東京大学大学院新領域創成科学研究科
社会文化環境学専攻

2021 年度

修士論文

五軸 CNC 切削機を用いた
ひねり嵌合形式の木組みの開発
(Development of twist fitting type wooden joints using 5-axis CNC cutting machine)

2022 年 1 月 17 日提出
指導教員　佐藤　淳　准教授

宮野　公輔
Miyano Kousuke

目次

第一章　序論

1.1　研究背景

1.1.1　日本の木組み技術

日本の伝統木造社寺建築物に用いられている「木組み」の技術は、木材を精度高く刻むことにより金物を使用せずに材と材を組み上げることができる。この技術は宮大工によって何百年と発展し、様々な凹凸加工を施して工夫されたその仕口や継手の種類は 200 以上とも言われている（図 1-1）。

また近年 SDGs といった世界中で環境問題に対する関心の高まりから木造建築にも注目が集まっており、その中でも釘などの金物を一切使わない日本の伝統技術である「木組み」は特に注目が集まっていると言える。

図 1-1. 日本の伝統的な「木組み」の例
出典：木組みの美　岡部材木店　https://zaimokuten.com/ie/kigumi　（最終閲覧日 2021 年 9 月 20 日）

1.1.2　建築設計におけるデジタル技術の普及

近年デジタル技術の発達によりパラメトリックデザインやコンピューテーショナルデザインといったデザイン手法が普及し、建築分野においてもこれらの手法を用いた複雑な建築物の設計が実践されるようになった。また 3D プリンターやレーザーカッター、ロボットアームなどデジタルファブリケーションの普及により CAD 等のコンピュータ内のモデリングに留まらず現実に施工される事例も増加している（図 1-2）。

今まで作製が非常に困難であったものが作製可能になり、より自由なモノづくりが新たなイノベーションを生み出すとの期待から、デジタルファブリケーションへの期待は非常に高まっており、今回使用する五軸 CNC 切削機もその一つである。

図 1-2. パラメトリックデザインの実施例（当研究室設計の木組み棚「すすき」）

1.1.3　五軸 CNC 切削機の普及

デジタルファブリケーションの一つとされる CNC（computerized numerical control）切削機は XYZ の直線三軸で構成されるものが長い間主流であったが、近年この三軸に複数の回転軸を加えた五軸や多軸の CNC 切削機が普及し始めている。

一般的に三軸 CNC 切削機は複数の面を加工する場合、材料を手動で回転し固定しなおさなければならず、人為的な作業が加わることで、時間がかかる上、置き直した際に誤差が生じて製品精度にバラつきが生じることもある。五軸や多軸 CNC 切削機では機械の制御が三軸のものに比べ煩雑になるものの、前述のような問題を避けることができる（図 1-3）。

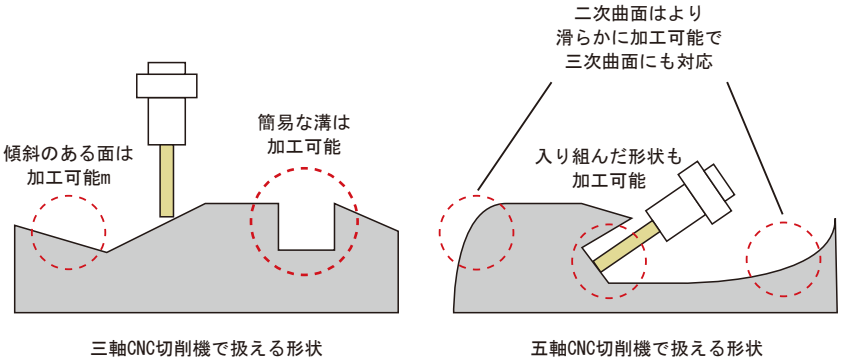


図 1-3. 三軸 CNC 切削機と五軸（多軸）CNC 切削機で削る形状の違い

この普及により建築生産の現場でも特定の形状の木材プレカットとして積極的に五軸（多軸）CNC 切削機が利用されるようになり、いままで三軸 CNC 切削機でも加工可能な簡単な形状の仕口や継手部材しか生産することができなかったがより複雑なものが生産されるようになった。

1.2　既往研究における本研究の意義

木組み技術について、宮大工個人のもつ技術力に大きく依存しない、人の手による木組みを「既存の木組み」と定義すると、少なくとも平安時代から続き 200 種類以上あるとされる「既存の木組み」技術は新たな木組みの開発余地は殆どないといえる。実際に「既存の木組み」に関する研究は、日本建築学会ホームページで確認できるものだけで 1975 年以降 300 件以上あり、殆どが構造特性、歴史、プレカットに関することで新規の木組みを提案するような研究は殆ど存在しない(2021 年 9 月 3 日、日本建築学会ホームページにて検索。キーワード「木組み」で 349 件、「仕口　木」で 363 件該当)。

一方で近年の木組みに関する研究は、コンピューテーショナルデザイン、デジタルファブリケーションの発展に伴ってそれらデジタル技術を関連させたものが増えており、五軸（多軸）加工機に限れば 2015 年に野口らによって「汎用的な部品加工を目的とした五軸加工機の制作」が発表された。この研究では五軸加工機のツールは丸ノコに限定されているが、その後も五軸加工機の加工性能を向上させる研究として、中村ら（2018）の「五軸加工機による丸ノコを用いた加工パス導出に関する研究」やツールにドリル機能の追加を施した古庄ら（2019）の「ATC 付き五軸加工機のフォリー制作を通した可用性検討」などがある。

また五軸加工機による新たな構法の可能性を示すものとして、中村ら（2017）の「五軸加工機による新しい校倉構法に関する研究」、中村ら（2018）の「丸ノコツールのパス自動生成システムの開発と留めで構成する木質ドームの制作 多軸加工によるあたらしい建築構法の創出研究　その 2」、RANDI ら（2020）の「五軸加工機による木造レシプロカル・タワーの制作」などがあり、五軸加工機を使用することによって「既存の木組み」をパラメトリックデザインに応用させることができると示された。

さらに中村ら（2018）の「五軸加工機による校倉構法の断面加工に関する研究「−多軸加工によるあたらしい建築構法の創出研究　その 1 −」のなかでは「既存の木組み」について、「建築構法の観点からみれば、部品部材の取合い・納まりが、工具・加工機の特性や制約から決定されているものも多い。逆に言えば、自由度の高い加工機械を前提にすれば、いままで実用化が難しかったディテールの実現もありうる」と述べられており、これら五軸加工機に関する研究によって、「既存の木組み」の枠を広げ「新たな木組み」を生み出す余白が生まれたといえる（図 1-4）。

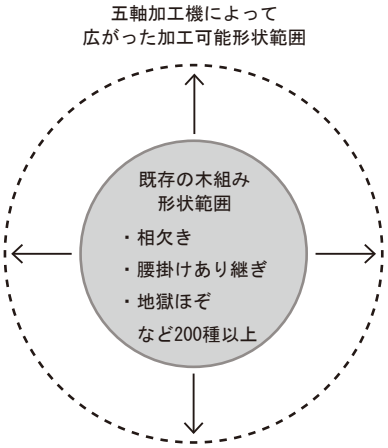


図 1-4. 五軸加工機に関する研究によって広がる木組みの形状の枠組

しかし五軸加工機と木組みに関する研究は、上記のような五軸加工機の機能向上、構法の提案により加工可能形状の範囲を広げることを目的としたものが殆どで、広がった範囲に存在するであろう「新たな木組み」の提案は海外での少数事例のみである。実際の海外の事例としては Mikayla ら（2018）の金輪継ぎや Alfredo ら（2020）でのレシプロカル構造を前提とした「self-aligning joints」の相欠きなどがある。

これら事例は刻みに曲面を使用することによって「既存の木組み」の形状を変化させ、それぞれの「既存の木組み」が持つ煩わしさや欠点を解消している。Mikayla ら（2018）の金輪継ぎ（図 1-5）では、従来のほぞの部分とその周辺を S 字の曲線に改良することによって二つの部材がカチッと嵌るだけで組合わせることができる。曲率による摩擦を制御することにより簡単に外れることを防ぎ、従来の金輪継ぎでの組合わせた際の応力集中を低下させている。

図 1-5. 既存の金輪継ぎ（左）とその改良（右）

出典（左）：金輪継ぎ　アキラ工務店　https://www.kyoto-araki.jp/kyomachiya/kigumi/kanawa.html
（最終閲覧日 2022 年 1 月 11 日）

出典（右）：Diagrams and model of Curved Splice iterations.　Mikayla Heesterman、Kevin Sweet（2018）.
「Robotic Connections: Customisable Joints for Timber Construction」.
『XXII CONGRESSO INTERNACIONAL DA SOCIEDADE IBEROAMERICANA DE GRÁFICA DIGITAL』.　p.6

また Alfredo ら（2020）の「self-aligning joints」の相欠き（図 1-6）では、通常の相欠きで材と材が直交しない場合に不安定になってしまう欠点に対し、刻み部分を曲面で構成することにより直交でなくても成立し、材が自動的に滑らかに滑りながら意図した位置に嵌る。

図 1-6. 既存の相欠き（左）と「self-aligning joints」の相欠き（右）

出典（左）: DIY の幅が広がる『相欠き継ぎ』のやり方　99% DIY 心地よい暮らしを自分でつくる
https://99diy.tokyo/aigakitugi/（最終閲覧日 2022 年 1 月 11 日）

出典（右）: Alfredo Salgado Ferrer、Fabrizio Tozzoli（2020）.「Digital Timber Reciprocity」issuu
https://issuu.com/alfredosalgadoib/docs/portfolio.final　p.10（最終閲覧日 2022 年 1 月 9 日）

これら二つの事例とも材を手動で何度も置きなおし複数回切削を行えば三軸 CNC 切削機で切削できるものの、大量生産や同じ材に複数の刻みが必要になる場合があることから五軸 CNC 切削機を使用しており（図 1-7）、まさに五軸 CNC 切削機によって実用化可能になった「新しい木組み」といえる。

図 1-7. Mikayla ら（2018）による切削（左）と Alfredo ら（2020）による切削（右）

出典（左）: Diagrams and model of Curved Splice iterations.　Mikayla Heesterman、Kevin Sweet（2018）.
「Robotic Connections: Customisable Joints for Timber Construction」.
『XXII CONGRESSO INTERNACIONAL DA SOCIEDADE IBEROAMERICANA DE GRÁFICA DIGITAL』.　p.6

出典（右）: Alfredo Salgado Ferrer、Fabrizio Tozzoli（2020）.「Digital Timber Reciprocity」issuu
https://issuu.com/alfredosalgadoib/docs/portfolio.final　p.72（最終閲覧日 2022 年 1 月 9 日）

一方でこれら二つは共通して「既存の木組み」と同様の直線動作による嵌合形式であり、今後の「新たな木組み」の発展を考慮すると発想の自由度を高めるために回転動作による嵌合形式の可能性も考える必要がある。そこで本研究では五軸 CNC 切削機を用いて、回転操作による嵌合形式の「新たな木組み」を提案し、さらなる「新たな木組み」の発展を加速させることを目的とする（図 1-8）。また今後の展望として、このような新たな木組みの開発が進み現状の加工機によって広がった範囲が埋まるようになれば、人の腕の持つ自由度と同じ七軸、さらにそれを超えるような八軸加工機などの開発に進展していくと考え、結果として木組みの形状のバリエーションが増えることによって、木造建築のデザインの幅も広げることができると考える。

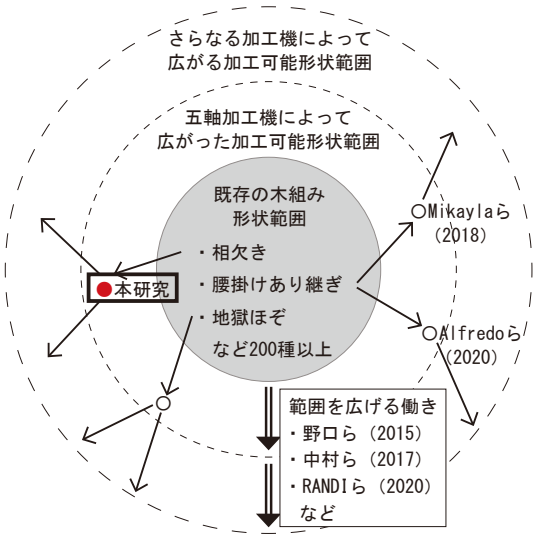


図 1-8. 本研究と既往研究の位置づけ及び今後の展望

1.3　研究目的

本研究では五軸 CNC 切削機を用いて、回転動作による嵌合形式の「新たな木組み」形状を提案することによって、今後のデジタルファブリケーションによる木組み開発を促し、木組み技術全体の発展の一助となることを目的とする。

第二章　研究方法

2.1　使用する機械及びソフトウェア

今回は五軸 CNC 切削機として 5AXISMAKER 社の 5axismaker 5xm600XL（フレームサイズ 600mm × 600mm × 600mm）を使用し（図 2-1）、CNC 機械を動かすためのプログラミング言語である g-code の読み込み及び機械への命令を行うソフトウェアとして Mach3 を使用する（図 2-2）。また 3D モデリングや g-code の生成には 3D モデリングツールの Rhinoceros とそのアプリケーション内で実行されるビジュアルプログラミング言語および環境である Grasshopper を用いる。

