

ChatGPT + Geometry

Arch 2710 - Fall 2024

Prof Earl Mark

Purpose

This tutorial is intended to show how you can use tools like “Co-Pilot” and “ChatGPT” to model complex geometries. This is primarily done by asking these programs for a series of equations that will generate 2D or 3D geometry. While we will focus on how this can allow for the easier production of simple geometric shapes, this workflow has the capacity to allow you to work collaboratively with these tools to produce complex forms.

Process

Example #1

1. Find the name of the shape that you are looking to produce (eg: heptagram)
2. Ask ChatGPT to explain how to produce that geometry in Grasshopper:

Can you explain how to create a heptagram using a series of equations in the grasshopper plugin for rhino

3. Confirm the results that ChatGPT has given you, does it seem reasonable and appropriate?

1. Understanding the Geometry of a Heptagram

A heptagram is a star polygon with 7 vertices and intersecting edges. To construct it:

- Divide a circle into 7 equal parts.
- Connect every n -th vertex, where n is a skip step (e.g., 2 or 3) to form the star.

For a heptagram:

- The step size determines the type:
 - $7/2$ (connect every second point).
 - $7/3$ (connect every third point).

4. Read through directions given to you by ChatGPT

2. Components and Workflow in Grasshopper

Inputs:

- A circle as the base geometry.
- Parameters for divisions and step size.

Steps:

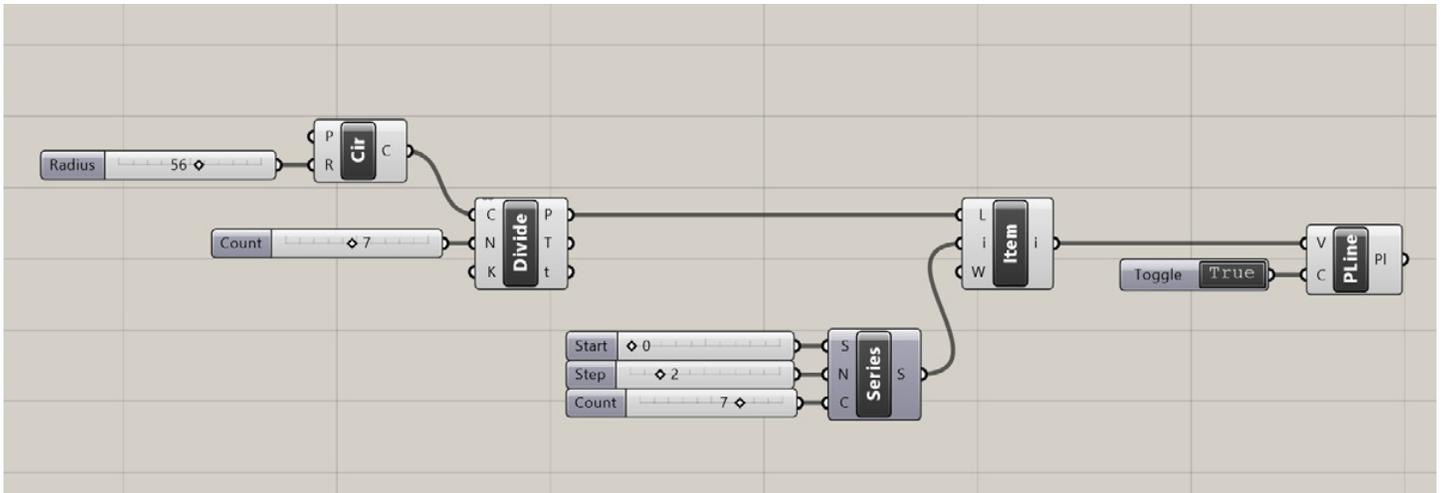
1. **Create the Circle:**
 - Use the `Circle` component to generate the base circle.
 - Input a radius using a `Number Slider`.
2. **Divide the Circle:**
 - Use the `Divide Curve` component to divide the circle into 7 equal parts.
 - Set the `Number of Divisions` to 7.
3. **Generate Point Indices for the Star:**
 - Use a `Series` component:
 - **Start:** 0 (first point index).
 - **Step:** Skip step (e.g., 2 or 3).
 - **Count:** Equal to the number of points (7).
 - This generates a sequence of indices that define the star's connection pattern.
4. **Shift Indices for Wraparound:**
 - Since Grasshopper works with sequential indices, ensure the series "wraps around" at the end using the `Wrap` option.
5. **Connect the Points:**
 - Use the `List Item` component to extract the points based on the generated indices.
 - Use the `Polyline` component to connect the points in the desired order and close the shape.
6. **Visualize the Heptagram:**
 - The `Polyline` component will display the heptagram shape.
 - Use `Extrude` or other transformations if a 3D version is needed.

