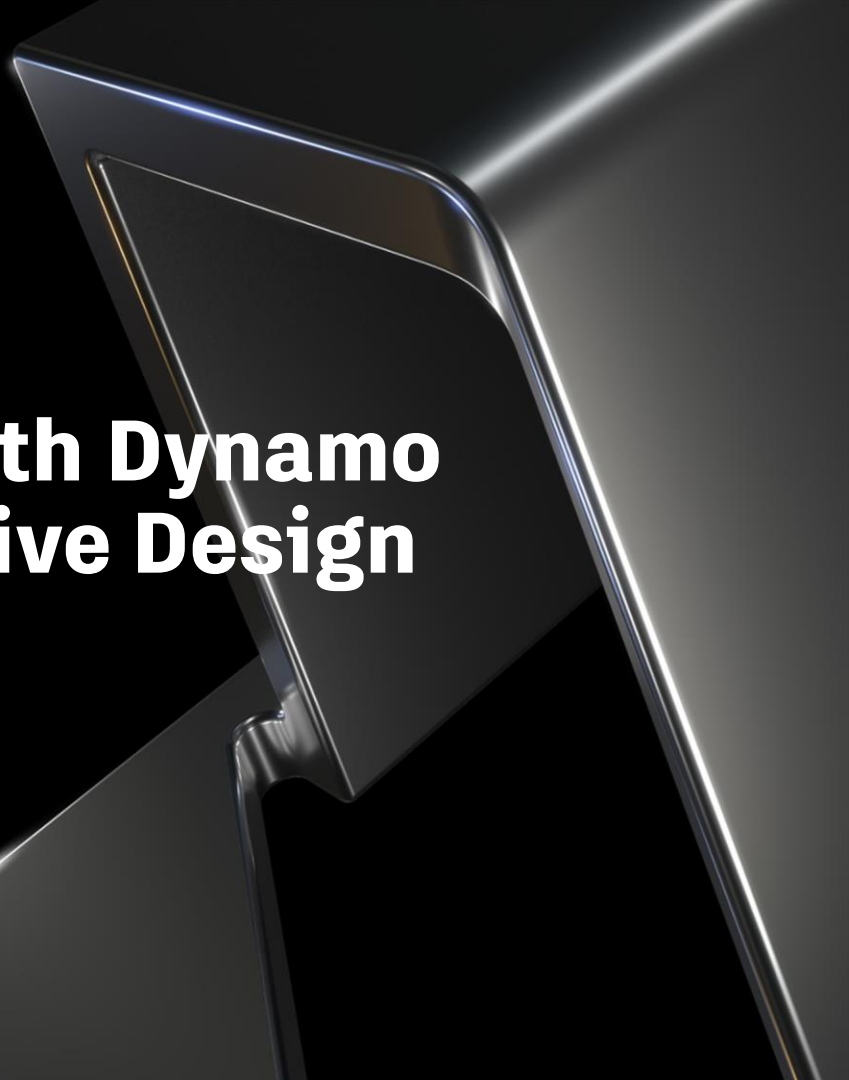
AUTODESK UNIVERSITY

Optimize road design with Dynamo for Civil 3D and Generative Design

Paolo Emilio Serra

Principal Implementation Consultant | @PaoloESerra



Paolo Emilio Serra

Principal Implementation Consultant

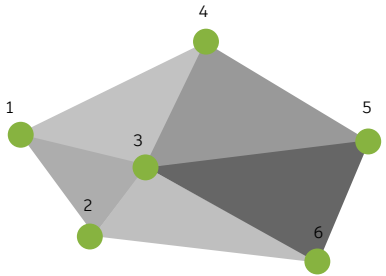
- 2008 – 2014 BIM Manager
- 2013 – Revit API & Dynamo
- 2014 – Autodesk Consulting
  - Automation & Dynamo
  - Linear Structures
  - Generative Design for AEC
  - DfMA + Industrialized Construction
- Contacts
  - paolo.serra@autodesk.com
  - @PaoloESerra
  - puntorevit.blogspot.com

Learning Objectives

- Define a computational design approach for road design
- Leverage generative design to optimize the design and increase the insights of the design challenge
- Automate the creation of corridor models
- Assess the next steps for new use cases and implementations