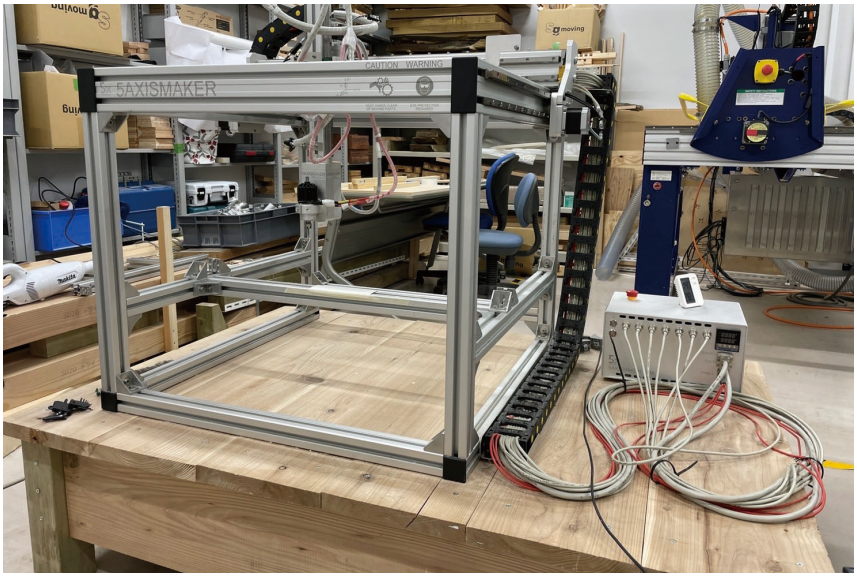


図 2-1. 5axismaker 5xm600XL

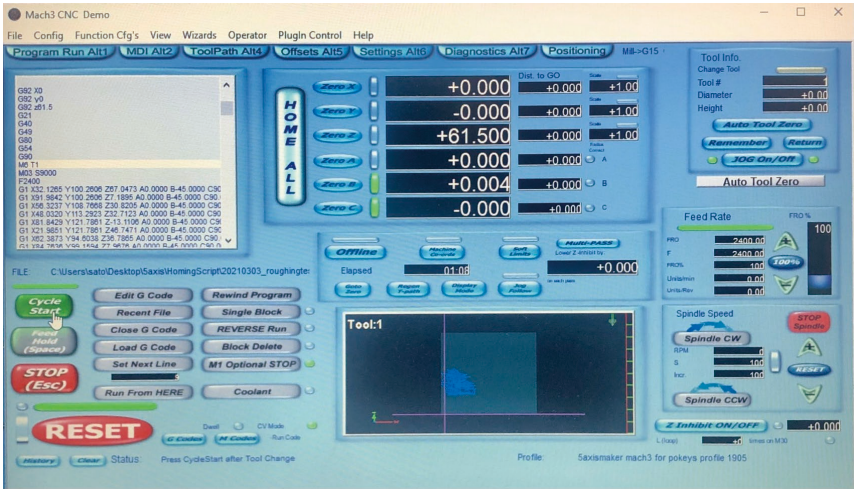


図 2-2. Mach3 の操作画面

また今回使用する五軸 CNC 切削機（5axismaker）について、ドリルの付いたヘッドの直線軸 XYZ 及び二つの回転傾斜軸 BC の方向を以下の図 2-3 に示す。

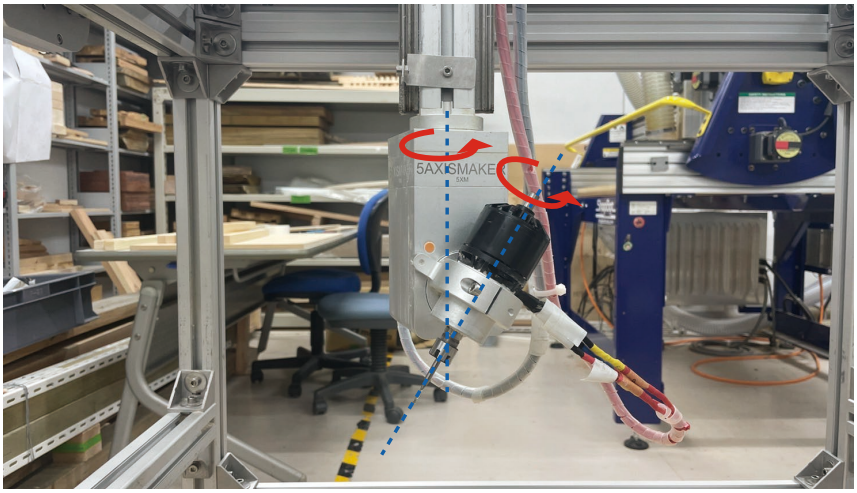


図 2-3. 5axismaker の直線軸 XYZ 及び二つの回転傾斜軸 BC の方向

2.2　木組み開発の流れ

木組み開発の流れ及びそれぞれの段階で使用する機械及びソフトウェアを示したフロー図を以下に示す（図 2-4）。

図 2-4. 木組み開発のフロー図

2.3　木組み 3D モデルの作成

一般的な相欠きの木組みを元に材と材が交差する木組みを設計する。組み合った時の材と材がなす角を交差角、それぞれの材の軸線周りの回転をひねり角と定義し（図 2-5）、本研究ではパラメトリックデザインに対応するため、交差角及び各材のひねり角が自由に設定可能で、かつ回転動作によって嵌合する木組みモデルを提案する。

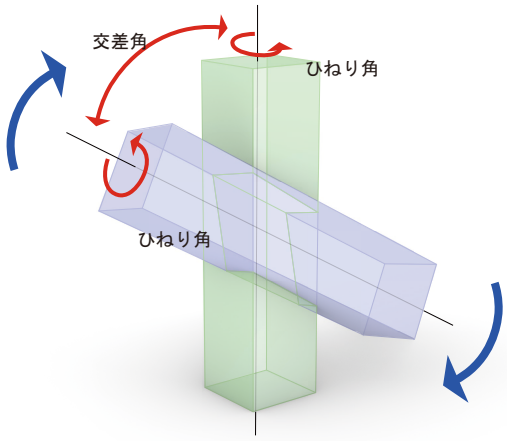


図 2-5. 交差角及びひねり角の定義

2.4　g-code の生成

ドリルの先端が動く軌跡（以下 toolpath とする。）及びそれを実現するヘッドの動作（XYZBC で表現）の g-code 化までを Grasshopper で設計した。g-code は使用するドリルの形状に応じて木組みの「g-code surface」（面の切削）と「g-code edge」（エッジの切削）の二つを生成した。「g-code surface」は「g-code edge」に対し toolpath が非常に長くデータが重くなるため、Grasshopper ファイルを「Roughing」、「Finishing 1」、「Finishing 2」、「g-code generator」の四つに分割した。

g-code 生成にあたり、複数の Grasshopper ファイルそれぞれがデータの受け渡しをしており、その流れを以下の図 2-6 に示す。また各ファイルの詳細を以下に示す。

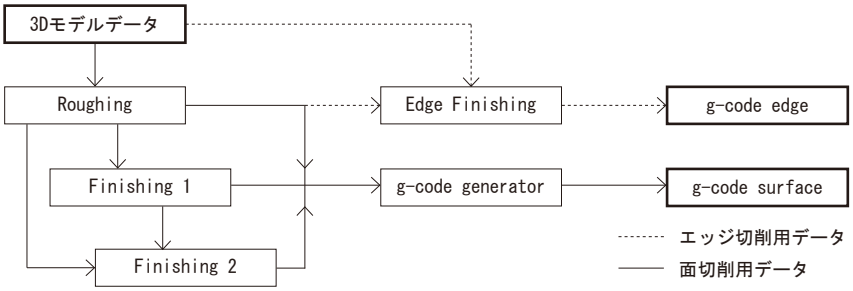


図 2-6. それぞれの g-code 生成までの流れ

2.4.1　3D モデルデータ

作成した木組みの 3D モデルデータとその木組みを実現するために木材から削り出す部分の 3D モデルデータ（図 2-7）を「Edge Finishing」、「Roughing」に受け渡す。また、提案する木組みが上手く嵌るか確認するため 3D プリントするためのデータも作成した。

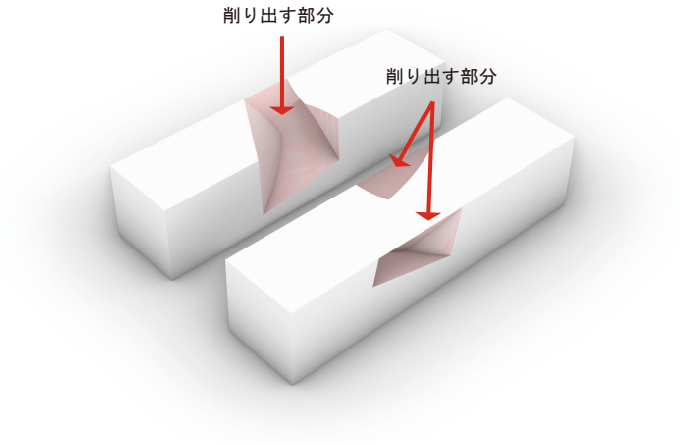


図 2-7. 他の Grasshopper に受け渡す 3D モデルデータ（赤い部分が削りだす部分）

2.4.2　Roughing

一般的に切削を行う際、直接個々の面を削る前に大まかに全体を削る必要がある。Rouughing ではこの大まかに削る部分の toolpath 及び g-codee を生成する（図 2-8）。
生成までの Grasshopper の操作を以下の図 2-9 に示す。詳細のアルゴリズムは付録に示す。

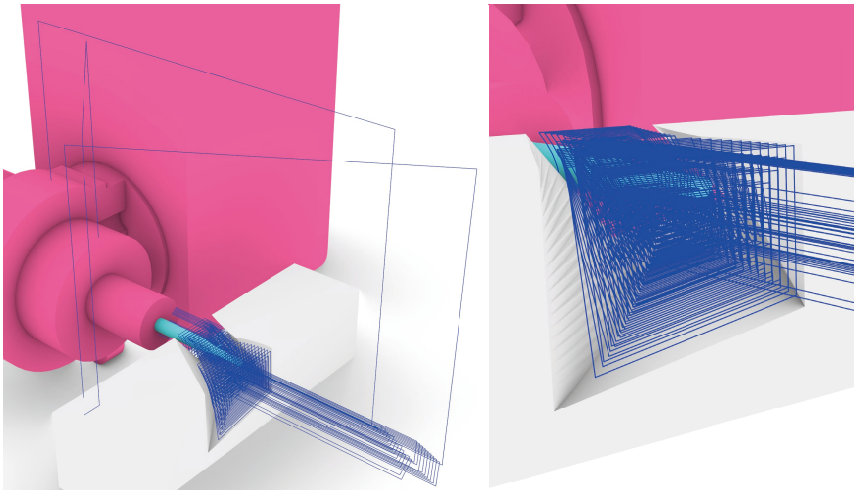
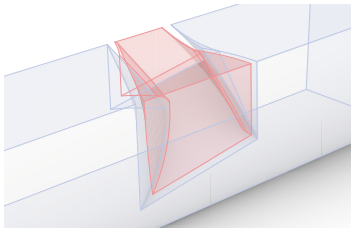


図 2-8. Roughing の toolpath

1. 削る部分のBrepを作成

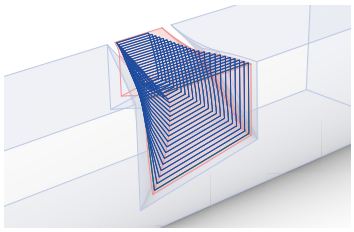
Roughingでは先端が半球状のドリルを使い、その半球の中心を基準に動かしていくので、元の削る部分をその半径分オフセットしたBrepを用意してその範囲を削る。



↓

2. 削る部分をコンターに

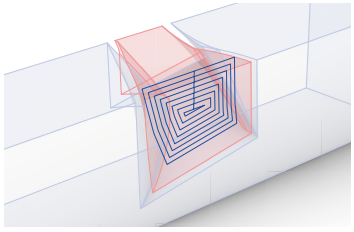
1で作ったBrepをコンター分割する。その分割で得られた平面の外周がToolpathの一部になる。
コンター幅は基本ドリル刃の半径とし、コンターにする方向はドリルの進む方向で基本X軸方向（木材の短短辺方向）とする。また奥の平面が手前の平面をはみ出さないよう注意する。



↓

3. コンターを削るための経路作成

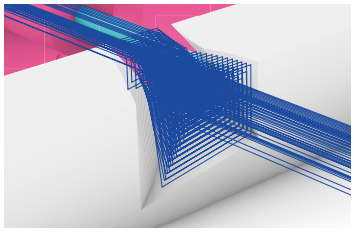
2で作ったコンター平面の外周をドリル刃の半径分ずつオフセットしたポリラインを作成し、それらを同じ平面ごとに一本のポリラインにする。これがRoughing Toolpathの一層分になる。



↓

4. 経路全体を一本のToolpathに

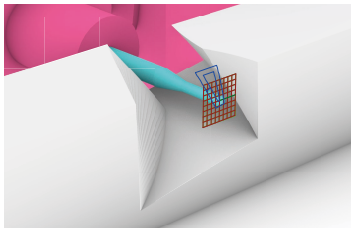
一つの層を削ったら木材を傷つけないように一度ドリルを離し次の層を削りに行く、という動きのToolpathを作成し、それぞれの層のToolpathとつなげて一つのToolpathとする。



↓

5. 各Toolpath pointでのドリルの向きを決める

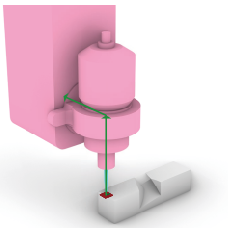
4で作成したToolpathはポリラインであるのでいくつも頂点を持っておりそれをToolpath pointと呼ぶ。この点ごとにドリルの方向を指定するためplaneを設定する。Roughingでは基本的にドリルの向きは2のコンター方向と一致するため調整する必要はない。



↓

6. 各Toolpath planeからheadの位置、傾きを計算

5Axismakerではheadの中心位置と回転軸BCの傾きを設定することドリルの先端位置を決定する。
5で作ったToolpath planeからheadの中心位置と回転軸BCの傾きを逆算しg-code生成に必要なデータを取る。



↓

7. g-codeを生成

得られたheadのXYZ座標とBCの傾きのデータをg-codeに変換する。この時BCの回転が最小になるように角度の値を変換する。

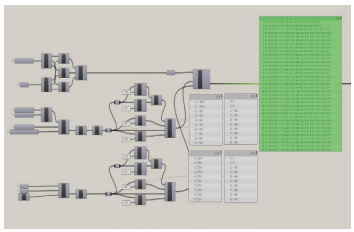


図 2-9. Roughing toolpath/g-code 生成までの Grasshopper 操作

2.4.3 Finihsing 1

ここでは木組みを実現するため、実際の刻みの各面を仕上げとして削る toolpath 及び g-codee を生成する（図 2-10）。また Roughing 同様に生成までの Grasshopper の操作を以下の図 2-11 に示す。

