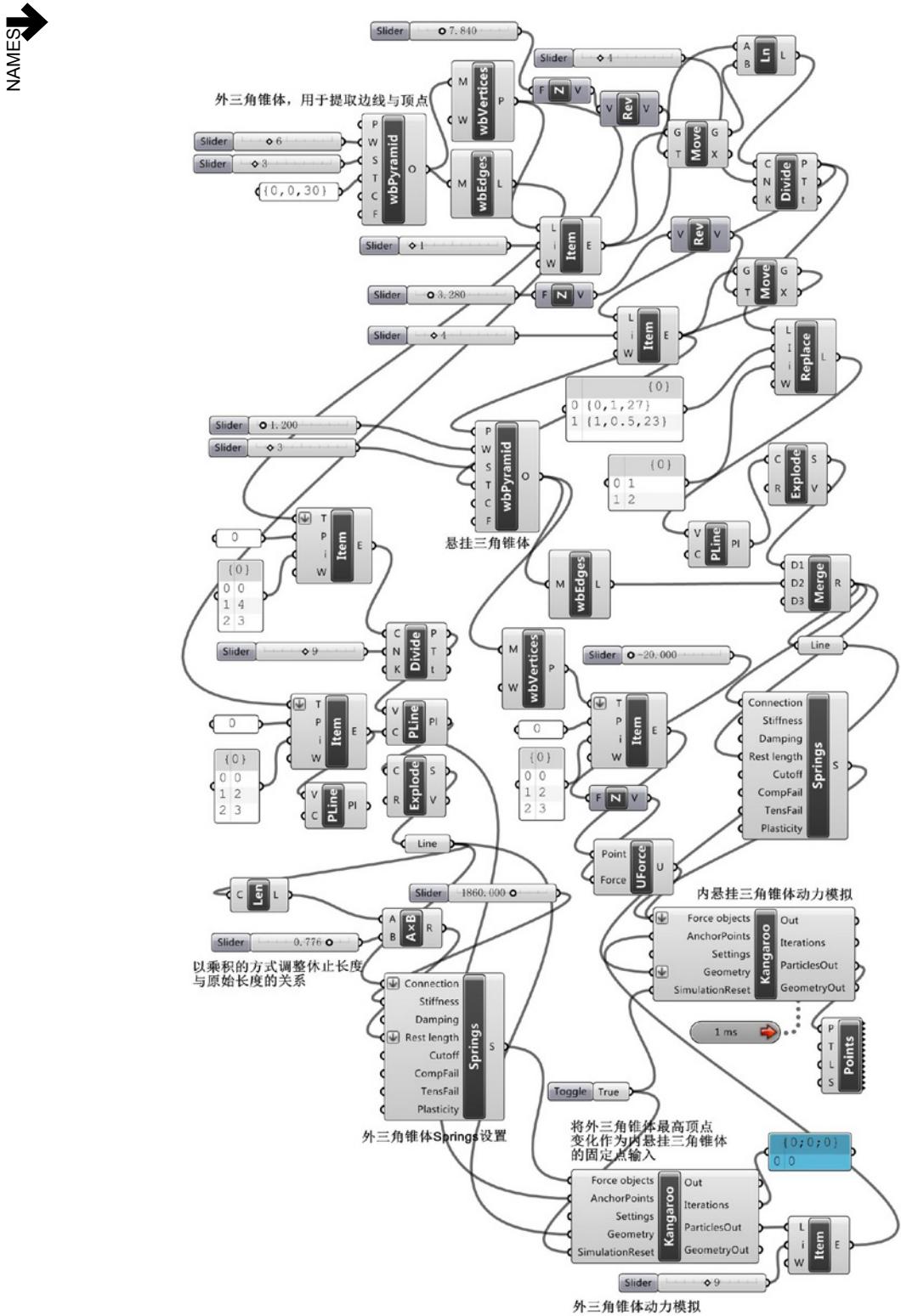


模拟过程中可以调整各项参数，这里
调整休止长度，会获得不同的模拟结
果：





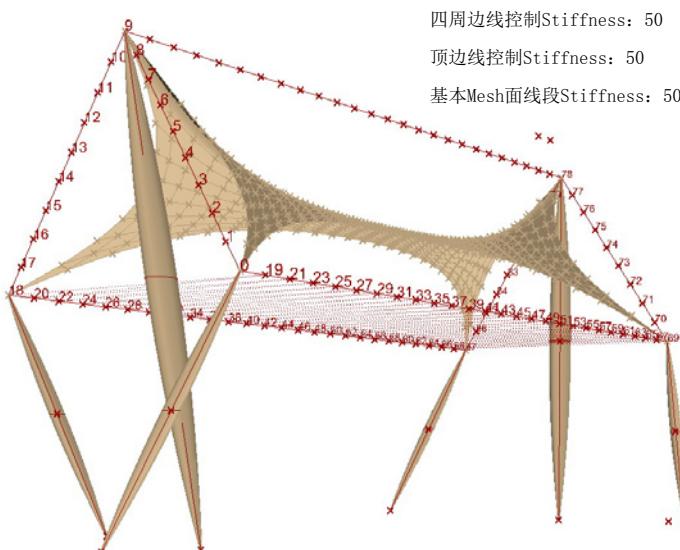
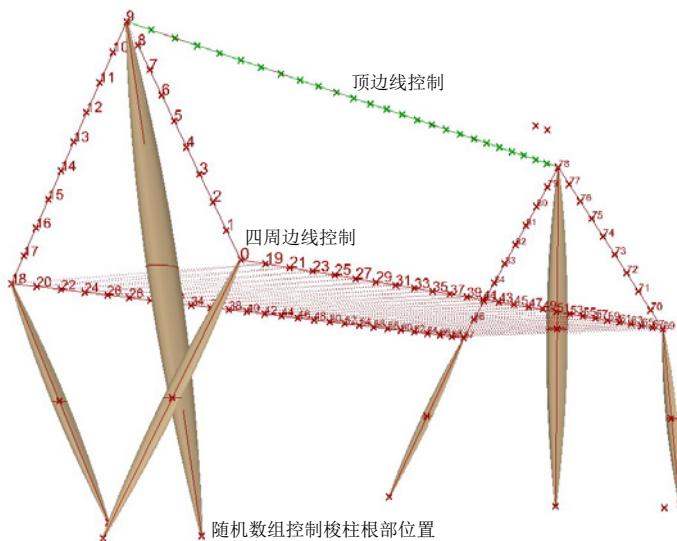
ICONS

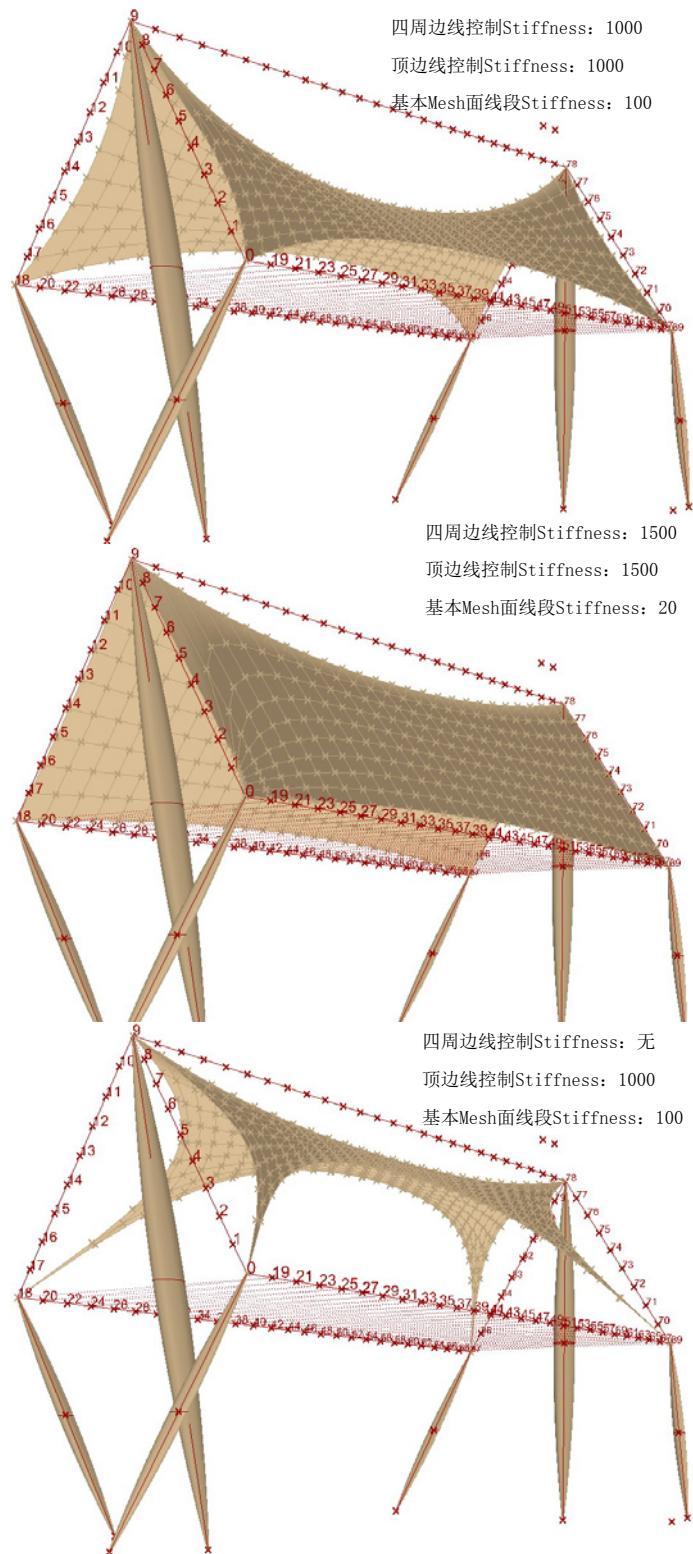


面的张力

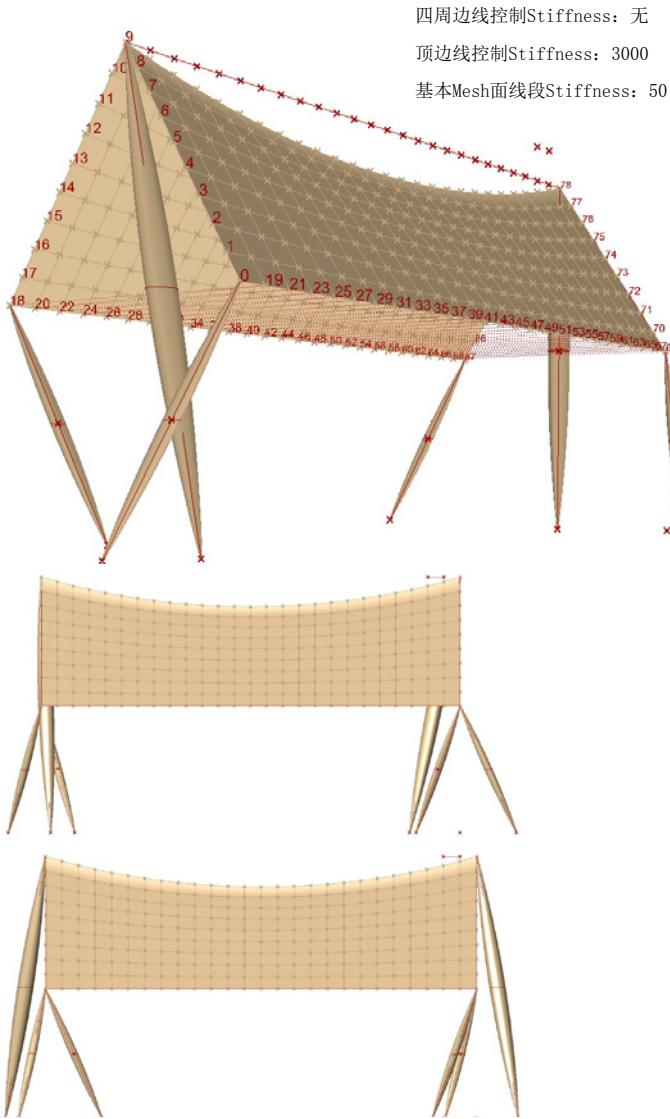
Springs可以处理单独的线段，而Mesh面有很多线段构成，所以可以对其各线段施加弹力，从而影响整个Mesh面的变化。对面的控制，可以根据设计意图设置不同线段具有不同的弹力，以及选取需要控制的点作为固定点。在这个模拟过程里，不同线段的不同弹力可以连接到同一个解算器上，各个弹力参数值则在各自的Springs组件上设置。

本例中的模拟，为了获得面的边线使用了Python语言进行编写。对于棱柱部分，为了增加棱柱位置的自由性，使用随机数组进行设置，以圆作为变化的范围，随机选取圆上一点作为棱柱的根部。



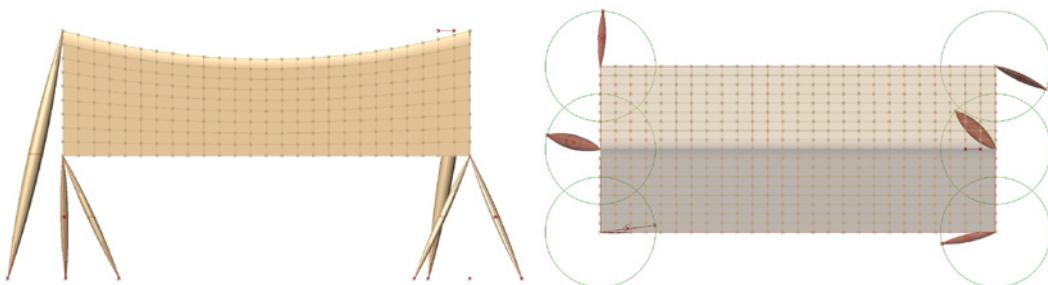


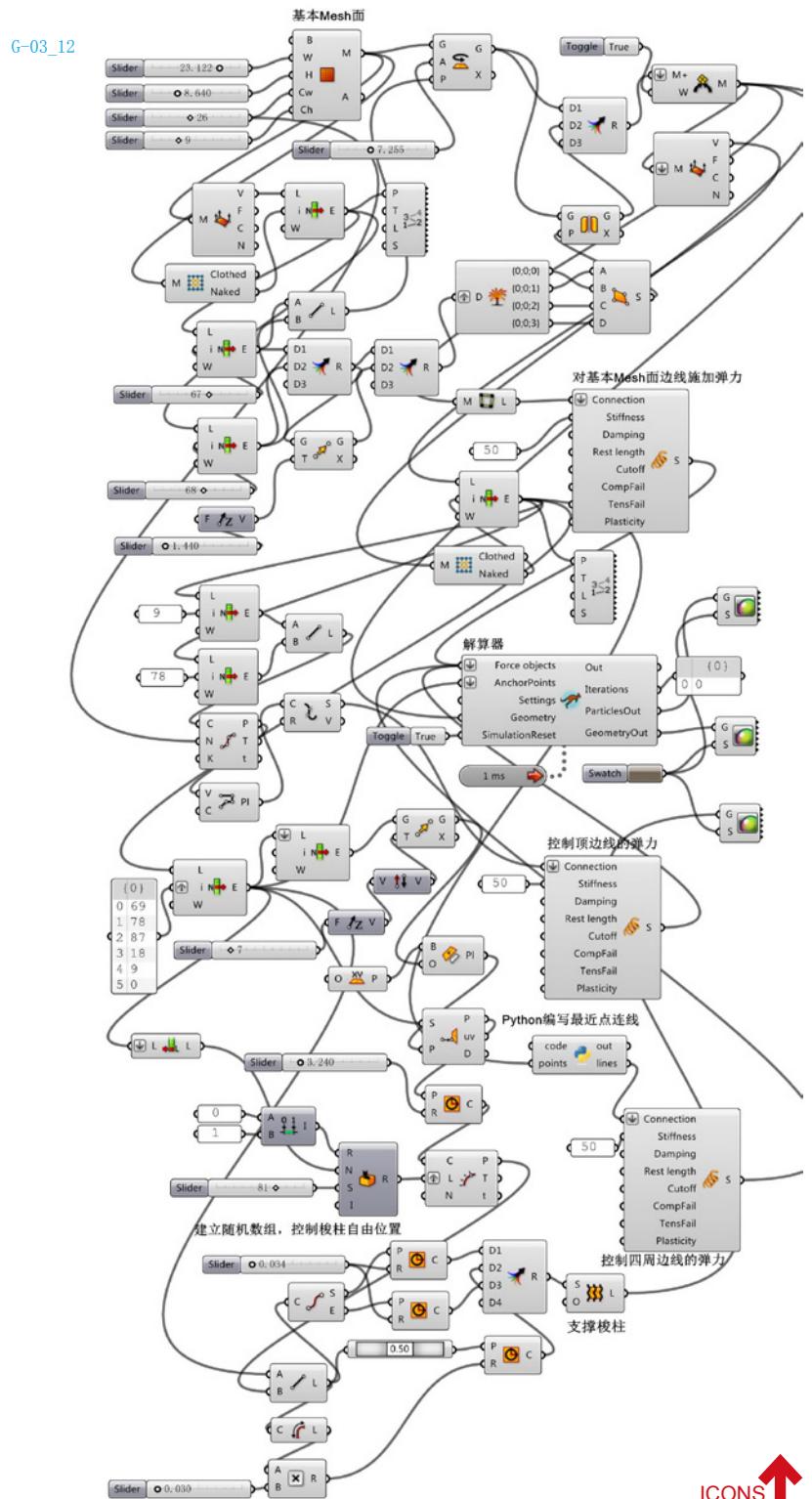
将连入AnchorPoints的6个顶点修改为四周一圈控制点：

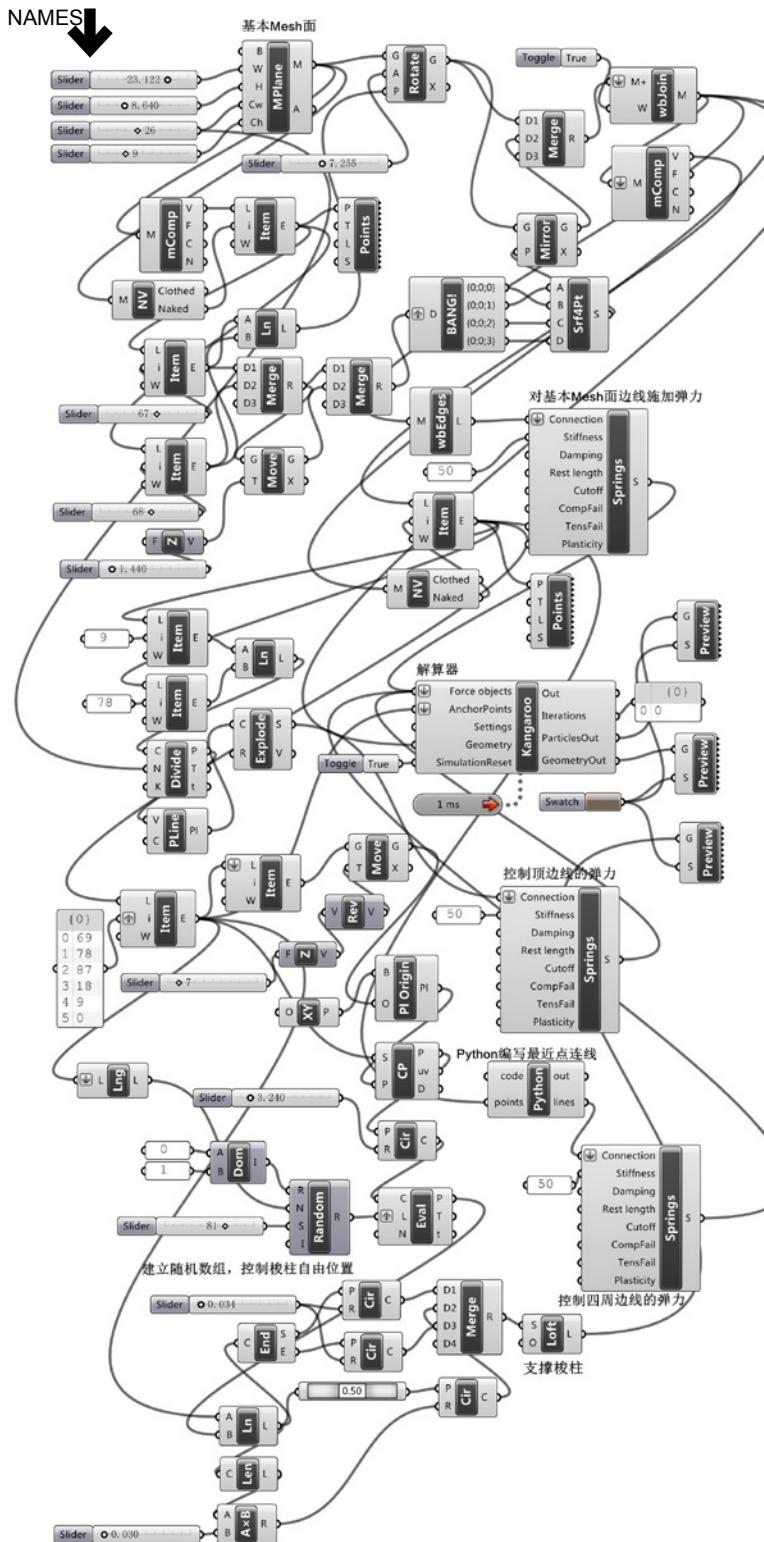


棱柱根部随机位置的变化

将顶部的固定点投影到地面，以地面的投影点为圆心画圆，使用随机数组随机获取圆上一点。







Python编写最近点连线:

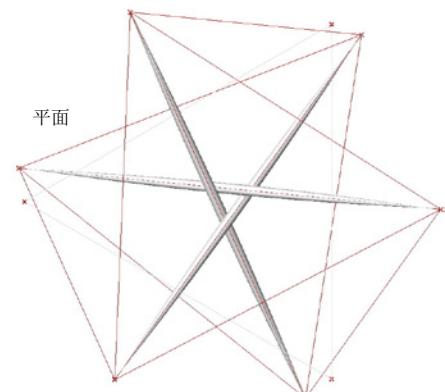
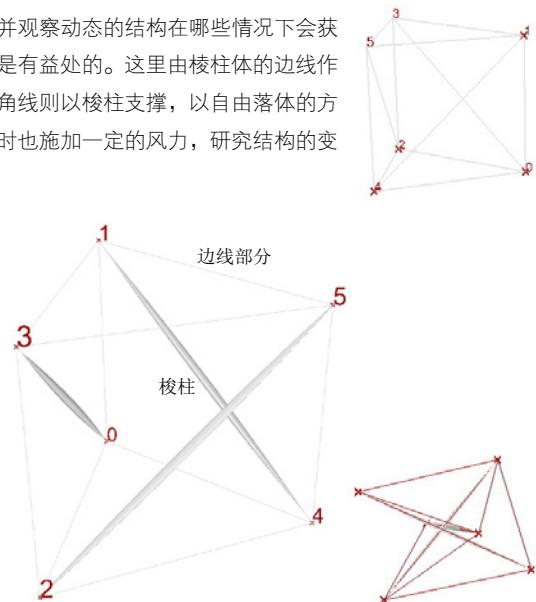
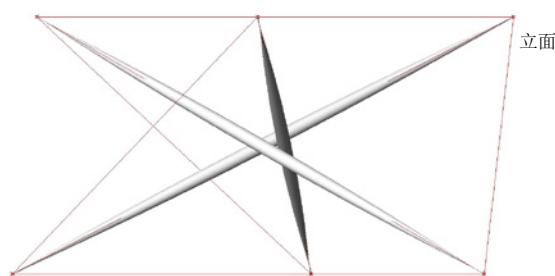
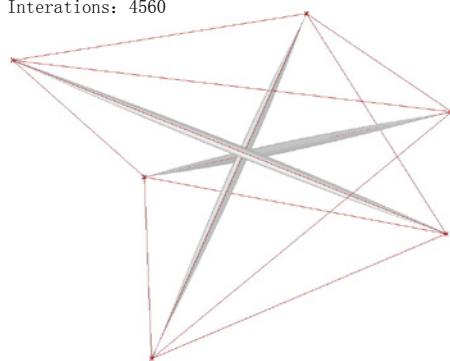
```
import rhinoscriptsyntax as rs
points=points
print(points)
sorted=rs.SortPointList(points) #对输入的点按最近距离排序
line=rs.AddPolyline(sorted)#用排序后的点连线
lines=rs.ExplodeCurves(line)#炸开连线为各个小的线段
```



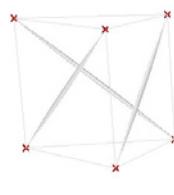
连锁杆件

使用动力学可以模拟多种情况来协助设计的推敲，并观察动态的结构在哪些情况下会获得各方力的平衡而静止，这对于结构本身的合理性判断是有益处的。这里由棱柱体的边线作为设计的主要结构，对其施加弹力模拟绳索部分，而对角线则以棱柱支撑，以自由落体的方式使其坠落，并与地面发生碰撞，观察其运动情况，同时也施加一定的风力，研究结构的变化情况。

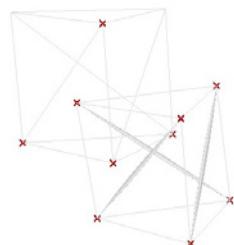
```
绳索部分Stiffness: 50
棱柱部分Stiffness: 1000
重力: 有
风力: 无
绳索部分RestLength:=初始长度
Iterations: 4560
```



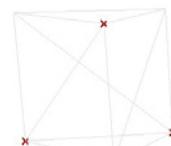
Iterations: 0



Iterations: 110



Iterations: 320



绳索部分Stiffness: 50

梭柱部分Stiffness: 1000

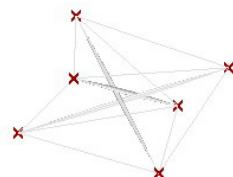
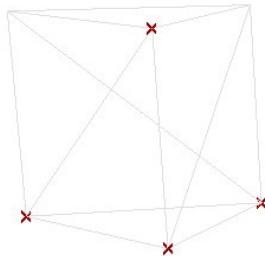
重力: 有

风力: 有

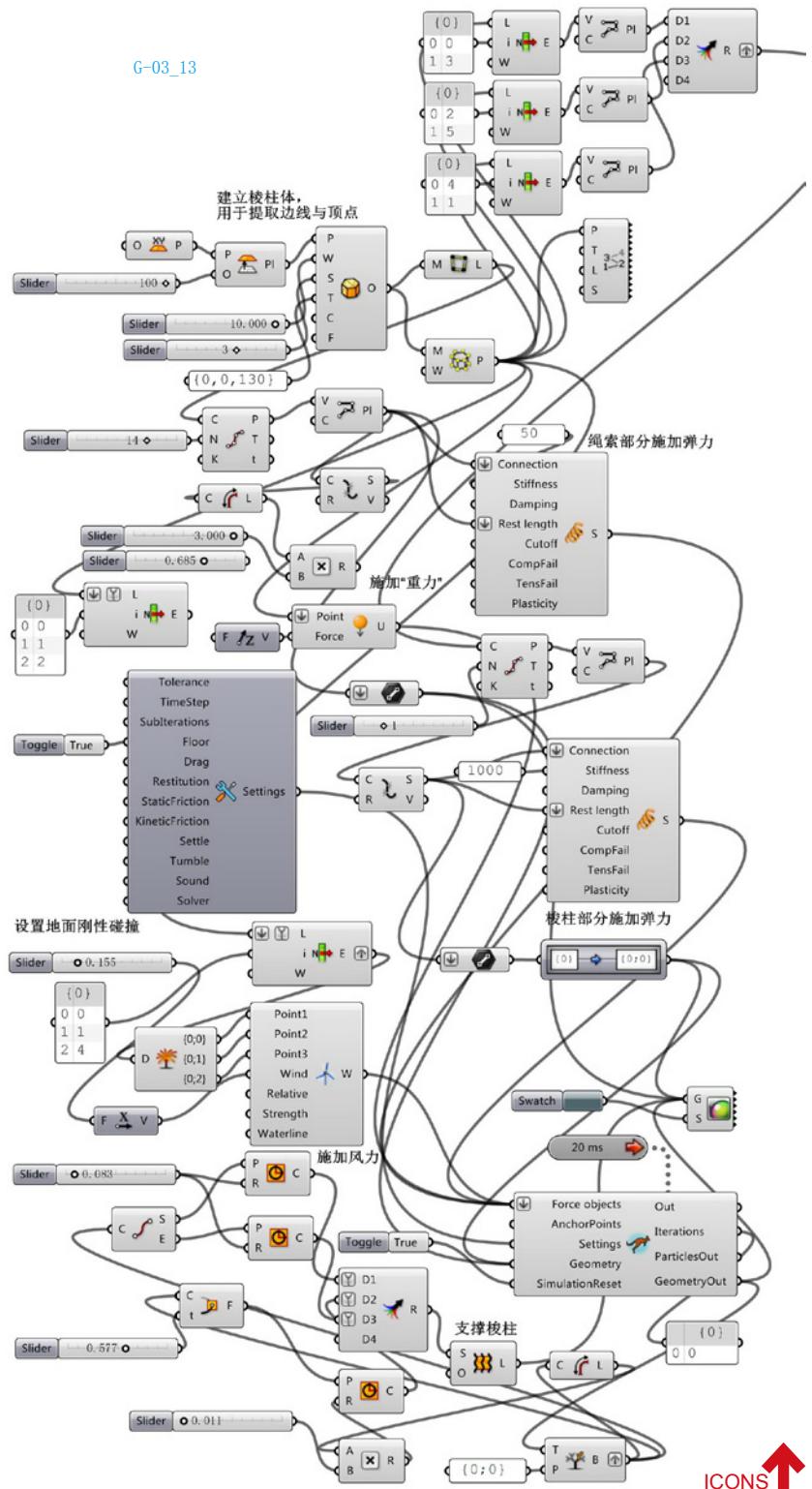
绳索部分RestLength:=初始长度



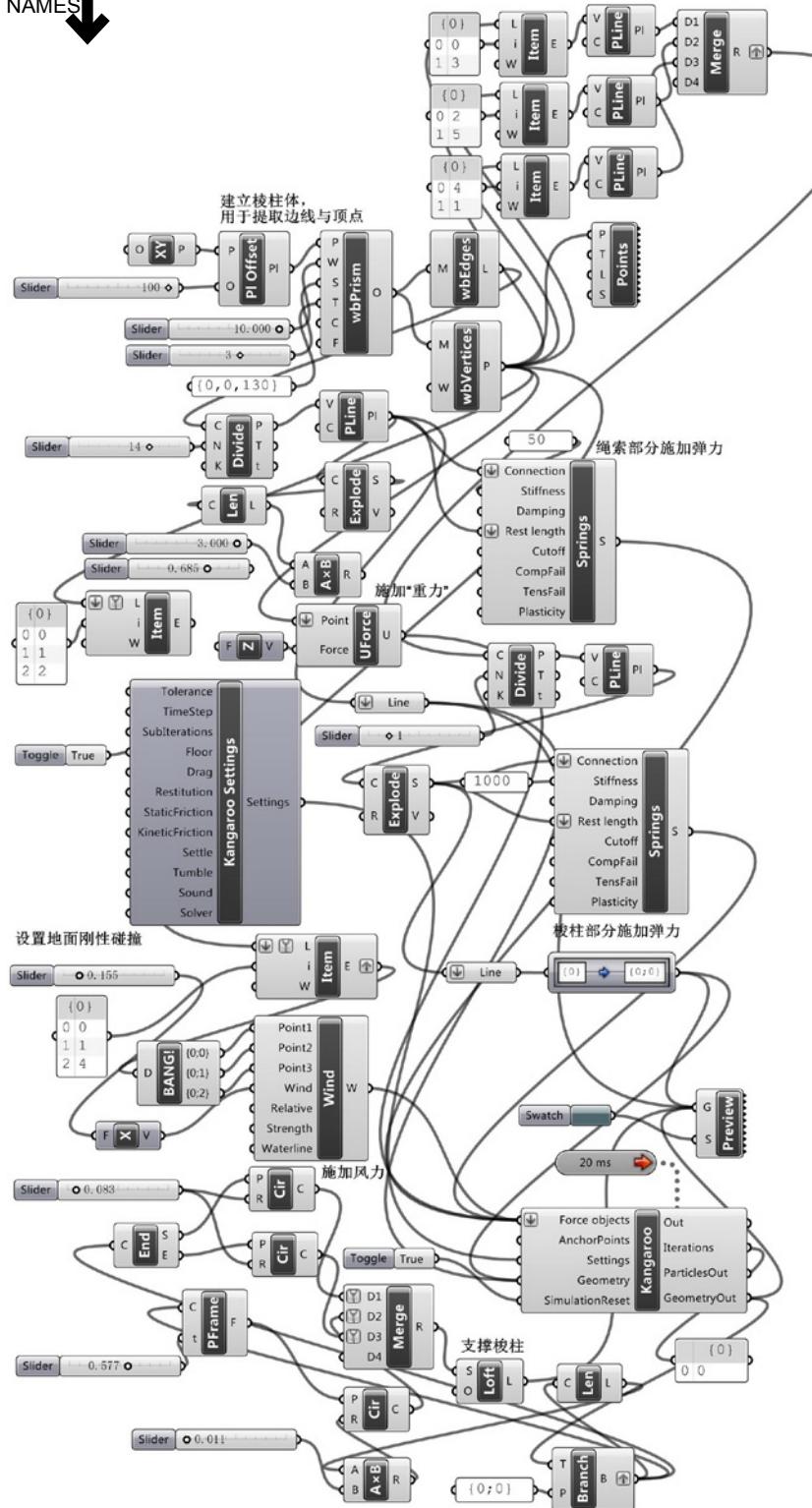
Iterations: 2540



G-03_13



NAMES

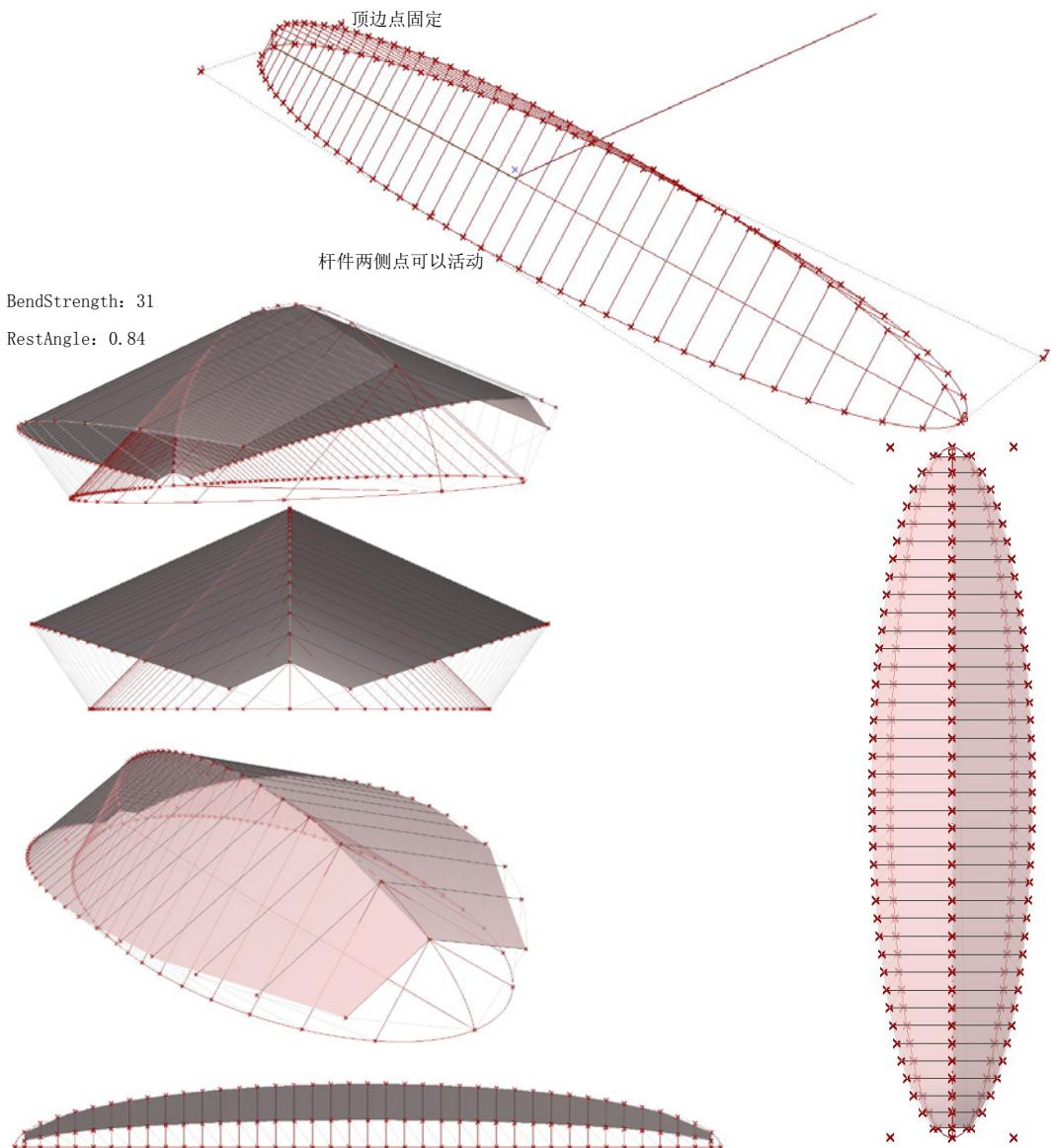


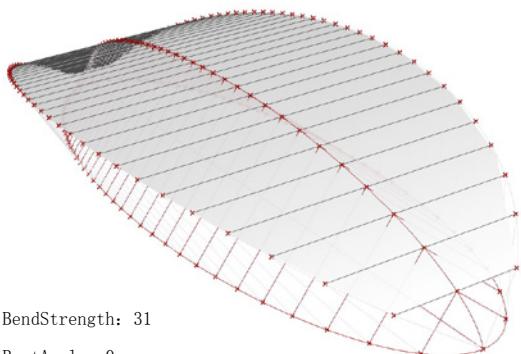
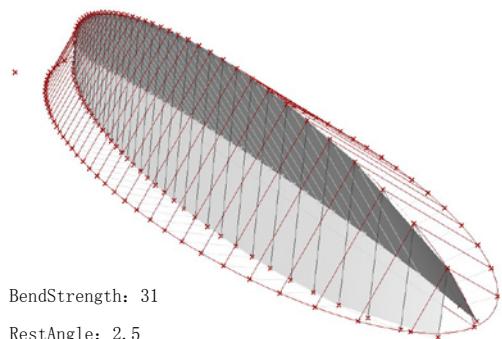
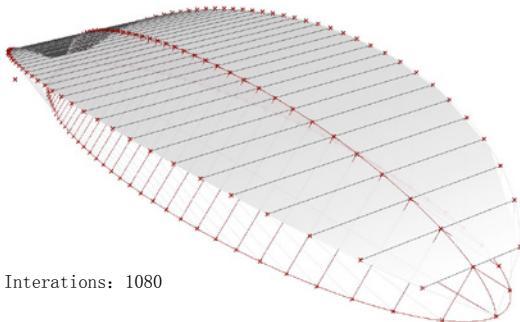
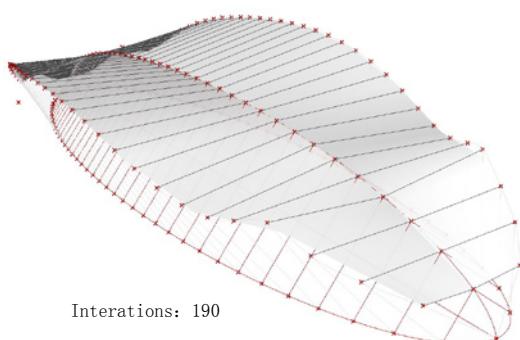
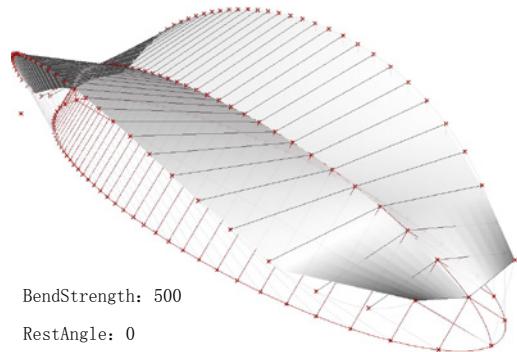
CALATRAVA

圣地亚哥·卡拉特拉瓦（Santiago Calatrava）具有建筑师和工程师的双重身份，他对结构和建筑美学之间的互动有着准确的把握，他认为美态能够由力学的工程设计表达出来。在他的诸多著名的作品中都洋溢出一种动力的美感，似乎是在捕捉舞者，或者鸟儿舞蹈与飞翔的瞬间，反映在具体建筑的几何形体上就体现出每个构件似乎都可以折叠，并且可以不断地变化。

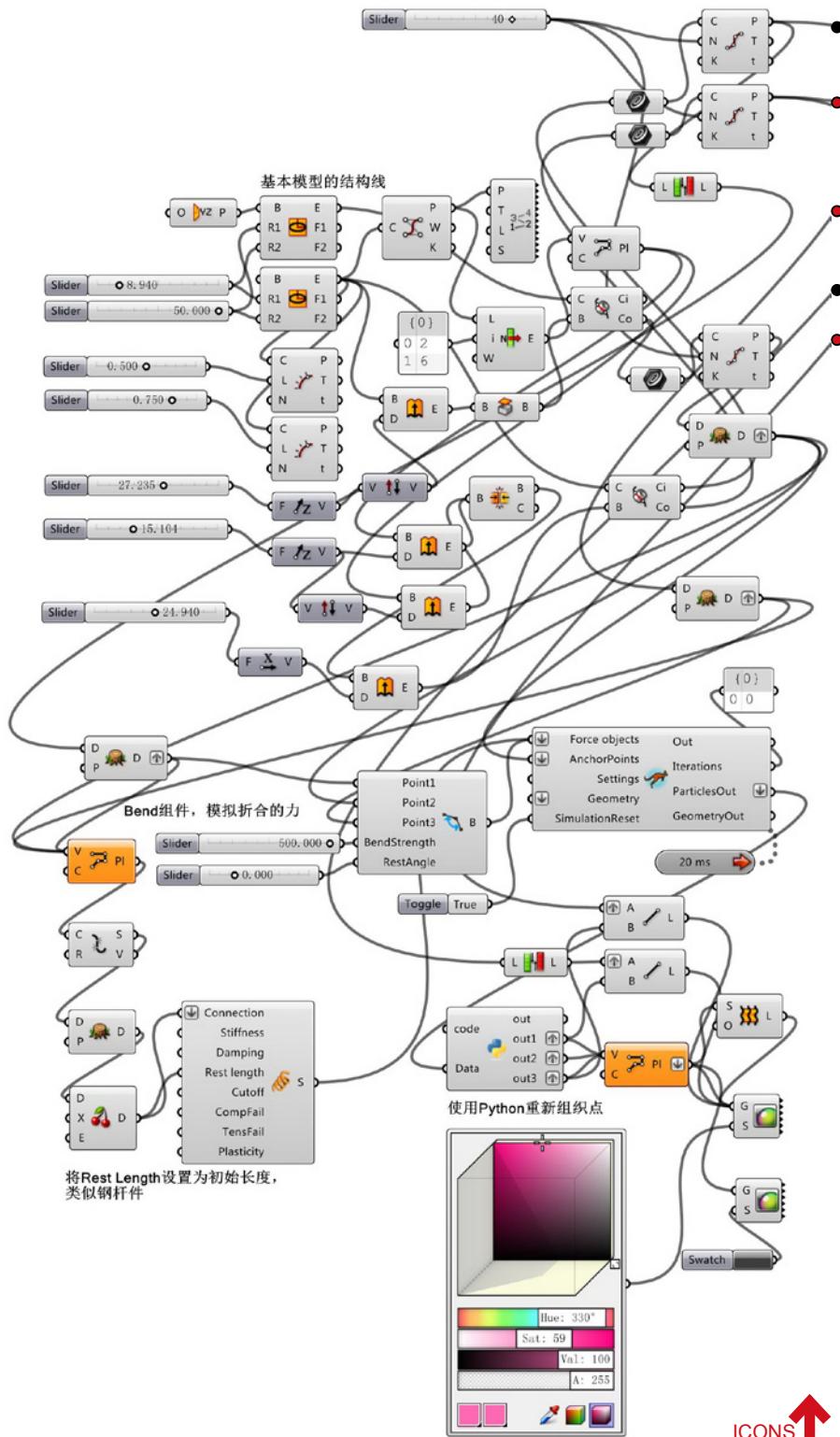
在没有计算机动力模拟情况下，作为研究该类模型的手段，直接的方法就是实际模型的搭建，而到目前所有这些都可以在计算机中完成，这也为设计师涉足具有动力美感的几何领域予以技术上的支持。

Kangaroo的部分组件可以协助设计师进行此类的初步模拟，本例以基本几何体椭圆为基本结构，使椭圆部分间的连线作为活动的杆件，固定最顶一排的点，使杆件两侧可以活动，设置时使用了Springs组件，将Rest Length值与初始长度值一致，使其长度不发生改变，并使用Bend折合组件改变折合角度来捕捉杆件折合时瞬间的动态。



