20.211
Introduction to Design Computation
Fall 2019

Instructor

Jason Lim

Teaching Assistants

Tan Ying Yi
Kateryna Konieva

Course Description

This course is an introduction to concepts and methods, as well as practical techniques in architectural design computation. It is comprised of three learning segments: (a) foundations in computational geometry; (b) principles of algorithmic design; and (c) associative/parametric design. Students will gain experience in geometric modelling, visual programming, and reading and writing simple generative computer programs through a series of active learning sessions and design exercises.

In this course, students will learn to create, analyze and evaluate computational and geometric constructs in a design context. The course provides students with a basis for developing a critical approach towards using computational tools over the course of their architectural education and beyond. The success of students is evaluated not solely on technical accomplishments, but according to the integrity of the produced design processes and products in engaging the given theoretical concepts.

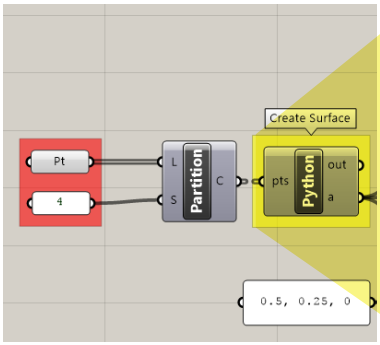
Learning Objectives

- Understand computational design approaches in the field of architecture
- Formulate design problems so that they can be solved via computational methods
- Acquire technical skills in geometric modeling, basic scripting and visual programming
- Explore and develop design solutions using computational means

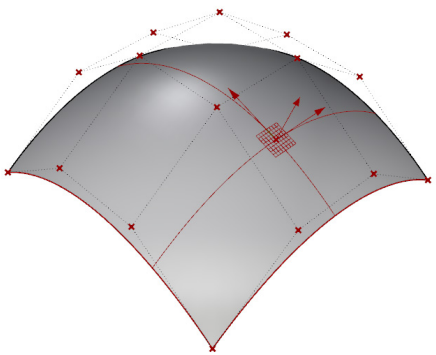
Measurable Outcomes

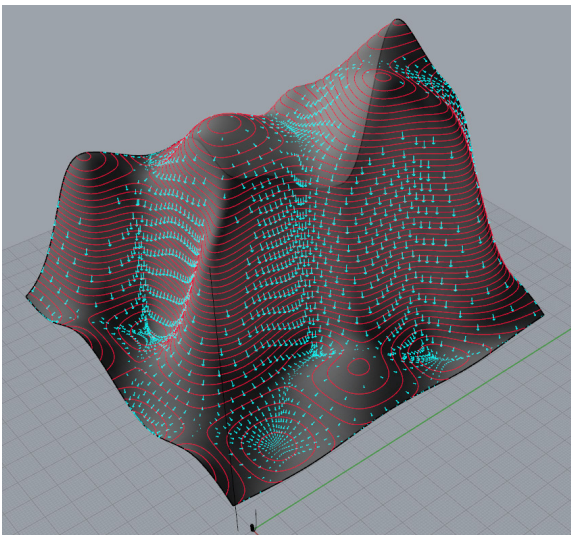
- Demonstrate a broad base of knowledge about design computation
- Gain and demonstrate knowledge in geometric modelling, scripting and visual programming
- Apply computational methods to generate, describe, and evaluate designs

Students are introduced to a hybrid approach to programming, involving the use of a textual programming language—Python—within a visual programming environment—Grasshopper. Through weekly exercises, students learn procedural and object oriented styles of programming in Python, as well as the visual dataflow paradigm in Grasshopper.