Grasshopper Scripting with Equations

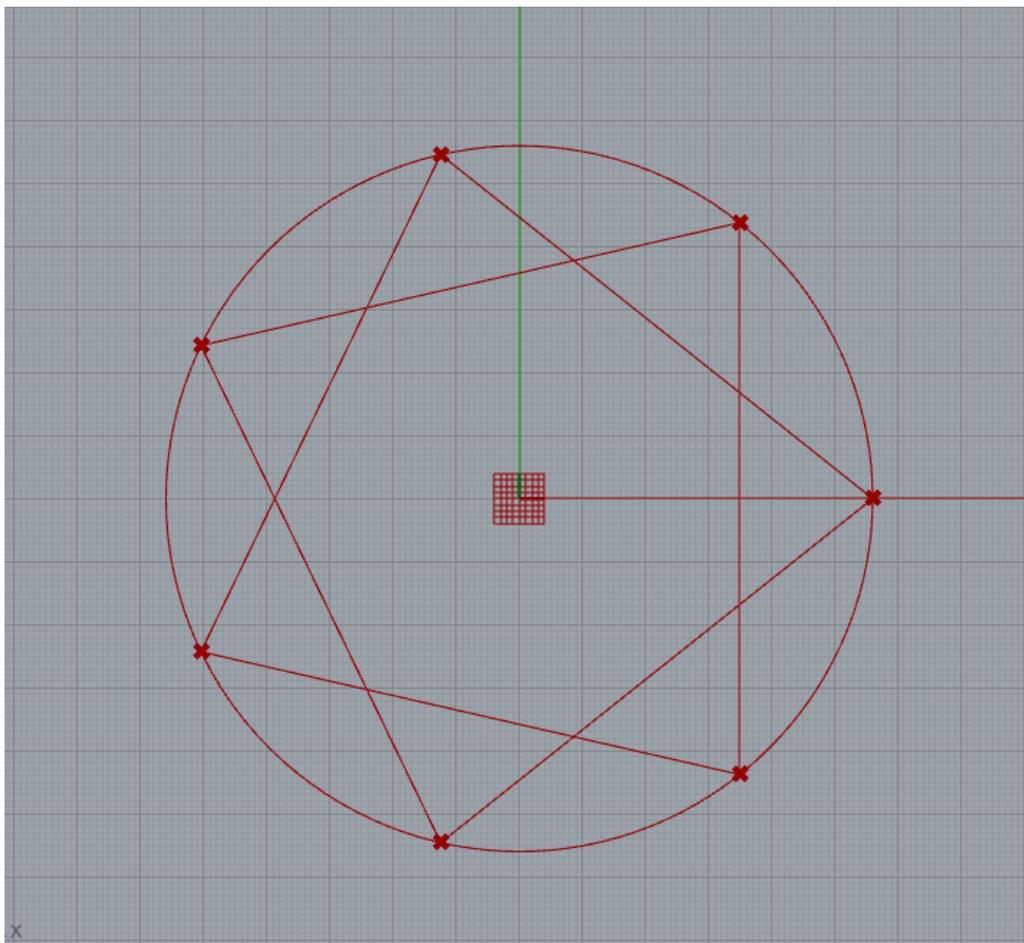
To use equations for precise control:

- Parameterize the circle using polar coordinates:
 - $x = r \cdot \cos(\theta)$
 - $y = r \cdot \sin(\theta)$
- Generate points programmatically by iterating angles in multiples of $2\pi/7$ (modulo 7).