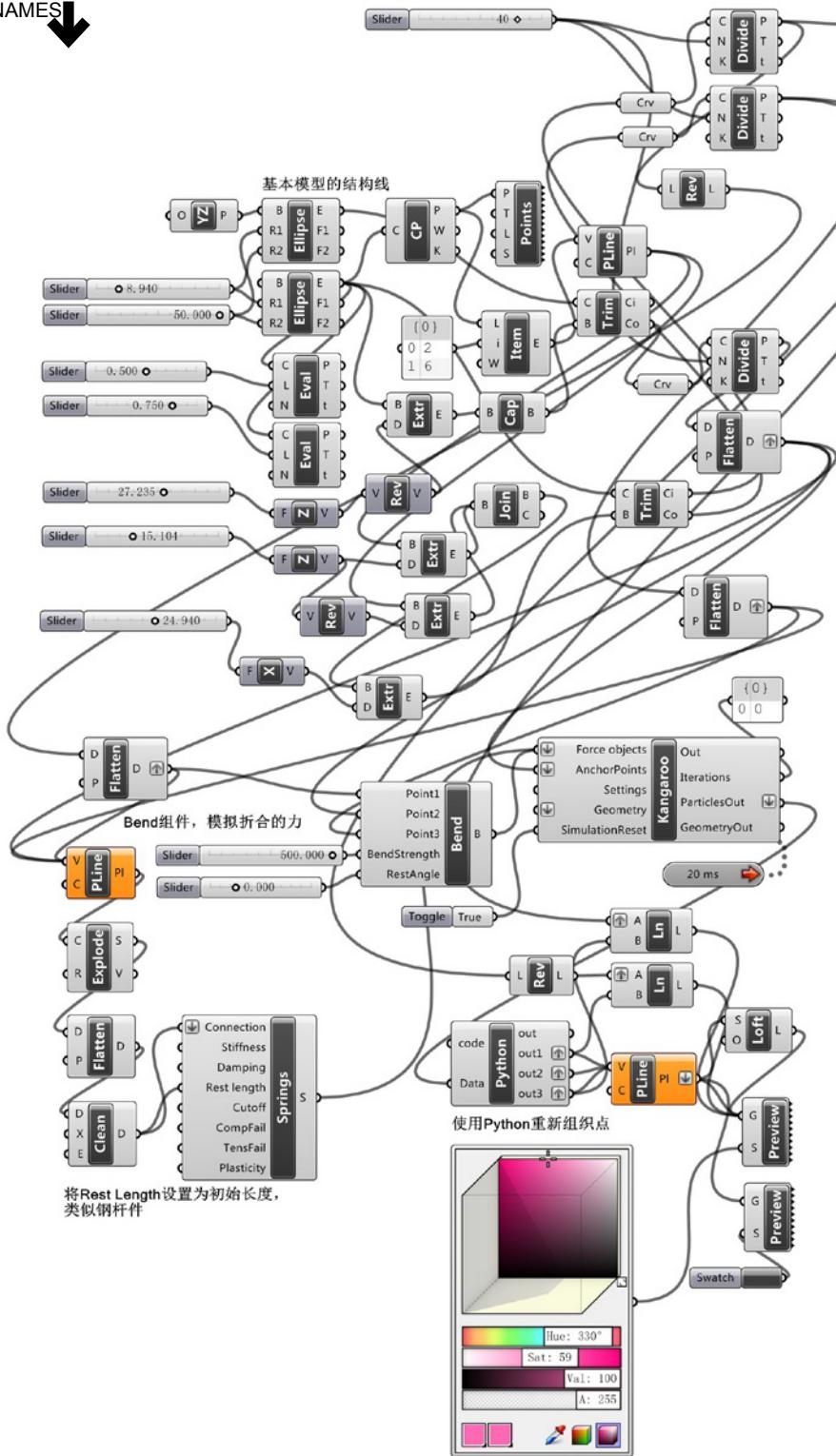


G-03_14

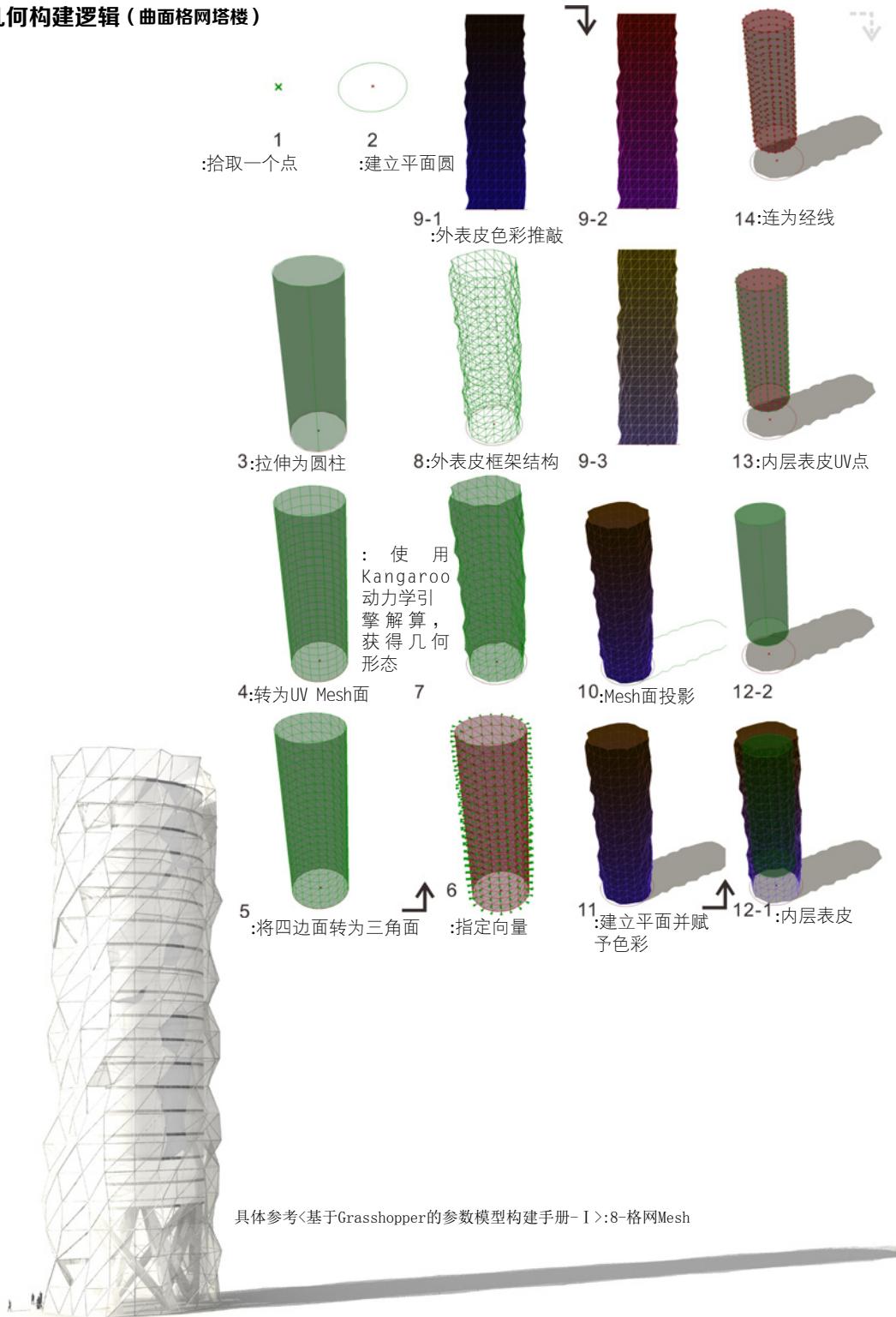


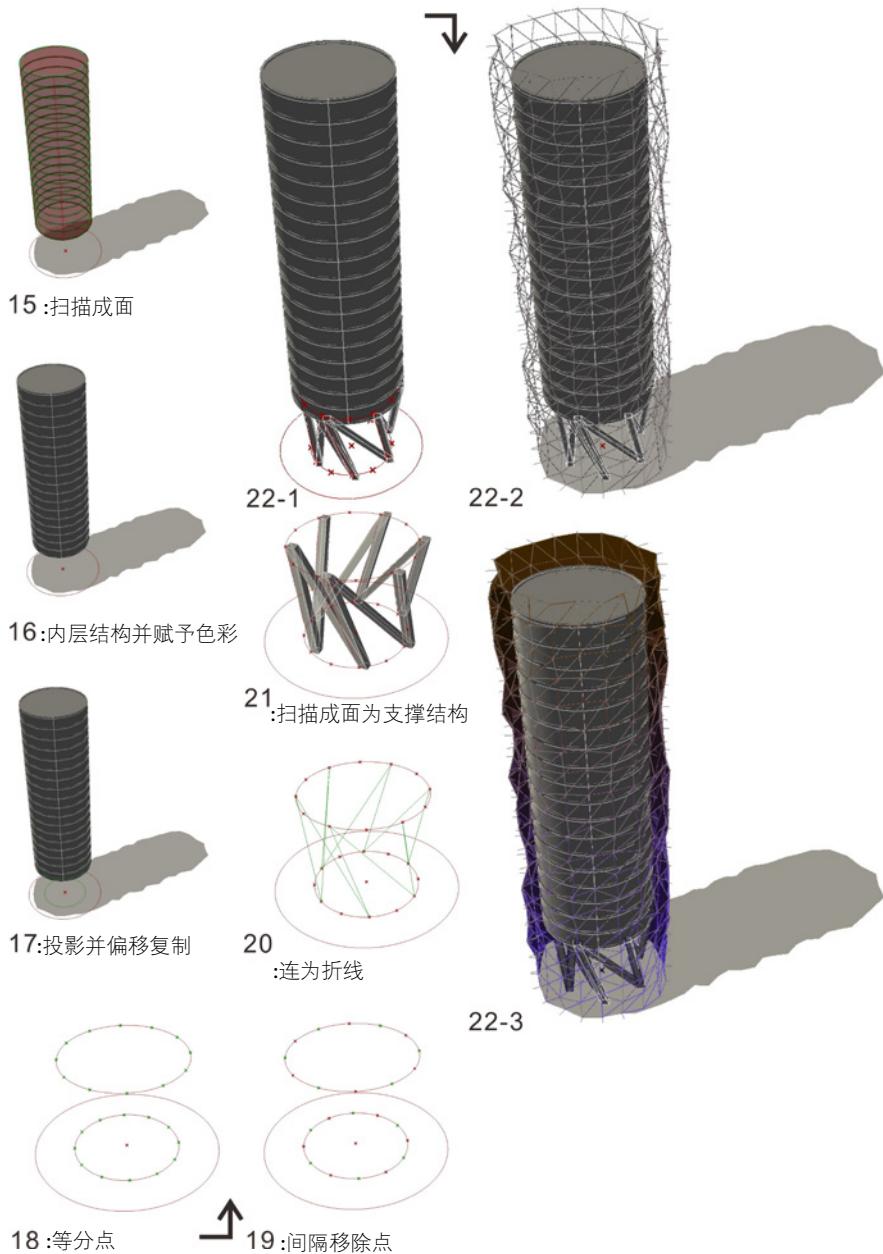
NAMES

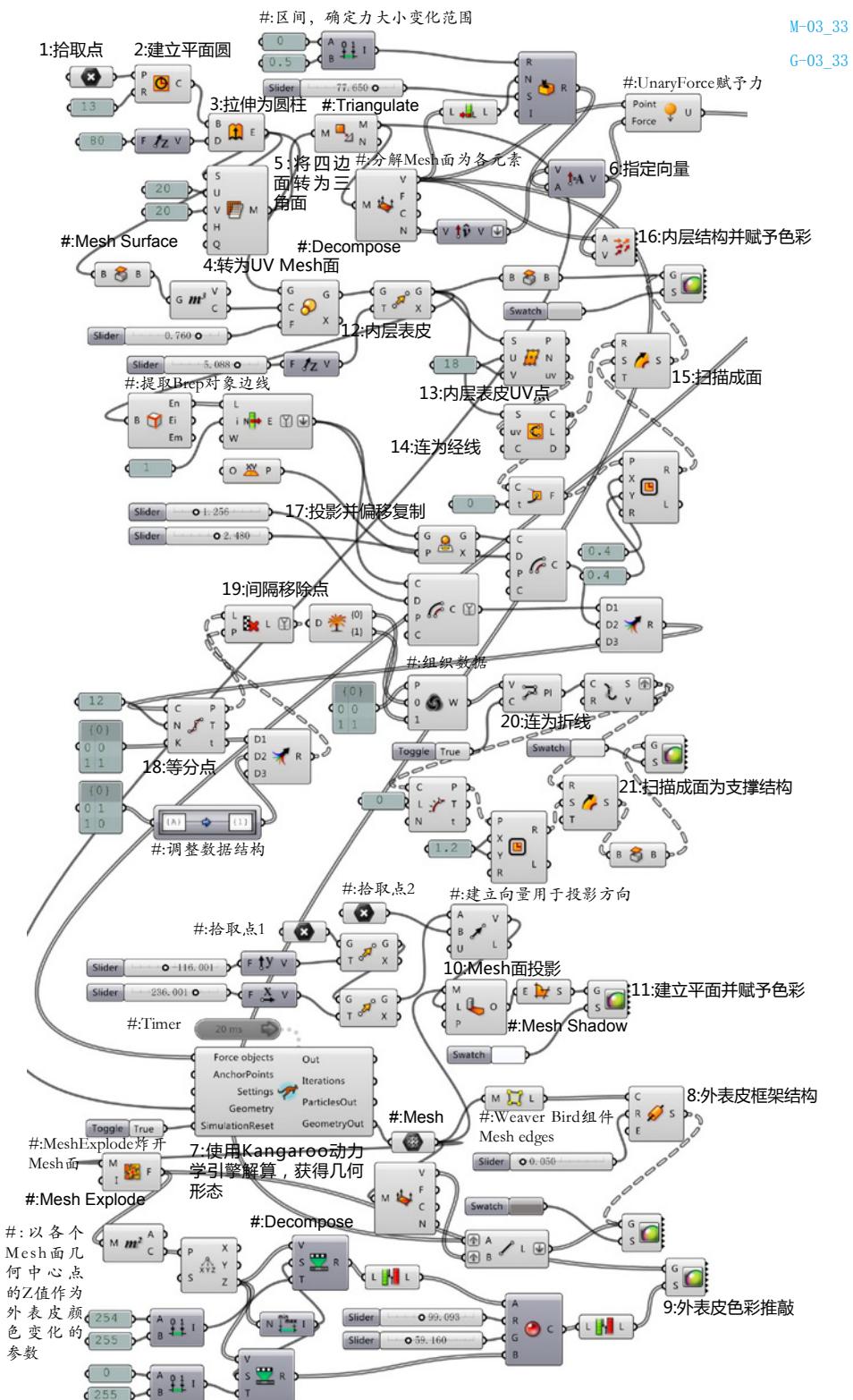




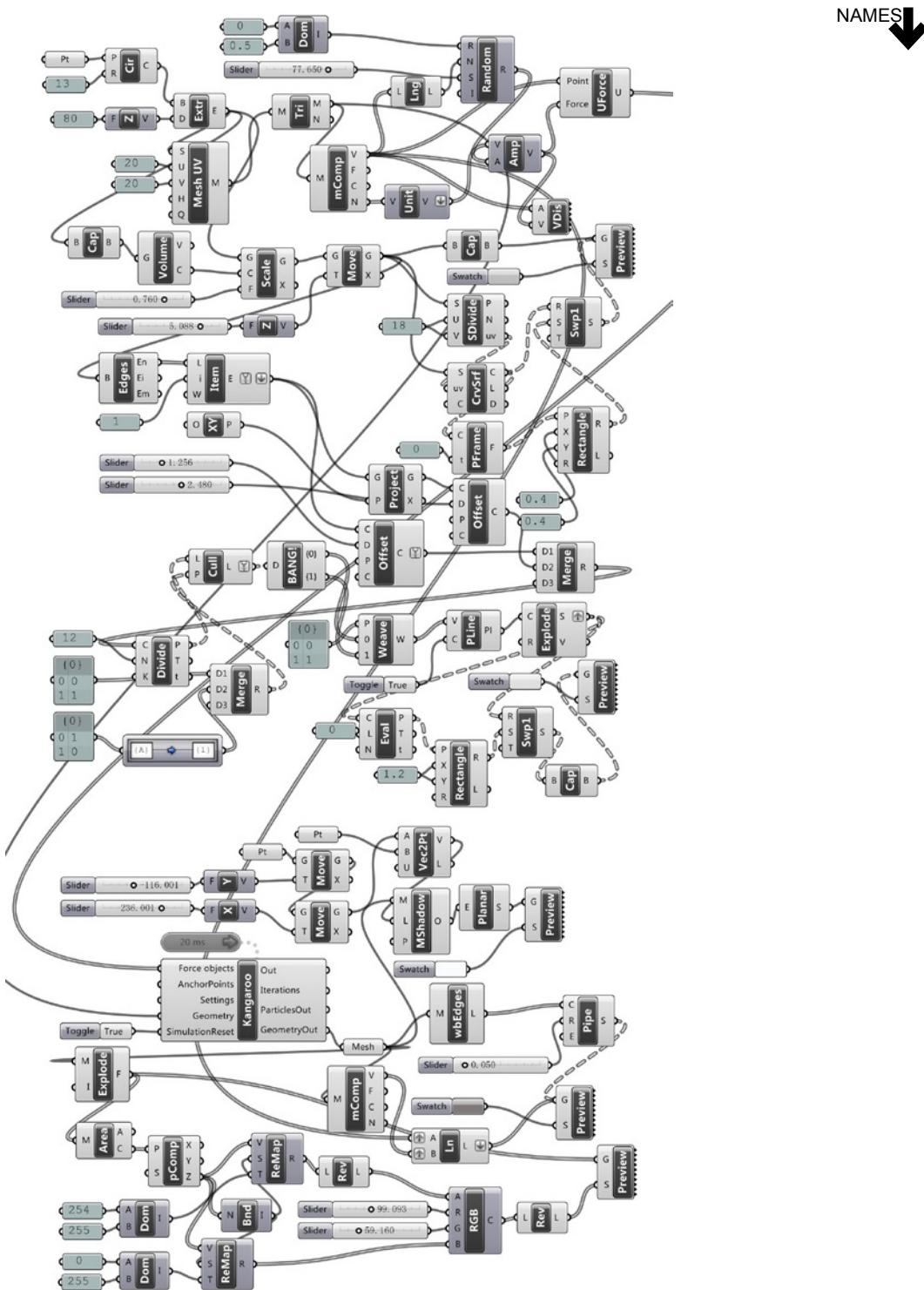
几何构建逻辑 (曲面格网塔楼)







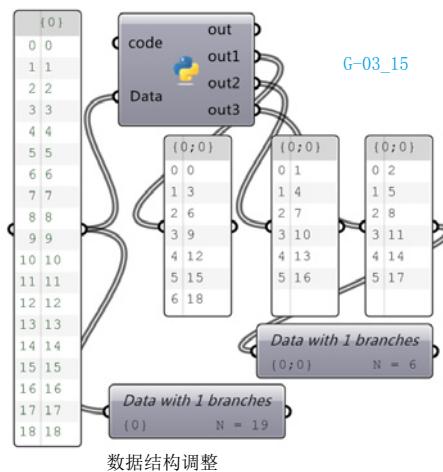
↑ ICONS



```

import rhinoscriptsyntax as rs
data=Data
out1=[]
out2=[]
out3=[]
length=len(data)
print(length)
n1=0
n2=1
n3=2
while n1<length:
    out1.append(data[n1])
    n1+=3
    print(n1)
while n2<length:
    out2.append(data[n2])
    n2+=3
    print(n2)
while n3<length:
    out3.append(data[n3])
    n3+=3

```



G-03_15

4 结构协同

结构部分一般是由结构师完成，并且在项目开始，对设计已经有了初步概念的时候，结构师就会联手建筑师或者景观设计师共同探讨概念几何与实现结构的问题，确保其设计的合理性。

就Grasshopper (GH) 而言，已经基于此平台开发了一些协助设计师分析结构方面的工具，但是因为计算方法，权威性等问题，并不能以此为根据进行最终结构施工图的依据，基于GH的结构分析目前仅能作为设计师初步的结构判断来协调建筑（构筑）设计。

Karamba

<http://twl.uni-ak.ac.at/karamba/> 官方网址



Karamba是基于GH的有限元Finite Element结构分析，主要是静力结构分析，能够计算在外力作用下，梁结构三个方向上的受力情况。Karamba完全被嵌入到GH的参数化环境中，因此能够轻松地联合参数化模型构建，同时Karamba的优化计算方法类似于GH的组件Galapagos。

Karamba目前并不能承诺其计算的完全正确性，在使用的过程中可能会出现错误，因此使用Karamba进行结构计算会带有风险。但是使用Karamba进行优化设计，以及结构比较，能够一定程度上成为设计师研究分析结构的有效手段。



Karamba各组件单元说明：

Algorithms: 算法，静力模型Statical Models的计算组件；

Cross Section: 截面，建立和选择截面类型；

Ensemble: 集合，通过确定参数的设置建立梁Beam模型，以及预位移Preliminary Displacement和影响结构计算各种参数的集合组件Assemble等；

Loads: 载荷，设置外力组件；

Materials: 材料：定义材料属性；

Params: 参数，静力模型对象的类；

Results: 结果，拾取计算结果；

Utils: 工具，辅助结构力学模型建立的额外组件；

ZZZ-Deprecated: 包含早些Karamba的组件，能够使使用早期版本的组件在当前版本中正确加载；

Karamba基本模型建立与计算流程：

1–在RHINO或者GH中建立点线几何用于结构分析模型；

2–将点线几何转化为Karamba中的梁Beams；

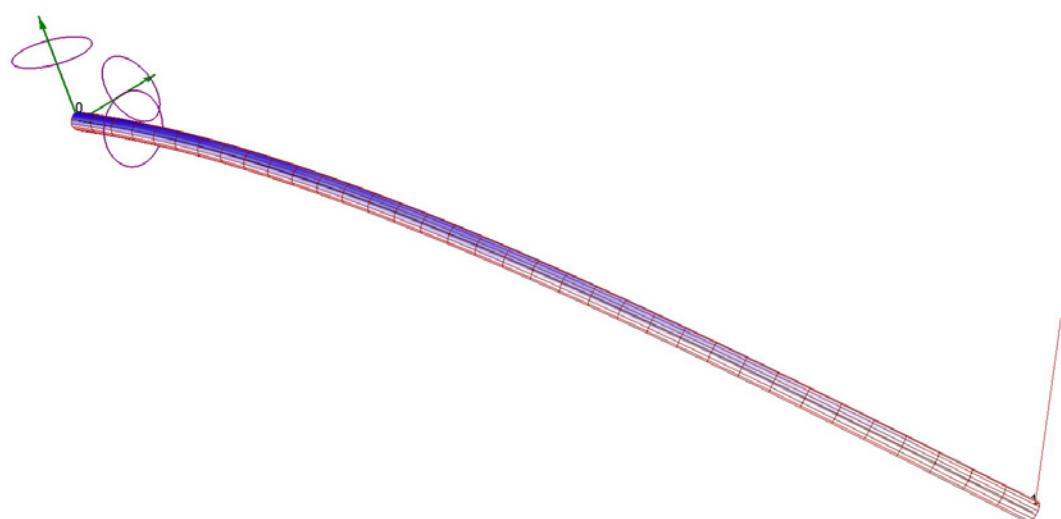
3–定义约束Suppors以及载荷Loads；

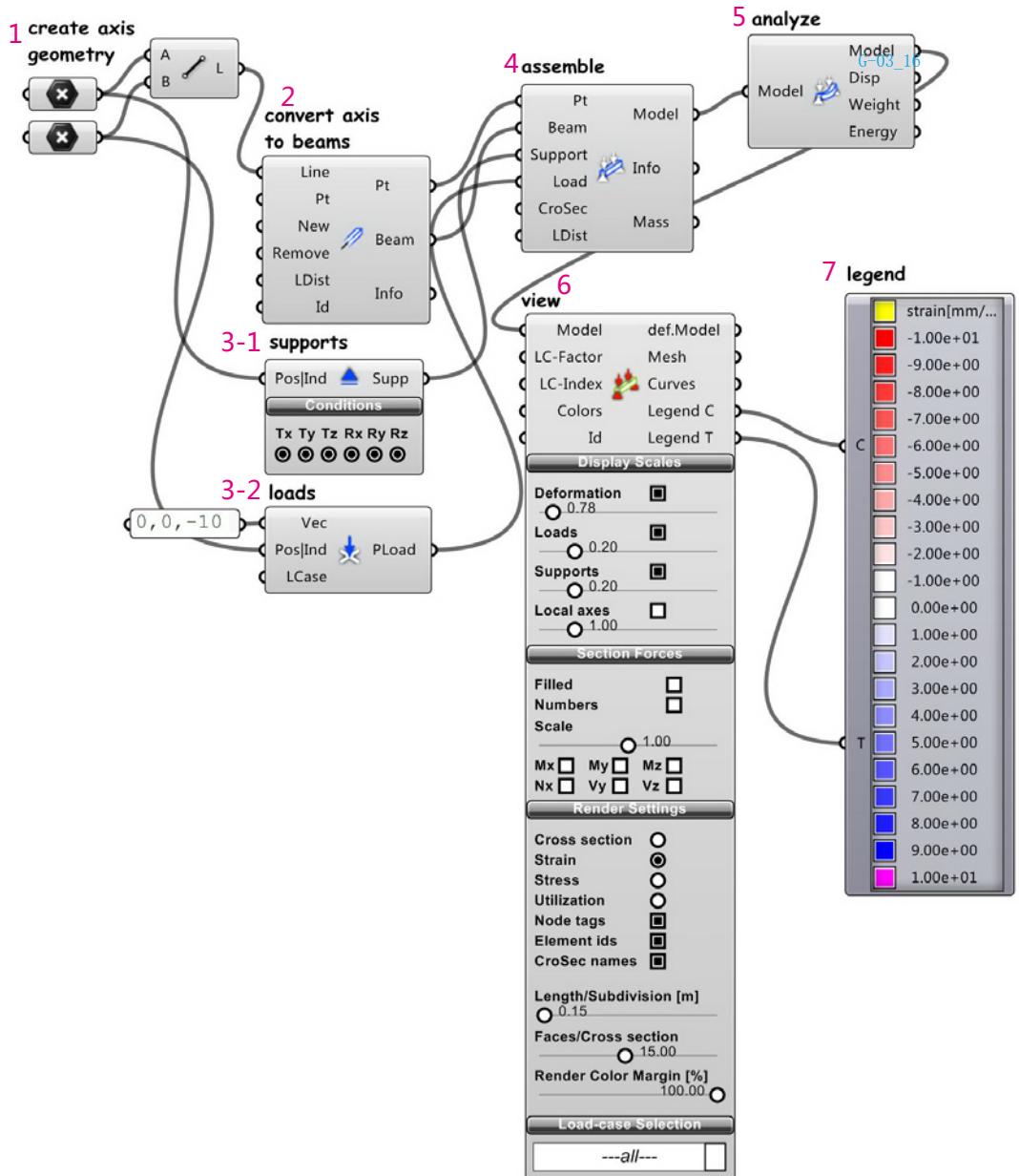
4–将影响结构分析的各项参数点、梁、约束与载荷进行集合；

5–解算；

6–显示分析结果。可以设置比例的变形（偏离）Deflections，压力Stress，拉力Strain等。对于多个载荷可以单独或者一起显示其分析结果。

7–可以显示图例；





进行结构力学模拟需要两部分知识结构，一个是**材料力学**：用于研究单个构件的内力（应力），变形（强度、刚度、稳定）；二是**结构力学**：研究由构件组成的体系，确定各构件的内力。

材料力学基础

A: 拉伸与压缩

1—力的内部效应

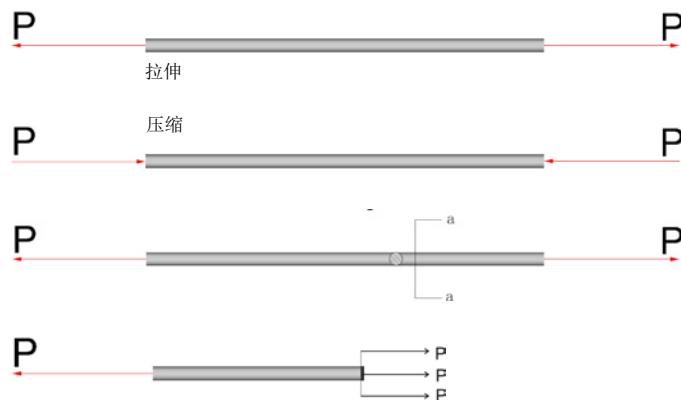
研究作用于物体（不再是完全刚性的）上力的内部效应，计算受力物体的形变是材料力学研究的基础课题。

轴向受载杆件：

简单情况：一等截面金属直杆在其两端承受一对共线，反向力的作用，两个力的作用线与各截面形心组成的纵向轴线重合，为了满足静力学平衡条件，这两个力的数值必须相等。如果两个力的方向是离开此杆的，杆件将承受拉伸；如果是指向此杆件的，则杆件承受压缩。

在这两个力的作用下，杆的内部将产生抗力。可以用位于杆轴某处，且与杆轴垂直的假想截面来研究杆件的内部抗力，如图a-a。为了分析计算，可以考虑将此截面右侧的杆段除去。因而，必须补充右侧杆段对左侧杆段的作用。用此处引入的截面法，初始的内力便成为保留杆段的外力。为使左侧杆段平衡，这种效应在数值上等于水平力P。然而，沿截面a-a法向作用的力P实际上是截面上法向分布力合成的结果。

为此，必须对这些力的分布方式作某些假设。通常，通过形心作用的力是假设均匀分布在横截面上的。



法应力：

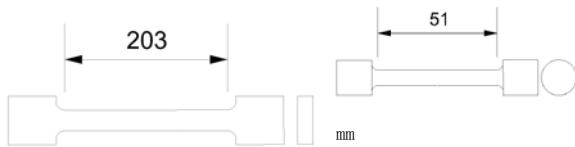
作用在某处小面积上的内力，最好转化为处理单位面积上法向力。单位面积上法向力的强度称为法应力，是用单位面积上作用力的单位表示，即N/m²。如果杆端的力使杆拉伸，杆内就产生拉应力，如果杆是受压缩的，杆内产生压应力。施加在杆端的力作用线必须通过每一个截面的形心。

试件：

轴向载荷经常出现在结构设计中，为了在实验室中模拟这种轴向载荷，试件应夹持在电子或液压万能实验机的夹头中。这两种试验机通常在材料实验室用作施加轴向拉伸载荷。

在材料试验技术规范标准中，美国材料实验室协会颁布了详细的使用说明。例如，一种厚度超过4.76mm的金属板试件，和一种直径为38mm的金属圆棒试件，试件的中部区域尺寸略小于两段尺寸。因而，破坏不会发生在夹持端，过渡圆角保证试件两种尺寸的过渡部位不会发生应力集中。供测量伸长的标距长度为203mm和51mm。

试件的伸长可以使用机械的，光学的引伸计，也可以使用粘贴在试件表面的电阻应变片来测量，这种电阻应变片由若干个沿试件轴向，很细的电阻组成。当试件伸长时，电阻丝的电阻就会发生改变，电阻的改变由惠斯通电桥测量并转换为伸长。



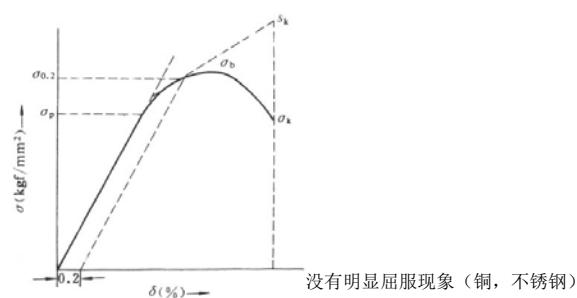
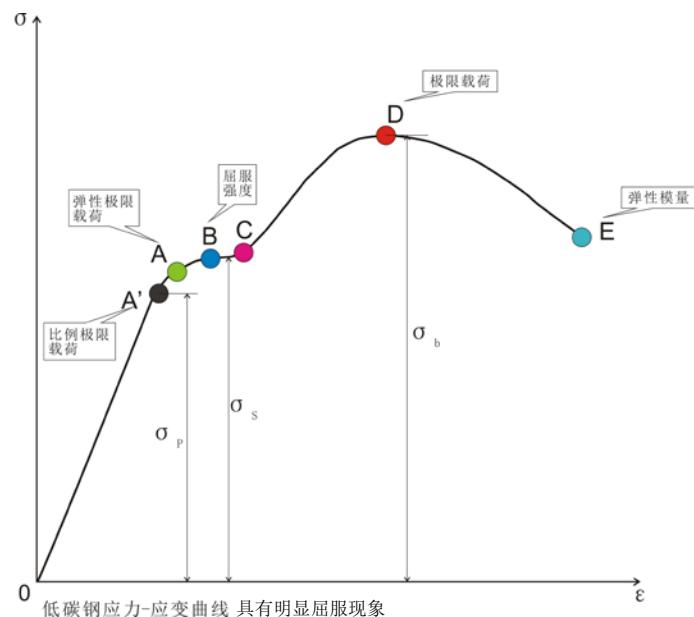
法应变:

假设一支试件已经装卡在拉伸-压缩试验机上，并且拉力逐渐的施加在试件的两端。对于每一个事先指定的轴向载荷增量都要测量相应标距的伸长。由这些数值得到单位长度的伸长，并将之定义为法应变，用 ε 表示，可以用标距长度L去除总伸长 Δ 来得到法应变，即 $\varepsilon = \Delta/L$ 。法应变通常用mm/mm为单位，无量纲量。

应力-应变曲线:

当轴向载荷以某种增量逐渐增长时，对于每一个载荷增量都要测量标距长度的总长度，直至试件断裂。已知试件的初始截面积，对于轴向载荷的任意值，可以得到用 σ 表示的法应力： $\sigma = P/A$

P 是以N表示的轴向载荷， A 是初始截面积。因此可以得到法应力 σ 与法应变 ε 组成的很多数据对，然后分别以法应力为纵轴，法应变为横轴，用上述实验数据对绘图，即材料的应力-应变曲线或拉伸图。对于不同的材料，应力-应变图有很多不同的形式。



弹性极限：在应力-应变曲线中，OA段为弹性变形阶段，此时卸掉载荷，试样恢复到原来尺寸。A点对应的应力为材料承受最大弹性变形的应力称为弹性极限，用 σ_p 表示。

比例极限：其中OA'部分为一斜直线，应力与应变呈比例关系，A'点所对应的应力为保持这种比例关系的最大应力称为比例极限，用 σ_e 表示。由于大多数材料的A点和A'点几乎重合在一起，一般不做区分。

弹性模量E：在弹性形变范围内，应力与应变的比值。弹性模量是材料最稳定的性质之一，它的大小主要取决于材料的本性，除随温度升高而逐渐降低外，其他强化材料的手段如热处理、冷热加工、合金化等对弹性模量的影响很小。

刚度：材料受力时抵抗弹性变形的能力，可以通过增加横截面积或改变截面形状的方法来提高零件的刚度。

强度与塑性

强度：材料在外力作用下抵抗变形和破坏的能力。根据加载方式的不同，强度指标有许多种，其中以拉伸试验测得的屈服强度和抗拉强度两个指标应用最多。

1. 屈服强度

屈服现象：应力超过B点后，材料将发生塑性变形。在BC段，材料发生塑性变形而应力不会增加的现象。

屈服强度：B点所对应的应力称为屈服强度，用 σ_s 表示。屈服强度反映材料抵抗永久变形的能力，是最重要的零件设计指标之一。

2. 抗拉强度

颈缩现象：CD段为均匀变形阶段。在这一阶段，应力随应变增加而增加，产生应变强化。变形超过D点后，试样开始发生局部塑性变形，即出现颈缩。

抗拉强度：随应变增加，应力明显下降，并迅速在E点断裂。D点所对应的应力为材料断裂前所承受的最大应力，称为抗拉强度，用 σ_b 表示。抗拉强度反映材料抵抗断裂破坏的能力，也是零件设计和材料评价的重要指标。

塑性：是指材料受力破坏前承受最大塑性变形的能力，指标为伸长率和断面收缩率。

伸长率：为试样拉断后，标距部分的残余伸长与原始标距之比的百分率。 $\delta = (l_1 - l_0) / l_0 * 100\%$ ， l_0 为原长， l_1 为断裂后长度。

断面收缩率：为试样断裂后，横截面积最大缩减量与原始横截面积之比的百分率。 $\phi = (F_0 - F_1) / F_0 * 100\%$ ， F_0 为试件原始截面积， F_1 为断口处的截面积。

韧性材料和脆性材料：

金属工程材料通常区分为韧性材料和脆性材料，在试件断裂之前，韧性材料具有较大的拉伸变形（如结构钢和铝），脆性材料具有较小的拉伸应变。法应变 0.05mm/mm 经常被取做这两类材料的分界线，铸铁和混凝土是典型的脆性材料。

胡克定律：

对于应力-应变曲线中，碳素结构钢，合金钢，硬质钢等，当应变相当小时，其应力，应变关系是线性的。轴向力和由它引起的伸长之间的线性关系（因为这些量各自和应力，应变只相差一个常数因子）是罗伯特·胡克（R·Hooke）首先于1678年发现的，成为胡克定律。为了描述材料的初始线性区，可以写作： $\sigma = E \epsilon$ ，其中E为应力-应变曲线中直线段OA'的斜率。

弹性模量 E：

E（亦即单位应力和单位应变之比）是材料拉伸时的弹性模量，又叫杨氏模量。各种工程材料的E值被制成表格，编入手册中。因为应变的单位是纯数，因而E的单位与应力的单位相同为N/m²。很多常用工程材料压缩时的弹性模量和拉伸弹性模量几乎相同。

2 - 材料的力学性质

应力应变曲线可以用来表示材料的几个强度指标（前文部分已述）

比例极限: (已述)

弹性极限: (已述)

弹性区和塑性区:

应力 – 应变曲线由原点至比例极限的范围叫做弹性区，由比例极限至断裂点的范围叫做塑性区；

屈服点:

应力 – 应变曲线中，B点的纵坐标叫做材料的屈服点，由该处开始，应变增长而应力不再增长，载荷增长到C点，材料发生屈服。某些材料的应力 – 应变曲线上存在两个应变增长而应力不增长的点，这些点分别叫做上屈服点和下屈服点；

极限强度或拉伸强度: (已述)

断裂强度:

E点的纵坐标叫做断裂强度；

回弹模量:

单向拉伸力逐渐由零增长到比例极限对应的数值，材料单位体积内所做的功被定义为回弹模量。此功可以用应力 – 应变曲线下由原点到比例极限的面积来计算，单位是N·m/m³ (SI)，因而，材料的回弹模量是材料在弹性区吸收能量的能力；

韧性模量:

单向拉伸力逐渐的由零增长到引起断裂的数值，材料单位体积内所做的功被定义为韧性模量。这个量可以用应力 – 应变曲线下由原点到断裂的面积来计算。材料的韧性模量是材料在塑性区吸收能量的能力；

断面收缩率: (已述)

延伸率(伸长率): (已述)

工作应力:

工作应力经常被选作屈服应力或极限应力除以某个安全因数，安全因数的选择依赖于设计者的识别力和经验。安全因数的规范一般在设计标准规范中可以找到；

应变强度:

如果韧性材料能够加载到超过屈服点而不失效，就说材料已被应变强化。很多结构钢都发生这种情况；

屈服强度:

当载荷加到某一数值后卸载，如果材料残留事先指定的永久变形，应力 – 应变曲线上与之对应的纵坐标叫做材料的屈服强度。这个事先指定的永久变形通常选为0.002或0.0035mm/mm；

切线模量:

应力相对于应变的变化率叫做材料的切线模量。这是由 $\epsilon_t = d\sigma / d\epsilon$ 给出的瞬时模量；

线膨胀系数:

线膨胀系数定义为，温度变化一度时，直杆单位长度的变化，它通常用 α 表示。线膨胀系数的值与长度的单位无关。但是与温度的标准有关。例如钢的膨胀系数为 $12 \times 10^{-6}/^\circ\text{C}$ 。温度变化如同加载一样，也会在结构内部产生应力；

泊松比:

当杆件受单向拉伸时，杆将沿载荷的作用方向产生伸长，但是，垂直于载荷的横向尺寸要产生缩短。横向应变与轴向应变之比叫做泊松比。通常用 μ 表示。对于大多数材料， μ 位于0.25~0.35的范围。

比强度:

这个量被定义为极限(或拉伸)强度与比重(单位体积内的重量)的比值。在SI系统中表示为：(N/m²) / (N/m³) = m，即，比强度为长度的单位。对于比较材料的效率性能，这个量是有用的；

比模量：

这个量被定义为杨氏模量与比重（单位体积内的重量）的比值。比模量单位为长度的物理单位；

3-动力效应

为了用拉伸或压缩试验测定材料的力学性能，加载速率有时会对测试结果有显著的影响。一般来说，韧性金属对加载速率的变化最敏感，而加载速率对脆性材料（如铸铁）的影响则可以忽略。例如，特别快速施加轴向力时，低碳钢的屈服点可以提高170%之多。然而此时，总形变与低速加载相比保持不变。

4-材料的分类

以上所有的讨论都是基于如下两个普遍采用的，关于材料特征的假设：

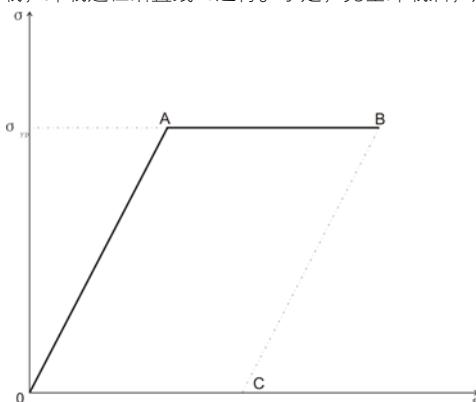
1-均匀假设，亦即物体内所有点都具有相同的弹性性质 (E, μ)；

2-各向同性假设，亦即物体内任一点的所有方向上都具有相同的弹性性质；

并非所有的材料都是各向同性的，如果材料不具有任何弹性对称性，称之为各向异性，有时也称之为偏等性。这类材料与各向同性材料不同，具有21个独立的弹性常量数，而不是2个独立的弹性常数。如果材料具有三个相互垂直的弹性对称平面，则称之为正交各向异性材料。这种情况下，独立的弹性常数有9个。例如，纤维增强复合材料是典型的各向异性材料。

5-弹塑性分析

在某些结构中，经常允许材料的应力和形变处于塑性范围，例如某些建筑规范允许个别构件发生塑性形变，此外飞行器和火箭结构的某些构件故意设计为在塑性区工作，以减少重量。此外，很多金属加工过程，本身就包括材料的塑性形变。对于低碳或中碳钢塑性小应变问题，应力-应变曲线通常理想化为两条直线，其中一条的斜率为 E ，表示弹性区；另一条的斜率为零，表示塑性区，称为理想塑性材料。如果产生了点B对应的应变，然后卸载，卸载过程沿直线BC进行。于是，完全卸载后，产生与应变OC对应的残留将永久形变。



材料	室温下常用工程材料的性质				20℃ 泊松比
	比重 kN/m ³	杨氏模量 Gpa	极限应力 Mpa	线胀系数 10 ⁻⁶ /℃	
1. 板，棒或块状金属试件					
铝合金	27	70~79	310~500	23	0.33
黄铜	84	96~110	300~590	20	0.34
紫铜	87	112~120	230~380	17	0.33
镍	87	210	310~760	13	0.31
钢	77	195~210	550~1400	12	0.3
钛合金	44	105~120	900~970	8~10	0.33
2. 板，棒或块状非金属试件					
混凝土	24	25	28~41	11	
玻璃	26	48~83	70	5~11	0.23
3. 丝状材料（晶须）试件（直径<0.025mm）					
氧化铝	38	690~2410	13800~27600		
碳化钡	25	450	6900		
玻璃		345	70000~20000		
石墨	22	980	20000		
4. 复合材料（沿载荷方向单向增强）					
环氧树脂	19	210	1365	4.5	
S-玻璃纤维增强环氧树脂	21	66.2	1900		
5. 其他					
石墨增强环氧树脂	15	104	1310		
Kevlar-49 环氧树脂	13.7	86	1520		

引自《金美经典学习指导系列-材料力学(第四版)》

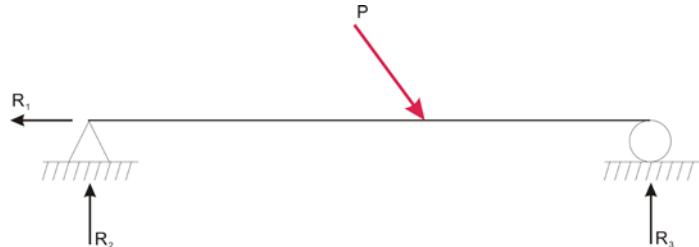
B: 超静定力系—拉伸与压缩

1—静定力系的定义

如果作用在物体上的所有外力，都能仅依靠静力学平衡方程解出，该力系就是静力的，称为静定力系。

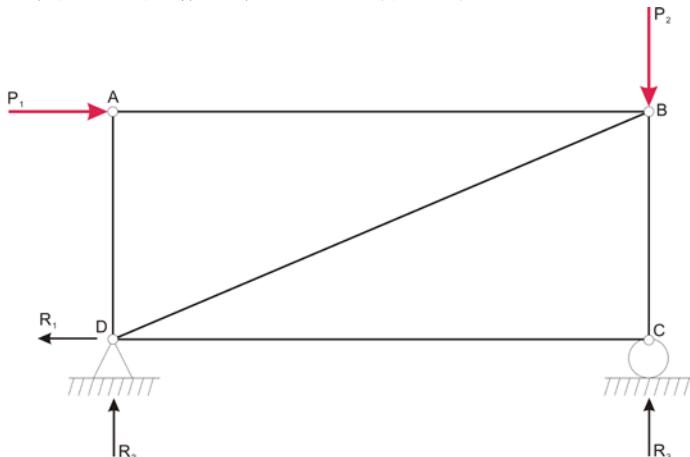
例-1：

承受力P作用的杆，约束反作用力为R1、R2和R3。此例中有三个独立有效地静力平衡方程，足以求解三个未知的约束反作用力，因而力系是静定的。



例-2：

桁架ABCD承受力P1和P2作用，约束反作用力为R1、R2和R3。同样有三个独立有效地静力平衡方程，可以求解三个未知的约束反作用力，因而力系是静定力系。

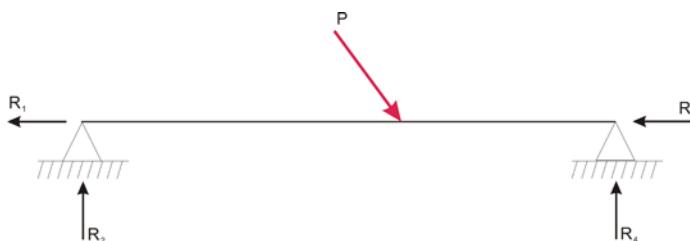


2—超静定力系的定义

在很多情况下，只依靠静力学平衡方程无法解出作用在物体上的所有力，原因是存在多于平衡方程数目的未知力，这种力系称为超静定力系。

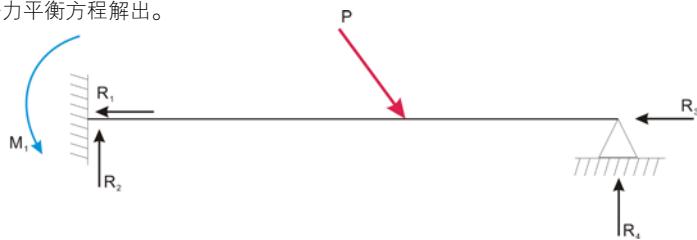
例-3：

杆件承受力P作用，约束反作用力是R1、R2、R3和R4。因为有四个约束反作用力而只有三个平衡方程，这样的力系称为一次超静定力系。



例-4:

杆件的外力系是两次超静定的，因为有五个未知的反力， R_1 ， R_2 ， R_3 ， R_4 和 M_1 。但是只有三个静力平衡方程，从而使约束反作用力不能依靠静力平衡方程解出。



C: 剪切

剪切力: 通过物体作一截面，作用在这个截面内的力称为剪切力，或剪力，用 F_q 表示；

剪应力: 剪切力除以其作用面积，所得结果称为剪应力，或切应力， τ 表示：

$$\tau = F_q/A;$$

剪应力是沿着与杆件轴线垂直平面作用的应力，而法应力是垂直于该平面的应力，因此两者具有不同的作用方向。

剪应变: 发生在原始矩形单元体角点处的角度变化定义为剪应变。它是用弧度（rad）来度量的，通常表示为 γ ；（切应变）

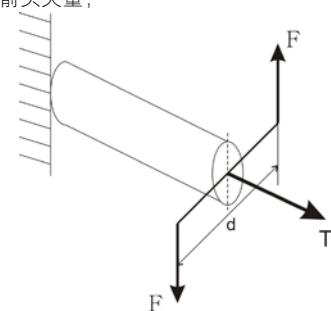
剪切模量: 剪应力 τ 和剪应变 γ 的比值称为剪切模量，通常用 G 表示，即： $G = \tau / \gamma$ ， G 也称为刚性模量。

G 的单位与剪应力相同，即 N/m^2 ，原因为应变是无量纲量。

焊接连接通常有气体焊接，电弧焊接工艺，另外还有电子束焊接和激光束焊接等。

D: 扭转

扭转: 考虑一个一端固定，另一端承受扭转力偶（转矩） $T=F_d$ 作用的杆件，该扭转力偶作用在杆轴垂直的平面内，这样的杆件处于扭转之中。扭转力偶的另一个表示方法是一个沿杆轴的双箭头矢量；



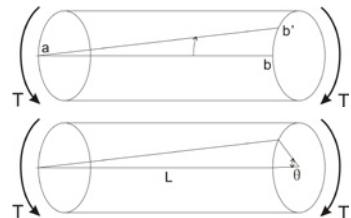
扭矩: 有时可能会有若干个力偶沿轴的长度作用，在这种情况下，为方便引入一个新的物理量—扭矩。任意一个截面内的扭矩等于其一侧作用在轴上的力偶矩的代数和，至于选择哪一侧，在所有情况下均是任意的；

极惯性矩: 对于外径 D_0 ，同心内孔直径 D_i 的空心圆轴，其横截面的极惯性矩通常用 J 来表示，计算公式为 $J = (\pi/32) (D_0^4 - D_i^4)$ ，实心圆轴的极惯性矩可由令 $D_i=0$ 得到，极惯性矩 J 是横截面的几何（数学）性质。通常会在研究圆轴扭转应力时用到；

扭转剪应力: 无论实心圆轴还是空心圆轴，扭转剪应力 τ 表示为： $\tau = T \rho / J$ ，式中 ρ 是计算剪应力处与圆轴中心的距离。注：轴内任一点处的剪应力均不应超过材料（剪应力）的比例极限；

剪应变: 如果在未加载的圆轴上沿母线作一标记直线 $a-b$ ，那么在扭转力偶矩 T 作用之后，线 $a-b$ 将运动至 $a-b'$ 处，在圆轴表面上用弧度度量的，母线起始及终止位置之间的角度 γ ，定义为圆轴之上的剪应变。同样的定义也适用于圆轴内部的各点；

剪切模量：剪应力与剪应变的比值叫做剪切模量， $G = \tau / \gamma$ ；



扭转角：如果长度为L的圆轴，在其两端受到常量力偶T的扭转，则其一段对另一端的扭转角 θ 为， $\theta = TL/GJ$ ，其中J是截面的极惯性矩；

E: 剪力和弯矩

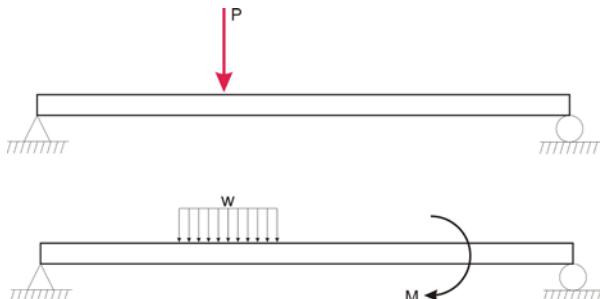
梁：一个杆件承受位于包含其轴线的纵向平面内的力和力偶的作用，承受的作用力应垂直于杆件纵向轴线；

悬臂梁：如果梁只在一端支承，并且由于此种支承而使梁的轴线不能在支承点处转动，这种梁称为悬臂梁。梁的一端可以自由挠曲，另一端完全固定，称为固定端，其支承壁对梁的约束反力包括铅垂力和作用在加载平面内的力偶；

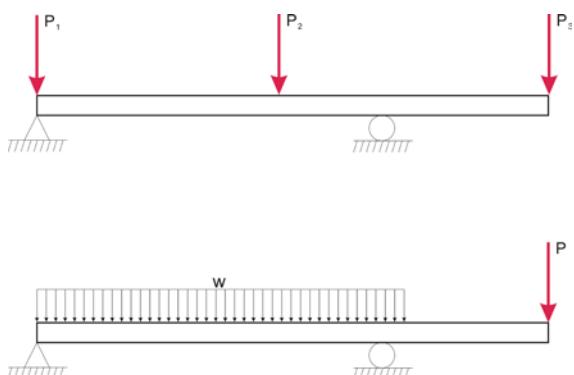
简支梁：在其两端自由支承的梁，此处所谓的“自由支承”，系指端部对梁的支承只提供约束反力而不能提供任何约束反力偶。因此当梁在载荷作用下发生挠曲时，支承处不会限制端部的转动；

