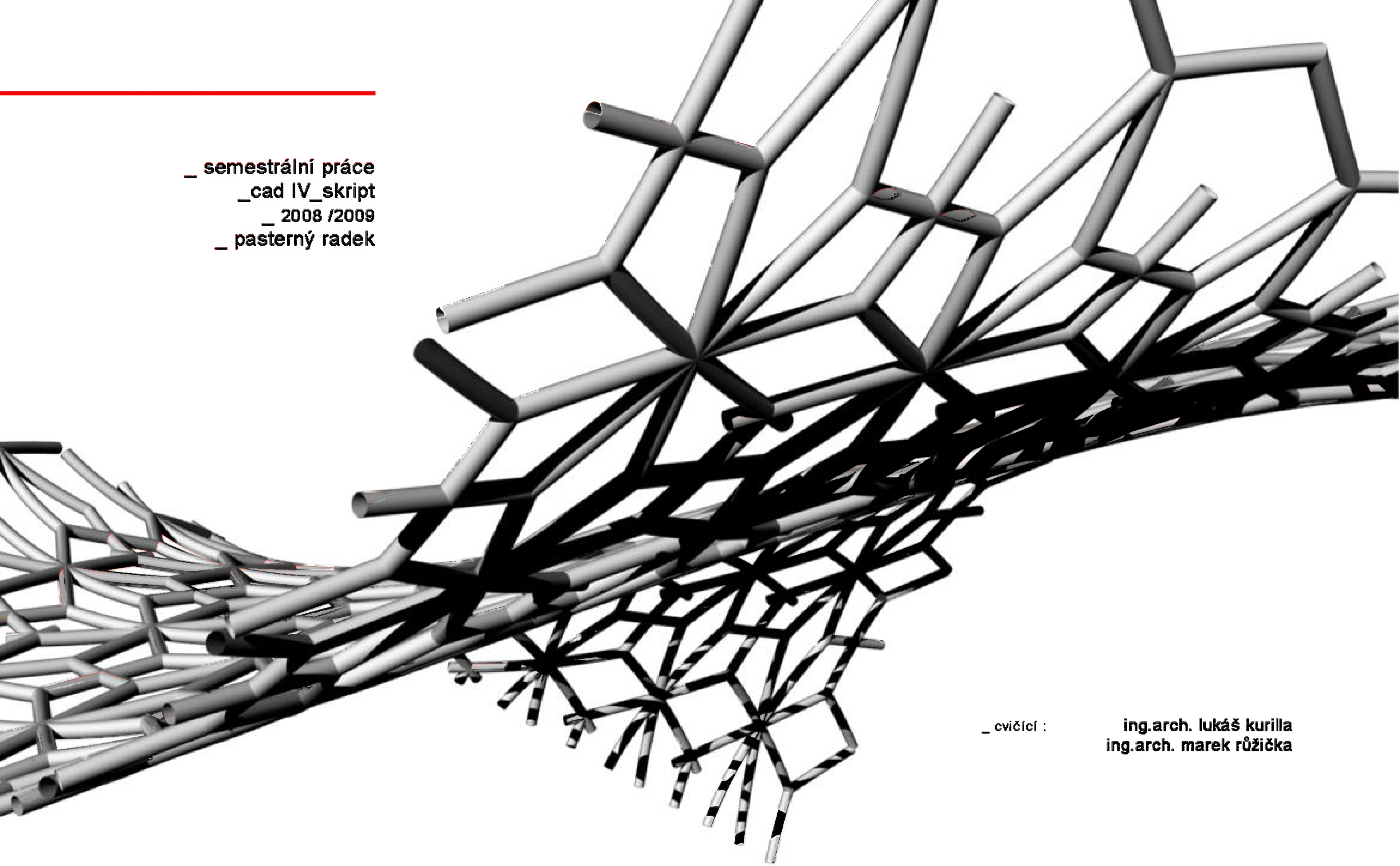


OPS | oktapolygonstructure

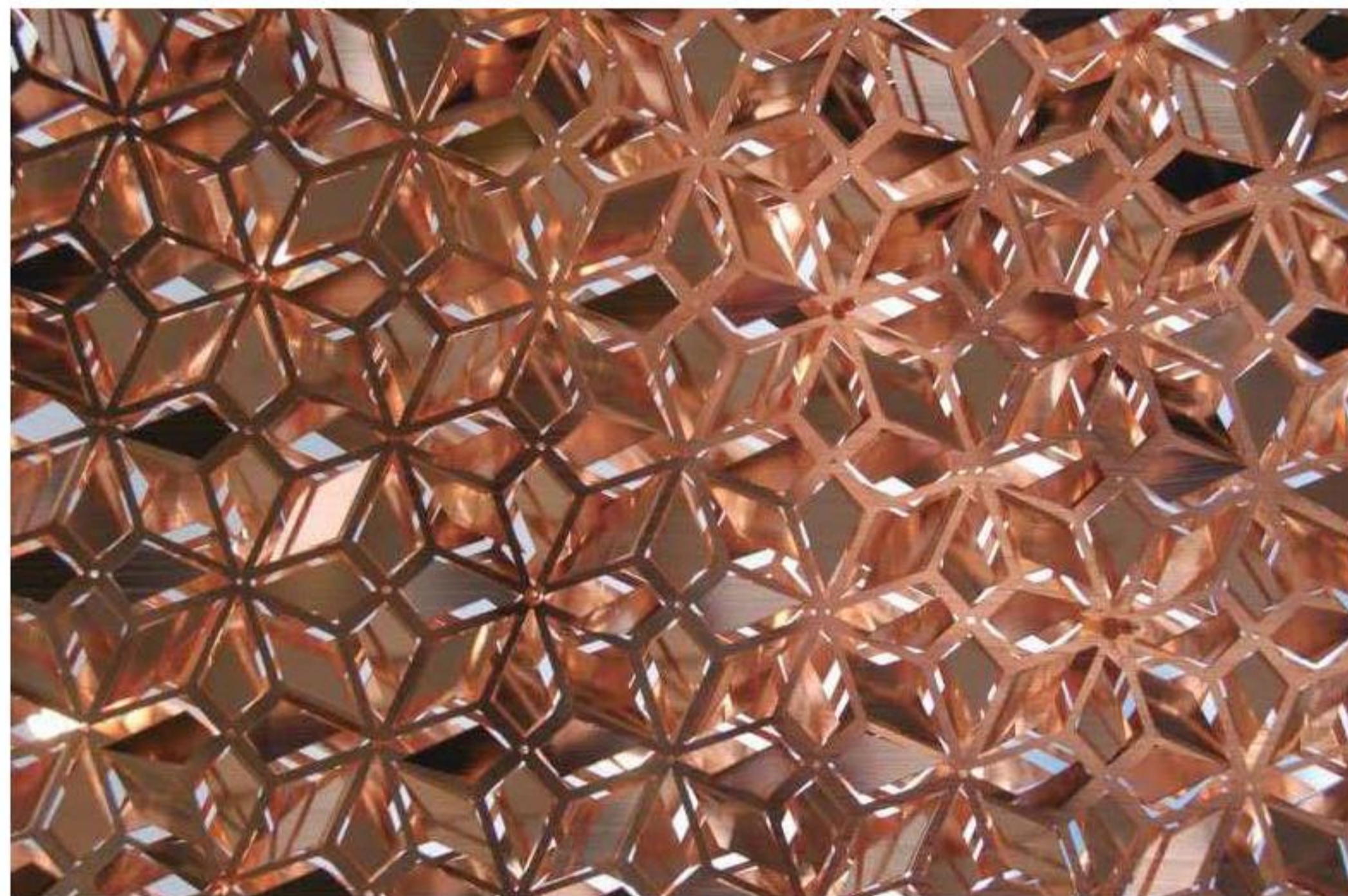
_ semestrální práce
_ cad IV_ skript
_ 2008 /2009
_ pasterný radek



_ cvičitel :

ing.arch. lukáš kurilla
ing.arch. marek růžička

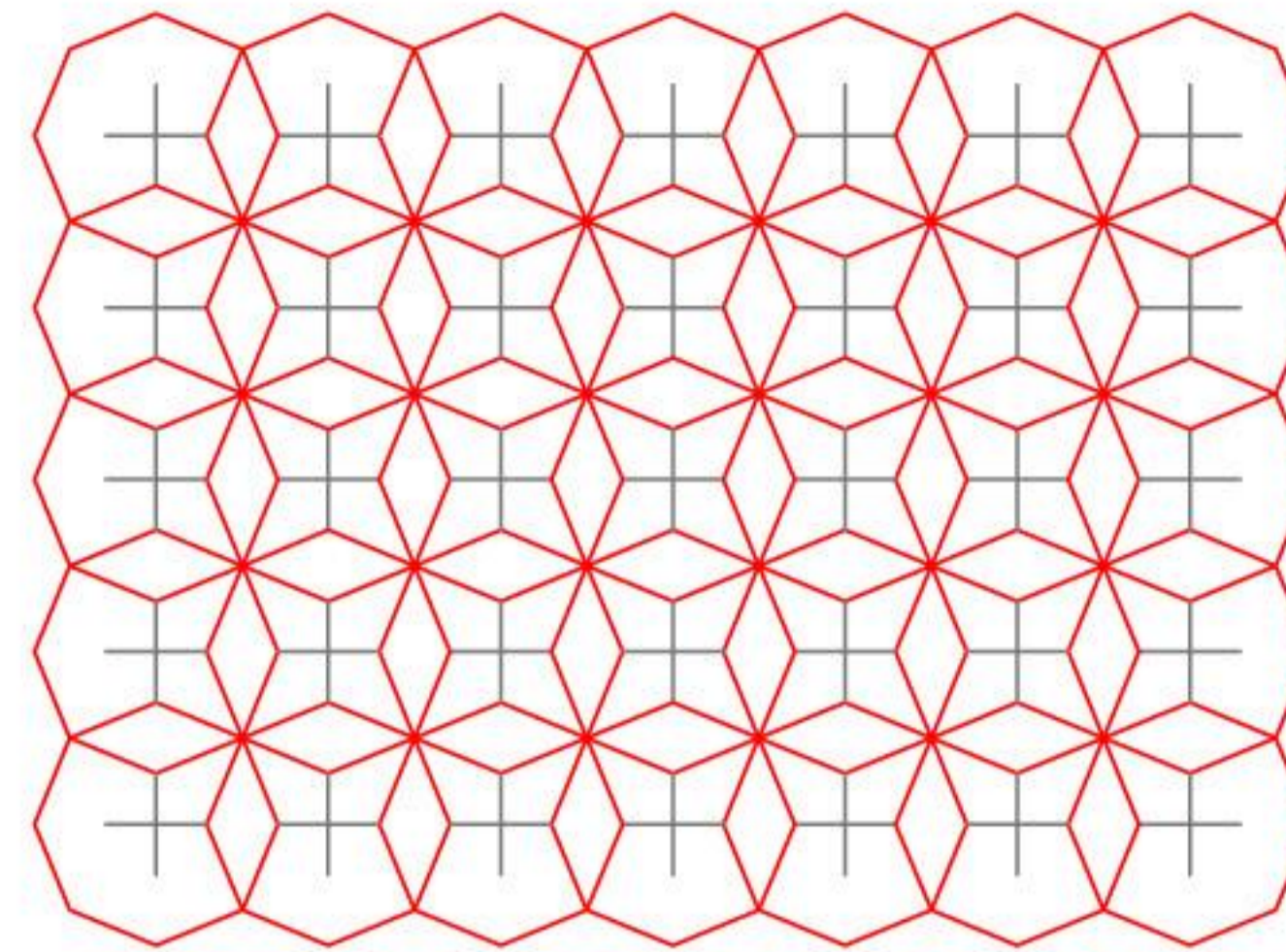
Mým úkolem byl rozbor vybraného ornamentu a jeho následné parametrické vytvoření v programu Grasshopper 0.6.0012, což je zásuvný skriptovací modul programu Rhinoceros 4.0.



fotografie vybraného ornamentu

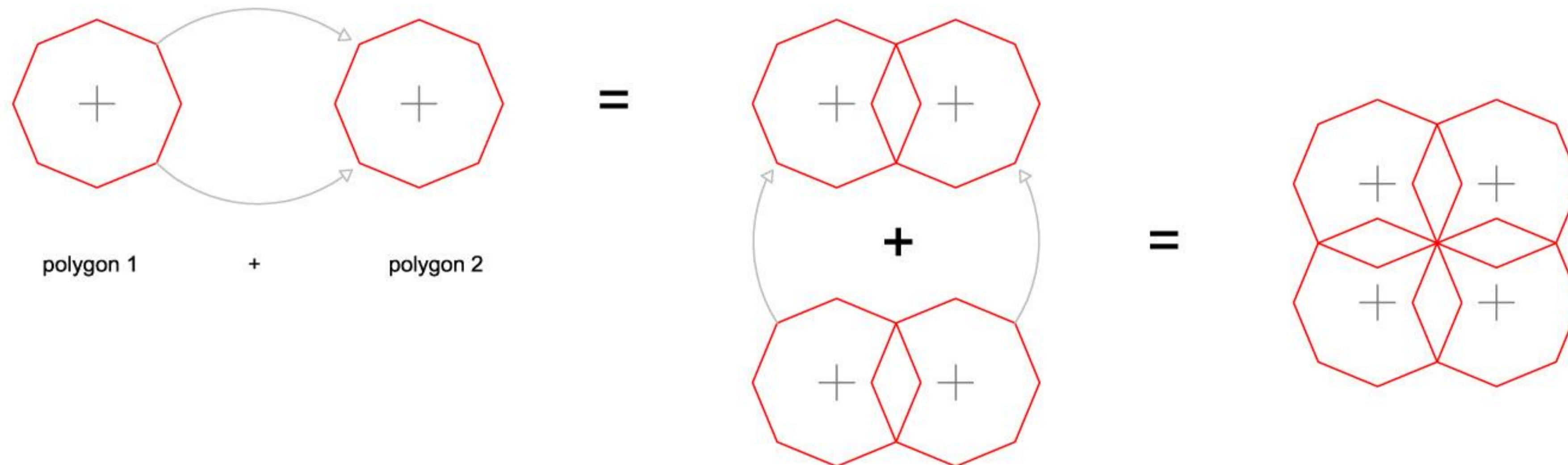
rozbor ornamentu

Ornament se skládá z pravidelných osmiúhelníků a přímk. Průnikem osmiúhelníků vzniká čtyřcípá pravidelná hvězdice, jejichž vrcholy, které jsou u vnitřních tupých úhlů, spojují úsečky dotvářející celý ornament. Úsečky se protínají vždy ve středu daného polygonu. Tento střed se pak stává "řídícím bodem". Tyto body pak vytváří společně pravidelnou síť řídících bodů.



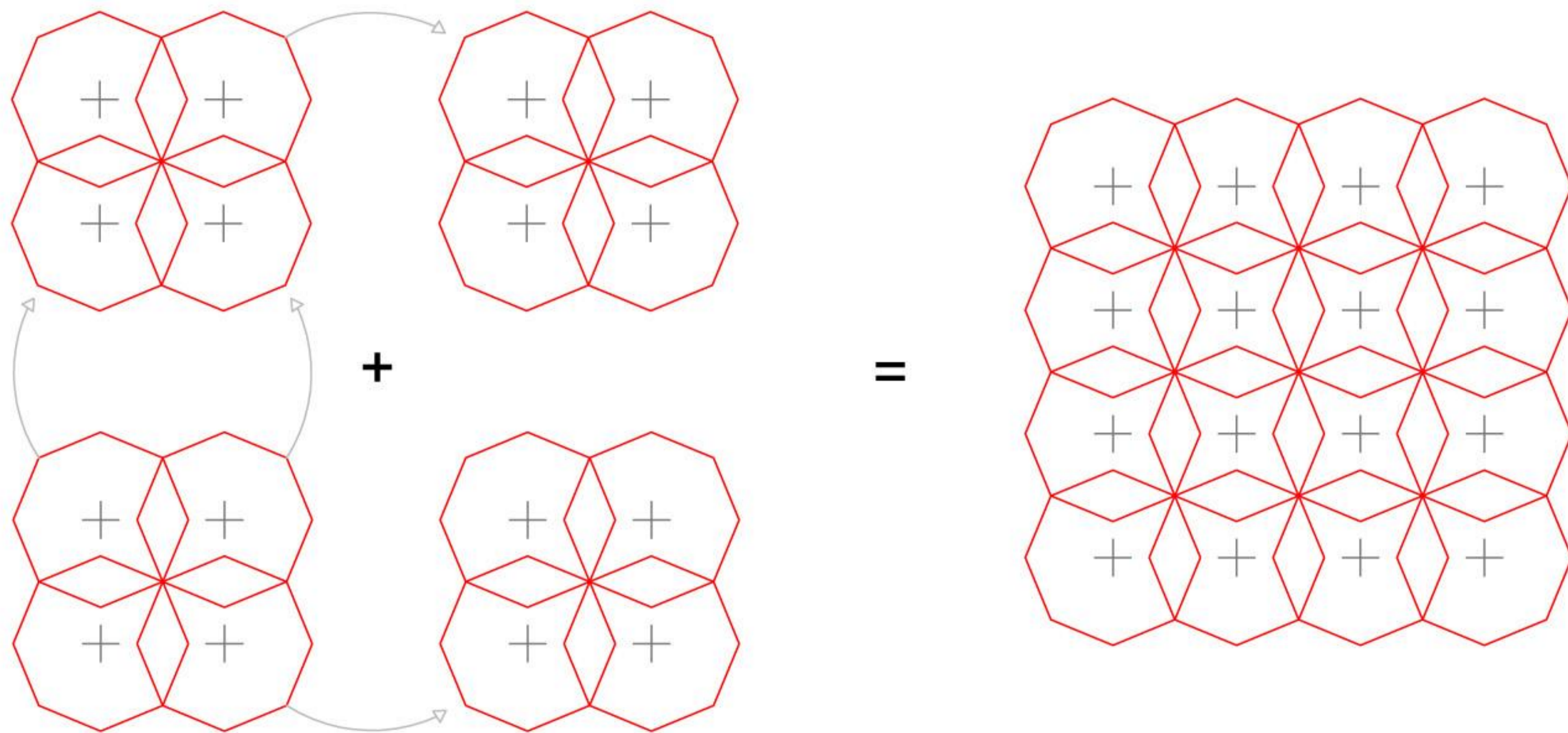
schématické znázornění vytváření ornamentu

Obrázek znázorňuje vzájemné napojování jednotlivých polygonů a postupné vytváření obrazce



