User Guide
FEA Interface for Lattice Structure

Sep 2015

Version 0.1

ADML @ McGill University

Written by: Yunlong Tang

DOCUMENT REVISIONS

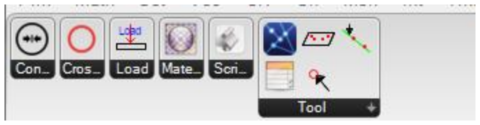
Date	Version Number	Document Changes
05/09/2015	0.1	Initial Draft

TABLE OF CONTENTS

1 INTRODUCTION

1.1 SCOPE AND PURPOSE

FEA-Interface is a group of grasshopper components which can provide a link between lattice design and FEA solvers. This document is a brief manual for using this component to do the analysis of designed lattice structure generated by Intralattice core module or other lattice design and generation components. The standard process of using this group of components will be described in the first section. Then, detailed discussion for each component will be made.



1.2 PROCESS OVERVIEW

The overall lattice simulation process is shown in Figure. It can be divided into three major steps. They are lattice design, FEA script generation, problem solving and post-processing. This manual primarily focuses on the FEA interface between lattice frame and commercial FEA solver like Nastran. The set of tools described in manual plays a role as a pre-processing software for FEA solver. In the current analysis, in order to avoid a heavy computational load, lattice struts are regarded as beam elements. The developed Grasshopper assembly provides a solution to convert designed lattice structure into FEA model with beam elements. This model is output in the format of scripts which can be read and solved by the commercial FEA solver. It should be noted the unit system use for this assembly is (Kg, N, mm, MPa)

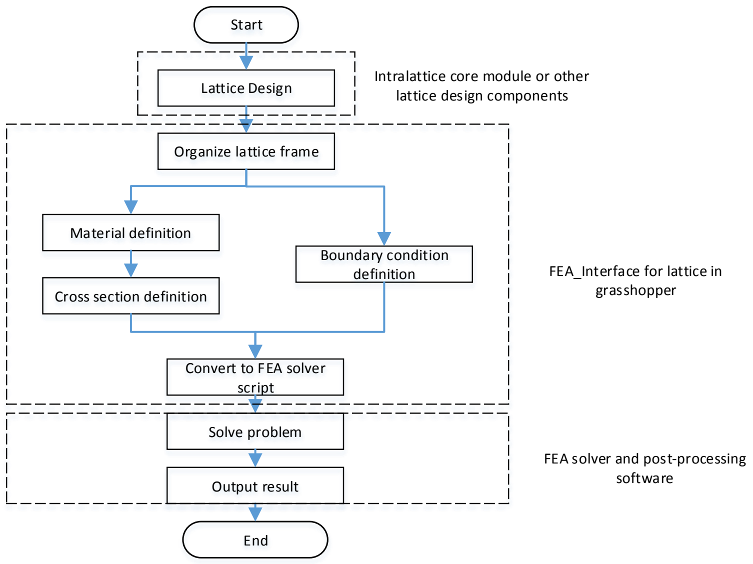

Start
Lattice Design
Intralattice core module or other lattice design components
Organize lattice frame
Material definition
Boundary condition definition
Cross section definition
FEA_Interface for lattice in grasshopper
Convert to FEA solver script
Solve problem
Output result
FEA solver and post-processing software
End

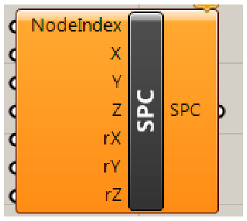
Figure 1 Process Overview

2 Components Introduction

This section will discuss the main function of each component individually. General, the tools described in this manual can be divided into six sub-categories according to its property. They will be discussed respectively in the following contents.

2.1 Constraints

2.1.1 Single Point Constraints (SPC) Component



Function: SPC component is used to define a single point constraint. In a single point constraint one or more degrees of freedom are fixed for a given node.

Input:

NodeIndex: Index of node. This index can be obtained from the node selection components in tool sub-category

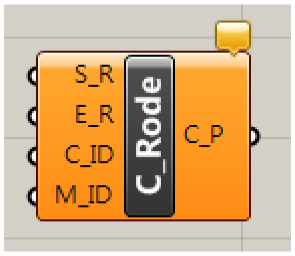
Freedom of constraints (x,y,z,rx,ry,rz): There are totally six constraints of freedom. Three of them (x,y,z) are translational constraints along x,y,z axis. Another three are rotational constraints. Use bool value true of false to control the whether this degree of freedom is fixed. If true, this degree of freedom is fixed, vice-versa.

Output:

SPC: Constraints script used for FEA script generation

2.2 Cross Section

2.2.1 ROD section



Function: This component is used to define the cross section of the beam. Particularly, it is used to define the circular cross section.