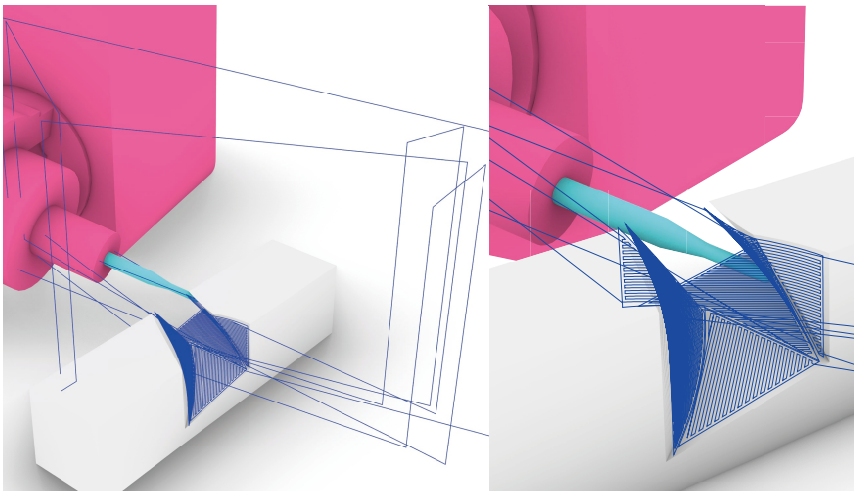
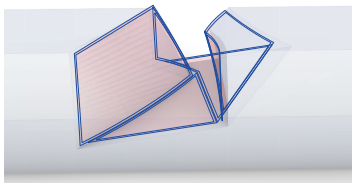


図 2-10. Finishing 1 の toolpath 及び削る部分

①削る面の処理

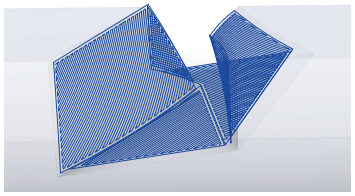
ドリルの先端が半球であることを考慮して実際の面より半球の半径分オフセットした面を作成（半球の中心が面内を移動）。



↓

②面ごとの経路作成

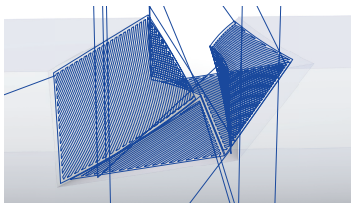
①で作成した面をコンター化し、作られた曲線を面ごとに一本の経路にする。
コンター幅はドリルの直径以下で狭ければ狭いほど表面が滑らかになる。



↓

③経路を一本のToolpathに

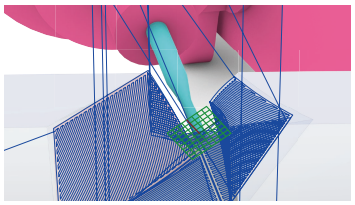
面ごとにドリルが抜ける方向を指定し、面と面を移動する時のToolpathを作る。その後全ての面のToolpathをつなげて一本にする。



↓

④Toolpath planeの作成

Toolpath pointでのplaneを指定してドリルの向きを指定する。



↓

残りはRoughigの⑥⑦と同じ

図 2-11. Finishing 1 toolpath/g-code 生成までの Grasshopper 操作

2.4.4 Finishing 2

ここでは出来上がる木組みの表面をより滑らかにするため Finishing 1 とは異なる角度で面を削っていく（図 2-12）。Grasshopper については Finishing 1 のものと面を削る際の角度設定以外は同じなので省略する。

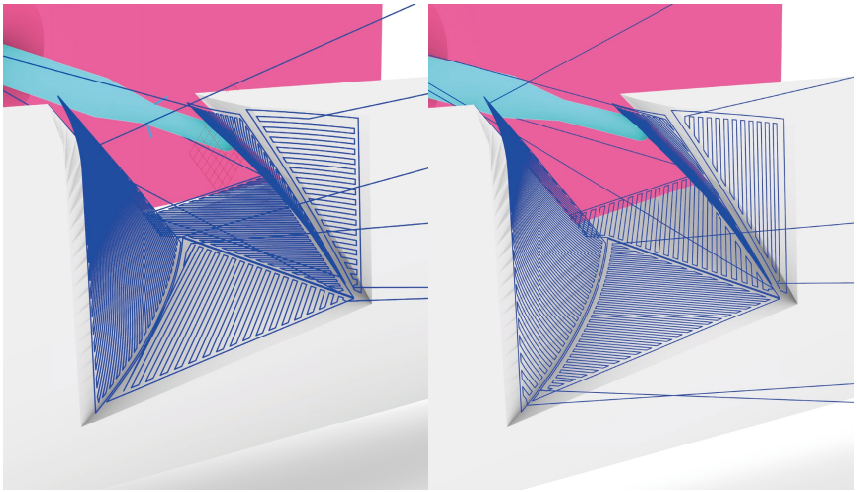


図 2-12. Finishing 1 と 2 の面を削る角度の違い（左が Fin.1、右が Fin.2）

2.4.5 g-code generator

ここでは Roughing、Finishin 1、Finishing 2 で生成した g-code を一つの g-code にまとめる操作をする。またここでドリルの回転速度と移動させる速度を設定する（図 2-13）。

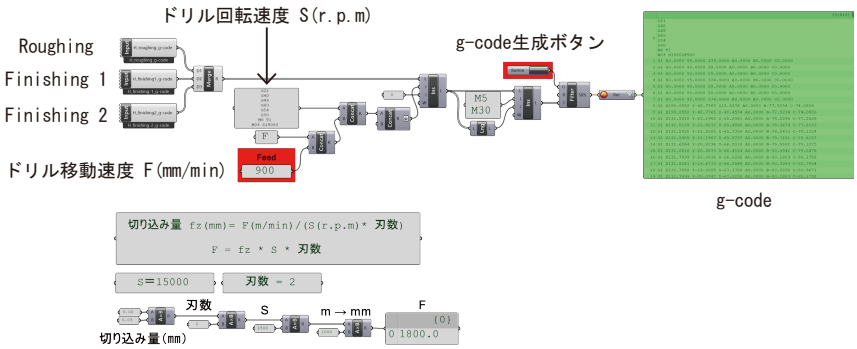


図 2-13. 3 つの g-code をまとめる Grasshopper

2.4.6 Edge finishing

Roughing、Finishin 1、Finishing 2 では面を切削するのに適した先端が丸いドリルを使用するため三つをまとめた g-code を生成した。一方で先端が丸いドリルを使用すると凹なエッジが丸みを帯びてしまうため Edge finishing では先端が尖ったドリルを使用し、凹なエッジを本来の形に切削する。削るエッジを図 2-14 に、この toolpath 及び g-code 生成までの Grasshopper の操作を図 2-15 に示す。

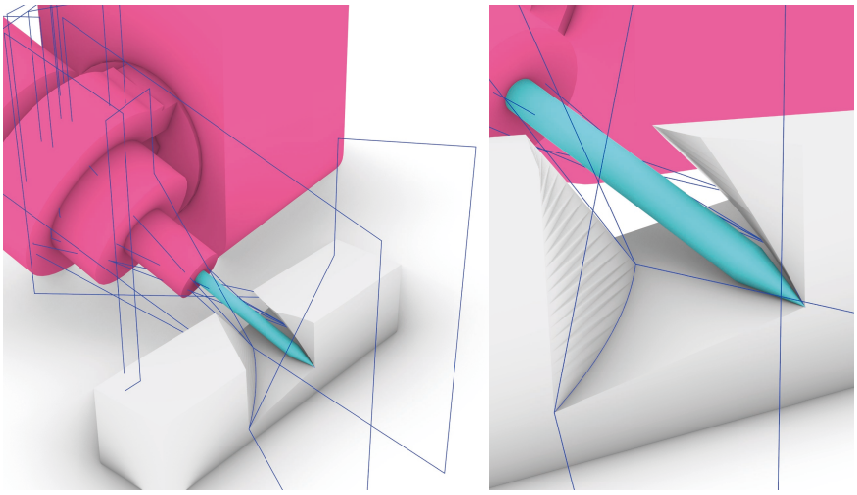


図 2-14. Edge finishing の toolpath 及び削る部分

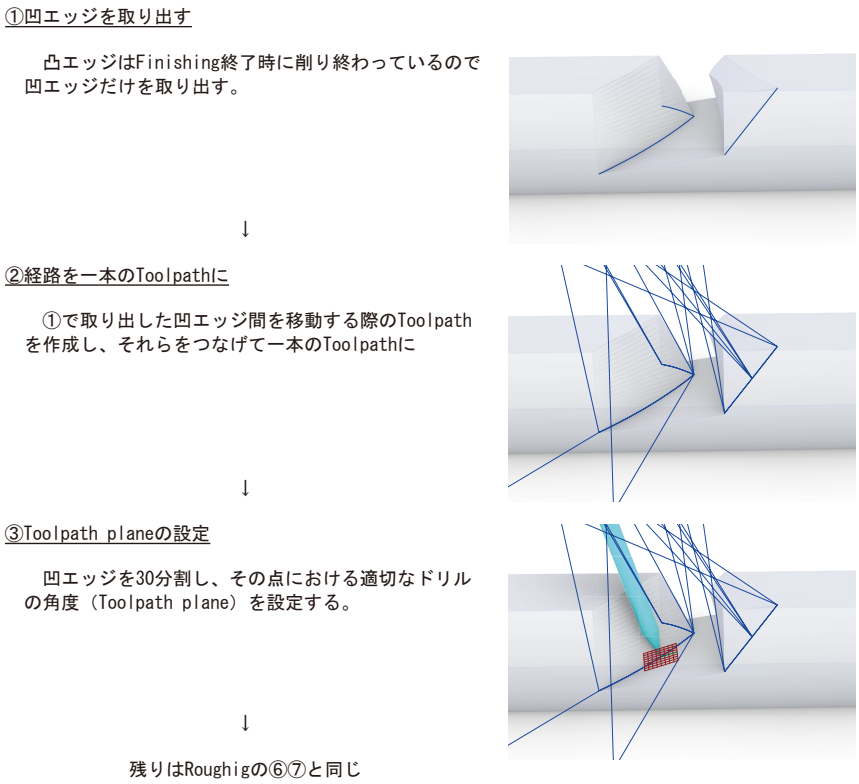


図 2-15.Edge finishing toolpath/g-code 生成までの Grasshopper 操作

2.5　5Axismaker での切削

木材を削るための g-code が生成できた後はドリルの設定および機械のキャリブレーションを行う必要がある。

2.5.1　ドリルの設定

ドリルには先端の形状によって様々な種類があるが、削りたい形状に合わせて適切なドリルを選択する必要がある。今回使用した 2 種類のドリルについて述べる。

一つ目は Roughing、Finishin 1、Finishing 2 で使用する先端の丸い ballnose と呼ばれるドリルで、ドリルの軸と平面の法線ベクトルの方向が一致していなくても使用することができる。また曲面の切削に適している。

二つ目は Edge finishing で使用する先端が尖った V 字溝用ドリルで、名前の通り V 字の溝を削るためのドリルである。V 字の角度はできるだけ鋭いものを選択し、より多くのエッジの切削に対応できるようにした。

どちらのドリルも MISUMI-VONA で注文し、ballnose と V 字溝それぞれの型番は RSB230-R1-10（先端半径 1mm、ドリル長 50 ㎜、刃長 10mm、シャンク径 φ 4）、SVEM4-4-0.05-30-7（先端角度 30°、ドリル長 50mm、刃長 7mm、シャンク径 φ 4）とした。

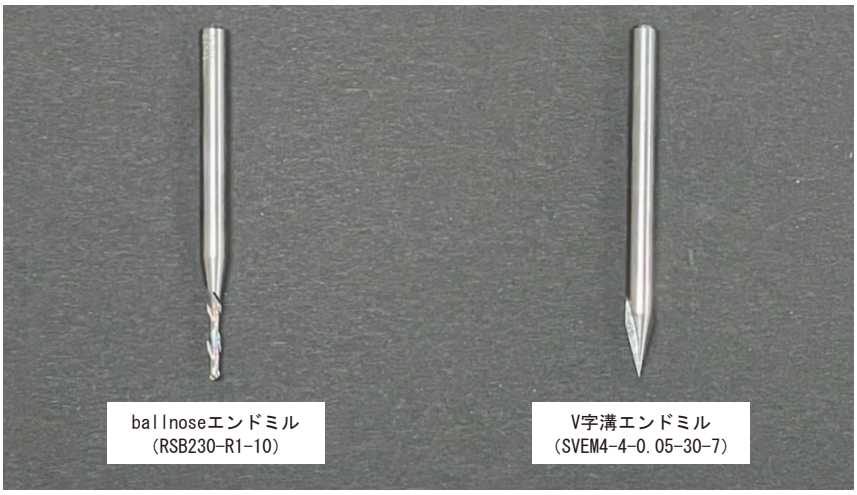


図 2-16. ドリルの種類

2.5.2　マシンのキャリブレーション

生成した g-code を実行するために Grasshopper 上での原点（0,0,0）と機械の原点を一致させる必要がある。Mach 3 を用いて 5axismaker のドリルを操作し、ドリルが下向きの状態でドリルの先端が部材の（0,0,0）と一致するように目視で移動させ、そのドリルの位置を 5axismaker 上の（0,0,0）とする。その時の様子を以下の図 2-17 に示す。

図 2-17. キャリブレーション中の様子（左：右奥が Y 軸正方向、右：奥が X 軸負方向）

第三章　本論

3.1　ひねり嵌合形式の木組み

木組みを開発するにあたって、①材と材の交差角、各材のひねり角がある程度自由に設定できる、②組合わせた際に空洞がなく外から見ても元の木材に欠けている部分がない、という二点を条件とした。

またモデルを作製する際、まず相欠きの特徴を整理しその要素に対してオズボーンのチェックリスト（アイデア抽出の手法:転用、適合、変更、拡大、縮小、代用、置換、逆転、結合の九つの観点で新たなアイデアを考える）を用いて新たに開発する木組みが持ちうる要素を書き出た。その後その中から実際に適用させる要素を選び開発を行った（図 3-1）。