简支梁至少有一个支承是能够发生水平位移的，因此在梁内不会产生轴向力。

如果没有任何一端对于水平位移是自由的，那么在受载荷作用下变形时，就会产生某些轴向力；

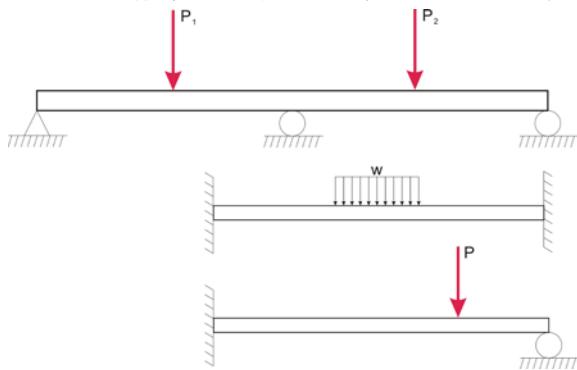


外伸梁：一个梁自由支承在两点处，并且由一端或两端延伸到这些支承之外，这样的梁称为外伸梁；



静定梁：悬臂梁、简支梁、外伸梁等，其约束反力都可以利用静力学平衡方程计算得到，这些约束反力的数值和梁的形变无关，这样的梁称为静定梁；

超静定梁：如果作用在梁上的约束反力数目超过静力平衡方程的数目，那么必须用基于梁变形的方程去补充静力平衡方程。在这种情况下，梁称之为超静定的；



载荷的类型

载荷是作用在结构上的主动力。载荷作用的范围和分布情况，通常将其简化为分布载荷和集中载荷。分布载荷是指连续分布在结构某一部分上的载荷，它又可分为均布载荷和非均布载荷。例如等截面杆件的自重即可简化为沿杆长作用的均布载荷；池壁上的水压力和作用在挡土墙上的土压力，均可简化为集度按直线变化的非均布载荷，又称线分布载荷。集中载荷是指作用在结构上某一点的载荷，当实际结构上分布载荷的受载区域尺寸远小于结构的尺寸时，为了计算方便，即可将此区域内分布荷载的总和视为作用在区域内某一点上的集中载荷。

作用于结构上的载荷，按其作用时间，可以分为恒载荷活载。恒载是指永久作用于结构上的载荷，如自重，结构上固定设备的重量等。活载是指暂时作用在结构上且位置可以变动的载荷，如结构上临时设备，人群和移动吊车的重量，以及风力、雪重、水压力等。恒载作用下结构强度的计算可以通过内力分析进行，而对活载，还要涉及影响线和包络图的概念。

根据载荷作用的性质，又可以分为静力载荷和动力载荷。静力载荷是指逐渐增加，不致使结构产生显著冲击或振动，因而可略去惯性力影响的载荷。恒载和上述大多数活载都可视为静力载荷。动力载荷是指作用在结构上，对结构产生显著冲击或引起其振动的载荷，在这类载荷作用下，结构将会发生不容忽视的加速度。例如动力机械的振动，爆炸冲击，地震等引起的载荷就是动力载荷。

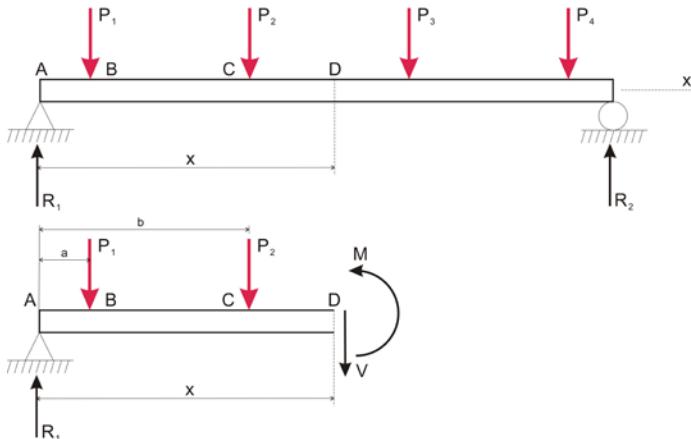
注：结构除了承受载荷外，还可能受到其它外在因素的影响，如温度变化，支座位移，材料收缩等，这些因素也会对结构的受力和变形产生影响。

楼层上常用载荷		
项次	类别	载荷值 (kN/m^2)
1	住宅，宿舍，幼儿园	1.5
2	教室，阅览室	2.0
3	展览室	3.0
4	商店，候车室	3.5
5	藏书库，档案室	5.0

梁的内力和内力矩：当梁受到载荷，如力和力偶作用时，内部要产生应力，通常既有法应力又有剪切力。为了求得梁的任意截面上这些应力的数值，必须知道作用在该截面内的合力和合力矩。这些合力和合力矩可用静力学平衡方程求得：

如果要求研究距梁的左端 x 处的D截面上的应力，为此将梁在D处切开，并将切面右侧部分除去。此时应将除去部分的影响施加在被保留的D截面的左侧部分上，这个影响包括铅直剪力和力偶，可分别用 V 和 M 表示。

力 V 和力偶 M 使得梁的左侧部分在与力 P_1 、 P_2 、 P_3 共同作用下保持平衡；



抗力矩: 力偶 M 称为截面D处的抗力矩。抗力矩的数值可以用静力学方程式求得;

抗剪力: 铅直力 V 称为截面D处的抗剪力;

弯矩: 截面D一侧的外力关于通过D的轴之矩的代数和称为D处的弯矩; 弯矩和抗力距方向相反, 数值相等, 通常也以 M 表示。在计算中通常使用弯矩, 而不是抗力距;

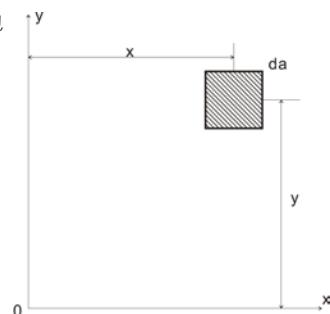
剪力: 截面D一侧, 譬如左侧的全部铅直力的代数和称为该截面处的剪力。剪力和抗剪力方向相反, 数值相等, 通常用 V 表示, 在计算中通常使用剪力而不是抗剪力;

符号规则: 使梁发生弯曲, 并且凹面向上的力产生正弯矩, 使梁发生剪切, 并且左段相对于右段向上的力产生正剪力;

F: 平面面积的形心, 惯性距

单元面积的一次矩: 单元体面积 da 关于平面内任意轴的一次矩, 定义为单元体的面积乘以单元体与该轴垂直距离之积; 例如, 单元体面积 da 关于x轴的一次矩定义为: $dQ_x = yda$, 关于y轴的一次矩为: $dQ_y = xda$;

有限面积的一次矩: 有限面积关于平面内任意轴的一次矩等于有限面积内包含的所有单元体面积对该轴一次矩之和, 求和常用积分来计算;



平面的形心: 平面的形心是这样一个点, 假想的将平面面积集中在此点处, 对于任何轴的一次矩都保持不变。例如一块金属的薄平板, 如果支撑在它的形心之下, 那么它将会在水平平面内保持平衡。

有少数平面的形心位置是很清楚的, 在对称图形如圆形或矩形中, 它的形心和图形的几何中心是重合的;

单元体面积的二次矩或惯性矩: 单元体面积 da 关于平面内任一轴的二次矩或惯性矩, 定义为单元体的面积乘以单元体与该轴垂直距离的平方;

有限面积的二次矩或惯性矩: 有限面积关于平面内任一轴的二次矩或惯性矩等于有限面积内所有包含的全部单元体对该轴惯性矩之和, 经常使用积分来计算;

单位: 惯性矩的单位是长度的四次方, m^4 ;

G: 梁的挠度

梁挠度: 用梁始于未受力位置时的挠度表示梁的变形是最方便的。挠度的测量是从梁开始的中性层到变形后的中性层。变形后中性层描述的构形称为梁的弹性曲线。

位移 y 被定义为梁的挠度，经常需要确定沿梁长每个 x 值对应的挠度 y 。这个关系可以写为等式，此等式通常叫做梁的挠曲线或弹性曲线方程；



梁挠度的重要性: 与限制应力一样，梁的设计规范经常要限制梁的变形。例如，在很多建筑规范中，梁的最大许可挠度不能超过梁长的1/3000；

H: 梁的塑性设计

塑性设计: 结构设计的某些情况可以允许构件产生适量的永久形变。在这种情况下，许可载荷有可能大于弹性理论给出的许可载荷，因为弹性理论不允许结构中任何点处的应力达到材料的屈服点。这种结果有利于更有效地使用材料，被称之为塑性设计。这种更有效地设计是可行的，其原因在于某些材料，如结构钢，在达到屈服点之后，仍能承担相对较大的塑性形变；

塑性铰: 伴随作用在梁上的横向载荷不断增加，在梁上某个特殊位置的最外层纤维开始屈服，并且屈服相当快的向内部扩展。最终，中性轴一侧拉伸纤维的应力全部达到拉伸屈服应力，另一侧压缩纤维的应力全部达到压缩屈服应力时，该位置将产生流动或铰链作用，称为塑性铰，通过塑性铰传递的弯矩保持不变；

结构力学基础

A: 杆件结构力学

土木工程中各类建筑物和构筑物，例如房屋、桥梁、水池、挡土墙等，在使用过程中，都要承受各种载荷的作用。这种承受载荷的建筑物和构筑物或其中的某些构件，都可以称为结构。

结构的类型多种多样，就几何特征区分，有杆件结构、薄壁结构、实体结构。杆件的基本特征是它的长度远大于其它两个方向的尺度—截面的宽高，杆件结构便是由若干这种杆件组成的。薄壁结构是厚度远小于其他两个尺度的结构。平面板状的薄壁结构称为薄板，由若干块薄板组成其它各种薄壁结构。具有曲面外形的薄壁结构，称为薄壳结构。实体结构是指三个方向的尺度大约为同一量级的结构，例如挡土墙，堤坝，块式基础等。

依照空间特征区分，杆件结构可以分为平面杆件结构和空间杆件结构两类。凡组成结构的所有杆件的轴线都位于同一平面，并且载荷也作用于该平面内的结构，称为平面杆件结构。否则，便是空间结构。严格来说，实际的结构都是空间结构，不过在进行计算时，常可以根据其实际受力情况的特点，将它分解为若干个平面结构进行分析，以使计算简化。但是需要注意，并非所有情况都能这样处理，有些时必须作为空间结构来研究的。

杆件结构力学的任务是研究结构的组成规律和合理形式，以及结构在外因作用下的强度，刚度和稳定性的计算原理和方法。研究组成规律的目的在于保证结构各部分不至于发生相对运动，使它能承受载荷并维持平衡。进行强度和稳定性计算的目的在于保证结构的安全并使之符合经济的要求。计算刚度的目的在于保证结构不至于发生过大的，在实用上不能容许的位移。研究结构的合理形式是为了有效地利用材料，使其受力性能得到充分发挥。上述强度、刚度和稳定性的计算，不仅在设计结构时需要进行，当已有结构所承受的载荷情况改变时，也应加以核算，判明是否需要采取加固措施。

B: 杆件结构计算简图

对结构进行力学分析之前，需要将实际结构进行简化，分清结构受力，变形的主次，用一个简化的理想模型来代替实际结构，称为结构的计算简图。

结构的计算简图一般包括杆件的简化，支座的简化和结点的简化等。

1-杆件的简化

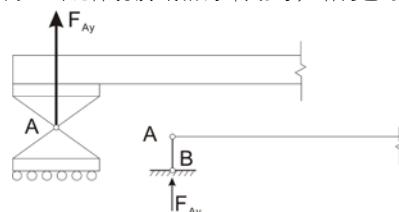
根据杆件受力后变形特点，各种杆件在计算简图中均用其轴线来代替。等截面直杆的轴线是一直线，曲杆是一曲线。变截面杆件也都近似以直线或曲线来代替。

2-支座的简化和分类

将结构与基础或其它支撑物联系，并用以固定结构位置的装置称为支座。在建筑结构中，从支座对结构的约束作用来看，常用的计算简图分为三类：

1.活动铰支座

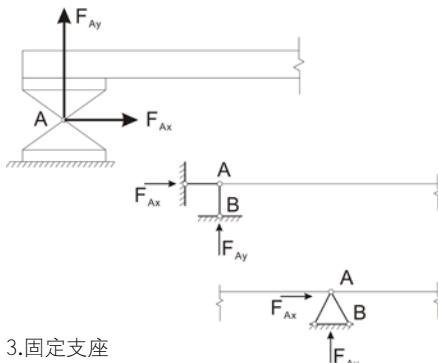
活动铰支座对结构的约束作用是只阻止结构上的A点沿垂直于支承平面方向移动，结构既可以绕铰A转动，又可沿着与支撑平面平行的方向移动。因此，当不考虑支撑平面上的摩擦力时，活动铰支座的反力将通过铰A的中心并与支撑平面垂直，其作用点和方向是确定的，只是大小未知，用 F_{Ay} 表示。根据上述特点，这种支座在计算简图中可用一根链杆来表示，因为与该链杆相连的结构不仅可绕铰A转动，而且当链杆绕铰B做微小转动时，结构也可以在垂直于链杆的方向作微小移动。链杆AB的内力既代表支座的反力。



2.固定铰支座

固定铰支座的约束作用是不允许结构上A点发生任何移动，但是可以绕铰A转动。因此，固定铰支座的反力将通过铰A的中心，但其方向和大小都是未知的，可以用两个沿确定方向的未知反力 F_{Ax} 和 F_{Ay} 来表示。这种支座在计算简图中常用交与点A的两根链杆来表示。

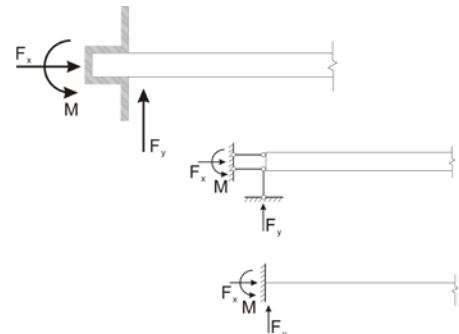
在实际结构中，凡属不能移动而可作微小转动的支承情况，都可以视为固定铰支座。例如插入钢筋混凝土杯形基础中的柱子，当用沥青麻丝填缝时，则柱的下端便可视为固定铰支座。



3.固定支座

固定支座不允许结构发生任何移动和转动，它反力大小、方向和作用点都是未知的。因此，可以用水平和竖向的反力 F_x 和 F_y 及反力偶 M 来表示。固定支座也可用三根既不全平行又不全交于一点的链杆表示，这三根链杆的内力是与这种支座的三个反力等效的，因为若将两根水平链杆的内力均向杆件截面的中心平移后，便可合成一个沿杆轴作用的水平反力 F_x 和一个反力偶 M 。

在实际结构中，凡嵌入墙身的杆件，其嵌入部分有足够的长度，以致使杆端不能有任何移动和转动时，该端就可以视为固定支座。又如插入杯形基础中的柱子，如果用细石混凝土填缝，则柱的下端一般可以看作固定支座。



3-结点的简化

在杆件结构中，几根杆件相互联结处称为结点。根据结构受力特点和结点的构造情况，在计算中常将其简化为两种类型：

1. 铰结点

铰节点的特点是它所联结的各杆件都可以绕结点自由转动。

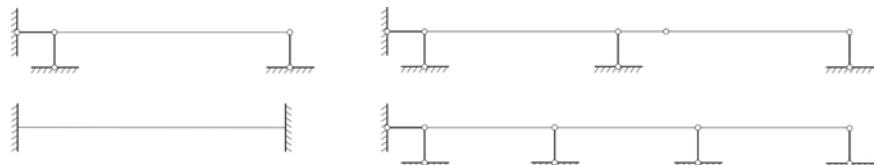
2. 刚结点

刚结点是它所联结的各杆件变形前后在结点处杆端的夹角保持不变，即各杆端转动的角度相等。

C: 杆件结构的类型

1-梁

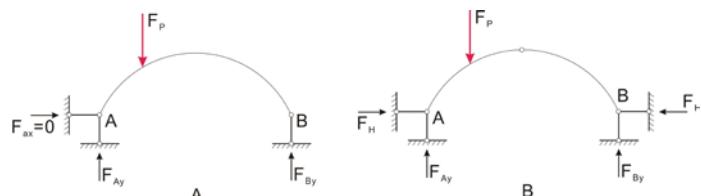
梁是一种受弯杆件，可以是单跨也可以是多跨。



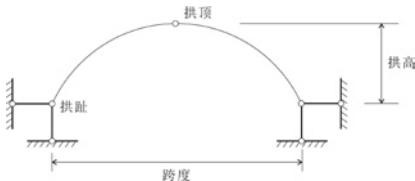
2-拱

拱是轴线为曲线，且在竖向载荷作用下支座会产生水平反力的杆件结构。这种水平反力将使拱的弯矩远小于跨度，载荷及支承情况相同的梁的弯矩，又称水平推力，或简称推力。拱结构与梁结构的区别，不仅在于外形不同，更重要的是在竖向载荷作用下是否产生水平推力。

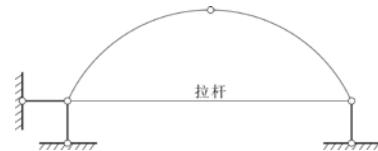
图A、B所示的两个结构，虽然它们的杆轴都是曲线，但是A所示结构在竖向载荷作用下不产生水平推力，其弯矩与同跨度，同载荷的相应简支梁的弯矩相同，所以这种结构不是拱结构而是曲梁。但是B图，结构两端都有水平支座链杆，在竖向载荷作用下将产生水平推力，所以属于拱结构。由于水平推力的存在，拱中各截面的弯矩将比相应的曲梁或简支梁的弯矩更小，并且会使整个拱体主要承受压力。因此，拱结构可采用强度较高而抗拉强度较低的砖、石、混凝土等材料来建造。



拱结构最高的一点称为拱顶，三铰拱的中间通常安置在拱顶处。拱的两端与支座联结处称为拱趾，或者称为拱脚。两拱趾在同一水平线上称为平拱，否则为斜拱。两个拱趾间的水平距离L称为跨度。拱顶到两拱趾连线的竖向距离f称为拱高，或者称为拱矢。拱高于跨度之比称为高跨比或矢跨比。（拱的主要力学性能与高跨比有关）

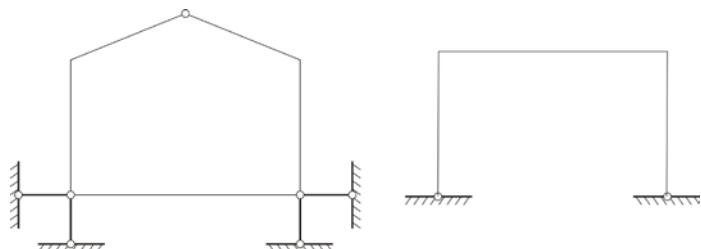


用作屋面承重结构的三铰拱，常用两支座之间设置的水平拉杆代替一根水平支座链杆，以拉杆产生的拉力代替推力的作用，使竖向荷载作用下支座只产生竖向反力，故称为具有拉杆的拱，或者称为拉杆拱。但是，这种结构的内部受力情况与三铰拱完全不同，其优点在于消除了推力对支承结构（例如墙）的影响。



3-刚架

刚架是由梁和柱组成的结构，各杆件主要受弯。刚架的结点主要是刚结点，也可以有部分铰结点或组合结点。刚架的内力，是指构成刚架各杆件垂直于杆轴截面上的弯矩 M ，剪力 F_q 和轴力 F_N 。在计算静定刚架时，通常先由整体或某些部分的平衡条件，求出各支座反力和各铰结处的约束力，然后在逐杆绘制内力图。



4-桁架

桁架是由若干杆件在两端用铰联结而成的结构。桁架各杆的轴线都是直线，当仅受作用于节点的载荷时，各杆只产生轴力。

桁架结构在土木工程中应用很广泛。特别是在大跨度结构中，桁架更是一种重要的结构形式。桁架的形式，桁架杆件之间的联结方式及采用的材料是多种多样的。在分析桁架时，必须选取既能反映其本质又便于计算的简图。各种桁架的共同特征：由于在结点载荷作用下，桁架中各杆的内力主要是轴力，而弯矩和剪力很小，可以忽略不计，因而联结各杆件的结点所起作用和铰结点是接近的。一般桁架简图引用下列假定：

- 1-各杆在两端用绝对光滑而无摩擦的理想铰相互联结；
- 2-各杆的轴线都绝对平直且处于同一平面内，并通过铰的中心；
- 3-载荷和支座反力作用在结点上，并且都位于桁架所在的平面内；

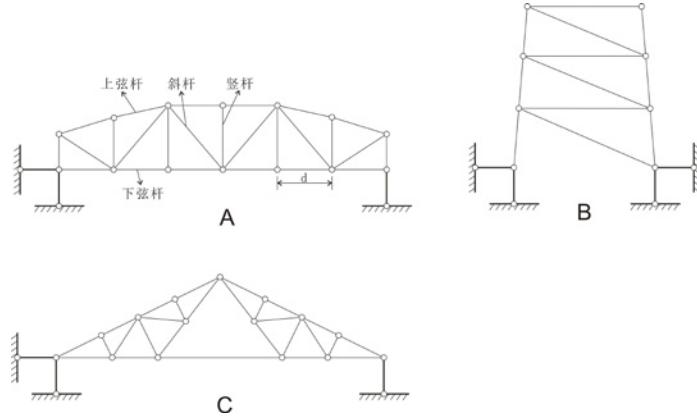
在上述理想情况下，桁架各杆均为两端铰接的直杆，仅在两端受约束力作用，故只产生轴力。这类杆件也称为二力杆，在轴向受拉或受压时，由于截面上的应力均匀分布且同时达到极限，故材料能得到充分利用。

实际的桁架常不能完全符合上述理想情况。例如桁架的结点具有一定的刚性，有些杆件在结点处可能连续不断，或者各杆之间的夹角几乎不可能变动。另外，各杆轴无法绝对平直，结点上各杆的轴线也不一定会交于一点，或者载荷并不都能作用于结点上等。因此，桁架中某些杆件必将发生弯曲而产生不均匀分布的力。通常把按理想情况下计算出来的内力称为主内力，由于理想情况不能完全实现而产生的附加内力称为次内力。

常用桁架一般按下列两种方式组成：

- 1—由基础或由一个基本铰结三角形开始，一次增加二元体组成桁架，图A、B所示。这样的桁架称为简单桁架；
- 2—由几个简单桁架按照几何不变体系的简单组成规则联成一个桁架图C。这样的桁架称为联合桁架，其中把简单桁架联结在一起的杆件，称为联合杆。

桁架的杆件，依其所在位置的不同，可分为弦杆和腹杆两类。弦杆是指桁架上、下外围的杆件，上边的杆件称为上弦杆，下边的杆件称为下弦杆。桁架上弦杆和下弦杆之间的杆件成为腹杆。腹杆又分为竖杆和斜杆。弦杆上相邻两点之间的区间称为节间，其距离d称为节间度。



基于Karamba的结构力学分析与优化

从材料力学与结构力学的基础出发，进一步探索基于计算机辅助下的结构协同设计。

不可否认的是，只有在计算机辅助之下，设计师才更容易的进入结构师的领域，从繁琐的计算公式解脱出来，直接面向设计与结构的协同，而将计算交与计算机来处理。同时，进一步拓展设计的方法，将设计的合理性与结构同步优化比较协同起来，在参数与结构分析的结合上，可以在无数可能性中找到最佳的结构选型，例如由一定数量的随机点随机连线构成的框架结构，在一定载荷影响下，哪种联结方式使得框架位移最小。

基于GH参数平台的结构分析，使分析研究的可能性得以进一步深入，Karamba官方网站中提供了多样的案例，从不同方面解释Karamba在协助设计方面的表现，可以作为设计师在研究设计时的程序样本。

未设置Karamba材料时，使用其默认的材料，为钢Steel材料，等级为S235，即屈服应力Yields为 $f_y = 23.5[\text{kN}/\text{m}^2]$ 。

E: Young's Modulus, 杨氏模数 kN/cm^2 ;

G: Shear Modulus, 剪切模量 kN/cm^2 ;

gamma: Specific Weight, 比重 kN/m^3 ;

fy: Yield Stress, 屈服强度 kN/cm^2 ;





关于比重与密度：

密度Density：某种物质的质量和其体积的比值，即单位体积的某种物质的质量，叫做该物质的“密度”。用 ρ 表示，国际单位制的单位为千克每立方米 (kg/m^3)；

比重Specific Weight：物体的重力和它的体积的比值，即单位体积的某种物质的重力，称作该物体的“比重力”。用 γ 表示，国际单位制的单位为牛顿每立方米 (N/m^3)；

除气体外，任何物质某体积的重力，和4℃时同体积的水的重力相比，即称为该物质的比重。水的密度在4℃时等于1 g/cm^3 。物质的重力和其质量成正比，则其单位体积的重力，必然和其密度成正比，所以任何物质的比重，又等于该物质和水在4℃的密度相比。

故物质的比重常和其密度的数值相等。唯密度须以1 g/cm^3 为单位，比重则仅为纯数字，所以意义上绝不相同。根据1978年国际纯粹应用物理学协会所属符号单位和术语委员会的文件建议，我国已取消比重的概念，而以密度的概念代替。

关于MPa：

1 MPa (兆帕) = 1000 Kpa (千帕) = 1000000 pa (帕)； 1 GPa = 1000 MPa；

1 Pa = 1 N/m²，1 Pa是1 N的力均匀的压在1 m²面积上所产生的压强；

则1 MPa = 1000 kN/m² = 0.1 kN/cm²； 1 GPa = 100 kN/cm²

$cc=m=cm^3; g/cc=g/cm^3=10kN/m^3$; (1立方厘米(毫升)物质的重力)

抵抗材料变形(位移)的Material Stiffness材料刚度属性使用杨氏模量

Young's Modulus反映，值越高材料刚性越好，所使用的单位为 kilo Newton per square centimeter kN/cm²；

一般材料的杨氏模量：

Type of Material 材料类型	E [kN/cm ²] 杨氏模数(弹性模量)
Steel 钢	21000
Aluminum 铝	7000
Reinforced Concrete 钢筋混凝土	3000
Glass Fiber 玻璃纤维	7000
Wood (Spruce) 云杉	1000

在拉伸材料时，材料不仅在长度上发生变化，同时截面面积缩小，然而对于较大的梁跨度与横截面的比，该种影响较小。Shear Modulus剪切模量可以用于描述该类问题。然而目前版本，Karamba并不考虑该材料属性。

对于Specific Weight，单位为kilo Newton per cubic meter [kN/m³]。通常重力加速度Gravitational Acceleration的值为 $g = 9.81[\text{kg}\cdot\text{m}/\text{s}^2]$ ，对于结构变形的计算，其值可以约等于 $10[\text{kg}\cdot\text{m}/\text{s}^2]$ 。在Karamba计算中，仅当施加重力载荷时，材料的比重属性才起作用。

关于材料的属性可以查询 <http://www.matweb.com/材料物性数据库> 获得材料信息，或者查询国内相关数据库。

从MatWeb上获得的材料属性数据：

Overview of materials for AISI 4000 Series Steel

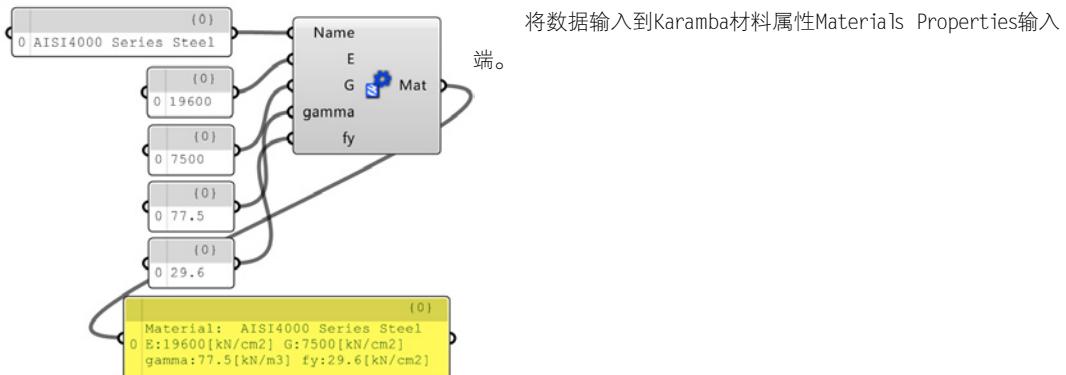
Categories: [Metal](#); [Ferrous Metal](#); [Alloy Steel](#); [AISI 4000 Series Steel](#)

Material Notes: This property data is a summary of similar materials in the MatWeb database for the category "AISI 4000 Series Steel". Each property range of values reported is minimum and maximum values of appropriate MatWeb entries. The comments report the average value, and number of data points used to calculate the average. The values are not necessarily typical of any specific grade, especially less common values and those that can be most affected by additives or processing methods.

Vendors: No vendors are listed for this material. Please [click here](#) if you are a supplier and would like information on how to add your listing to this material.

Physical Properties	Metric	English	Comments
Density	7.75 - 7.85 g/cc	0.280 - 0.284 lb/in ³	Average value: 7.85 g/cc Grade Count:591
Particle Size	6.70 - 12.0 µm	6.70 - 12.0 µm	Average value: 9.27 µm Grade Count:12
Mechanical Properties	Metric	English	Comments
Hardness, Brinell	121 - 578	121 - 578	Average value: 280 Grade Count:553
Hardness, Knoop	140 - 616	140 - 616	Average value: 309 Grade Count:578
Hardness, Rockwell B	68.0 - 112	68.0 - 112	Average value: 96.3 Grade Count:533
Hardness, Rockwell C	10.0 - 62.5	10.0 - 62.5	Average value: 30.2 Grade Count:525
Hardness, Vickers	36.0 - 614	36.0 - 614	Average value: 299 Grade Count:578
Tensile Strength, Ultimate	450 - 1970 MPa	65300 - 285000 psi	Average value: 977 MPa Grade Count:534
Tensile Strength, Yield	296 - 1860 MPa	42900 - 270000 psi	Average value: 749 MPa Grade Count:532
Elongation at Break	8.00 - 34.0 %	8.00 - 34.0 %	Average value: 19.2 % Grade Count:532
Reduction of Area	16.0 - 74.4 %	16.0 - 74.4 %	Average value: 54.9 % Grade Count:445
Modulus of Elasticity	196 - 213 GPa	28400 - 30900 ksi	Average value: 205 GPa Grade Count:589
Compressive Yield Strength	1650 - 1800 MPa	239000 - 260000 psi	Average value: 1700 MPa Grade Count:12
Bulk Modulus	140 - 170 GPa	20300 - 24700 ksi	Average value: 140 GPa Grade Count:589
Poissons Ratio	0.270 - 0.300	0.270 - 0.300	Average value: 0.290 Grade Count:589
Charpy Impact	15.0 - 23.0 J	11.1 - 17.0 ft-lb	Average value: 19.8 J Grade Count:5
Izod Impact	9.00 - 138 J	6.64 - 102 ft-lb	Average value: 63.6 J Grade Count:68
Fatigue Strength	138 - 772 MPa	20000 - 112000 psi	Average value: 528 MPa Grade Count:11
Machinability	50.0 - 75.0 %	50.0 - 75.0 %	Average value: 59.7 % Grade Count:560
Shear Modulus	75.0 - 82.0 GPa	10900 - 11900 ksi	Average value: 80.0 GPa Grade Count:589
Electrical Properties	Metric	English	Comments
Electrical Resistivity	0.0000200 - 0.0000263 ohm-cm	0.0000200 - 0.0000263 ohm-cm	Average value: 0.0000241 ohm-cm Grade Count:585
Thermal Properties	Metric	English	Comments
CTE, linear	10.4 - 14.6 µm/m-°C	5.78 - 8.11 µin/in-°F	Average value: 12.7 µm/m-°C Grade Count:220
Specific Heat Capacity	0.473 - 0.477 J/g-°C	0.113 - 0.114 BTU/lb-°F	Average value: 0.475 J/g-°C Grade Count:518
Thermal Conductivity	42.6 - 48.1 W/m-K	296 - 334 BTU-in/hr-ft ² -°F	Average value: 44.1 W/m-K Grade Count:585
Component Elements Properties	Metric	English	Comments
Carbon, C	0.0900 - 1.29 %	0.0900 - 1.29 %	Average value: 0.298 % Grade Count:604
Chromium, Cr	0.300 - 3.30 %	0.300 - 3.30 %	Average value: 0.795 % Grade Count:373
Iron, Fe	91.9 - 99.1 %	91.9 - 99.1 %	Average value: 96.9 % Grade Count:603
Manganese, Mn	0.300 - 3.00 %	0.300 - 3.00 %	Average value: 0.703 % Grade Count:604
Molybdenum, Mo	0.0800 - 0.750 %	0.0800 - 0.750 %	Average value: 0.256 % Grade Count:604
Nickel, Ni	0.650 - 3.80 %	0.650 - 3.80 %	Average value: 2.17 % Grade Count:312
Phosphorous, P	0.0200 - 0.0400 %	0.0200 - 0.0400 %	Average value: 0.0341 % Grade Count:593
Silicon, Si	0.150 - 1.05 %	0.150 - 1.05 %	Average value: 0.240 % Grade Count:604
Sulfur, S	0.0200 - 0.0500 %	0.0200 - 0.0500 %	Average value: 0.0380 % Grade Count:593

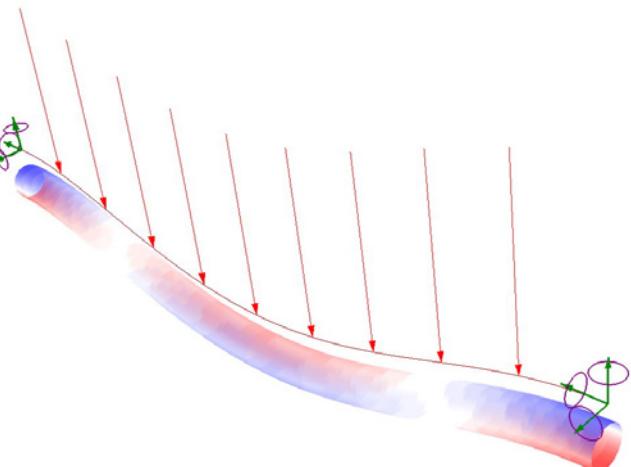
Some of the values displayed above may have been converted from their original units and/or rounded in order to display the information in a consistent format. Users requiring more precise data for scientific or engineering calculations can click on the property value to see the original value as well as raw conversions to equivalent units. We advise that you only use the original value or one of its raw conversions in your calculations to minimize rounding error. We also ask that you refer to MatWeb's [terms of use](#) regarding this information. [Click here](#) to view all the property values for this datasheet as they were originally entered into MatWeb.



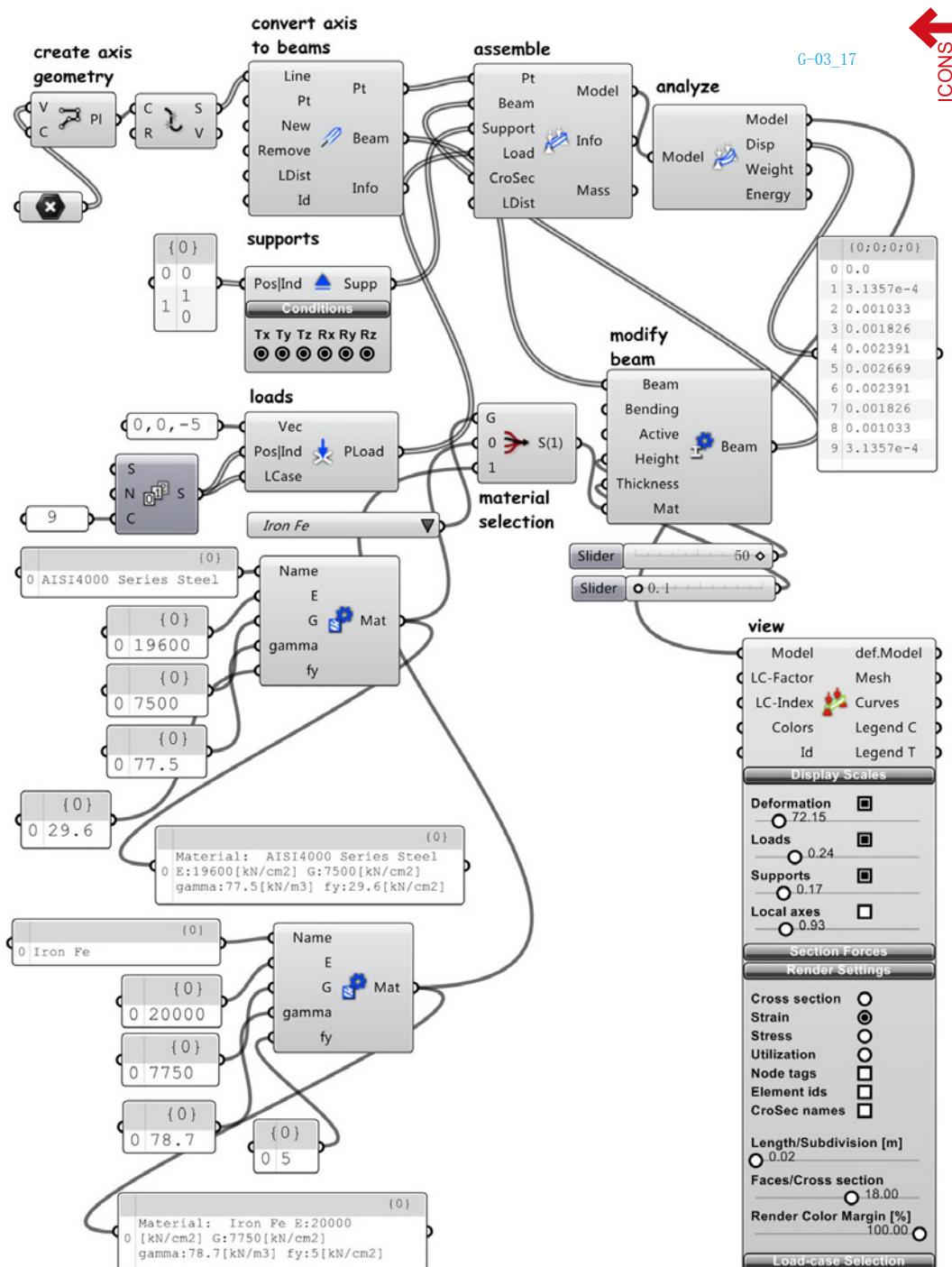
A: 同一载荷Load与约束(支撑) Support作用下不同材料的位移(变形) 比较

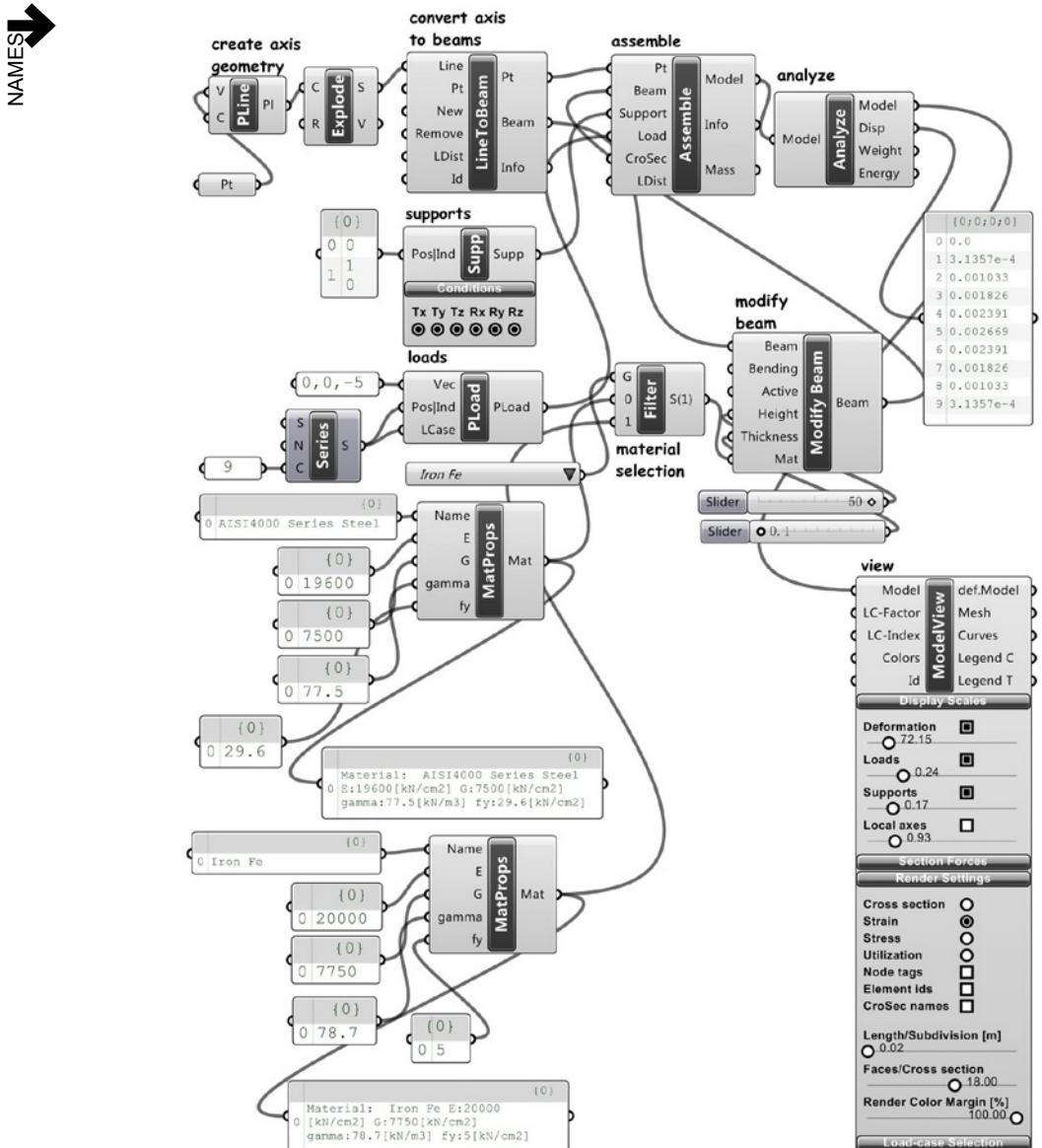
在载荷与支撑条件不变的条件下可以输入不同的材料属性，比较位移变化情况。该例中梁两端为固定铰支座，载荷分布于9个均匀分布的点上，比较在MatWeb材料物性数据库中获的钢Steel (AISI4000 Series Steel)与铁Iron (Fe)的结构变化情况。

图中材料弯曲的情况并不代表实际材料的位移，是为了便于观察，对位移调整了比例进行显示，该部分可以在Display Scale卷展栏中设置。



AISI4000 Series Steel各点位移(m) :	Iron Fe各点位移(m):
0. 0.0	0. 0.0
1. 3.1996e-4	1. 3.1357e-4
2. 0.001054	2. 0.001033
3. 0.001863	3. 0.001826
4. 0.00244	4. 0.002391
5. 0.002723	5. 0.002669
6. 0.00244	6. 0.002391
7. 0.001863	7. 0.001826
8. 0.001054	8. 0.001033
9. 3.1996e-4	9. 3.1357e-4





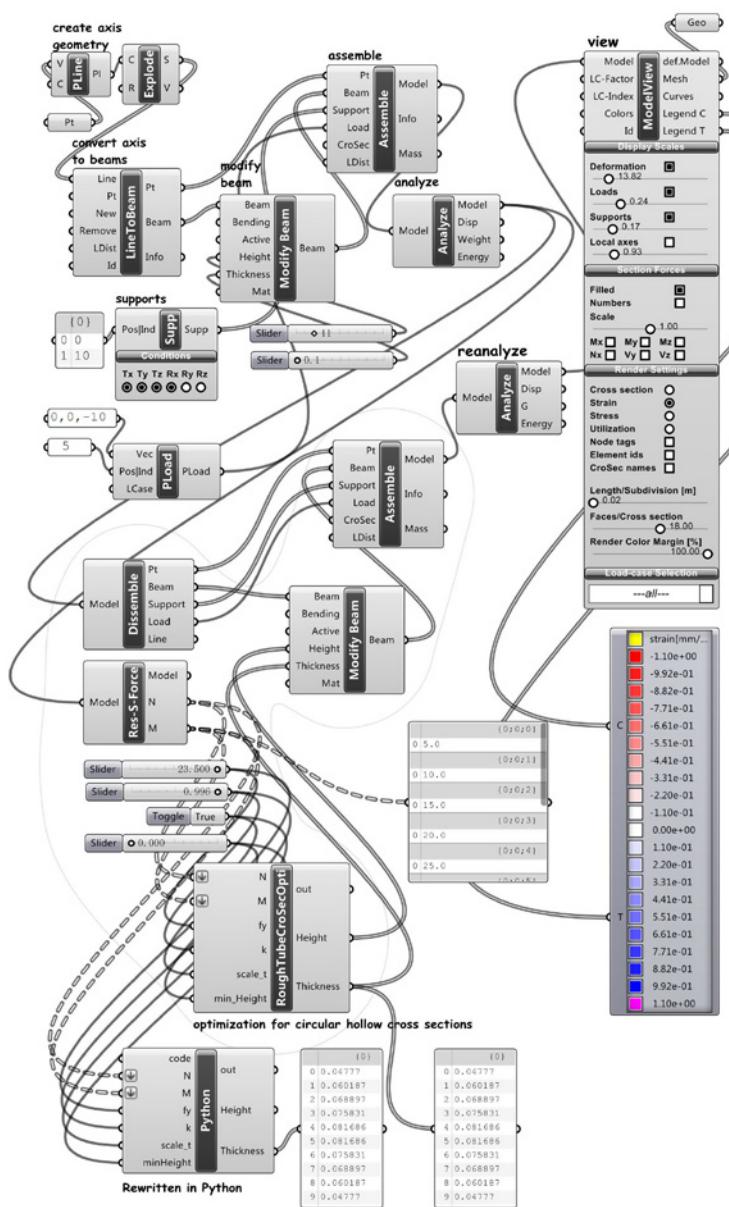


NAMES

B: 梁截面尺寸的优化

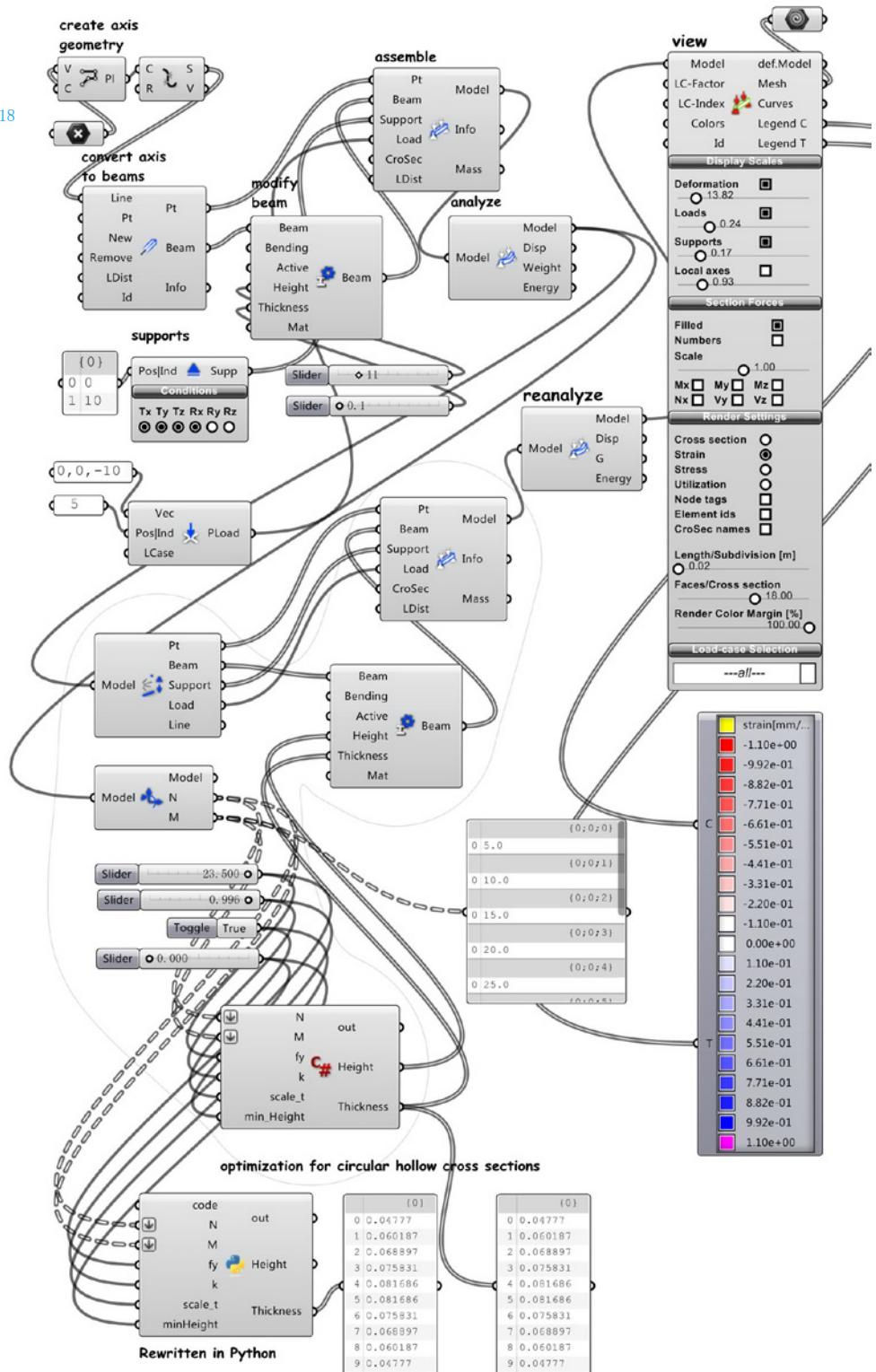
由结构优化获得的变截面

在结构计算中可以使用程序语言C#, Python等编写相应的程序对结构截面进行优化。



ICONS

G-03_18



```

private void RunScript(double N, double M, double fy, double k, bool scale_t, double min_Height, ref object
Height, ref object Thickness)
{
    //---
    // written by Clemens Preisinger
    // 11/14/2011
    //---

    // how it works:
    // first outer diameter and wall thickness of the tube is determined
    // based on the moment M. If the outer diameter is smaller than
    // min_Height then a diameter of min_Height is assumed.
    //

    // In order to accomodate the additional strain
    // due to the normal force N are is then added to the cross section:
    // either by increasing the wallthickness (if scal_t is true) or by
    // increasing the tubes diameter.
    //

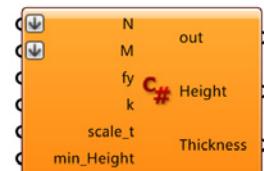
    // input paramaters:
    // N      normal force [kN]
    // M      resultant bending moment [kNm]
    // fy     yield stress [kN/cm2] this describes the strength of a material.
    //        The default is 23.5[kN / cm2] which corresponds to steel.
    //        Sometimes it is not the stranght of a material that
    //        determines the size of a cross section but the maximum
    //        deflection. In such a case scale down this value to limit
    //        deflections.
    // k      approximate quotient between inner diameter and outer
    //        diameter of the tube.
    // scalt_t   if true then the wallthickness is scaled otherwise the
    // diameter
    //        for accomodating stresses due to N.
    // min_Height minimum diameter from bending stresses
    //---

    // calculate outer diameter from moment
    double Dm = Math.Pow(100 * Math.Abs(M) / (Math.PI / 32 * fy * (1 - Math.Pow(k, 4))), 1.0 / 3.0);

    Dm = Math.Max(Dm, min_Height);

    double Dn;
    double t;
}