©ADML lab

Input:

S_R: Start-radius of rod cross section

E_R: End-radius of rod cross section

C_ID: cross section ID (Integer)

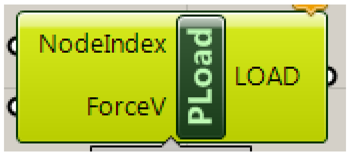
M_ID: material ID (Integer), material used for this cross section

Output:

SPC: Cross section script definition

2.3 LOAD

2.3.1 Point Load



Function: Define the point load

Input:

NodeIndex: Index of a node. This index can be obtained from the node selection components in tool sub-category

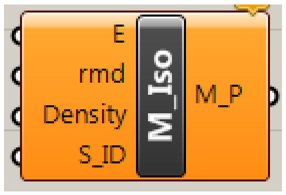
ForceV: Force Vector. It represents the direction and magnitude of the force

Output:

LOAD: script for the load

2.4 MATERIAL

2.4.1 M_Iso (Material_Isotropic material)



Function: Define the material properties for the FEA analysis

Input:

E: Young's modulus of material

rmd: Possion Ratio of material

©ADML lab

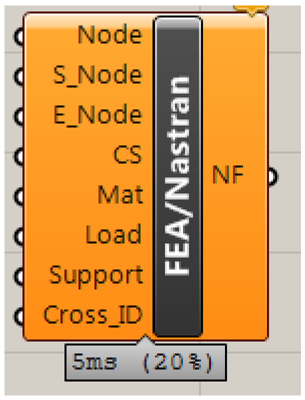
Density: Density of material

Output:

LOAD: Material properties

2.5 Script Generator

2.5.1 FEA/Nastran



Function: Generate script for Nastran solver

Input:

Node: Node list of lattice frame. This list can be generated by "Frame Organize" component.

S_Node: Start node index of struts. This list can be generated by "Frame Organize" Component.

E_Node: End node index of struts. This list can be generated by "Frame Organize" Component.

CS: Cross section. This input can be read from the cross section definition components.

Mat: Material properties. This input can be read from the material definition components.

Load: Load for the FEA analysis. This input can be read from the load definition components.

Support: Constraints. This input can be read from the constraints component.

Cross_ID: This input indicates the relationship between ID of struts and ID of the cross section. If you want to indicate the strut with ID equal to 5 has a cross property's ID equal to 2, you can input a string "5, 2". This can be also get value from the tool called LinkStrutToCross. By default (no input), all strut will be assigned with the cross section whose ID equal to 1.

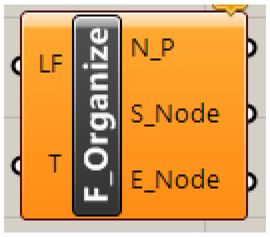
Output:

LOAD: Nastran script which can be used for FEA analysis

©ADML lab

2.6 Tools

2.6.1 F_Organizer (Frame Organizer)



Function: To organize the output of lattice frame and provide input for the FEA_Interface

Input:

LF: Line frame, which is used to define lattice frame

T: Tolerance. If the distance of two nodes is smaller than this value. These two nodes will be merged.

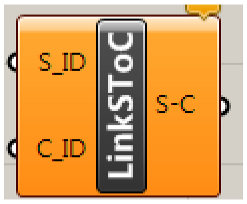
Output:

N_P: Node position list.

S_Node: Start node index list.

E_Node: End node index list.

2.6.2 LinkSToC (Link strut to cross section)



Function: To provide input for script generator to determine the relationship between strut and cross section

Input:

S_ID: Strut ID. This ID can be obtained by Component ”Strut selection by line”

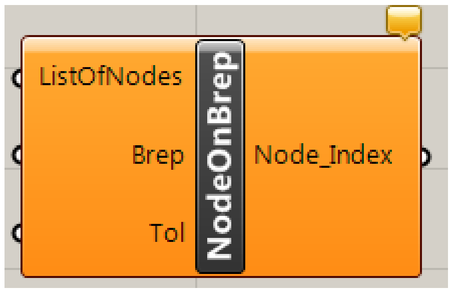
C_ID: Cross section ID

Output:

S-C: Strut-Cross section relationship can be regarded as the input for the script generator.

©ADML lab

2.6.3 Node On Brep



Function: Obtained node index of given Brep

Input:

ListOfNodes: List of Node position which can be obtained from the component Frame_Organize

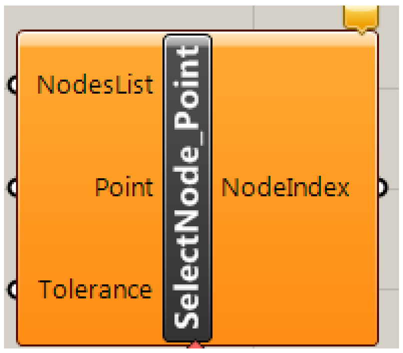
Brep: The brep where the node is on.