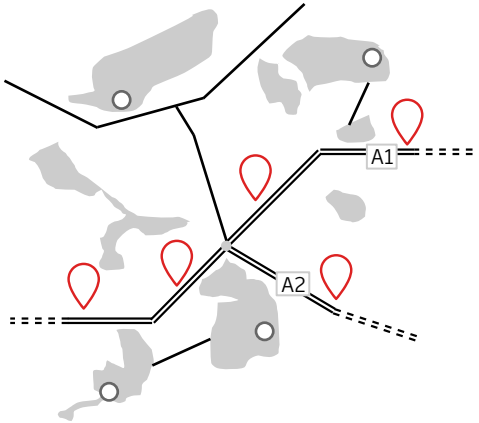
Key Concepts

Road Design | Inputs



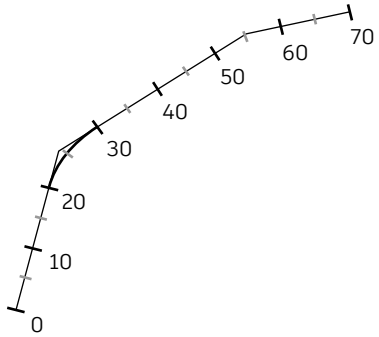
Surfaces

Obstacles
+
Boundaries

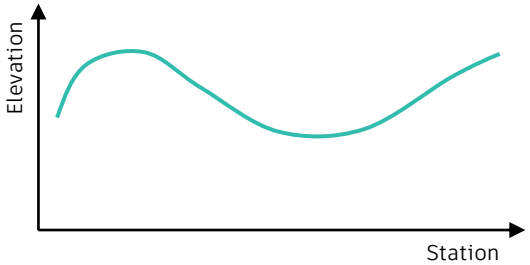


Key-Locations

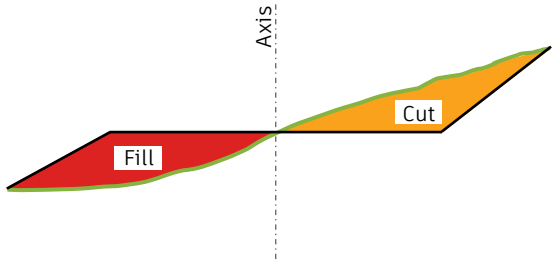
Road Design | Outputs



Horizontal Alignments

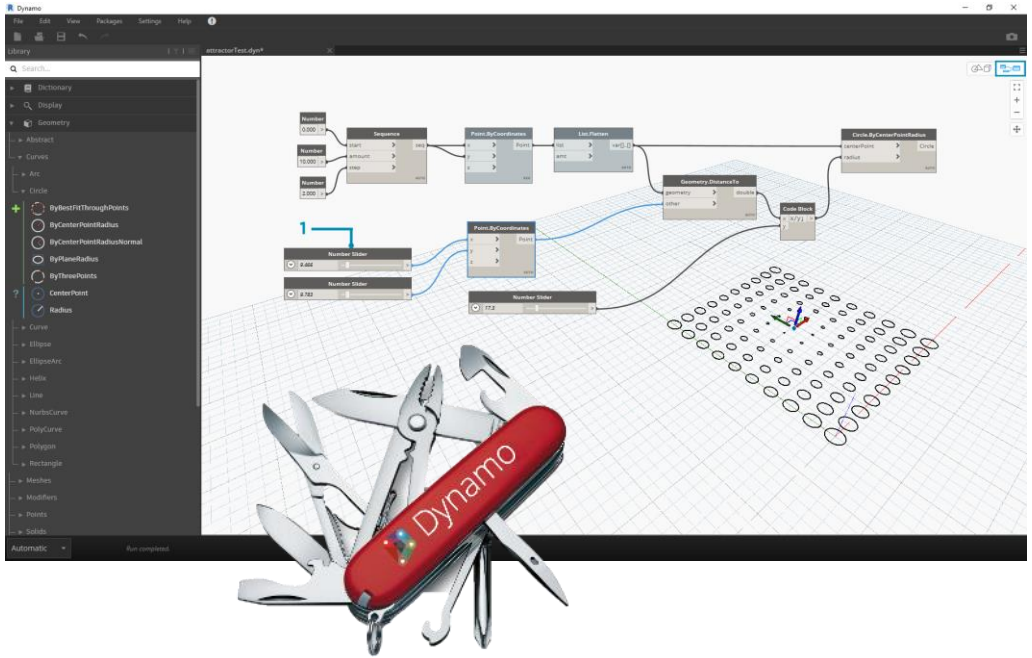


Vertical Profiles



Cross-Sections

Computational Design



C++

C# .NET
Python

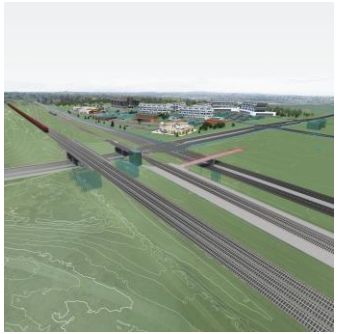
Dynamo

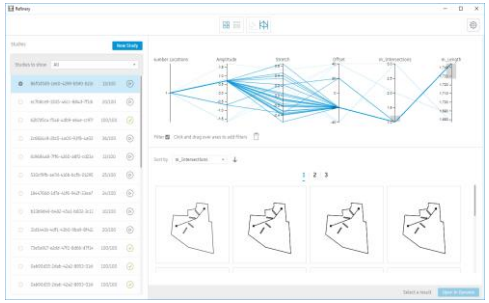
Interactive
Tools

Pencil +
Clay

CES322249

Design Optimization





Evolve

GDiR

Gather Data → Generate → Evaluate → Select

Defining Model

Computational Strategy

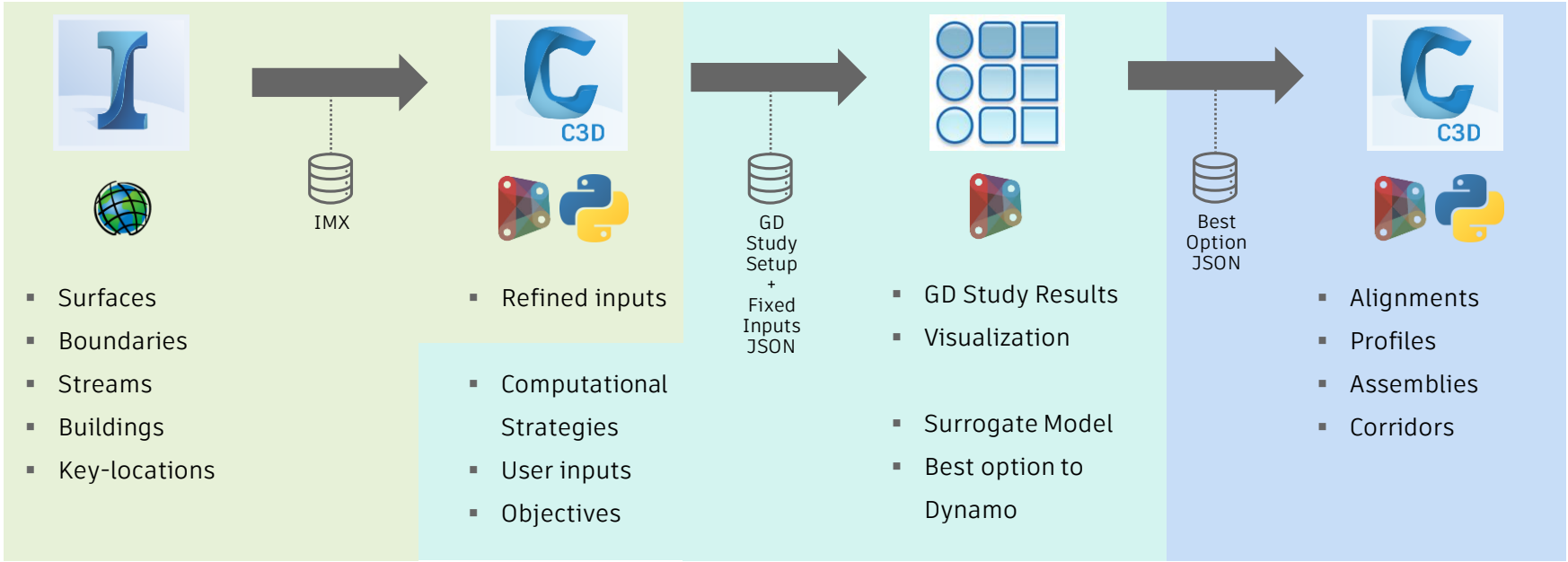
Objectives Scores

Data Driven

BES471869

Generative Design Workflow

Data Flow


IMX
C3D
GD Study Setup + Fixed Inputs JSON
Best Option JSON
C3D
Surfaces
Boundaries
Streams
Buildings
Key-locations
Refined inputs
Computational Strategies
User inputs
Objectives
GD Study Results
Visualization
Surrogate Model
Best option to Dynamo
Alignments
Profiles
Assemblies
Corridors
Inputs
Generative Design
Automation

Approach

Dynamo enabled workflows

- **Serialize** | Capture C3D fixed inputs
  - Data.Remember / JSON
  - It should run in Dynamo Sandbox

- **Optimize** | Use GDiR to generate options
  - Define user inputs + objectives
  - Select best option > Data.Gate / JSON

- **Finalize** | Create Corridors in C3D
  - Best option > Civil Objects

Setup Generative Design

- Computational Strategies
  - Minimum Spanning Tree > Alignments
  - Reduce # of triangles > Bounding Boxes
  - Cellular Automaton > Profiles

- Objectives
  - # of clashes [-]
  - Cut/Fill balance [-]
  - Length [-]
  - Visibility [+]
  - Top Surface [-]

BES471869 AS473693-L

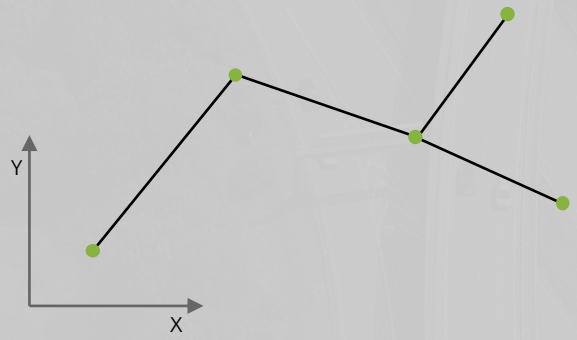
How To…

Alignments

How To...

Create alignments

- The Key-Locations (KLs) are defined in XY plane, these are nodes to connect
- Calculate Minimum Spanning Tree (MST)
- The “tree” represents multiple alignments connecting the KLs

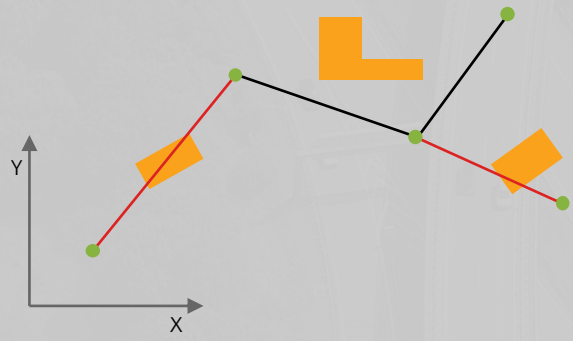


Minimum Spanning Tree

How To...

Create alignments

- In a real scenario, the MST may fail due to obstacles in the way of the branches

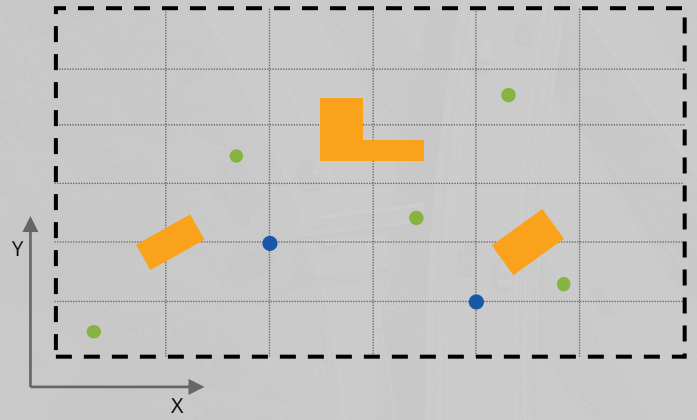


The minimum spanning tree may fail with obstacles

How To...

Create alignments

- A possibility is to add more nodes to the KLs and compute a new MST
- A simple option is to create a grid under the control of the designer
- The GD study can control how many and which nodes to add for the computation of the MST



Adding extra nodes sampling the space

How to create the extra nodes?
Designer input for grid, fixed during the study

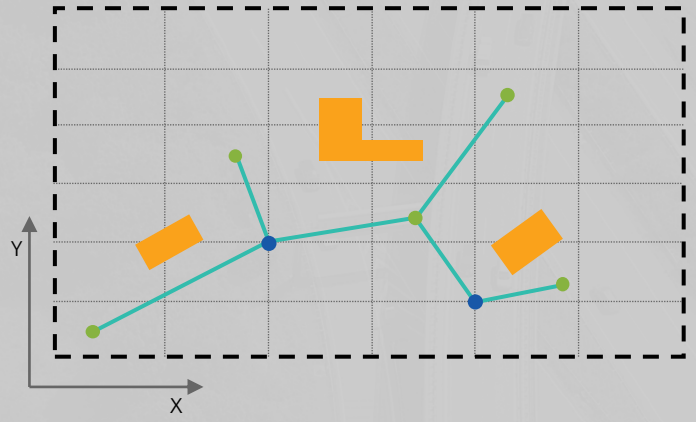
How many nodes to add?
Variable for the GD Study

Which nodes to add?
Variable for the GD Study

How To...

Create alignments

- Sample nodes within the boundary
  - I: XY grid sizes
  - V: # of points
  - V: which points, selector function
- Add extra nodes to key-locations
- Calculate Minimum Spanning Tree
  - In case of branching edges, create multiple alignments
- Add fillet for curves

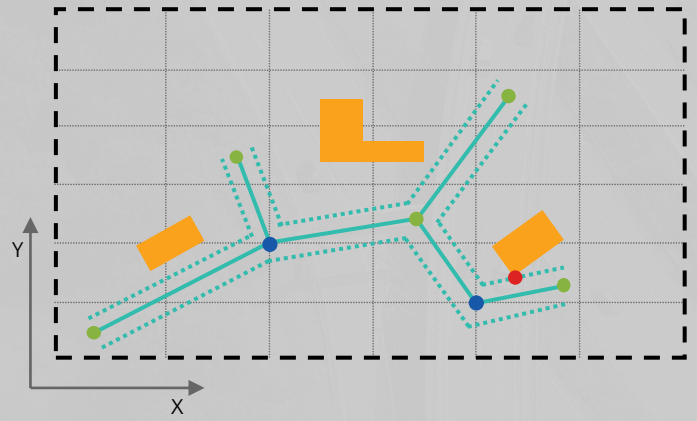


Calculate a new minimum spanning tree

How To...