ひねり動作の回転軸の違いから、2 パターンのモデルを開発しそれぞれを 3.1.1、3.1.2 に示す。

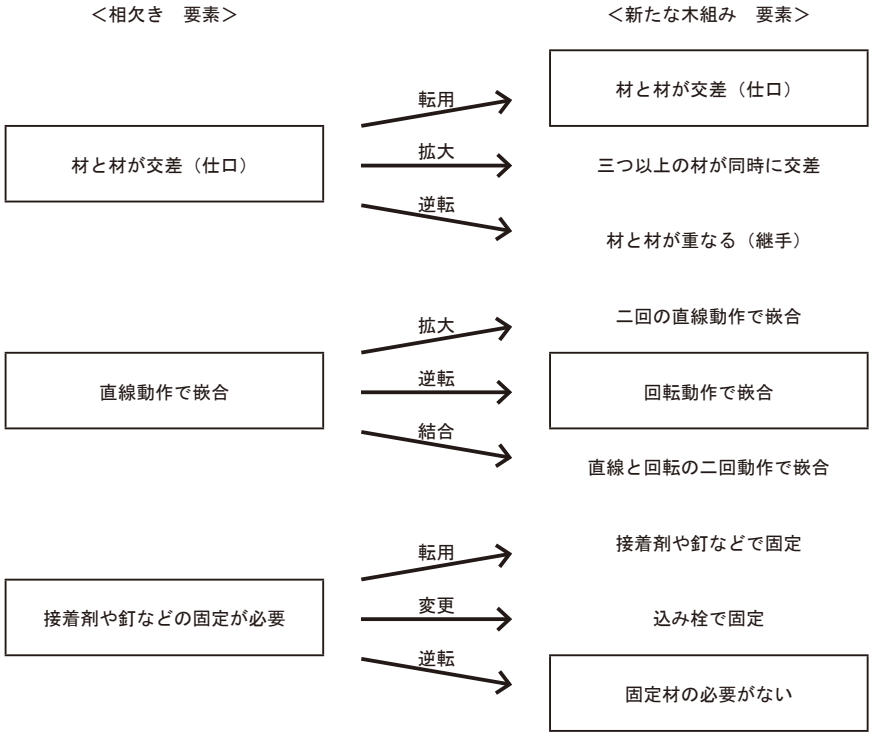


図 3-1. モデル開発の思考法

3.1.1　モデル①

垂直な材をV、もう片方をHとする。この木組みモデルにおけるパラメータはVとHの材軸間の距離（d (mm)）、軸の交差角（x (°)）、VとHそれぞれのひねり角（v (°)、h (°)）の四つで構成される。（図3-2ではd = 10 mm, x = 60°, v = 10°, h = 15°で各部材太さは24 mm × 24 mm）。このモデルではこれら四つのパラメータを決定することによって刻みの形状が自動的に決定し、ひねる際の回転軸（図3-2の赤い矢印）はVとHの交差部（図3-2の赤い六面体）の一辺に決まる（六面体の対角位置の辺でも成立するが、図3-2のひねり角をそれぞれ逆回転させ全体を180°反転させたものと同一なため無視）。また設定したパメータによって交差部が六面体となる場合、全てこの木組みの刻みが成立する。

実際の寸法を図3-3に、刻み方とそれが噛み合っている様子を図3-4に示す。またひねりながら噛み合う様子を図3-5に示す。

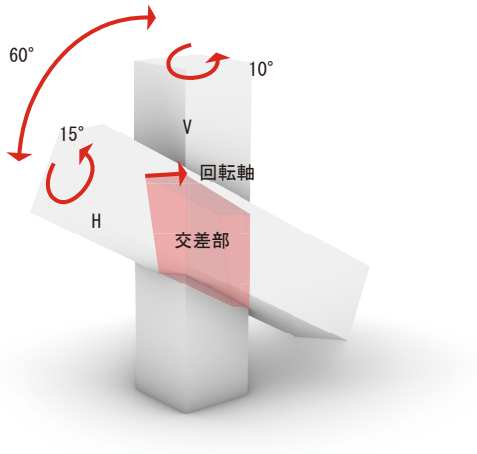


図3-2. モデル①の交差部と回転軸

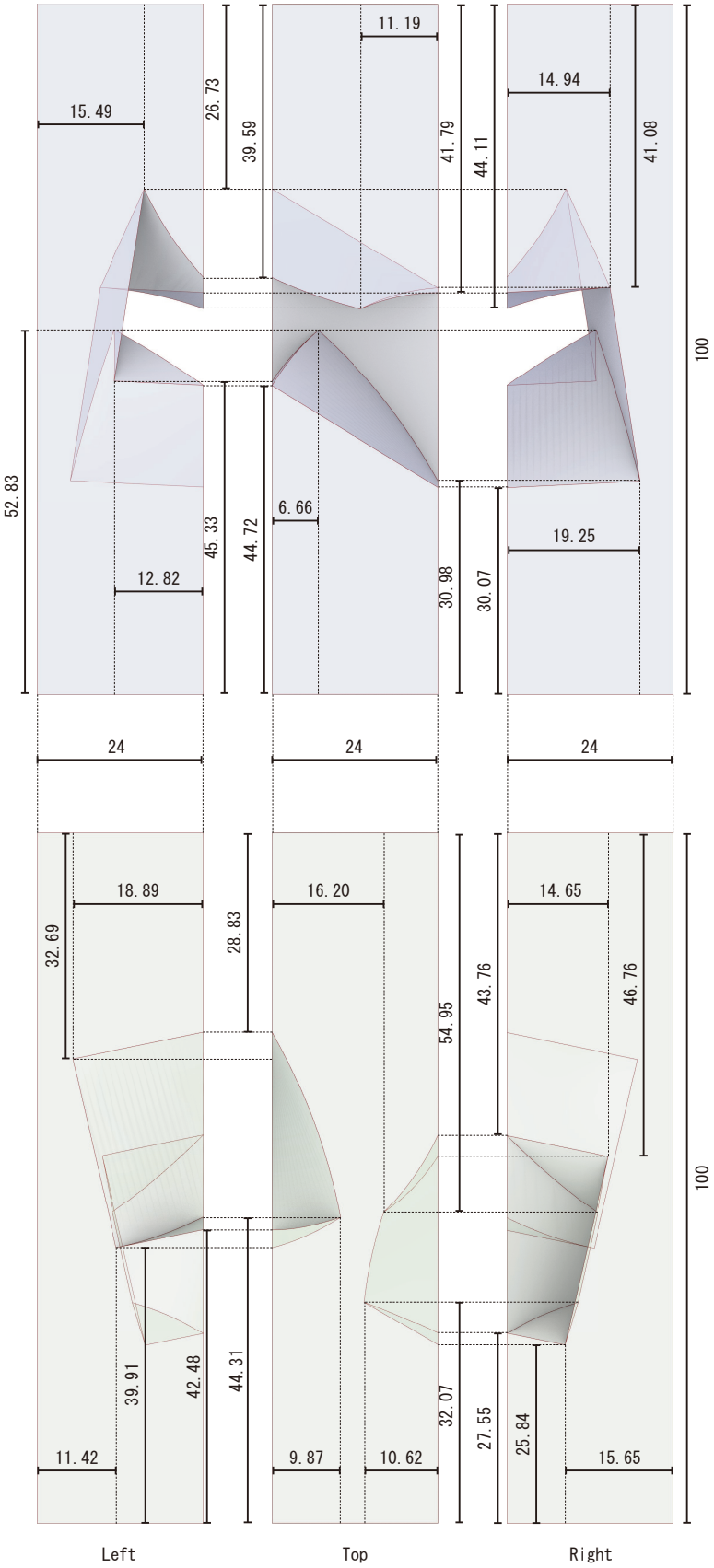


図 3-3. モデル①（図 3-1 とパラメータ同一）の寸法（原寸）

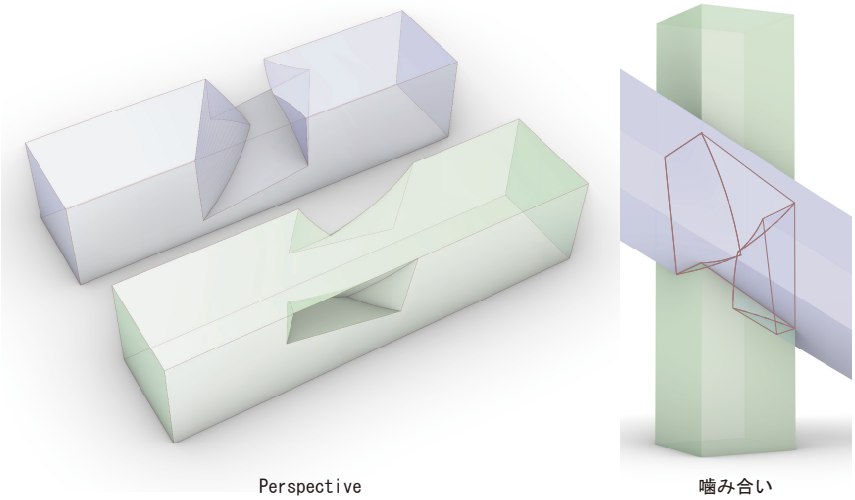


図 3-4. モデル①（図 3-1 とパラメータ同一）の刻みと噛み合い

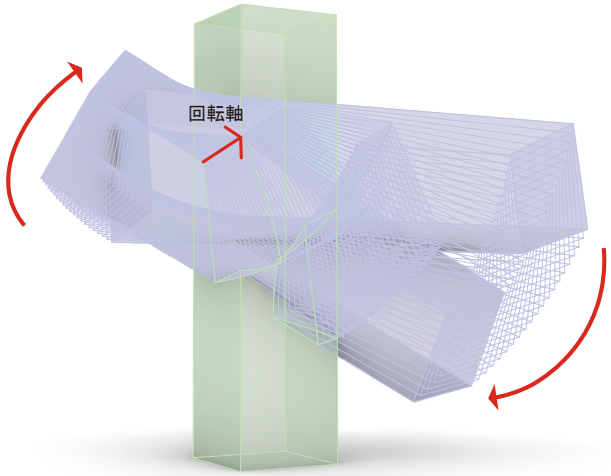


図 3-5. モデル①（図 3-1 とパラメータ同一）のひねりの様子

3.1.2　モデル②

モデル①では V、H の軸間距離、交差角とそれぞれのひねり角を決めることによって回転軸が決まり刻みが決定したが、このモデルではモデル①の四つのパラメータの他に回転軸を交差部の辺と無関係に設定することができる。ただし回転軸の方向次第で、ひねり動作の際に V と H が緩衝することがあり、「組合わせた際に空洞がなく外から見ても元の木材に欠けている部分がない」という条件を満たさない場合があり自由度はモデル①と比較すると高くない。しかし、施工時に複数部材を同時に接続しなければならない場面では複数交差部の回転軸を揃える必要があるため、回転軸を故意に設定することができるという点は非常に効果的に働く。

モデル①との違いをわかりやすく示すため回転軸以外のパラメータは同様にし、回転軸の比較の図を図 3-6、モデル②における実際の寸法を図 3-7、木組みの刻みと噛み合いの図を図 3-8、ひねりの様子を図 3-9 に示す。