Tol: Tolerance, if the distance between the node and given brep is smaller than this value, the node is regarded on the Brep.

Output:

Node_Index: Indexes of selected nodes.

2.6.4 SelectNode_Point



Function: Obtained node index of given Brep

Input:

ListOfNodes: List of Node position which can be obtained from the component Frame_Organize

Point: Select the point which is at the position of the node

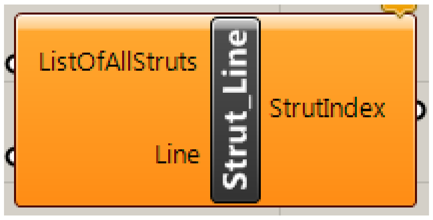
Tol: Tolerance, if the distance between node and selected point is smaller than tolerance value, the node will be selected.

Output:

©ADML lab

Node_Index: It is the indexes of selected nodes.

2.6.5 Strut_Line



Function: return the index of selected line

Input:

ListOfAllStruts: It is lattice frame which is generated from the lattice design module.

Line: Selected line.

Output:

StrutIndex: Index of selected line.

3 Example

In this section, we will give you an example to show how to use the provided components for FEA analysis.

3.1 Lattice generation

In this example, we first generate lattice with Intrallatice assembly which is shown on the Figure 2.

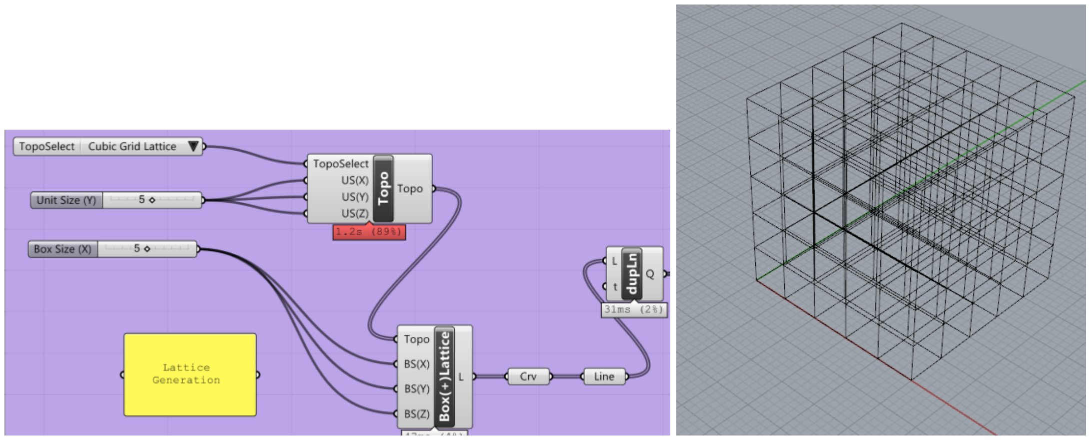


Figure 2 Lattice generation and frame view

©ADML lab

3.2 FEA Script generation

3.2.1 Frame organization

After we got the lattice frame. The next step is to organize the lattice frame. If the lattice frame has already been organized in the lattice frame generation component. This step can be skipped.

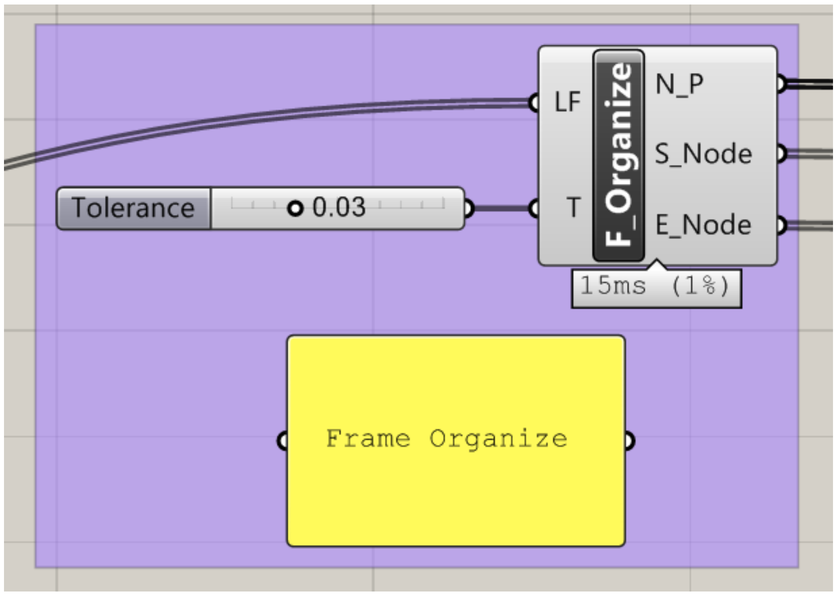


Figure 3 Frame organization

3.2.2 Material definition and cross section definition

Then, we need to define the material and cross section properties. First to decide the material properties. Select iso-material components and input the parameters and ID (shown on Figure 4). Then based on defined material properties we create two rod cross section (shown on Figure)