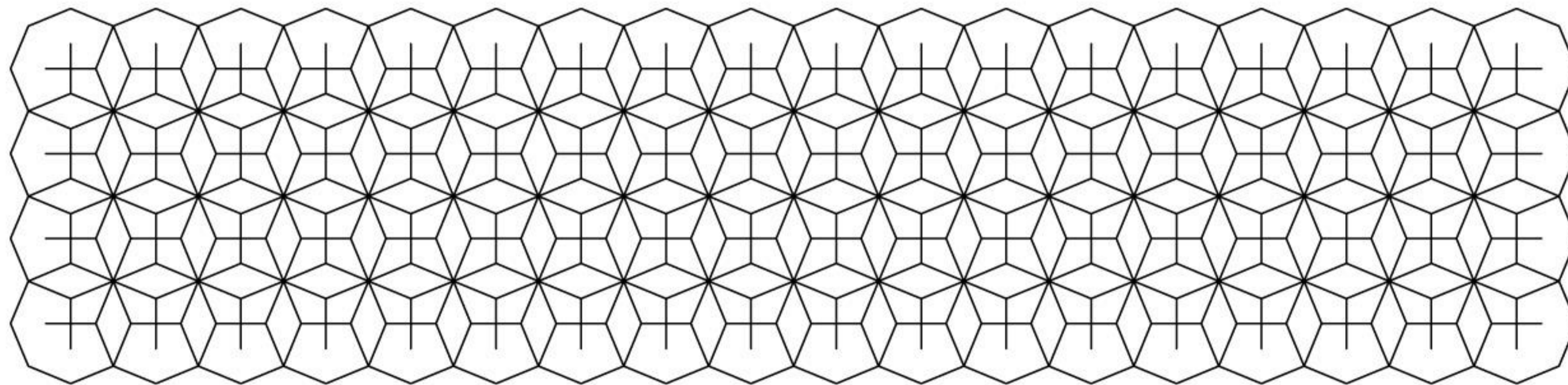
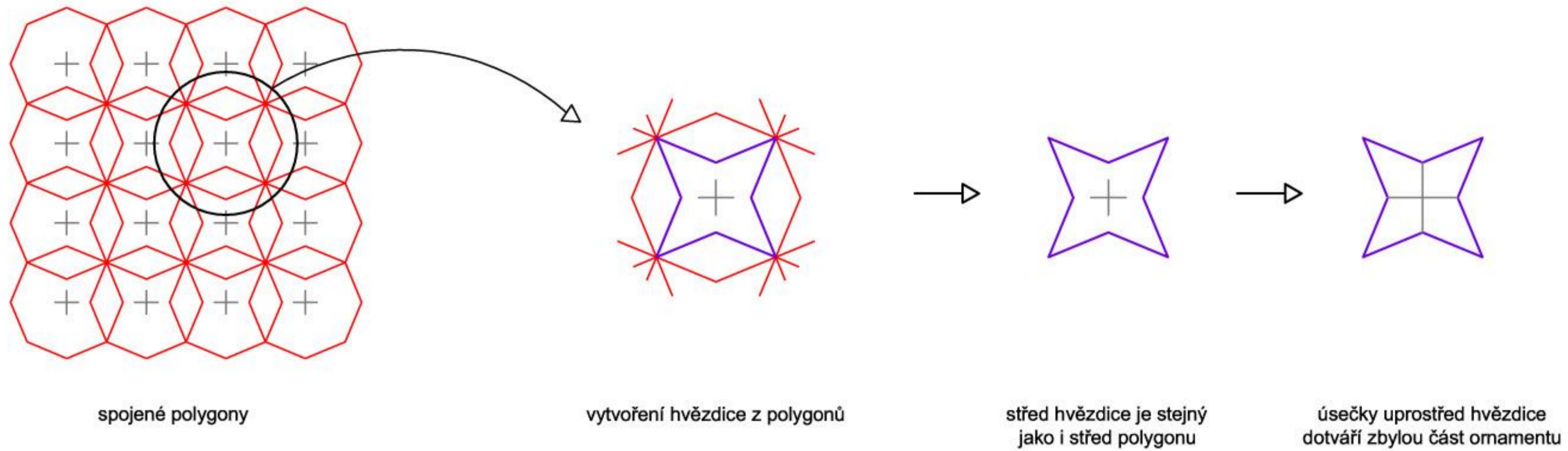
rozbor ornamentu

Obrázek znázorňuje vzájemné napojování jednotlivých polygonů a postupné vytváření obrazce



rozbor ornamentu

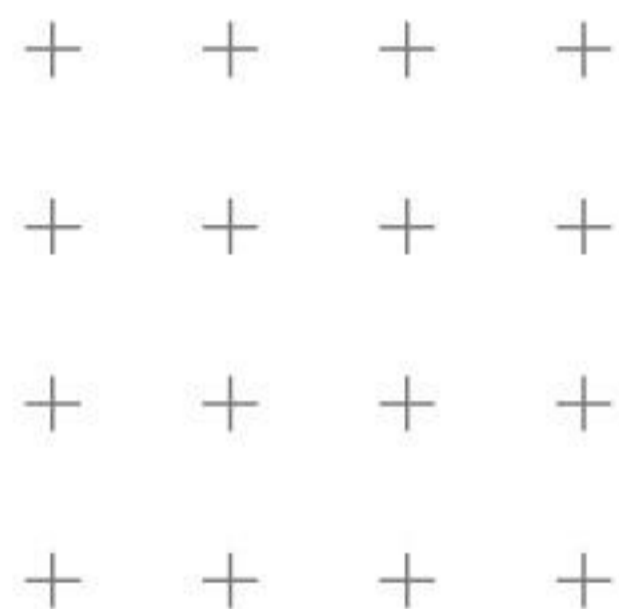
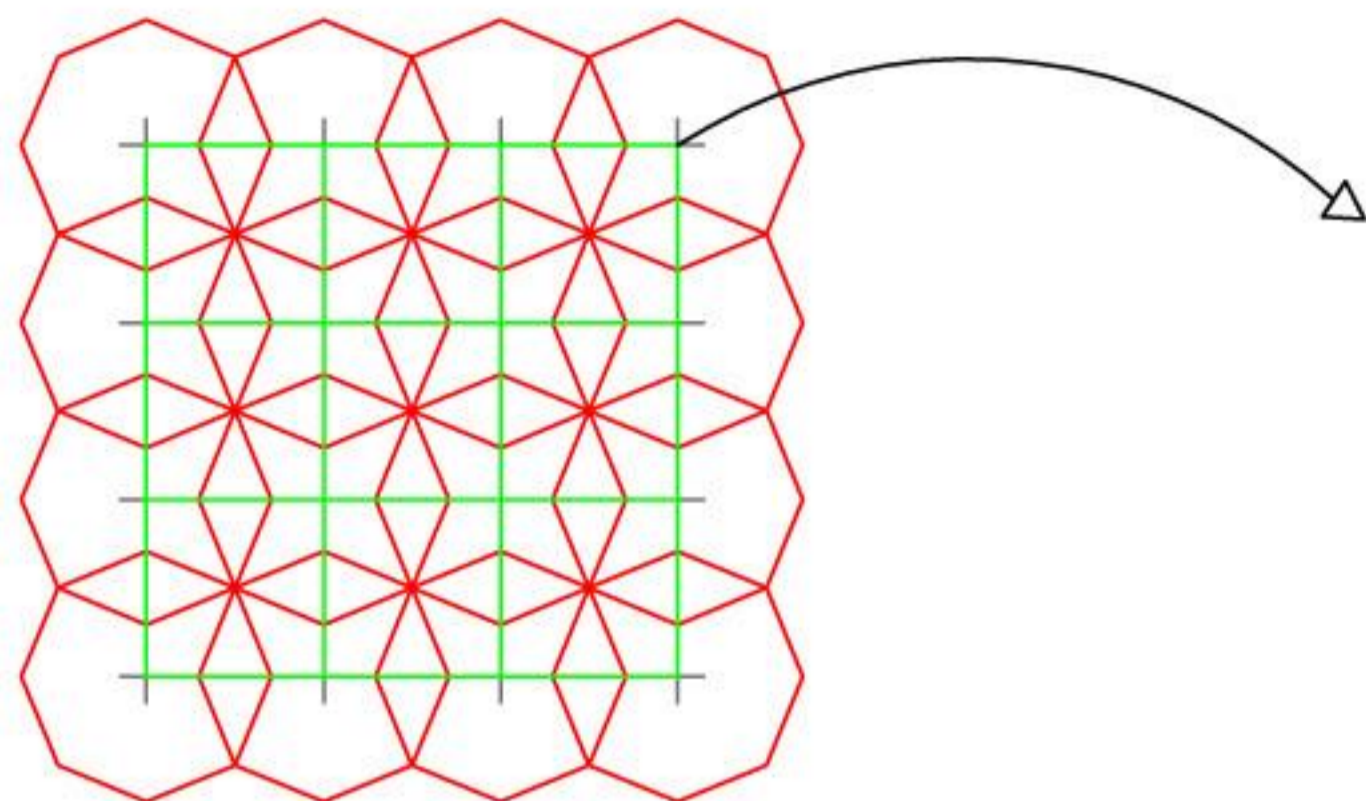
Obrázek ukazuje vytváření hvězdice z průniků osmiúhelníků.



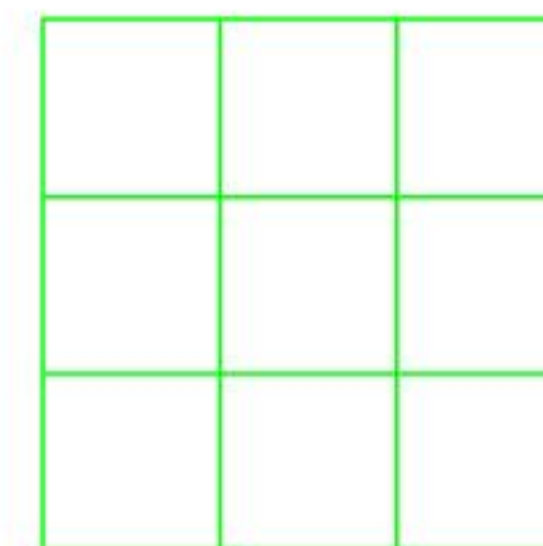
Výsledný ornament

rozbor ornamentu

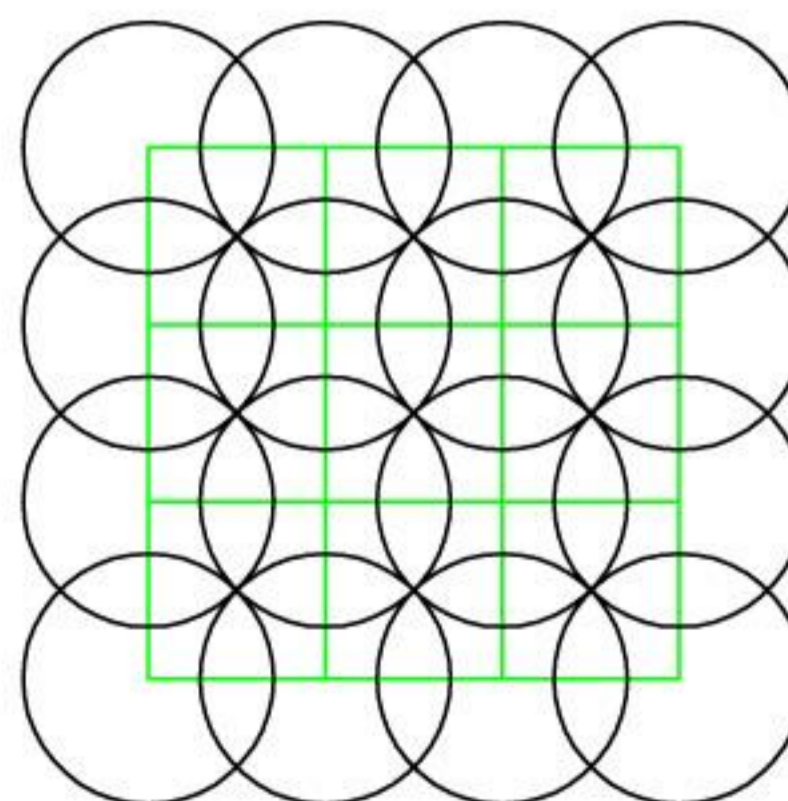
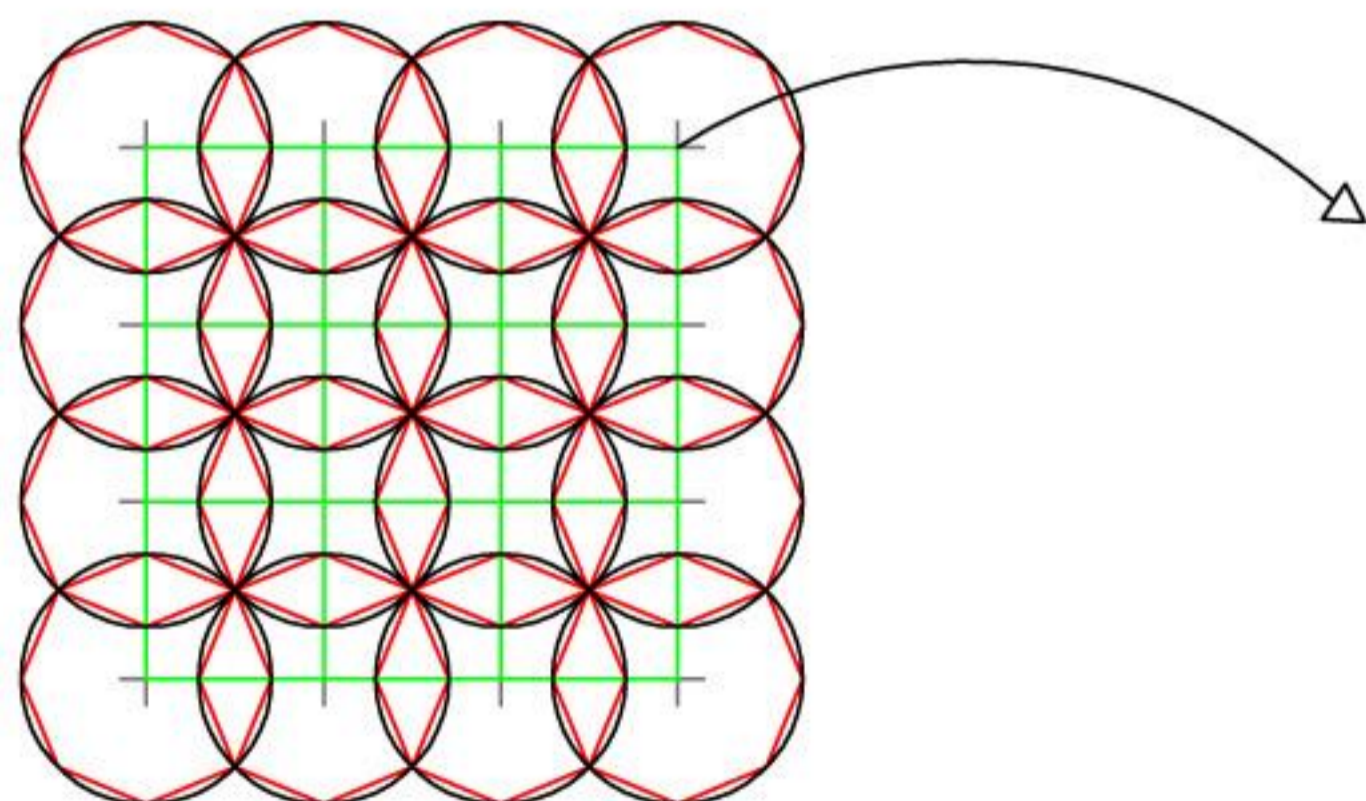
řídící body



střed polygonů - řídící body



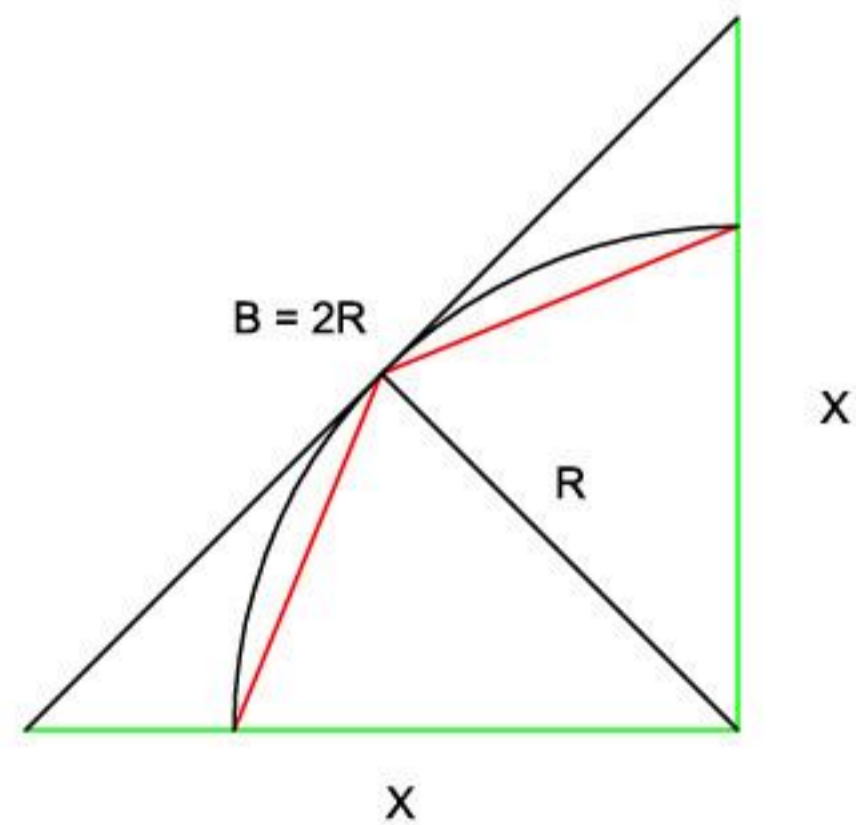
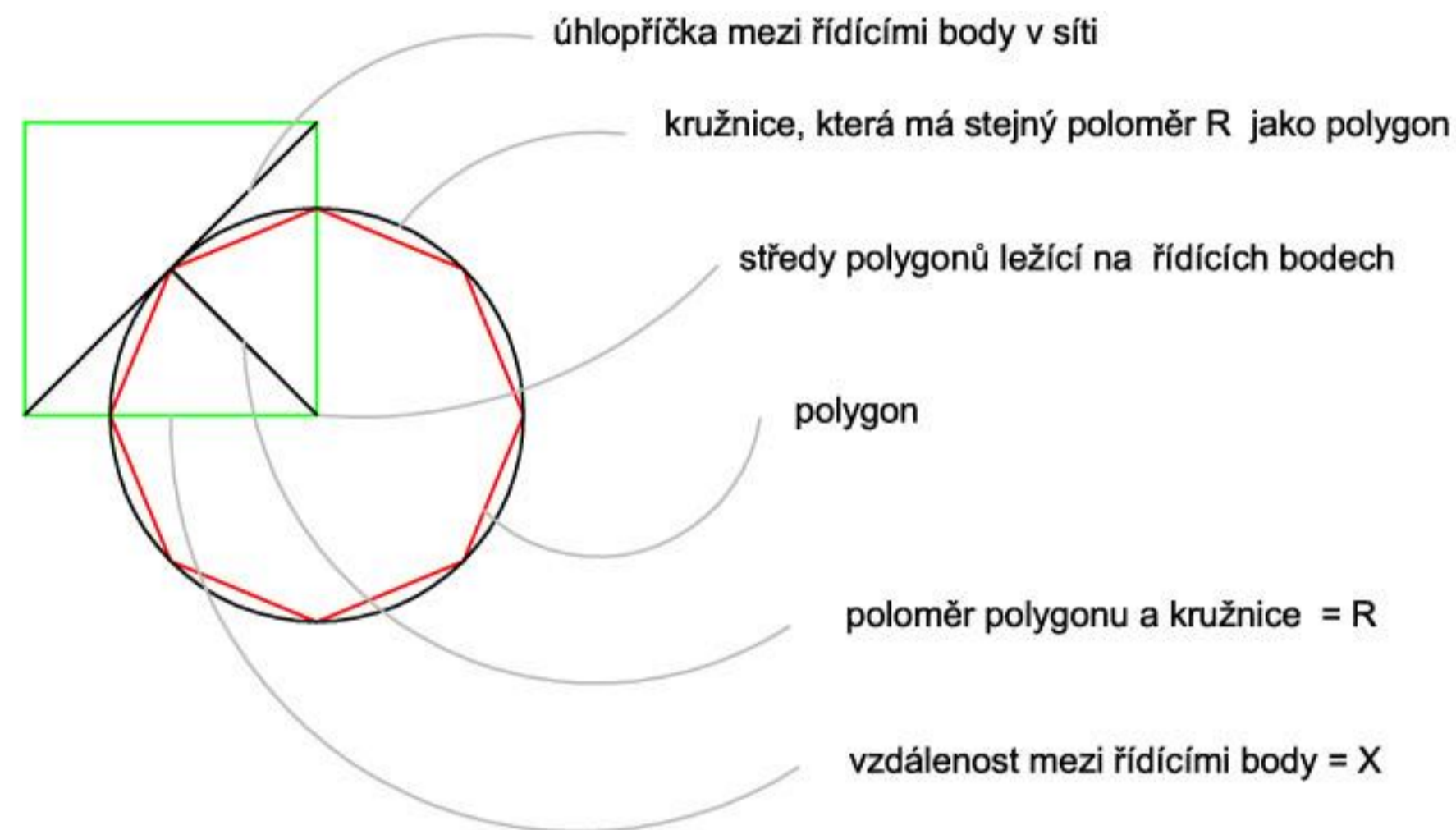
síť přímek
průsečíky přímek
=
řídící body



Kružnice opsaná polygonu,
má stejný střed jako polygon,
v našem případě střed = řídící bod.
Protože je tato kružnice opsaná polygonu,
má samozřejmě i stejný poloměr jako
naš osmiúhelník (polygon).

rozbor ornamentu

závislost polygonu na distanci mezi řídicími body



Pro vytvoření parametrického zadání jsem potřeboval zjistit, jak je poloměr polygonu závislý na měnící se distanci mezi řídicími body. K tomu mi pomohl předešlý obrázek a jeho následný rozbor a vytvoření rovnice, kdy poloměr R vyjádřím pomocí X, kde X je právě vzdálenost mezi řídicími body.