Create alignments

- Consider a buffer to check for intersections with obstacles and count them
- The objective is to minimize their number
- It is OK to have a logic that ALSO produces imperfect solutions
- The goal is to let Generative Design find the optimum within the study parameters
  - Next iteration, refine input grid



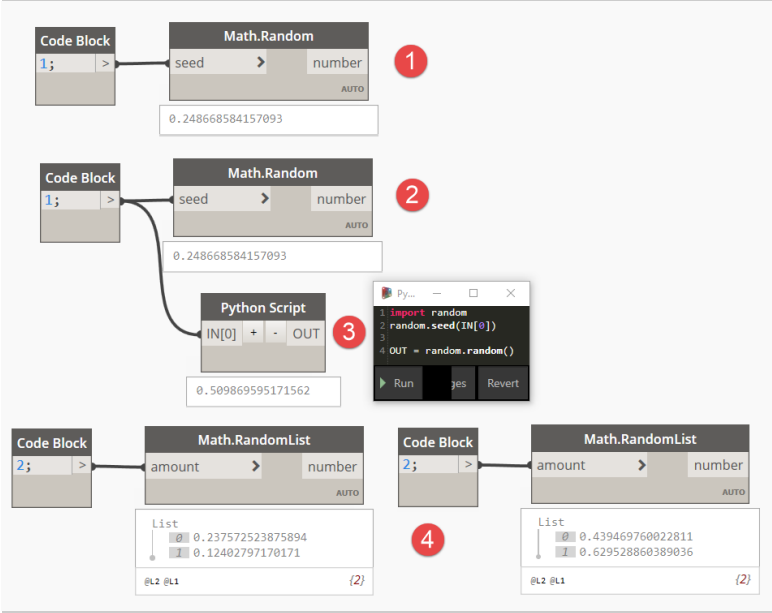
Calculate Intersections with a buffer

Selector Function

Selector Function

Random Generators Have Limits

- How to generate a dynamic number of variables
- Random number generators are OK for exploration but NOT for optimization
- Clear relationship between input and output
  - If small changes to input produce small changes to output, the algorithm can narrow down intervals for optimum input values
  - It needs to be consistent to be able to reproduce the values

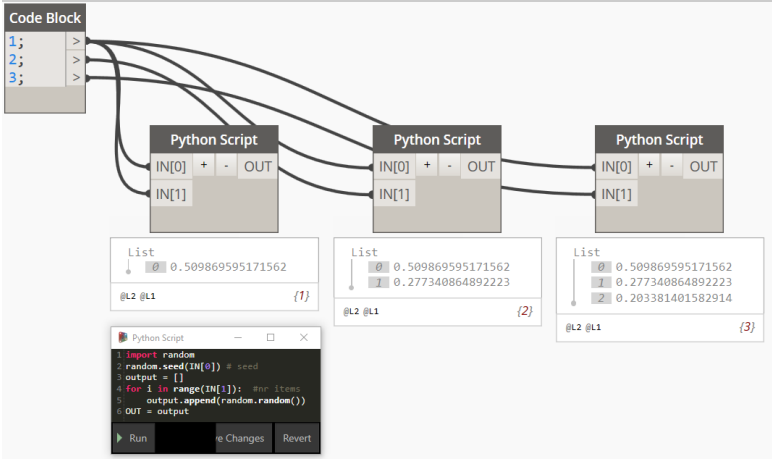


4. Dynamo Random List are not consistent

Selector Function

Random Generators Have Limits

- How to generate a dynamic number of variables
- Random number generators are OK for exploration but NOT for optimization
- Clear relationship between input and output
  - If small changes to input produce small changes to output, the algorithm can narrow down intervals for optimum input values
  - It needs to be consistent to be able to reproduce the values

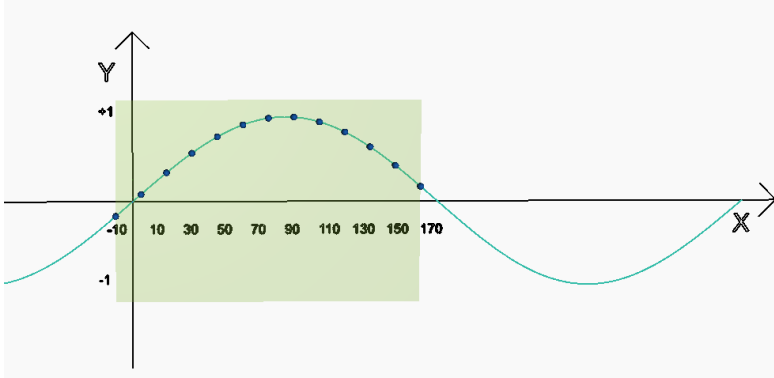


Python random numbers are consistent but there is no clear relationship between inputs and outputs (by design)

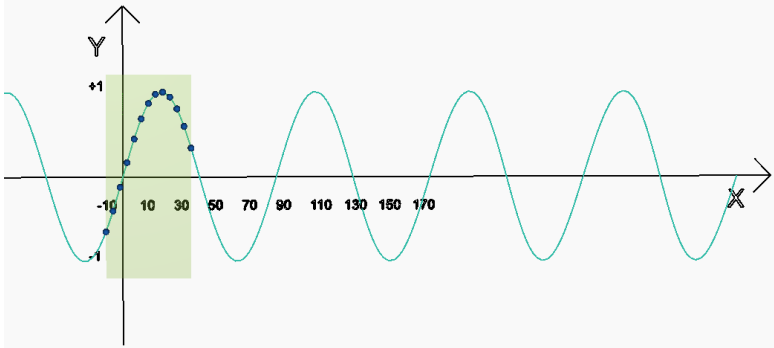
Selector Function

Oscillating function

- Limited number of variables for GD study
  - A: Frequency Coefficient
  - B: Start Angle
  - C: Range Width
- Produces a dynamic set of numbers with only 3 variables