```



```

if (scale_t || Dm == 0) {
    // add area needed for N by scaling the wall thickness
    //---
    // the result will lie a on the safe side because the additional
    // moment of inertia of the tube is not taken into account
    Dn = Math.Sqrt(Math.Abs(N) / (Math.PI / 4 * fy) + Dm * Dm);
    t = (Dn - k * Dm) / 2;
} else {
    // add area needed for N by scaling the diameter
    //---
    // the result will lie a on the safe side because the additional
    // moment of inertia of the tube is not taken into account
    t = Dm * (1 - k) / 2;
    Dn = Math.Abs(N) / (fy * Math.PI * t) + Dm;
}

Height = Dn;
Thickness = t;

```

使用Python 改写原C#编写的截面计算方法：

```

N=N
M=M
fy=fy
k=k
scale_t=scale_t
minHeight=minHeight
Dm = math.pow(100 *abs(M) / (math.pi / 32 * fy * (1 - math.pow(k, 4))), 1.0
/ 3.0)
Dm = max(Dm, minHeight)
if scale_t==0 or Dm==0:
    Dn = math.Sqrt(abs(N) / (math.pi / 4 * fy) + Dm * Dm)
    t = (Dn - k * Dm) / 2
else:
    t = Dm * (1 - k) / 2
    Dn = abs(N) / (fy * math.pi * t) + Dm
Height = Dn
Thickness = t

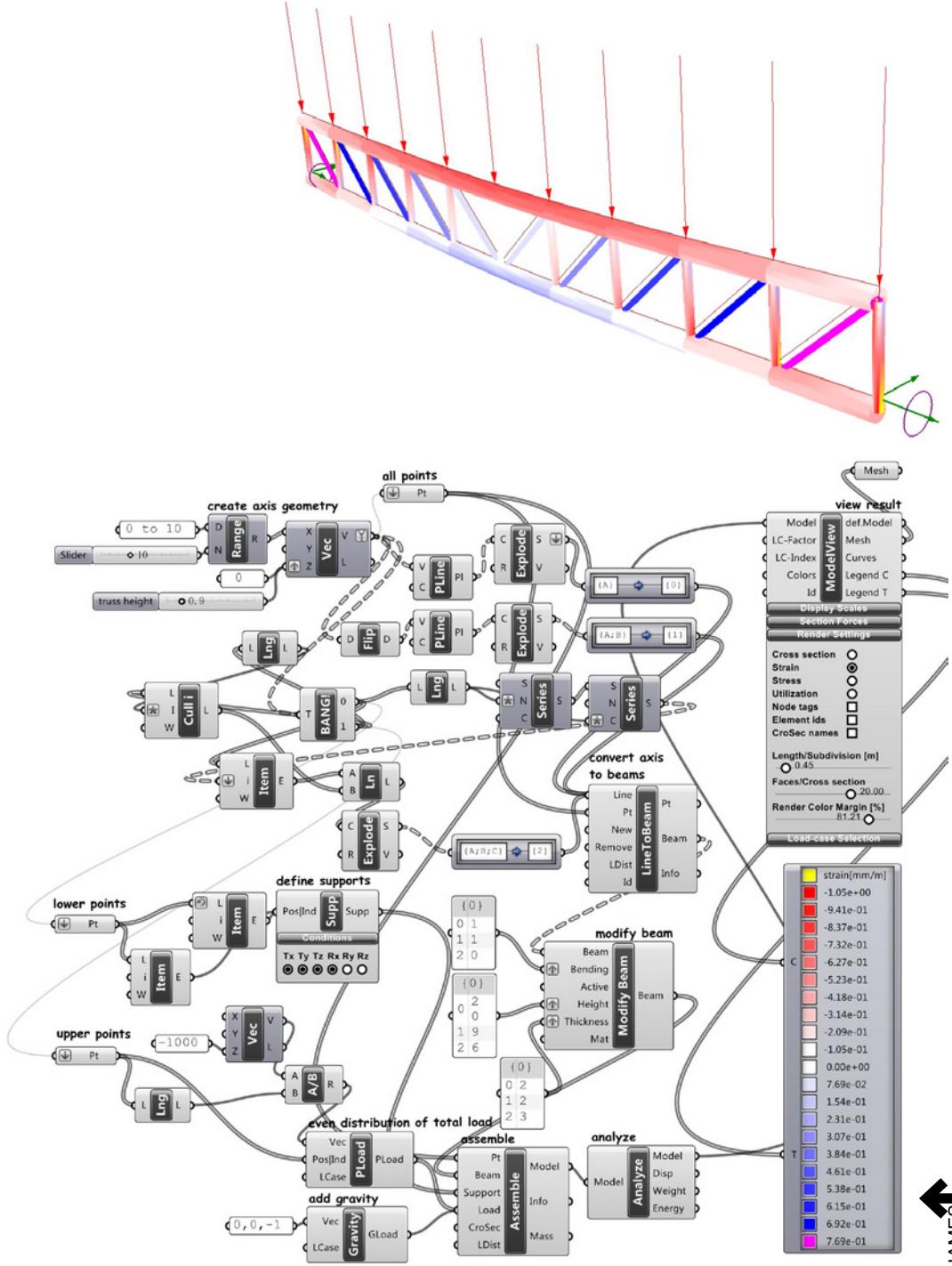
```



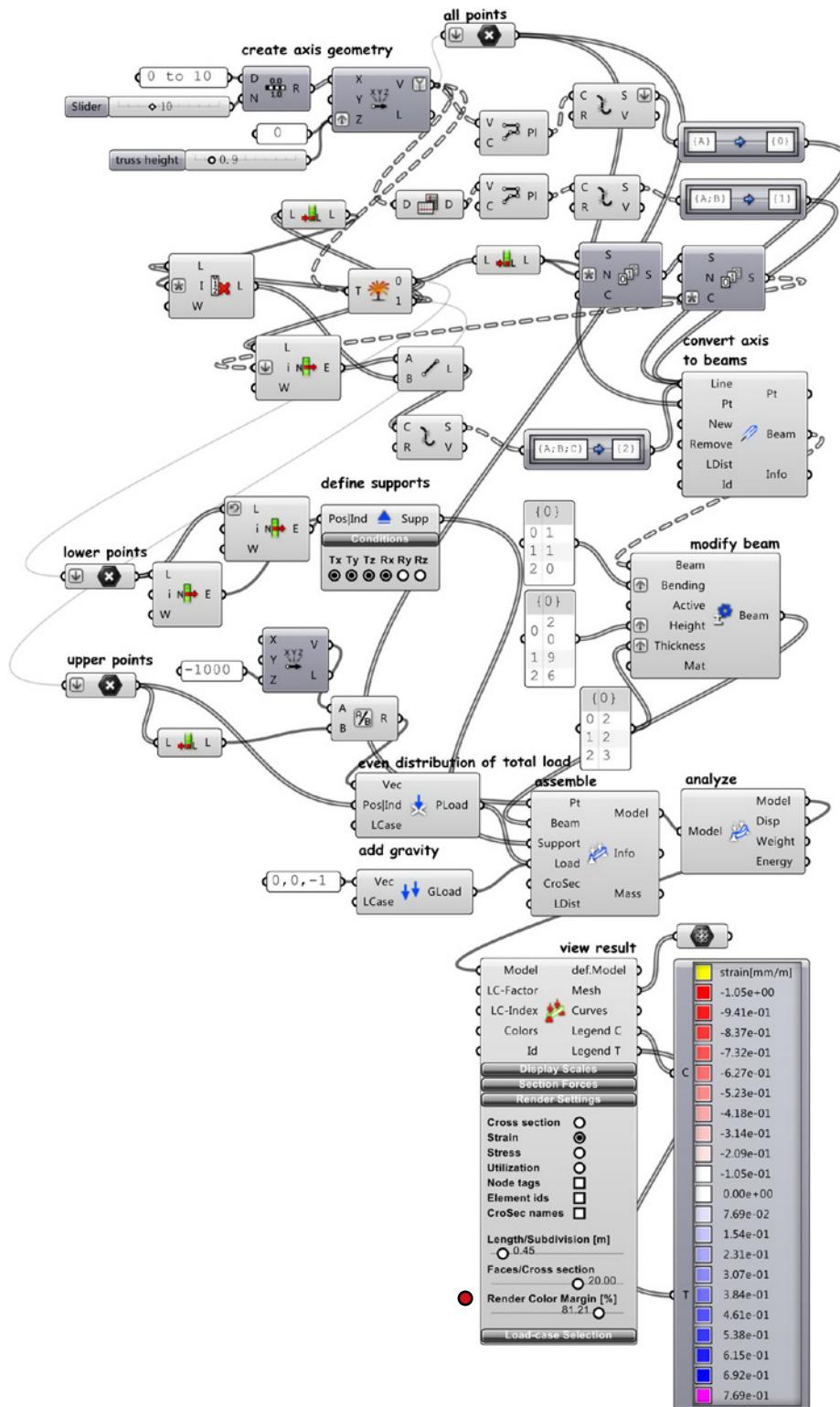
在改写过程中完全使用C#的计算方法，只是将C#与Python不同的语法表示方式进行转换，例如C#中的Math.Abs(M)在Python中为abs(M)，Math.Pow()为math.pow()，Math.Max(Dm, minHeight)为max(Dm, minHeight)，Math.PI为math.pi，scale_t || Dm == 0为pi==0 or Dm==0等。

C: 使用“Render Color Margin”滑条获取过载的区域

通过结构计算可以获得桁架在一定载荷作用下的内力变化情况，并通过ModelView组件下的Render Color Margin滑条获取感兴趣的受力区域进行进一步研究。同时使用Karamba可以设置桁架不同部分杆件的截面尺寸。

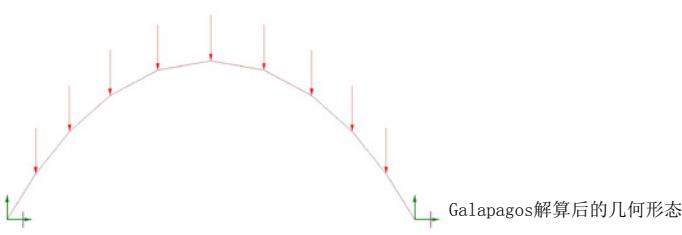
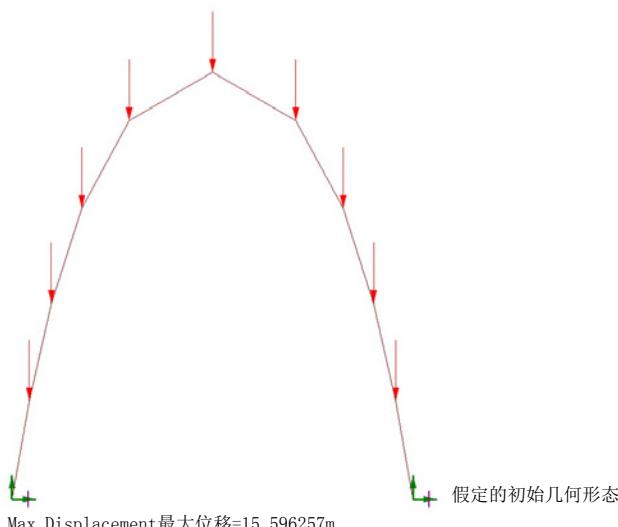


G-03_19

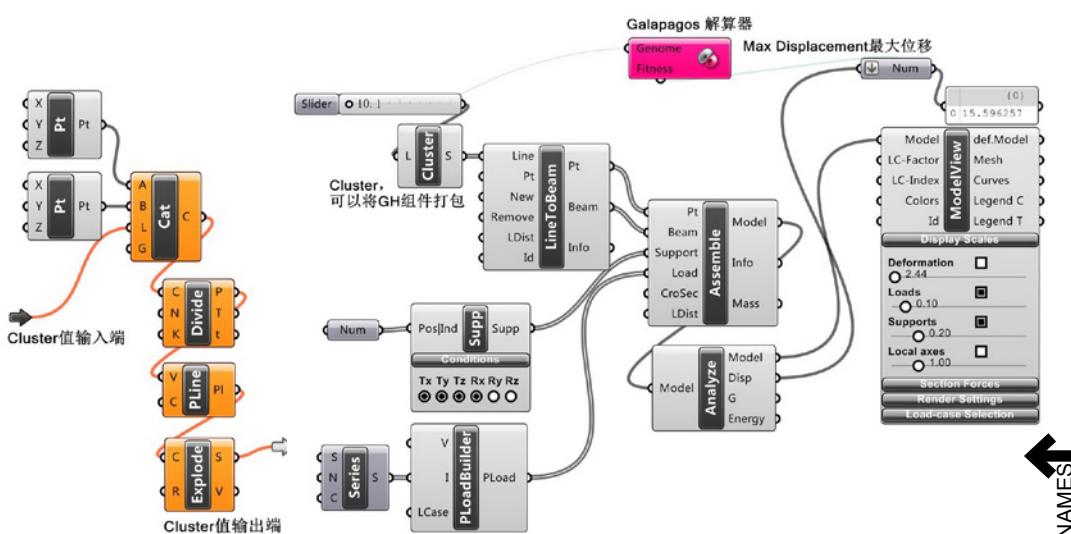


ICONS →

D-1: 几何结构优化 (Galapagos +Karamba)

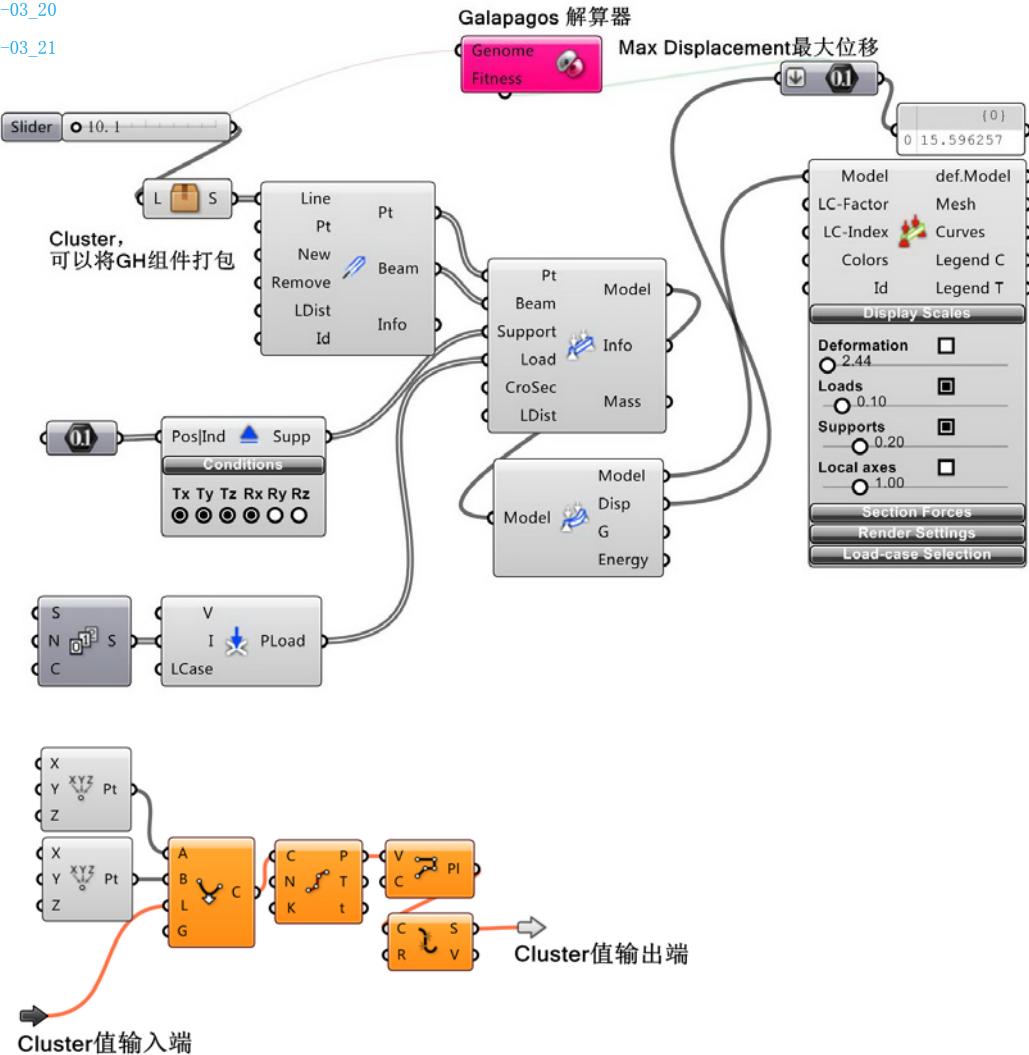


在载荷一定的条件下，拱顶点的位置决定拱的几何形态，将其作为Galapagos的基因输入端，而结构优化的目的在于将Max Displacement最大位移值最小化，因此将其连入Fitness适合输入端。遗传解算后，可以获得结构优化的几何，其位移由原15.60优化为0.96。



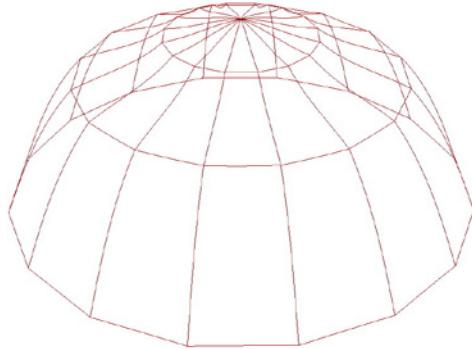
G-03_20

G-03_21

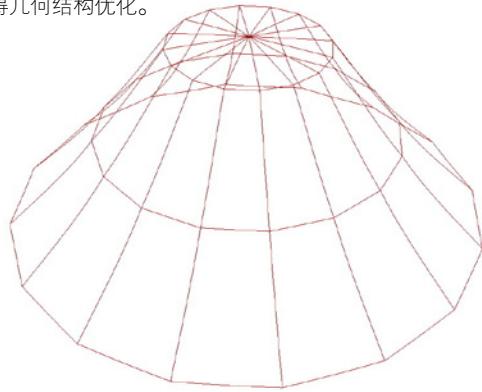


D-2: 几何结构优化 (Galapagos +Karamba)

本例三个水平面类圆的多边形直径是确定的，将各水平面所处的高度作为Galapagos的基因输入端的条件，通过解算不同水平面高度变化使获得最大位移值最小，从而获得几何结构优化。

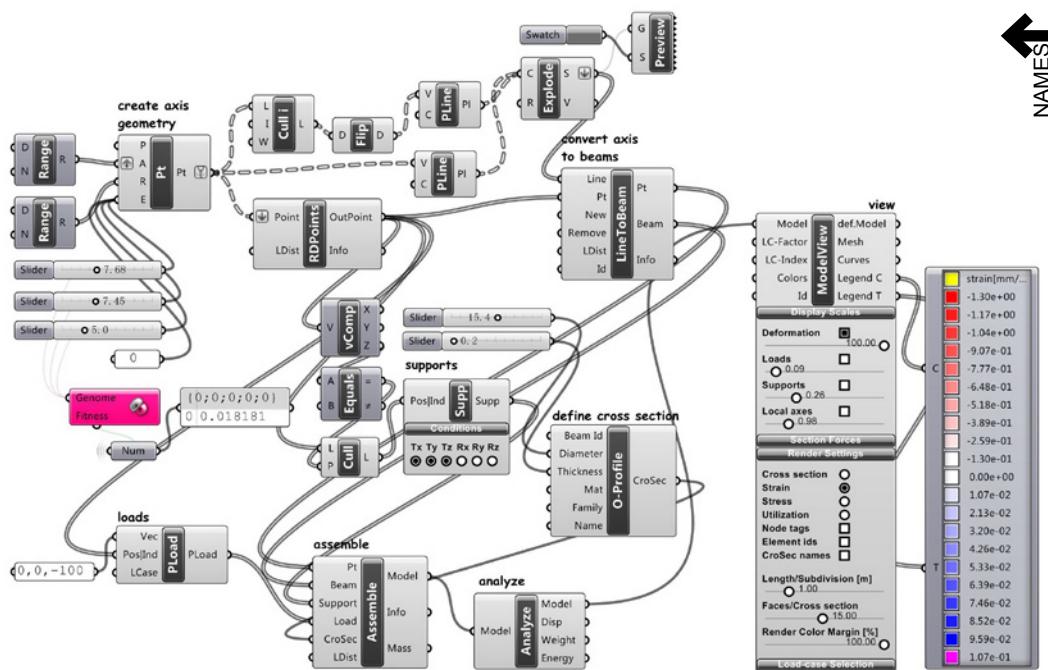
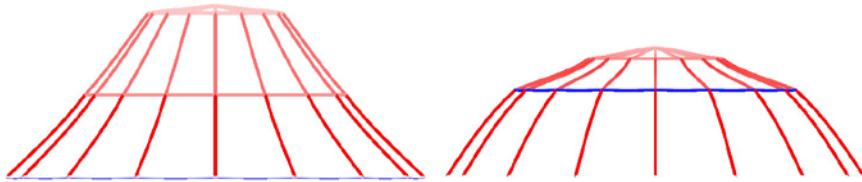


Max Displacement 最大位移=0.018747m

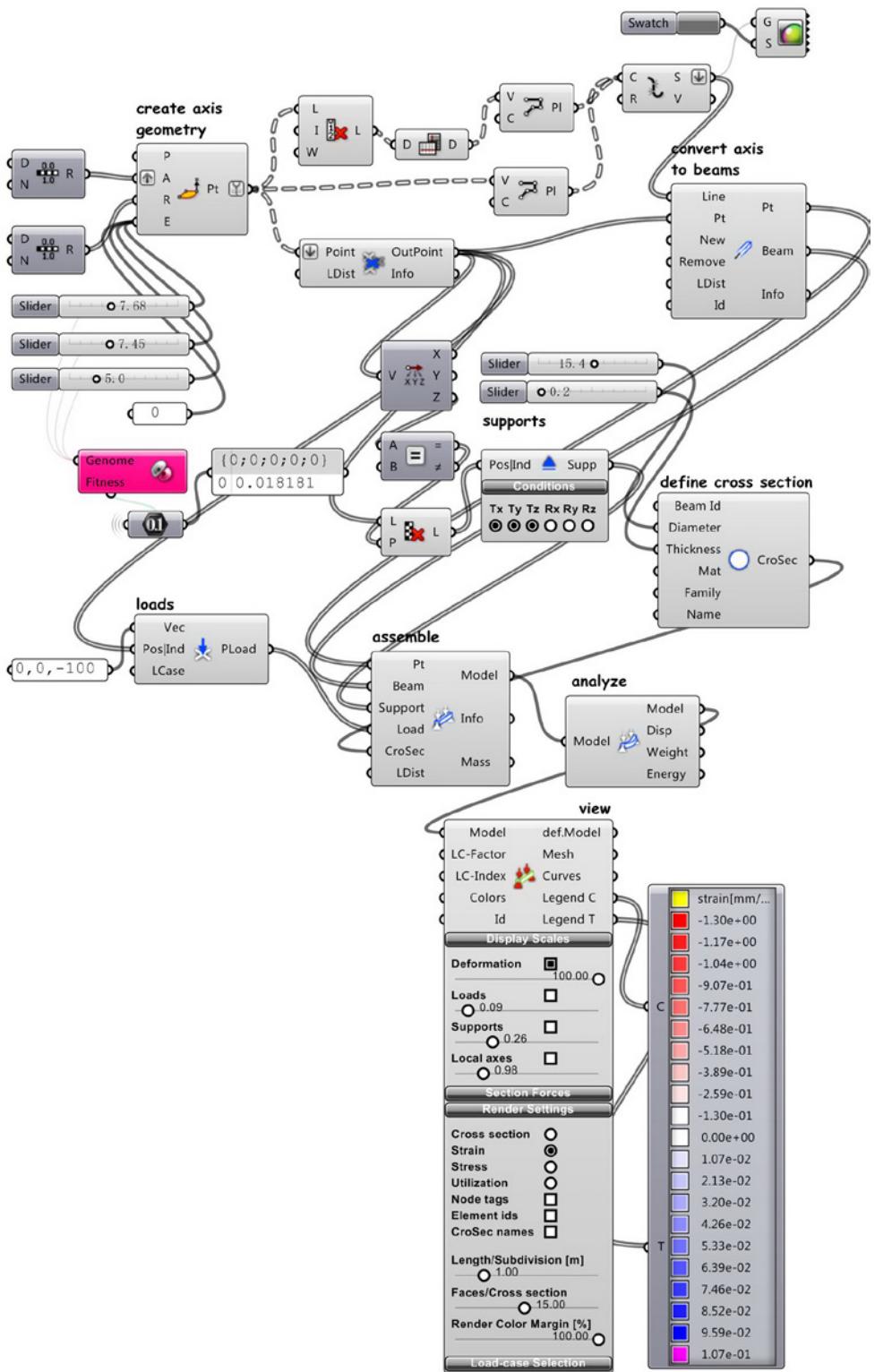


Max Displacement 最大位移=0.01386m

不同几何形式，会改变横梁承受最大内力的位置，解算前（右）第二根横梁承受最大内力，解算后（左）位于地平的支座承受最大的内力。

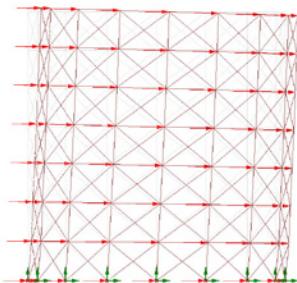
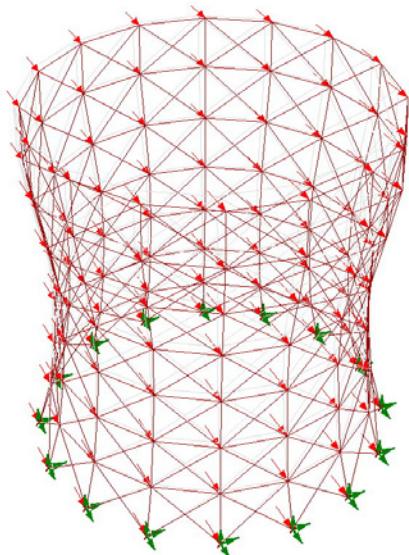


G-03_22

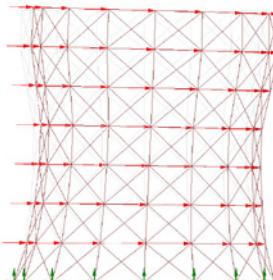


D-3: 几何结构优化 (Galapagos +Karamba)

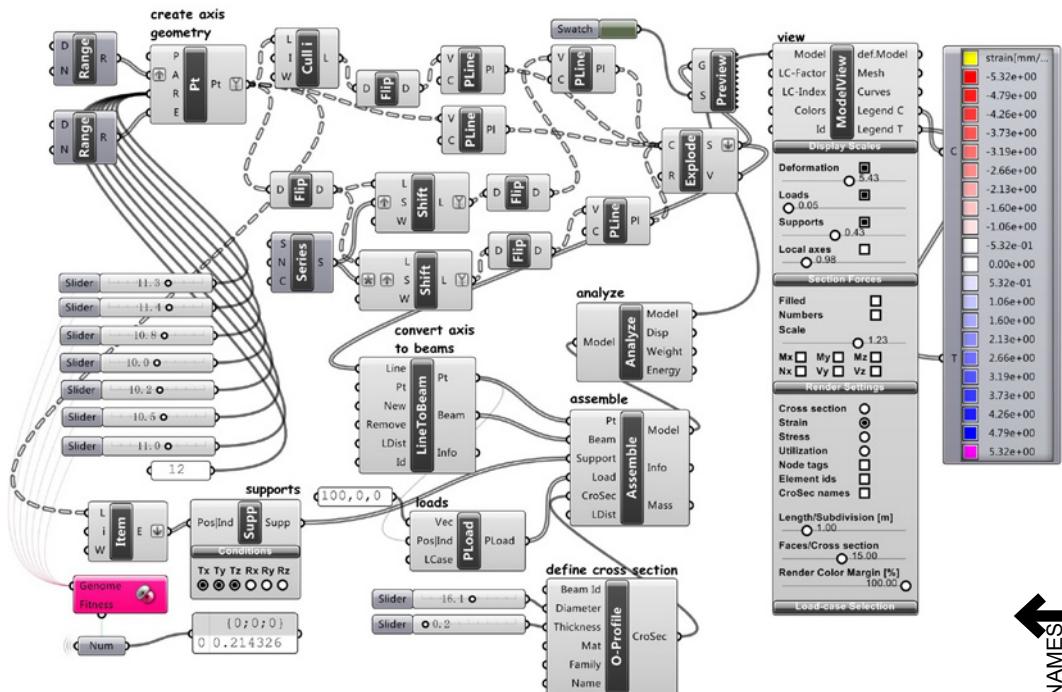
Max Displacement 最大位移=0.238135m



Max Displacement 最大位移=0.211795m

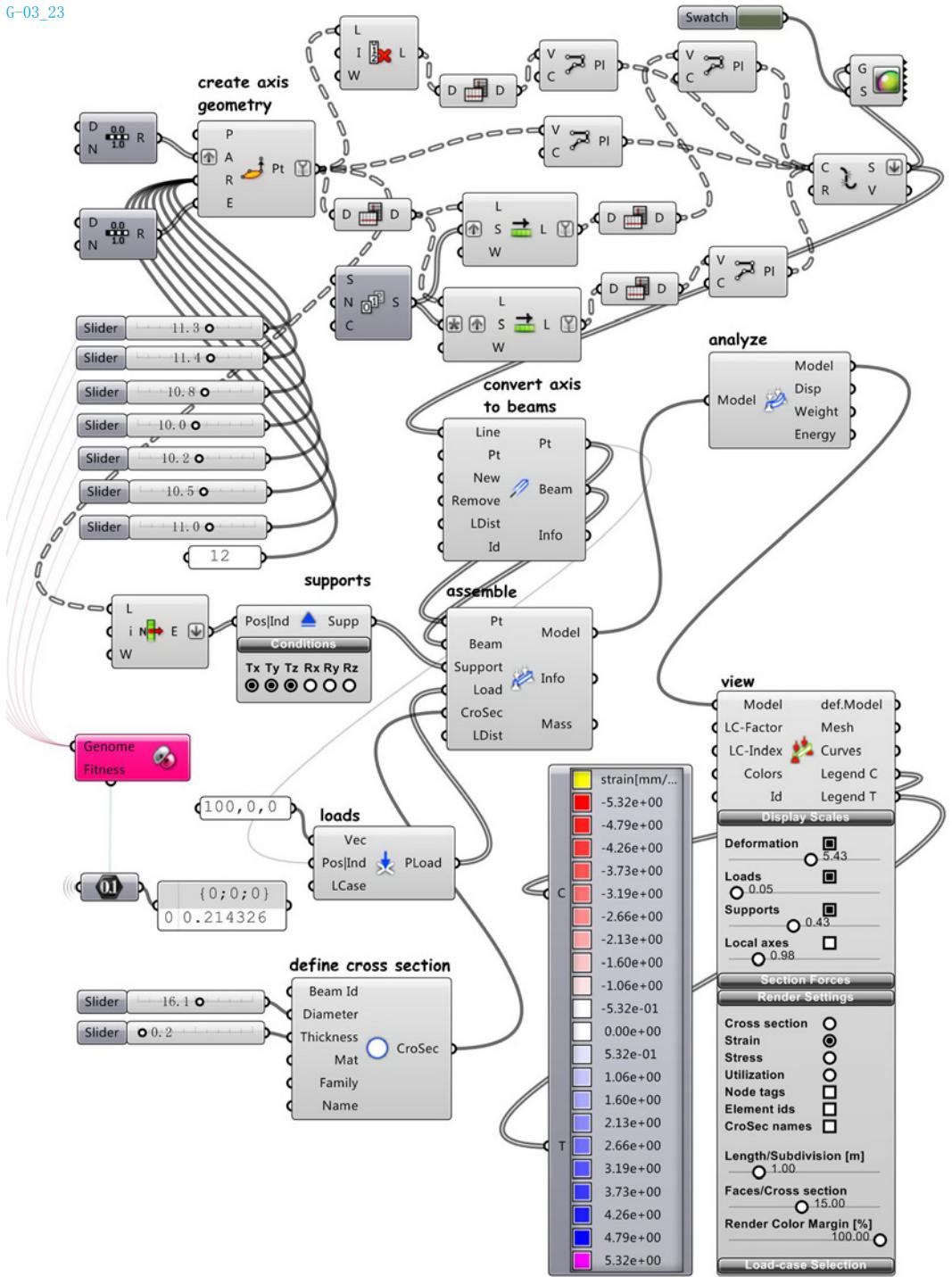


在结构几何上增加了水平载荷，可以理解为风载荷，受到水平载荷影响，结构发生位移，为了获得最小最大位移，将控制几何体类圆多边形的半径输入Galapagos的基因输入端进行解算。



← NAMES

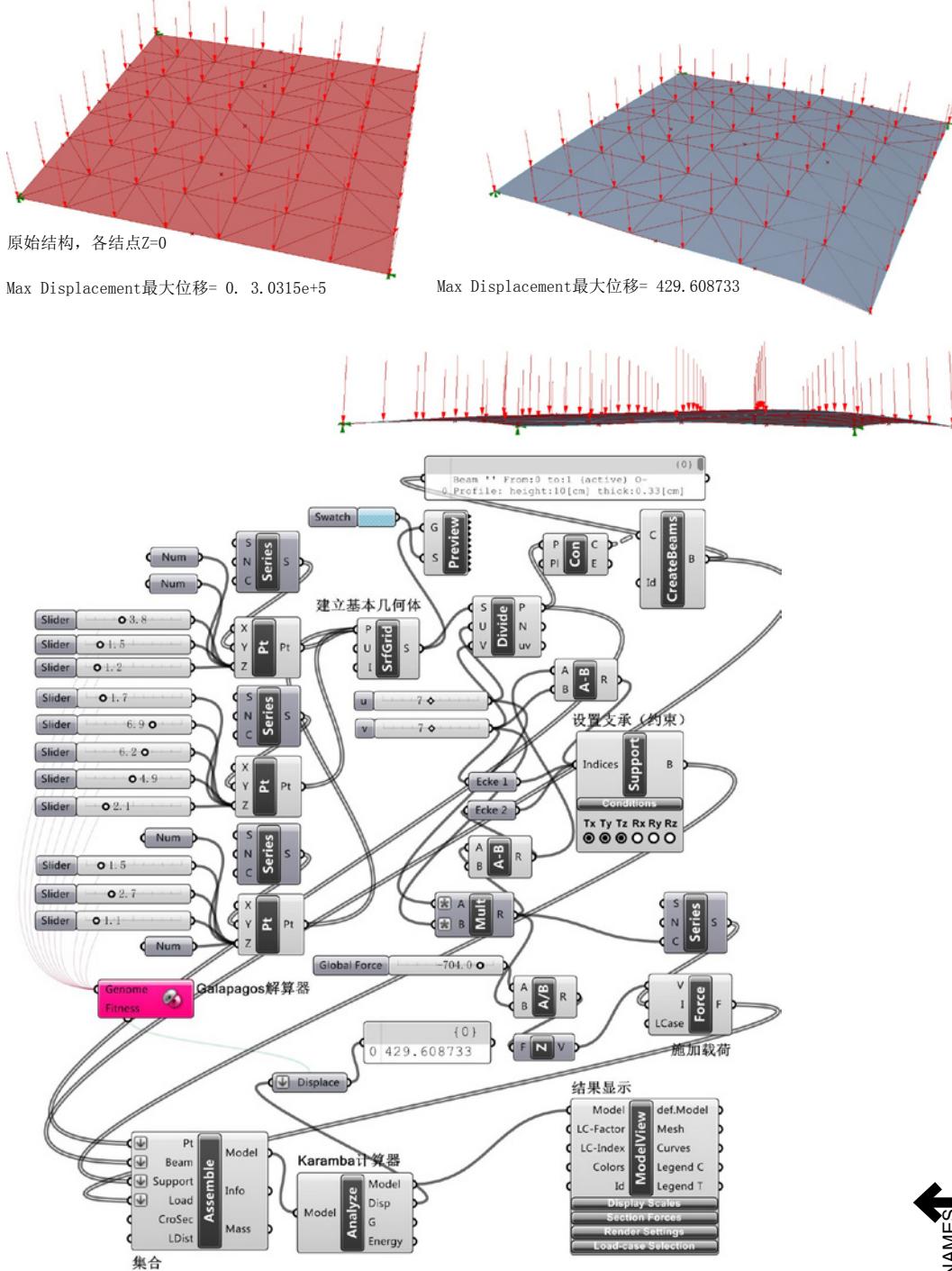
G-03_23



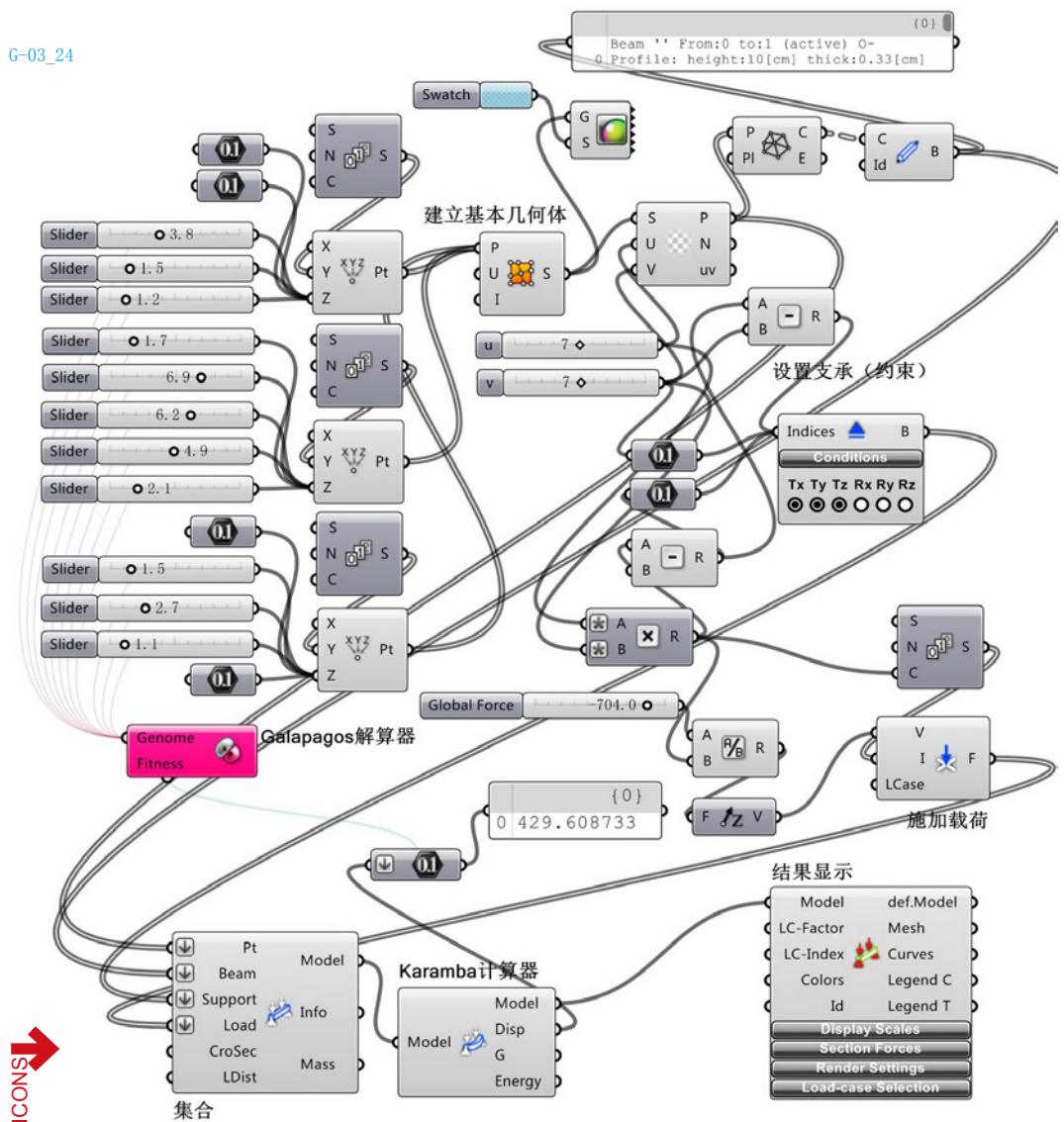
ICONS →

D-4: 几何结构优化 (Galapagos +Karamba)

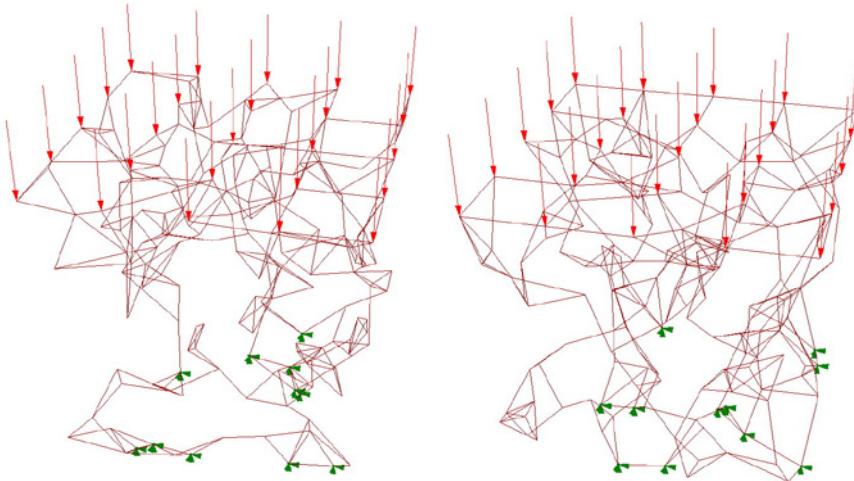
一个平面的网架结构，在受一定载荷作用下，发生极大的变形，将各结点的Z值输入到Galapagos的基因输入端，优化结构几何，可以获得理想的网架几何结构，协助进一步设计。



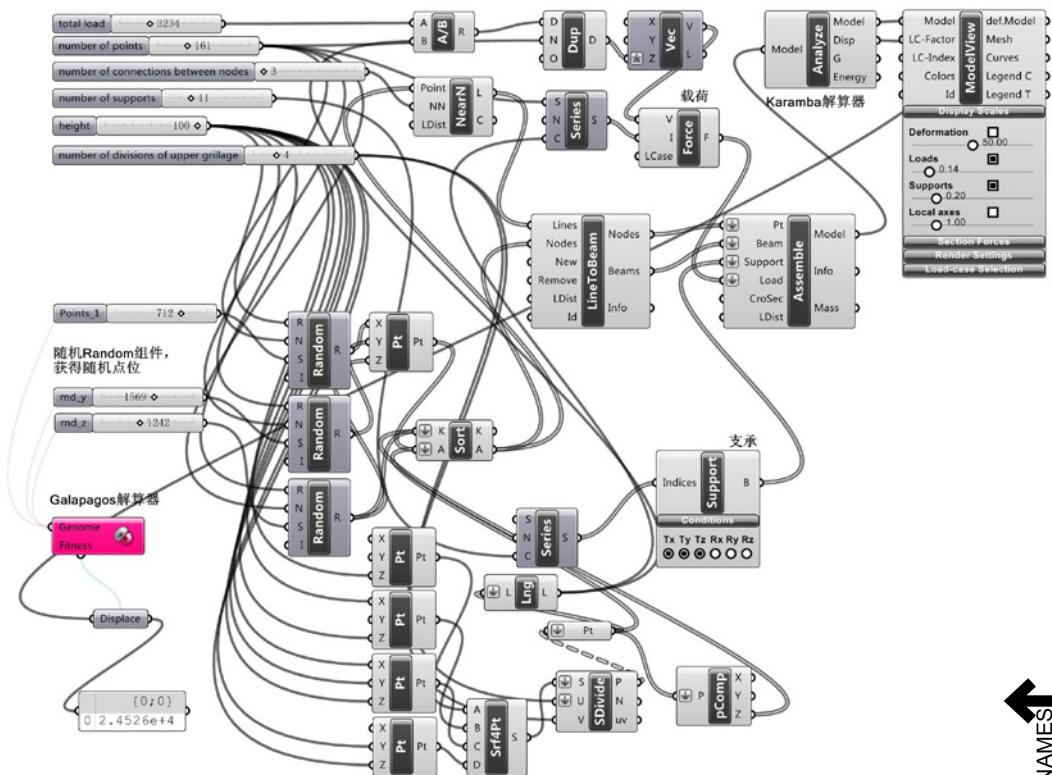
G-03_24



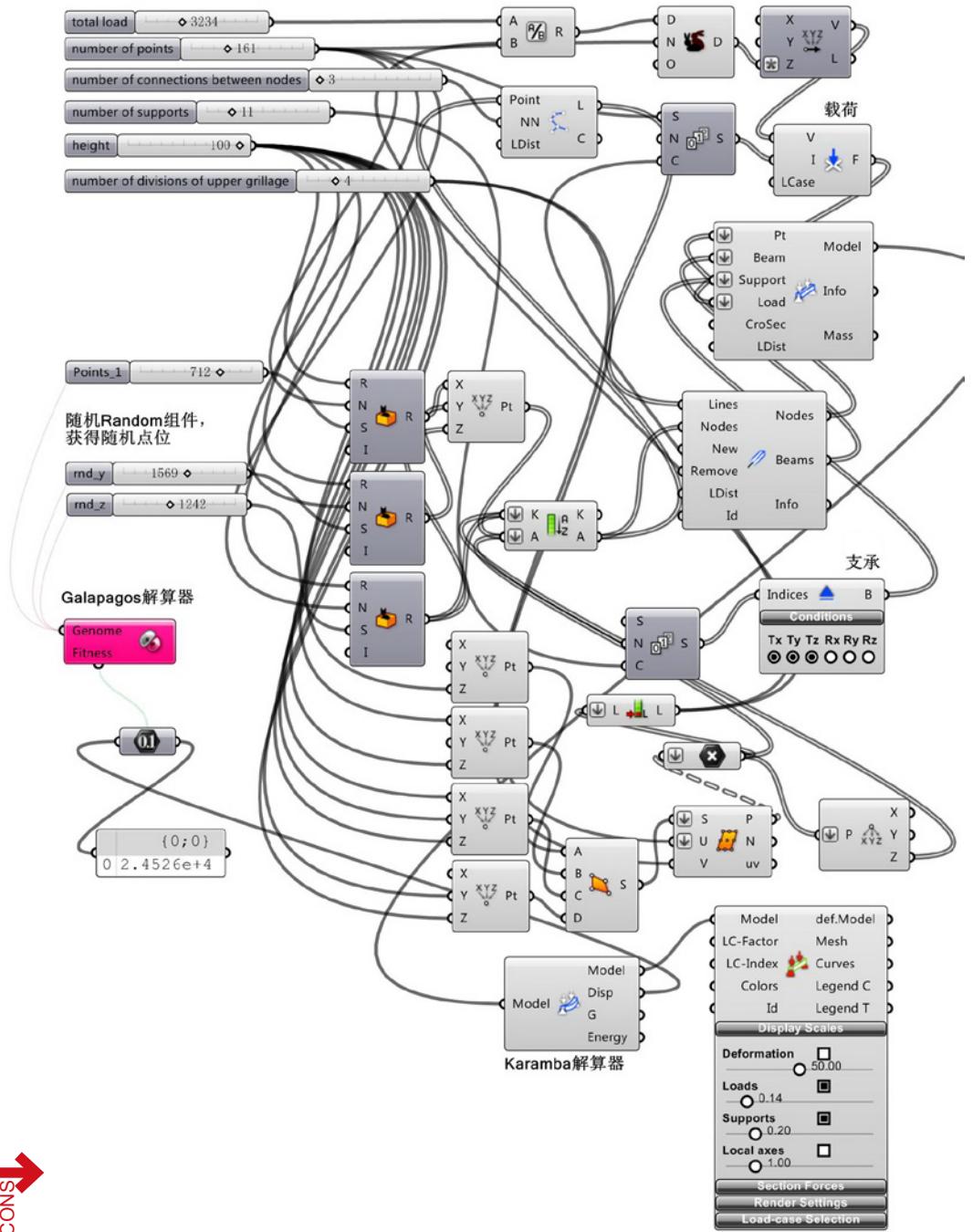
D-5: 几何结构优化 (Galapagos +Karamba)