Všimněte si, že uhlopříčka v dané síti je vlastně tečnou kružnice, která má stejný poloměr jako náš polygon. Polovina ozrcadlené uhlopříčky v tečném bodě je tedy poloměrem kružnice a tedy i polygonu. Právě tuto délku potřebujem zjistit pro vytvoření parametrického modelu obrazce. Tuto délku určuje následující rovnice:

X distance mezi řídicími body
 B uhlopříčka
 R poloměr kružnice - hledaná neznámá

Rovnice vychází z pravidel v pravoúhlém trojúhelníku tedy Pythagorovi věty.

$$X^2 + X^2 = B^2$$

$$B^2 = 2X^2$$

$$B = X \cdot 2^{1/2}$$

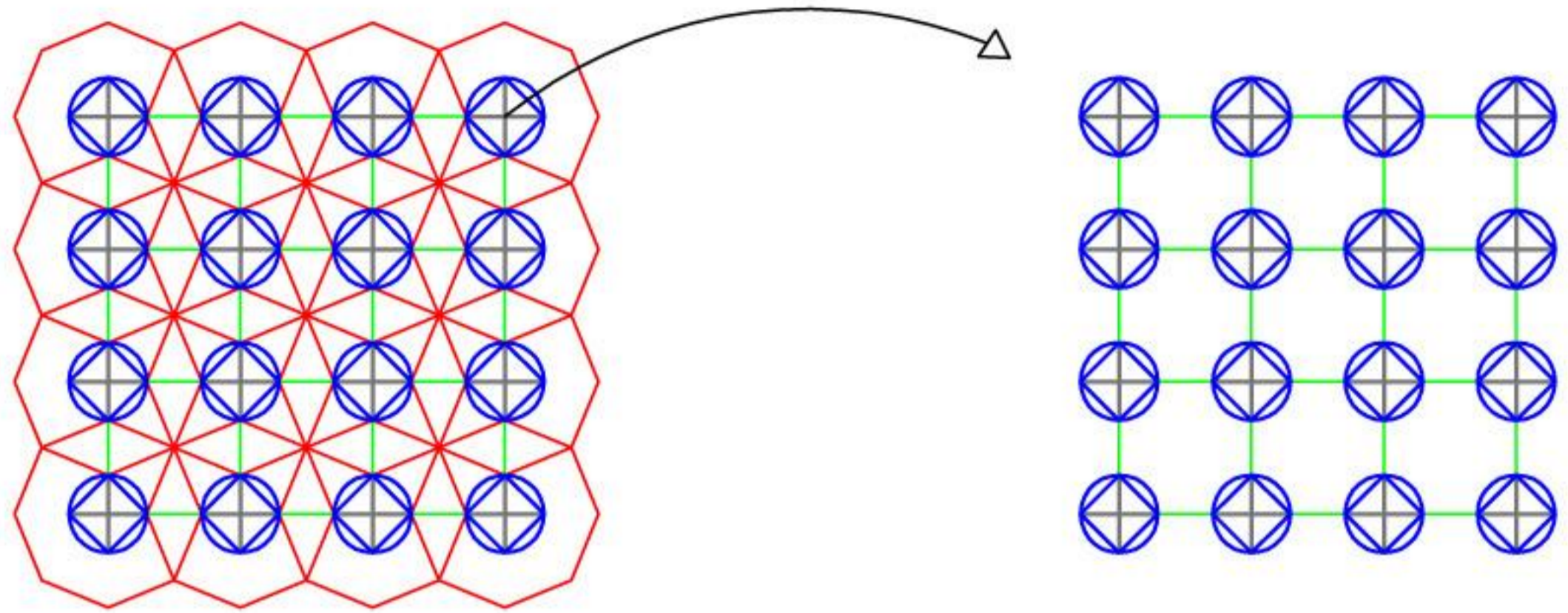
$$2R = X \cdot 2^{1/2}$$

$$R = X \cdot 2^{1/2} / 2 \quad \text{protože v programu grasshopper nebyla odmocnina možná, nahradil jsem část } 2^{1/2} / 2 \text{ za } \sin(45^\circ)$$

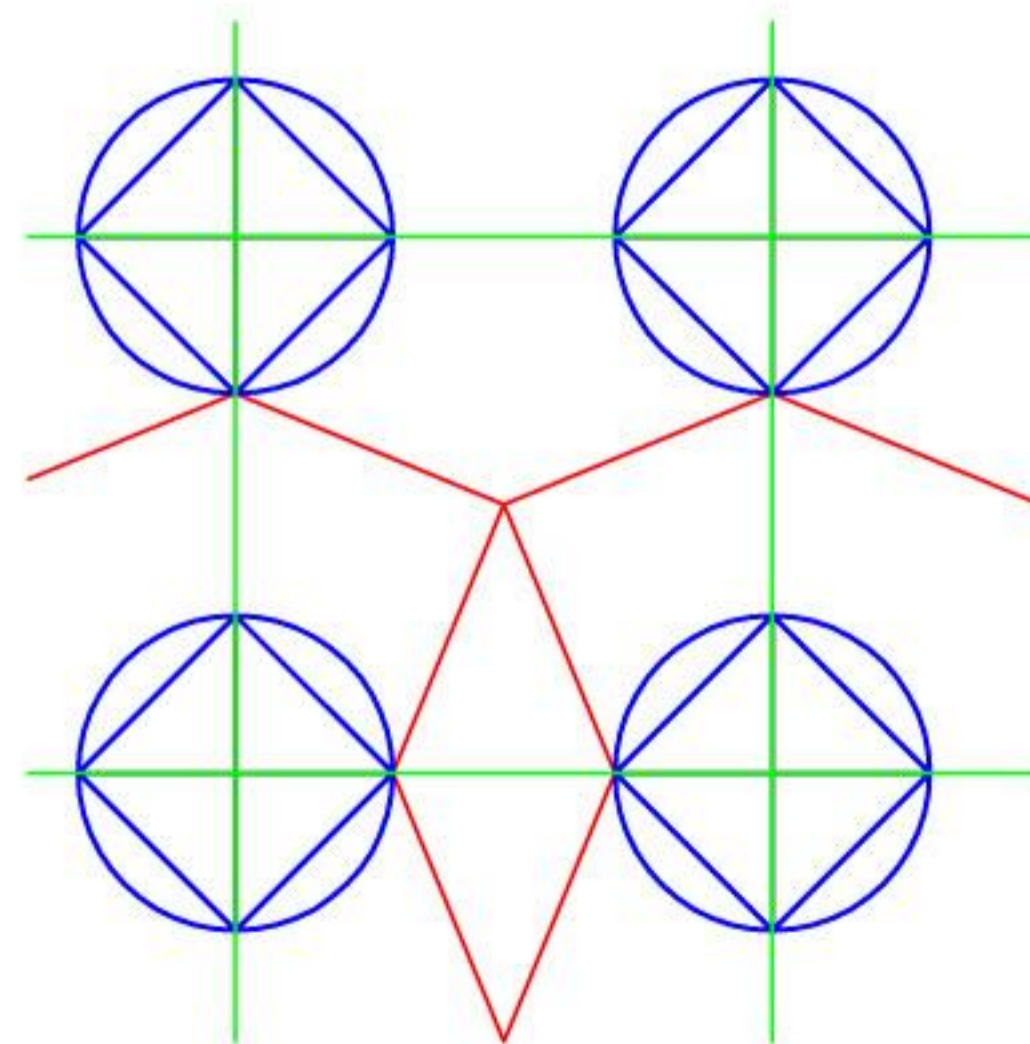
$$R = X \cdot \sin(45^\circ)$$

rozbor ornamentu

závislost polygonu na distanci mezi řídicími body



Pro dotvoření parametrického zadání jsem potřeboval také zjistit, jaká je závislost poloměru polygonu, který jsem si vytvořil opsáním vrcholů úseček, jenž dotváří zbytek obrazce, na distanci řídicích bodů. I tomuto polygonu (nyní čtyřúhelník) lze opsat samozřejmě kružnici, která má stejný poloměr jako daný polygon.



X

$$R_2 = X - (X * \sin(45^\circ))$$

$$R = X * \sin(45^\circ)$$

Rovnice pro poloměr nového polygonu, lze zjistit velice snadno. Pokud si uvědomíme, že poloměr nového polygonu je zbytkem distance mezi řídicími body.

Využijeme tedy této skutečnosti a předešlého výpočtu k vyjádření poloměru nového polygonu, tento poloměr označíme jako R_2 .

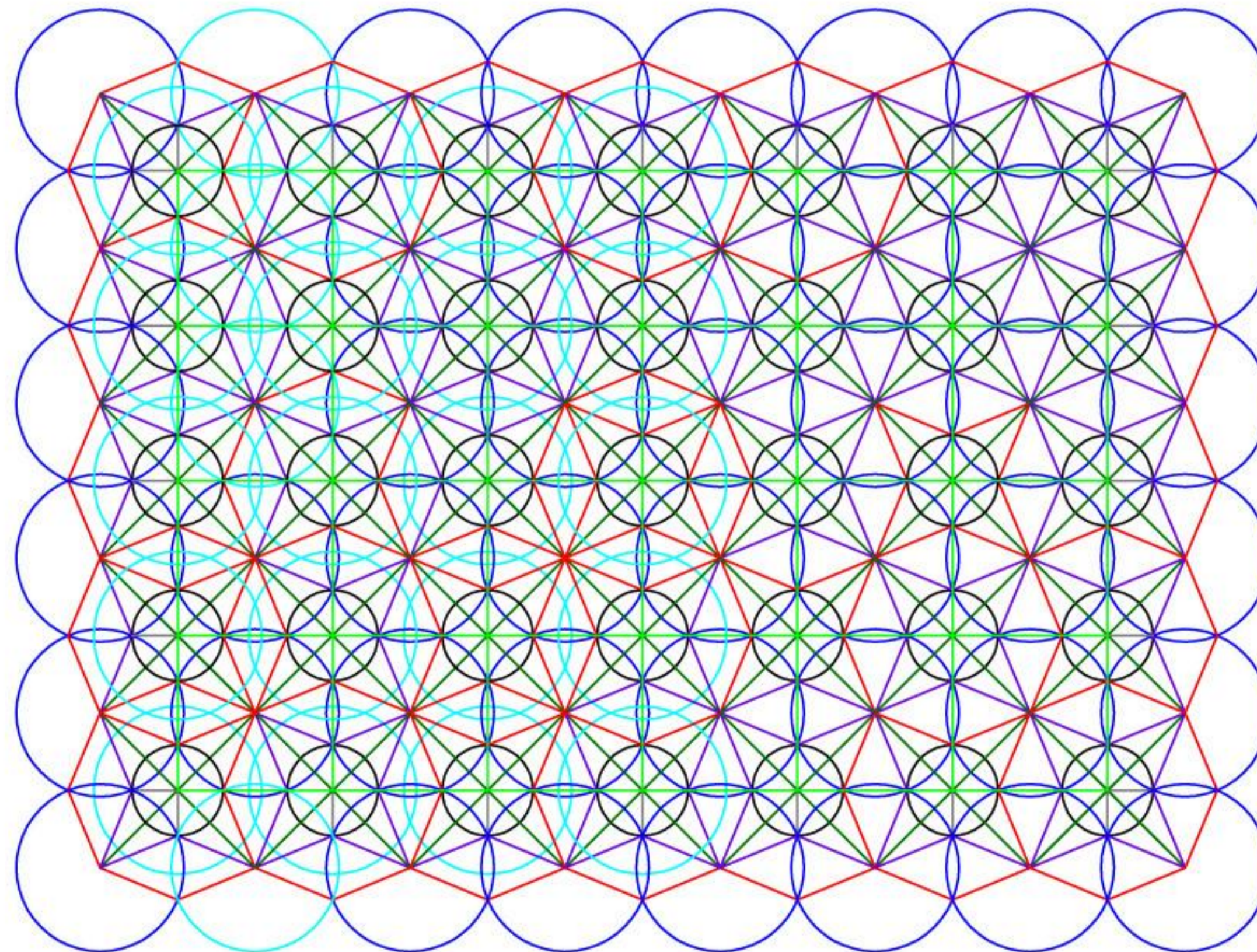
- X distance mezi řídicími body
- R poloměr osmiúhelníku
- R_2 poloměr nového polygonu (čtyřúhelníku) - hledaná neznámá

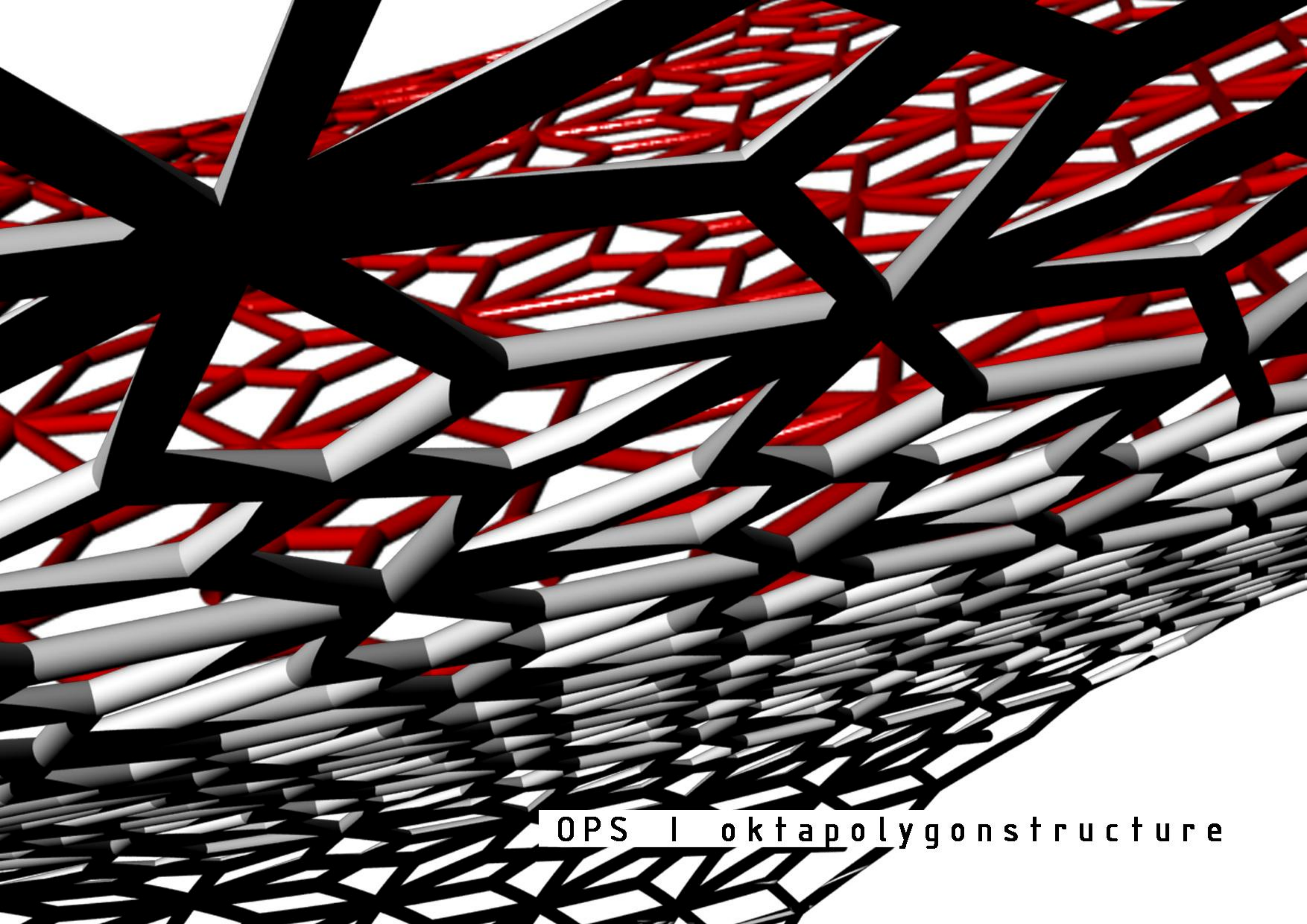
$$R_2 = X - R$$

$$R_2 = X - X * \sin(45^\circ)$$

rozbor ornamentu

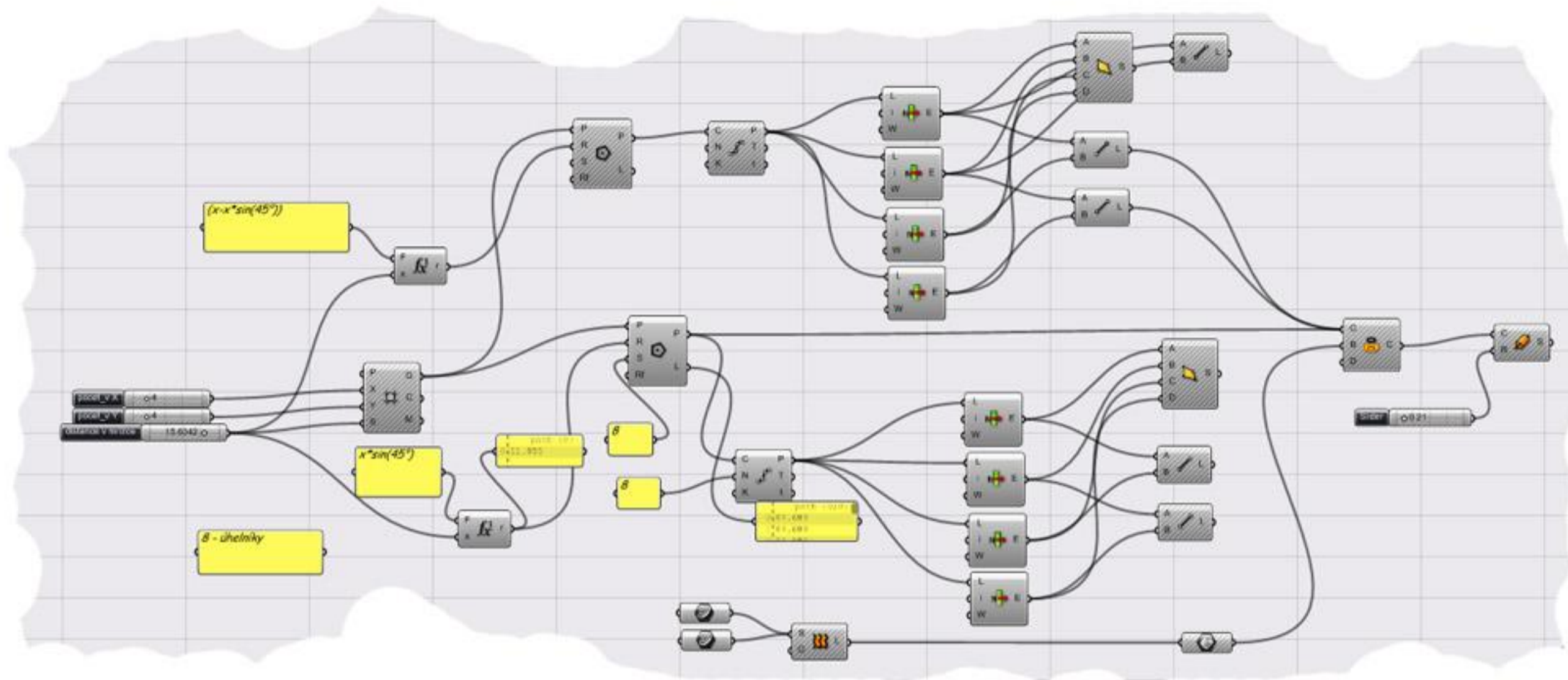
Během rozboru ornamentu jsem našel i další souvislosti, pravidla a různá závislá zobrazení.
Ovšem tyto regule jsem již nepotřeboval znát pro vytvoření parametrického návrhu v programu Grasshopper.
Možná sami najdete tyto pravidla v následujícím obrázku. :-)