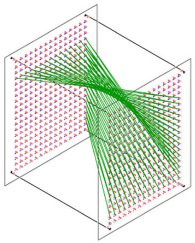

```python
from Rhino.Geometry import NurbsSurface

bIsRational = False
u_degree = v_degree = 3
u_cv_count = v_cv_count = 4
u_knot = v_knot = [0.0, 0.0, 0.0, 1.0, 1.0,  1.0]

# Creates internal uninitialized arrays for control
points and knots
nurbs_surface = NurbsSurface.Create(3, bIsRational, u_
degree + 1, v_degree + 1,u_cv_count, v_cv_count)
```





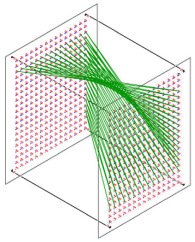
The course also introduces computational geometry concepts. It covers topic such as: points, lines, vectors, curves, surfaces and transformations. Students learn the mathematical basis of such concepts, as well as how to create and manipulate geometric types via programming.

RESULT 1A



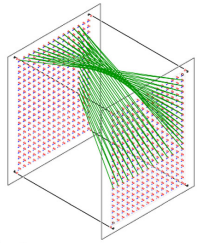
p1 start index = 15

RESULT 1B



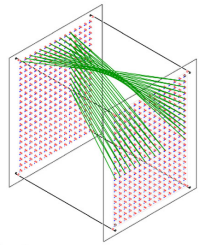
p1 start index = 30

RESULT 1C



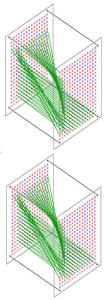
p1 start index = 90

RESULT 1D



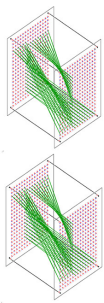
p1 start index = 120

RESULT 2A