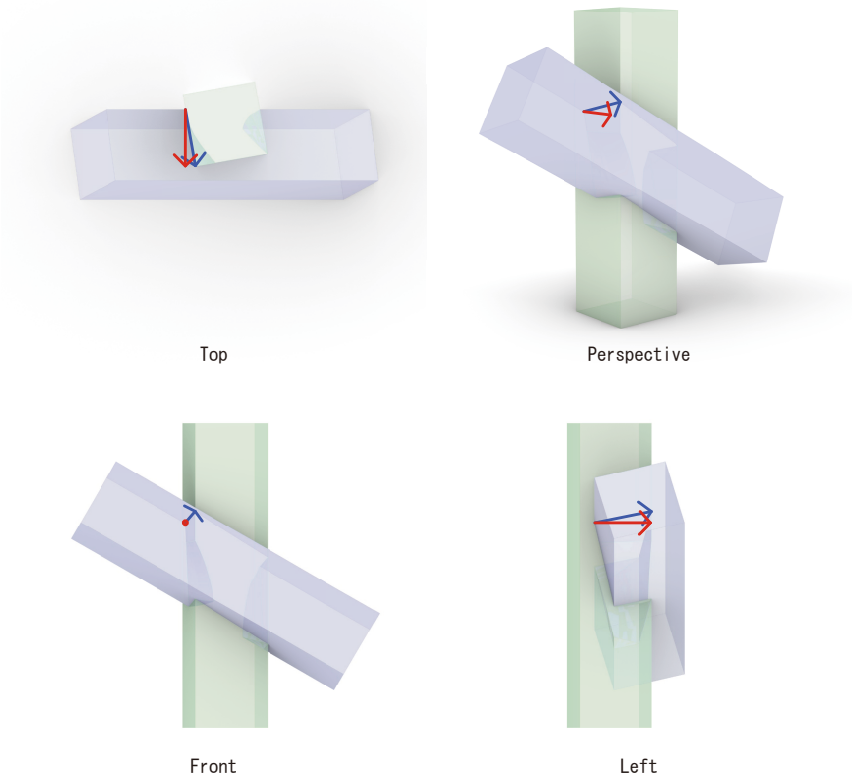


図 3-6. モデル①と②の回転軸比較（青がモデル①、赤がモデル②）

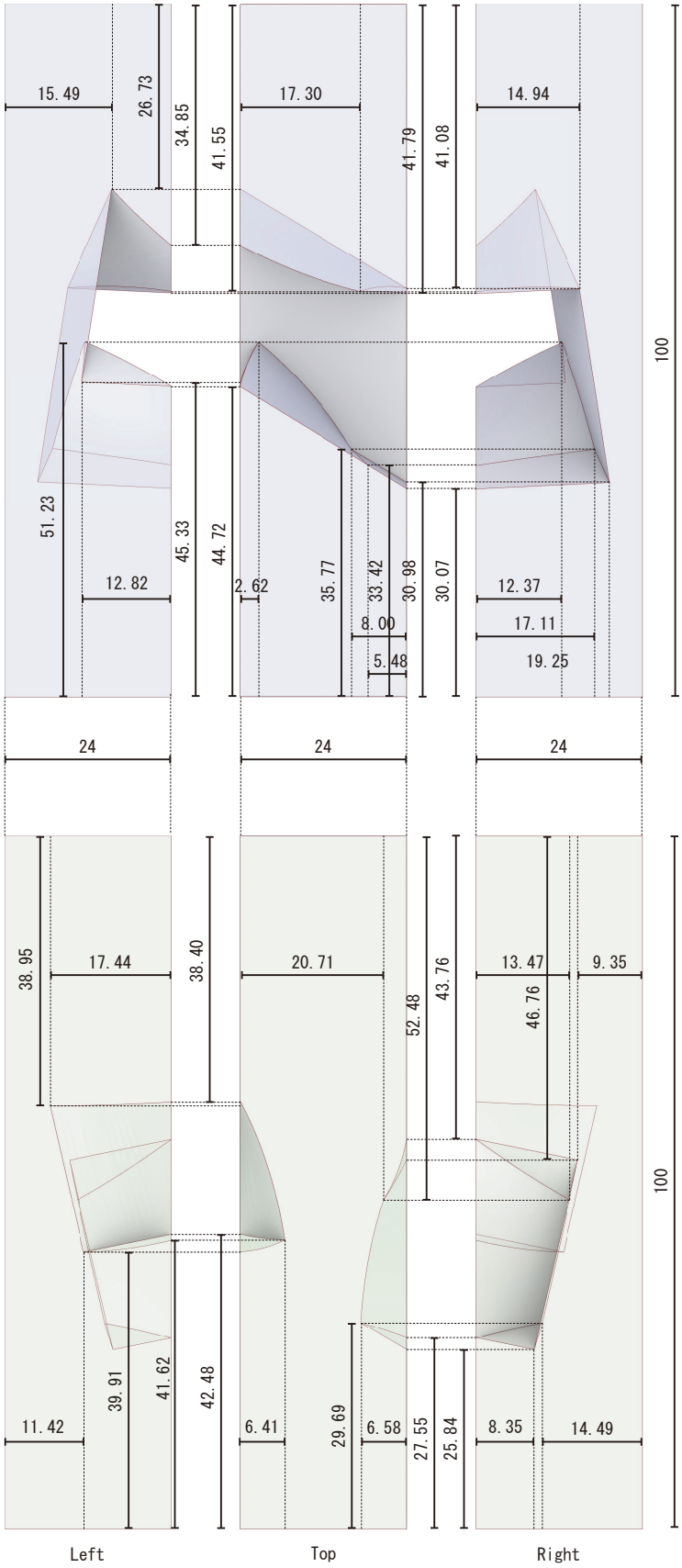


図 3-7. モデル②の寸法（原寸）

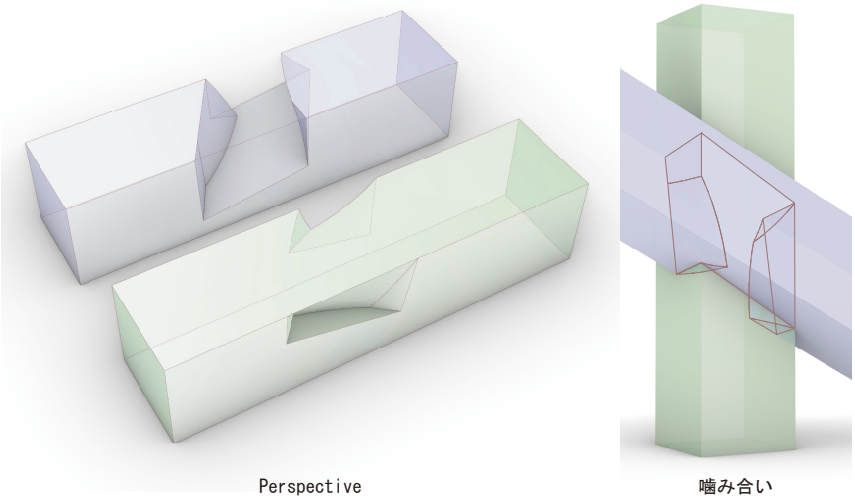


図 3-8. モデル②の刻みと噛み合い

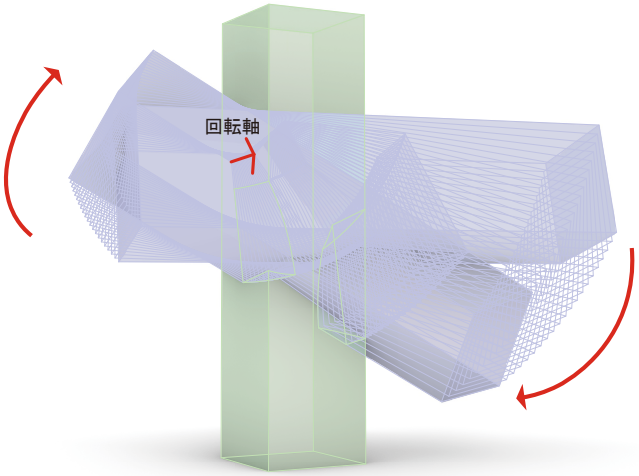


図 3-9. モデル②のひねりの様子

3.2　木組みモデルの考察

開発したモデルを 3D プリントし、その挙動について分析を行った。また 3D プリントしたものは材と材の隙間が殆どないものであった。

まずねじり回転動作について述べる。動作中は木組みの刻み面の殆どが曲面であるため、材と材の位置関係が一意に決まり、嵌め始めてから最後締まるまで設定した軌跡から外れることはなかった（図 3-10）。また､締まり切った場面では図 3-11 の赤丸の部分（以降「爪」とする）が引っかかることと刻み面の摩擦により、回転軸方向の力を加えても簡単に外れることはなかった。特に材が回転しないようにした状態で力を加えるとかなりの抵抗力を感じることができた（図 3-12）。一方で回転軸周りの木組みがゆるむ方向モーメントに対しては比較的簡単に外れた（図 3-13）。以上の事より、本モデルはひねり回転運動の一自由度ではめ込むことができると言える。

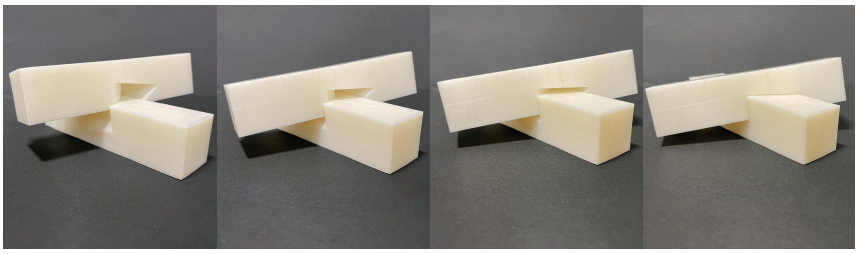


図 3-10. ひねりながら嵌め込む様子

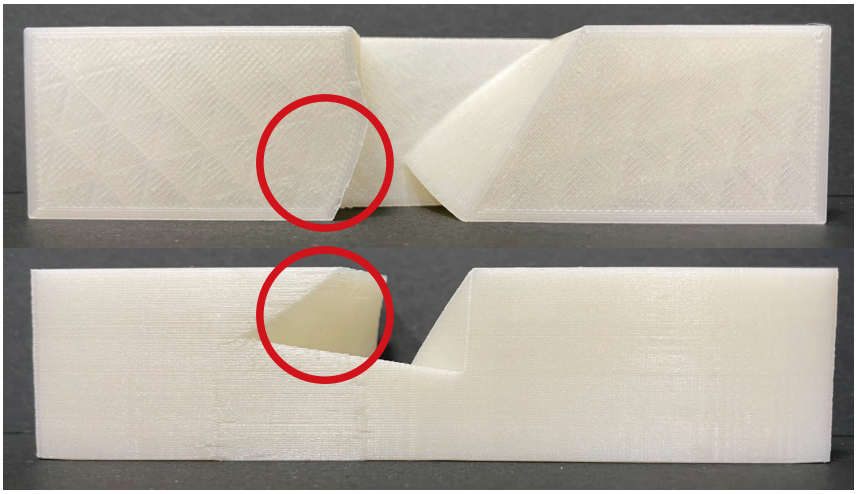


図 3-11. 回転軸方向の力に抵抗する「爪」の部分

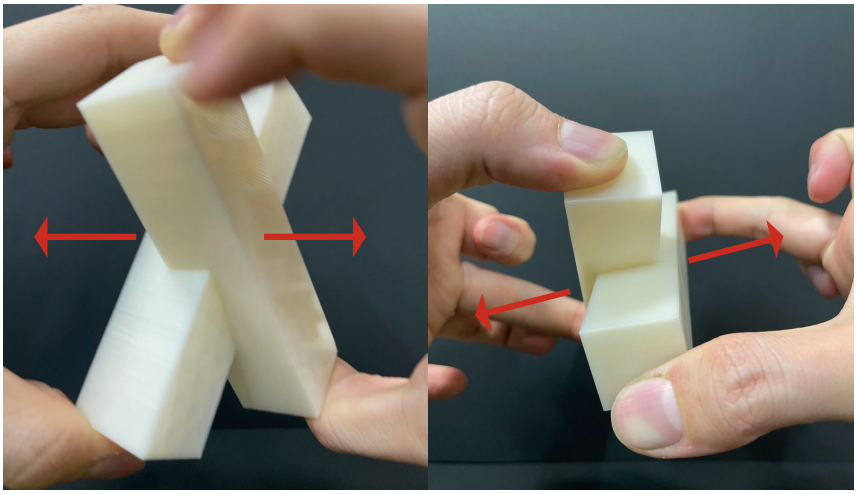


図 3-12. 回転軸方向に引っ張る

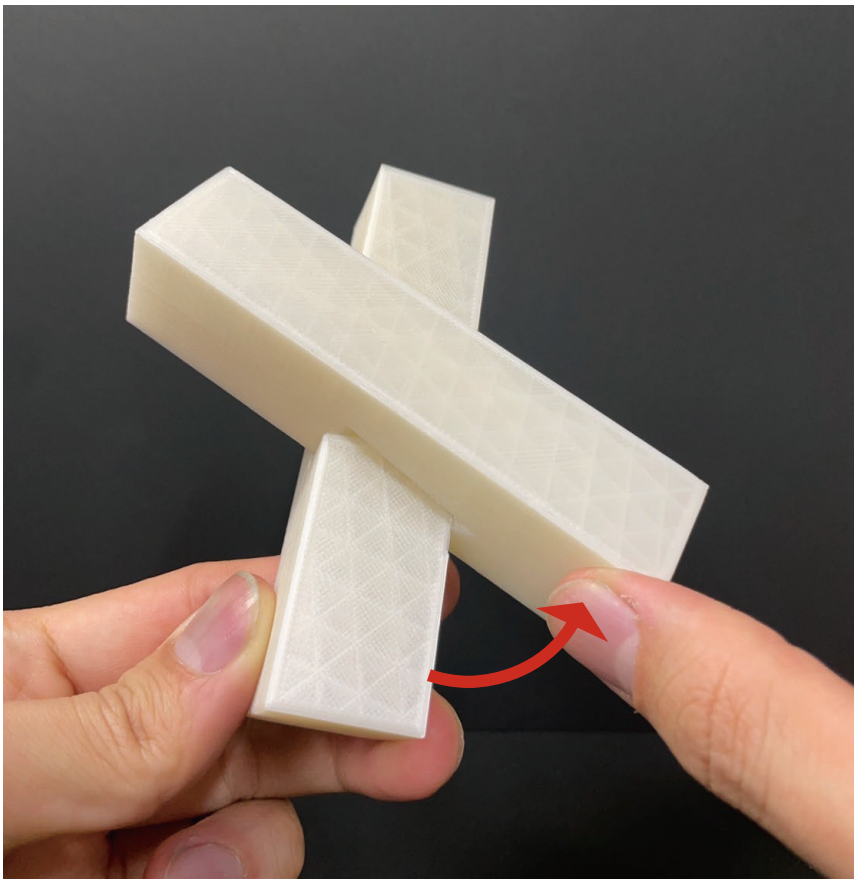


図 3-13. 回転軸回り、ゆるむ方に力をかける

次に相欠きとの比較を行う。相欠きは接着剤や釘などによる固定が必要だが、本研究の木組みは木組みが締まる方向に力が加わる状態ではそれらが必要でない。また交差角が小さいほど爪の部分が大きくなり、刻み面の総面積が増え摩擦力も大きくなるため締まった時により強固に噛み合うが、一方で交差角が 90°に近づくと爪が小さくなり、刻み面の総面積が減少し摩擦力が低下、形状も相欠きに近づいていき、交差角が 90°の時には回転運動ができずこの木組みが成立しない（図 3-14）。以上の事を考慮すると交差角が小さいときは本研究の木組み、90°に近いときは相欠き、と建築デザインの場面に応じて使い分けが可能だと考える。

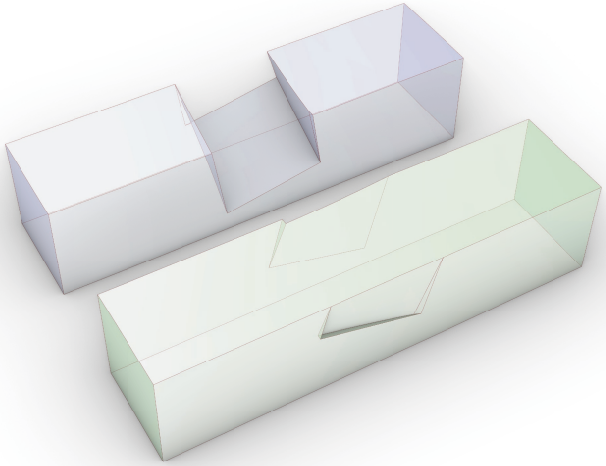


図 3-14. 交差角が 85°の本研究の木組み

最後に本研究の木組みについて今後について述べる。この木組みの実用化には力学的特性、パラメータの適用可能範囲、施工性を明らかにする必要がある。

力学的特性に関して、木組みが締まった時に主に力学的に効いているのは爪の部分であり、その部分を明らかにする必要がある。具体的には木組みが締まる方向にモーメント力が加わったときに材のめり込みの影響や、爪の根本はエッジが鋭角であることによるその部分が割けていく可能性などである。