5. Build Grasshopper script based on ChatGPT instructions (checking for validity)



6. Check the geometry in Rhino



Example #1

1. Find the name of the shape that you are looking to produce (eg: heptagram)
2. Ask ChatGPT to explain how to produce that geometry in Grasshopper:

Can you explain how to create the 2D geometry astroid using a series of equations in the grasshopper plugin for rhino

3. Confirm the results that ChatGPT has given you, does it seem reasonable and appropriate?

An **astroid** is a 2D curve classified as a type of hypocycloid. Its characteristic shape is a four-cusped star, and it can be described mathematically using parametric equations.

Here's how to create the 2D geometry of an astroid using equations in Grasshopper for Rhino.

4. Read through directions given to you by ChatGPT

1. Mathematical Background

The parametric equations for an astroid are:

$$x = a \cdot \cos^3(t)$$

$$y = a \cdot \sin^3(t)$$

where:

- t is the parameter (angle) ranging from 0 to 2π ,
- a is a scaling factor (radius).

2. Workflow in Grasshopper

Inputs:

- a : A parameter to control the size of the astroid.
- t : A range of values for the angle (parameter).

Steps:

1. Define the Angle Range:
 - Use the **Range** component to create a list of evenly spaced t -values from 0 to 2π .
 - Connect a **Number Slider** to control the number of divisions for the range (e.g., 100 for a smooth curve).
2. Calculate x and y :
 - Use the **Expression** component to compute x and y values for the astroid using the parametric equations:
 - For x : $a * (\text{Cos}(t)^3)$
 - For y : $a * (\text{Sin}(t)^3)$
 - Define the scaling factor a using a **Number Slider**.
3. Combine x and y :
 - Use the **Construct Point** component to assemble the x and y values into a list of points.
4. Create the Curve:
 - Use the **Interpolate** component to create a smooth curve through the points.

Grasshopper Implementation

Component Setup:

1. Parameterize t :
 - **Range**:
 - Domain: 0 to 2π (**Construct Domain** with start 0 and end 2π).
 - Steps: Set via a slider for resolution.
2. Calculate x :
 - **Expression**: $x = a \cdot \cos^3(t)$
 - Input a (scaling factor) and t (range values from **Range**).
3. Calculate y :
 - **Expression**: $y = a \cdot \sin^3(t)$
 - Input a and t .
4. Assemble Points:
 - **Construct Point**: Use x and y from the above calculations.
5. Draw the Curve:
 - **Interpolate**: Connect the points to create the astroid.

5. Push back! ChatGPT seems to be giving a general framework for how to implement this approach rather than an explicit series of steps. Try to request more specific instructions based on the information given.

Can you give explicit instructions for how to use those formulas to create the astroid? Including which grasshopper components to use and how to connect them

6. Read through directions given to you by ChatGPT

To create an **astroid** in Grasshopper using its parametric equations, follow these **explicit instructions**, step by step.

Astroid Parametric Equations

$$x = a \cdot \cos^3(t)$$

$$y = a \cdot \sin^3(t)$$

Where:

- t : Parameter ranging from 0 to 2π .
- a : Scaling factor for the size of the astroid.

Step-by-Step Instructions

1. Create the Range for t (Parameter)

1. Component: Range

- **Purpose:** Generate evenly spaced values for t from 0 to 2π .
- **Steps:**
 1. Add a Range component.
 2. Connect a Construct Domain component to the Domain input:
 - Set the start to 0 and end to 2π using a Number Slider (min: 0, max: 6.28).
 3. Use another Number Slider to set the number of steps (e.g., 100 for a smooth curve).

2. Compute $x = a \cdot \cos^3(t)$

1. Component: Expression

- **Purpose:** Calculate the x -coordinate of each point.
- **Steps:**
 1. Add an Expression component.
 2. Enter the formula: $a \cdot \cos(t)^3$ into the Expression field.
 3. Create a Number Slider for a (scaling factor, e.g., range 0 to 10) and connect it to the input a .
 4. Connect the output of Range (parameter t) to the input t .

3. Compute $y = a \cdot \sin^3(t)$

1. Component: Another **Expression**

- **Purpose:** Calculate the y -coordinate of each point.
 - **Steps:**
 1. Add a second **Expression** component.
 2. Enter the formula: `a * Sin(t)^3` into the **Expression** field.
 3. Use the same **Number Slider** for a as the previous step and connect it to the input a .
 4. Connect the output of **Range** (parameter t) to the input t .
-

4. Assemble Points

1. Component: **Construct Point**

- **Purpose:** Combine x - and y -coordinates into 2D points.
- **Steps:**
 1. Add a **Construct Point** component.
 2. Connect the output of the first **Expression** (for x) to the X input.
 3. Connect the output of the second **Expression** (for y) to the Y input.
 4. Leave the Z input empty (default is 0, for 2D points).