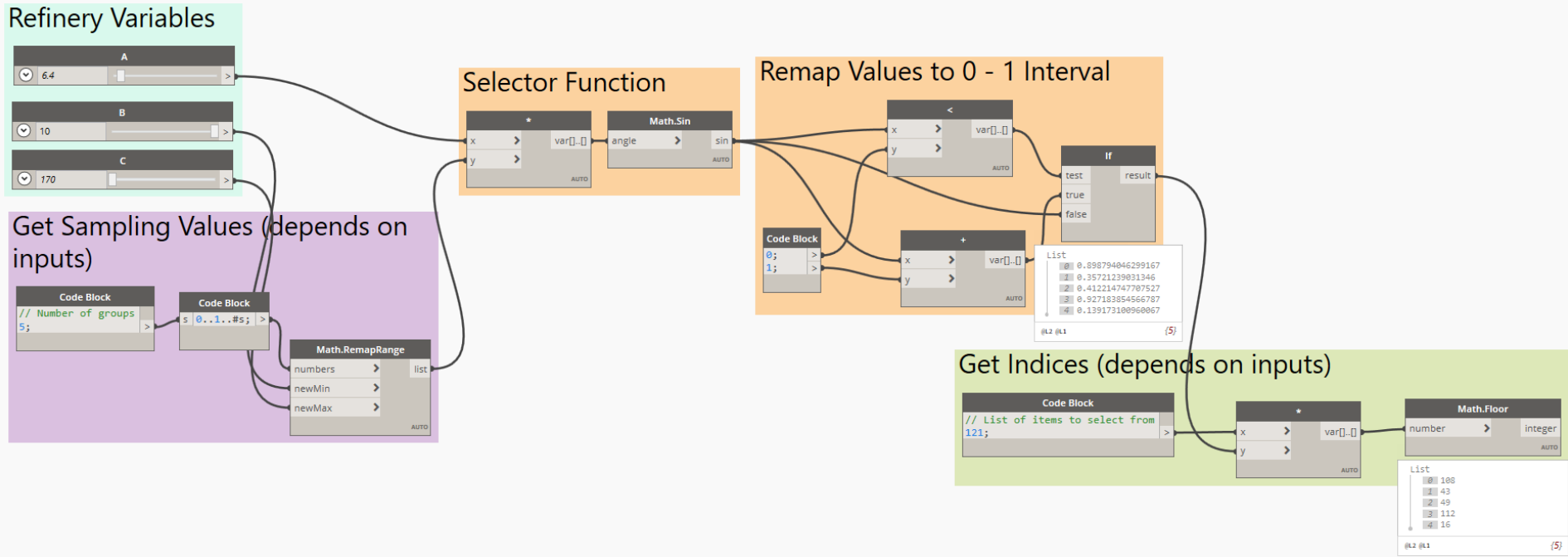
A=1, B = -10, C = 180



A=4, B = -10, C = 50

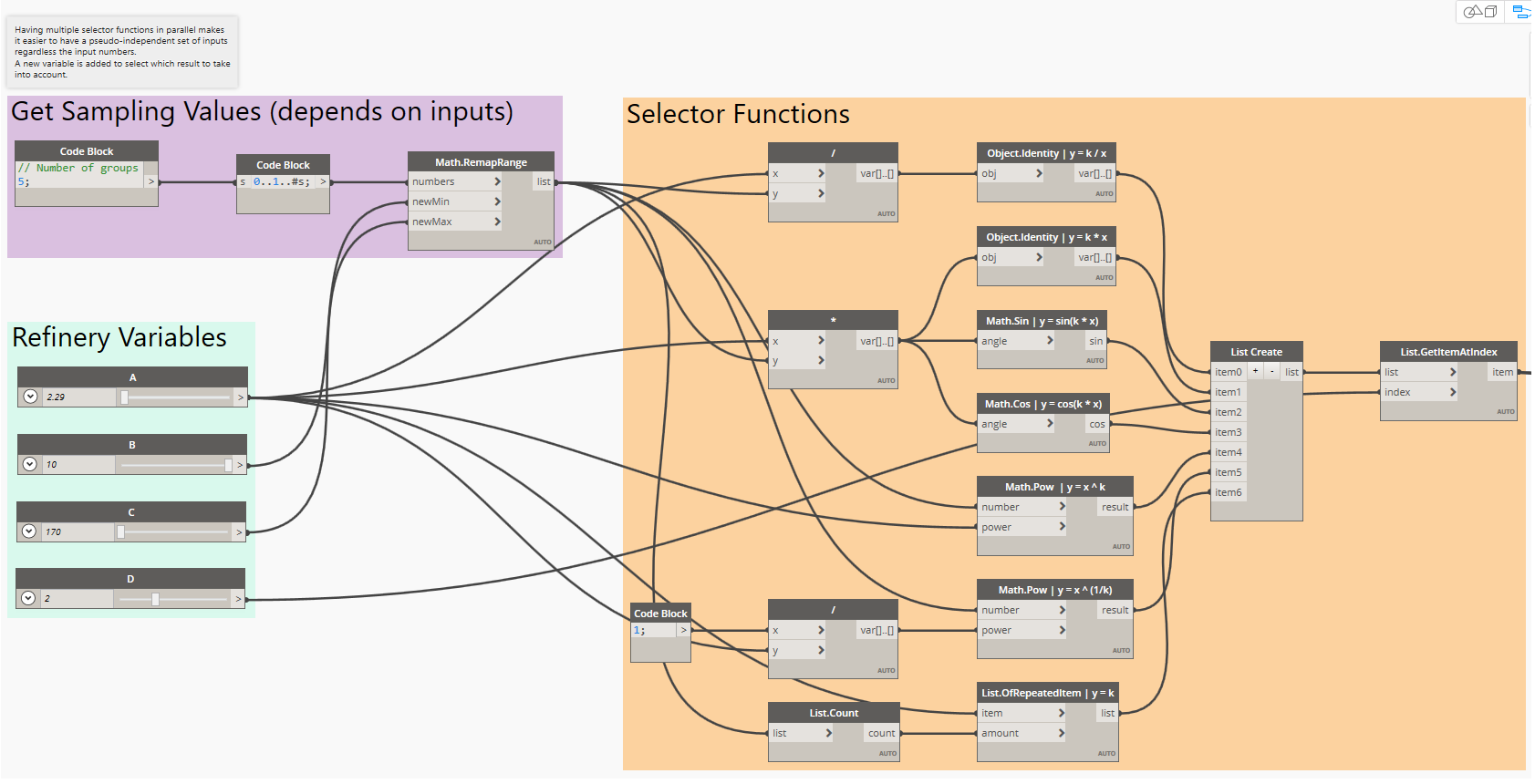
Selector Function

Oscillating function



Selector Function

Freedom to experiment which function works best

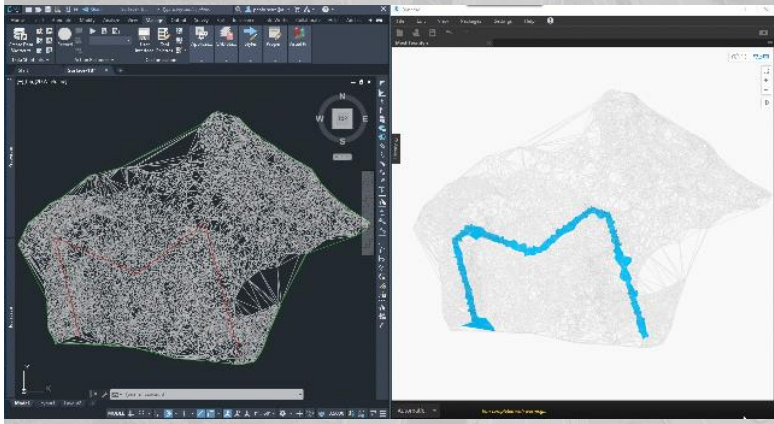


Surfaces

How To…

Deal with LARGE surfaces and A LOT of triangles

- Extract terrain mesh triangles
- Define a reasonable “buffer” for alignments
  - Create curve by offsetting PolyCurve
  - Extrude curve > buffer Solid
- Select only triangles intersecting buffer Solid
- Create an optimized terrain surface
  - PolySurface
  - Recommended to use recursion
- Use Python or custom nodes to improve performance

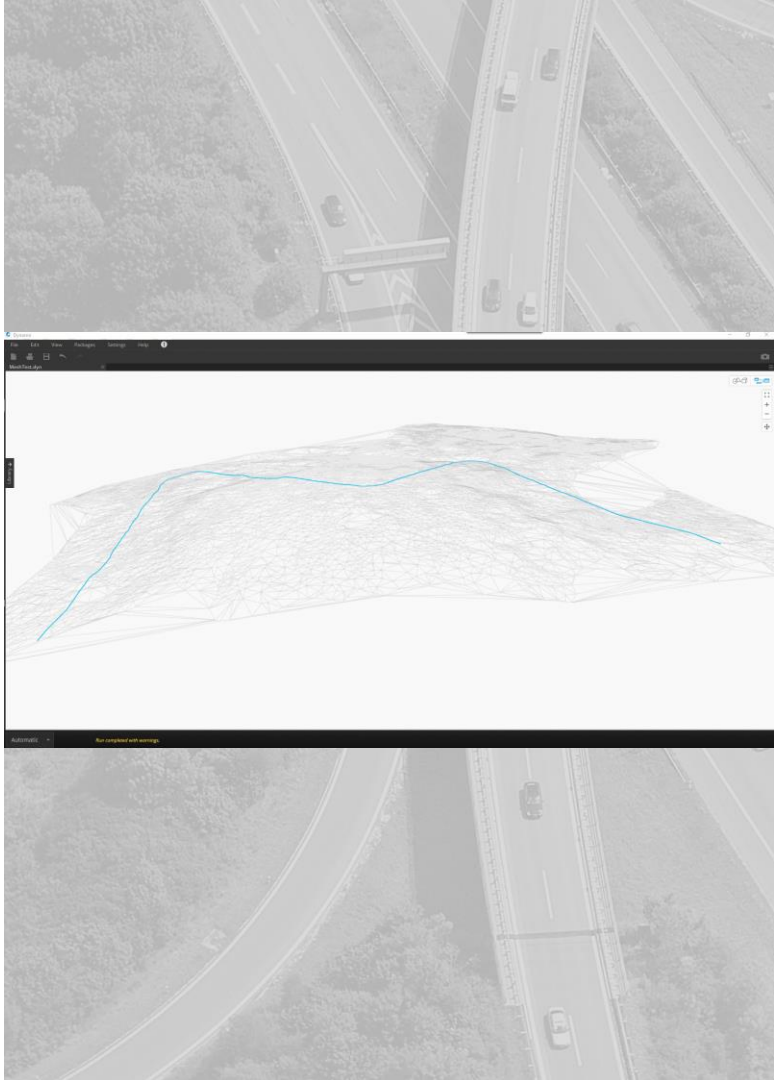


Profiles

How To...

Get EG profile without using Civil 3D nodes

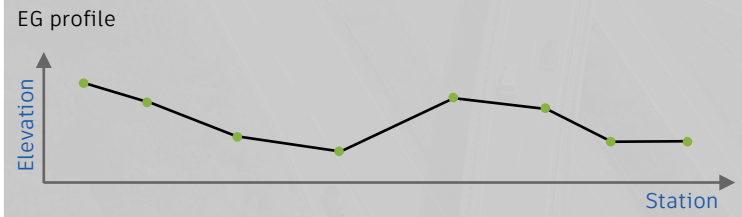
- Get optimized terrain PolySurface
- Extrude Alignment along Z axis
- Find intersections with PolySurface
- Join segments into PolyCurve



How To...

Create a design vertical profile

- Project EG Profile Points onto Alignment curve
- For each Point
  - Station = Alignment.ParameterAtPoint(p) * Alignment.Length
  - Elevation = Point.Z



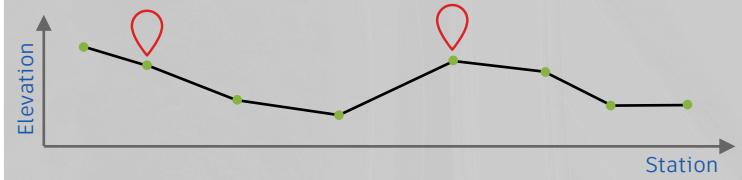
How To...

Create a design vertical profile

- Key-locations (K-L) stations and elevations on EG profile

EG profile

Add Key Locations



How To...

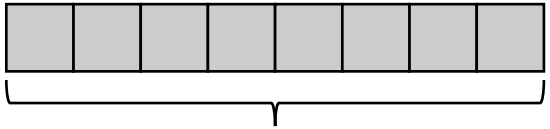
Create a design vertical profile

- Key-locations (K-L) stations and elevations on EG profile
- Apply design constraints
  - e.g., K-L station ± 50.0m constant elevation
- For intermediate station ranges
  - Opt. 1: sample points on EG profile
    - Adds variables to the GD study
  - Opt. 2: apply Cellular Automaton
    - Calculates best option following the rules

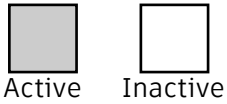
How To...

Implement a cellular automaton for profiles

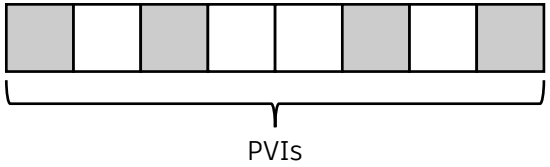
EG Profile



PVI States



Design Profile



Rules

- Max slope In/Out
- Max slope delta In/Out
- Minimum visibility distance

EG profile, all PVIs are active

How To...