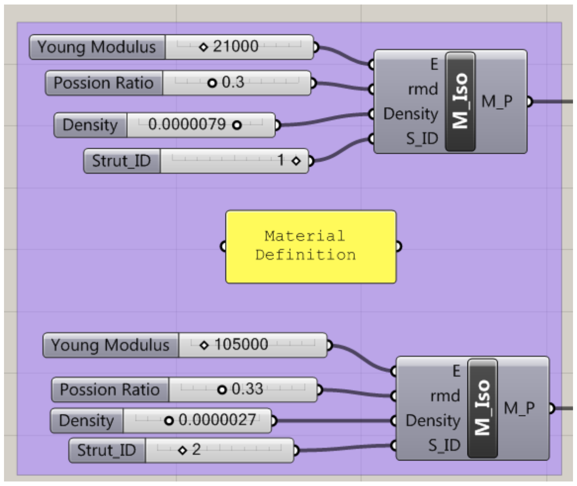


Figure 4 Material properties definition

©ADML lab

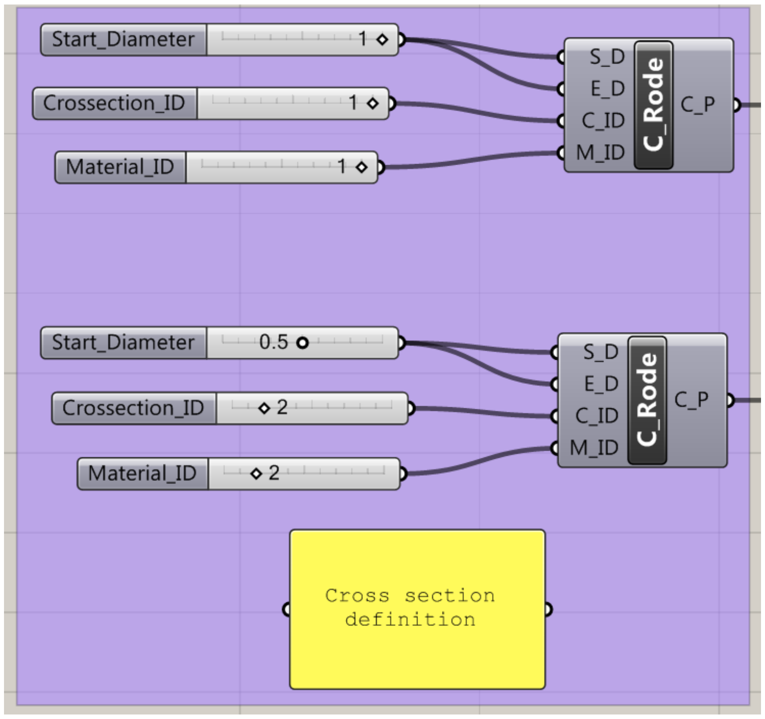


Figure 5 Cross section definition

3.2.3 Boundary Condition

Then, we need to define boundary condition. To define the boundary condition we need to first bake the nodes (shown on Figure 6) and Boundary brep. It had better to build different layer to bake different elements on Rhino.