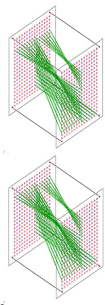
p1 start index = 3
p1 start index = 18

RESULT 2B



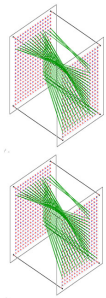
p1 start index = 6
p1 start index = 21

RESULT 2C

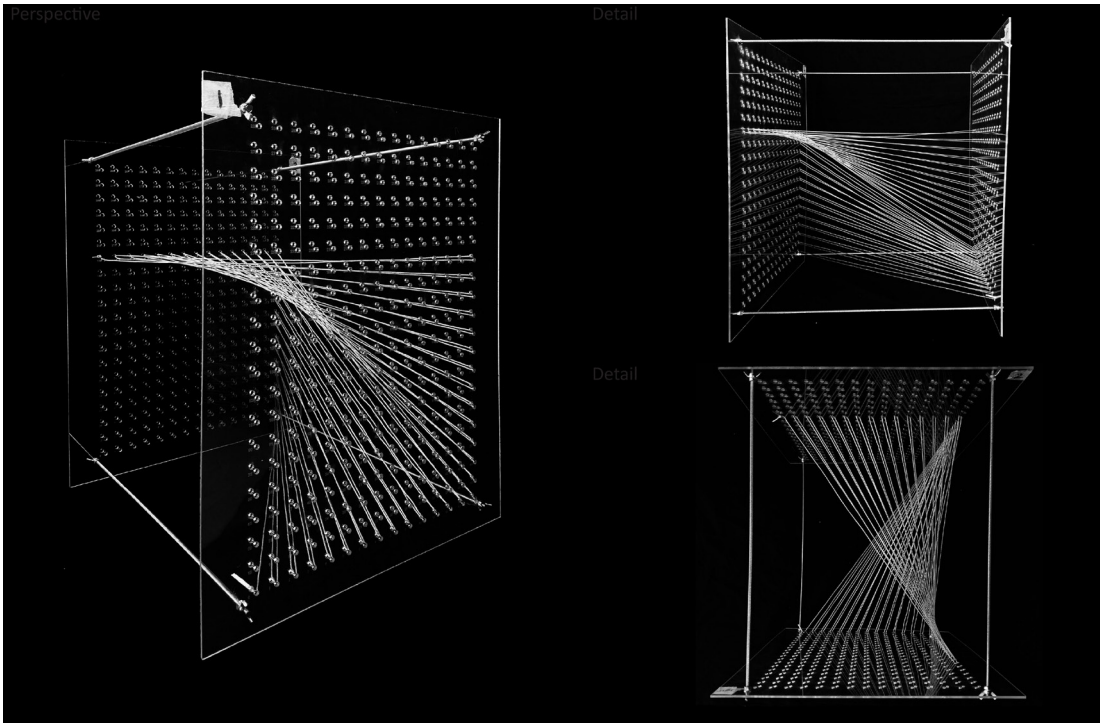


p1 start index = 9
p1 start index = 24

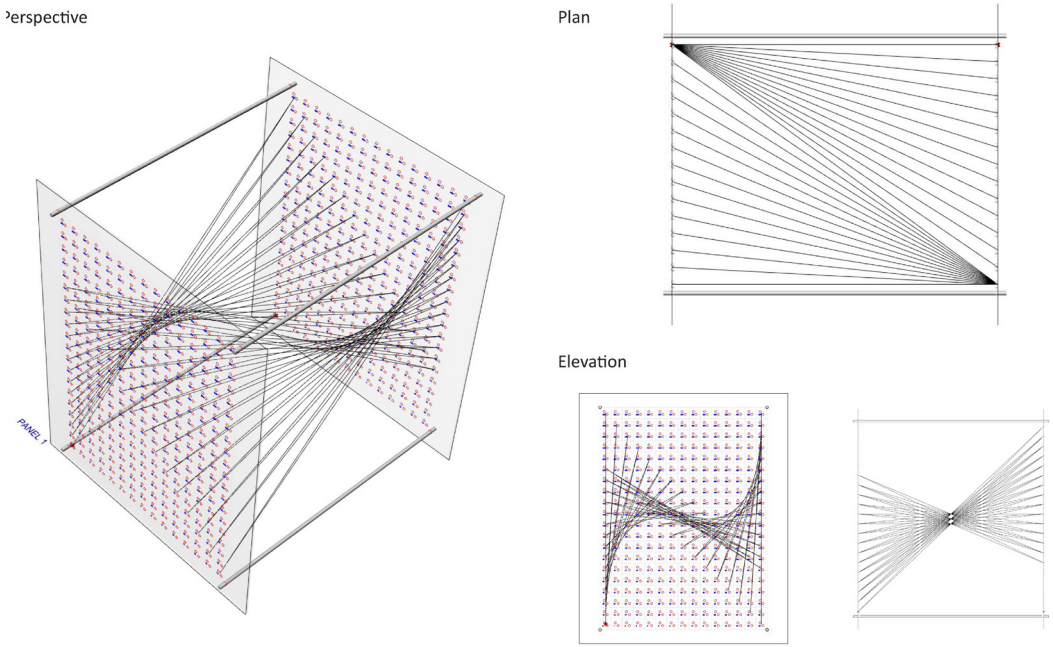
RESULT 2D

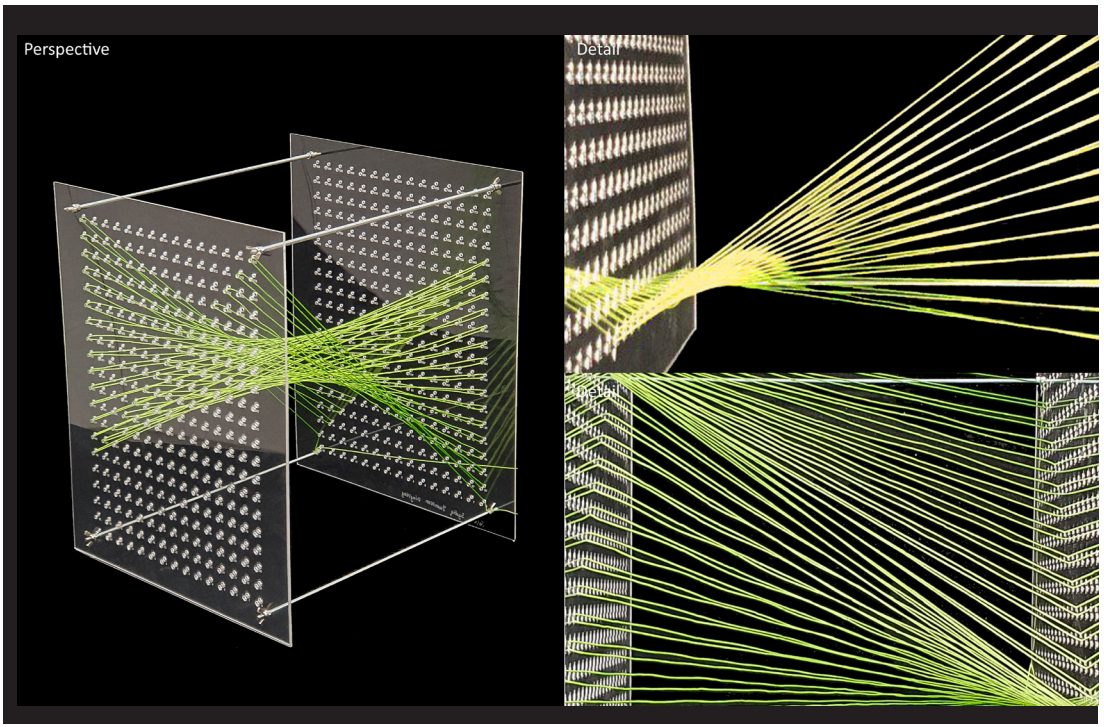


p1 start index = 12
p1 start index = 27

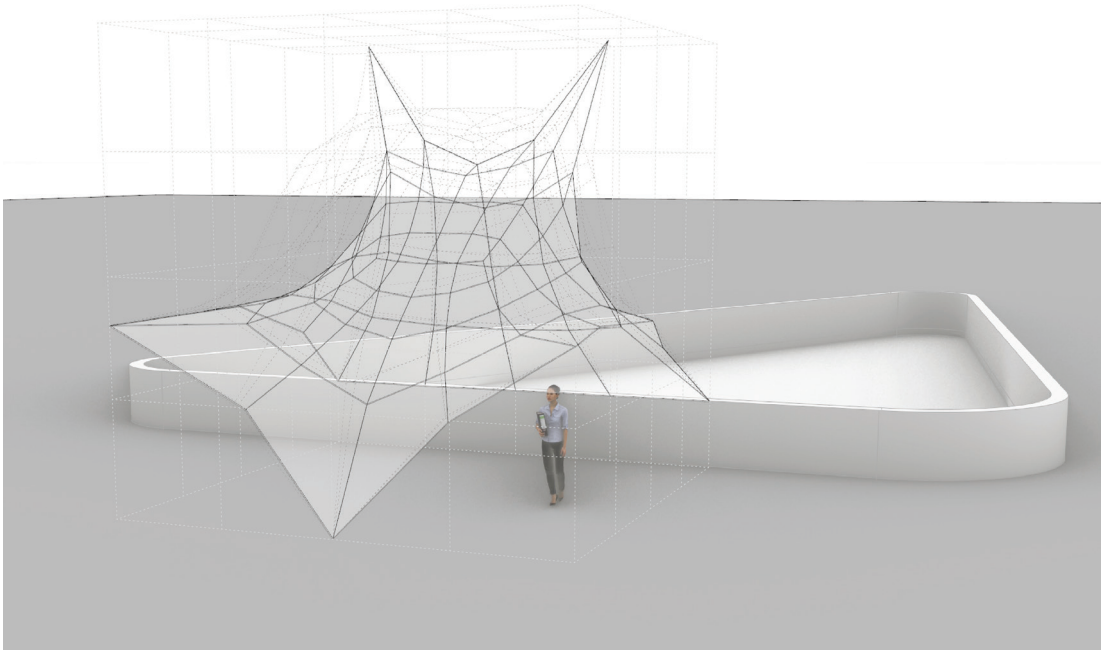


Assignment 1: A Single Thread by Chong Shi Qing, Han Xianhe, Melvin Wong and Renee Tan