OPS | oktapolygonstructure

vytvořený skript v programu Grasshopper 0.6.0012

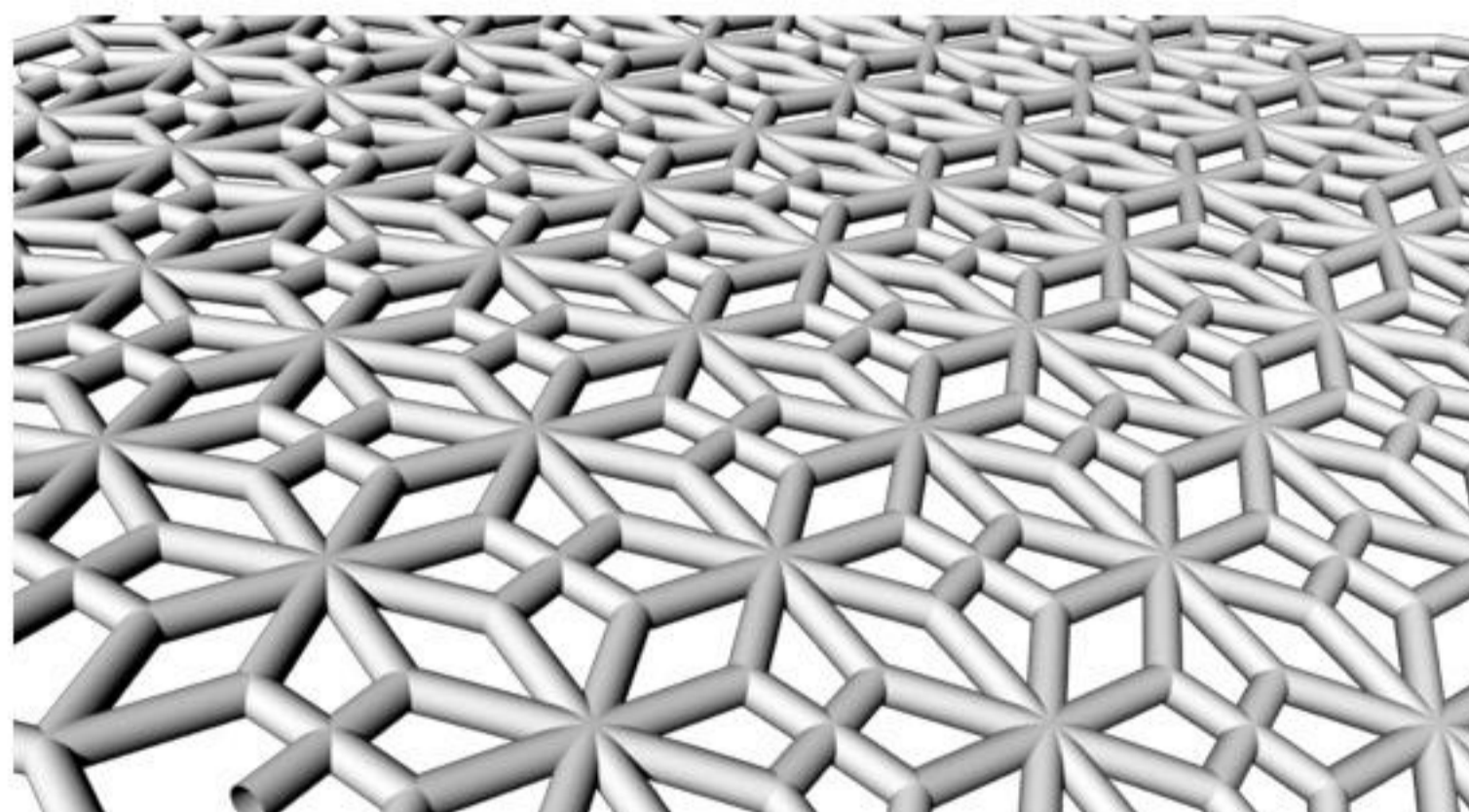


Program Grasshopper 0.6.0012 je zásuvný skriptovací modul programu Rhinoceros 4.0. Výhoda navrhování v tomto modulu spočívá v tom, že v reálném čase vidíme v Rhinu, co se děje s naším virtuálním modelem, pokud měníme jeho parametry. Parametry a celý skript vychází z určitých pravidel (regulí), které bychom měli znát před začátkem tvorby v tomto programu. Následně lze v programu Grasshopper využít již přednastavených funkcí, ale je možné si skripty psát sám pomocí Vb skriptu.

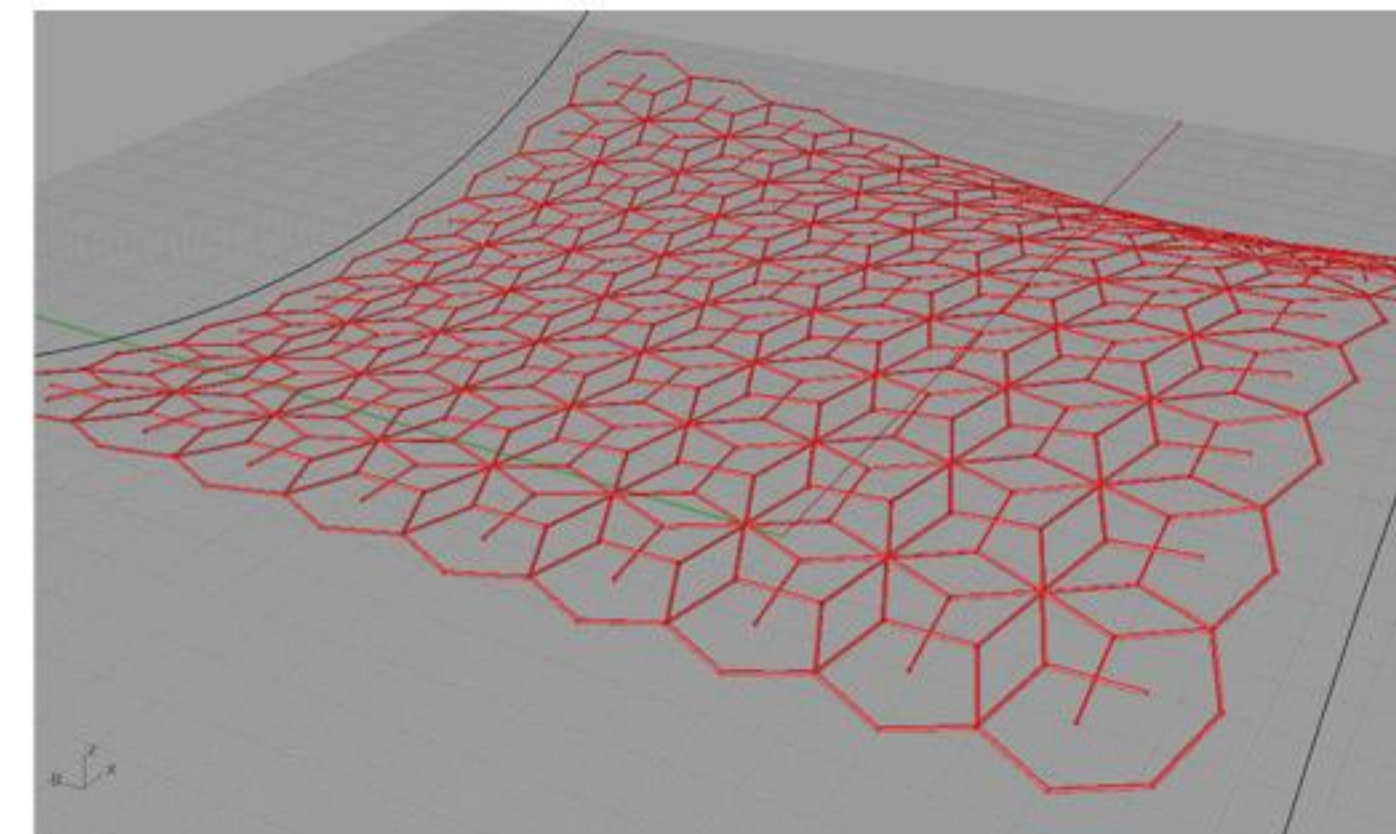
V mém případě jsem si nejprve udělal rozbor mého obrazce, tyto pravidla jsem pak využil k vytvoření parametrického modelu tohoto ornamentu.

Po získání výsledku, který se mi líbil jsem převedl geometrii z programu Grasshopper do programu Rhinoceros, kde jsem pomocí zásuvného modulu V-Ray vytvářel rendry hotové struktury

pohled na hotový skript



Rendr hotové struktury



Geometrie vytvořeného skriptu se zobrazuje v reálném čase v Rhinoceros

vytváření skriptu v programu Grasshopper 0.6.0012

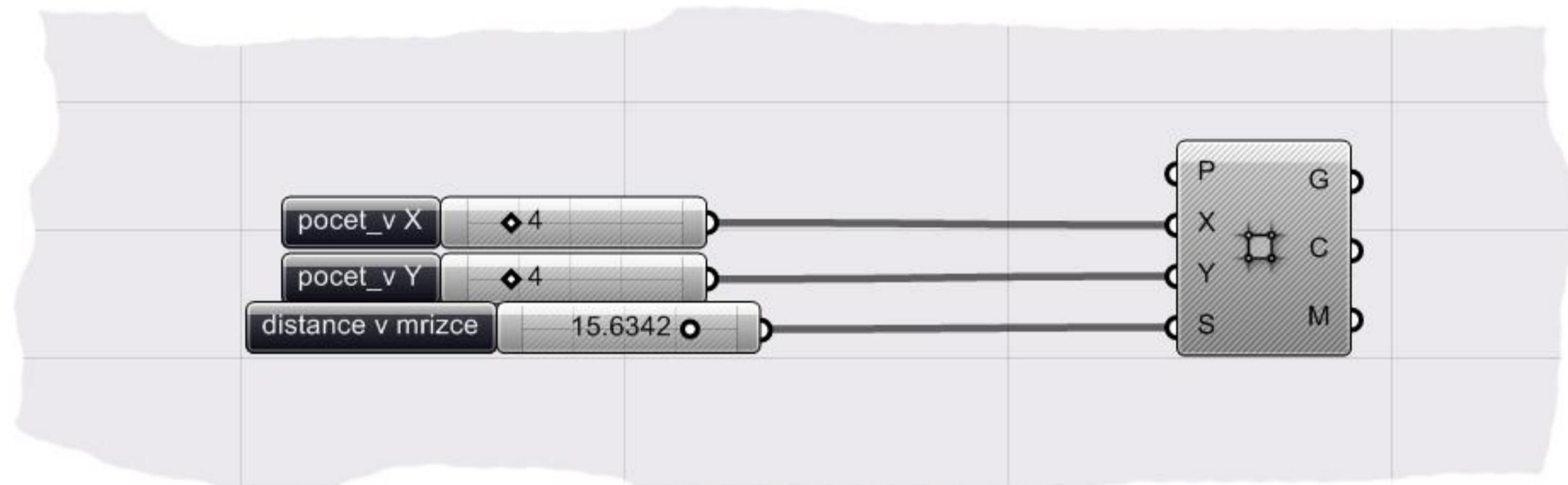
Funkcí Grid Rectangular jsem vytvořil pravidelnou mřížku bodů, v našem případě to jsou řídicí body.
Na levé straně ikony této funkce se nalézají vstupní data:

- X - počet kroků v souřadnicích X
- Y - počet kroků v souřadnicích Y
- S - vzdálenost mezi body

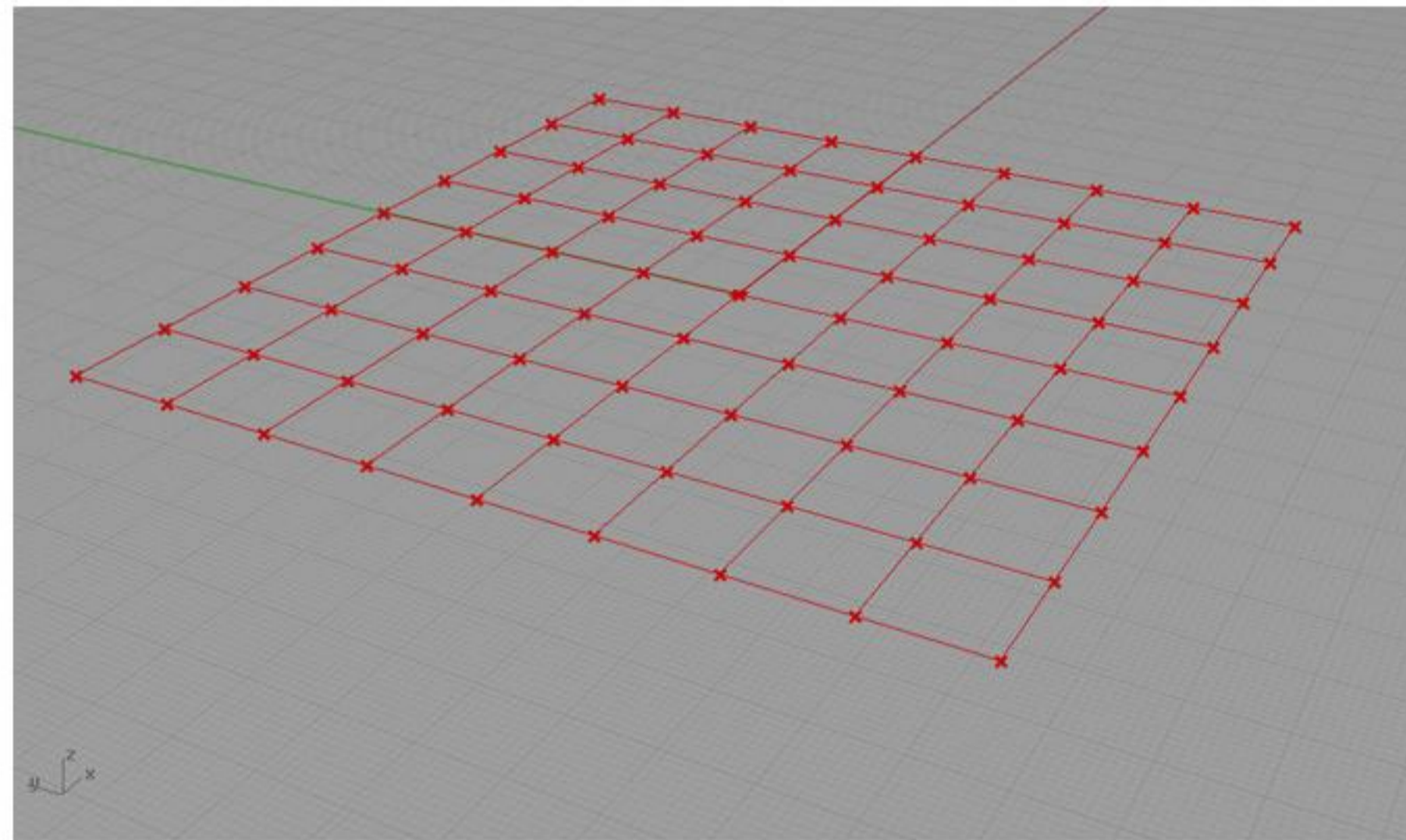
K těmto vstupním datům jsem připojil parametry (slidery), pomocí nichž budu moci během celé doby měnit vstupní data této základní sítě řídicích bodů.