Implement a cellular automaton for profiles

EG Profile

PVI States

Active Inactive

PVIs

Design Profile

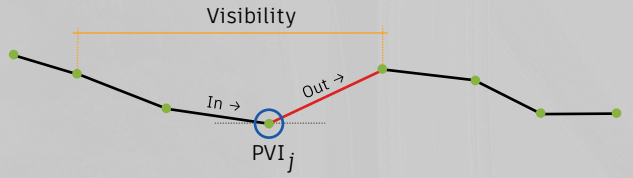
PVIs

Rules

- Max slope In/Out
- Max slope delta In/Out
- Minimum visibility distance

EG profile, all PVIs are active

For each PVI



Out Slope > Max slope

Slope In – Slope Out > Max slope delta

→ PVI_j state will change

How To...

Implement a cellular automaton for profiles

EG Profile

PVI States

Active Inactive

PVIs

Design Profile

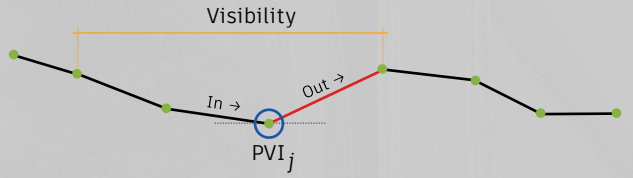
PVIs

Rules

- Max slope In/Out
- Max slope delta In/Out
- Minimum visibility distance

EG profile, all PVIs are active

For each PVI



Out Slope > Max slope

Slope In – Slope Out > Max slope delta

→ PVI$_j$ state will change

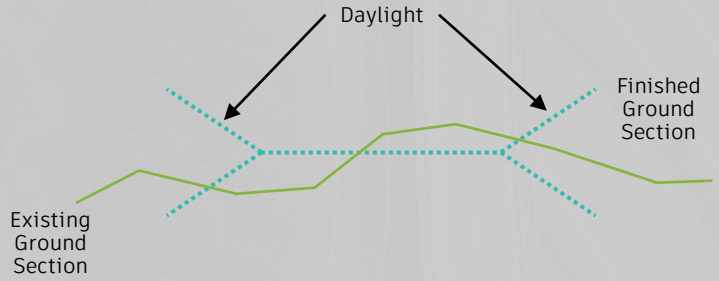
Repeat until best profile option is found:
Nr. slope failures and slope changes [-]
Slope failures and slope changes intensity [-]
Overall visibility [+]

Cross-Sections

How To...

Generate cross-sections

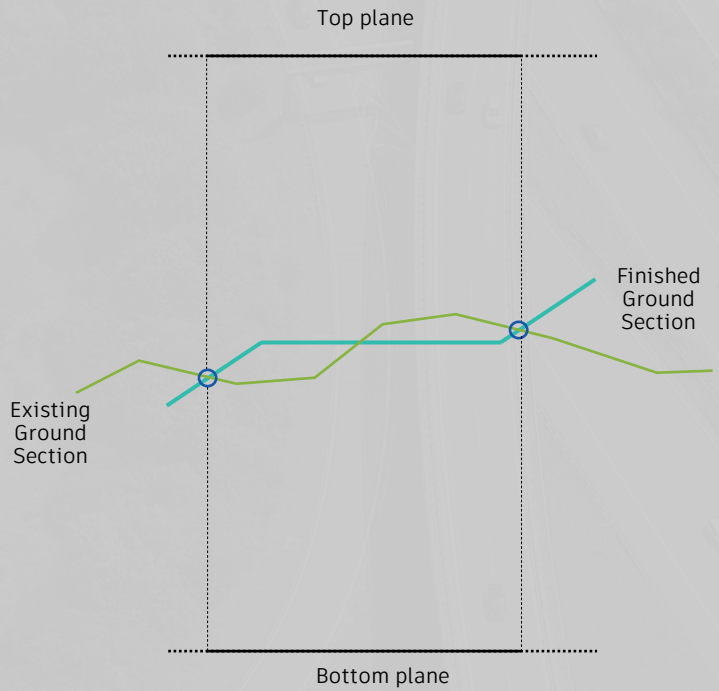
- Create road cross-section geometry (XSG) in the origin (including daylight)
- Get Coordinate Systems (CS) on Alignment PolyCurves, adjust elevation as per design profile (define frequency and add geometry stations)
- For each CS transform the XSG
  - Finished Ground



How To...

Generate cross-sections

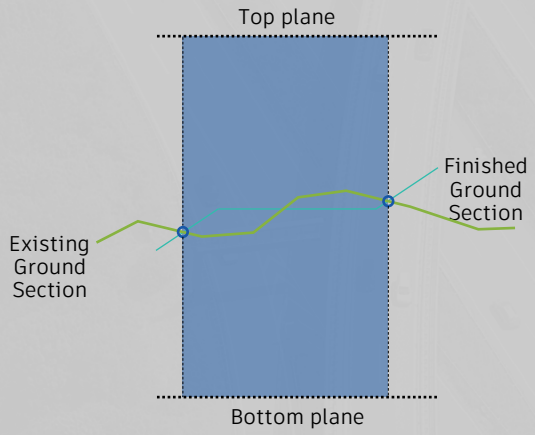
- For each station calculate the leftmost and rightmost intersection points between Existing and Finished ground
- Define two arbitrary horizontal planes above and below the surfaces (top and bottom)
- Project the intersection points onto the top and bottom planes
- Create lines connecting the projections on top and bottom planes



How To...

Generate cross-sections

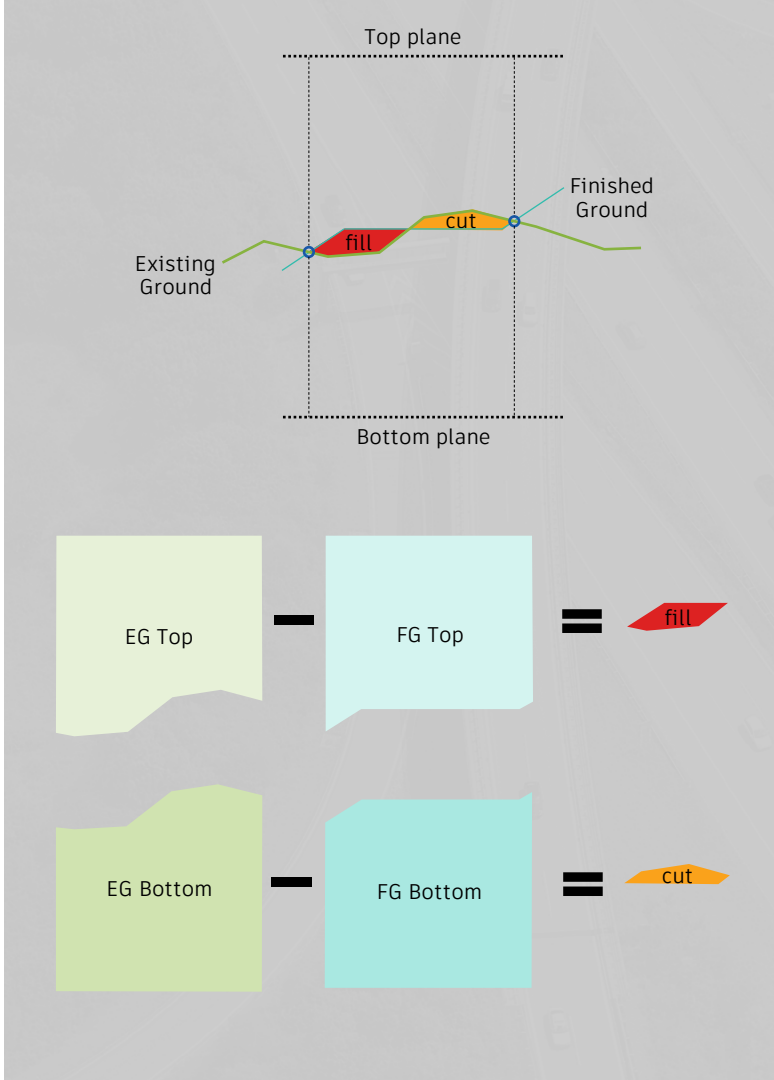
- Create a loft surface between top and bottom lines
- The surface can be split in two
  - EG Top, EG Bottom with Existing Ground
  - FG Top, FG Bottom with Finished Ground



How To...

Generate cross-sections

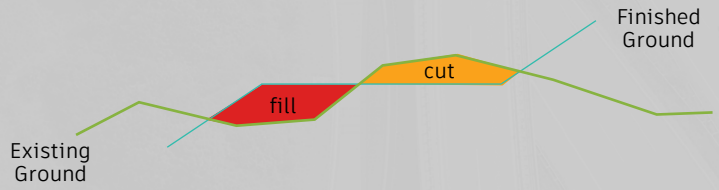
- The difference between the Top surfaces returns the Fill surface in the station
- The difference between the Bottom surfaces returns the Cut surface in the station



How To...