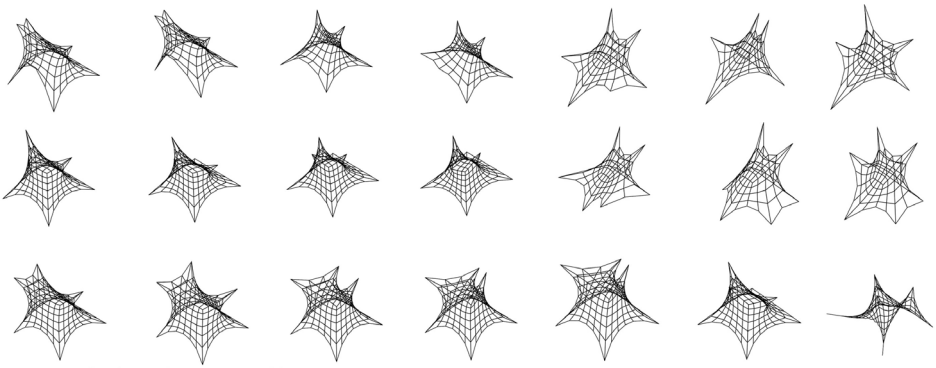




Assignment 1: A Single Thread by Ching Yuan Wen, Muhammad Syafiq Bin Norkhalim, See Tow Jo Wee and Thon Way Way