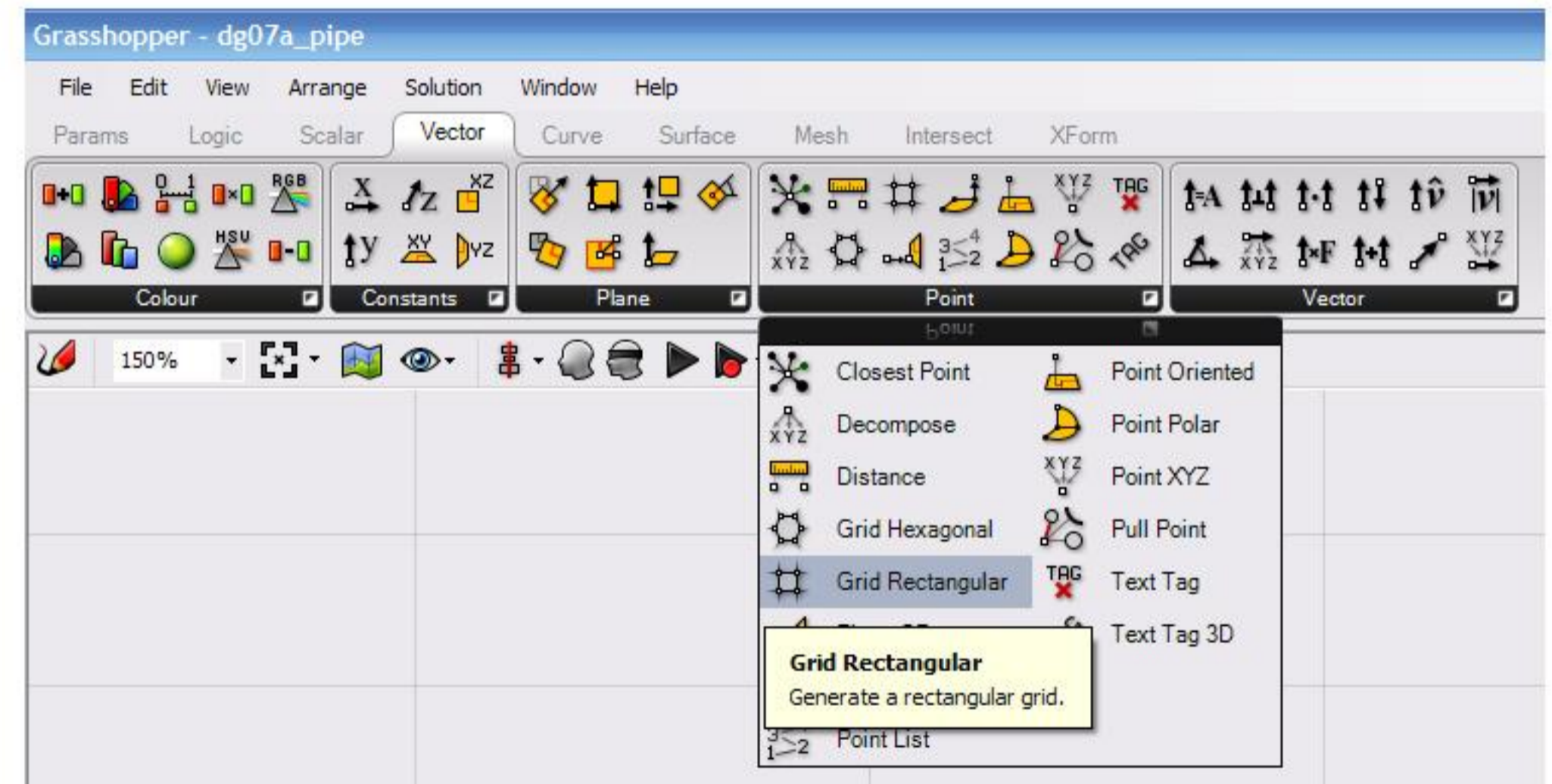
Editace slideru



použití funkce Grid Rectangular v Grasshopperu

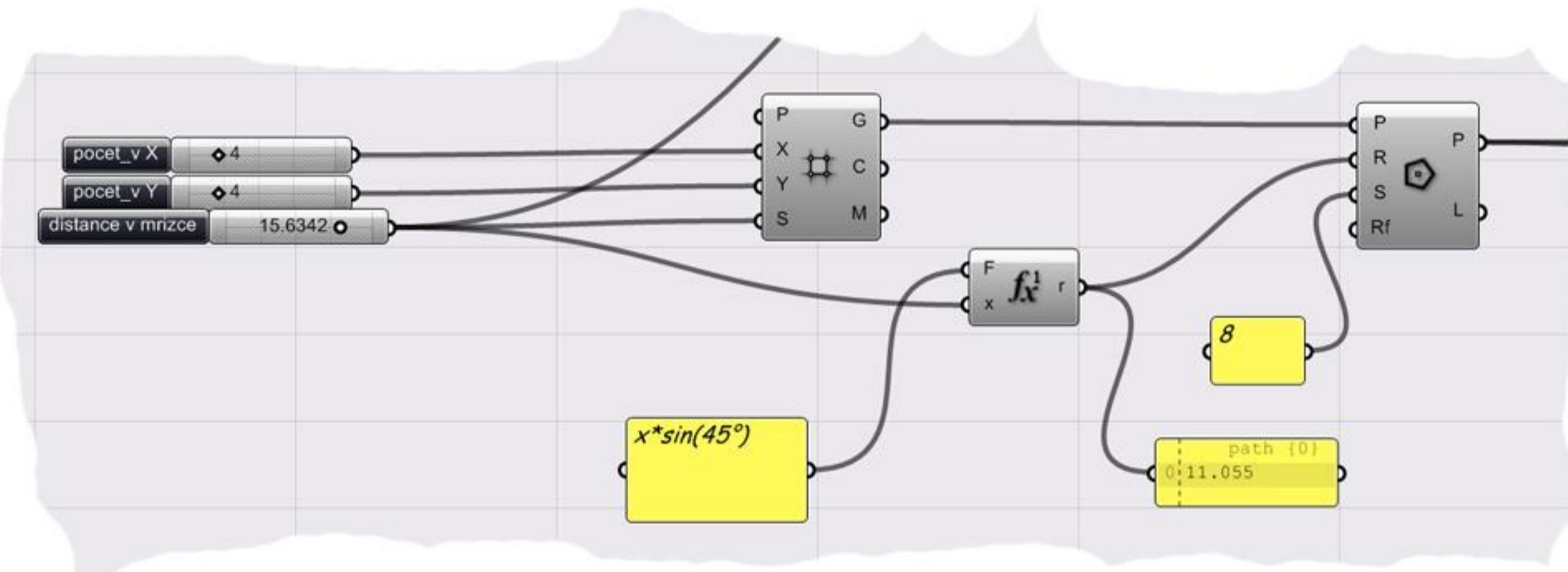


zobrazení v reálném čase v Rhinu



Funkci Grid Rectangular najdeme v záložce Vector / Point / Grid Rectangular

vytváření skriptu v programu Grasshopper 0.6.0012

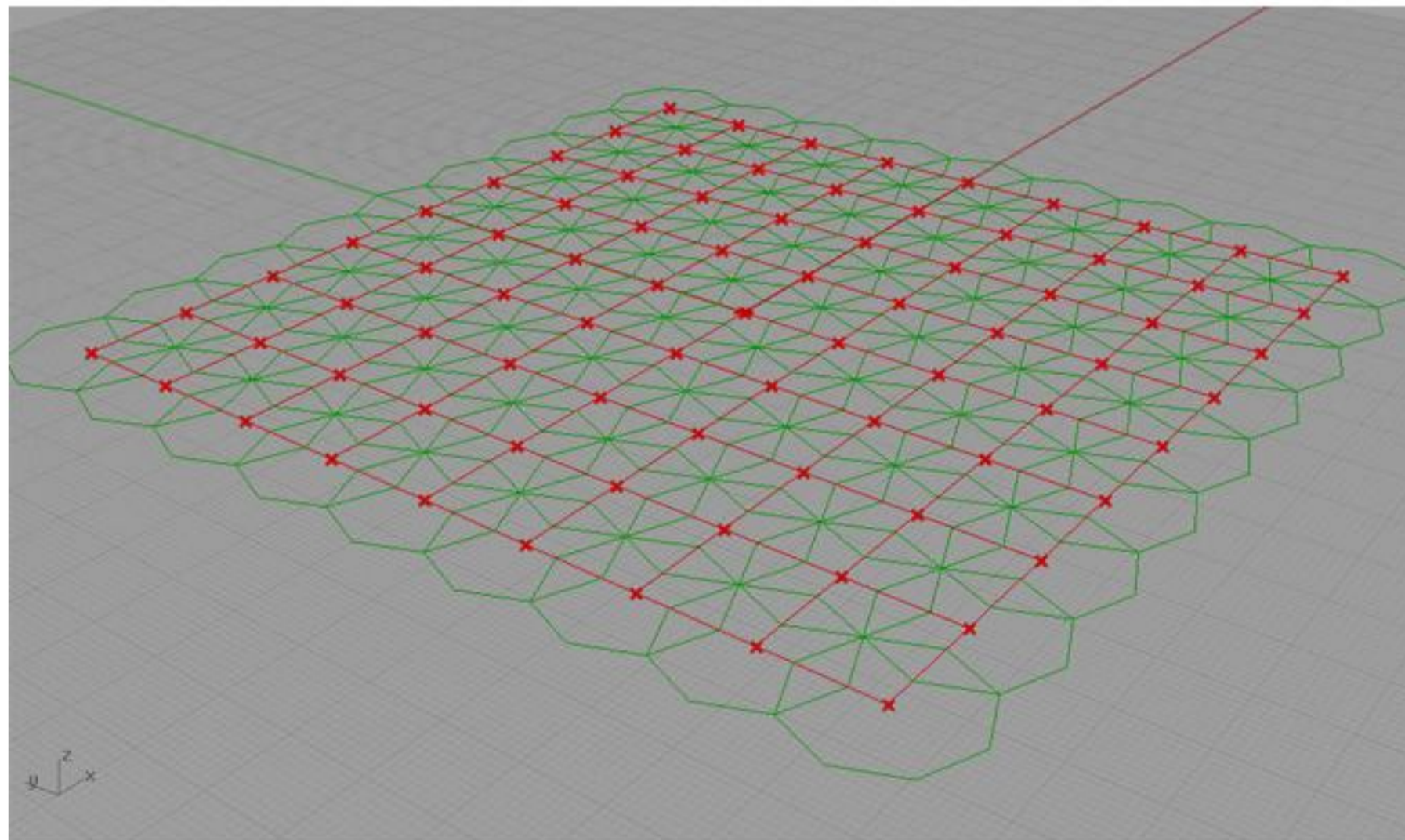


Vytvoření osmiúhelníka, proběhlo pomocí funkce Polygon (v záložce Curve / Primitive / Polygon).
Jako vstupní data zde jsou :

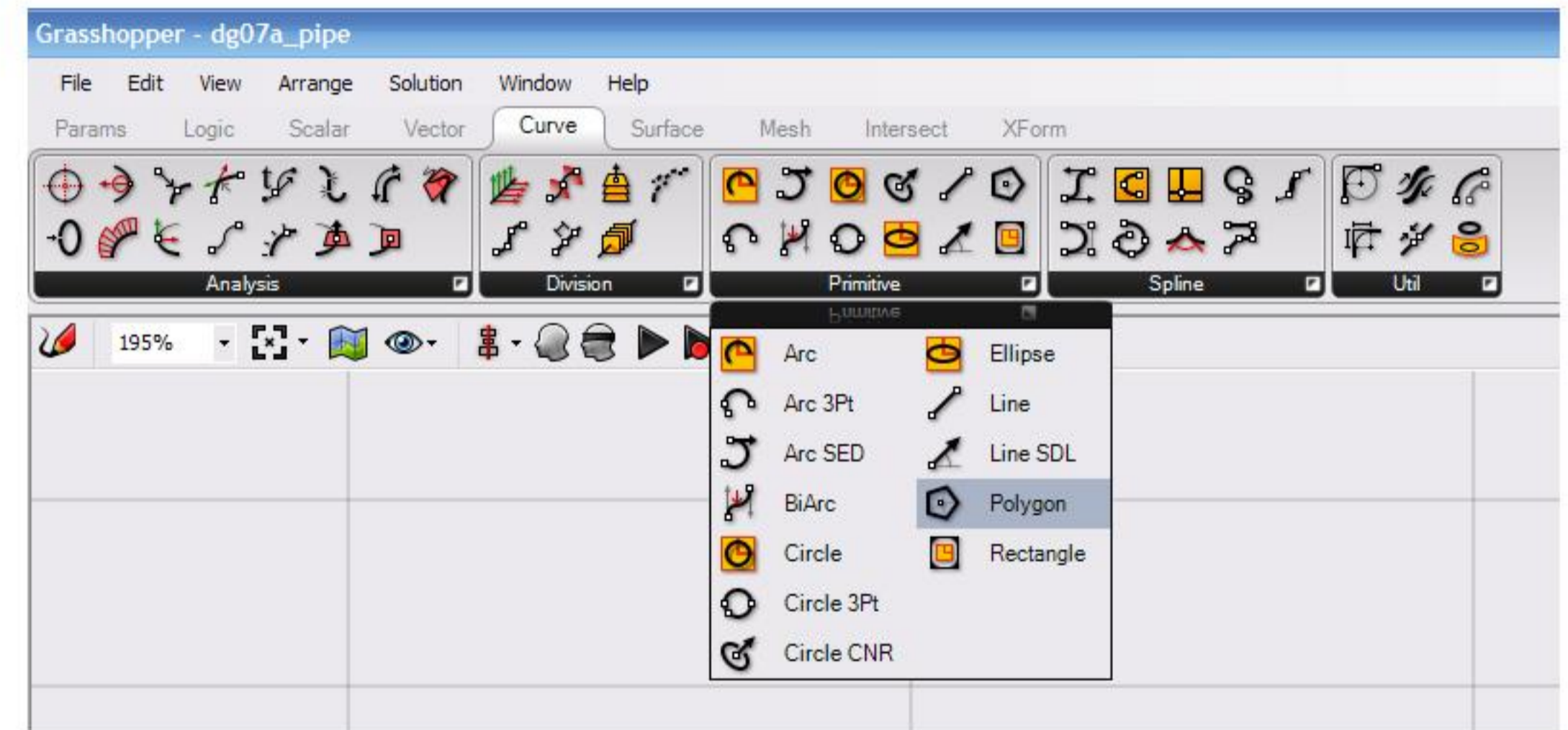
- P - plane - stejná jako základní mřížka
- R - poloměr polygonu - závislá na distanci v mřížce, ale protože v našem ornamentu je pravidlo určující poloměr polygonu, museli jsme tuto rovnici vložit i do Grasshopperu.
Zde pomocí funkce F1, kterou najdeme v záložce Logic / Script / F1. Sem následně vložíme naši rovnici, aby se poloměr polygonu měnil v závislosti na změně distance mřížky.
- S - počet stran polygonu - my zadáváme 8

Máme tedy vytvořenou část ornamentu z osmiúhelníků.

vytváření osmiúhelníku

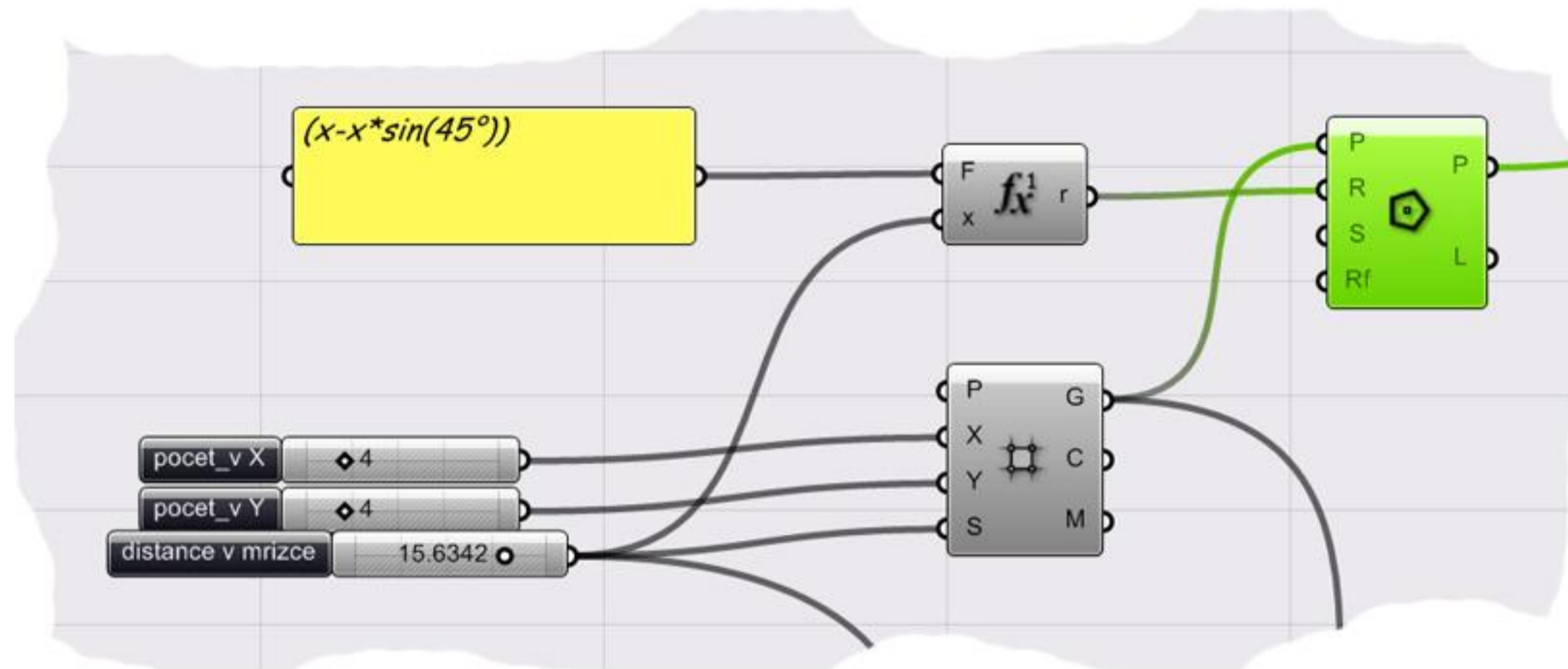


zobrazení v reálném čase v Rhinu



Funkci Polygon najdeme v záložce Curve / Primitive / Polygon

vytváření skriptu v programu Grasshopper 0.6.0012

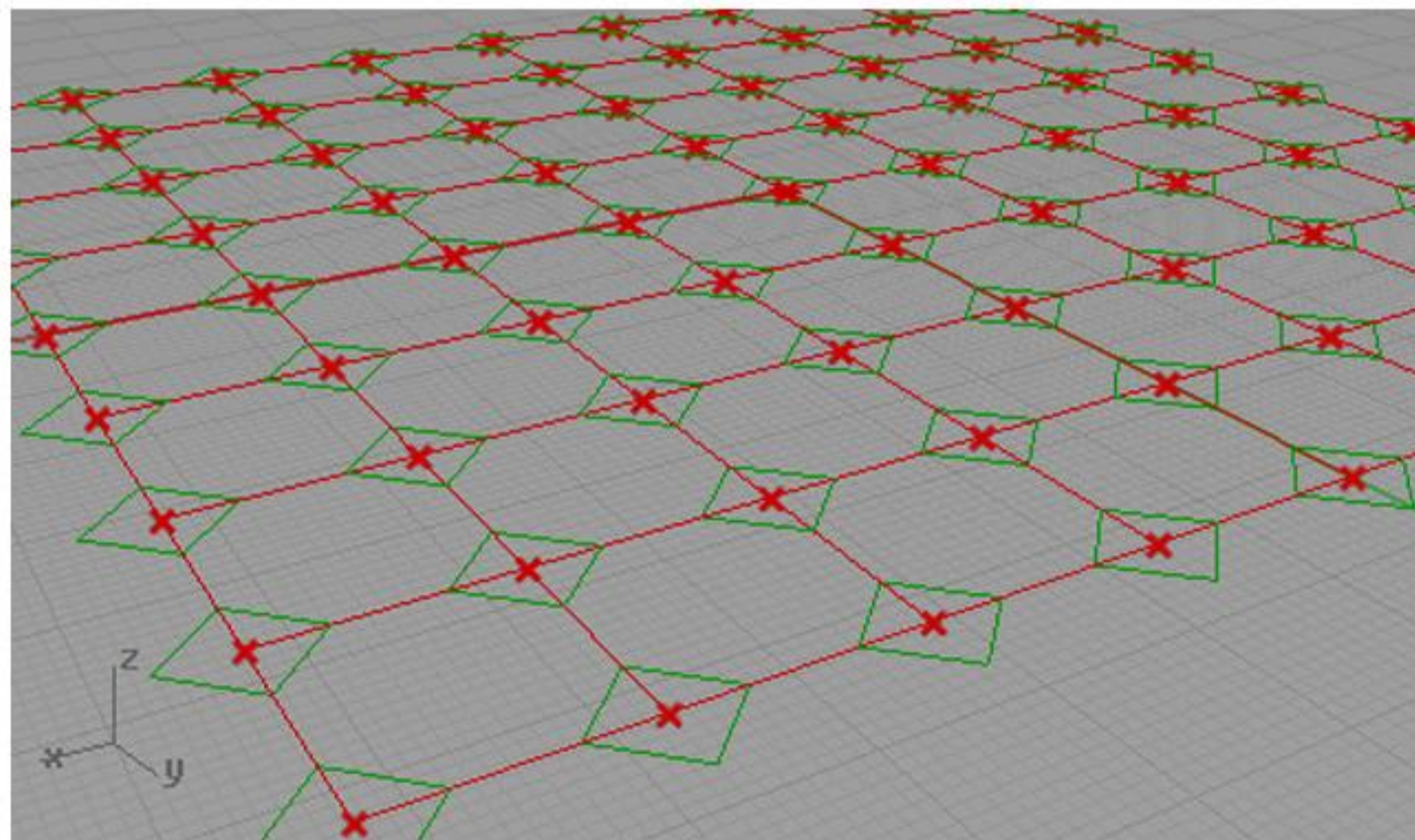


vytváření čtyřúhelníku

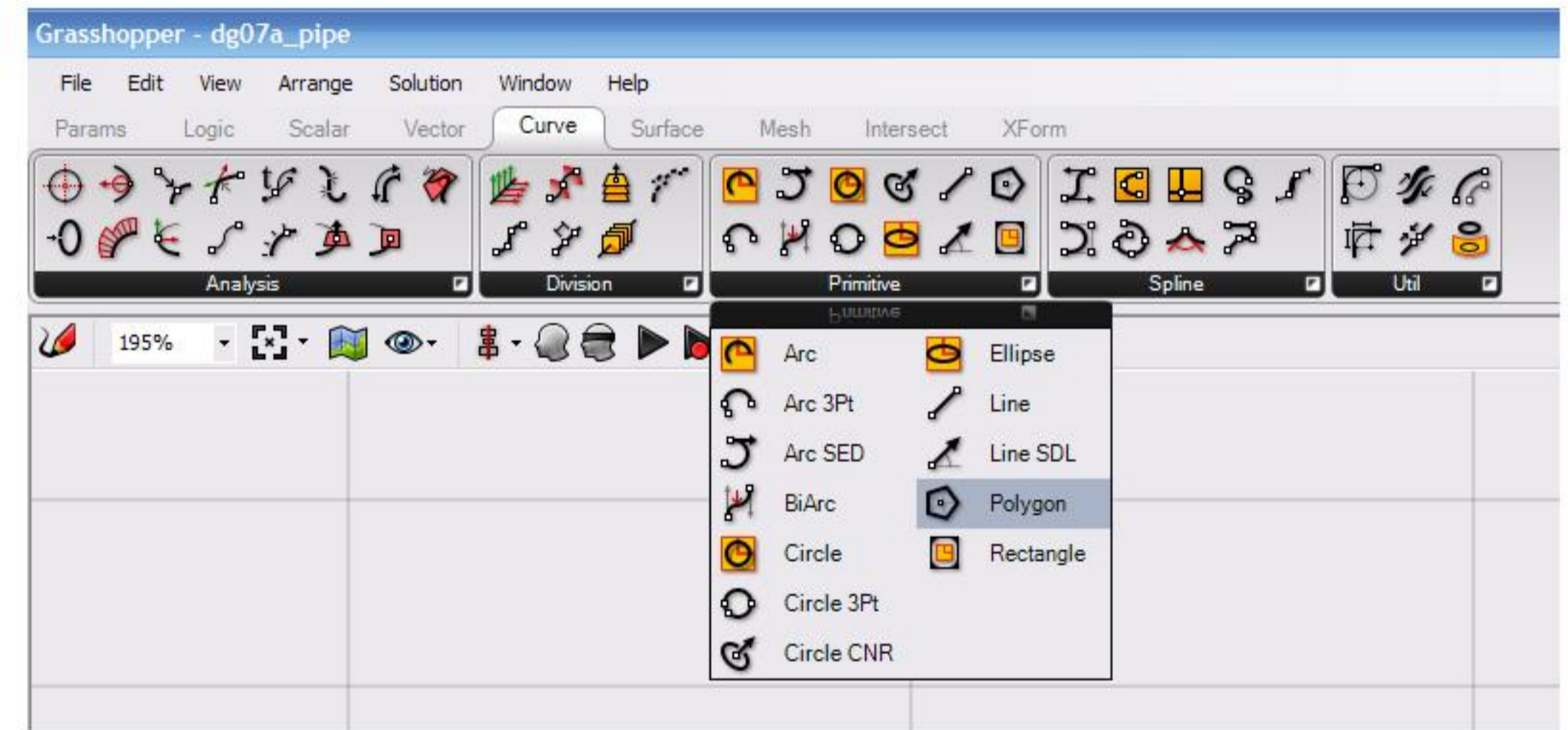
Vytvoření čtyřúhelníka, proběhlo pomocí funkce Polygon (v záložce Curve / Primitive / Polygon).
Postup je obdobný jako v předešlém případě.
Jako vstupní data zde jsou :