パラメータの適用範囲について、モデル①に関して材と材の交差部分が六面体となる範囲であれば成立することがわかっているものの、交差部が六面体となる各材のねじり角と交差角の関係は明らかにされていない。またモデル②についてはモデル①のパラメータに回転軸の方向が加えられることにより、適用範囲を明らかにすることがさらに複雑になる。しかしパラメータの適用範囲が明らかになることはこの木組みが実用化されるためには非常に重要である。

施工性に関して、回転での接続は非常に正確さが求められるが、鉄骨などと比較して木材には反りなどがあるため、構築物全体としてそのズレを吸収する仕組みが必要となる。

3.3　切削結果

今回使用した木材は 24 ㎜× 24 ㎜のパイン材である。

今回の木組みの形状について、材面法線一方向からだけでは削ることができない形状であり、三軸 CNC 切削機では何度か人力で材を回転させなければならず、また場所によっては材をひねった姿勢で固定する治具が必要となるため五軸 CNC 切削機を使用するのに適しているといえる。切削した形状はモデル①でパラメータは 3.1.1 で例として示したものと同じものである。

切削結果について初めに Roughing の結果について述べる。前述のとおり Roughing は木組み形状を大まかに削るものであり、面の表面がきれいになっているかよりも目標の形状に近いかを評価基準とする。実際の切削の様子を図 3-15 に示す。また Roughing の実行結果を図 3-14 に示す。

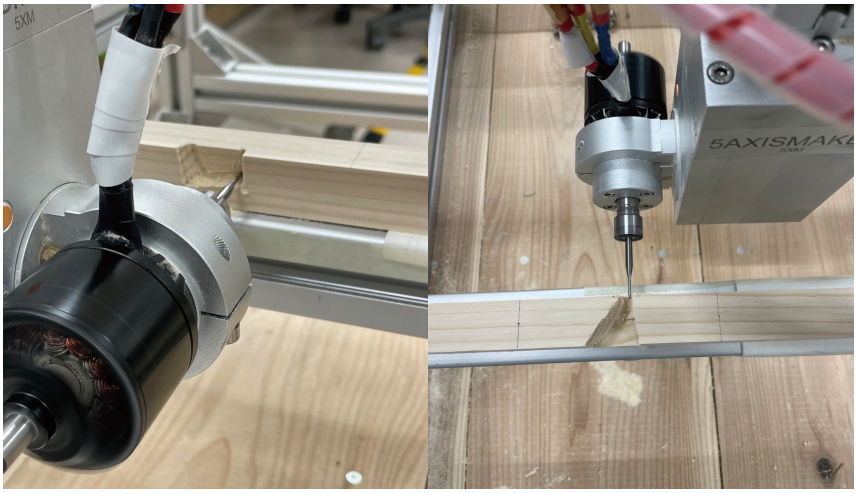


図 3-15. Roughing 実行の様子

Roughing 実行結果について、大まかな形状の実現は達成しているが、図 3-16 の V の中央の写真を確認すると、右側の部分が余計に削られていることがわかる。この原因は後に述べる五軸 CNC 切削機自体のズレ、ドリルの長さの測定誤差とドリルの回転によって余計な部分が巻き込まれて削れてしまったことによると考えられる。

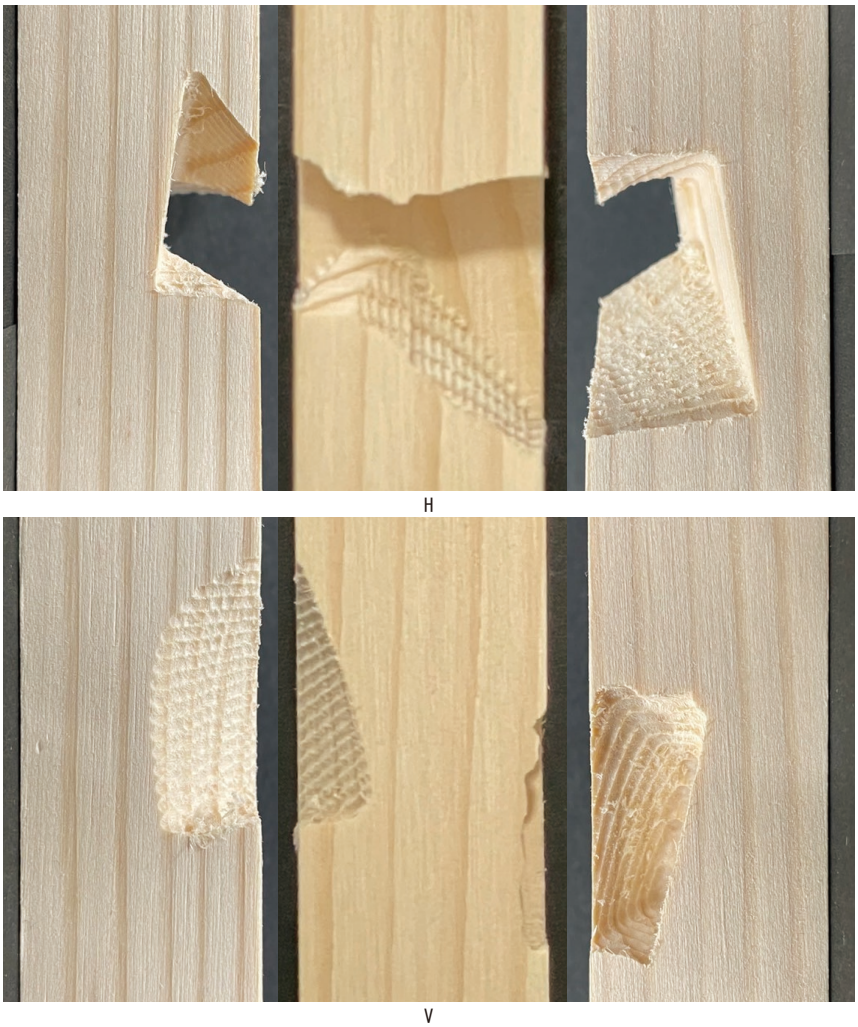


図 3-16. Roughing 実行結果

次に Finishing について述べる。Finishing は Finishing 1 と Finishing 2 によって構成され面の切削を二方向から行うことによって面をより滑らかに加工する。実際の切削の様子を図 3-17 に示す。Finishing 1 後の面の様子と Finishing 2 まで行った時の様子を比較すると（図 3-18）、明らかに Finishing 2 まで行ったもののほうが滑らかになっているのが確認できる。このことから Finishing 2 までやることの有効性を示すことができ、さらに滑らかな仕上げを目指す場合、1、2 とは異なる方向で削ることで実現することができる。

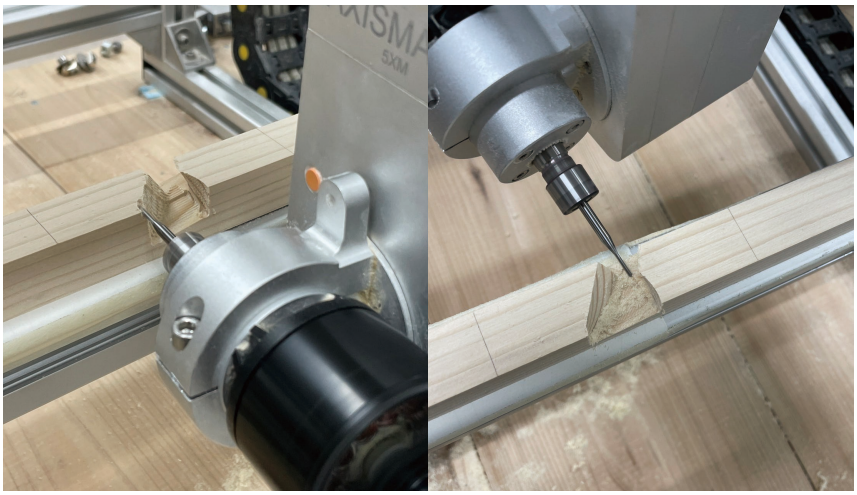


図 3-17. Finishing の様子

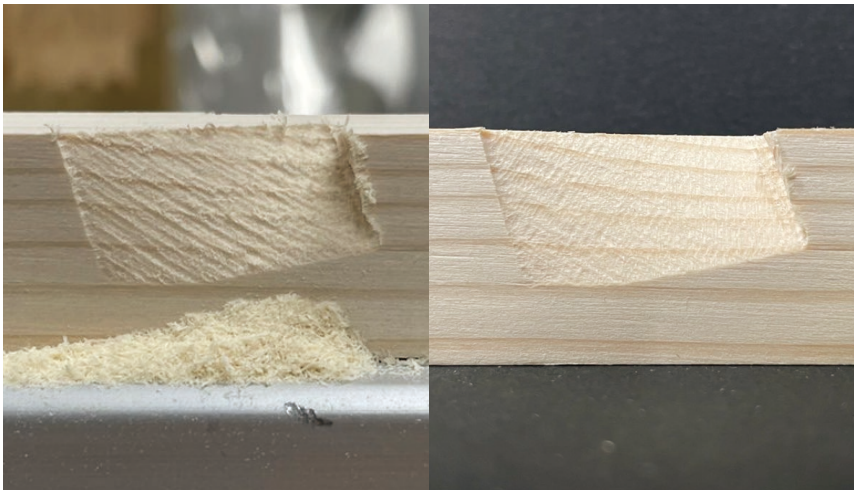


図 3-18. Finishing 1 後（左）と Finishing 2 後（右）

また Finishing の実行結果を図 3-19 に示す。ドリルの動きを制御して全体としての形状を再現できたことは評価できるものの、H 中央写真の右下部分は実際に想定していた形状ではない。H も V も全体的に中央によってしまったことが原因でできた形状だと考えられる。Roughing 同様に機械のズレ、ドリルの長さの測定誤差、ドリルの巻き込みに問題があると考えられる。

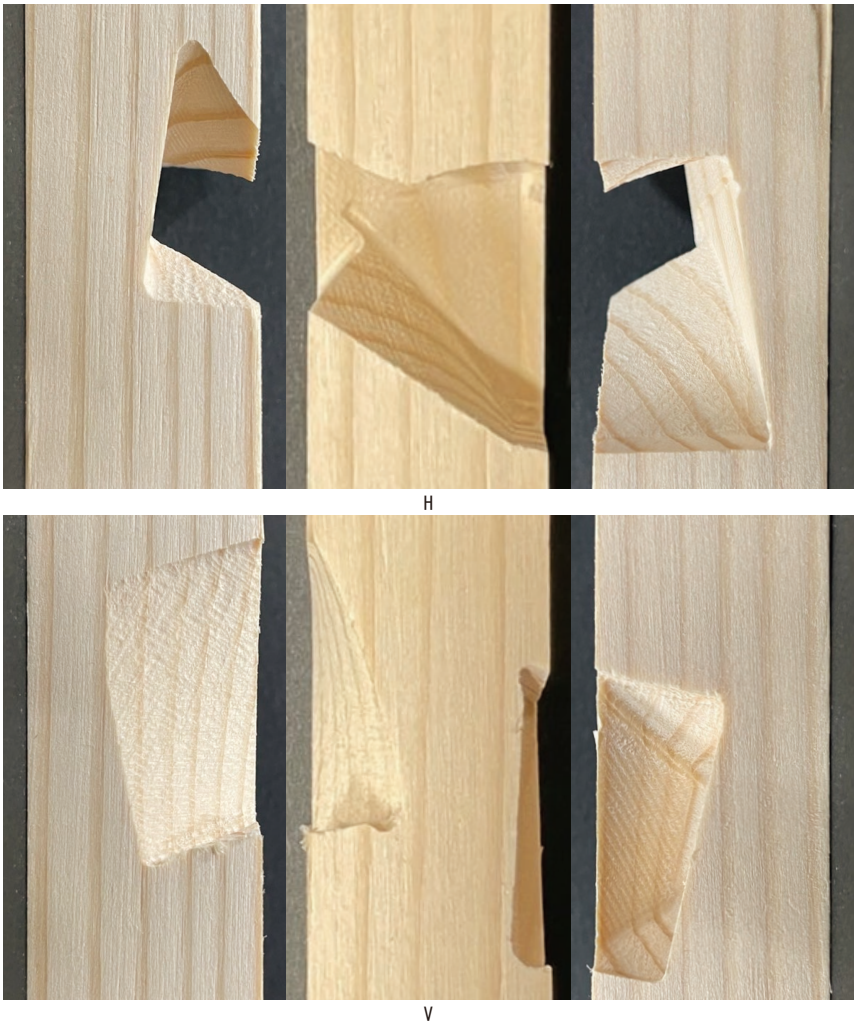


図 3-19. Finishing 実行結果

Edge finishing について、名前の通り木組み形状のエッジの部分を整えるものである。実際の Edge finishing の様子を図 3-20 に示し、その結果を図 3-21 に示す。

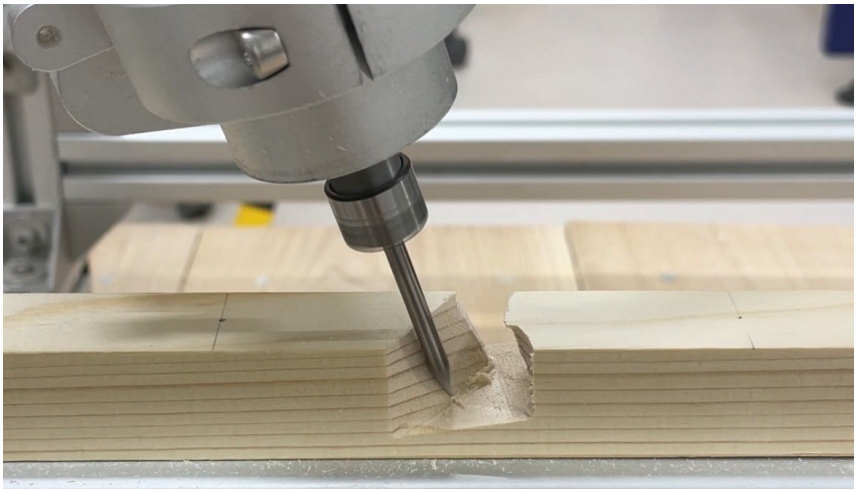


図 3-20. Edge finishing 実行の様子

Edge finishing の実行結果について、Grasshopper で設計したプログラム通りドリルが動いたためエッジの部分の見た目は非常にシャープになった。しかし前述の Roughing、Finishing と同じ問題で削りすぎてしまう部分も生じた。