5. Create the Astroid Curve

1. Component: **Interpolate**

- **Purpose:** Create a smooth curve through the computed points.
 - **Steps:**
 1. Add an **Interpolate** component.
 2. Connect the output of **Construct Point** to the input of **Interpolate**.
-

Optional Adjustments

1. Smoothness of Curve:

- Increase the number of steps in the **Range** slider for a finer resolution.

2. Size of Astroid:

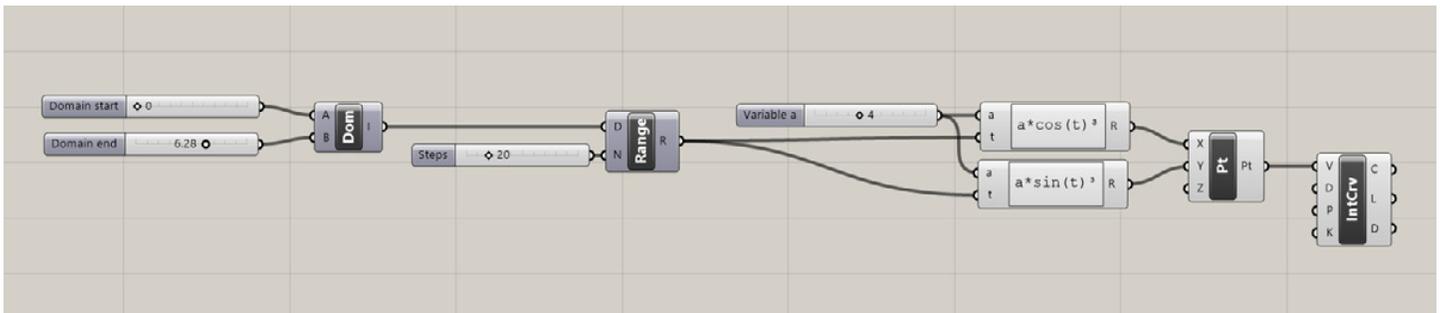
- Adjust the scaling factor a using the connected slider.

Complete Grasshopper Layout

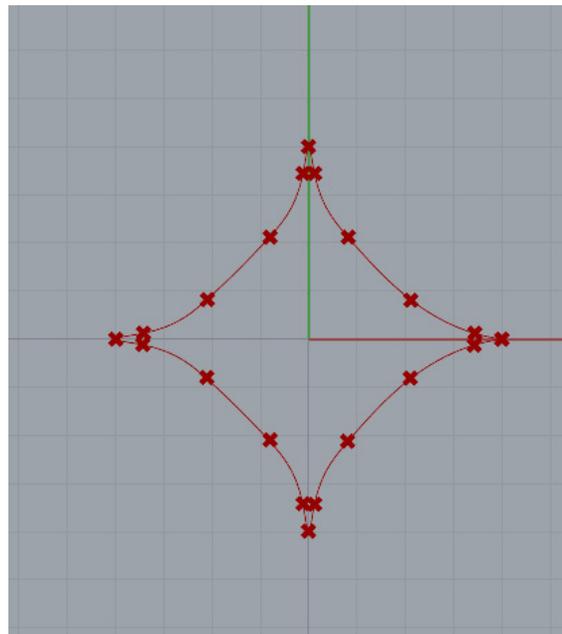
Here's the component list and their connections:

1. **Number Slider** : Set a (scale factor).
2. **Range** : Generate t -values.
3. **Construct Domain** : Define 0 to 2π as the domain for t .
4. Two **Expression** components:
 - First for $x = a \cdot \cos^3(t)$,
 - Second for $y = a \cdot \sin^3(t)$.
5. **Construct Point** : Combine x - and y -coordinates into points.
6. **Interpolate** : Create the astroid curve.