- P - plane - stejná jako základní mřížka
- R - poloměr polygonu - závislá na distanci v mřížce, ale protože v našem ornamentu je pravidlo určující poloměr polygonu, museli jsme tuto rovnici vložit i do Grasshopperu.
Zde pomocí funkce F1, kterou najdeme v záložce Logic / Script / F1. Sem následně vložíme naši rovnici, aby se poloměr polygonu měnil v závislosti na změně distance mřížky.
- S - počet stran polygonu - my zadáváme 4

Máme tedy vytvořenou pomocnou část pro úsečky..

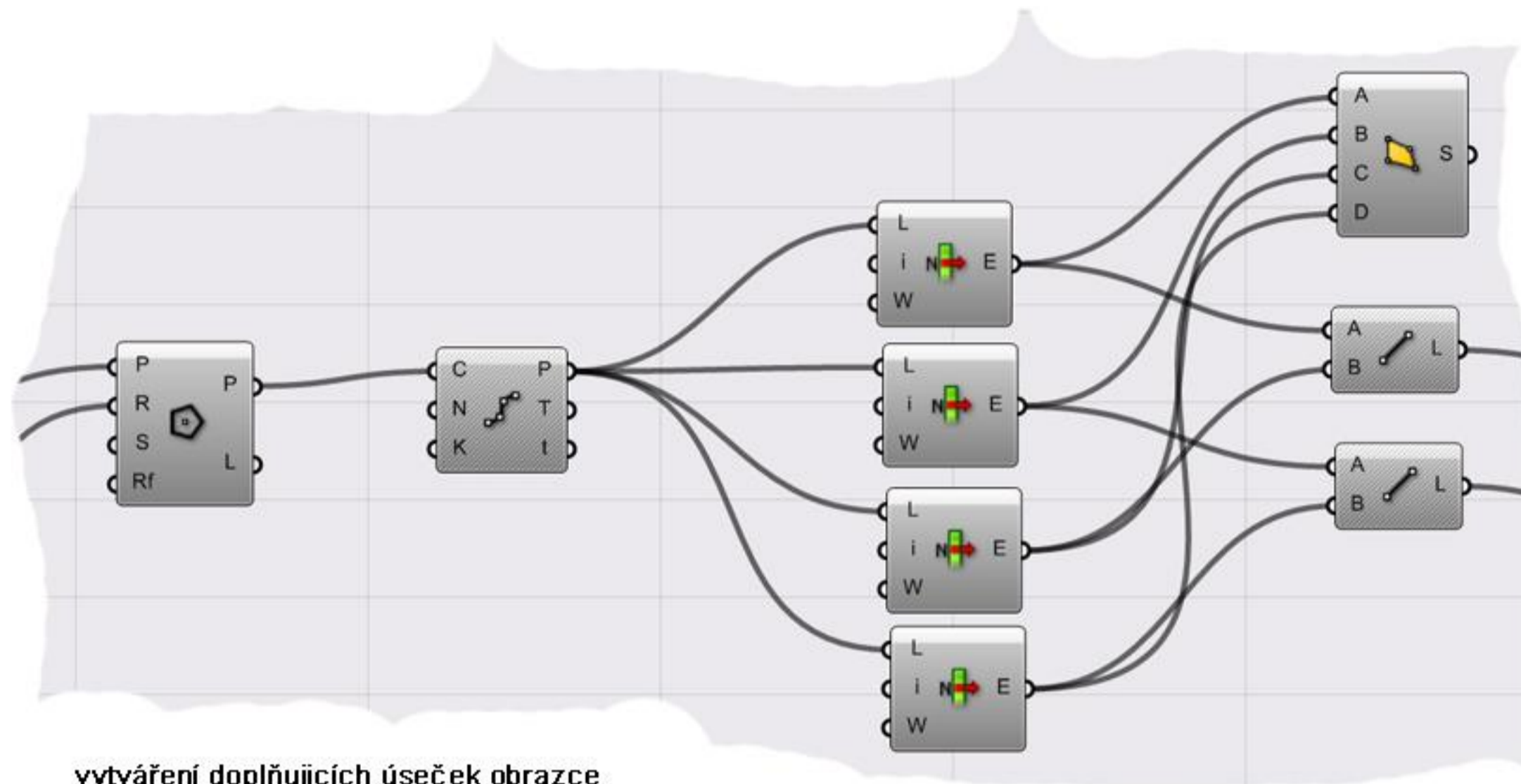


zobrazení v reálném čase v Rhinu



Funkci Polygon najdeme v záložce Curve / Primitive / Polygon

vytváření skriptu v programu Grasshopper 0.6.0012



vytváření doplňujících úseček obrazce

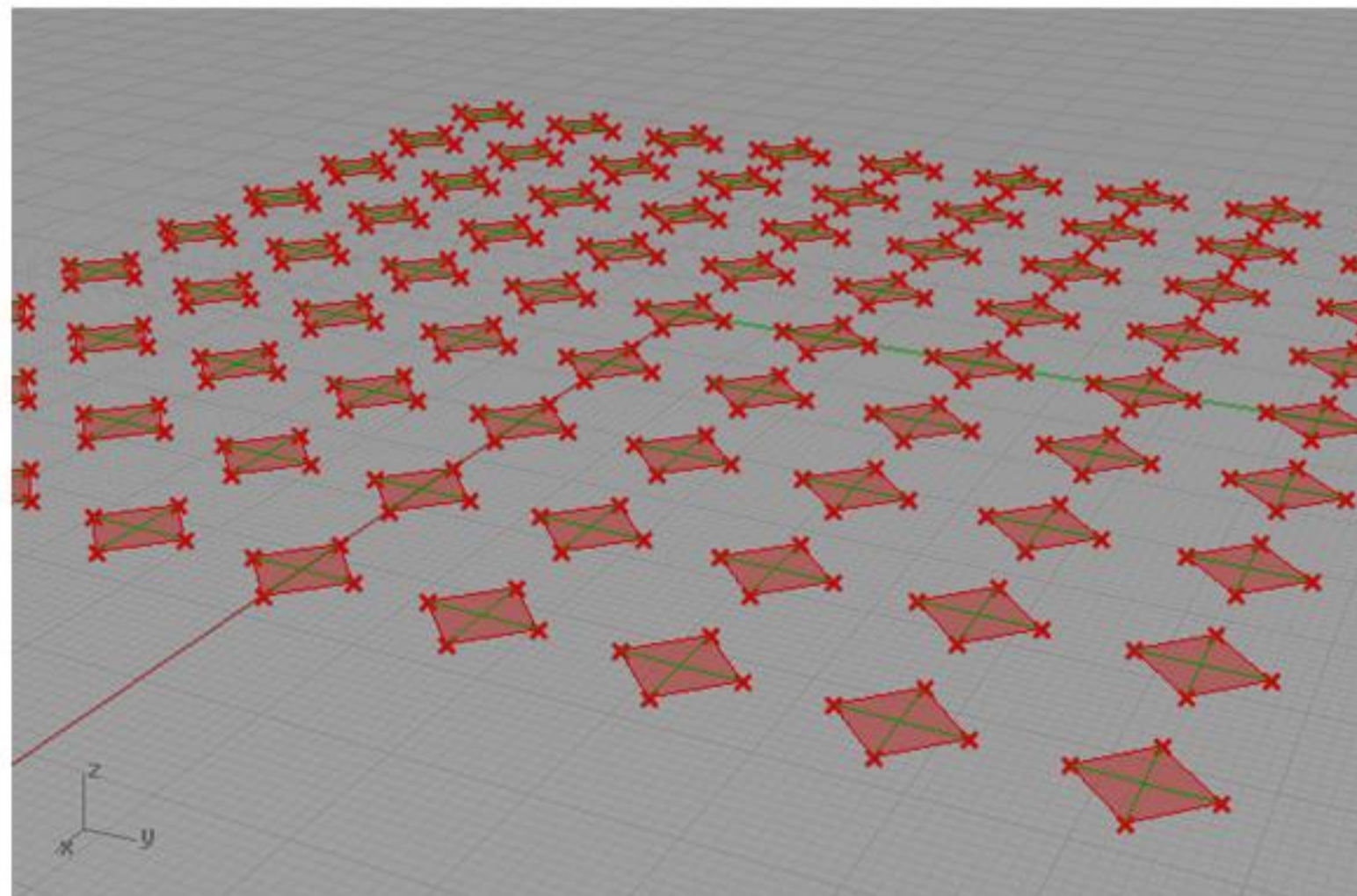
Vytvořený čtyřúhelník jsem si pomocí funkce Divide Curve (v záložce Curve / Division / Divide Curve) rozdělil tento polygon na 4 stejné úsečky. Tak jsem získal jednotlivé vrcholy tohoto polygonu.

Na výstup P (tedy points) jsem vložil funkce List Item (v záložce Logic / List / List Item). Tímto vlastně označíme jednotlivé vrcholy, na vstupu i pak postupně zadáme 0, 1, 2, 3, čímž vytvoříme pořadí těchto bodů.

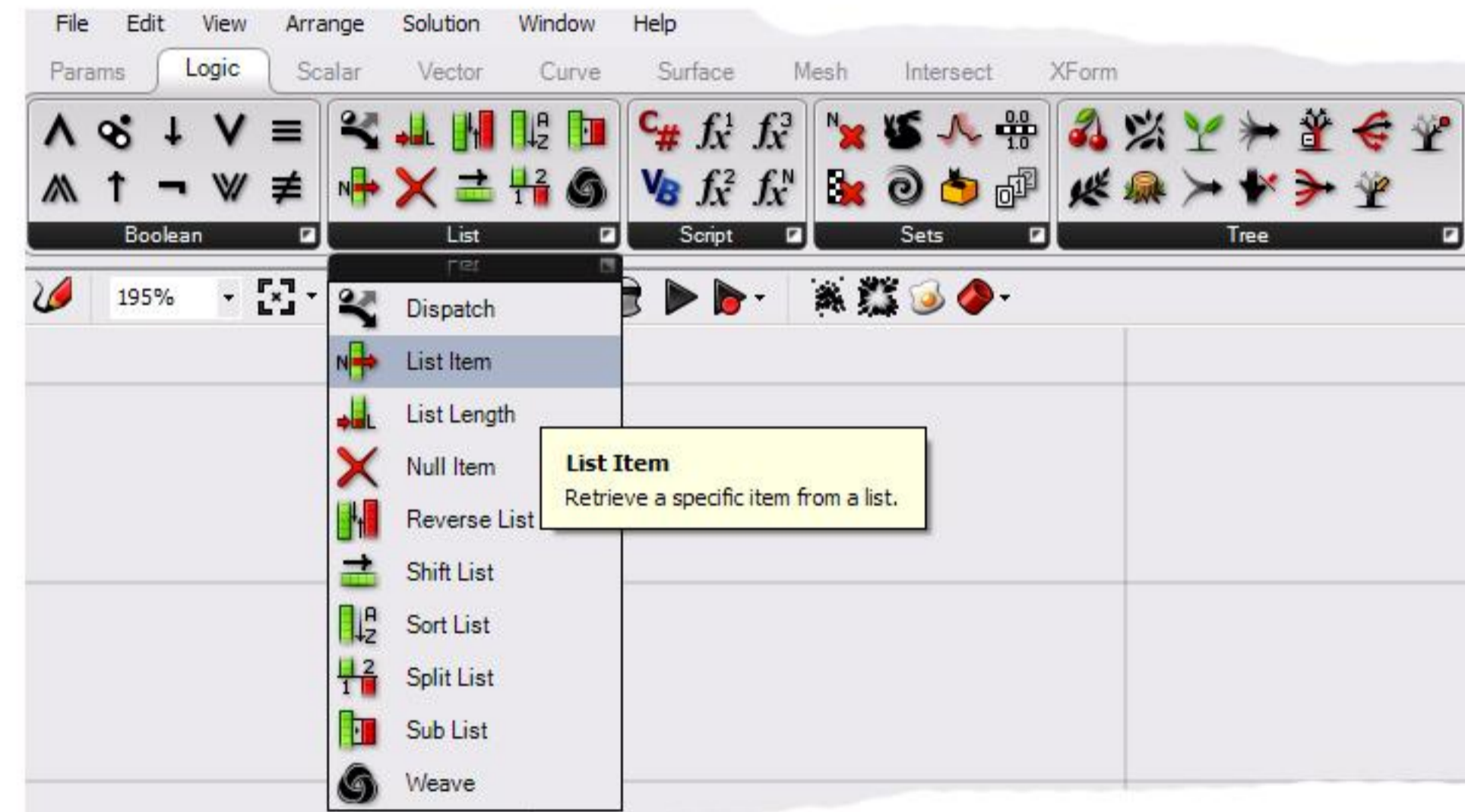
Poté vložíme funkci Line (Curve / Primitive / Line). Abychom vytvořili úhlopříčky musíme spojit body 0 a 2, z čehož nám vznikne první úhlopříčka, druhá vznikne spojením bodů 1 a 3.

plocha z těchto bodů vznikne pomocí funkce 4Points Surface (Surface / Freeform / 4Point Surface).

Tímto krokem jsme parkicky vyřešili 2D parametrický ornament.

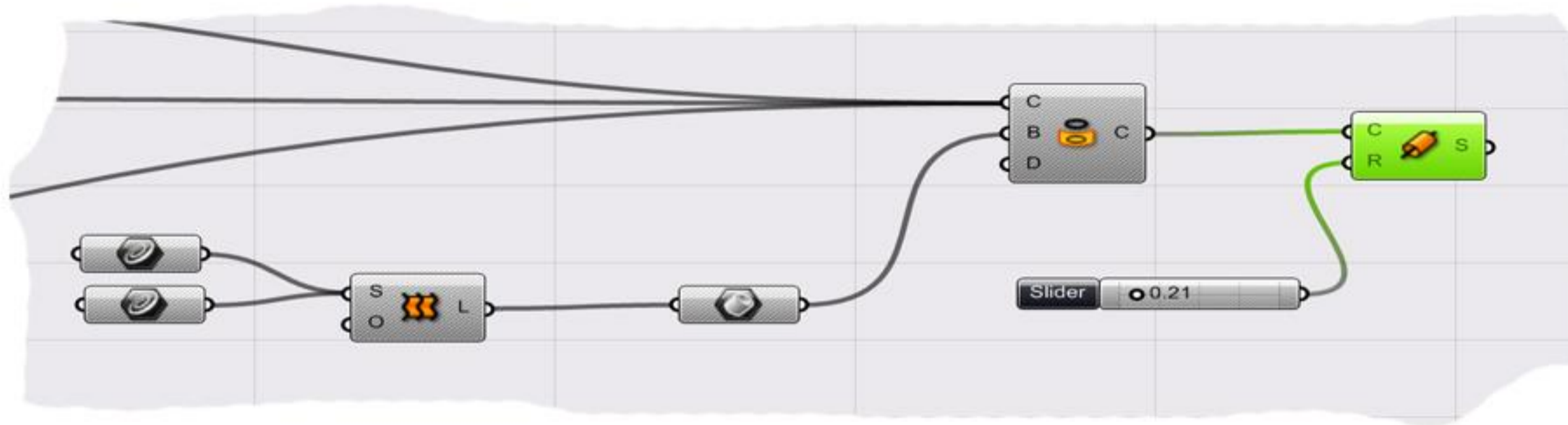


zobrazení v reálném čase v Rhinu



Funkci List Item najdeme v záložce Logic / List / List Item

další možná pokračování skriptu

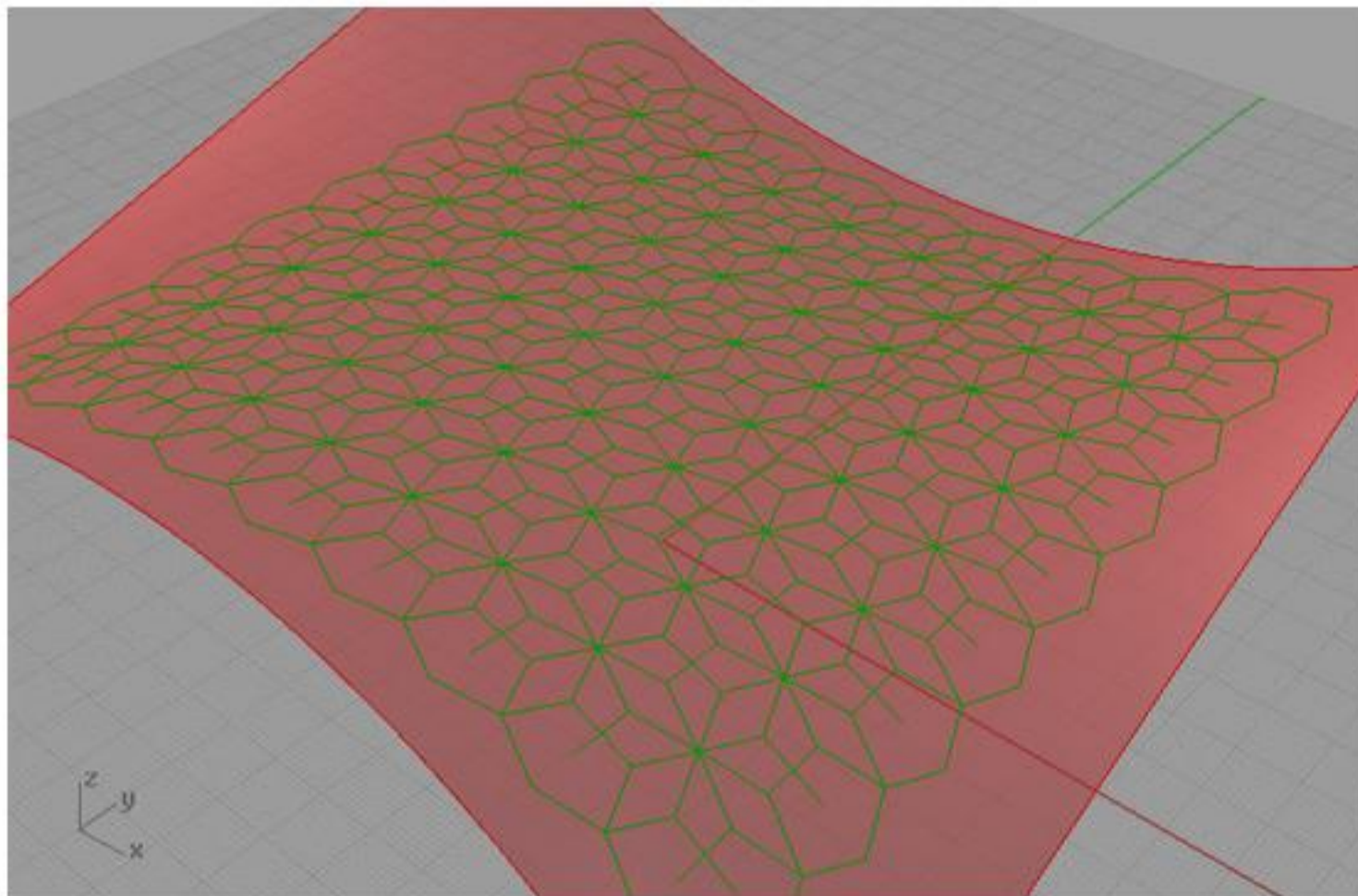


Nyní máme vytvořený 2D parametrický model. Dál se může skript začít vyvíjet mnoha směry, některé mohou být i velmi komplikované a jejich vytvoření v programu Grasshopper by bylo velice složité. Ovšem do budoucna stojí za prozkoumání.