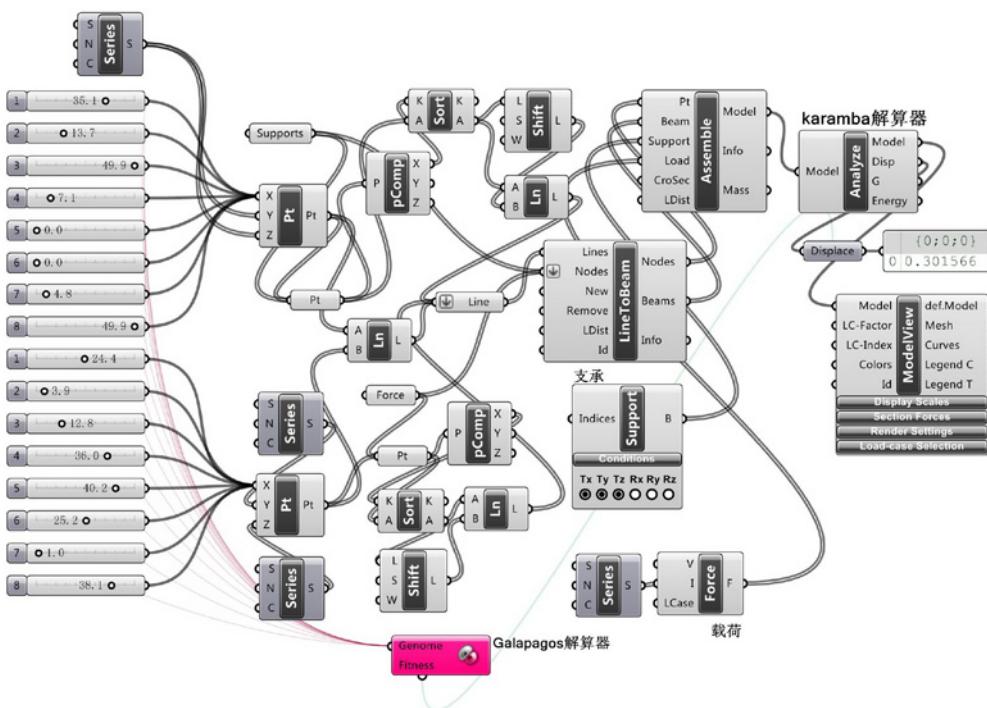
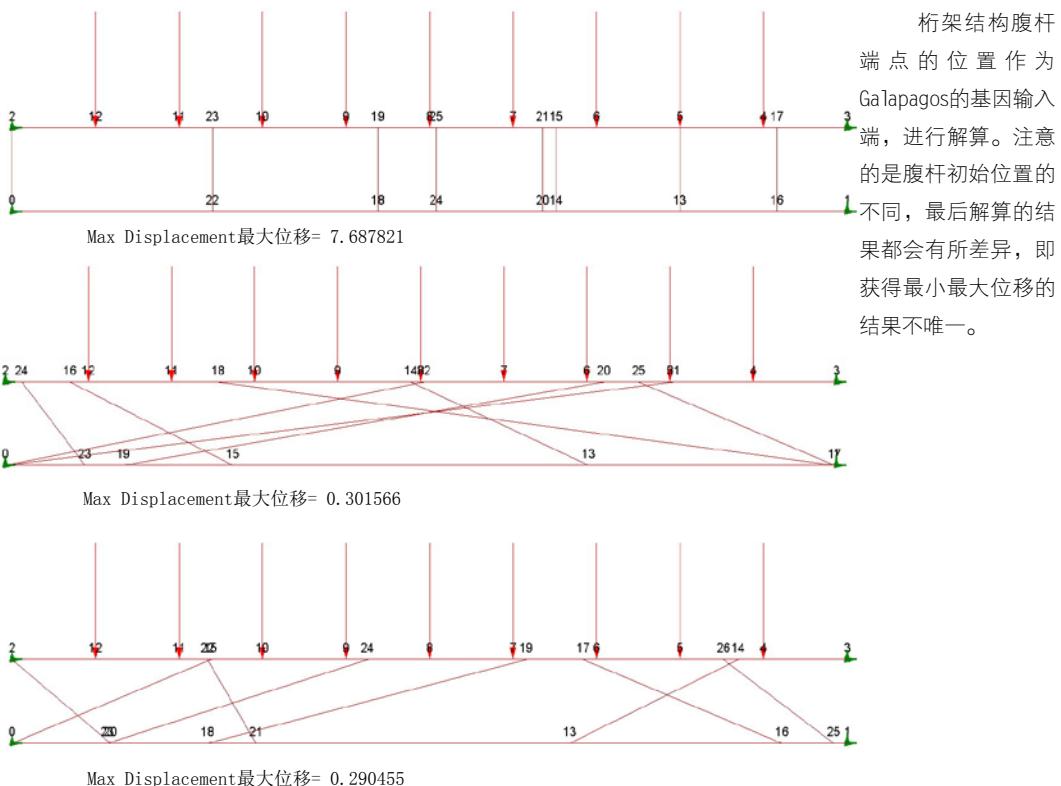
在设计过程中，经常使用随机组件获得自由的设计形式，因此结构合理性的问题变的更加复杂，随机因子的改变可以获得无数的几何形态，哪种几何形态是合理的？使用传统的结构解算方法，而不借助于计算机的模拟，这个问题看来不宜解决。将随机因子连接到Galapagos的基因输入端进行解算，可以获得相对较小最大位移值时的点位。



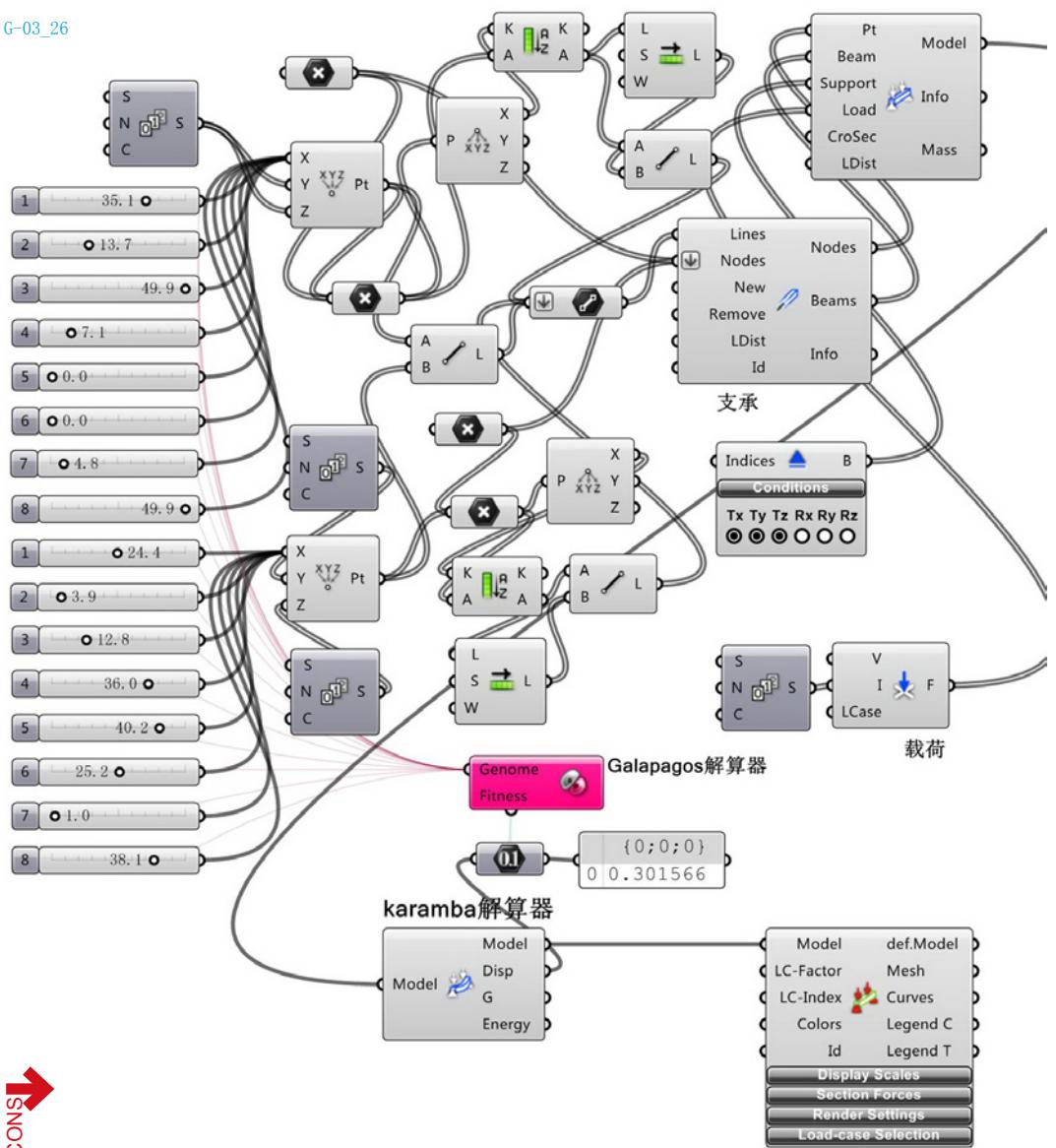
G-03_25



D-6: 几何结构优化 (Galapagos +Karamba)



G-03_26



ICONS

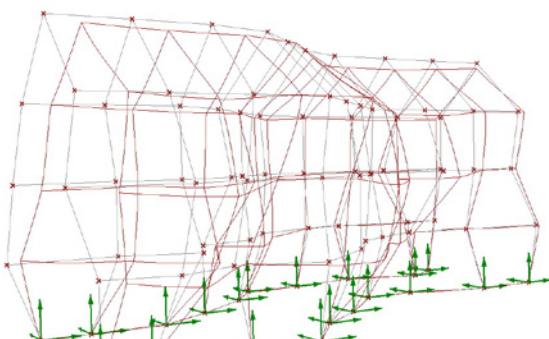
D-7: 几何结构优化 (Galapagos +Karamba)

将之前论述GHPython所建立的建筑在此以Galapagos+Karamba的方式进行结构优化选型，可以获得相对稳定的结构，这与之前随机选型存在本质的区别。在自由变化下更加合理性的设计趋势。

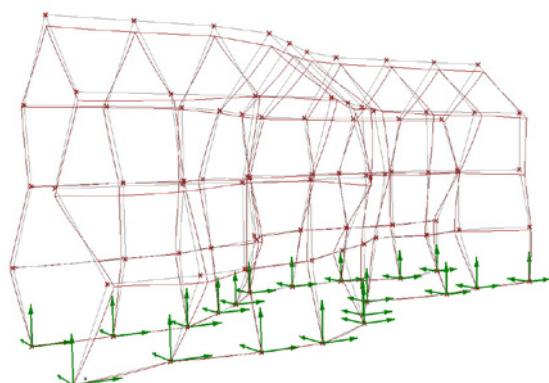
在没有相关可供设计师使用的计算机辅助结构计算平台条件下，在基本的几何形体出来之后，需要与结构师的直接配合，确定几何的合理性，一方面由结构师进行计算，获得判定结果，由建筑师和结构师共同对不合理的几何进行修正，并再次交与结构师进行结构计算；或者在交与结构师进行计算判定之前，由建筑师先使用计算软件做相关的结构优化，再与结构师共同探讨其合理性，对于推动设计的发展是不无益处的。

传统计算结构的方法一般是独立于设计存在，这就在设计与结构之间构成了一定的阻力。在三维软件中构建完设计几何后，需要单独进行处理为结构计算所能识别的文件格式，再在其中加载，在复杂的几何条件下，多数需要重新处理，并设置支承，施加载荷，选择截面，赋予材质，进而进行分析。即使做到此处，但是如何通过控制参数条件优化设计结构呢？如何在设计几何与结构分析之间不断地协同？对于单独处理结构分析的程序存在一定的障碍（这不排除以后设计几何平台与结构分析平台之间接口的开发，获得设计几何与结构分析的同步）。

GH平台的兼容性，本身参数系统以及节点操作模式，和Python等程序语言的共容，使其获得无限的发展空间，可以将几何模型数据，生态设计数据，地理信息数据，动力模拟数据，以及结构分析数据在同一平台下处理并结合，这本身对于设计师而言，就是获得了无限的各类分析模拟，构型与设计的协同，也要求设计师必须调整好自身的知识体系结构，这是与手绘时代截然不同的设计模式，也是与纯粹AUTOCAD计算机绘制设计截然不同的设计模式，这里所反映的是CAD、CAE、CAM相互协同下的设计，而且是同一平台下，最为直接的联合，这与目前软件发展的趋势是一致的，例如IES VE对各种热、光、风等生态环境分析进行整合，CATIA对参数设计，结构分析等各专业联合，跨平台管理等的整合。

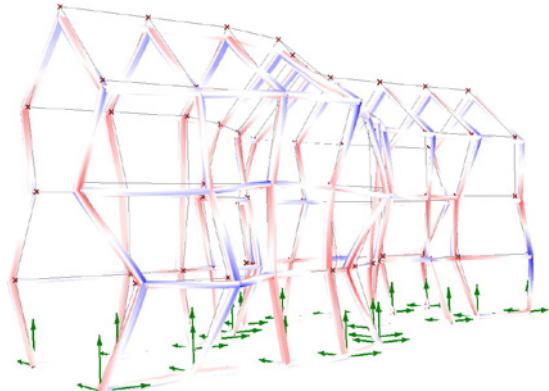


Max Displacement 最大位移= 0.083968(随机选择的位置)

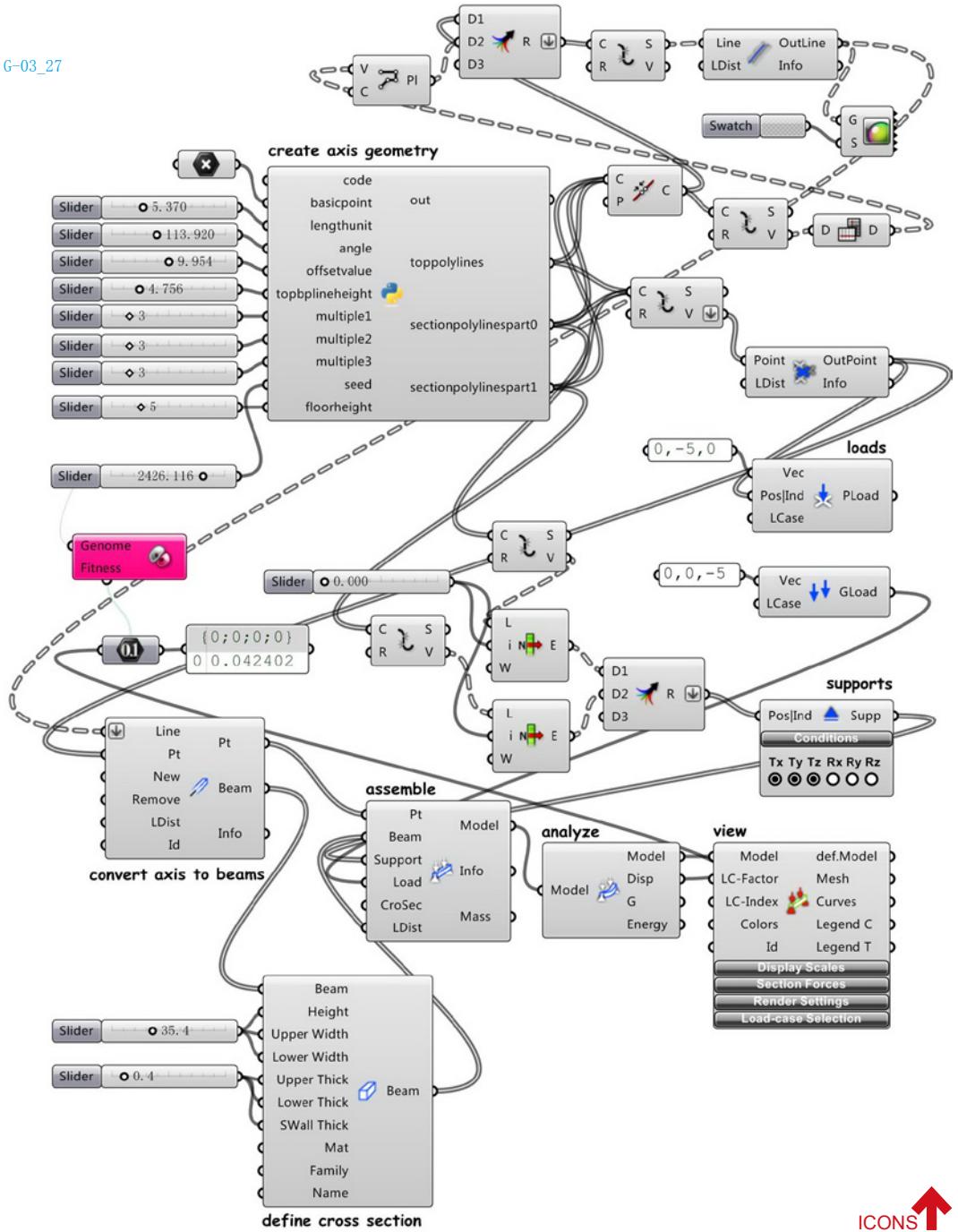


Max Displacement 最大位移= 0.019647(Galapagos解算后)

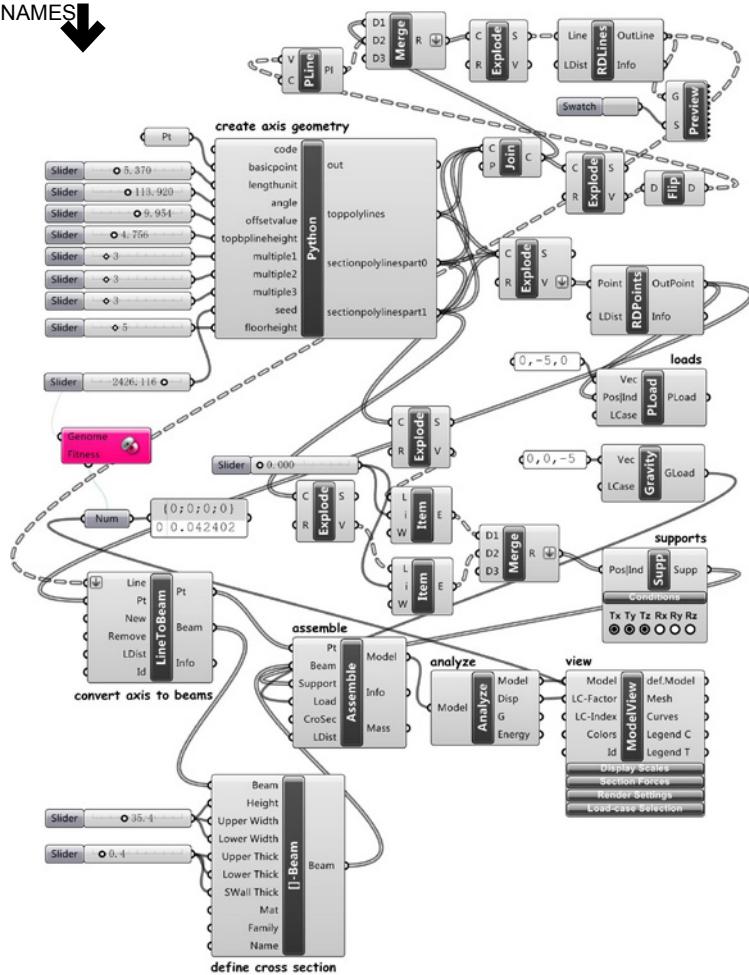
在Galapagos解算之前，通过调整随机因子可以获得一系列相似的几何模型，可是并不能获知哪种模型的结构趋于合理，因此将随机因子Seed端数值输入到Galapagos的基本输入端进行解算，以获得位移最小的结构体系。其中灰色线是未进行结构分析时所建立的模型。



G-03_27

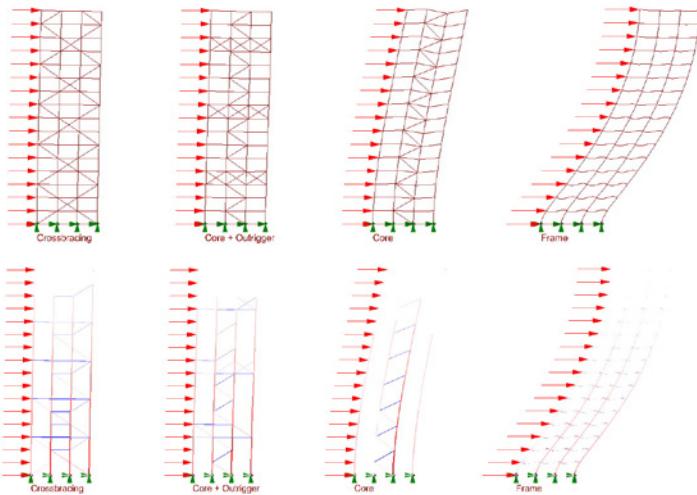


NAMES

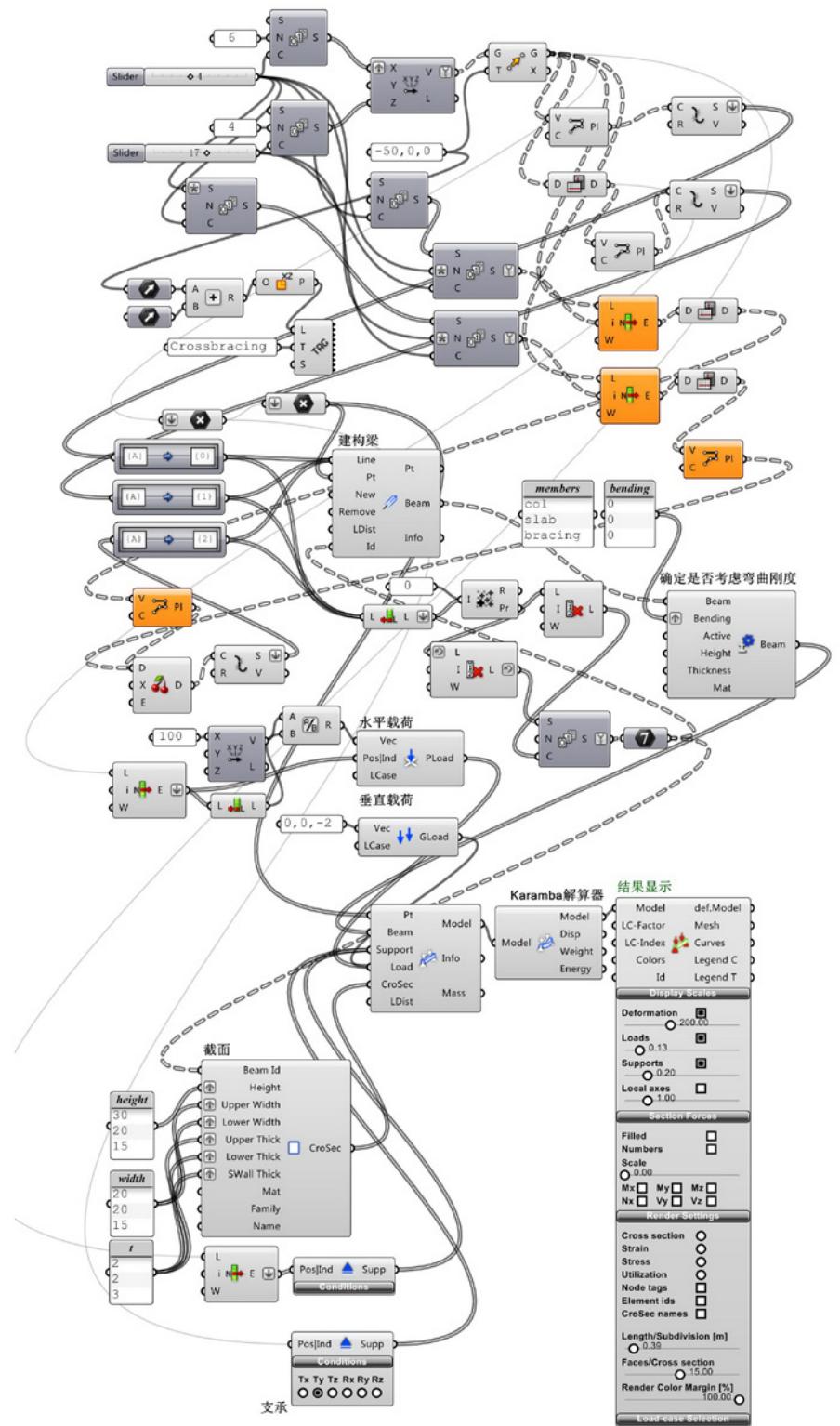


E: 不同桁架结构的位移比较

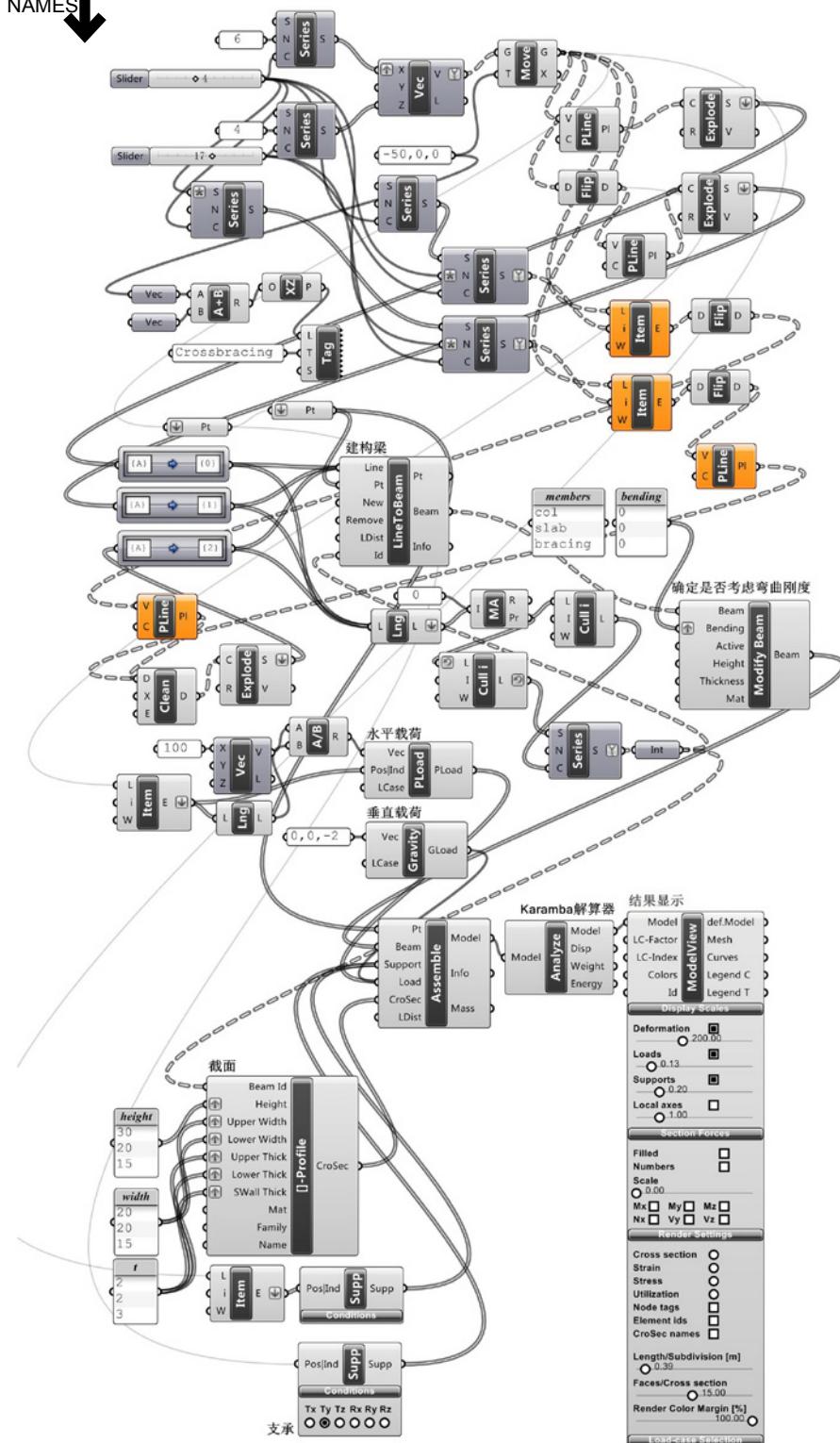
在设计过程中经常涉及到结构选型，对于哪种结构更加合理，可以通过同等载荷条件下，进行不同结构几何的解算进行比较研究。



G-03_28

↑
ICONS

NAMES



G: Eigen-modes of Structure结构本征模

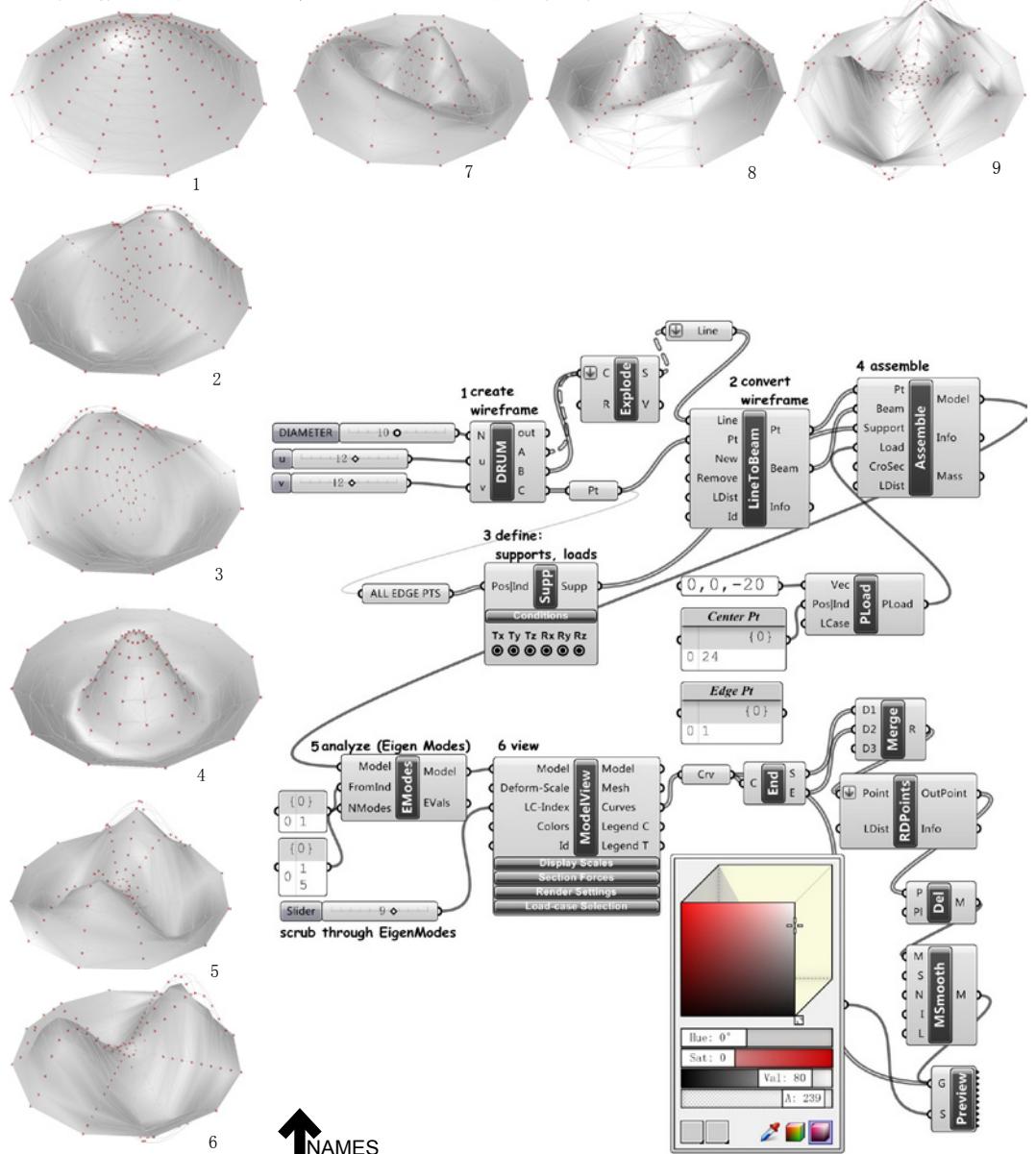
结构的本征模描述了结构几何最易变化的一连串形态趋势，而变化的第一个结构几何是最易获得的。当模数设置的越高，越多的外力将作用与结构几何。本征模在工程领域主要有两个应用：

1-结构动力学；

2-临界纵向载荷的稳定性分析；

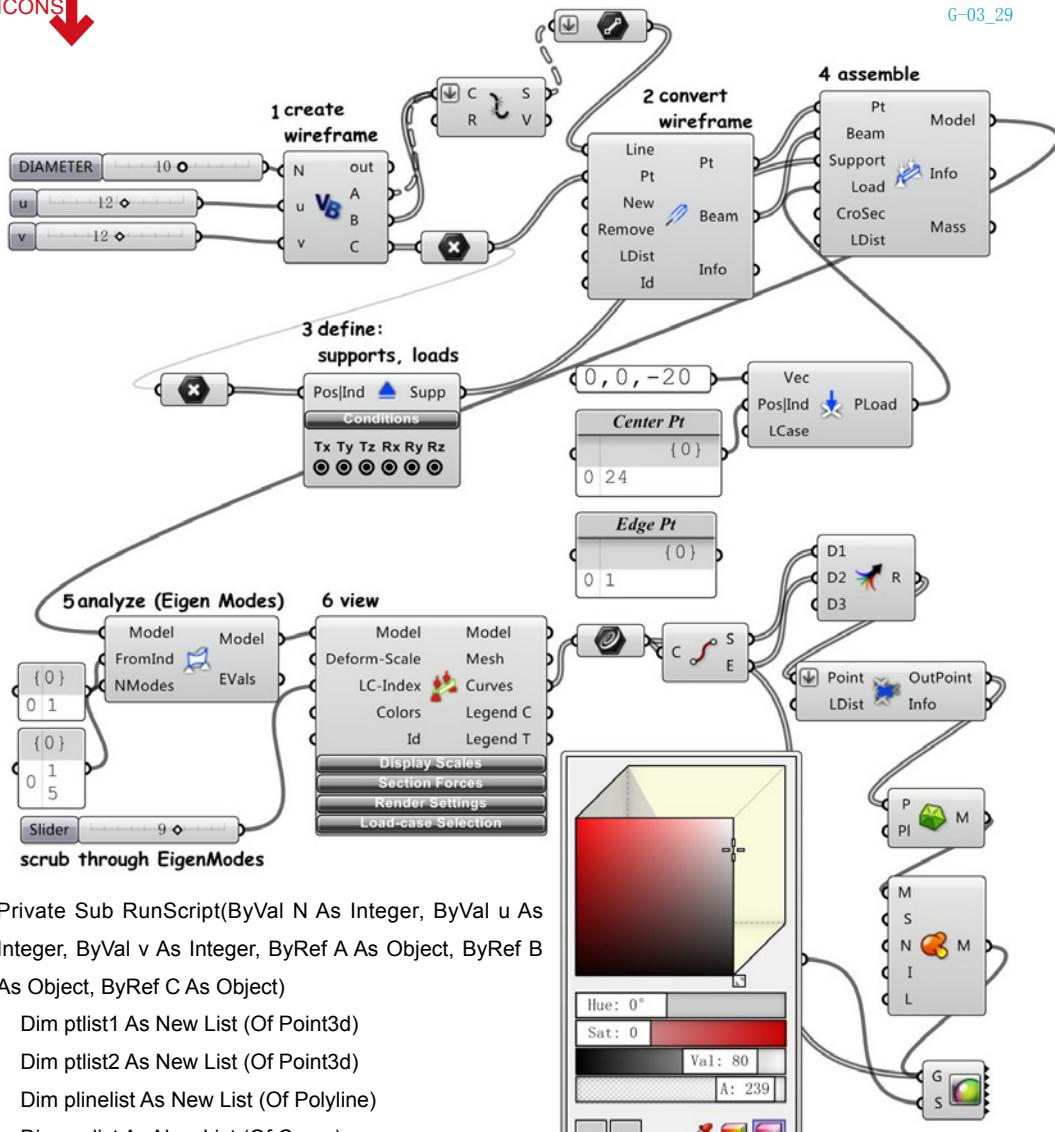
1.结构动力学：

当敲打鼓的中心时，会听到鼓皮的振动，这一系列的振动模式由第一个自然振荡所确定。同时振动的模式受到频率与结构的刚度和支承影响，以及质量的分布和击打的位置。



ICONS

G-03_29



```
Private Sub RunScript(ByVal N As Integer, ByVal u As Integer, ByVal v As Integer, ByRef A As Object, ByRef B As Object, ByRef C As Object)
```

```
    Dim ptlist1 As New List(Of Point3d)
    Dim ptlist2 As New List(Of Point3d)
    Dim plinelist As New List(Of Polyline)
    Dim crvlist As New List(Of Curve)
    Dim circlist As New List(Of Circle)
```

```
    Dim dtree1 As New DataTree(Of Polyline)
    Dim dtree2 As New DataTree(Of Point3d)
    Dim dtree3 As New DataTree(Of Point3d)
    Dim dtree4 As New DataTree(Of Point3d)
```

```
'CREATE CIRCLES
```

```
For i As Integer = 1 To u
```

```
    Dim cpoint As New Point3d(0, 0, 0)
```

```

Dim circ1 As New Circle(i * (N / u)) : circ1.Center = cpoint
Dim domain As New Interval(0, 1)
Dim crv As Curve = circ1.ToNurbsCurve : crv.Domain = domain
crvlist.Add(crv)

```

Next

'ADD CONCENTRIC POLYLINES

```
Dim arraycount As Integer = 0
```

```
'Dim stp As Double = 1 / v
```

```
For i As Integer = 0 To crvlist.Count - 1
```

```
    Dim path As New GH_Path(arraycount)
```

```
    For j As Double = 0 To (v - 1)
```

```
        Dim jval As Double = j * (1 / v)
```

```
        Dim pt As Point3d
```

```
        Dim crv1 As Curve = crvlist.Item(i)
```

```
        pt = crv1.PointAt(jval)
```

```
        dtree2.Add(pt, path)
```

```
        dtree3.Add(pt, path)
```

Next

```
    dtree2.Add(dtree2.Branch(arraycount).Item(0), path)
```

```
    Dim pline As New Polyline(dtree2.Branch(arraycount))
```

```
    dtree1.Add(pline, path)
```

```
    arraycount = arraycount + 1
```

Next

```
Dim arraycount2 As Integer = 0
```

```
For i As Integer = 0 To dtree3.Branch(0).Count - 1
```

```
    Dim path As New GH_Path(arraycount2)
```

```
    For j As Integer = 0 To dtree3.BranchCount - 1
```

```
        dtree4.Add(dtree3.Branch(j).Item(i), path)
```

Next

```
    arraycount2 = arraycount2 + 1
```

Next

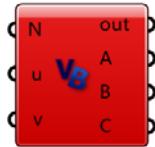
```
CreatePolyLines(dtree4, plinelist)
```

```
A = dtree1
```

```
B = plinelist
```

```
C = dtree3.Branch(dtree3.BranchCount - 1)
```

```
End Sub
```



'<Custom additional code>

Sub CreatePolyLines(ByRef dtree As DataTree (Of Point3d), ByVal plinelist As List (Of Polyline))

For i As Integer = 0 To dtree.BranchCount - 1

Dim ptlist As New List (Of Point3d): Dim pt As New Point3d(0, 0, 0): ptlist.Add(pt)

For j As Integer = 0 To dtree.Branch(i).Count - 1

ptlist.Add(dtree.Branch(i).Item(j))

Next

Dim pline As New Polyline(ptlist)

plinelist.Add(pline)

Next

End Sub

'</Custom additional code>

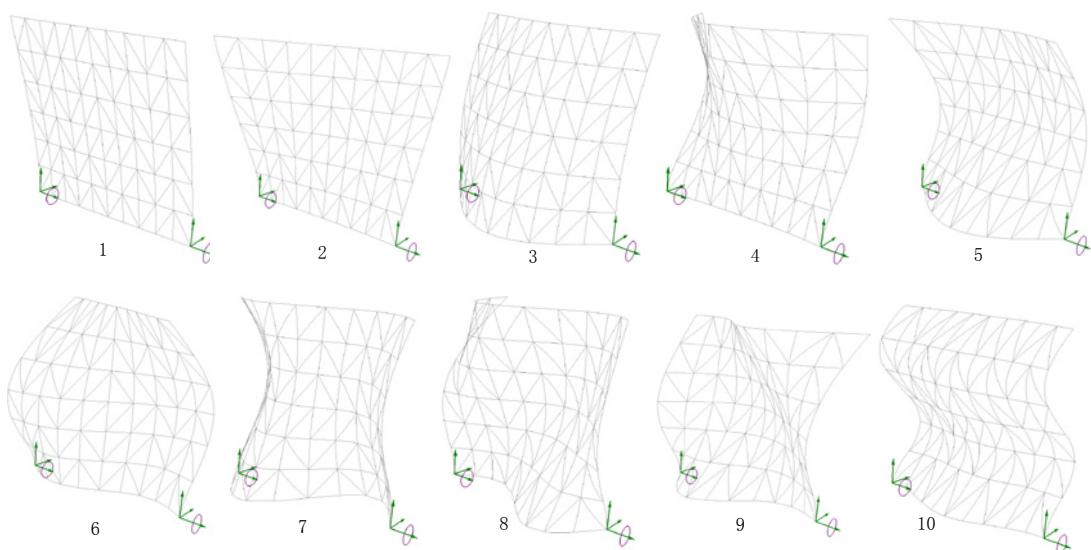
End Class

2. 临界纵向载荷的稳定性分析：

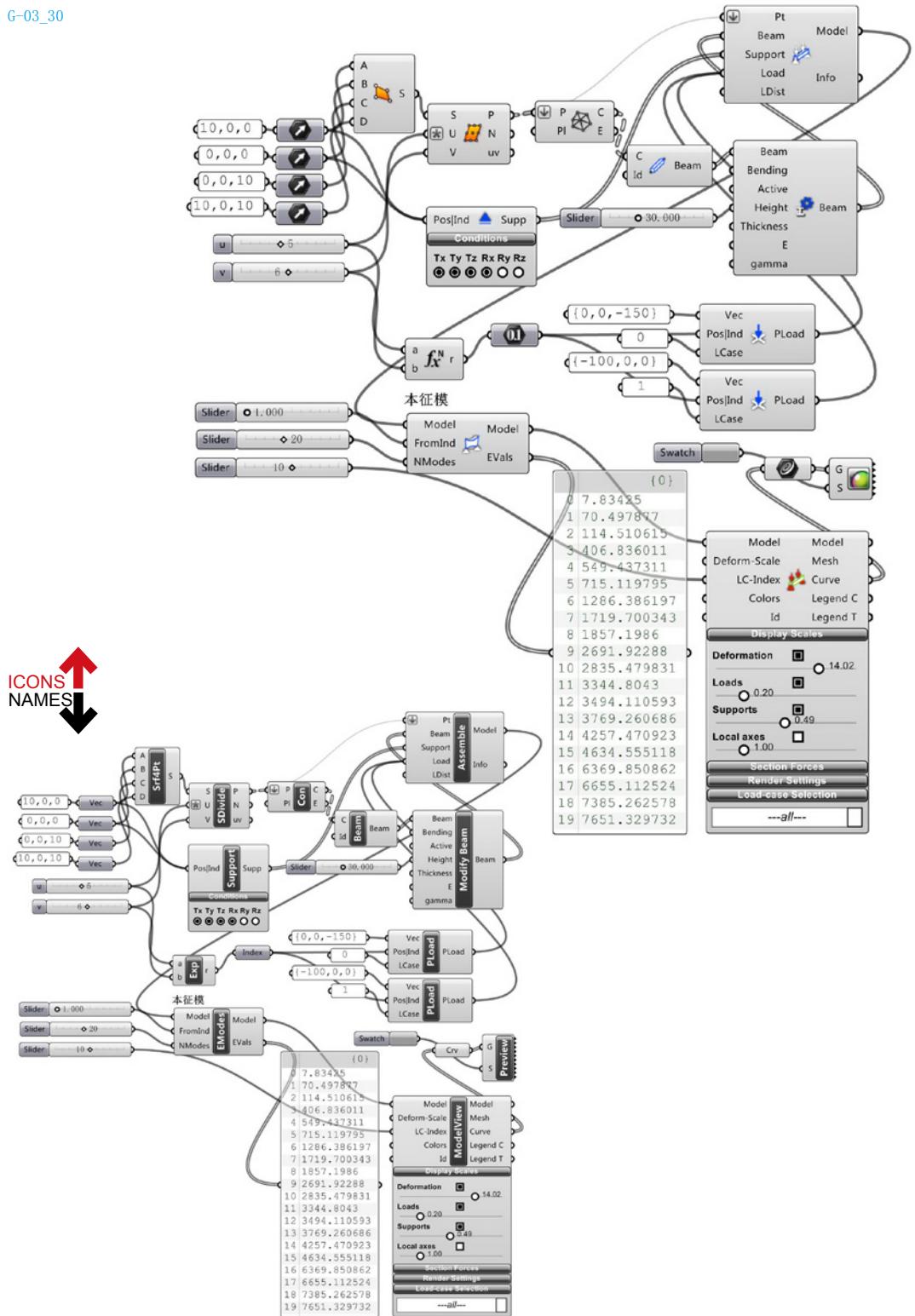
非稳定结构可以被看作是振动缓慢的静力系统，使永远保持其初始的状态。这也是在变形计算中，惯性力不起主要作用的原因。否则振动明显，具有相似变形。作用于结构上的载荷决定结构变形的方式。

本征模描述了不同支承条件下梁的变形。越多固定的自由度，需要越高的载荷使其变形。

Karamba的本征模分析EigenMode可以计算本征模，并回应以结构的本征值用于非惯性力下的变形分析。FromInd输入项为当前计算的本征模阶段，NModes输入项可以设置计算次数。输出结果可以在ModelView中显示，以便获取几何形式或者分析结构优化。而使用本征模分析时，所有的载荷都被移除。