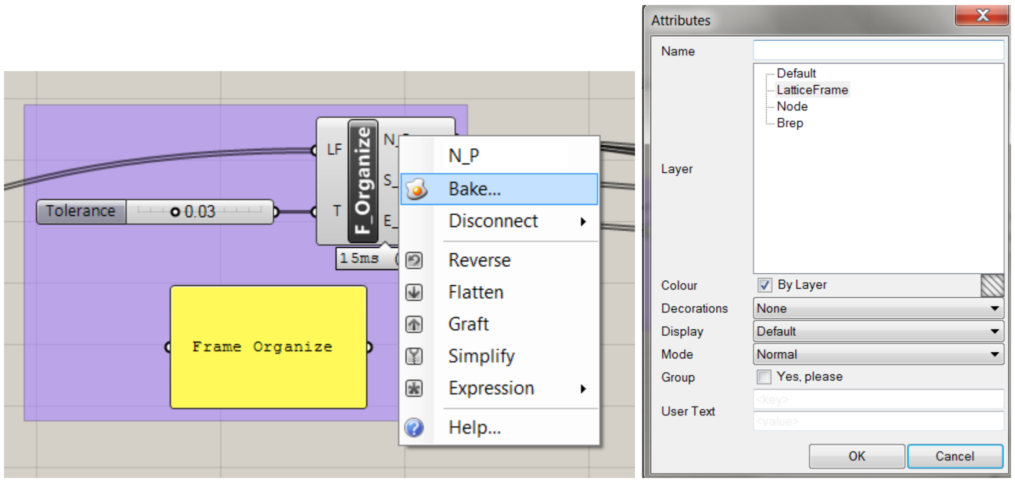


Figure 6 Bake node on Rhino

As to bake brep, it should be noted that we deconstruct the brep into surfaces. This is because we need we want to set the nodes on a single surface for constraints and loading.

©ADML lab

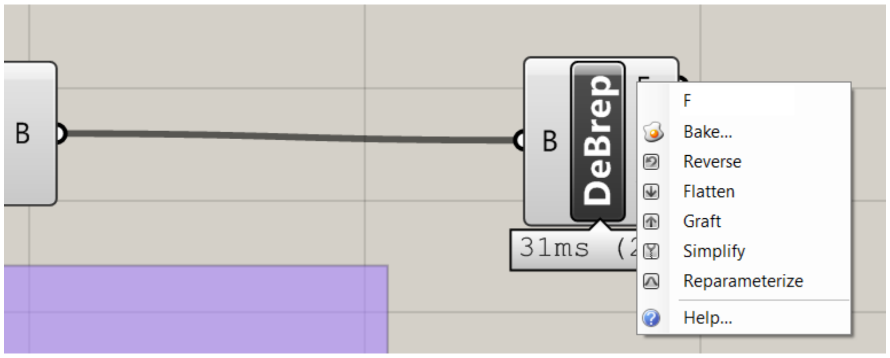


Figure 7 Bake decronstructed Brep

After we bake the deconstructed Brep, we can select the bottom of the box and use component called "NodeOnBrep" to find the indexes of nodes on this surface. The output of this component can be used as input for SPC component which is shown on Figure 8.

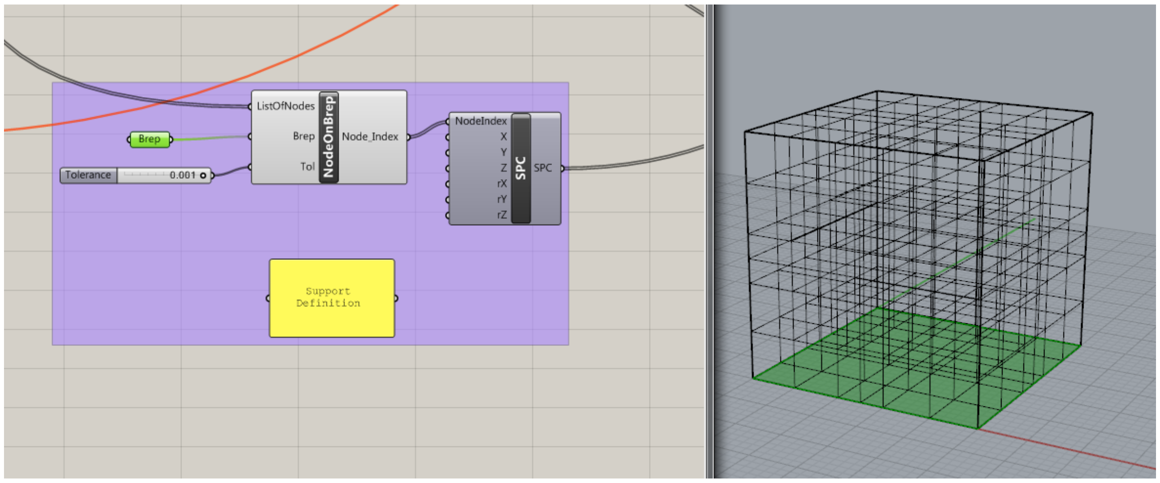


Figure 8 Define constraints

Then we can use the same method to define the load on the top of the box which is shown on Figure 9.