5. Build Grasshopper script based on ChatGPT instructions (checking for validity)



6. Check the geometry in Rhino



Example #3

1. Find the name of the shape that you are looking to produce (eg: helicoid)
2. Ask ChatGPT to explain how to produce that geometry in Grasshopper:

Can you explain how to create the 3D geometry helicoid using a series of equations in the grasshopper plugin for rhino

3. Confirm the results that ChatGPT has given you, does it seem reasonable and appropriate?

To create a **helicoid** in Grasshopper for Rhino using equations, follow this step-by-step guide. A helicoid is a type of ruled surface that spirals around an axis, defined by parametric equations.

4. Read through directions given to you by ChatGPT

Mathematical Representation of a Helicoid

The parametric equations for a helicoid are:

$$x = u \cdot \cos(v)$$

$$y = u \cdot \sin(v)$$

$$z = c \cdot v$$

Where:

- u : Controls the radius, ranging from $-r$ to r (radial extent).
- v : Controls the angle, ranging from 0 to $2\pi \cdot h$ (where h determines the number of twists).
- c : Scaling factor for the vertical rise per revolution.

Step-by-Step Instructions for Grasshopper

1. Define the Parameters u and v

1. Component: `Range` (for u and v):

- **Purpose:** Generate evenly spaced values for u (radius) and v (angle).
- **Steps for u :**
 1. Add a `Range` component for u .
 2. Connect a `Construct Domain` component to the `Domain` input of `Range`:
 - Set the domain from $-r$ to r using `Number Sliders` for r (e.g., -5 to 5).
 3. Use a `Number Slider` to set the **number of steps** (e.g., 50 for smoothness).
- **Steps for v :**
 1. Add another `Range` component for v .
 2. Connect a `Construct Domain` component to the `Domain` input:
 - Set the domain from 0 to $2\pi \cdot h$ (e.g., $6.28 \cdot 3$ for three twists).
 3. Use a `Number Slider` for h (number of twists) and c (vertical rise per revolution).

2. Generate a Grid of Points

To create a surface, you need a 2D grid of points based on the parameters u and v .

1. Component: `Construct Domain2`

- **Purpose:** Combine the u and v ranges into a 2D grid.
- **Steps:**
 1. Use `Construct Domain2` to combine the u -range and v -range outputs.

2. Component: `Mesh Grid` OR `Cross Reference`

- **Purpose:** Create a 2D grid of combinations for u and v .
- Connect u and v as inputs.

3. Compute the Coordinates

1. Component: Three **Expression** components (for x , y , and z).

- **Purpose:** Compute the x , y , and z coordinates for each u , v pair.
- **Steps for Each:**
 - Add an **Expression** component.
 - Enter the formulas:
 - $x = u \cdot \cos(v)$
 - $y = u \cdot \sin(v)$
 - $z = c \cdot v$.
 - Connect the corresponding u - and v -outputs to each **Expression**:
 - u to **u**, v to **v**, and c to **c**.

4. Assemble the 3D Points

1. Component: **Construct Point**

- **Purpose:** Combine x , y , and z into 3D points.
 - **Steps:**
 1. Add a **Construct Point** component.
 2. Connect the outputs of the x -, y -, and z -**Expression** components to the respective **X**, **Y**, and **Z** inputs.
-

5. Create the Helicoid Surface

1. Component: **Surface from Points**

- **Purpose:** Generate a surface from the 3D points.
- **Steps:**
 1. Add a **Surface from Points** component.
 2. Connect the output of the **Construct Point** component to the **Points** input.
 3. Set the **U Count** (number of points in the u -direction) to match the resolution of the u -range.