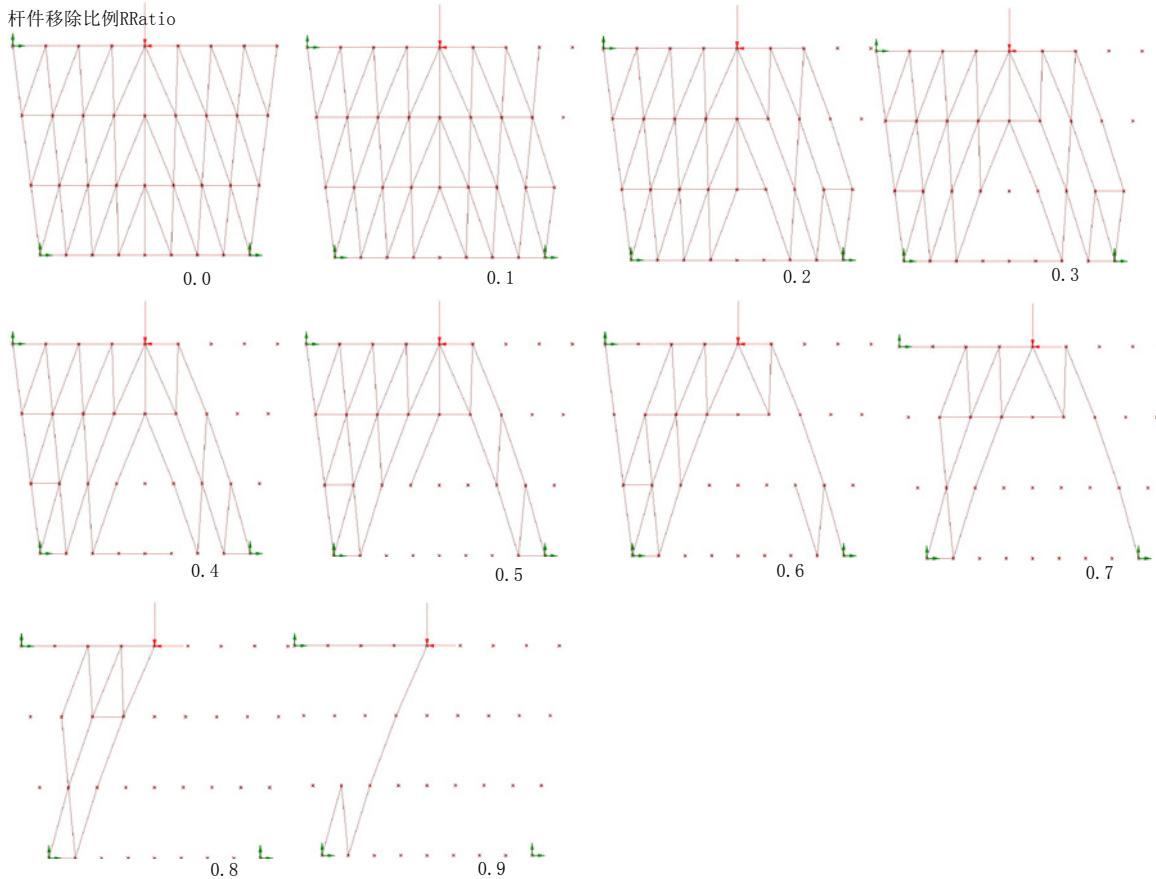


G-03_30

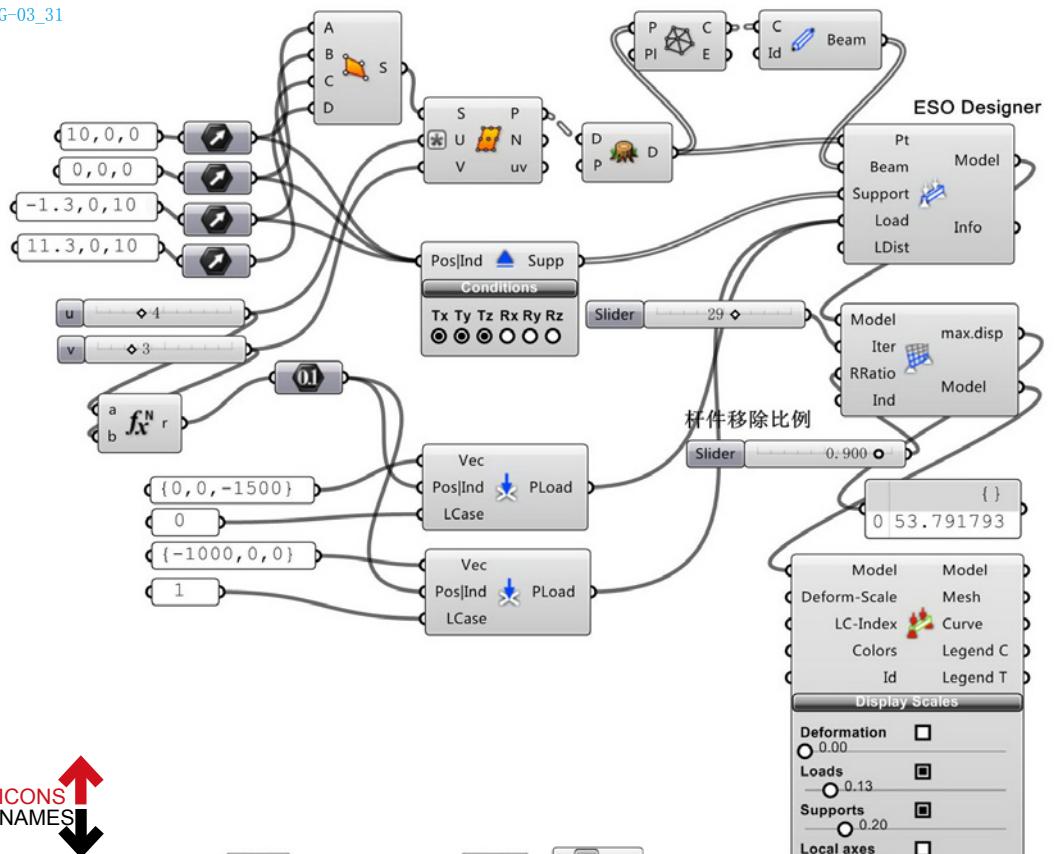


H: Evolutionary Structural Optimization(ESO) ESODesigner 组件

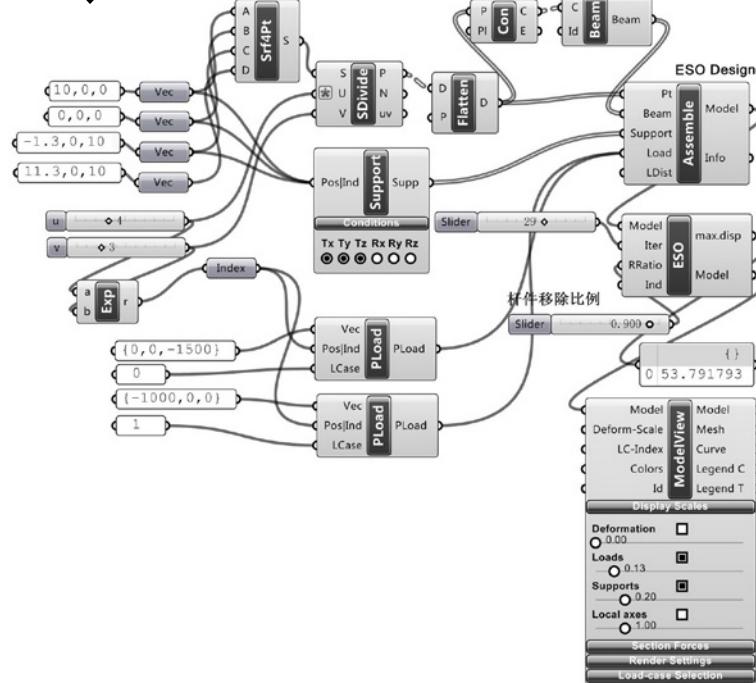
使用ESO Design可以在保证最大位移值保持最小的条件下移除结构杆件，在本例中可以明显看到杆件移除的次序从对受载荷影响最小的区域开始。



G-03_31



ICONS
NAMES



Ansys Workbench

<http://www.ansys.com/> 官方网址



ANSYS软件是融结构、流体、电磁场、声场和耦合场分析于一体的大型通用有限元分析软件。由世界上最大的有限元分析软件公司之一的美国ANSYS公司开发，它能与多数CAD软件接口，实现数据的共享和交换，如Pro/Engineer、NASTRAN、Alogor、I-DEAS、AutoCAD等，是现代产品设计中的高级CAE工具之一。因此它可应用于以下工业领域：航空航天、汽车工业、生物医学、桥梁、建筑、电子产品、重型机械、微机电系统、运动器械等。

软件结构：

软件主要包括三个部分：前处理模块，分析计算模块和后处理模块。

前处理模块提供了一个强大的实体建模及网格划分工具，用户可以方便地构造有限元模型。

分析计算模块包括结构分析（可进行线性分析、非线性分析和高度非线性分析）、流体动力学分析、电磁场分析、声场分析、压电分析以及多物理场的耦合分析，可模拟多种物理介质的相互作用，具有灵敏度分析及优化分析能力。

后处理模块可将计算结果以彩色等值线显示、梯度显示、矢量显示、粒子流迹显示、立体切片显示、透明及半透明显示（可看到结构内部）等图形方式显示出来，也可将计算结果以图表、曲线形式显示或输出。

软件提供了100种以上的单元类型，用来模拟工程中的各种结构和材料。该软件有多种不同版本，可以运行在从个人机到大型机的多种计算机设备上，如PC、SGI、HP、SUN、DEC、IBM、CRAY等。

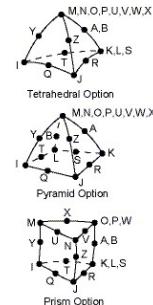
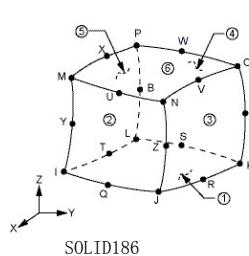
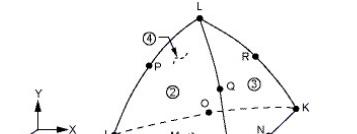
Ansys Workbench下的线性静力结构与动力学分析

Ansys Workbench下的线性静力结构分析是用来分析结构在给定静力载荷作用下的响应。一般关注的是结构的位移，约束反力，应力以及应变的参数。在Workbench中线性静力学分析由Mechanical模块求解，一般可以使用集成的Static Structural分析系统。

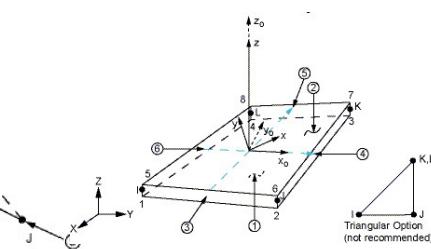
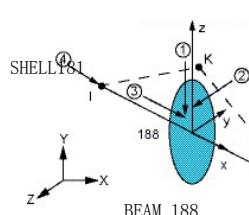
几何模型：

静力结构分析中，Mechanical一般支持实体、壳体、梁和点质量（Point Mass）四种：

- 实体，程序默认的单元是10节点的四面体单元（SOLID187）和20节点的六面体单元（SOLID186）；



- 壳体，首先要确定其厚度，其单元出程序采用4节点的四边形壳单元（SHELL181）来划分，其截面（和偏置）不需定义；



- 梁，其截面形状和方向，在DesignModeler中预先指定，并自动传到仿真模型中。梁实际是采用2节点的梁单元（BEAM188单元）划分，该单元支持截面的定义及偏置；
- 质量点（Point Mass），没有明确的建模重量，但是只有面实体才能定义质量点。可以使用坐标系（x, y, z）来定义其位置，也可以选择点/边/面定义位置。其重量/质量大小在Magnitude项输入。在结构静力分析中，质量点只受到“加速度”（标准重力加速度）和“旋转速度”的作用，只有惯性力才会对质量点起作用。线性静力结构分析中引入质量点是为了考虑结构中没有建模的附加重量，同时有惯性力出现。

材料属性：

在线性结构分析中，材料属性一般只需定义杨氏模量及泊松比：

- 假如有任何惯性载荷，必须定义材料的密度；
- 热膨胀系数和传热系数在有热载荷的时候才需要确定；
- 若要进行疲劳分析，需要Fatigue Module add-on license；

装配体：

工程中经常会涉及到装配体，当在Ansys Workbench中导入装配体时，程序就会在两个实体之间自动生成接触面：

- 在两个不同的实体边界上允许有不匹配的单元划分（不同实体上单元节点可以不重合）；
- 可以在Contact菜单中，指定自动探测接触距离的容差；
- 接触区域中一个表面构成“接触”面，则此区域的另一个表面构成“目标”面。在默认情况下，Workbench对装配体定义的是对称接触，并可以根据分析需要选择非对称接触；

Ansys Workbench中有四种接触类型，绑定接触，不分离接触，无摩擦接触和粗糙接触。其中绑定和不分离接触是基础的线性行为，求解时仅需要迭代一次；无摩擦和粗糙接触时非线性行为，求解需要迭代多次。

载荷及约束：

在Ansys Workbench中有四种类型的结构载荷：

- 惯性载荷，其对整个系统均有作用，并且当涉及到质量的时候，必须输入材料的密度；
- 结构载荷，作用在结构上的力或者力矩；
- 结构支承（约束），利用约束来限制结构部件在一定范围内的移动；
- 热载荷，会产生温度并且在整个模型上引起热扩散；

Mechanical中常见载荷：

- 加速度Accelerate与重力加速度Atandard Earth Gravity（惯性载荷）

程序内部加速度时通过惯性力施加到结构上的，而惯性力的方向与所施加的加速度方向相反，因为惯性力将阻止加速度所产生的变化；

- 旋转速度Rotational Velocity

属于惯性载荷，默认情况下旋转速度的单位是每秒转过的弧度值；

- 力Force和压力Pressure载荷

力载荷可以施加在结构的最外面，边缘或者表面等位置。当一个力施加到两个同样的表面上时，每个表面将承受这个力的一半，即力可以通过定义矢量，大小以及分量来施加；

压力载荷只能施加在表面，并且方向通常与表面的法向方向一致，其正值代表进入表面（例如压缩），负值代表从表面出来；

- 力矩载荷 Moment

力矩可以施加在任意实体的表面。假如选择了多个表面，力矩将分摊在这些表面上。力矩也可以通过矢量及其大小定义。当用矢量表示时，力矩遵守右手螺旋法则；

- 远端载荷 Remote Force

对远端载荷的理解可以先理解理论力学（刚体力学）中力的平移定理，固体力学中的圣维南原理以及力学中的等效原理。例如，在某一面上加载了一个远端载荷后相当于在这个面上将得到一个等效的力加上由于偏置的力所引起的力矩，而这个力分布在表面上，但是包括由于力偏置而引起的外力偶矩；

- 螺栓载荷 Bolt Pretension

在Mechanical中加载螺栓连接，只能在3D模拟中加载。螺栓载荷能够加载在圆柱形表面上，但是需要一个以z轴为主方向的局部坐标系；

- 热载荷 Thermal Condition

热载荷可以施加在模型上，任何温度载荷都可以施加。在Mechanical中通常首先进行热分析，然后在结构分析时将计算所得到的温度作为外载荷输入。但是，模型中的温度会引起热膨胀，而热应变自身并不能引起应力，只有在约束，温度梯度或者热膨胀系数不相匹配时才会产生内应力；

Mechanical中常见的支承约束

- 固定约束 Fixed Support

固定约束加载于实体上，则限制x, y和z方向上的平移；加载在顶点，边缘或面上，则约束所有的自由度；若加载于壳或梁，则限制x, y和z方向的平移和转到。

- 给定位移 Displacement

可以在顶点，体边缘或面上加载已知的位移；

允许在x、y和z方向上给予强制位移；

当输入0值时，代表此方向上被约束；

如果不设定某个方向的值，则意味着实体在这个方向上不受约束，即能自由移动；

- 无摩擦约束 Frictionless Support

无摩擦约束实际是在面上施加了法向约束。对称实体受到对称的外载荷时，这个约束可以作为对称面的边界条件，因为对称面等同于约束了法向位移；

- 圆柱面约束 Cylindrical Support

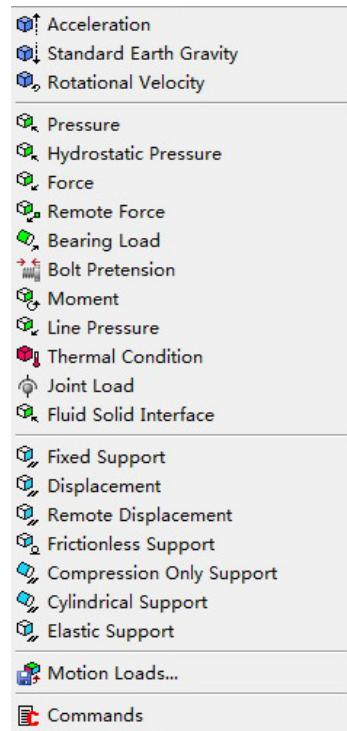
仅适用于小变形（线性）分析，可以施加在圆柱表面，并且可以指向轴向，径向或者切向自由度的约束状况；

- 只有压缩的约束 Compression Only Support

只有压缩的约束是指给几何体表面施加只有法向压缩的约束。这个约束仅仅限制这个表面的边缘或者顶点上，当然简单（支）约束仅限制平移，但是多有旋转都是自由的；

- 固定旋转 Fixed Rotation

主要应用与壳或者其他体的表面，边缘或者定点上，它仅约束旋转，平移的约束不受限制；



Ansys Workbench下的动力学分析可以处理不能够简化为结构静力学的工程问题，动力学的通用运动方程为：不同分析类型对应求解不同形式的方程，如：

$$[\mathbf{M}] \{ \ddot{\mathbf{u}} \} + [\mathbf{C}] \{ \dot{\mathbf{u}} \} + [\mathbf{K}] \{ \mathbf{u} \} = \{ \mathbf{F}(\mathbf{t}) \}$$

- 对于模态分析，则 $\mathbf{F}(\mathbf{t}) = 0$, $[\mathbf{C}]$ 一般忽略；
- 对于谐响应分析, $\mathbf{F}(\mathbf{t})$ 和 $\mathbf{u}(\mathbf{t})$ 都假设为谐函数, 例如 $Z \sin(wt)$, 其中 Z 是振幅, w 频率, 单位是 rad/s; 对于瞬态动力学分析, 方程保持上述形式;

求解通用运动方程有两种主要方法, 即模态叠加法和直接积分法。其中模态叠加法是确定结构的固有频率和模态, 乘以正则化坐标, 然后加起来计算位节点的位移解。这种方法可以用来进行瞬态和谐响应分析。直接积分法是直接求解运动方程。对于谐响应分析, 由于载荷与响应都假设是谐函数, 所以运动方程是力的频率函数, 而不是以时间函数的形式来写出并求解的, 至于瞬态分析, 运动方程保持为时间的函数, 这可以通过显式和隐式方法进行求解。其中隐式求解方法的特点:

- 要求矩阵求逆;
- 非线性要求平衡迭代;
- 积分时间步 Δt 可以很大, 但由于收敛问题而受到限制;

除了 Δt 要求很小的情况下, 隐式积分法对大多数问题都是有效的。而显式求解方法不要求矩阵求逆, 其特点:

- 可轻松处理非线性问题(无收敛问题);
- 积分时间步 Δt 必须很小(如 $1e-6$ 秒);
- 对于短时间的瞬态分析有效, 如波的传播, 冲击载荷以及高度非线性问题(如金属成型)。

ANSYS 的显式求解器是 AUTODYN 和 ANSYS-LS/DYNA。

关于模态分析:

模态分析是研究结构动力特性一种近代方法, 是系统辨别方法在工程振动领域中的应用。模态是机械结构的固有振动特性, 每一个模态具有特定的固有频率、阻尼比和模态振型。这些模态参数可以由计算或试验分析取得, 这样一个计算或试验分析过程称为模态分析。这个分析过程如果是由有限元计算的方法取得的, 则称为计算模态分析; 如果通过试验将采集的系统输入与输出信号经过参数识别获得模态参数, 称为试验模态分析。通常, 模态分析都是指试验模态分析。

进行模态分析时, 其与结构受什么样的载荷没有关系, 只要给定了质量、弹性模量、泊松比等材料参数, 并施加了边界约束就可以得到此状态下的各阶自振频率和振型(也称为模态)。

模态分析可以用于:

- 评价现有结构系统的动态特性;
- 在新产品设计中进行结构动态特性的预估和优化设计;
- 诊断及预报结构系统的故障;
- 控制结构的辐射噪声;
- 识别结构系统的载荷;

关于谐响应分析:

谐响应分析是分析结构在不同频率的简谐载荷作用下的动力响应, 是与结构所受载荷相关的, 只是结构所受载荷的都是简谐载荷, 而且载荷频率的变化范围在谐响应分析时要给出来。

关于响应谱分析:

响应谱分析是分析计算当结构受到瞬间载荷作用时产生的最大响应，可以认为这是快速进行接近瞬态分析的一种替代方案。响应谱分析的类型有两种：单点谱分析和多点谱分析。

目前响应谱分析广泛应用于分析核电厂建筑的地震响应，机载电子设备的冲击载荷分析和地震带的商业建筑分析等。在进行响应谱分析之前：

- 先进行模态分析后方可进行响应谱分析；
- 结构必须是线性，具有连续刚度和质量的结构；
- 进行单点谱分析时，结构受一个已知方向和频率的频谱所激励；
- 进行多点谱分析时结构可以被多个（上限20）不同位置的频谱做激励；

关于随机振动分析:

随机振动分析是指在一些随机激励作用下计算一些物理量，如位移或应力等的概率分布情况。目前随机振动分析在机载电子设备，声学装载部件，抖动的光学对准设备等设计上得到广泛应用。

关于瞬态动力学分析:

瞬态动力学分析能够的出任何结构关于时间载荷的响应。不同于刚体动力学分析，在Workbench中瞬态动力学的模型可以是刚体，也可以是变形体，对于变形体，还能考虑材料的非线性特征，最终得到变形体的应力和应变值。

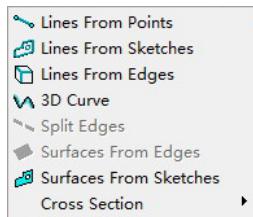
注：虽然ANASSYS Workbench已经推出了很多年，但是很多使用ANASYS 以及Fluent6.3（未被ANASYS收购前）的设计师大部分仍旧使用老版本和传统的操作模式，建议在ANASYS Workbench下处理各项分析，能够获得更好的体验。

概念建模Concept Modeler:

在进行有限元分析之前，需要建立几何模型，而ANSYSWorkbench中已经整合了DesignModeler来处理基本模型问题。虽然DesignModeler同样可以不依靠任何其它软件平台独立完成模型的建立，但是对于大多数的模型其实都不是在ANSYS平台下来完成建模工作的。设计师可以在自己熟悉的CAD系统中建好模型再导入到DesignModeler中即可。DesignModeler能够与大多数主流的CAD软件协同建模，读取外部CAD文件，甚至能够嵌入到主流CAD中。

DesignModeler12能够读入的外部模型格式有：

ACIS (*.sat)
Catia [V5] (*.CATPart;*.CATProduct)
IGES (*.igs;*.iges)
Inventor (*.ipt;*.iam)
Mechanical Desktop (*.dwg)
Monte Carlo N-Particle (*.mcnp)
OneSpace Designer (*.pkg;*.bdl;*.ses;*.sda;*.sdp;*.sdac;*.sdpc)
Parasolid (*.x_t;*.xmt_txt;*.x_b;*.xmt_bin)
Pro/Engineer (*.prt;*.asm*)
Solid Edge (*.par;*.asm;*.psm;*.pwd)
SolidWorks (*.SLDPRT;*.SLDASM)
STEP (*.stp;*.step)
Unigraphics (*.prt)
All Files (*.*)

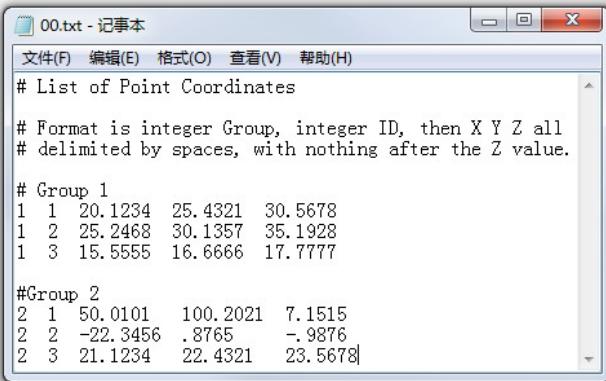


虽然DesignModeler目前可以识别实体和面体，但是无法识别线体，那么从外部导入线用于创建梁模型或者板壳模型就存在一定的障碍。在DesignMoedeler中提供了几种概念模型处理的方法：

这里阐述从外部（RHINO+Grasshopper）获得坐标文件导入到DesignModeler中生成线体的方法。

因为目前没有一个集各专业例如模型构建，结构分析，生态分析，甚至地理信息系统等的所有顶尖技术联合平台，同时可能也因为设计师操作习惯的差异影响这一统一平台的形成，但是这一追求在各专业软件平台发展各自走向成熟的同时，都在努力构建跨平台的联系，使其兼容性得以提升。

DesignModeler支持输入一定格式的点坐标文本文件（.txt），因此首先要清楚文本的格式内容，再使用Python搭建Grasshopper与DesignModeler之间的桥梁—文本。



```
# List of Point Coordinates
# Format is integer Group, integer ID, then X Y Z all
# delimited by spaces, with nothing after the Z value.

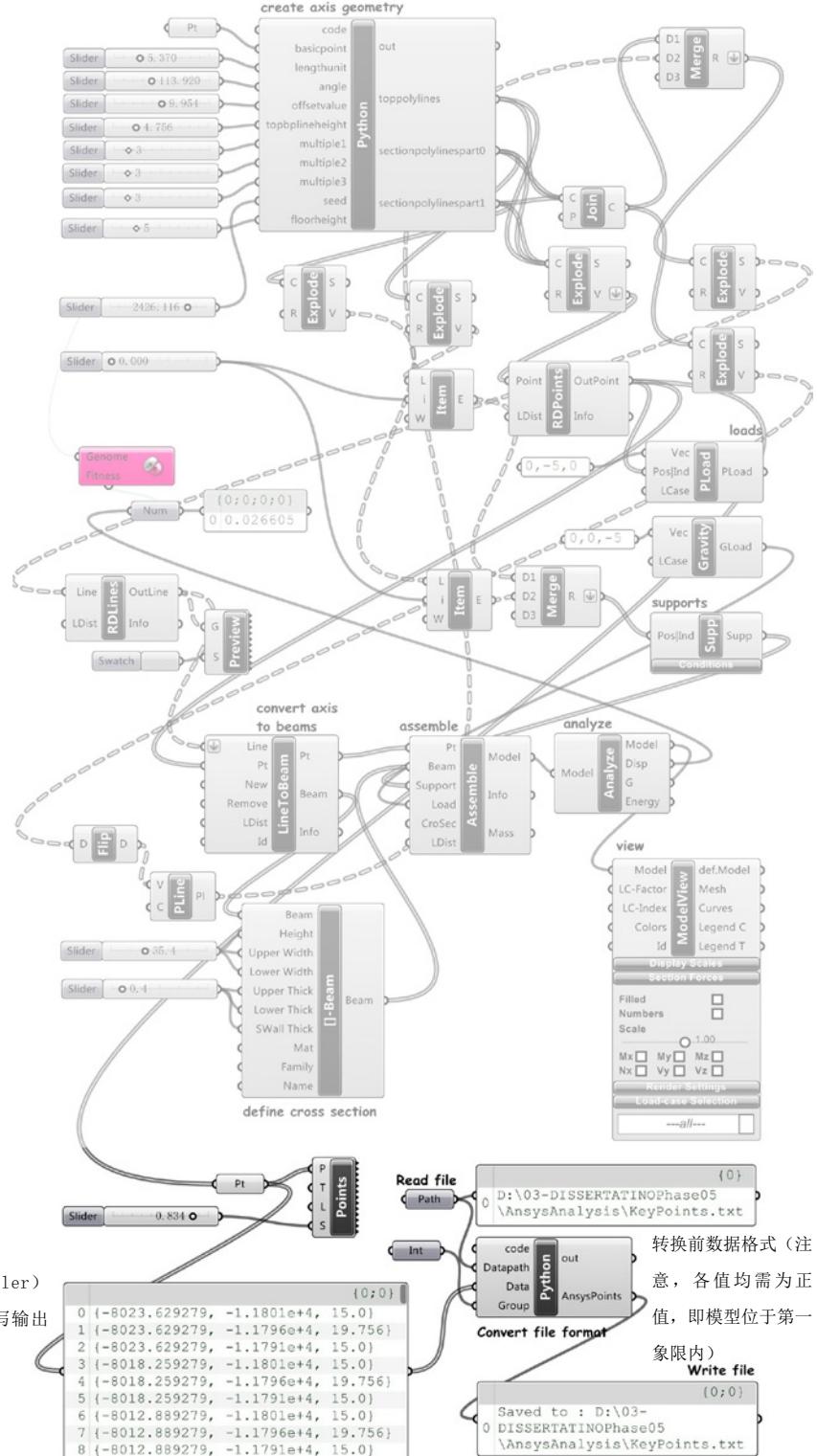
# Group 1
1 1 20.1234 25.4321 30.5678
1 2 25.2468 30.1357 35.1928
1 3 15.5555 16.6666 17.7777

#Group 2
2 1 50.0101 100.2021 7.1515
2 2 -22.3456 .8765 -.9876
2 3 21.1234 22.4321 23.5678|
```

ANSYS帮助文件中例举的文本格式

G-03_32

灰色部分前文已述，见D-7：几何结构优化 (Galapagos+Karamba)



```

import rhinoscriptsyntax as rs
import re #使用re模块（正则表达式）

datapath=Datapath
data=Data#输入数据
group=Group#组的序号

print(group)#通过Print 函数检查数据
f=open(datapath,'w')#打开文件，写入模式
print(data)
newlst=[]
for i in data:
    pat=re.compile("\d+\.\d+")#将正则表达式模式字符串编译为正则表达式对象
    newstr=re.findall(pat,i)#返回string中与pattern匹配的所有为重叠的值
    strlst="%s %d %s %s %s" %(group[0],data.index(i)+1,newstr[0],newstr[1],newstr[2])#字符串格式化
    f.writelines(strlst+'\n')#写入文件
    newlst.append(strlst)
print(newlst)

f.close()
ansyspoints=str('Saved to : '+datapath)
AnsysPoints=ansyspoints

```

关于re模块：

re模块用于在字符串中执行正则表达式匹配和替换。它同时支持Unicode和字节字符串。正则表达式模式是以包含文本和特殊字符序列的字符串形式指定的。由于模式大量使用特殊字符和反斜杠，所以它们通常写为“原始”字符串，如r'(?<int>\d+)\.(\d*)'。



模式语法：

正则表达式模式能够识别一下特殊字符序列：

字符	描述
text	匹配文字字符串text
.	匹配任何字符串，但换行符除外
\$	匹配字符串的开始标志
\$	匹配字符串的结束标志
*	匹配前面表达式的0个或多个副本，匹配尽可能多的副本
+	匹配前面表达式的1个或多个副本，匹配尽可能多的副本
?	匹配前面表达式的0个或多个副本，匹配尽可能少的副本
*?	匹配前面表达式的0个或多个副本，匹配尽可能少的副本
+?	匹配前面表达式的1个或多个副本，匹配尽可能少的副本
??	匹配前面表达式的0个或1个副本，匹配尽可能少的副本
{m}	准确匹配前面表达式的m个副本
{m, n}	匹配前面表达式的m到n个副本，匹配尽可能多的副本。如果省略了n，它将默认设置为0。如果省略了m，它将默认设置为无穷大
{m, n}?	匹配前面表达式的m到n个副本，匹配尽可能少的副本。
[...]	匹配一组字符，如r'[abcdef]'或r'[a-zA-Z]'。特殊字符（如*）在字符串中是无效的
[^...]	匹配集合中未包含的字符，如r'[^0-9]'
A B	匹配A或B，其中A和B都是正则表达式
(...)	匹配圆括号中的正则表达式（圆括号中的内容为一个分组）并保存匹配的子字符串。在匹配时，分组中的内容可以使用所获得的MatchObject对象的group()方法获取
(?aiLmsux)	将字符'a', 'i', 'L', 'm', 's', 'u'和'x'解释为与提供给re.compile()的re, A, re, i, re, L, re, M, re, s, re, u, re, x相对应的标志设置。“a”仅在Python3中可用
(?:...)	匹配圆括号中的正则表达式，但丢弃匹配的子字符串
(?P<name>...)	匹配圆括号中的正则表达式并创建一个指定分组。分组名称必须是有效的Python标识符
(?P=name)	匹配一个早期指定的分组所匹配的文本
(?#...)	一个注释。圆括号中的内容将被忽略
(?=...)	只有在括号中的模式匹配时，才匹配前面的表达式。例如，'hello(?!=world)'只有'world'不匹配时才匹配'hello'
(?!...)	只有在括号中的模式不匹配时，才匹配前面的表达式。例如，'hello(?!=world)'只有'world'不匹配时才匹配'hello'
(?<=...)	如果括号后面的表达式前面的值与括号中的模式匹配，则匹配该表达式，例如，只有当'def'前面是'abc'时，r'(?<abc)def'才会与之匹配
(?<!...)	如果括号后面的表达式前面的值与括号中的模式不匹配，则匹配该表达式，例如，只有当'def'前面是'abc'时，r'(?<abc)def'才会与之匹配
(?i name)ypat npat	检查id或name标识的正则表达式组是否存在。如果存在，则匹配正则表达式ypat。否则，匹配可选的表达式npat。例如r'(Hello)((1)World Howdy)'匹配字符串'Hello World'或'Howdy'

在正则表达式中，标准字符转义序列（如\n和\t）被认为是标准字符，例如r'\n+'可以与一个或多个换行字符匹配。此外，要在正则表达式中指定通常拥有特殊含义的文字符号，可以在它们前面加上反斜杠。例如，r'*'与字符*匹配。此外，还有许多反斜杠序列与特殊的字符集对应。

在Unicode字符串进行匹配时，特殊字符\d,\D,\s,\S,\w和\W的含义不同。在这种情况下，它们将匹配与所描述属性相匹配的所有Unicode字符。例如，\d匹配分类为数字的所有Unicode字符，如欧洲数字，阿拉伯数字和印度数字，它们分别占据了不同的Unicode字符范围。

字符	描述
\number	匹配与前面的组编号匹配的文本。组编号范围为1到99，从左侧开始
\A	仅匹配字符串的开始标志
\b	匹配单词开始或结尾处的空字符串。单词(word)是一个字母数字混合的字符序列，以空格或任何其他非字母数字字符结束
\B	匹配不在单词开始或结尾处的空字符串
\d	匹配任何十进制数。等同于r'[0-9]'
\D	匹配任何非数字字符，等同于r'[^0-9]'
\s	匹配任何空格字符。等同于r'[\t\n\r\f\v]'
\S	匹配任何非空格字符。等同于r'[^ \t\n\r\f\v]'
\w	匹配任何字母数字字符
\W	匹配\w定义的集合中不包含的字符
\z	仅匹配字符串的结束标志
\\\	匹配反斜杠本身

标志	描述
A或ASCII	执行仅8位ASCII字符匹配（仅适用于Python3）
I或IGNORECASE	执行不区分大小写的匹配
L或LOCALE	为\w, \W, \b, \B使用地区设置
M或MULTILINE	将^和\$应用于包括整个字符串的开始和结尾的每一行（在正常情况下，^和\$仅应用于整个字符串的开始和结尾）
S或DOTALL	使点(.)字符匹配所有字符，包括换行符
U或UNICODE	使用\w, \W, \b和\B在Unicode字符属性数据库中的信息（仅限于Python2）。Python3默认使用Unicode
X或VERBOSE	忽略模式字符串中未转义的空格和注释

函数：

以下函数用于执行模式匹配和替换：

- `compile(str[,flags])`

将正则表达式模式字符串编译为正则表达式对象。此对象可以作为模式参数传递给随后的所有函数。flags是以下标志的按位OR结果：

- `escapes(string)`

返回一个字符串，其中的所有非字母数字字符都带有反斜杠；

- `findall(pattern,string,[,flags])`

返回string中与pattern匹配的所有未重叠的值，包括空匹配值。如果模式包含分组，将返回与分组匹配的文本列表。如果使用了不只一个分组，那么列表中的每项都是一个元组，包括每个分组的文本。flags的含义和`compile()`相同；

- `finditer (pattern,string,[,flags])`

与`findall()`含义相同，但返回一个迭代器对象。迭代器返回类型为`MatchObject`的项；

- `match(pattern,string [,flags])`

检查string的开头是否有字符与pattern匹配。如果成功，则返回`MatchObject`；否则返回`None`。flags的含义与`compile()`相同；

- `search(pattern,string [,flags])`

在string中搜索pattern的第一个匹配项。flags的含义与`compile()`相同。如果成功，则返回`MatchObject`；如果未找到匹配值，则返回`None`；

- `split(pattern,string [,maxsplit=0])`

根据pattern出现的位置拆分string。返回字符串列表，其中包括与模式中任何分组匹配的文本。maxsplit是执行拆分的最高次数。默认情况下，将执行所有可能的拆分；

- `sub(pattern,repl,string [,count=0])`

使用替换值repl替换string中最左侧的，未重叠的pattern的出现位置。repl可以使字符串或函数。如果它是一个函数，则使用`MatchObject`调用它，并返回替换字符串。如果repl是一个ie字符串，则使用反向引用（如^\6'）来引用模式中的分组。序列'g<name>'用于引用各顶名称的分组。count是执行替换的最高次数。默认情况下，将替换所有出现的位置。尽管这些函数不接受flags参数，如`compile()`，但可以使用本节前面介绍的（？ iLmsux）实现相同效果。

- `subn(pattern,repl,string [,count=0])`

与`sub()`相同，但返回一个元组，其中包含新字符串和替换次数。

正则表达式对象：

由compile()函数创建的经过编译的正则表达式对象r具有以下方法和属性：

- r.flags

在编译正则表达式时使用flags参数，如果没有指定标志则使用0；

- r.groupindex

一个字典，将r'(?P<id>)’定义为符号分组名称映射到分组编号；

- r.pattern

一个模式字符串，正则表达式从它编译而来；

- r.findall(string [,pos [,endpos]])

等效于findall()函数。pos和endpos指定搜索的开始和结束位置；

- finditer(string [,pos [,endpos]])

等效于finditer()函数。pos和endpos指定搜索的开始和结束位置；

- r.match(string [,pos] [,endpos])

检查在string的开头是否有匹配的字符。pos和endpos指定要搜索的string范围。如果找到匹配值，则返回MatchObject；否则返回None；

- r.search(string [,pos] [,endpos])

在string中搜索匹配项。pos和endpos指定搜索的开始和结束位置。如果找到匹配值，则返回MatchObject；否则返回None；

- r.split (string [,maxsplit=0])

等效于split()函数；

- r.sub(repl,string [,count=0])

等效于sub () 函数

- r.subn(repl,string [,count=0])

等效于subn () 函数；

匹配对象：

search()和match()返回的MatchObject实例包含关于分组内容的信息，以及匹配值得位置数据。MatchObject实例m具有以下方法和属性：

- m.expand(template)

返回一个字符串，该字符串可通过在字符串template上置換正则表达式反斜杠来获取。数字反向引用（如“\1” 和“\2”）和命名引用（如“\g<n>”和“\g<name>”）可替换为对应分组的内容。注意，这些序列应该使用原始字符串指定，或使用原义反斜杠字符（如“r\1”或“\\1”）指定；

- m.group([group1,group2,...])

返回匹配值的一个或多个分组。其中的参数指定分组编号或分组名称。如果未给定分组名称，则将返回整个匹配值；如果仅给定了一个分组，将返回一个字符串，其值包含与该分组匹配的文本；否则返回一个元组，其中包含与所请求的每个分组匹配的文本。如果给定的分组编号或名称无效，将抛出IndexError；

- m.group([default])

返回一个元组，其中包含与模式镇南关所有分组匹配的文本。default是与未包含在匹配值中的分组返回值（默认值为None）；

- `m.groupdict([default])`

返回一个字典，其中包含所有匹配值得所有给定名称的分组。`default`是与包含在匹配值中的分组返回值（默认值为`None`）；

- `m.start([group])`

- `m.end([group])`

这两个方法返回与一个分组匹配的子字符串的开始和结束字符。如果省略`group`，将使用相匹配的整个子字符串。如果该分组存在，但未包含在匹配值中，则返回`None`；

- `m.span([group])`

返回一个二元组(`m.start(group),m.end(group)`)。如`group`未包含在匹配值中，则返回(`None, None`)。如果省略`group`，则使用整个匹配的子字符串；

- `m.pos`

传递给`search()`或`match()`函数的`pos`值；

- `m.endpos`

传递给`search()`或`match()`函数的`endpos`值；

- `m.lastindex`

相匹配的最后一个分组的数字索引。如果没有相匹配的分组，则为`None`；

- `m.lastgroup`

相匹配的最后一个给定名称分组的名称。如果没有相匹配的给定名称分组，或者模式中没有给定名称的分组，则为`None`；

- `m.re`

一个正则表达式对象，它的`match()`或`search()`方法生成次`MatchObject`实例；

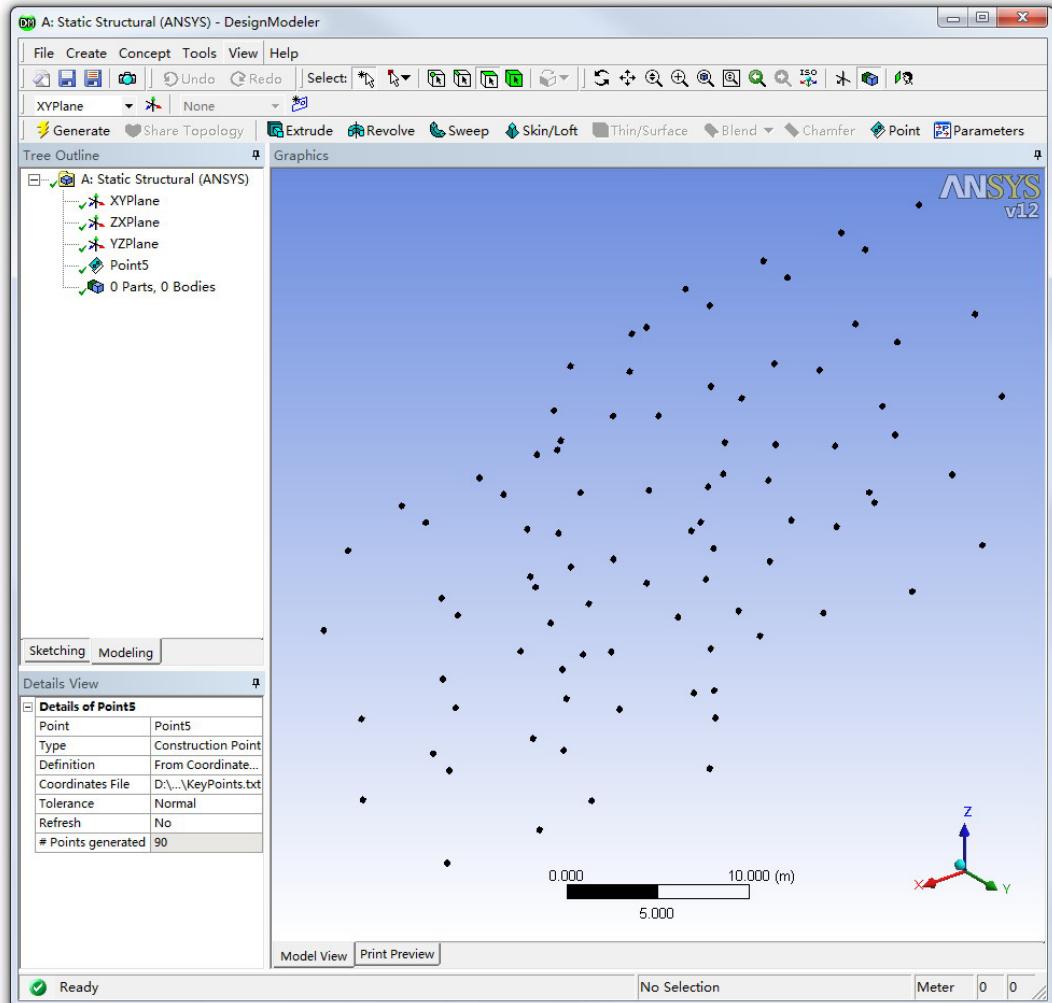
- `m.string`

传递给`match()`或`search()`的字符串；

在Grasshopper平台使用Python编写程序转换的文本格式：

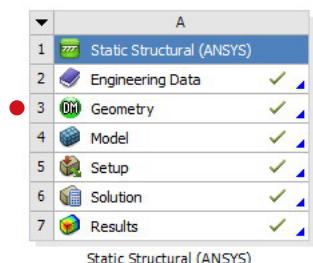
1	1	13.893688	7.459369	15.0
1	2	13.893688	2.482369	19.756
1	3	13.893688	2.494631	15.0
1	4	19.263688	7.459369	15.0
1	5	19.263688	2.482369	19.756
1	6	19.263688	2.494631	15.0
1	7	24.633688	7.459369	15.0
1	8	24.633688	2.482369	19.756
1	9	24.633688	2.494631	15.0
1	10	30.003688	7.459369	15.0
1	11	29.481971	1.153315	19.756
1	12	28.960253	5.152738	15.0
1	13	33.939689	3.806315	15.0
1	14	33.417971	2.499738	19.756
1	15	32.896254	8.805791	15.0
1	16	37.875689	0.153262	15.0
1	17	37.353972	6.152792	19.756
1	18	36.832254	12.458845	15.0

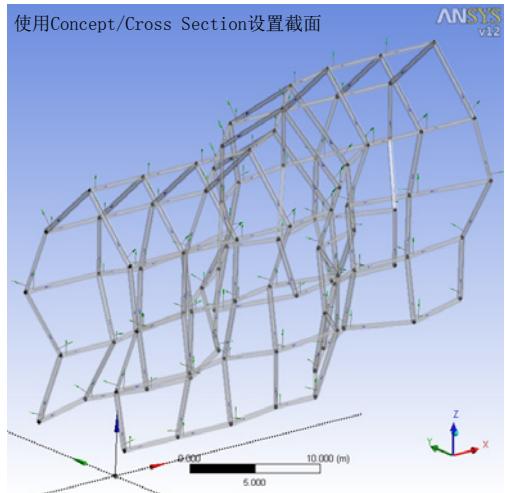
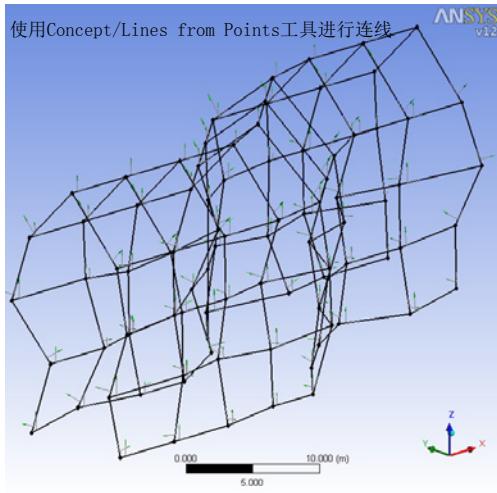
T-KeyPoints



在ANSYS Workbench平台下的DesignModeler中加载点坐标文本文件：

A-StructureAnalysis





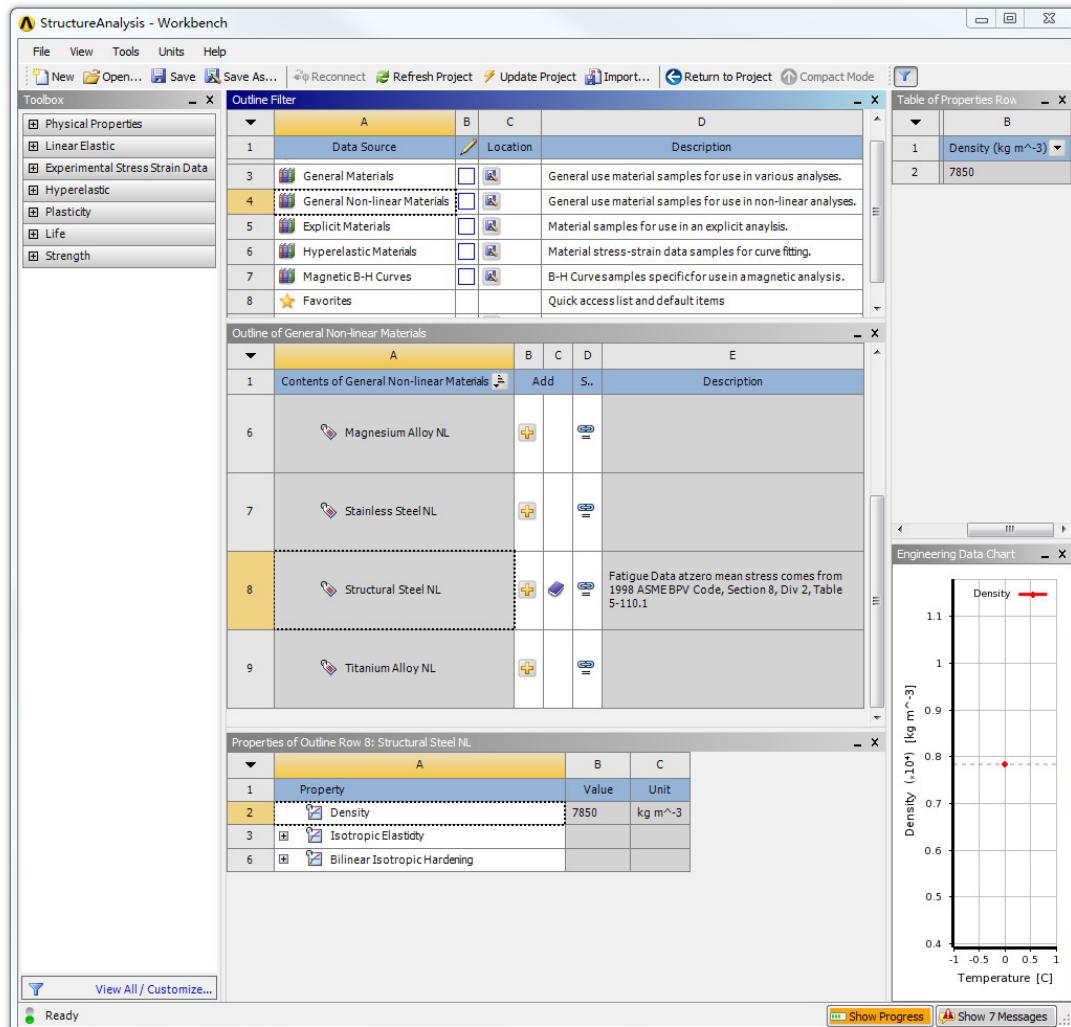
A	
1	Static Structural (ANSYS)
2	Engineering Data ✓
3	Geometry ✓
4	Model ✓
5	Setup ✓
6	Solution ✓
7	Results ✓

Static Structural (ANSYS)

注：各类报告均在Report Preview中可获得。

TABLE 1	
Unit System	Metric (m, kg, N, s, V, A) Degrees rad/s Celsius
Angle	Degrees
Rotational Velocity	rad/s
Temperature	Celsius