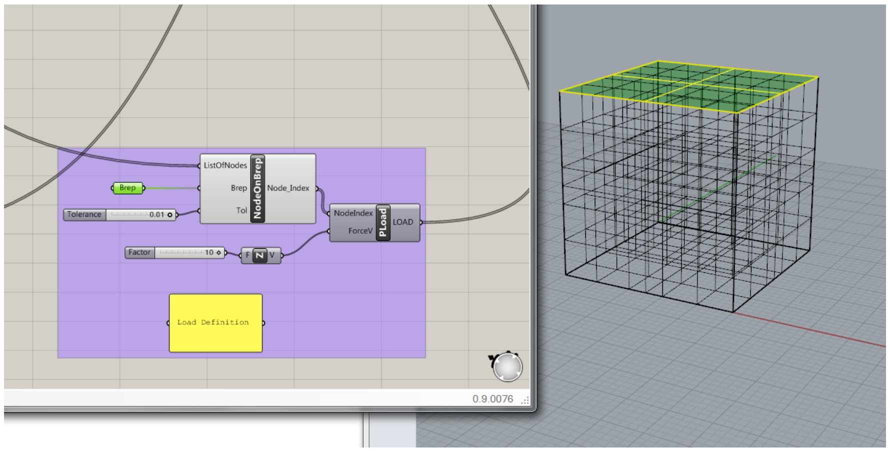


Figure 9 Define point load

©ADML lab

3.2.4 Define struts and cross section relationship

In this case we have two cross section types. In the default, all the lattice struts will use the cross section whose ID equals to 1. Thus, we need to define those struts with cross section 2. To do this, we use component "strut_line" to first select the strut we want to assign with cross section 2 which is shown in Figure 10. And then output those strut index into the component called "LinkSToC" to build a relationship between selected struts and defined cross section. The whole process is shown in Figure 11.