この Edge finishing について、見た目に関しては見栄えの良いものになるが、「爪」の根本など鋭角の部分は木材の裂けを助長してしまうため重要度はあまり高くないと考える。Roughing、Finishing とはドリルの形状が異なるため途中でドリルを付け替える必要があり、その点でも本研究の木組みに関しては重要度の低いものであった。

一方で鋭利なエッジを削れたという事実は今後新たな木組みを開発していく際の一つの考える材料になりえると言える。

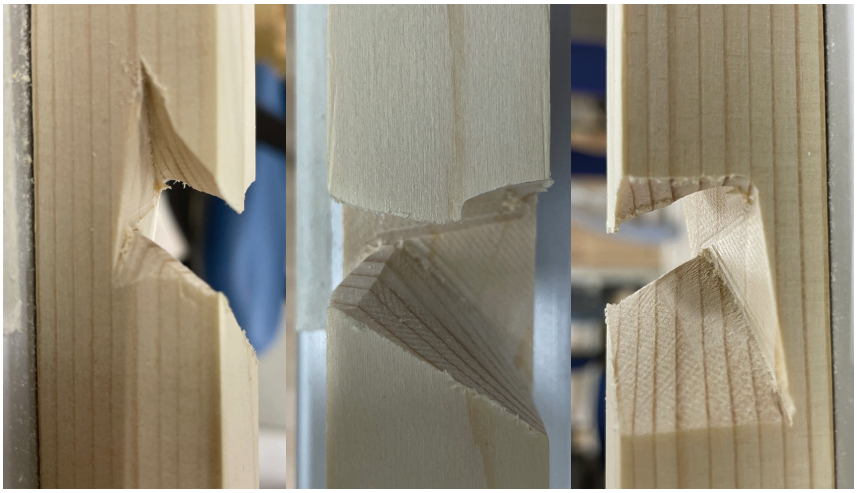


図 3-21. Edge finishing 実行結果

切削時間に関して、ドリルの形状が同じである Roughing、Finishing は連続して行い、例として示したモデル①の部材 H で 32 分、V で 26 分かかり、Edge finishing では同じ部材 H では約 5 分、V で 7 分であった。ドリルの付け替えやキャリブレーションの時間も考慮すると一部材当たり一時間弱で大量生産するにあたっては少々時間がかかりすぎているため、Toolpath の効率化などが今後求められると考える。

切削結果について、全体的にズレが出てしまうという結果になってしまった。この原因は切削を行うためのプログラム上のパラメータ設定の問題も考えられるが、五軸加工機の弱点である機械の繊細さの問題も考えられる。五軸加工機では軸を動かすモータが分かれて設置されているわけではなく、5 つの軸が集合しているため三軸加工機と比較して剛性が高くない。機械自体が力を受けた時に軸自体がその力をを吸収してしまうことで切削を何度も繰り返す内に軸が傾いていくことがあり、特に回転軸はその影響を受けやすい。軸のズレが数百分の一㎜であってもドリルの先端位置はドリルの長さが 100 ㎜であれば数ミリのズレになってしまう。そのため五軸加工機を利用する際は軸がズレてしまっていないか確認する必要がある。

他のズレの原因として、原点のキャリブレーションやプログラムを動かす上で必要なドリルの長さの測定など手動又は目視で決める部分が多数あることがあげられる。センシング技術などを用いて機械化できる部分はできる限り進めて人の手で決める部分を減らしていくことが加工精度を上げるにあたって重要である。

一方で今回の木組みに関して、多少ズレが生じても締まる方向に力が架かる場面で使用すれば材と材同士が離れてしまうことはなく問題はない。施工時のことを考えると、木材は反りなどの影響で施工位置がズレてしまうことも考えられるので、少し余裕を持たせた方が使用しやすいとも考えられる。

また同じ木材を使用しても、木目の方向によって切削面の滑らかさに大きな差があり、材料の繊維方向の違いによるドリルの回転速度や移動速度を調整する必要がある。よってこの木組み実用化にあたっては材料の選定や木材の種類による加工のしやすさや構造的強さを把握する必要があり、目的に応じてどの種類を使用するべきか、木のどの部位を選ぶべきかを考えていく必要がある。

3.4　全体形の提案

ここでは本研究の木組みの強みを活かした全体形の提案を行う。

この木組みの強みとして回転動作で嵌め込んだ後、締める方向に力が加わっていれば材と材が固く連結されることである。このことを活かす形状としてはアーチ形状があげられる。図 3-22 の基本形を元に考えていくと図のアーチ①や②などの形状が考えられ、3.2 で述べたように材と材の交差角が小さいほど強固に組み合わさるためアーチ①よりも②の方が構造的に強くなり、本数が増えれば増えるほど交差角が小さくなり強力になると考えられる。

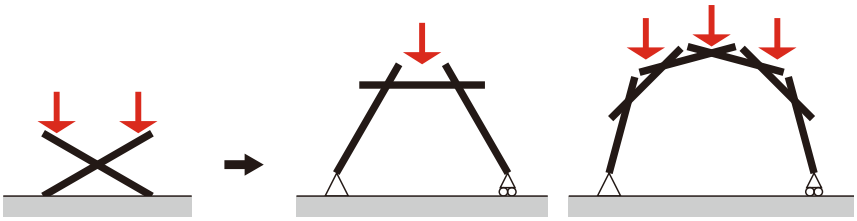


図 3-22. 本研究の木組みの特性を活かす全体形状

上記のアーチ形状を基本とし、それを活かした全体形を図 3-23 に示す。

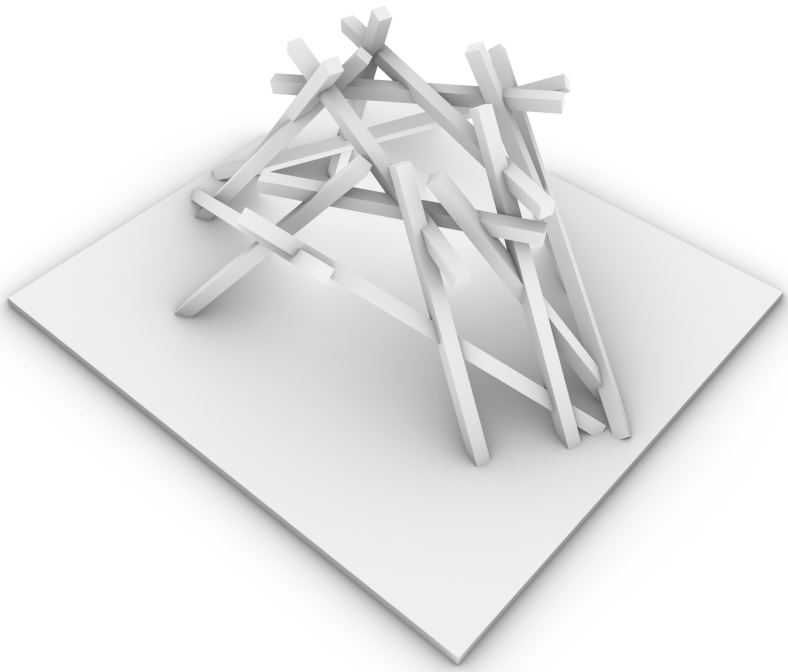


図 3-23. 提案する全体形状

この形状は三本の材からなるアーチを編み込むような形式で構成されている（図 3-24）。図の物は四つのアーチユニットを互い違いに合わせることで成立している。この形式であればアーチユニットの数を増やしていってヴォールト形状にしていくことも可能である。またこの全体形の足はピン支持を想定しており、アーチ方向（図 3-24 上からの図での上下方向）への広がりは本研究の木組みで抑えられているが、それと直行する方向（図 3-24 上からの図での左右方向）への広がりを対策するためである。実際に 3D プリンターを用いて模型を作製し（図 3-25）、土台に乗せた場合（ピン支持）とそうでない場合の広がりに対する抵抗を比較したところ、土台に乗せたほうが抵抗を強く感じることができた。

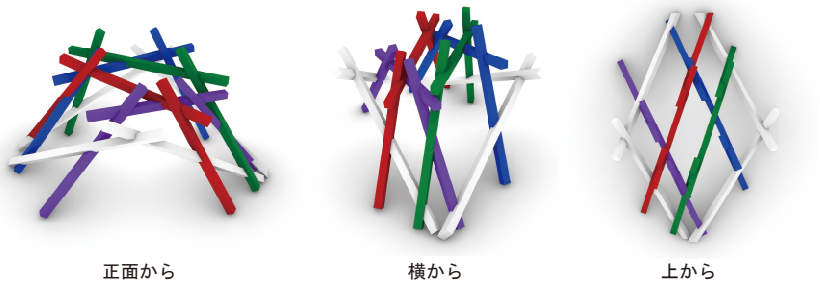


図 3-24. 全体形状の構成

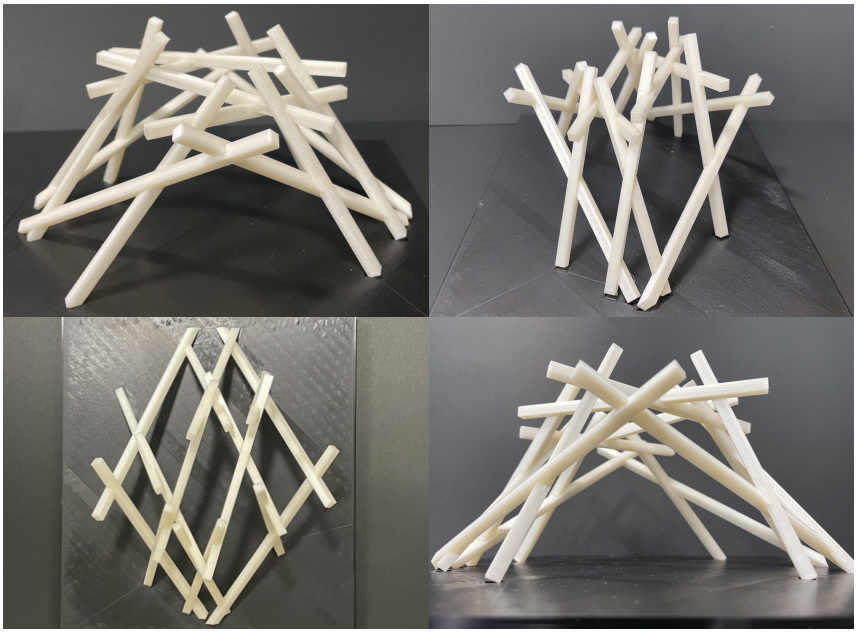


図 3-25. 3D プリントの模型

またこの全体形の施工法について、アーチユニットごとの接合部は本研究のモデル①を使用し、アーチユニット自体の接合部はモデル②を使用することを想定している。実際に施工をする際にはアーチユニットごとの接合ではなく図 3-26 のような材と材の✖（薄い赤の部分）を一つのユニットとして、ユニットごとに接合していく。この際、✖の端部の接合部（ユニット同士の接合部）では回転軸を一致させる必要があるためモデル②を使用している（図 3-27）。ただし図 3-26 の赤い丸の部分は施工全体の最後に接合する部分で、材と材を回転させることができないため相欠きを使用しなければならない。

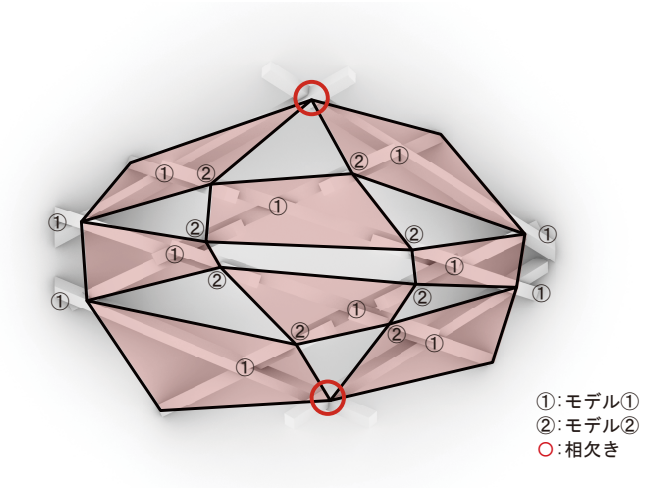


図 3-26. 構成ユニット

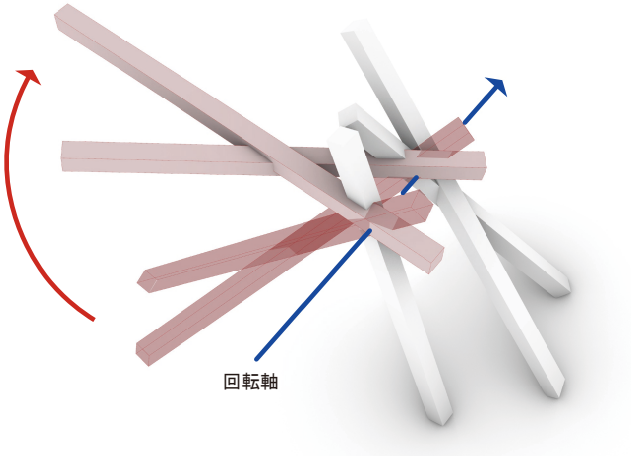


図 3-27. ユニット同士の接続

今回は実際に施工を行っていないが、各接合部の木組み形状から全体形の施工まで行うことによって現状把握されていない施工問題まで確認することができ、新たな施工方法の提案になると考える。