TABLE 2 Model (A4) > Geometry	
Object Name	Geometry
State	Fully Defined
Definition	
Source	D:\03-DISSERTATIONPhase05\ANSYS\StructureAnalysis_files\dp0\SYSIDM\SYS.agdb
Type	DesignModeler
Length Unit	Meters
Element Control	Program Controlled
Display Style	Part Color
Bounding Box	
Length X	46.028 m
Length Y	22.913 m
Length Z	19.756 m
Properties	
Volume	80.869 m³
Mass	6.3482e+005 kg
Scale Factor Value	1.
Statistics	
Bodies	1
Active Bodies	1
Nodes	17981
Elements	9026
Mesh Metric	None
Preferences	
Import Solid Bodies	Yes
Import Surface Bodies	Yes
Import Line Bodies	Yes
Parameter Processing	Yes
Personal Parameter Key	DS
CAD Attribute Transfer	Yes
CAD Attribute Prefixes	SDFEA:DDM
Named Selection Processing	No
Material Properties Transfer	No
CAD Associativity	Yes
Import Coordinate Systems	Yes
Reader Save Part File	No
Import Using Instances	Yes
Do Smart Update	No
Attach File Via Temp File	Yes
Temporary Directory	C:\Users\RICHIEDELL\AppData\Local\Temp
Analysis Type	3-D
Mixed Import Resolution	None
Enclosure and Symmetry Processing	Yes



在ANSYS Workbench平台下的Engineering Data中选择或编辑材料属性：

Structural Steel > Constants

Density	7850 kg m^-3
Coefficient of Thermal Expansion	1.2e-005 C^-1
Specific Heat	434 J kg^-1 C^-1
Thermal Conductivity	60.5 W m^-1 C^-1
Resistivity	1.7e-007 ohm m

Structural Steel > Compressive Yield Strength

Compressive Yield Strength Pa	2.5e+008
-------------------------------	----------

Structural Steel > Tensile Yield Strength

Tensile Yield Strength Pa	2.5e+008
---------------------------	----------

Structural Steel > Tensile Ultimate Strength

Tensile Ultimate Strength Pa	4.6e+008
------------------------------	----------

Structural Steel > Isotropic Elasticity

Temperature C	Young's Modulus Pa	Poisson's Ratio
	2.e+011	0.3

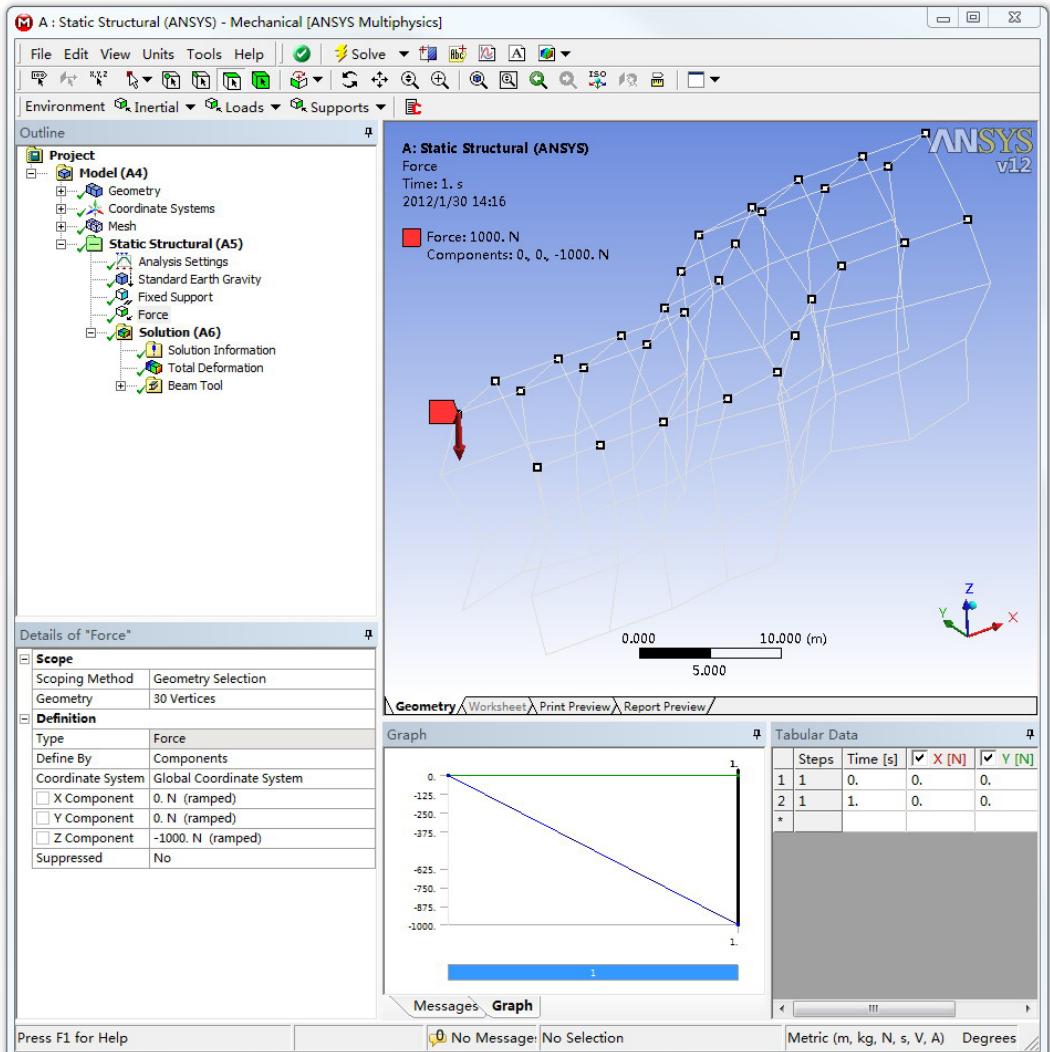
Structural Steel > Strain-Life Parameters

Strength Coefficient Pa	Strength Exponent	Ductility Coefficient	Ductility Exponent	Cyclic Strength Coefficient Pa	Cyclic Strain Hardening Exponent
9.2e+008	-0.106	0.213	-0.47	1.e+009	0.2

Toolbox:

- 1 Static Structural (ANSYS)
- 2 Engineering Data
- 3 Geometry
- 4 Model
- 5 Setup
- 6 Solution
- 7 Results

Static Structural (ANSYS)



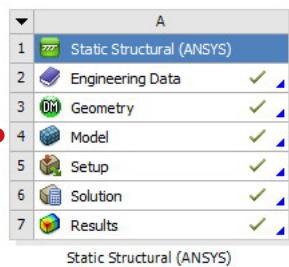
在ANSYS Workbench平台下的Mechanical中赋予材料，设置约束载荷以及划分网格：

Model (A4) > Analysis	
Object Name	Static Structural (A5)
State	Solved
Definition	
Physics Type	Structural
Analysis Type	Static Structural
Solver Target	ANSYS Mechanical
Options	
Environment Temperature	22 °C
Generate Input Only	No

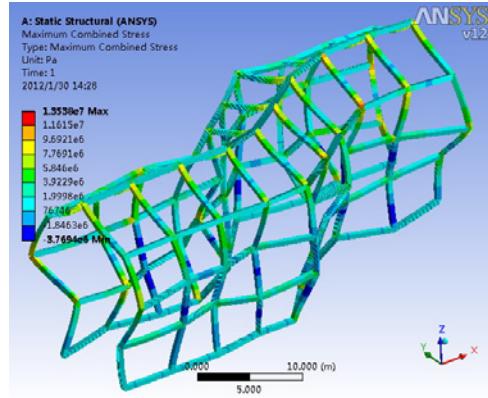
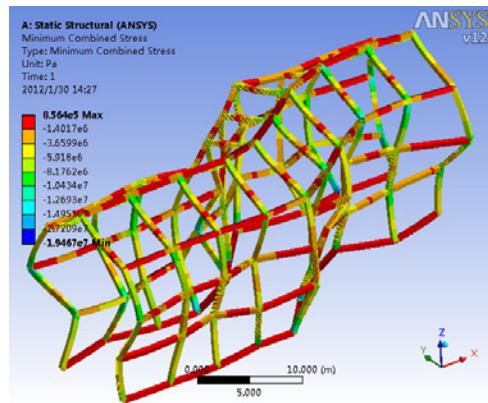
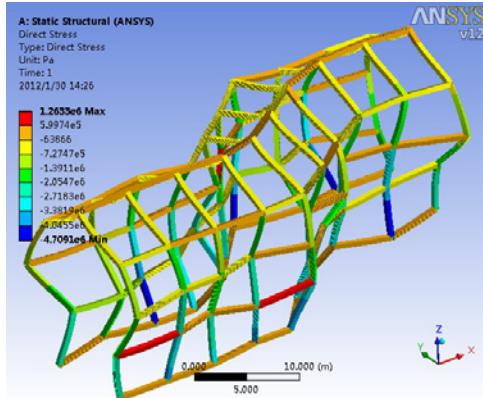
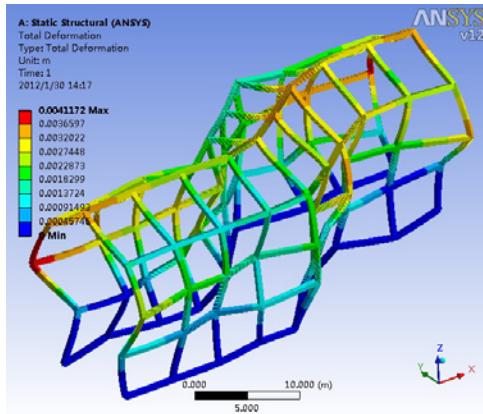
Model (A4) > Static Structural (A5) > Loads		
Object Name	Fixed Support	Force
State	Fully Defined	
Scope		
Scoping Method	Geometry Selection	
Geometry	20 Vertices	30 Vertices
Definition		
Type	Fixed Support	Force
Suppressed	No	
Define By	Components	
Coordinate System	Global Coordinate System	
X Component	0. N (ramped)	
Y Component	0. N (ramped)	
Z Component	-1000. N (ramped)	

Model (A4) > Static Structural (A5) > Accelerations

Object Name	<i>Standard Earth Gravity</i>
State	Fully Defined
Scope	
Geometry	All Bodies
Definition	
Coordinate System	Global Coordinate System
X Component	-0. m/s ² (ramped)
Y Component	-0. m/s ² (ramped)
Z Component	-9.8066 m/s ² (ramped)
Suppressed	No
Direction	-Z Direction



计算结果后处理显示：



Model (A4) > Static Structural (A5) > Solution

Object Name	Solution (A6)
State	Solved
Adaptive Mesh Refinement	
Max Refinement Loops	1.
Refinement Depth	2.

TABLE 12

Model (A4) > Static Structural (A5) > Solution (A6) > Solution Information

Object Name	Solution Information
State	Solved
Solution Information	
Solution Output	Solver Output
Newton-Raphson Residuals	0
Update Interval	2.5 s
Display Points	All

TABLE 13

Model (A4) > Static Structural (A5) > Solution (A6) > Results

Object Name	Total Deformation
State	Solved
Scope	
Scoping Method	Geometry Selection
Geometry	All Bodies
Definition	
Type	Total Deformation
By	Time
Display Time	Last
Calculate Time History	Yes
Identifier	
Results	
Minimum	0. m
Maximum	4.1172e-003 m
Information	
Time	1. s
Load Step	1
Substep	1
Iteration Number	1

模态分析：



Mode	Frequency [Hz]
1.	1.9398
2.	2.4461
3.	3.4197
4.	4.1622
5.	4.8743
6.	5.4672

TABLE 14
Model (A4) > Static Structural (A5) > Solution (A6) > Beam Tool

Object Name	Beam Tool
State	Solved
Scope	
Geometry	All Line Bodies



TABLE 15

Model (A4) > Static Structural (A5) > Solution (A6) > Beam Tool > Results

Object Name	Direct Stress	Minimum Combined Stress	Maximum Combined Stress
State	Solved		
Definition			
Type	Direct Stress	Minimum Combined Stress	Maximum Combined Stress
By	Time		
Display Time	Last		
Calculate Time History	Yes		
Use Average	Yes		
Identifier			
Results			
Minimum	-4.7091e+006 Pa	-1.9467e+007 Pa	-3.7694e+006 Pa
Maximum	1.2633e+006 Pa	8.564e+005 Pa	1.3538e+007 Pa
Information			
Time	1. s		
Load Step	1		
Substep	1		
Iteration Number	1		

注：各类报告均在Report Preview中可获得。

5 扩展组件

Grasshopper仍旧以强劲的趋势不断的拓展其应用领域，这里仅将其官方网站目前涉及的的扩展组件罗列出来，是读者能够对其发展的方向有个清楚的认知，如果对哪个方向有兴趣，希望进一步了解该扩展组件可以在其官方网站查询。

1. Kangaroo



官方网址 <http://www.grasshopper3d.com/group/kangaroo>

Kangaroo is a Live Physics engine for interactive simulation, optimization and form-finding directly within Grasshopper;

Kangaroo动力学引擎组件可以与Grasshopper结合用于交互式模拟、优化、寻找符合力学的某种形式；

2. Hummingbird



官方网址 <http://ghhummingbird.wordpress.com/>

Hummingbird is a set of Grasshopper components that facilitate the creation of Revit native geometry. This process utilizes a simplified pseudo-language in Excel to describe many properties of the Revit BIM geometry. Translating Rhino primitive geometry to Revit has been vastly simplified without the need for linking or reference objects. This allows for the downstream Revit model to be modified and adapted for the project duration;

Hummingbird组件可以用于方便的创建Revit本地几何体。在Excel中使用简单的pseudo语言可以描述Revit BIM建筑信息模型的几何属性。在没有参考对象的条件下可以将Rhino的几何体转化为Revit几何对象，大幅度简化转化的过程，并且允许导入到Revit中的几何体进行修改适应项目的有效管理；

3. Geometry Gym BIM



官方网址 <http://ssi.wikidot.com/>

BIM (Building Information Modeling) enhancements for Rhino and Grasshopper, import/export to Revit, Digital Project, Bentley and Archicad amongst others. Also Structure modeling (including analytical) and Computational Geometry Tools such as mesh relaxation/inflation and tessellation tools;

Geometry Gym BIM可以提升Rhino和Grasshopper的BIM建筑信息模型能力，可以将几何模型导入导出到Revit、Digital Project、Bentley和Archicad等平台，同时可以构建结构模型、分析模型以及提供了部分计算几何体工具，例如Mesh面松弛和膨胀工具以及曲面离散技术等；

4. Starling



官方网址 <http://www.food4rhino.com/project/starling>

SL_0.1

Starling_0.1 mesh tools. These components enable mesh parametrization, so it behaves like a surface - you can evaluate points at any place etc. Is it accurate? - not at all, but give it just a minute and you'll probably fall in love. Mesh morphing, remeshing, smooth curves evaluation (see group image), 3d convex hull and last but not least - slFastMesh - which will enable you to stop thinking about meshing parameters and start playing.

What's that all about? Mesh is embedded to planar form (or spherical), with respect to topology. You draw point on this planar (or spherical) mesh representation and it automatically finds its place on original mesh.

Starling组件是Mesh编辑工具，能够将Mesh参数化，使其表现行为类似于曲面，例如你可以在其表面提取点，同时可以使用Mesh变形工具、重组Mesh面、三维凸包等工具协助Mesh面的编辑，还可以找到Mesh面上点在原始面上的位置等；

官方网址 http://www.hilojacobs.com/?page_id=464

5. Chameleon



Chameleon is a plugin for both Grasshopper and Revit with a focus on interoperability, simulation, and efficient practice workflows. Chameleon's main advantage is its ability to facilitate easy transfer of geometric data between Grasshopper and Autodesk Revit, but also includes other valuable tools to make life easier, both in Grasshopper and Revit;

Chameleon是基于Grasshopper和Revit交互的扩展组件，能够有效地转换二者之间的几何数据，同时提供了一些有用的组件工具；

6. DIVA for Rhino



官方网址 <http://www.diva4rhino.com>

DIVA-for-Rhino is a plugin which runs thermal, daylight, solar radiation, and glare simulations. Our goal is to bring validated environmental simulations directly to the conceptual design environments of Rhino and Grasshopper;

DIVA for Rhino组件可以在Grasshopper平台上用于热环境、光环境、太阳辐射以及眩光的模拟，能够有效、更直接的协同设计几何与环境的关系，对于概念设计阶段很有帮助；

7. Firefly Plug-in

官方网址 <https://www.box.com/firefly>

Firefly offers a set of comprehensive software tools dedicated to bridging the gap between Grasshopper (a free plug-in for Rhino) and the Arduino micro-controller. It allows near real-time data flow between the digital and physical worlds – enabling the possibility to explore virtual and physical prototypes with unprecedented fluidity.



As a generative modeling tool, Grasshopper offers a fluid visual interface for creating sophisticated parametric models, but by default, it lacks the ability to communicate with hardware devices such as programmable microcontrollers or haptic interfaces. Firefly fills this void. It is an extension to the Grasshopper's parametric interface; combining a specialized set of components with a novel communication protocol (called the Firefly Firmata or Firmware) which together enable real-time communication between hardware devices and the parametric plug-in for Rhino.

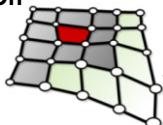
A notable distinction for Firefly is that it is the first visual microcontroller programming environment designed specifically for a 3-D parametric CAD package (such as Rhino). This feature means that real-world data, acquired from various types of sensors or other input devices (video cameras, internet feeds, or mobile phone devices, etc.) can be used to explicitly define parametric relationships within a Grasshopper model. Firefly completes the communication feedback loop by allowing users the ability to send information from Grasshopper back to the microcontroller in order to incite specific actuations (ie. lights, motors, valves, etc). Ultimately, this workflow creates a new way to create interactive prototypes.

Firefly提供了一套全面的工具致力于构建Grasshopper与Arduino微控制器之间的联系，能够获得数字世界与真实的物理世界之间几近实时的数据流的传输，探索虚拟世界与实体原形之间前所未有的互动联系。

作为模型构建工具，Grasshopper提供了建立复杂参数模型的可视化节点操作模式，但是默认条件下，它缺少对可编程微控制器或触觉接口等硬件设施的交互能力。Firefly弥补了这一不足，它是Grasshopper的扩展插件，其专门设置了一种新颖的通信协议帮助在硬件设施与参数模型之间实时的通信。

Firefly的显著特点在于它是专门设计的三维参数化CAD软件包（例如Rhino）首个可视化单片机编程环境。这意味着Grasshopper可以使用现实世界的数据，来自于各种类型的传感器或者其它输入设备例如摄像机、网络传送、移动电话等设备的数据来构建之间的参数化模型，能够通过Grasshopper发送信息到微控制器以完成特定的动作，例如控制灯的开启、电机以及电动阀相关的动作等来完成建立信息交互反馈回路，Firefly组件以一种新的方式创建交互原形；

8. Paneling Tools GH Add-On



官方网址<http://www.grasshopper3d.com/group/panelingtools>

Parametric tools to create and manipulate rectangular grids, attractors and support creative morphing of parametric patterns.

Paneling Tools工具有基于Rhino平台的版本和基于Grasshopper的版本，一般用于模型的表皮构建，能够建立和调整矩形栅格网，支持使用参数模式建立曲面变体等；

9. LunchBox



官方网址<http://nmillerarch.blogspot.com/>

A collection of useful geometry tools. The plug-in includes utilities for mathematical forms, paneling systems, structures, and workflow;

LunchBox工具提供了一组有使用价值的组件，包括数学形式、嵌镶系统、结构和工作流；

10. gHowl官方网址<http://www.grasshopper3d.com/group/ghowl>

gHowl is a set of components which extend Grasshopper's ability to communicate and exchange information with other applications and physical devices;

gHowl包含一系列的组件用于扩展Grasshopper与其它信息平台以及设备之间的信息交互；

11. GENERATION官方网址<http://antonioturiello.blogspot.com/>

Provides additional components to explore, animate and fabricate generative shapes with Grasshopper;
GENERATINO提供了一组额外的组件可以用于探索、激发和构建有生成能力的形式；

12. Centipede官方网址<http://www.ceedstudio.com/ceed3/?p=482>

Allows the user to keyframe the animation of multiple objects in Grasshopper along a single slider;
Centipede允许在Grasshopper中沿单一滑块对多个对象的动画建立关键帧；

13. Rhino.Python官方网址<http://www.food4rhino.com/project/ghpython>

Rhino.Python是Grasshopper平台上使用Python语言协助模型构建的编程语言组件；

14. [uto]tools Plug-ins官方网址<http://www.utos.blogspot.com>**Geco**

Geco is a set of components which establish a live link between Rhino/Grasshopper and Autodesk Ecotect to export, evaluate and import...

**MeshEdit**

MeshEdit is a set of components which extend Grasshopper's ability to manipulate meshes;

**FlowLines**

FlowLines is a set of components to create 2D or 3D-Streamlines and 2D-Equipotential lines;

FlowLines FlowLines可以创建一些有花纹流线的图案；

15. Mantis官方网址<http://zaghloul4d.blogspot.com/>

Mantis{Grasshopper,Mathematica} A new tool for grasshopper named Mantis which allows to interact with Mathematica directly, the first 3 tools are for generative Cellular Automata "CA";

Mantis工具允许Grasshopper与Mathematica之间直接交换数据,其前三个工具为元胞自动机"CA";

16. Goat



官方网址<http://www.rechenraum.com/goat>

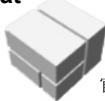
goat is an optimization solver add-on component. It perfectly complements galapagos, David Rutten's evolutionary solver based on a randomized core. goat pursues a mathematical rigorous approach and relies on gradient-free optimization algorithms, delivering fast and deterministic results. At every run, goat will yield the same optimal result.

goat is a drop-in replacement for galapagos. It is based on David Rutten's galapagos GUI and interfaces NLOpt, a collection of mathematical optimization libraries;

Goat是一个优化的求解组件，能够完美的弥补galapagos进化组件的不足，David Rutten的进化求解基于一个随机的内核。Goat则诉求于精准与严格的方法，依靠梯度变化的最佳化算法，实现更快和更确定性的结果。同时每次运行都会趋向于同样的优化结果。

Goat可以作为Galapagos的替代，它也是基于David Rutten的galapagos的GUI用户界面；

17. Packrat



官方网址<http://volatileprototypes.com/projects/packrat/>

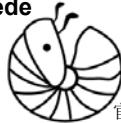
PackRat is a component for Grasshopper that facilitates the arrangement of rectangular items within rectangular containers with the aim of improving efficient use of space. In other words, it attempts to approximate optimal solutions to the 3D bin-packing problem.

Packrat accepts a list of containers and a list of items to be packed. Only rectangular items are recognized, so if the input consists of arbitrarily shaped BRepS, their bounding boxes will be considered. It outputs a GH_Structure of items per container, and a supplementary list of items that did not fit in any container.

There is an option to choose whether sorting of elements and containers should be performed before packing. While sorting usually yields better results, disabling allows the user to perform their own sorting before inputting the elements to PackRat；

Packrat组件试图以三维装箱问题的近似最优解将输入的多个盒体有效率的放置于矩形容器里；Packrat目前只接受矩形对象，对于Brep等对象将转化为其边界盒体来求解；在Packrat输入端S项，可以确定是否对对象排序，如果排序将会获得较好的结果；

18. Millipede



官方网址<http://www.sawapan.eu>

Millipede is a structural analysis and optimization component for grasshopper. It allows for very fast linear elastic analysis of frame and shell elements in 3d, 2d plate elements for in plane forces, and 3d volumetric elements. All systems can be optimized using built in topology optimization methods and have their results extracted and visualized in a variety of ways.

In addition millipede implements a few basic geometric features [extraction of iso surface meshes from volumetric scalar fields or and extraction of curved contours over any mesh] and a few numerical analysis tools [Fast fourier transforms in 1 and 2 dimensions, sparse linear system solver and eigenvalue calculation for large matrices using functionality from the intel math kernel library];

Millipede是结构分析和优化的组件，可以非常快的进行框架以及二维和三维的外壳元素的线弹性分析。所有系统都能够被内建的拓扑优化方法进行优化，并且提取结果并可视化。

另外Millipede提供了几个基本的几何特征工具，例如从体积标量场提取ISO表面网格或者任何网格的曲线轮廓和几个数值分析工具，例如一维和二维的快速傅立叶变换，稀疏线性系统求解和大型矩阵特征值的计算；

19. Scarab



官方网址<http://www.food4rhino.com/project/scarab>

Scarab is a set of components that exposes Maxwell Render materials to Grasshopper. Materials can be assigned and baked to Grasshopper geometry;

Scarab是材质赋予工具，可以将材质赋予和烘培到Grasshopper对象上；

20. Quokka

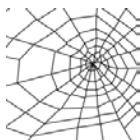


官方网址<http://quokkalab.com/>

Quokka is a software tool for connecting Microsoft Kinect with Grasshopper in Rhino. Quokka gives the ability to use the Kinect as a 3D scanner. This will help architects, designers, engineers, students and anyone who is interested in working in Rhino to parametrically manipulate physical data;

Quokka可以连接Microsoft Kinect微软体感游戏设备到Grasshopper参数平台，将其作为一个三维的扫描器，这能够帮助对参数感兴趣的建筑师、设计师、工程师以及学生通过参数控制物理数据；

21. SpiderWeb



官方网址<http://www.gbl.tuwien.ac.at/Archiv/digital.html>

SpiderWeb is a plugin providing simple graph tools in Grasshopper;

SpiderWeb是能够构建简单图形的工具；

22. Horster Tools



官方网址<http://www.grasshopper3d.com/group/horster-tools>

horsterReference tools make it easy to reference objects by type, layer and other attributes from Rhino to Grasshopper.

horster cameraControl allows user to control camera in Rhino viewport via Grasshopper.

HorsterReference Tools可以根据Rhino空间中对象的类型、所在层或其它属性将对象调入到Grasshopper空间中；另外可以使用cameraControl组件在Grasshopper空间中控制Rhino空间的视图；

23.Karamba



官方网址<http://www.karamba3d.com>

Karamba interactively calculates the statical response of three dimensional beam structures under the action of external loads;

Karamba是基于Grasshopper的有限元Finite Element结构分析，主要是静力结构分析，能够计算在外力作用下梁结构三个方向上的受力情况，一般会结合Galapagos组件优化结构；

24.Hoopsnake



官方网址<http://www.volatileprototypes.com/projects/hoopsnake/>

Hoopsnake is a component that allows feedback loops within Grasshopper;

HoopSnake, apart from a legendary creature, is simple component for Grasshopper. What it does in principle is to create a copy of the data it receives at its input upon user request and store it locally. This duplicate is made available through a standard Grasshopper parameter output.

What turns this operation from mundane to actually promising is the fact that the input of the component includes some custom programming to escape Grasshopper's recursive loop avoidance check.

What this means in practice is that it is possible through the use of HoopSnake to send the output of a set of components back to its input in a looping fashion. With the help of a little bit of Data Tree / Data List manipulation, this should open up some new possibilities for iterative process modeling；

Hoopsnake在Grasshopper可以反馈环路，是用于迭代的工具，而且相对于使用Python语言编程，其操作容易，将初始值输入，Hoopsnake会存储该值，经过自定义的一连串函数变化后，再将变化后的值连回Hoopsnake组件，构成回路不断往复计算，直到某个值不满足要求时，获得布尔值为False终止计算。通过使用Hoopsnake组件，可以帮忙处理树形数据和单一列表数据，获得不断变化的值，打开迭代处理数据获得模型构建新的可能性；

25.Slingshot



官方网址<http://the provingground.wikidot.com/slingshot>

Slingshot! is a plug-in aimed at enabling Grasshopper to connect and share information with remote data sources；

Slingshot! currently features an extensive collection of components for reading, writing, and managing relational database systems using SQL；

Slingshot能够建立Grasshopper与远程数据源的链接和信息共享，工具组件能够使用SQL读取、写入和管理相关数据系统；

26.SPM Vector Components



官方网址<http://www.food4rhino.com/project/spm>

SPM Vector Components is a collection of tools for integrating and manipulating vector fields;

SPM Vector Components组件可以用于管理矢量场;

27.Finches:Local



The Finches components allow for easy import, export, batch processing, and (coming soon) importing GIS data into Rhino and Grasshopper;

Finches组件可以帮助导入、导出、批处理和快速的即时导入GIS数据到Grasshopper平台;

28.Elk



官方网址 <http://www.food4rhino.com/project/elk>

Elk is a plugin used to generate topographies and street maps using data from OpenStreetMap.org and Shuttle Radar Topography Mission (SRTM) data from NASA/Jet Propulsion Laboratory;

Elk工具可以从OpenStreetMap.org和NASA/Jet Propulsion实验室的Shuttle Radar Topography Mission(SRTM)数据建立地形和街道图;

29.HAL



官方网址 <http://www.thibaultschwartz.com/>

HAL is a Grasshopper plugin for ABB industrial robots programming. With its large library (78 components), the user can simulate complex multi-robots installations and generate entire RAPID programs in real-time. With its special programming packs covering a large panel of RAPID instructions, HAL allows the user to create advanced programming structures including I/O management and error handling. New tooling packs, for hotwire cutting and milling, ease the programming process of innovative fabrication strategies;

HAL是ABB工业机器人程序在Grasshopper平台控制的扩展插件,带有78个组件的庞大程序库, 可以模拟复杂的多数机器人设备和实时生成整个PAPID程序。使用其特殊的编程包可以转化PAPID大型面板的指令, HAL允许用户创建包含I/O管理和错误处理的高级编程结构, 新的工具包包含热线式切割、铣切、新颖的加工程序处理等;

30.Genoform



官方网址 <http://www.genoform.com>

This version includes grasshopper components that are able to communicate with the external genoform (main). You can create a generative designs by simply replacing your slider bars with genoform slider bars and by dragging in the genoform (grasshopper) component that establishes a communication link with the genoform (main);

Genoform工具可以交互外部的Genoform程序，Genoform是一个老牌的衍生设计外挂程序，早期挂在Solidwork下面，利用自动输入参数化方式来大量产生设计方案，现在则可以实现与Grasshopper的互动，通过使用genoform滑块替代Grasshopper的数值滑块产生设计；

31. Minimal Surface Plugin

Creates minimal surfaces from either 4 boundary curves or 2 curves (open or closed)

使用闭合或者不闭合的四根或两根边界曲线建立最小曲面；

第四部分 进化计算与迭代

1 Galapagos

Evolutionary Computing (EC) 进化计算



Evolutionary Computing (EC) 进化计算已经在各领域中广泛应用，但是它主要被程序员层级的人员所使用。在设计中如何使用EC的逻辑，构建一个EC平台让非程序员来处理更加广泛的问题，是Galapagos作者David Rutten的目的。

在Karamba一节处理结构优化问题时，就是以Galapagos为解算核心，获得合理的结构体系。其主要解算的就是那些可以以数值作为影响几何模型构建的参数与一定条件下所获得的结果（该结果应该具有最大值，或最小值的需求目标）之间的联动关系，以其期望结果来获得模型构建的初始条件。

David Rutten在解释Galapagos时（2010年9月21日维也纳AAG10会议）并不是对其本身具体程序语言的解释，毕竟使用Galapagos的对象不再是纯粹的程序员，而是其使用者-设计师，因此在作出解释时，如同前文所编写Python代码时所使用的几何构建逻辑，以更直观的方式解释程序运作的过程。

Evolutionary Principles Applied to Problem Solving

用于问题处理的进化计算原则

目前对于Evolutionary Solvers进化解算或者Genetic Algorithms遗传算法并没有什么特别的。它最开始来自于上个世纪60年代，Lawrence J.Fogel里程碑式的论文“On Organization of Intellect”，是首次对进化计算的尝试。在70年代初，Ingo Rechenberg 和 John Henry Holland通过种子生长机制进一步印证进化算法。然而该算法并没有得到广泛的关注，直到计算机程序的发展，1986年Richard Dawkin“*The Blind Watchmaker*”一书的问世，主要为一个基于人类基因选择，无止境生长的生物形态“Bio-morphs”。80年代个人计算机的发展也使进化算法被用于个人项目，使其得到更广泛的应用。

Pros and Cons

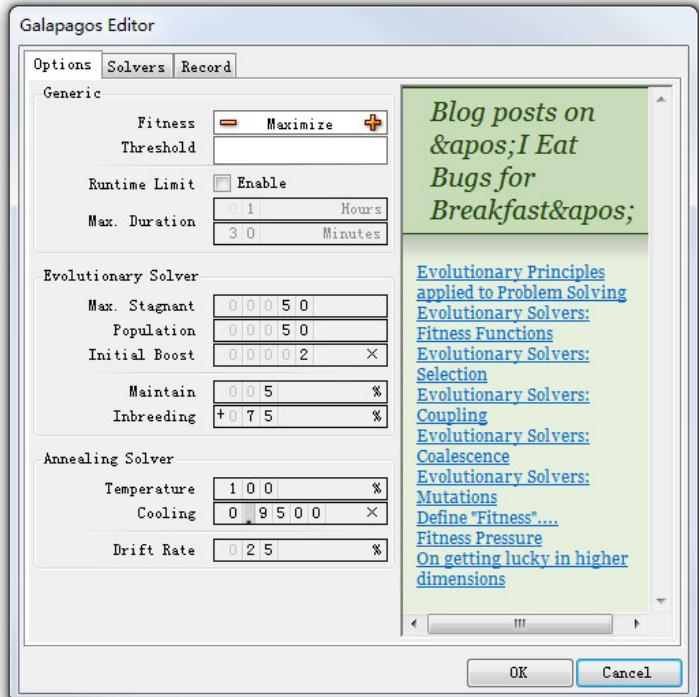
利弊

在开始解释Galapagos之前有必要说明下该解算器的利弊。对于所有问题的解决策略似乎很难找到非常完美的答案，每种方法都有其缺憾，正如进化算法被广泛的获知其解算的速度不尽人意。甚至一个单独的处理可能需要花费几天甚至星期的时间，尤其在解决复杂的问题时。

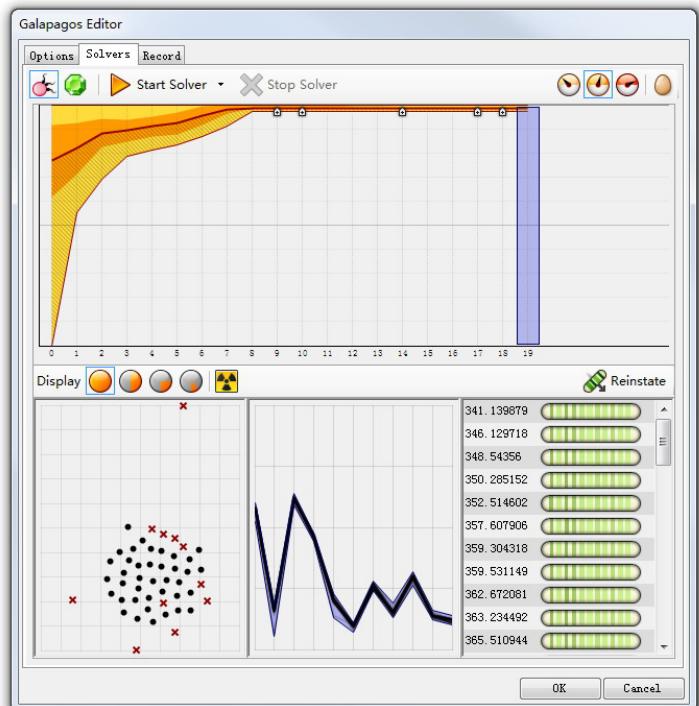
进化算法同时也并不能保证是否能解算问题，除非预先确定一个足够好的值。否则，可能程序永远运行下去，永远找不到答案，又或即使已经达到目的，但是却无法正确识别。

但是，相对于所有计算方法，EC却有很强的优势，甚至有些独特。它具有很好的变通性，可以解决很广泛的问题。而且它具有很好的宽容性。使用EC的方式进行解算，每一步迭代都是正向的，即后一个解算答案总比前一个有所改进，因此在解释的过程中，就可以选择优化于最开始的设置，甚至当在计算过程中失败时，该值也是一个可以考虑的答案。

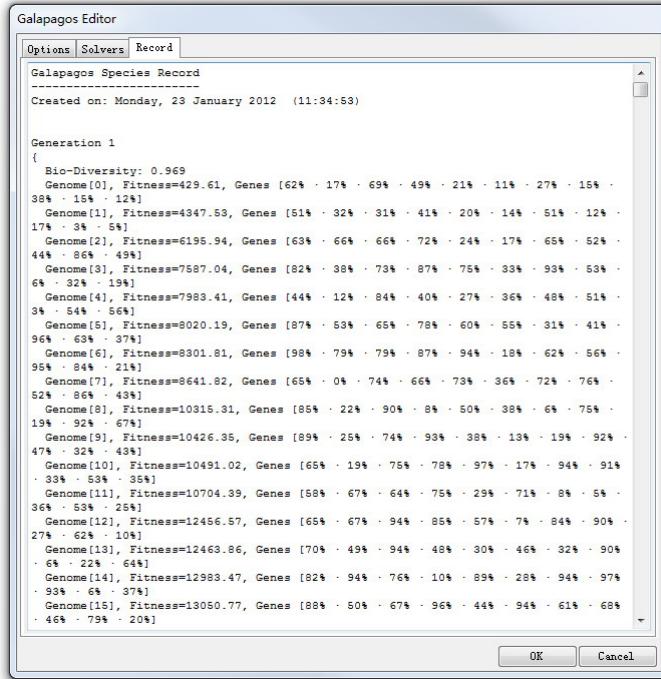
EC也具有很好的交互性。尤其在解算具有一定广度的问题时。它的运行过程是高度透明的，因此存在很多机会创建人机交互的对话窗口，引导解算跨过障碍，或者探索一些并不是很理想的答案，甚至一条死胡同。



Galapagos的对话窗口



Galapagos的解算窗口

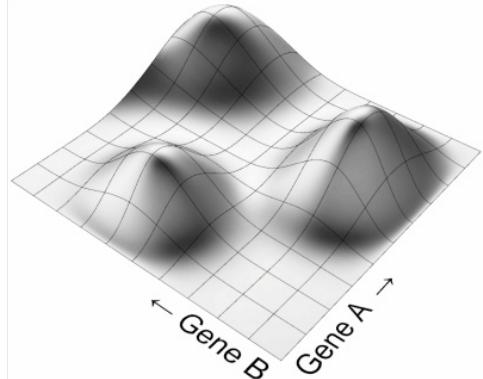


Galapagos的记录窗口

The Process

进化计算过程

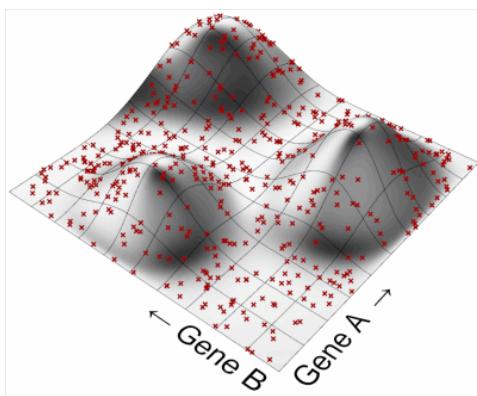
David Ruttent以图示的方式来解释进化计算的过程。每张图片都显示了进化过程某一时刻的状态。

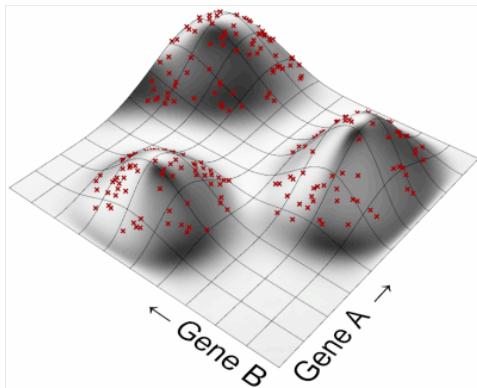


可以把图中的鼓起物看作山脉，并存在两个变量GeneA与GeneB，实际代表Galapagos的Genome基因输入端。当变量GeneA或者GeneB独自或同时发生变化时，模型中代表求解目标的Fitness，即Galapagos的Fitness适合输入端将会发生改变（此时的图示还没有产生标志点，见后图）。此处所寻找的Fitness适合是找到山峰顶点。

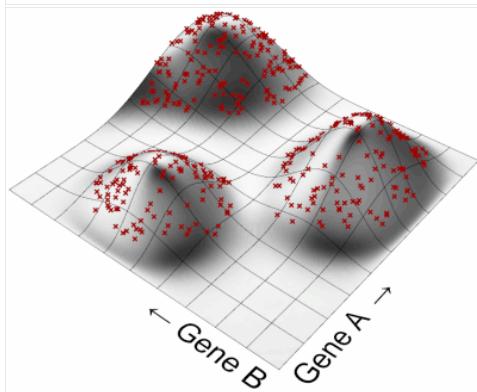
当然输入条件并不局限于本例中的两个，X, Y方向的变量，以及Z方向的Fitness适合求解，构成的3维体系，还可以更多，12个输入变量，13维体系都是可能的，求解问题也会更加复杂。

在开始计算时，必须给些初始的条件然后才可以进一步判断进行优化选择，这里初始条件是随机产生的红点，可以理解为基因组（Genomes），此时每个基因除了具有其位置坐标值（x—GeneA, y—GeneB, z—Fitness）外，并没有其它的什么。每个随机获得的基因都是下一步计算所需要的。

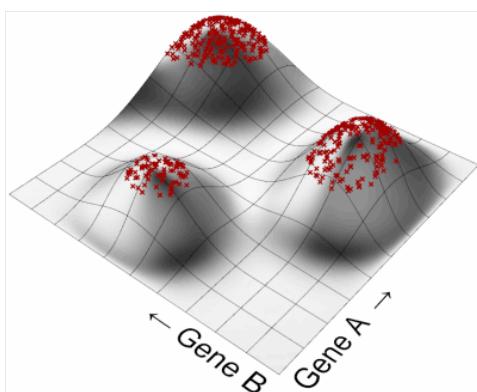




一旦开始下一步计算，因为解算的目的是获得最高点，因此只要找到Z值，即Fitness最大的值即可，每一个高处的基因都比低处的基因更接近答案，所以可以把不适合低处的基因排除掉。



如果一直排除掉不好的基因，所剩下的基因不能被确定是最好的，因为一开始的基因都是随机产生。因此，在排除不好基因的同时，需要在优良的区域范围内继续繁育新的基因组。

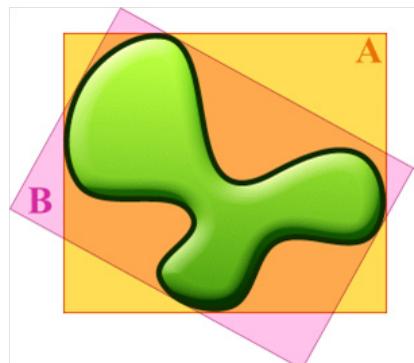


现在有了新的基因组，所获得的基因也不再是纯粹随机的了，而是向目标点集聚，继续的解算就是不断地重复上述的步骤，直至获得最佳的答案，此处是山峰的最高点。

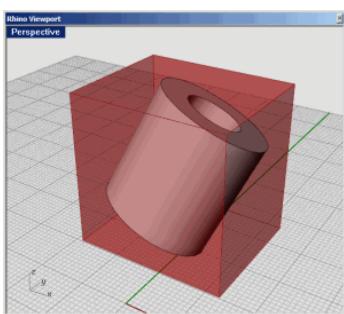
Fitness Functions

Fitness适合函数

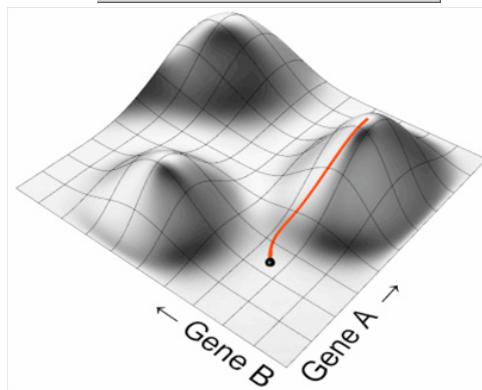
实际上很难具体的去说明Fitness，这需要结合实际的求解目的，例如Karamba中对最小最大位移的求解，以获得初始控制基因组的参数条件，或者上述对最高山峰点的求解，因此Fitness可能是任何事物，最强的，最快的，甚至最不好的，但是它们都有一个目的，所有答案中最大的Maximize或者最小的Minimize。



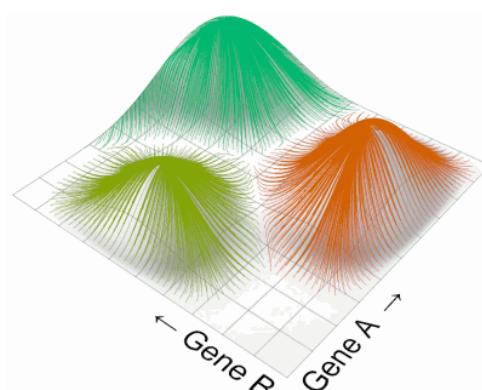
这是一个很简单的例子，如何获得绿色有机图形面积最小的正交外接矩形，图中给出了A和B两个选择，B的面积比A要小，因此更趋向于适合Fitness。



如果想使用一个最小体积的正交立方体来切割出想要的零件，使所消耗的材料最小，可以使外接立方体的X和Y轴可以自由旋转，最高角度为360度，而对于正交立方体，90度就可以了，将该区域作为Galapagos的基因端输入，进行解算。但是同时需要考虑旋转的精度，对于一个4位小数精度的旋转会产生810, 000, 000, 000次计算，因此应考虑求解的精度问题。

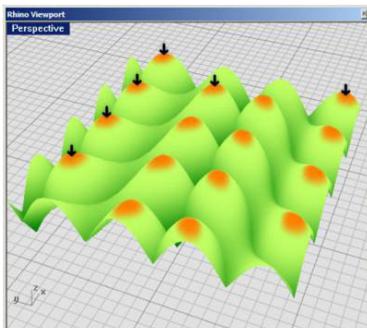


跟随解算过程中一个基因的演化历史，虽然会有相当数量的随机后代，但是具有一个明显的基因演化趋势，正像沿着山脉最低陡坡向下流淌的溪流。这里则是沿着最陡的方向向上攀爬。

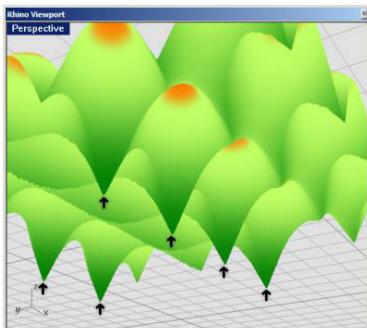


每一个基因个体都试图最大化自身的适合度，只有具有较高适合度的基因才会被解算器认可，并获得进一步的繁殖。因此橘色的基因繁殖具有更高的繁殖优势。

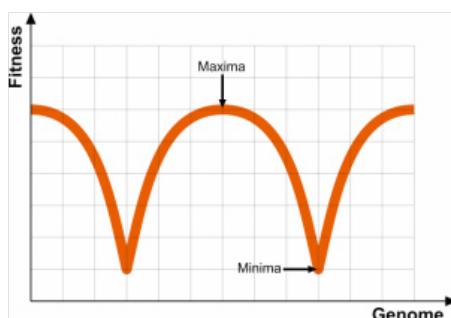
因为每一个基因都向上方攀爬，每个山峰都有一个围绕它的吸引流域。这些流域表示模型中所有点进一步向最高山峰点收敛。需要注意的是，这些流域并不代表山峰的品质。实际上，一个糟糕的解算可能有非常多的吸引流域并且这些流域又很小。类似这样的解算就会有些棘手，解算也容易卡住。



首先，进一步分析下那个最小外接正交立方体的问题，它并没有图中看起来那么简单，如果将此与山峰解算问题连同起来，可以认为GeneA和GeneB分别代表x轴和y轴的旋转。因此每一对AB就代表一个两个方向旋转的共同结果。

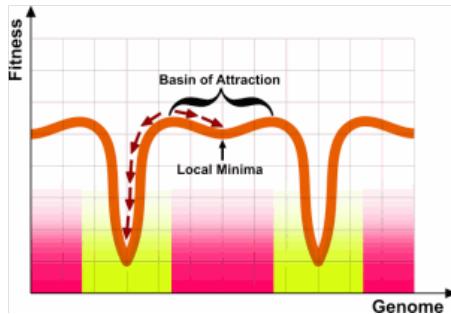


需要注意的是此时山脉是周期性的，在两个方向上每90度就会重复一次。同时对于最小外接正交立方体的解算也不再是寻找山峰的最高点，而是与之相反。在寻找适合的最低点时，会发现围绕最低点的适合流域很小，并且很陡峭，其最低点的值基本相当。



可以把这些具有相同质量的解算以二维图示表达。山峰鼓起处为最大值，陡峭最低处为最小值。

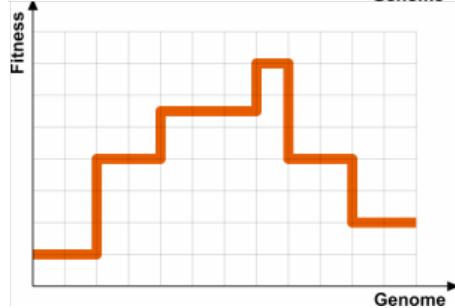
然而不管你的祖先们是何种基因，后代都会向适合的流域繁衍。在各自繁衍的过程中，可能会出现哪个基因后代最先到达最适合的流域，此时是最低点，但是这个最低点却不是正确的解。