Generate cross-sections

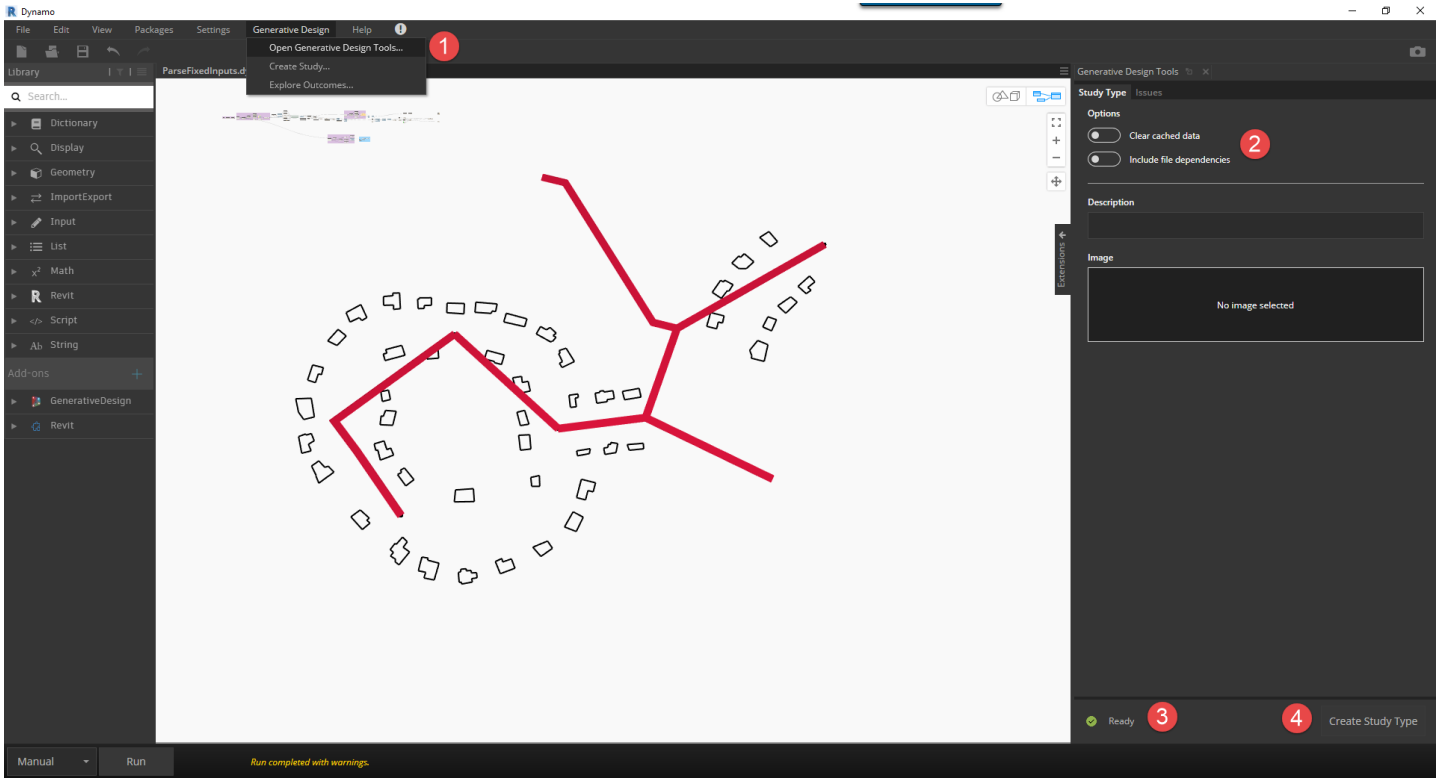
- Cumulate the areas of cut and fill for all cross-sections
  - Total Cut [-]
  - Total Fill [-]
  - Math.Abs(Total Cut – Total Fill) [-]
- Loft the Finished Ground Cross-Sections
  - Total Top Area [-]