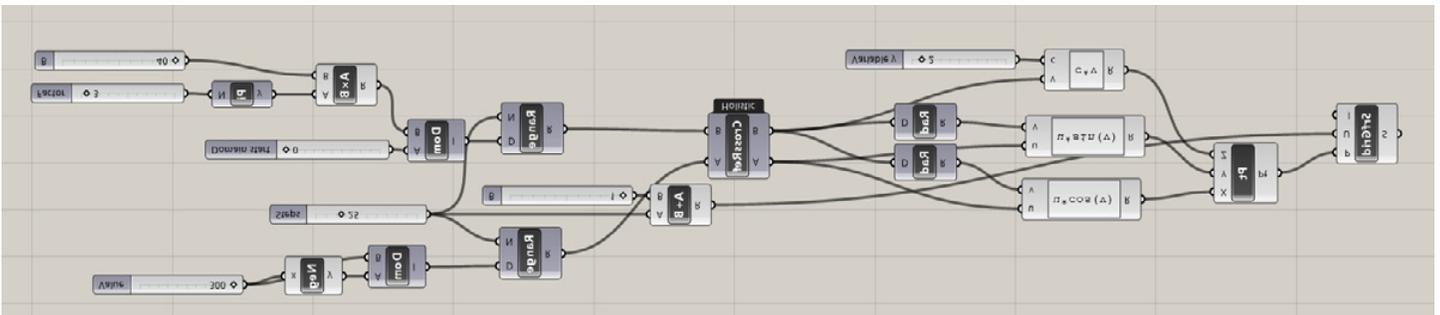
Optional Adjustments

- Use sliders to adjust:
 - Radius r (domain of u),
 - Number of twists h (extent of v),
 - Vertical scaling c .
- Increase the resolution (number of steps in u and v) for a smoother helicoid.

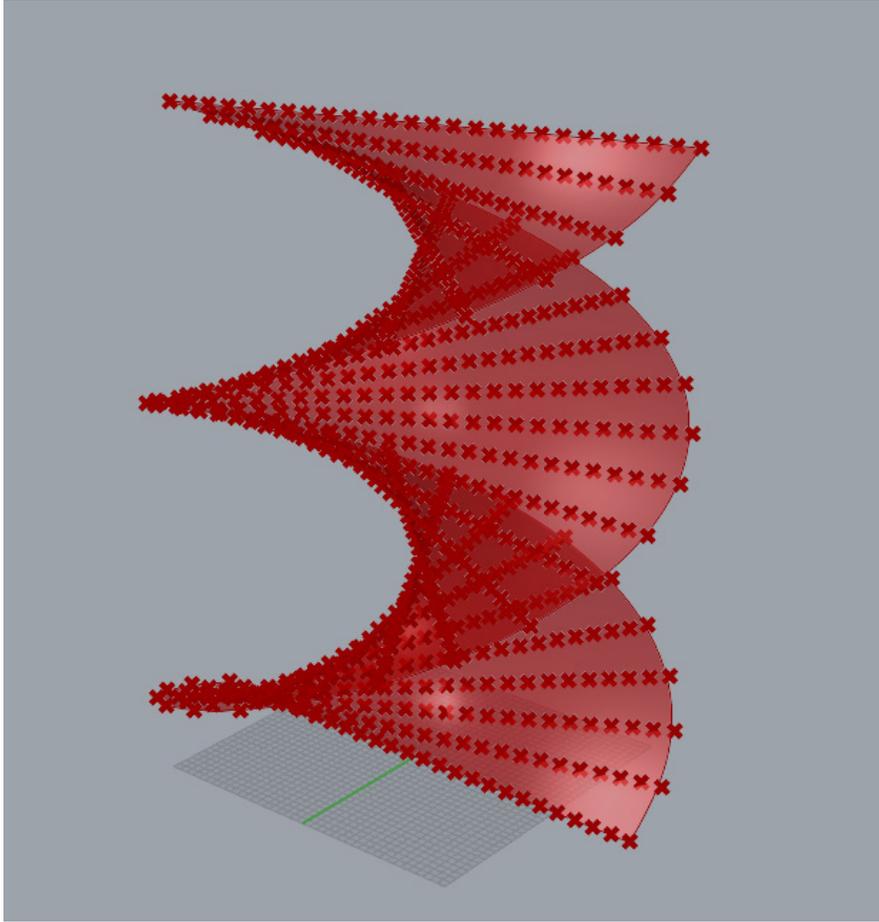
Component Summary

1. **Range** (2): For u and v parameters.
2. **Construct Domain**: Define the domain for u and v .
3. **Construct Domain2** or **Cross Reference**: Combine u and v into pairs.
4. **Expression** (3): Calculate x , y , z using the helicoid equations.
5. **Construct Point**: Combine x , y , z into 3D points.
6. **Surface from Points**: Create the helicoid surface.