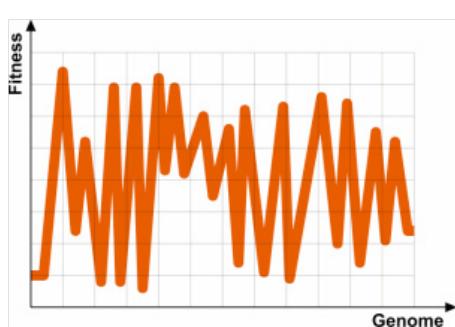
图中有两个发展的趋势，较高的解算位于山谷最低处，黄色区域，而质量较低的解算位于粉色区域。此时吸引流域给出了两个解答，而且一半以上的区域都位于低质量的求解区域。



更糟糕的是山峰非常的陡峭，在随机获得最初的基因时，很可能错过右侧的山峰，而捕获了左边的，所产生的后代会在此处迅速的繁衍，而其余的基因逐渐消失，找到右侧山峰最高点的机会逐渐消失。使用进化算法处理此类问题有些棘手。



甚至当山峰变得平坦时，基因总是会在原地打转，因为其值都是相等的，直到一个基因很幸运的向高处跳到一个台阶时，它会迅速的统治基因库，但是会再一次不断地游离，直到下一个偶然发生。



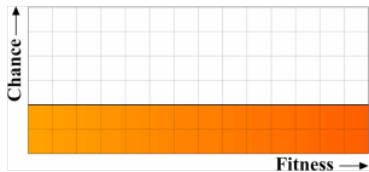
更坏的情况是类似于噪音似的山脉，很难明智的判别出区域适合还是全局适合。

Selection Mechanisms

选择机制

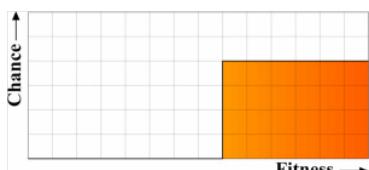
达尔文的自然选择进化理论学说，可以简单的认为自然选择影响着基因库的选择方向，判断在时间的进程中谁能够产生子代。一个极端的例子可以是因为一个特别的基因而使其在生育年龄而无法繁殖后代。而自然选择的高度复杂性，使得不太可能记录特定基因繁殖的足迹。

然而，目前人类也可以通过人工的干预获得子类。如同在使用进化计算方法时，可以进行人为的选择。这个选择要比自然选择容易的多，只需要判断由谁来产生子代。

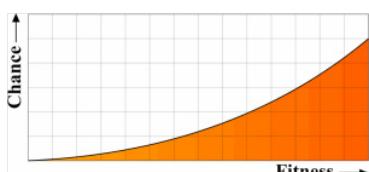


可以想象下最简单的一种选择，等方性选择（Isotropic Selection）。实际上，这是一种不存在的选择进化方式。在该类选择中，每个基因都可以繁殖后代。

不管在这个Fitness图示中哪个位置，都具有成为最适选择的机会。这种选择也许是毫无意义的，但是自然界中存在类似的例子，例如风力传粉或者珊瑚产卵。等方性式的选择方式可以被看作一种安全方式的进化算法。



Exclusive Selection排它选择的方式，只有顶端N% 的区域种可以获得繁殖的机会。

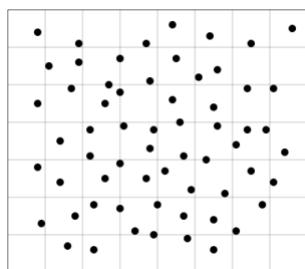


Biased Selection偏向选择，则是适合繁殖的种具有更多的繁殖机会，这与现实世界中的自然选择很相似。可以使用幂函数表达。

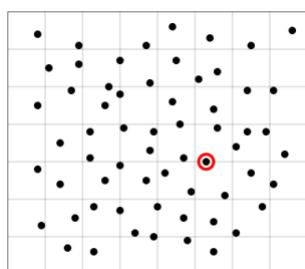
Coupling Algorithms

配对算法

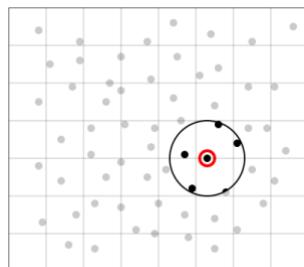
配对就是找到合适的配偶，进而繁殖后代，一旦根据选择算法获得了雄性，就需要在基因（个体）中挑选雌性来完成配对。在选择过程中存在很多方式，但是在Galapagos中仅允许选择一个，通过基因间的距离。



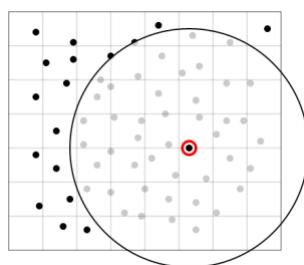
可以使用基因图谱来进行说明，在网格中以黑点的方式显示了特定种群所有的基因（个体）。黑点间的距离近似为基因个体之间的距离。而一个基因组是一类基因的群体，基因组之间的距离是一个N维的值，N等于基因的数量。当然在二维空间中很难表达N维空间中的点，仅能作为一种估计。通过基因图只能看到越近距离的越具有相似性，反之则具有不同品质。



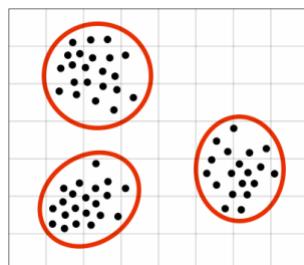
想象一下，红色圈住的基因是首先被选择的雄性。



可以在一定近距离内进行基因的配对，但是这如同近亲婚姻，存在隐性的基因缺陷。在进化计算中则是过快的衰减，阻断了更丰富的多样性增长，将计算区域限制在一个局域的适合流域中，限制找到全局中的最适配偶。

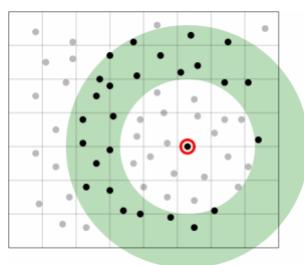


另一个极端是排除所有近距离的基因，但是因为较远的距离其各自的品质会更加相对，并不能确定这个配对是适合的。



在被称为 zoophilic 配对中，不再是一组基因组，而是多个，每个基因组都会在各的区域内找到最适配对。

当然，也希望在不同子基因组间进行配对，那么子代将会被放置于基因图中间的某个位置而进入一个不同的适合流域。



最好的方式就是在近亲与远亲之间的某个区域获得配偶。目前配偶的选择并没有考虑配偶的适合性，但是即使没有高级的选择算法，解算器也是可以工作的。

Coalescence Algorithms

联合算法

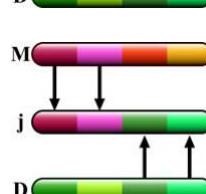
一旦选择了配偶，就会产生子代。实际的生物进化过程中基因重组是相当复杂的。而参数变量则相对要简单。因为进化算法并不是与生物基因重组相似。虽然，生物基因更加的数字化。而在18世纪60年代，Mendel发现生物基因并不是连续的变量，更像一个开关在进行控制。



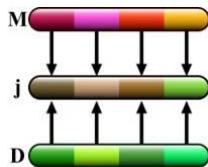
在配对基因组时，需要确定子代基因的值。获得该值的方法，Galapagos给出了几种方式。



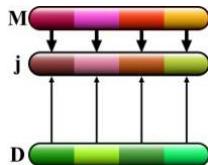
可以想象下有四个基因的两个基因组（染色体组）M和D。



在Crossover mating方式下，子代继承了来自母亲的随机基因，余下的则来自父亲。



Blend Coalescence方式获得的基因值则是来自于父母的平均值。



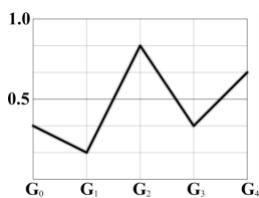
当然在子代中，也可以从更适合的母亲或父亲中获得较多的遗传基因，使其在子代的基因中占据更多的分量。

Mutation Factories

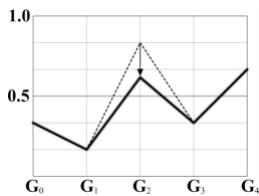
变异

前面讨论了选择Selection，配对Coupling，以及联合Coalescence用于改进解算的质量。然而，这也降低了演化的多样性。可以使用变异Mutation 产生多样性。在Galapagos核心算法中有几种变异的方式。目前在Grasshopper中仅限于点的变异。

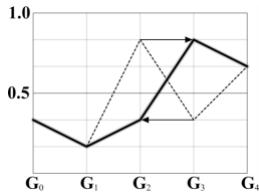
这里 David Rutten仍然使用图式的方式对其进行解释。多维的点可以在二维中粗略的表示，为了表示大量点的存在，将典型的点进行连接。



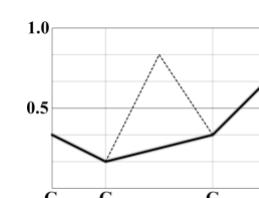
这里使用了包含5个基因的基因组。G0位于1/3的位置，其值位于最大和最小范围内。此种图式能够很容易的读出种群里的亚种，以及独立的个体。当使用变异时，每个独立的基因组都具有独立的图式。



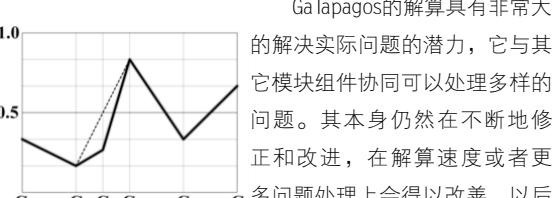
左图中显示了一个点的突变，单独的一个基因值发生了改变，目前只有该类突变在Galapagos中是可行的。



Inversion mutations倒置变异，只在特别关系下是有用的。他试图对基因有较大的变动，并因此使得Fitness适合变化较为剧烈。多数情况下这都是不利的。



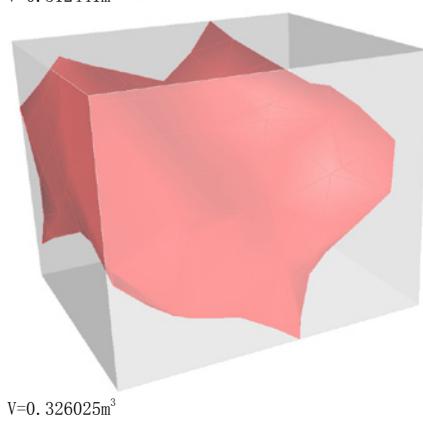
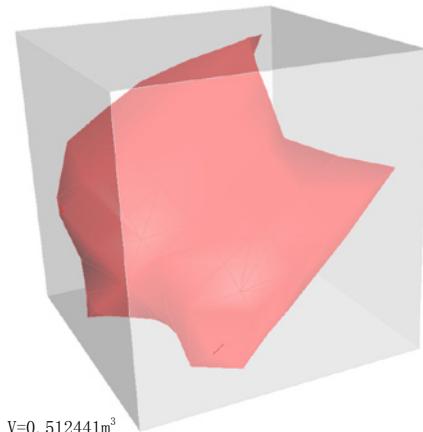
这两个变异示例，需要一个确定的基因值被增加Addition 或者被移除Deletion。



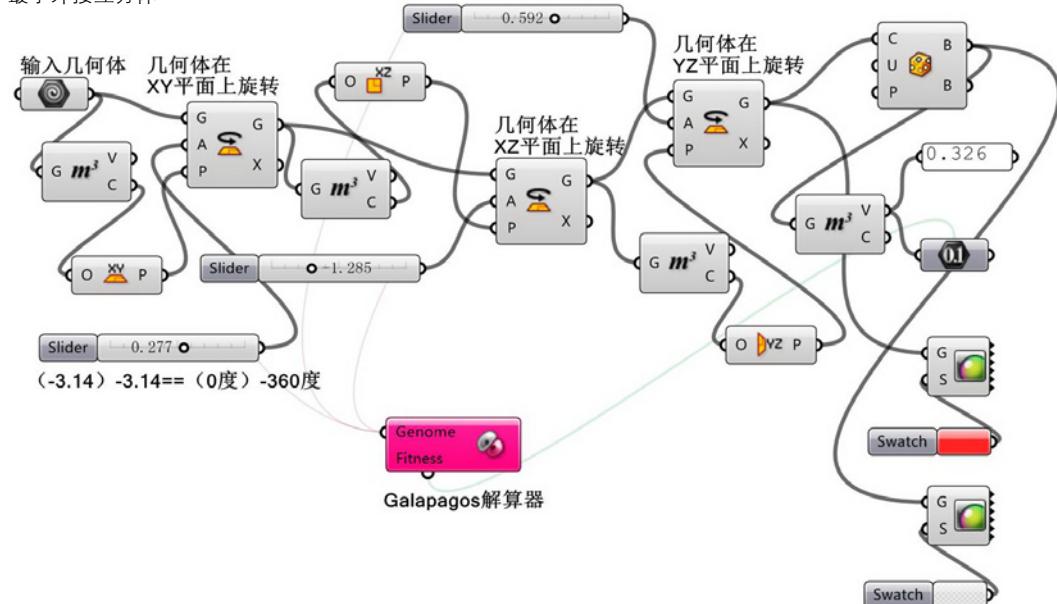
Conclusion

结论

Galapagos的解算具有非常大的解决实际问题的潜力，它与其它模块组件协同可以处理多样的问题。其本身仍然在不断地修正和改进，在解算速度或者更多问题处理上会得以改善，以后Galapagos的进化算法仍然会给设计师以更多的惊喜。



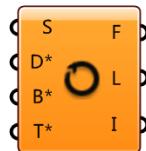
最小外接立方体



2 HoopSnake

<http://www.grasshopper3d.com/group/hoopsnake>

Iterative Process 迭代



在Grasshopper中参数化实现过程基本存在两种状态，一种是绝大部分组件所具有的静态参数化过程，输入，输出的数据都是以列表的方式出现，如果想要以时间前后的方式逐一使用列表中的数据进行某种行为的操作，就需要遍历列表，使用另一种动态的参数化过程，在Python程序语言中可以是For循环，或者进一步增加处理的难度，将第一次产生的结果作为下一次初始条件的迭代过程往复循环。

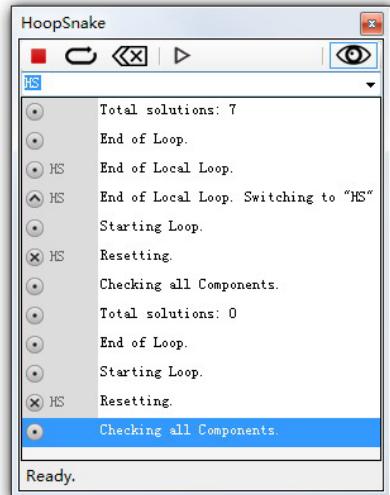
HoopSnake可以以节点的方式来处理动态的参数化过程（在语言中，例如Python, C#等很容易实现），获得程序语言中迭代的作用，将每次处理后的结果作为下一次的开始条件再次循环，并记录每次循环的结果。

S: 开始值，迭代开始前输入的第一个参数值；

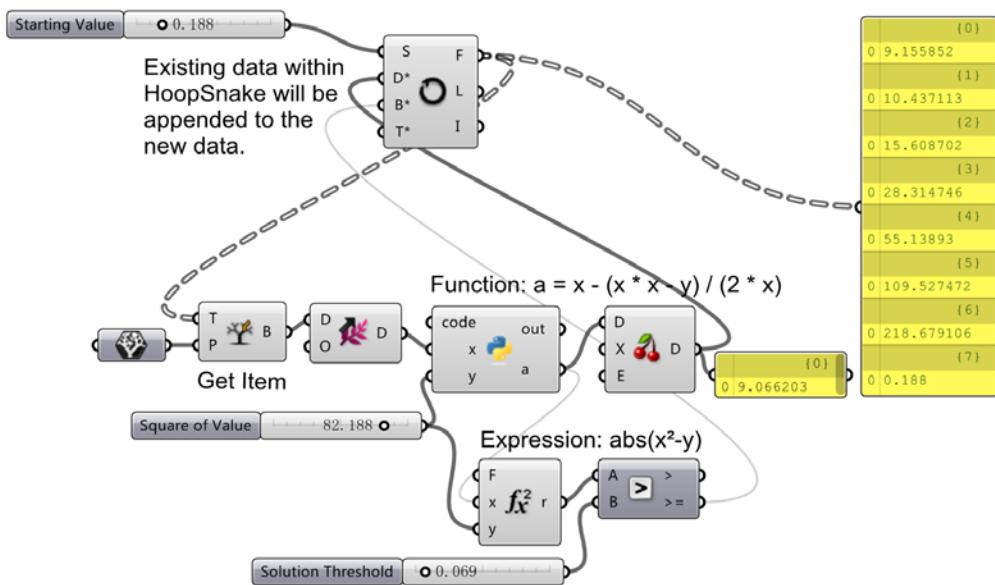
D: 返回值，迭代开始后经过一系列组件处理获得的结果值；

B: 控制值，给定迭代终止的条件；

T: 循环控制开关；



G-04_02



迭代+迭代器+生成器+递归

迭代 iteration 是重复反馈过程的活动，其目的通常是为了逼近所需的目标或结果。每一次对过程的重复被称为一次“迭代”，而每一次迭代得到的结果会被用来作为下一次迭代的初始值。

在计算机科学中，迭代是程序中对一组指令（或一定步骤）的重复。它既可以被用作通用的术语（与“重复”同义），也可以用来描述一种特定形式的具有可变状态的重复。

在第一种意义下，递归是迭代的一个例子，但是通常使用一种递归式的表达。比如用 $0!=1$ ， $n!=n*(n-1)!$ 来表示阶乘。而迭代通常不是这样写的。

而在第二种（更严格的）意义下，迭代描述了在指令式编程语言中使用的编程风格。与之形成对比的是递归，它更偏向于声明式的风格。

在 Python 中迭代序列（或者其它可迭代的对象）时，有一些非常有用的函数，例如并行迭代，编号迭代以及翻转和排序迭代等。

有些迭代的函数则位于 `itertools` 模块中，`itertools` 有很多工具用来创建和联合迭代器（或者其它可迭代的对象），还可以实现：将可迭代的对象链接起来，创建返回无限连续整数的迭代器（和 `range` 类似，但是没有上限），从而通过重复访问可迭代对象进行循环等等。

```
dir(itertools)
['__doc__', '__name__', '__package__', 'accumulate', 'chain', 'combinations',
'combinations_with_replacement', 'compress', 'count', 'cycle', 'dropwhile',
'filterfalse', 'groupby', 'islice', 'permutations', 'product', 'repeat', 'starmap',
'takewhile', 'tee', 'zip_longest']

#Python 语言
variablenames=['v1','v2','v3','v4','v5']
values=[12,78,32,102,23]
variablelst=[]
for i in range(len(variablenames)):#for循环迭代
    variablelst.append((variablenames[i],values[i]))
print(variablelst)#variablelist=[('v1', 12), ('v2', 78), ('v3', 32), ('v4', 102), ('v5', 23)]
```

```
zlst=list(zip(variablenames,values))#使用zip函数的并行迭代
print(zlst)#zlst=[('v1', 12), ('v2', 78), ('v3', 32), ('v4', 102), ('v5', 23)]
```

```
sortlst=sorted(values)#排序
print(sortlst)#sortlist=[12, 23, 32, 78, 102]

reverselst=list(reversed(sortlst))#翻转
print(reverselst)#reverselst=[102, 78, 32, 23, 12]
seq=['102','78','32','23','12']
for index,val in enumerate(seq):#使用内建enumerate函数的编号迭代
    if '78' in val:
        seq[index]='578'
```

```
print(seq) #seq=['102', '578', '32', '23', '12']
```

```
#迭代前初始化 a=0
#Python 语言 lst=[]
lst.append(a)
#循环5次 for i in range(5):
#a值增加i a+=i
lst.append(a)
print(lst)
```

迭代结果 `lst=[0, 0, 1, 3, 6, 10]`

迭代器 iterator—有时又称光标 (cursor) 是程式设计的软件设计模式，可在容器物件 (container，例如list或vector) 上遍访的接口，设计人员无需关心容器物件的内容。

各种语言实作Iterator的方式皆不尽同，有些面向对象语言像Java, C#, Python, Delphi都已将Iterator的特性内建语言当中，完美的跟语言整合，称之为隐式迭代器 (implicit iterator)，但像是C++语言本身就没有Iterator的特色，但STL仍利用template实作了功能强大的iterator。

Iterator另一方面还可以整合Generator。有些语言将二者视为同一接口，有些语言则将之独立化。

Python的_iter_方法：

_iter_是迭代器规则 (iterator protocol)的基础，它返回一个迭代器 (iterator)，使其具有__next__方法 (这个方法在调用时不需要任何参数) 的对象。在调用next方法时，迭代器会返回它的下一个值。如果next方法被调用，但迭代器没有值可以返回，就会引发一个StopIteration异常。

使用迭代规则可以避免因为使用列表一次性获取所有值而占用过多的内存。另外迭代器更加简单也更加优雅。

迭代器在Python 3.0中有些变化，在新规则中，迭代器对象应实现__next__方法，而不是next。而新建函数next可以用于访问这个方法。换句话说，next (it) 等同于3.0之前版本中的it.next()。

class Fibs:#定义斐波那契数列类

```
def __init__(self):#初始化所有实例
    self.a=0
    self.b=1
def __next__(self):#实现next方法，返回下一个值
    self.a,self.b=self.b,self.a+self.b
    return self.a
def __iter__(self):#返回一个迭代器 ( iterator)
    return self
fibs=Fibs()
for f in fibs:
    if f>1000:#查找斐波那契数列中比1000大的数中最小的数
        print(f)#f=1597
        break
```

生成器Generator—连同迭代器一样是Python最强大的两个特性。但是生成器的概念更高级一些。它的创建如同创建函数一样简单。一般生成器包含yield语句，或者说任何包含yield语句的函数都称为生成器。而生成器的行为与普通函数有很大差别。其在于它不像return那样返回值，而是每次产生多个值。每次产生一个值 (使用yield) 语句，函数就会被冻结，即函数停在那点等待被激活。函数被激活后就从停止的那点开始执行。

P-04_02

```
def Fibs():
    a=0
    b=1
    yield a#可以理解为此时a值被存储到yield的生成器中
    yield b#可以理解为此时b值被存储到yield的生成器中
    while 1:
        next=a+b
        yield next#可以理解为此时next值被存储到yield的生成器中
        a=b
        b=next
import itertools
lst=list(itertools.islice(Fibs(),10))
print(lst)#lst=[0, 1, 1, 2, 3, 5, 8, 13, 21, 34]
```

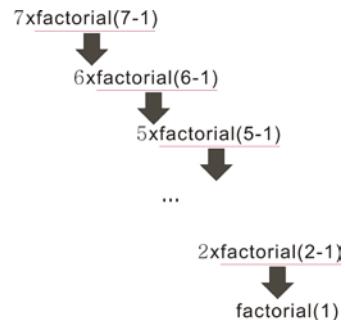
递归Recursion—在数学与计算机科学中，是指在函数的定义中使用函数自身的方法。一般来讲递归包含两个部分：

- 当函数直接返回值时有基本实例（最小可能性问题）；
- 递归实例，包括一个或者多个问题最小部分的递归调用；

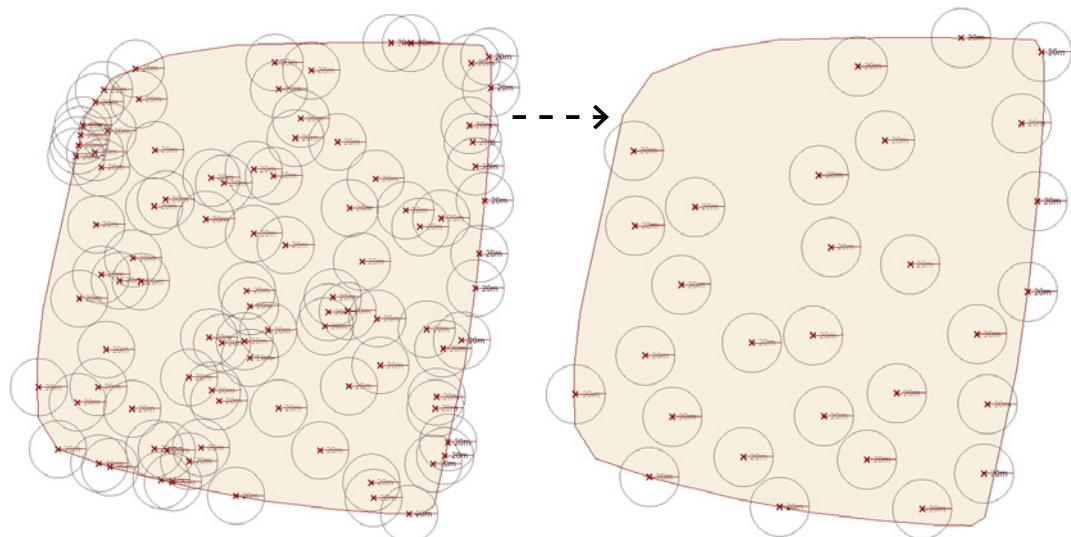
这里的关键就是将问题分成为小部分，递归不能永远继续下去，因为它总是以最小可能性问题结束，而这些问题又存储在基本实例中。每次函数被调用时，针对这个调用的新命名空间会被创建，意味着当函数调用“自身”时，实际上运行的是两个不同的函数（或者说同一个函数具有两个不同的命名空间）。

```
def factorial(n):#定义阶乘函数
    if n==1:#最小可能性，即1的阶乘是1
        return 1
    else:
        return n*factorial(n-1)#大于1的数的阶乘是n* (n-1) 的阶乘
print(factorial(7))#factorial(7)=5040
```

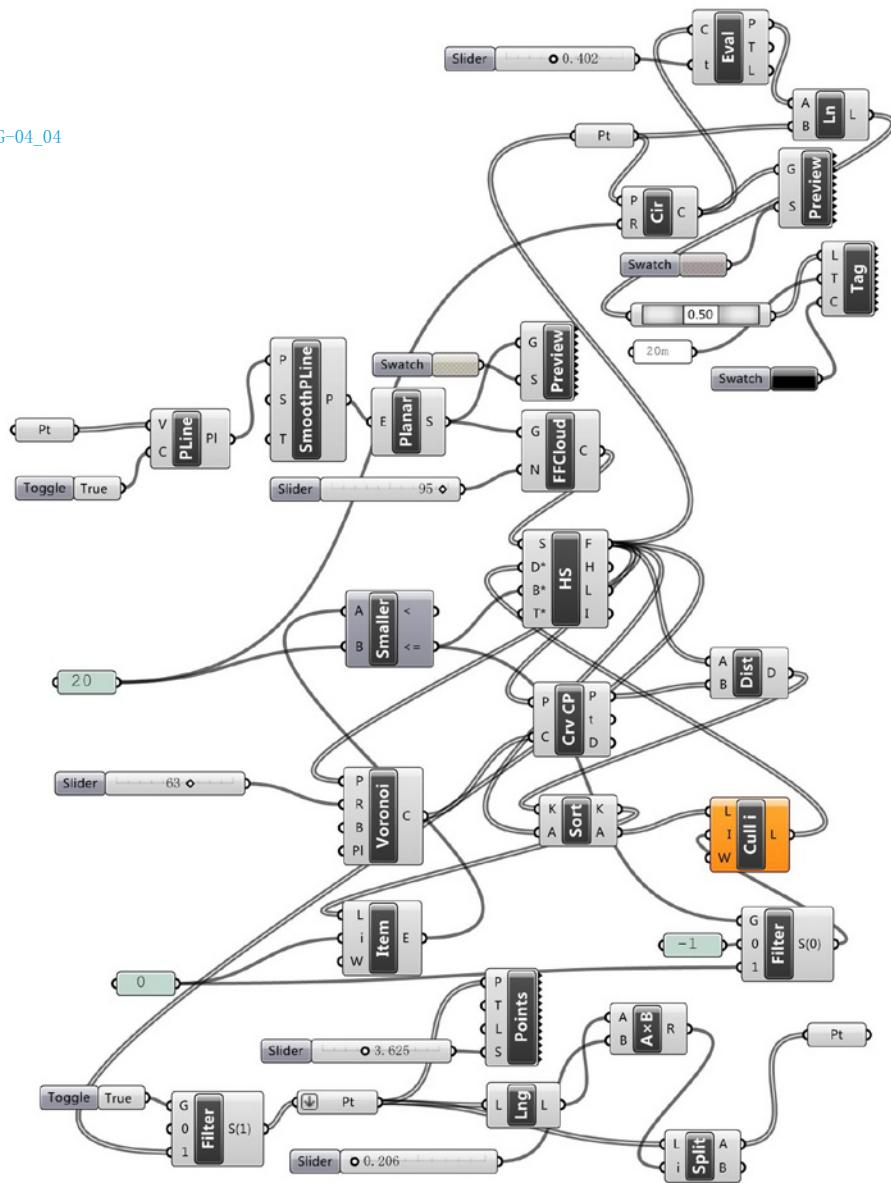
P-04_03



使用HoopSnake计算满足建筑间距要求的点位：



G-04_04



第五部分 表达

模型构建主要是辅助设计处理各种几何问题，包括各项分析以及几何建立，在处理这些过程中以及最后的成果，都需要图解和渲染来表达。针对表达也主要分为这两个方面：图解+渲染。

这里的图解是指能够辅助设计师梳理思维程序，并能够呈现给设计团队成员或委托方的各类分析图式，而渲染主要针对的是设计模型效果的表达。在整个设计过程里，过去即使目前往往只强调渲染方面的表达处理，即如何处理好效果图的问题，而容易忽略图解的表达。实际上在最后设计成果文本中，表达设计是如何处理的逻辑和效果图相比同等重要。

进行图解表达时，估计设计师最为常用的工具就是Adobe Photoshop和Microsoft Office Powerpoint，其实到目前已经有很多辅助此类的软件能够协助设计师快速的完成各项图解，从而使问题表达的更加清晰，节约不必要的时间损失。

渲染方面除非必要由专业效果图公司处理外，大部分图纸其实都是由设计师完成的，渲染是表达设计的必要和重要环节，但是将较多的时间用于渲染上而非设计推敲上，不免也是一种时间的浪费。因此一个优秀的渲染器，减少繁琐的设置，而又快速的获得理想的表达效果，应该是每个设计师都希望的。

1 图解



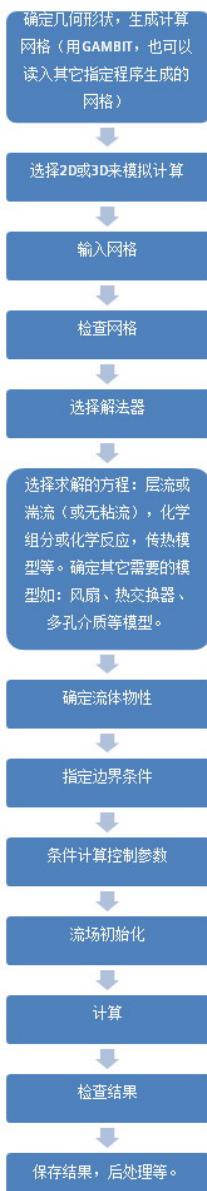
- Microsoft Office Powerpoint幻灯片
<http://office.microsoft.com/en-us/powerpoint/>

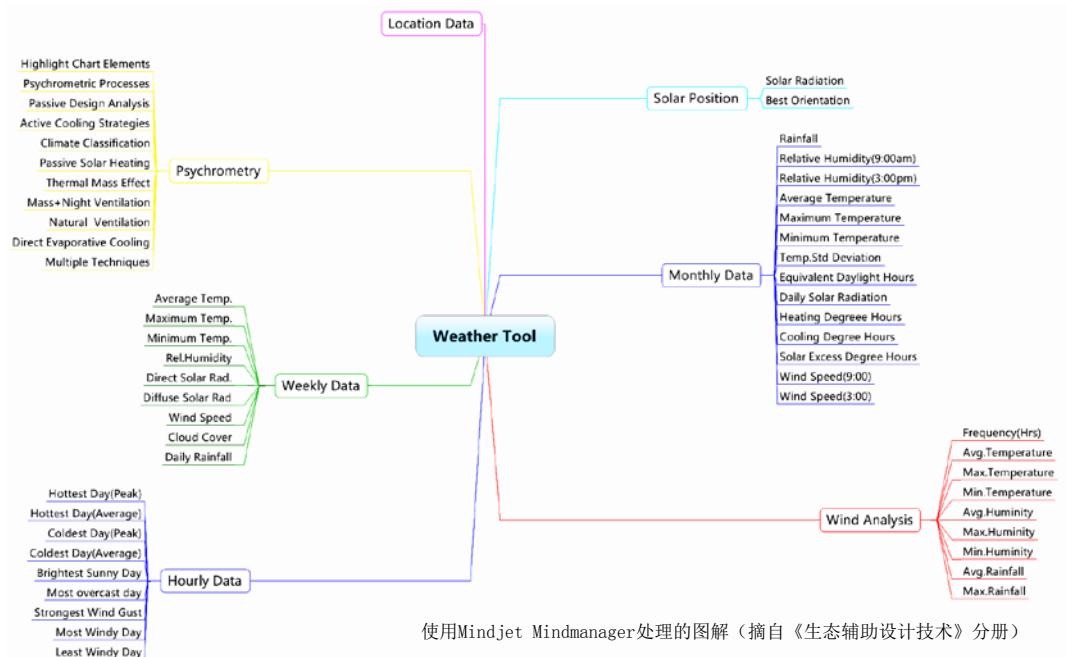
PPT主要是用于汇报的文件，但是其中很多功能图示可以协助设计师进行图解表达。而且最后汇报为了节约排版和制作汇报的时间，往往使用PPT进行排版，然后再摘取主要内容，直接处理为汇报文件。(注：对于文本和书籍的排版可以使用Adobe InDesign)

使用PPT处理的图解（摘自《生态辅助设计技术》分册）

- Mindjet Mindmanager思维导图
<http://www.mindjet.com/>

在小组头脑风暴的时候，经常会使用便签纸将各自的设计构思，想法写下来，然后分类粘贴在墙上来激发和梳理设计的概念，Mindjet Mindmanager类似于这样的行为。它在计算机中构建了一个创造，管理和交流思想的通用标准，有助于有序的组织思维，资源和项目的进程。





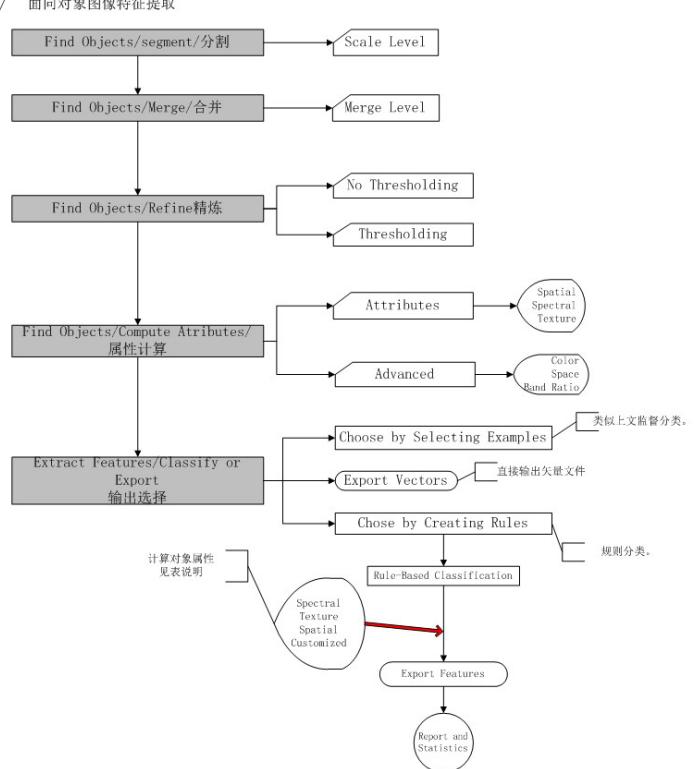
- Microsoft Office Visio流程图

<http://office.microsoft.com/en-us/visio/>

Feature Extraction(FX)
面向对象图像特征提取



在设计分析的时候，经常会需要处理分析的流程，Microsoft Visio可以协助设计师处理该方面的内容。它本身是Microsoft Office的一部分，其制作图表的范围十分广泛，例如地图，企业标志等，Visio支持将档案保存为svg, dwg等矢量格式。

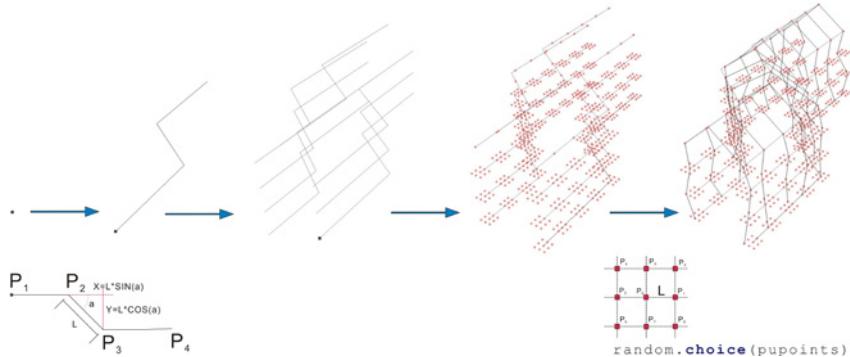


- CorelDRAW Graphics Suite平面设计

<http://www.corel.com/>



CorelDRAW是由加拿大Corel公司开发的图形图像软件，具有很强的设计绘制能力，，被广泛的应用与商标设计、标志制作、模型绘制、插图描画、排版及分色输出等等诸多领域。其优秀的交互式能力，智慧型绘图工具，降低了操作的难度，可以节省设计的时间。



使用CorelDRAW处理的图解（摘自《模型构建》本分册）

2 渲染

根据设计师所使用模型构建工具的不同，可能渲染的方式有所差异，每个三维建模软件基本都有自身的渲染器。同时也可以加载其它的渲染器，例如Vray。对于景观设计师而言，最为棘手的是植物表达，在传统设计中一般渲染的模型不包括树木等植被，而是在后期Photoshop平面处理，主要目的是植物的渲染会占据过多的CPU和内存，不是无法渲染就是要花费过长的时间。但是，使用后期平面处理的方法同样会消耗设计师宝贵的时间，而且这种处理基本是纯粹的图面美化，如果过多的经历花费于此，对于设计模型的真实性也会有所影响。

到目前为止，计算机的硬件以及软件的发展都有非常大的提升，尤其2010年左右，很多三维构建软件已经基本走向成熟，处理速度也已经满足日常个人电脑的要求，可以借助VUE，Lumion等后期处理软件来完成这项工作。

- V-ray

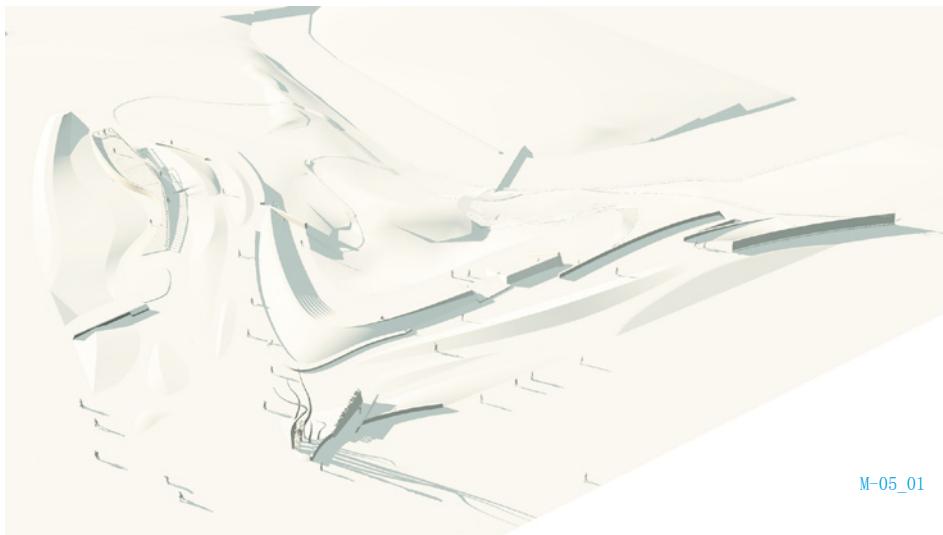
<http://www.vray.com/>



V-ray是由专业的渲染器开发公司chaosgroup开发的渲染软件，是目前业界最受欢迎的渲染引擎。

基于V-ray内核开发的有V-ray for 3Dmax、Maya、Sketchup、Rhino等诸多版本，为不同领域的优秀3D建模软件提供了高质量的图片和动画渲染。除此之外，V-ray也可以提供单独的渲染程序，方便使用者渲染各种图片。

V-ray渲染器提供了一种特殊的材质——V-raymtl。在场景中使用该材质能够获得更加准确的物理照明（光能分布），更快的渲染，反射和折射参数调节更方便。使用V-raymtl，可以应用不同的纹理贴图，控制其反射和折射，增加凹凸贴图和置换贴图，强制直接全局照明计算，选择用于材质的BRDF。



M-05_01

在RHINO中结合Grasshopper构建的模型，使用嵌入到RHINO中的VRAY进行渲染，可以非常真实的表达设计师对材质的构想，同时简单的设置即可达到真实的效果，渲染速度可以满足设计过程，模型构建中即使渲染，以便观察模型材质设计和设计模型体量感的需求。

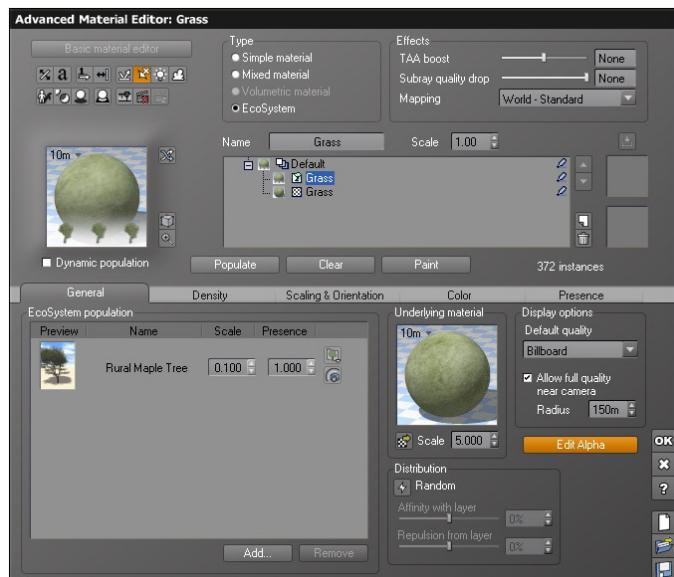
但是目前，Vray对于植被没有渲染支持，更多倾向于工业产品，纯粹建筑等的渲染表达。



- VUE景观设计
<http://www.e-onsoftware.com/>

Vue

对于景观设计师，最头疼的是植被以及自然环境的设计与表达，因为植被过多的面，使得CPU和内存无法支持其渲染，但是硬件性能的提升和软件技术的提高，这些都已经解决。VUE为专业人员提供解决方案，支持创建格外丰富和真实的数字自然环境。一般将设计模型存为OBJ等VUE可以识别的格式导入，再赋予材质，对于自然生态系统，可以使用内部的EcoSystem材质来处理大尺度的自然环境，而不是逐一的栽植每一颗树木。为了确定获得的自然系统更加符合设计的意图，例如某一个片区应该是没有树木，而另一个区则为疏林草地，或者茂密森林，可以将地面或者有起伏的地形各自根据设计目的单独为面导入，分别赋予EcoSystem材质进行设置即可。



V-05_01





- Lumion

<http://lumion3d.com/>



一直强调的是，设计师所希望的后期处理软件就是能够快速，高效获得设计的表达，并且能够达到令人满意的艺术效果。Lumion的出现无不切合这些基本条件，从而为设计师节省时间，精力和资金。但是在植被方面目前没有VUE的EcoSystem材质，大面积的植物种植仍旧比较耗时。

引自Lumion自带案例文件



参考文献：

1. [英]Cecil Balmond著. Informal异规. 李寒松译. 北京：中国建筑工业出版社，2008.
2. Kostas Terzidis. 2006. Algorithmic Architecture. Great Britain, Architectural Press.
3. Mohamad Khabazi. 2009. Algorithmic modelling with grasshopper(Rhino Plug-in).
4. Reiser+Umenoto. 2006. Atlas of novel tectonics. New York, Princeton Arhictecture.
5. Andrea. 2010. 3 years of DigitAG&. BlogBooker.
6. Lumen, Inc. 2009. Digital-botanic architecture2•eBook. New Mexico.
7. Michael Frame, Benoit Mandelbrot. 2009. Nial Neger. Fractal Geometry. Yale University.
8. Zubin Mohammad Khabazi. 2010. Generative algorithms.
9. Andrew Payne & Rajaa Issa. 2009. Grasshopper primer for version 0.6.0007.
10. ncf. 2009. 中文运算器教程Tutorial for components.
11. Michael Meredith, Aranda-lasch, Mutsuro Sasaki. 2008. From control to design:Parametric/Algorithmic architecture. Actar.
12. David Rutten. 2007. Rhinoscript for Rhinoceros. Robert McNeel&Associates.
13. 浦广益编著. Ansys workbench 12基础教程与实例详解. 北京：中国水利水电出版社，2010.
14. 李家宝, 洪范文主编. 建筑力学第三分册-结构力学(第四版). 北京：高等教育出版社，2006.
15. 王文栋编著. RhinoScript参数建模for Architecture. 北京：中国青年出版社，2011.
16. 增旭东, 王大川, 陈辉著. Rhinoceros&Grasshopper参数化建模. 武汉：华中科技大学出版社，2011.
17. [美]William Nash. 材料力学(第四版). 赵志岗译. 北京：科学出版社，2002.
18. David Rutten. 2010. Evolutionary principles applied to problem solving.
19. Clemens Preisinger. 2011. Parametric structural modeling Karamba user manual for version 0.9.007.
20. Skylar Tibbits, Arthur van der Harten, Steve Baer. 2011. Python for rhinoceros 5.
21. Daniel Piker. 2011. Kangaroo live physics for rhino and grasshopper using kangaroo(Grasshopper version) (Draft).
22. Zubin Mohammad Khabazi. 2010. Generative algorithms concepts and experiments:Weaving.
23. Rajaa Issa. 2011. Essential mathematics for computational design.
24. Zubin Khabzai. 2011. Generative algorithms concepts and experiments:Strip morphologies.
25. Magnus Lie Hetland. 2008. Beginning python from novice to professinoal, second edition. New York. Apress.
26. [挪]Magnus Lie Hetland. Begining Python from novice to professional second edition python基础教程, 司维, 曾军葳, 谭颖华译. 北京, 人民邮电出版社, 2010.
27. David M. Beazley. Python Essential Reference fourth edition python 参考手册. 谢俊, 杨越, 高伟译. 2011. 北京, 人民邮电出版社.
28. Rajaa Issa. 2011. Rhinoceros NURBS modeling for windows panelling tools.
29. Michael Zeiler. 1999. Modeling our world. New York. Environmental systems research institute, Inc.
30. Houdini Help.
31. Calatrava complete works1979–2007. 2007. Italy, TASCHEN GmbH.
32. a+u. Cecil Balmond赛西尔·巴尔蒙得. 北京：中国电力出版社，2008.
33. 张宝荣, 吕新欣等编著. Houdini学习总动员-基础卷. 北京：清华大学出版社. 2010.
34. 张宝荣, 吕新欣等编著. Houdini学习总动员-动力学卷. 北京：清华大学出版社. 2010.
35. 徐鹤山编著, ANSYS建筑工程实例分析. 北京：机械工业出版社. 2011.

后记

《基于Grasshopper的参数模型构建手册- II 》，与 I 不同，阅读本书一般需要对Grasshopper的参数模型构建有了一定认识之后，向更广领域研习的一本指南。自从进入到参数化领域，以数理逻辑的思维看待设计，融入了更多的相关学科协同设计，那么同时有必要了解相关学科的知识，例如结构力学，这样才能够正确地使用相关的组件，因此在该部分编入了涉及到的相关学科基本知识以方便查询。

Python语言在三维构建领域应用的范围在不断的扩大，在掌握了基本的Grasshopper参数化模型构建之后，有必要学习Python语言来增强自身构建与分析模型的能力。但是这不是一本关于Python的教程，而是讲述了Python应用的方法，能够使读者认识到Python应用的能力，在学习了该语言之后，可以阅读本书的Python程序来协助对该语言的学习。

同时讲述了两个特殊的组件Galapagos和Hoopsnake，它们都是具有迭代性质的组件，在《基于Grasshopper的参数模型构建手册- I 》中有对该组件详细讲解的相关案例，本书则侧重于对组件理论部分的解释，能够正确的理解组件的本质已达到对组件正确与熟练应用的目的。

实际上对Grasshopper参数设计的拓展远不止本书所涉及的这些，仅以此抛砖引玉，不管是自行拓展应用领域的组件来增强设计的能力，还是使用别人已经编写好的组件，参数化的领域仍旧在不断的拓展并不断的在完善，希望在这新兴的学科领域中坚持跟进，不至于形同陌路，是对此领域感兴趣的设计师要不断坚持的。