Optimization

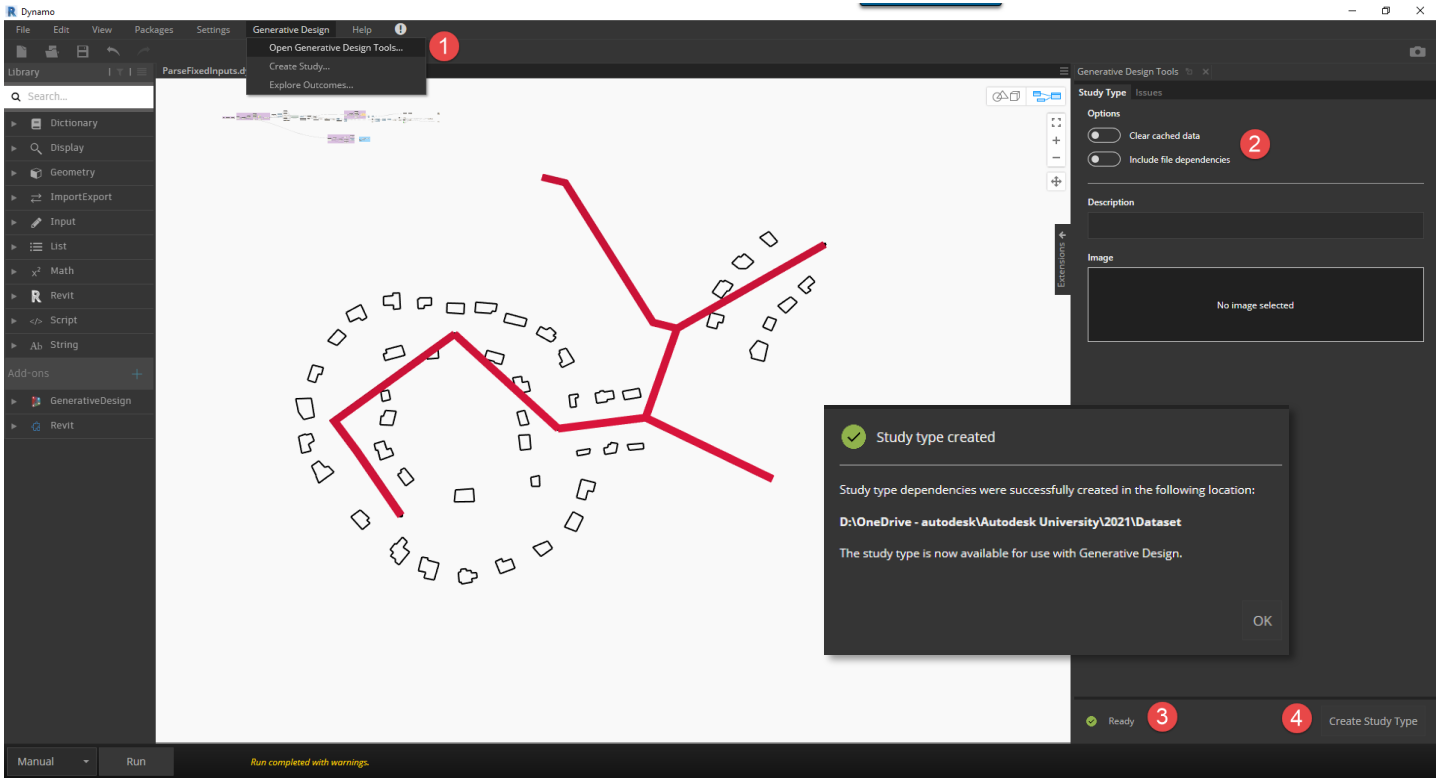
Create Study Type

Generative Design in Revit



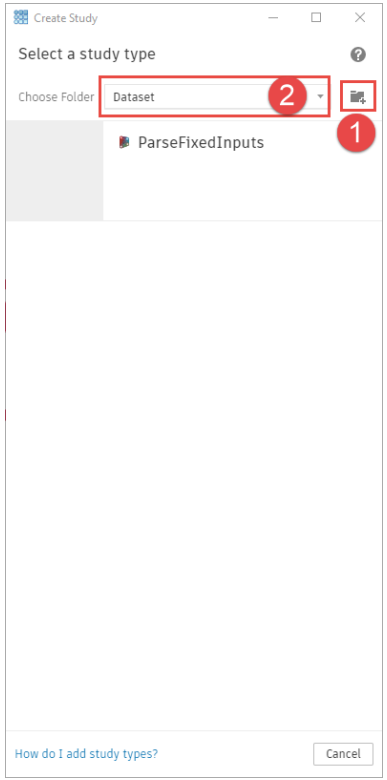
Create Study Type

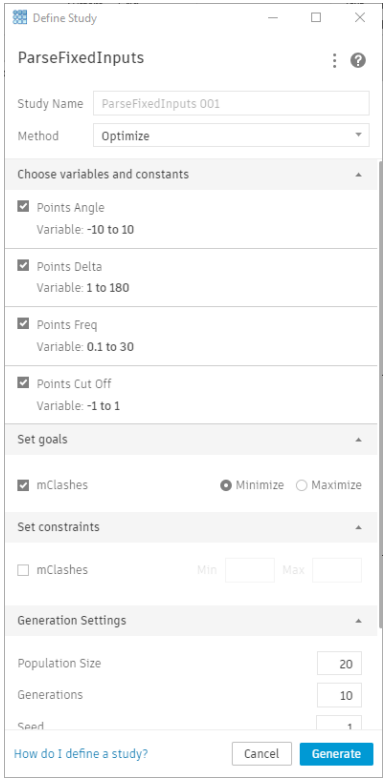
Generative Design in Revit

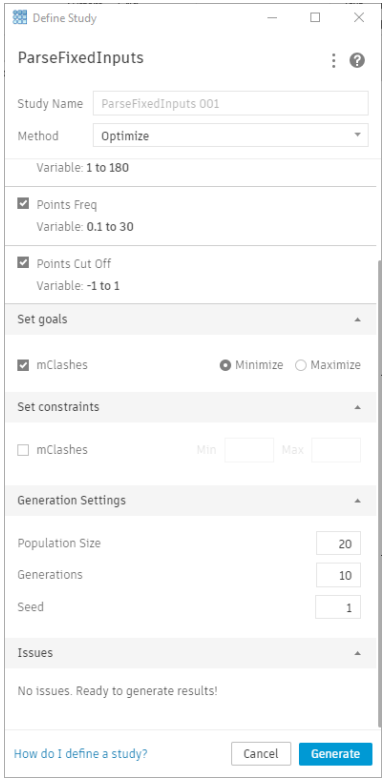


Define Study

Generative Design in Revit



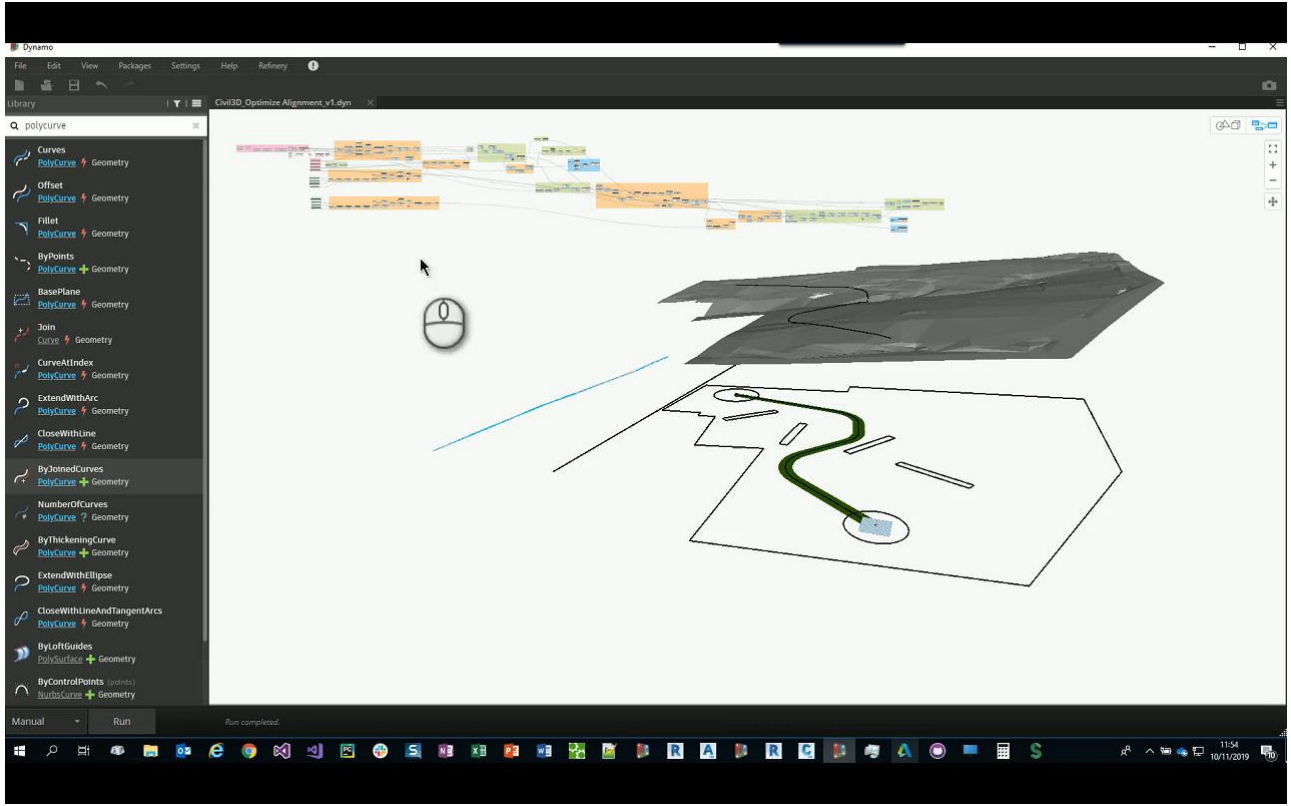




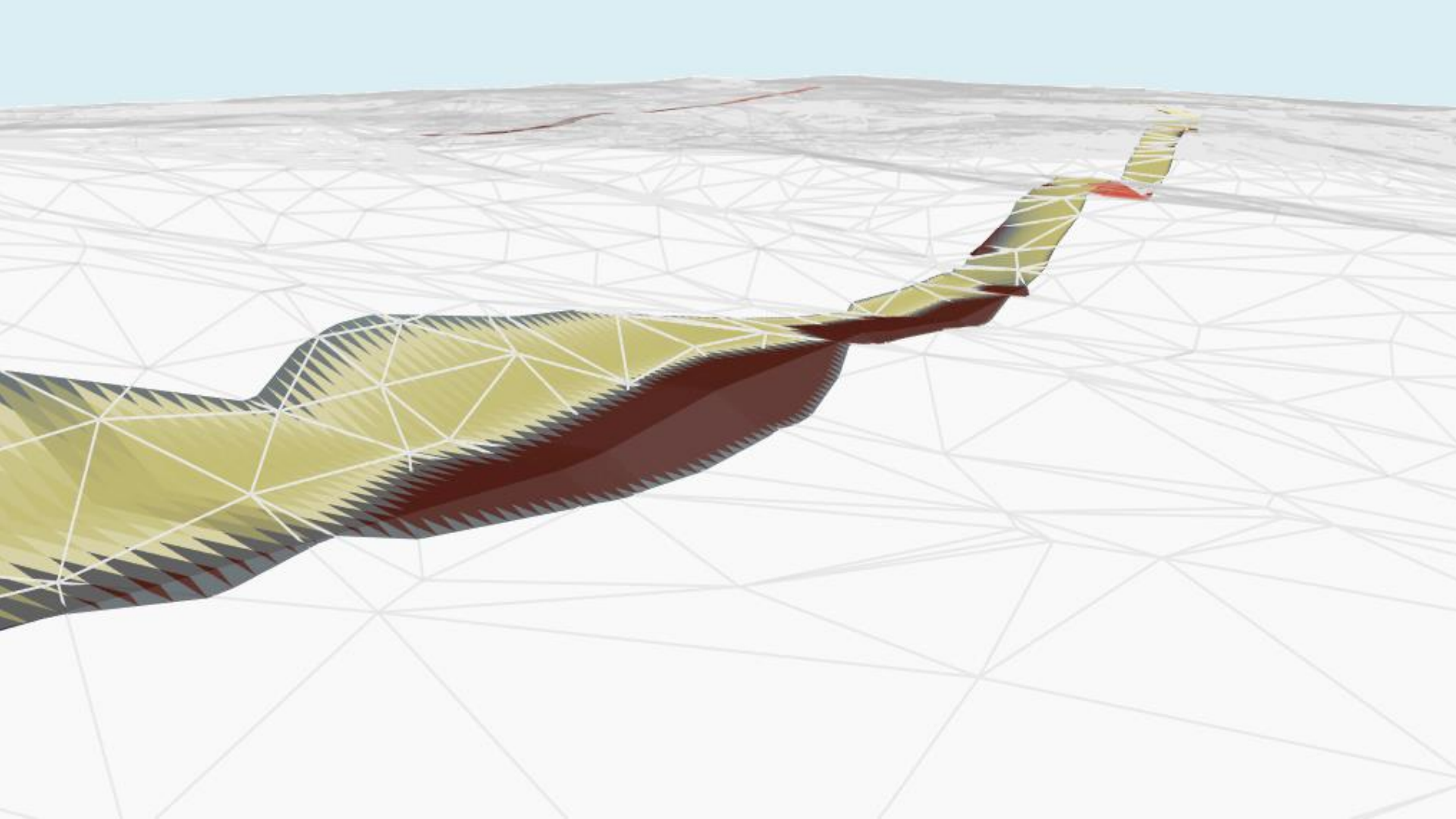
Explore the results

Explore Results

Data Driven Design



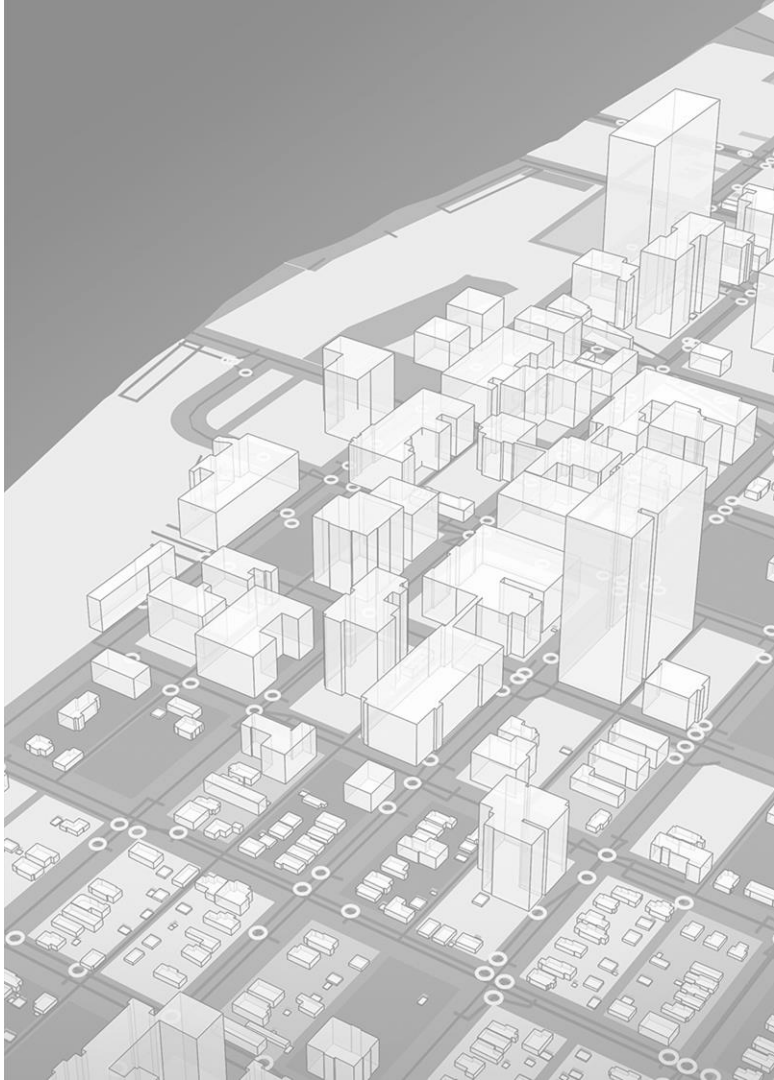
[Link To Video]