5. Build Grasshopper script based on ChatGPT instructions (checking for validity)



6. Check the geometry in Rhino

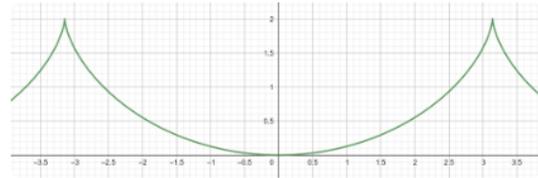


Other Techniques

Geometry Identification

1. Curve Recognition

Let me know if you'd like further clarification or examples in Grasshopper!



WHAT GEOMETRY IS THIS

The graph shown is the **inverted cycloid**, which is often described as the path traced by a point on the circumference of a circle as it rolls inside another fixed circle.

The mathematical representation of this geometry is typically expressed parametrically:

$$x = r(\theta - \sin \theta)$$

$$y = r(1 - \cos \theta)$$

Here:

- r is the radius of the rolling circle.
- θ is the angle of rotation.

If this is not what you were expecting, could you clarify your intent or provide additional details?

2 Geometry Recognition



Give me a series of formulas so that I can model the massing of this building in grasshopper

To model the **University of Virginia Rotunda** in Grasshopper, you can break down the building into its fundamental geometric components and define their parametric relationships. Here's how you can create the basic massing:

1. Cylindrical Base

The cylindrical base forms the main volume of the Rotunda.

- Formula for the Cylinder:

$$x = r \cdot \cos(\theta)$$

$$y = r \cdot \sin(\theta)$$

$$z = h$$

- Grasshopper Workflow:

1. Use a `Circle` component to define the base with radius r .
2. Extrude the circle vertically using the `Extrude` component with height h .

2. Dome

The dome is a semi-spherical volume sitting on top of the cylinder.

- Formula for the Dome:

$$x = r \cdot \cos(u) \cdot \sin(v)$$

$$y = r \cdot \sin(u) \cdot \sin(v)$$

$$z = r \cdot \cos(v)$$

Where:

- $u \in [0, 2\pi]$ (azimuth angle for full rotation).
 - $v \in [0, \pi/2]$ (elevation angle for the upper hemisphere).
- Grasshopper Workflow:
 1. Use a `Range` component to divide u and v .
 2. Use the `Construct Point` component to compute x, y, z for the dome.
 3. Use a `Surface from Points` component to create the dome surface.

3. Portico with Columns

The portico is a rectangular extrusion with cylindrical columns in front.

- **Portico:**
 1. Use a `Rectangle` component to define the footprint of the portico.
 2. Extrude the rectangle using the `Extrude` component with a small height.
 - **Columns:**
 1. Use a `Circle` component to define the base of each column.
 2. Array the circles horizontally to place multiple columns along the width of the portico.
 3. Extrude the circles vertically to create the column height.
-

4. Staircase

The staircase is a stepped series of rectangular extrusions.

- **Grasshopper Workflow:**
 1. Use a `Series` component to create stepped heights and offsets.
 2. Use a `Rectangle` component to define the footprint of each step.
 3. Use the `Move` component to stack the rectangles vertically.
 4. Extrude the rectangles to create the solid steps.

5. Arches and Basement

The lower portion has a series of arches.

- **Formula for an Arch:** Use a parametric semicircle for the arch opening:

$$x = r \cdot \cos(\theta)$$

$$y = r \cdot \sin(\theta)$$

Where $\theta \in [0, \pi]$.

- **Grasshopper Workflow:**
 1. Use a `Circle` component and trim it to a semicircle using the `Shatter` or `Trim` component.
 2. Array the semicircles along the base width to create multiple arches.
 3. Use `Boundary Surface` or `Extrude` to generate the 3D openings.

Sample Values for Massing

- **Cylindrical Base:**
 - Radius: ~10 units
 - Height: ~15 units
- **Dome:**
 - Radius: Same as the cylinder
 - Height: Half of the cylinder height
- **Portico:**
 - Width: Equal to the cylinder diameter
 - Depth: ~5 units
 - Column Count: 6 evenly spaced
- **Staircase:**
 - Step Height: 0.5 units
 - Step Depth: ~1 unit
 - Total Height: ~5 units
- **Arches:**
 - Radius: ~2 units
 - Spacing: 3-4 units between centers

Let me know if you'd like a more detailed implementation for any specific element!