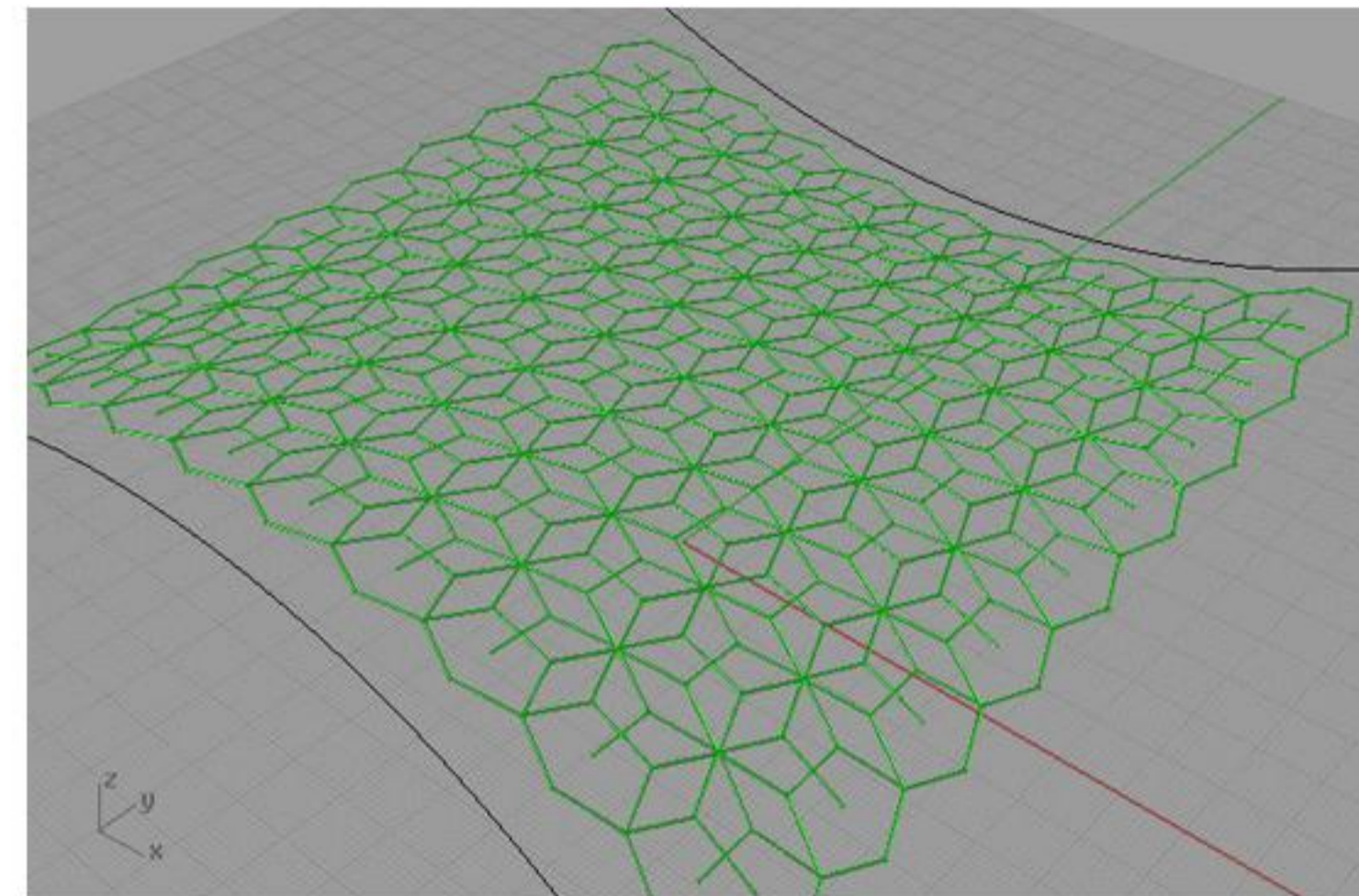
Já jsem se rozhodl tento ornament promítnout na plochu, která je řízena dvěma křivkami. Poté jsem nově vzniklé křivky potáhnul potrubím (podobné funkce jsou i v rhinu).

Pokoušel jsem se i o vytvoření dalších skriptů. Ošem buď bez větších úspěchů :-) nebo se mi nelíbil jejich výsledek.

konečná fáze skriptu

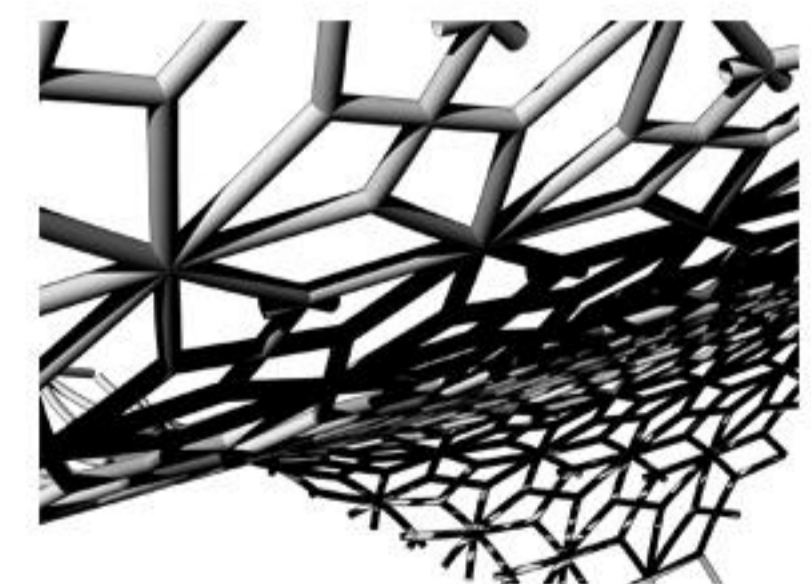
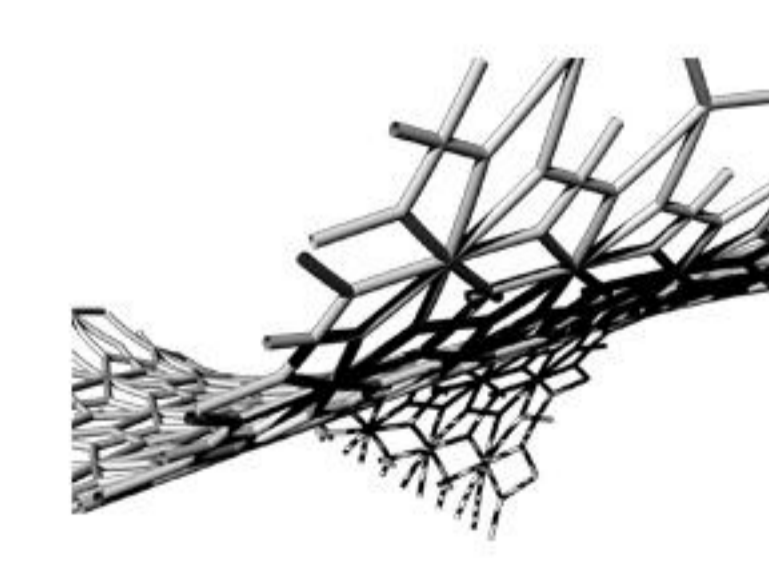
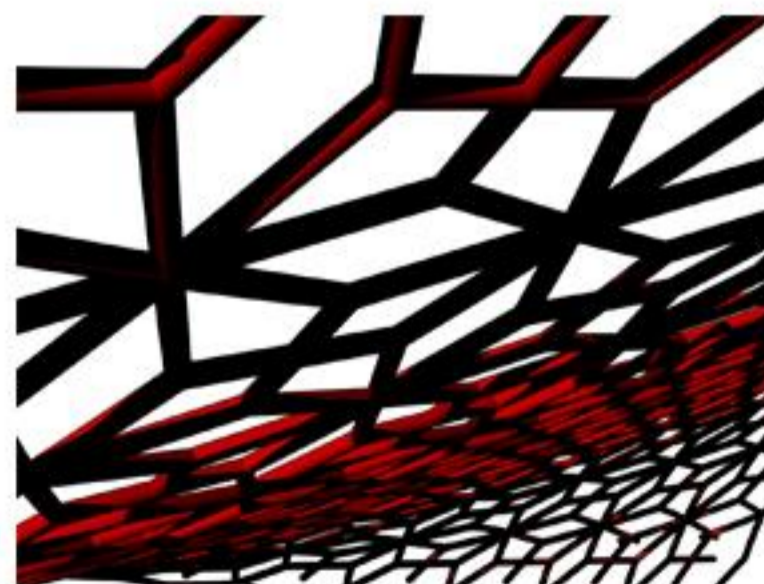
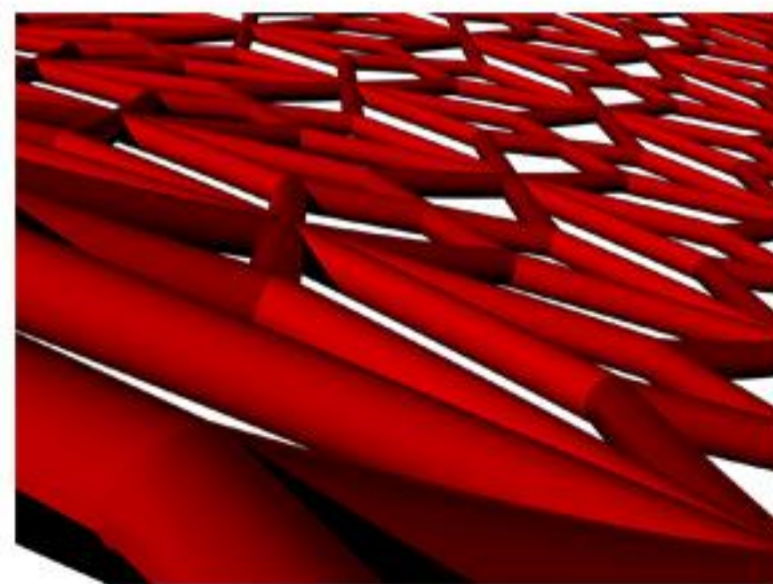
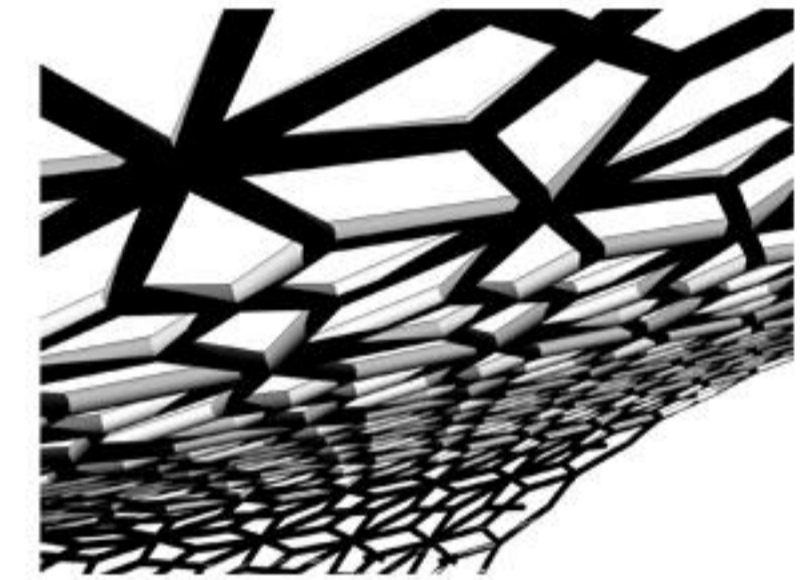
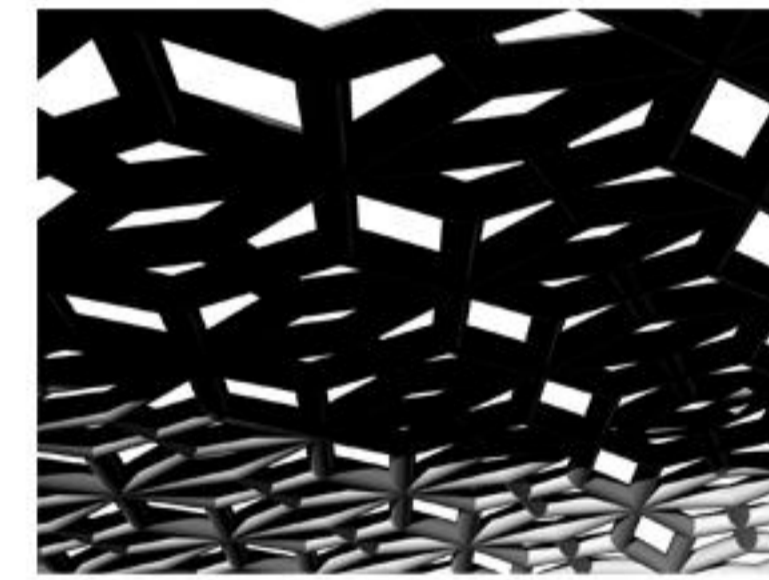
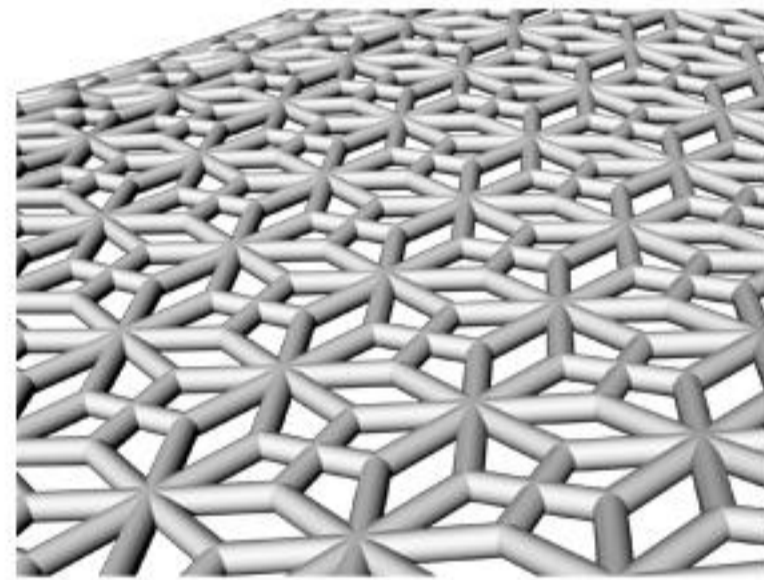
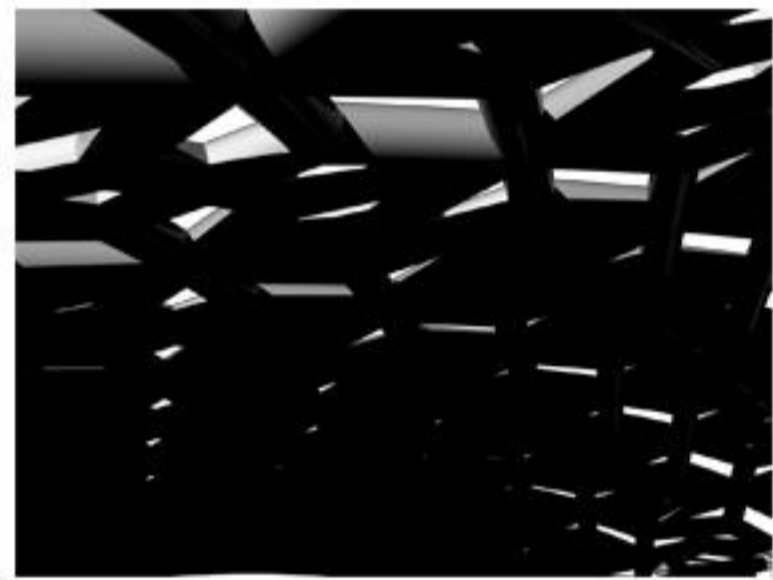
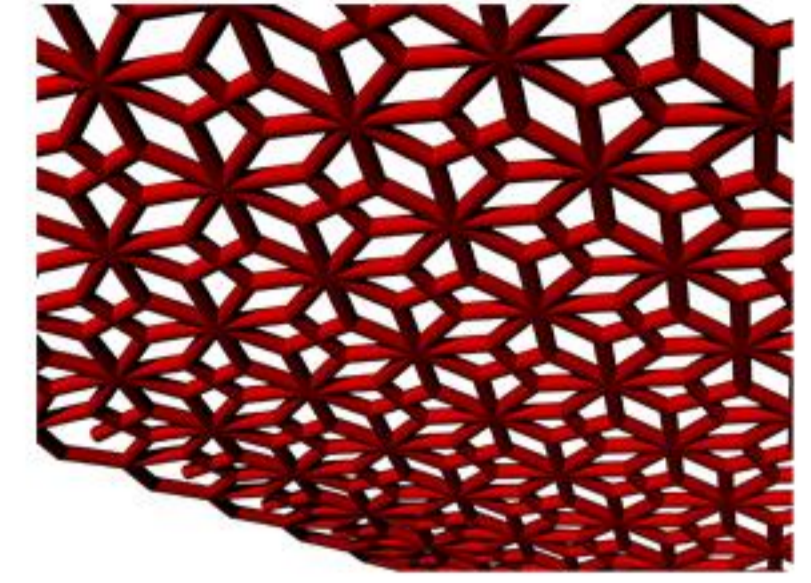
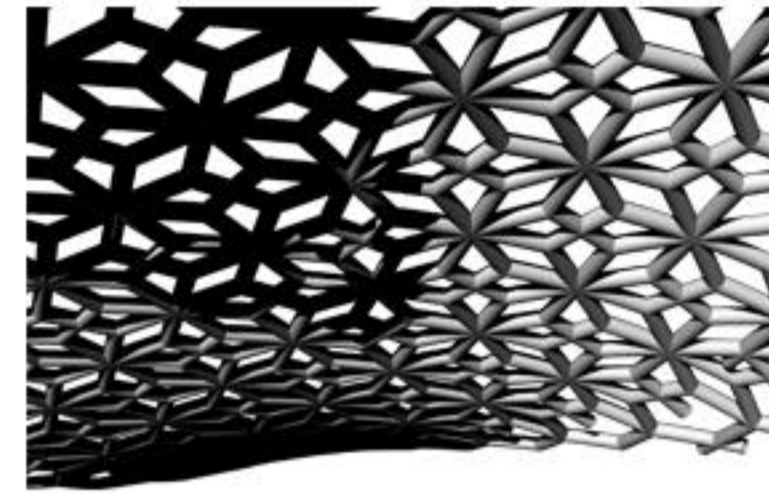
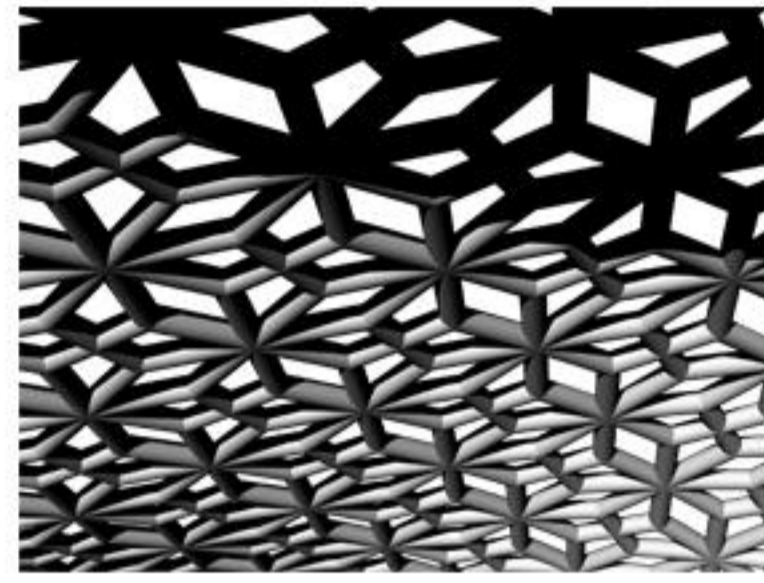
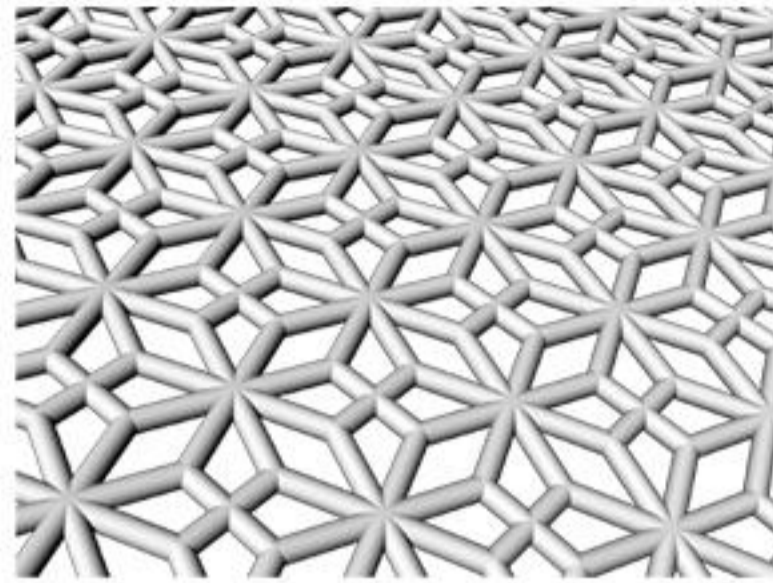