Read & Write JSON

Save the result of a computation

- Save the results at the end of a computation
- Use the results as input for a different graph
- OOTB Dynamo
- Python
  - Customized Dictionary



Automation

Corridor Automation

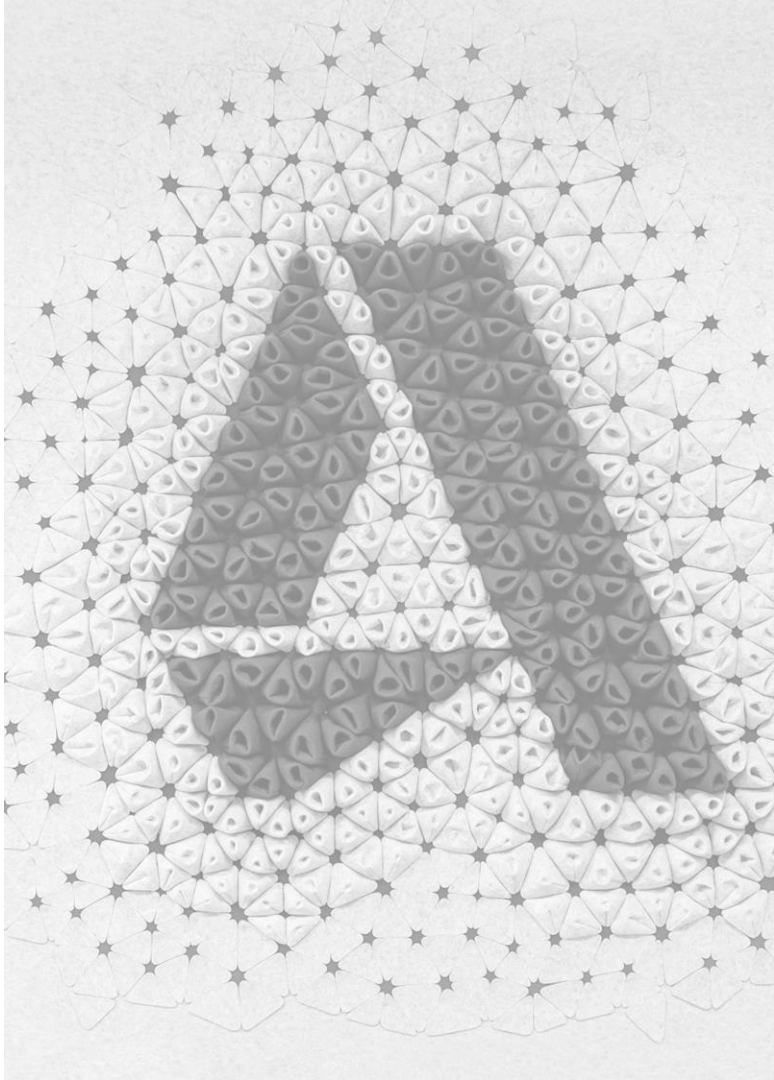
Dynamo enabled workflow

- Read Best Option from JSON
- Civil 3D Toolkit
  - Create Alignments
  - Create Profiles
  - Create Assemblies (* or create manually)
  - Insert Subassemblies (* or create manually)
  - Create Baseline
  - Create Baseline Regions
  - Set Targets
  - Rebuild Corridor

Conclusions & Next Steps

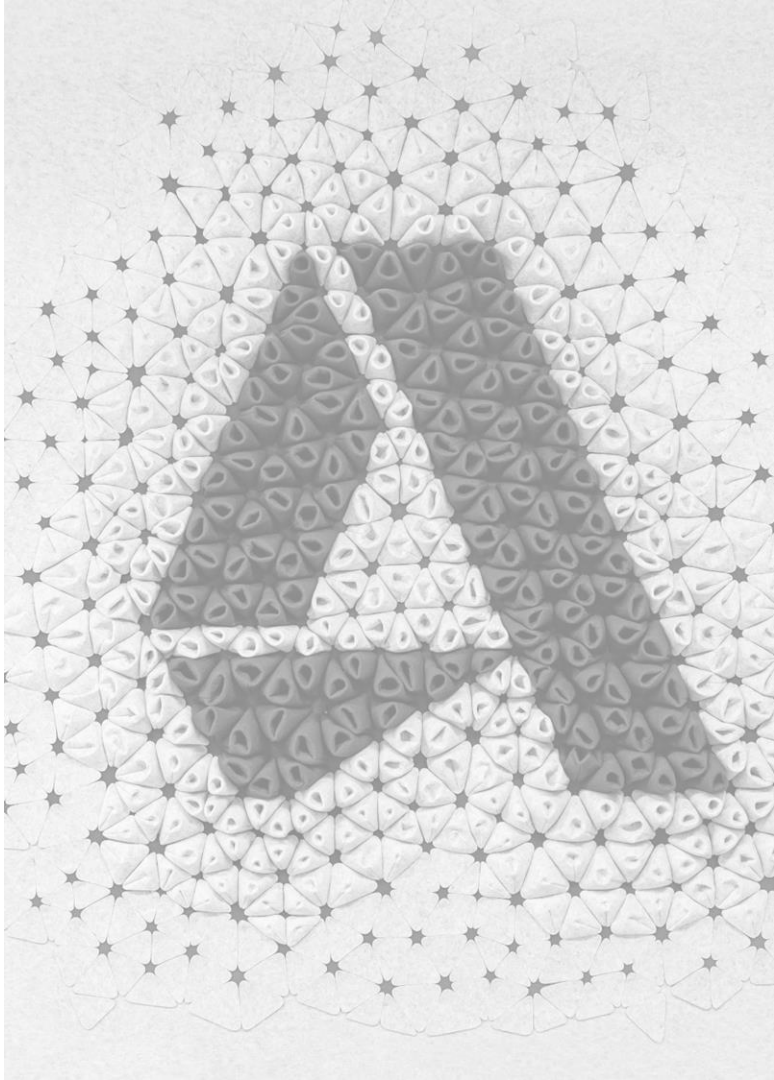
Conclusions & Next Steps

- Performance depends on
  - Number of triangles
  - Frequency of cross-sections
- Alignments
  - Add curves with variable radii
  - Add superelevation
- Profiles
  - Add curves



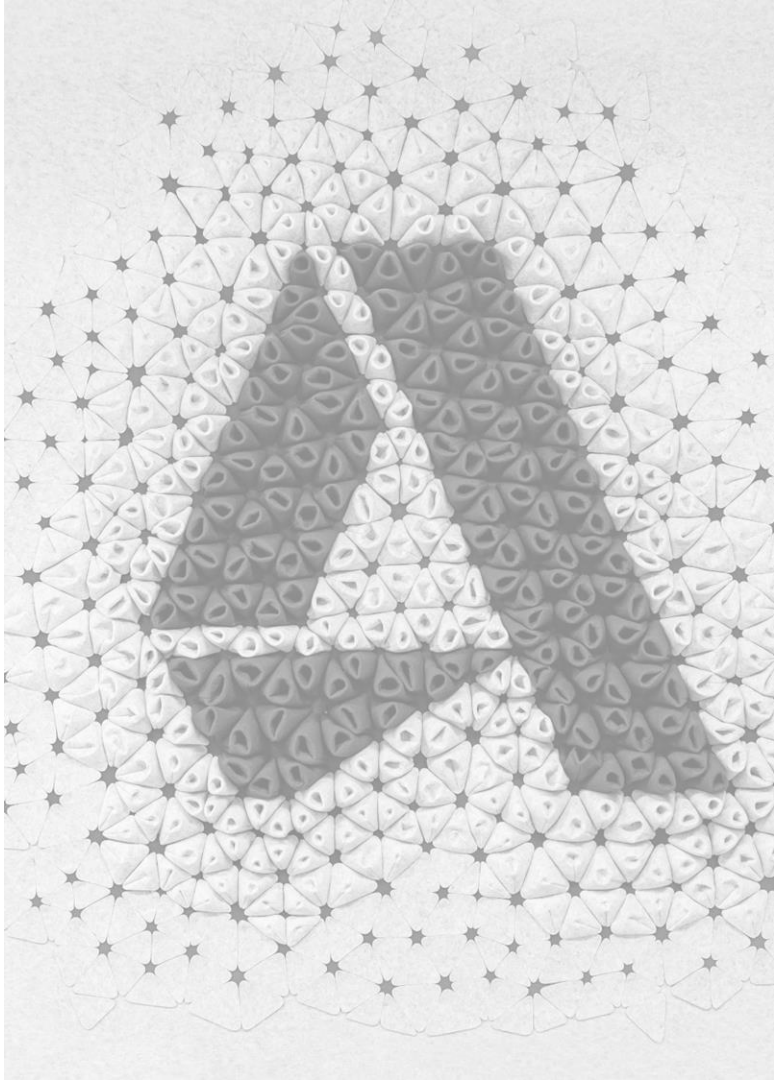
New Use Cases

- Analyze the problem to solve and change the strategies accordingly
  - Tunnels
  - Bridges
  - Retaining walls
- Experiment with different methods to provide the flexibility in the exploration of the design



Learning Resources

- Autodesk University
- https://primer.dynamobim.org/
- https://www.generativedesign.org
- https://forum.dynamobim.com/
- https://dynamobim.org/blog/



References

Autodesk University

- CES322249 Computational Design for Civil Engineers
- BES471869 Non-Geeks Guide to Optimizing Daily Workflows with Generative Design
- AS473693-L Generative Design at Hogwarts: Using Tech Instead of Magic
- CES473668 Supercharge Your Dynamo Graph with Civil 3D Toolkit