INITIAL ITERATIONS



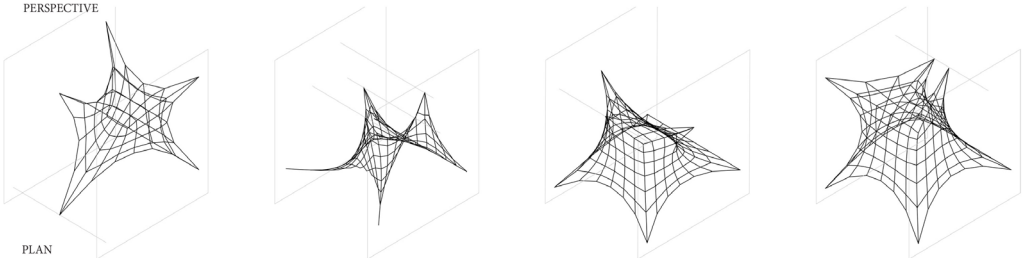
RESULT 1A

RESULT 1B

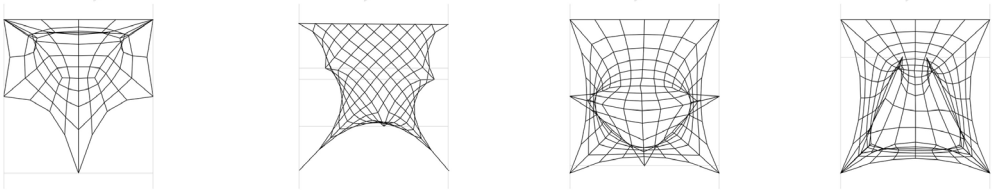
RESULT 1C

RESULT 1D

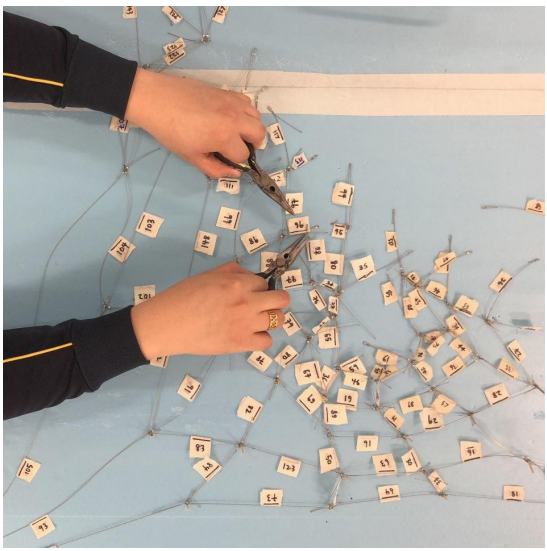
PERSPECTIVE

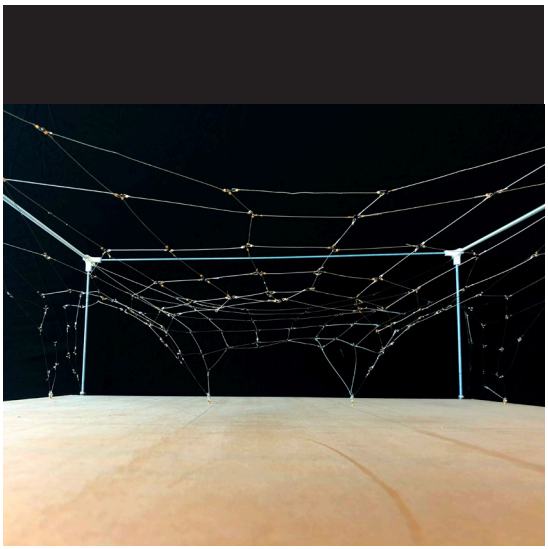


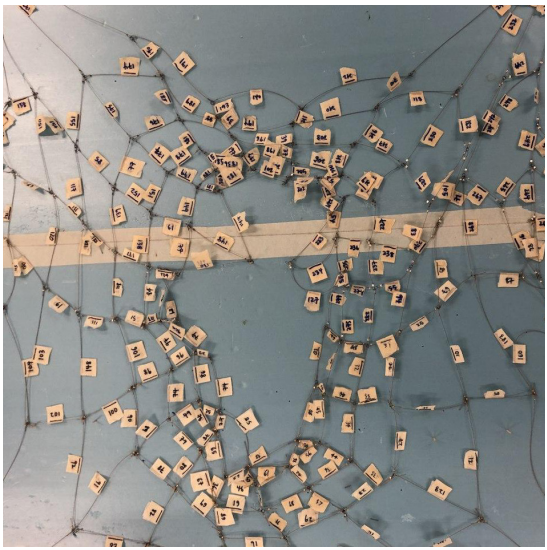
PLAN

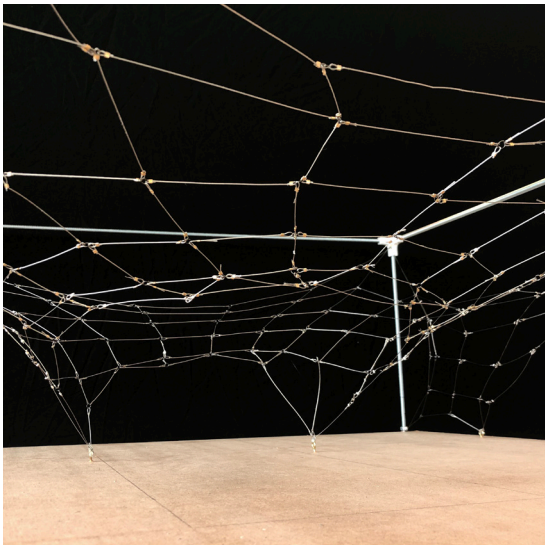


Assignment 2: The Pop-up structure (left) by Chong Yuan Wen, Keith Lim, Lynus Lim and Yoo Fei Yi. Students use a particle spring system for form-finding; they generate variations of a tensile struction based on different initial support conditions.









Assignment 2: The Pop-up structure (above) by Alastair Chew, Han Xianhe, Koh Fang Yun and Nurul Nazeera Binte Yazid. Students extract fabrication related information directly from the computational model in order to build a 1:10 steel wire model.