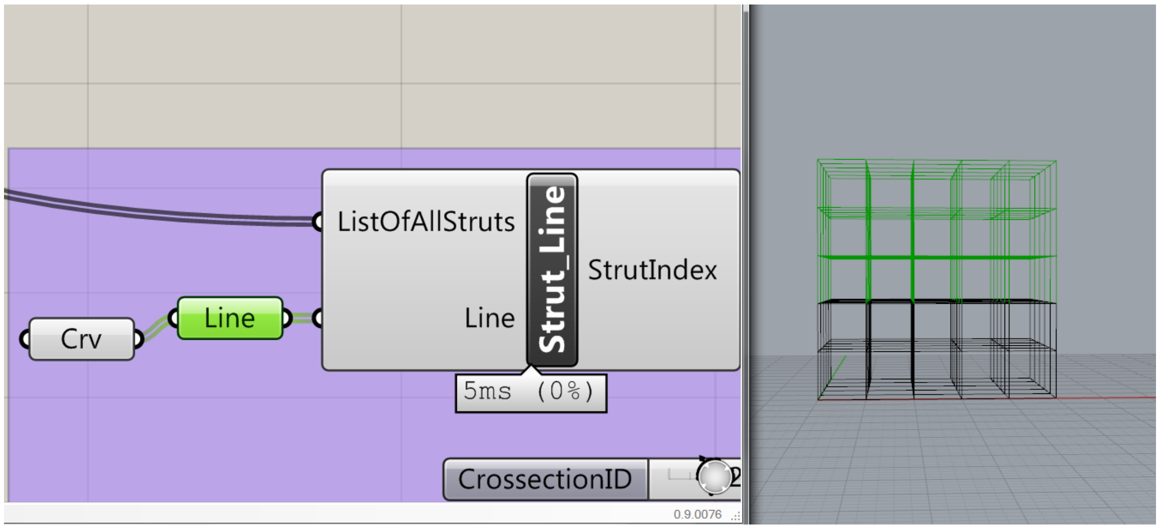


Figure 10 Select struts index by line

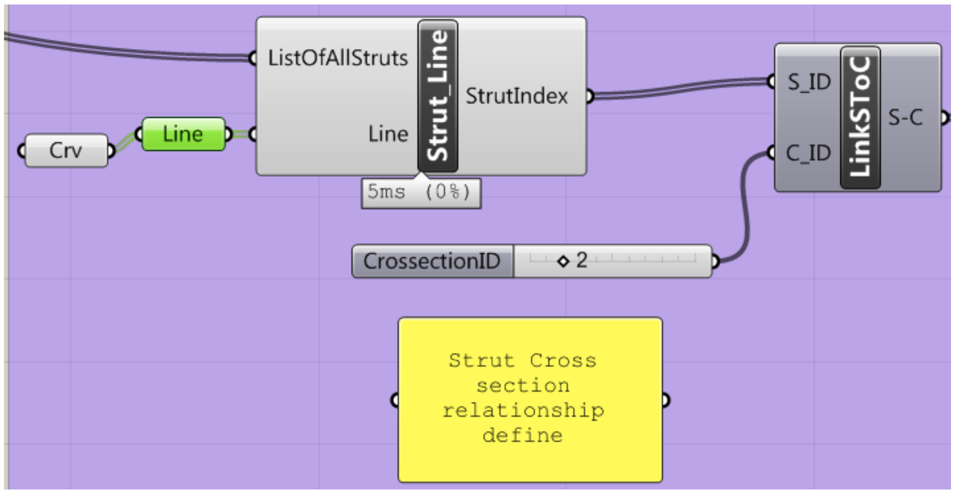
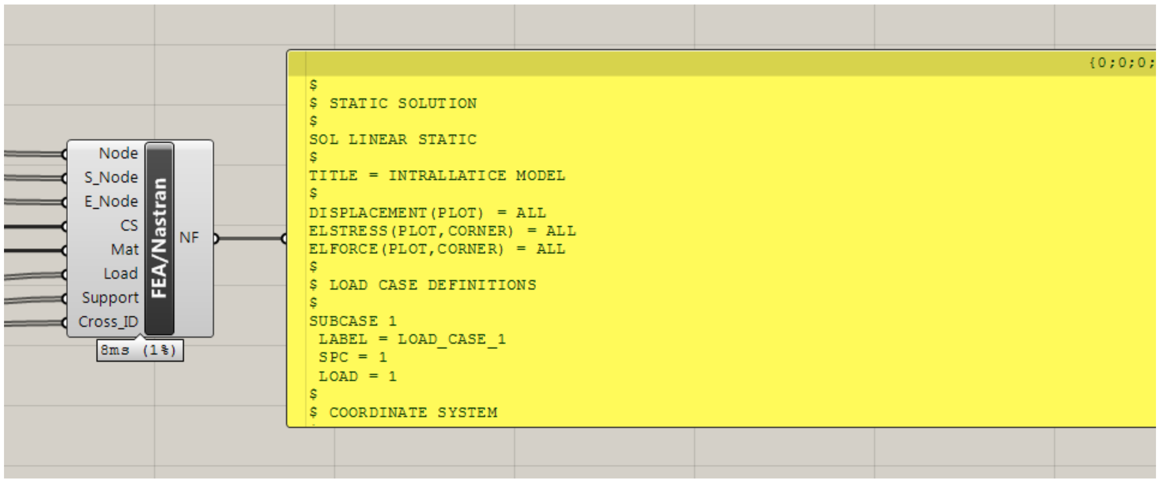


Figure 11 Define struts-cross section relationship

3.2.5 Generate Nastran script

The final step to generate FEA script is to input all the previously defined items into the script generator components shown on

©ADML lab



```
$
$ STATIC SOLUTION
$
SOL LINEAR STATIC
$
TITLE = INTRALLATICE MODEL
$
DISPLACEMENT(PLOT) = ALL
ELSTRESS(PLOT,CORNER) = ALL
ELFORCE(PLOT,CORNER) = ALL
$
$ LOAD CASE DEFINITIONS
$
SUBCASE 1
 LABEL = LOAD_CASE_1
 SPC = 1
 LOAD = 1
$
$ COORDINATE SYSTEM
```

Figure 12 FEA/Nastran script generator

There are two ways to save the script. The first way is copy and paste the script from the panel on the right hand side. Second, this component will automatically generate the script document called nastran.txt under path "Libraries\Documents". You only need to change its extension name to "*.nas" for running with Nastran solver.

3.3 SOLVE THE PROBLEM AND VIEW THE RESULT

In this sub-section, we are going to briefly talk about how to run the FEA analysis with output scripts.

In this case, we first open Auto-desk Nastran solver and then open the generate script file which is shown on the Figure 13.