zobrazení v reálném čase v Rhinu

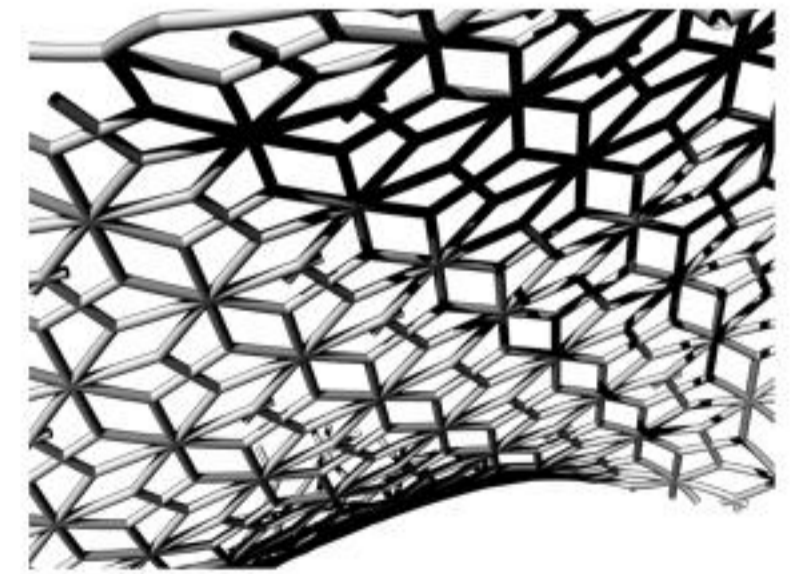
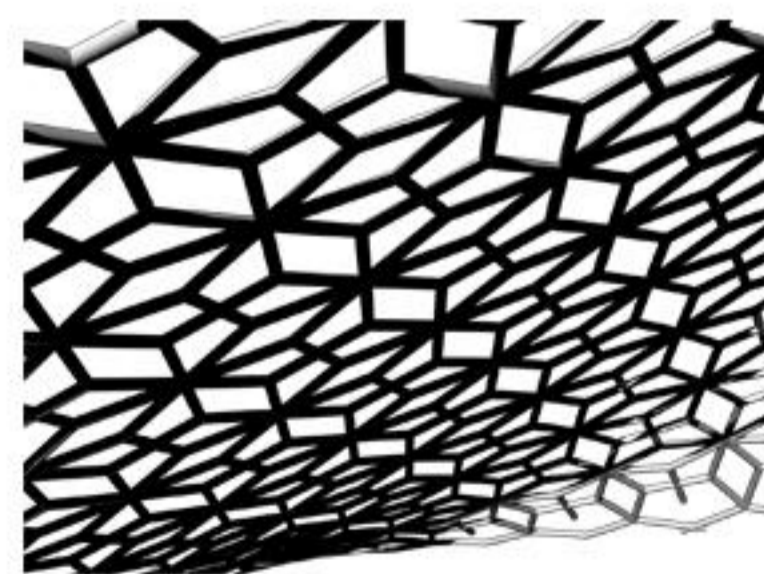
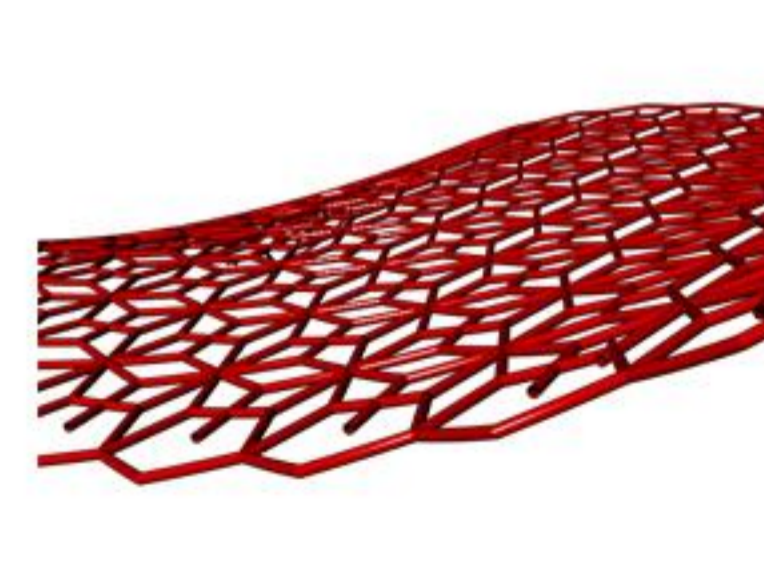
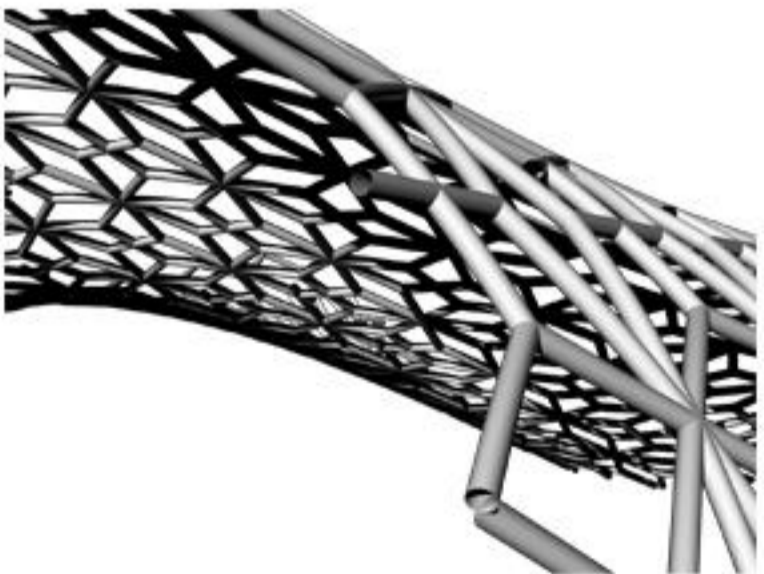
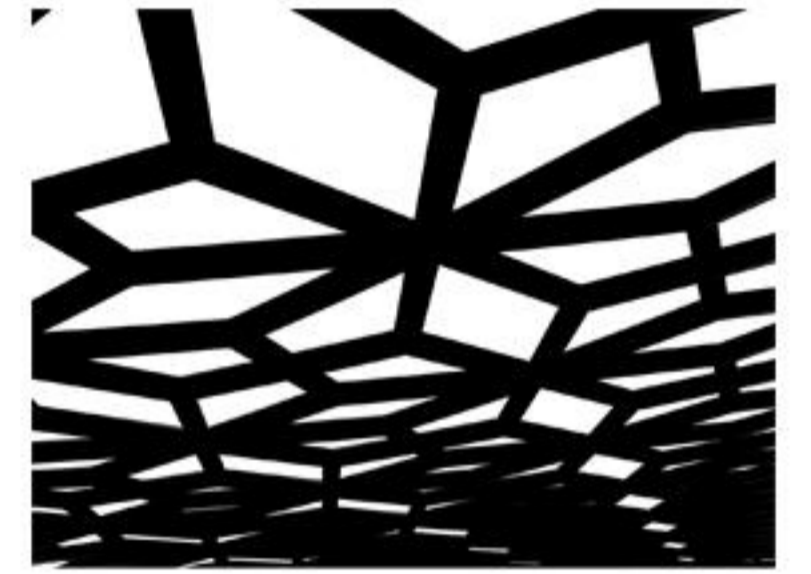
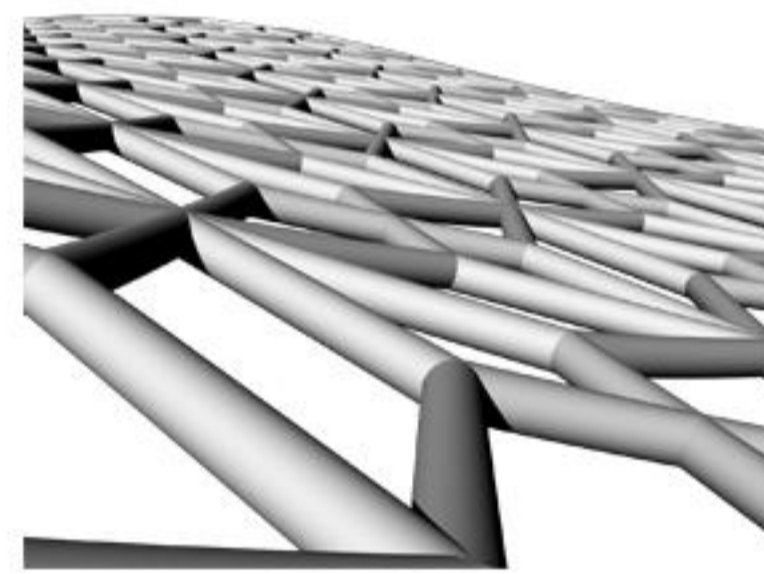
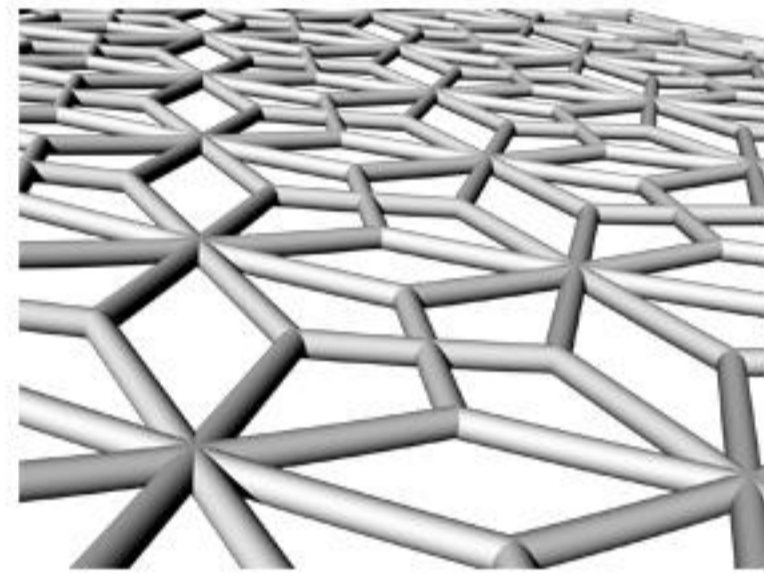
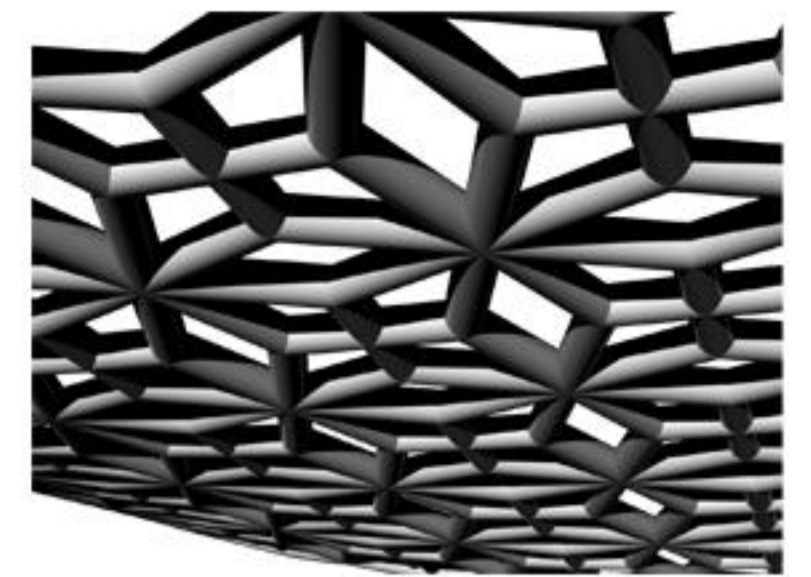
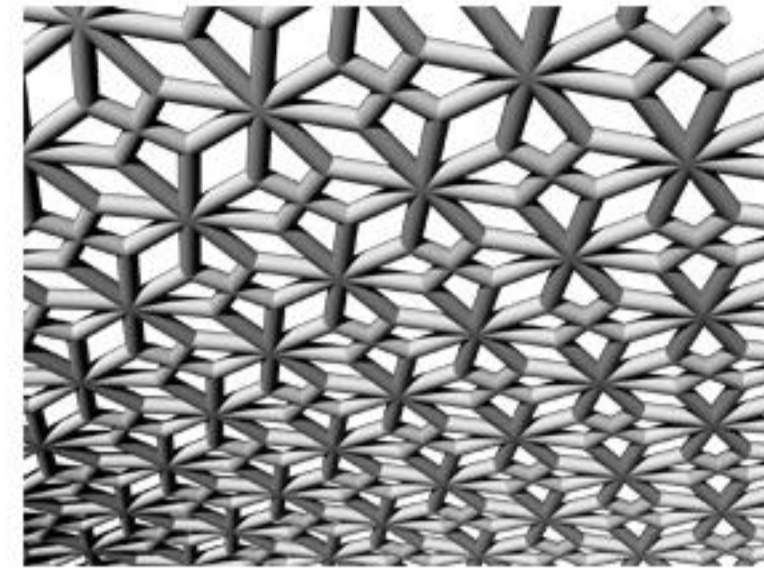
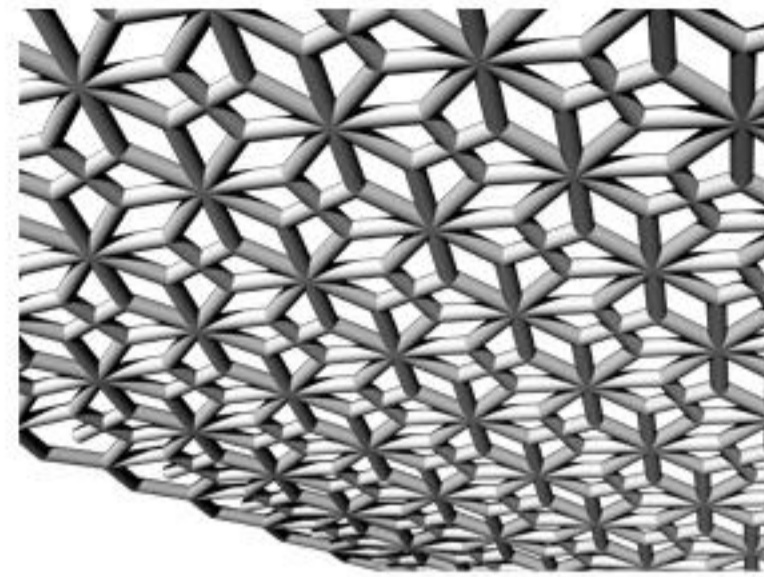
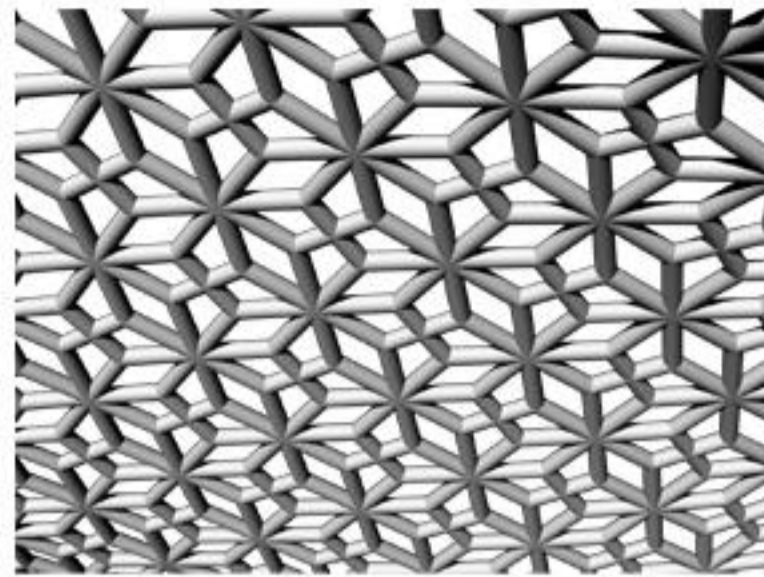


Funkci List Item najdeme v záložce Logic / List / List Item

vizualizace struktury



vizualizace struktury





OPS | oktapolygonstructure