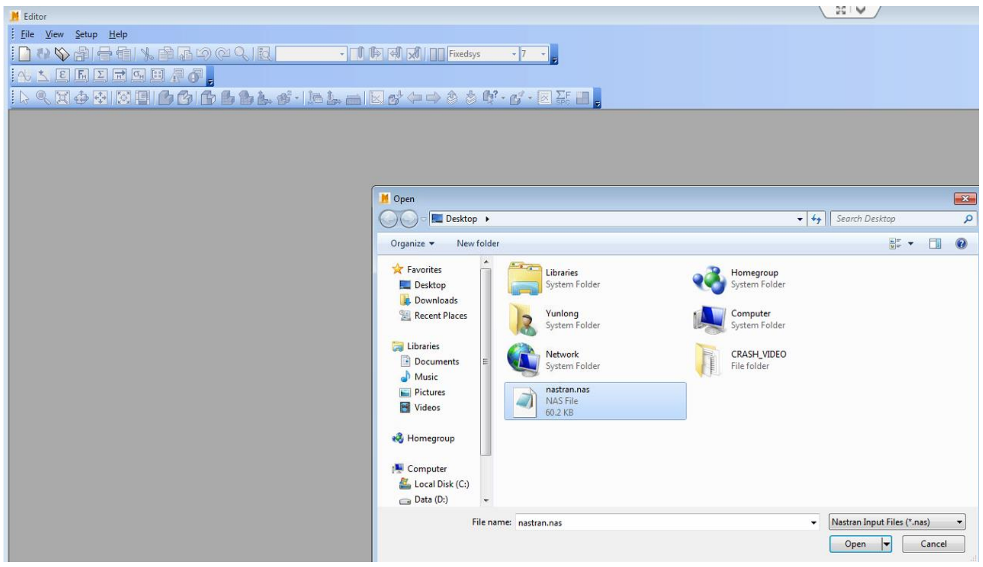
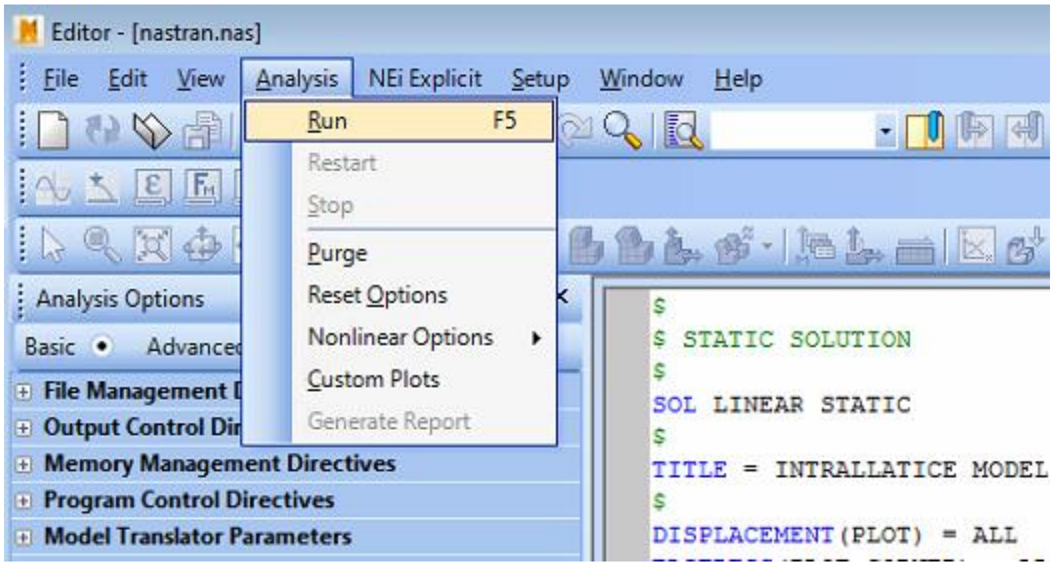


Figure 13 Open the file

Then we need to go to menu "analysis->run". Click run to solve the FEA problem.

©ADML lab



```
$
$ STATIC SOLUTION
$
SOL LINEAR STATIC
$
TITLE = INTRALLATICE MODEL
$
DISPLACEMENT(PLOT) = ALL
```

And the solver will be used to solve the defined problems. Then, we can go to the model results to view the result.

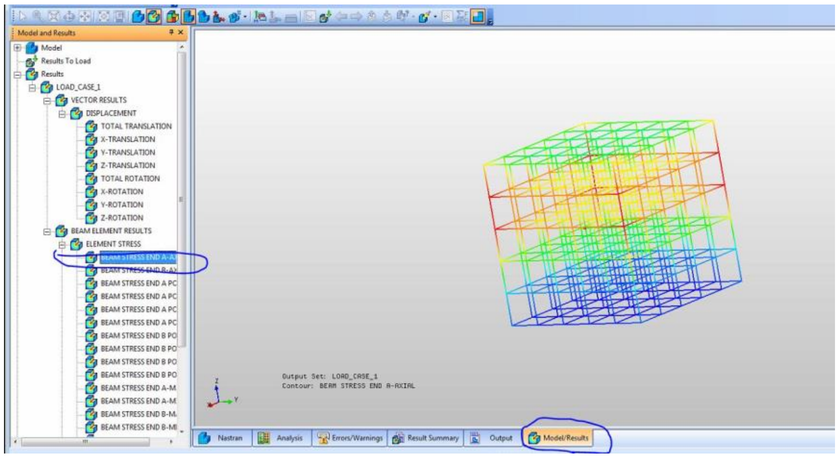


Figure 14 View the result

For the detailed use of Nastran solver, please reference the Nastran solver's user manual.

4 CONTACT INFORMATION

This component is developed by ADML lab, for any questions and feedback please contact Yunlong Tang. Email: tang.yunlong@mail.mcgill.ca.

©ADML lab