第四章　結論

本研究では「新たな木組み」を、相欠きの要素分析からインスピレーションを受け開発し、ねじり回転動作による嵌合形式の木組みを提案した。この木組みが持つ特徴として、①相欠きのような材と材が交差するような位置関係を取りつつ、各材のねじり角及び材同士の交差角を自由に設定できる、②木組みが締まる方向に力が加わったときにより材同士の接続が強固になる、の二点があげられ、今後の展望としてその特徴を活かした構造物の提案も行った。また五軸 CNC 切削機による切削では、誤差が生じてしまったものの三軸加工機ではなく五軸加工機によってこの木組みが加工可能であることを示した。

この木組みに関する今後の展望として、①加工精度向上②全体形構造物制作を通した施工及び実用性の提示、の二点があげられる。

加工精度向上について、本研究では五軸 CNC 切削機自体の軸のズレと g-code 生成プログラムの二つの問題点が生じた。機械自体のズレの影響を小さくするためには、まず機械の状況を把握する必要があり今後センシングなどの技術を駆使して新たな仕組み構築が必要である。これを行うことにより機械のキャリブレーションの自動化が可能になる可能性が考えられ、切削の効率化にもつながる。g-code 生成プログラムについてはズレが生じる部分のアルゴリズムの改善が必要であり、また Toolpath の効率化も求められる。

次に全体形構造物制作を通した施工及び実用性の提示について、本研究では全体形の提案は行ったものの実際に施工は行っておらず、今後これを行うことによって、新たに生じるであろう問題点を洗い出し改善することでこの木組みの実用可能性及び施工性、建築意匠の可能性を示すことができると考える。

最後に本研究での目的である木組み技術全体の発展について、本研究では今まで一般に使用されていなかったねじり回転動作による木組みを提案することに成功し、このことによってデジタルファブリケーションを用いた木組み開発に新たな気づきを与えることができたと考える。これにより今後の木組み開発が活発化しさらなるバリエーションを生むことを期待する。

参考文献

1. 木組みの美　岡部材木店　https://zaimokuten.com/ie/kigumi （最終閲覧日 2021 年 9 月 20 日）
2. アーカイブ検索　一般社団法人日本建築学会　https://www.aij.or.jp/paper/search.html （最終閲覧日 2021 年 9 月 3 日）
3. 野口 傑史、元池 遼、會田 健太朗、髙林 弘樹、田中 智己、平沢 岳人（2015）.「汎用的な部品加工を目的とした五軸加工機の制作」.『日本建築学会大会学術講演梗概集（2015 年 9 月）』. 37-38.
4. 中村 優介、漆山 生羽、林 真那、古庄 玄樹、加戸 啓太、平沢 岳人（2018）.「五軸加工機による丸ノコを用いた加工パス導出に関する研究」.『2017 年度日本建築学会 関東支部研究報告集Ⅱ（2018 年 3 月）』. 475-478
5. 古庄 玄樹、吉岡 直希、Randi HANTORO、大谷 星輝、中村 優介、加戸 啓太、平沢 岳人（2019）.「ATC 付き五軸加工機のフォリー制作を通した可用性検討」.『日本建築学会情報システム技術委員会 第 42 回情報・システム・利用・技術シンポジウム論文集』. 346-349.
6. 中村 優介、高橋 雅生、戸田 勇登、佐藤 清貴、髙林 弘樹、加戸 啓太、平沢 岳人（2017）.「五軸加工機による新しい校倉構法に関する研究」.『2016 年度日本建築学会 関東支部研究報告集Ⅱ（2017 年 2 月）』. 299-302.
7. 中村 優介、古庄 玄樹、林 真那、漆山 生羽、加戸 啓太、平沢 岳人（2018）.「丸ノコツールのパス自動生成システムの開発と留めで構成する木質ドームの制作 多軸加工によるあたらしい建築構法の創出研究　その 2」.『日本建築学会技術報告集　第 24 巻　第 58 号』. 1299-1302.
8. RANDI HANTORO、古庄 玄樹、中村 優介、加戸 啓太、平沢 岳人（2020）.「五軸加工機による木造レシプロカル･タワーの制作」.『日本建築学会大会学術講演梗概集（関東）（2020 年 9 月）』. 919-920.
9. 中村優介、高橋雅生、戸田勇登、林　真那、髙林弘樹、加戸啓太、平沢岳人（2018）.「五軸加工機による校倉構法の断面加工に関する研究　−多軸加工によるあたらしい建築構法の創出研究　その 1 −」.『日本建築学会技術報告集　第 24 巻　第 56 号』. 447-450.
10. Mikayla Heesterman、Kevin Sweet（2018）.「Robotic Connections: Customisable Joints for Timber Construction」.『XXII CONGRESSO INTERNACIONAL DA SOCIEDADE IBEROAMERICANA DE GRÁFICA DIGITAL』. 1-8.
11. Alfredo Salgado Ferrer、Fabrizio Tozzoli（2020）.「Digital Timber Reciprocity」 issuu https://issuu.com/alfredosalgadoib/docs/portfolio.final （最終閲覧日 2022 年 1 月 9 日）

謝辞

本論文の執筆にあたり、佐藤淳先生からは丁寧かつ熱心なご指導を賜りました。コロナ禍で例年のように対面でのミーティングを行うことができないこともありましたが、毎度たくさんのアドバイスをいただき、研究のためだけでなく今後の学びになるような気付きをたくさん得ることができました。ここに感謝の意を表します。ありがとうございました。

また、論文のみならず研究室活動の中でたくさんのサポートをしてくださった佐藤淳構造設計事務所の皆さまや、研究室の学生皆さんにも感謝しております。様々なプロジェクトの中で体力的に厳しい場面もありましたが、皆さんのおかげで楽しく充実した研究室生活を送ることが出来ました。特に Dr. のマリヤさんには五軸故障の時など何度も助けていただき、そのサポートが無ければ論文提出までたどり着けなかったかもしれません。ありがとうございました。そして、同期の氏岡、伊勢坊、今井、カザウィには 2 年間本当にお世話になりました。これからもよろしくお願いいたします。今後の皆さんの活躍を祈念します。

最後に、大学 4 年大学院 2 年に加え休学期間 1 年の計 7 年もの間、勉強する環境をつくり温かくサポートしてくれた家族に感謝します。

学生生活に関わってくださったすべての方に感謝の意を表し、謝辞にかえさせていただきます。

2022 年 1 月 17 日

宮野　公輔

付録

6　g-code 生成のための Grasshopper プラグイン

本章では 2.4.2 ～ 2.4.6 で示した各 Grasshopper ファイルのアルゴリズムについて、アルゴリズムのフロー図と実際のコンポ―メントの構成を示す。以下付録の目次。

6.1　Roughing

2.4.2 で大まかに示した Roughing アルゴリズムの流れを以下に示す。

0. 各種入力値の設定
1. 削る部分の 3D モデルを調整
2. 削る部分のスライス
3. スライスして得た各層での経路設計
4. 一本の経路（toolpath）にまとめる
5.toolpath 上のドリル制御平面の設定
6.toolpath planes から座標 X,Y,Z、回転角 B,C を算出
7.g-code 生成及びデータの受け渡し

これら各段階におけるアルゴリズムの詳細を以下に示す。また実際に作成した Grasshopper ファイルの画像を掲載する。

6.1.0　各種入力値の設定

⓪ドリル、ヘッドに関わる値の設定

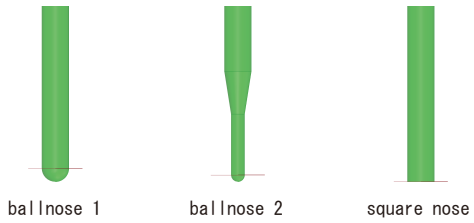
・endmill type

ドリルの形状（先端が半球形、円柱形、途中でテーパーしてる等）を数字で指定。後の分岐で使用。

0：ballnose 1
（先端が半球形、テーパーなし）

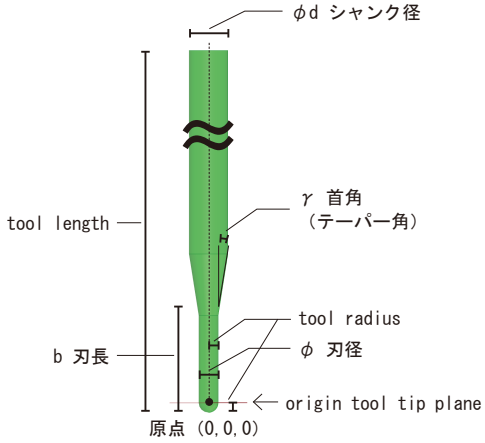
1：ballnose 2
（先端が半球形、テーパーあり）

2：square nose
（先端が円柱形）



・φ刃径，b刃長，γ首角，φdシャンク径

ドリル形状を決める値。右図参照。

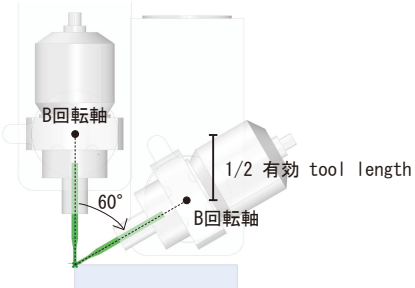


・tool radius

ドリル先端の半径。φ刃径の半分。

・origin tool tip plane

切削のシミュレーション時に使用する単位平面（座標とZ軸ベクトル）。ドリルの切削向き設定の回転原点にも使用。endmill typeの値に応じて、ドリル先端の形状が半球の場合その半球の原点（0,0,tool radius）、それ以外ではドリルの先端（0,0,0）に設定。

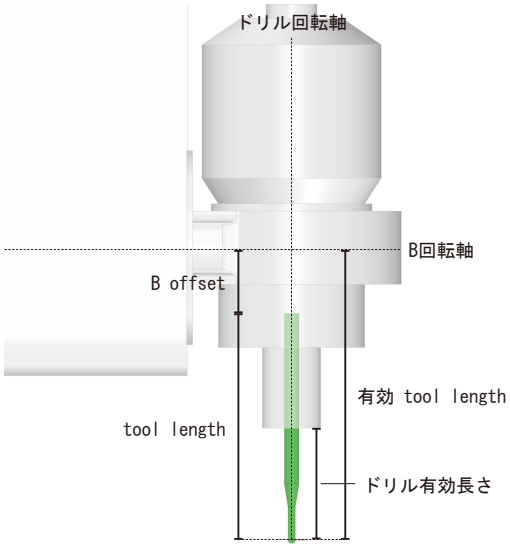


・1/2 有効 tool length

5Axismakerでドリルを下向きで材料原点に接触させた時とドリルを60°に傾けて接触させた時のZ座標の差。

・有効 tool length

1/2 有効 tool length を2倍し origin tool tip plane から B 回転軸までの長さを求める。



・B offset

B回転軸からのオフセット距離

・tool length

ドリルの描画長さ。

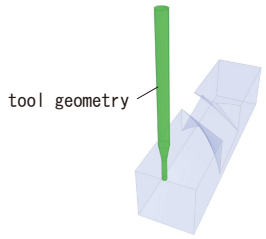
有効 tool length － B offset で定義。

・ドリル有効長さ

ヘッドから飛び出ているドリルの長さ。

・tool geometry

以上の値を用いて作成したドリルの 3D モデルデータ。



① 3D モデルデータの読込み

作成した加工 3D モデルを読み込む。

・木材

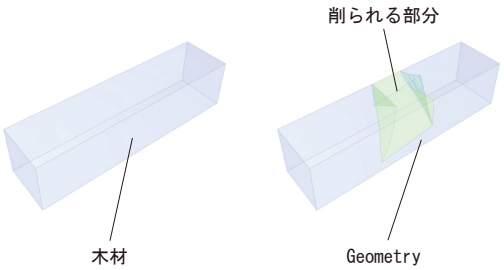
加工前の材料の 3D モデルデータ

・Geometry

加工後の 3D モデルデータ

・削られる部分

加工によって削られる部分の 3D モデルデータ。

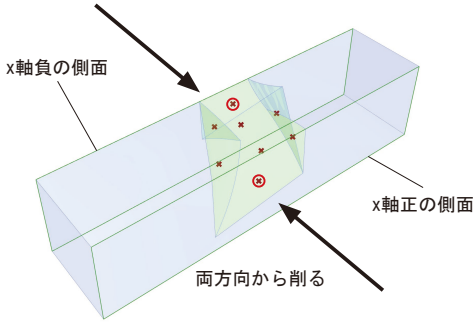


②削り方、木材の寸法、当たり判定モデル

・削り方

削られる部分各面の重心が、木材の x 軸正と負の側面上に存在するか判定。削られる部分が複数ある場合にはそれぞれの 3D モデルごとにこの操作を行う。

→正と負の側面どちらにも存在する場合、削られる部分が x 軸方向に貫通しているため x 軸正、負の二方向から切削を行う必要がある。（Ture or False）

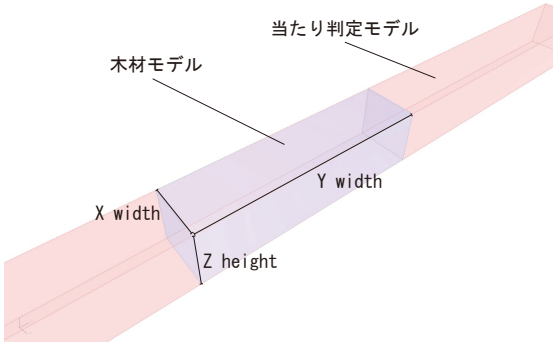


・木材の寸法

x,y,z それぞれの長さを求める。

・当たり判定モデル

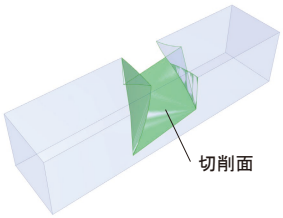
実際に切削を行う場合、長い材料を使用するため、木材 3D モデルを y 方向に拡大しヘッドの当たり判定を行う際に使用する。



③面の法線方向の整理等

・切削面の取り出し

Geometry と削られる部分の各面における中心を比較し、共通する面（切削面）を求める。



・各切削面の中心での法線ベクトル整理

取り出した切削面中心における単位法線ベクトルを求め、それぞれの中心点を法線ベクトル方向に移動。移動した点が Geometry の内か外か判定し、内であれば法線ベクトルが内向きになっているので切削面の表裏をひっくり返す。全ての法線ベクトルが外向きになるようにする。

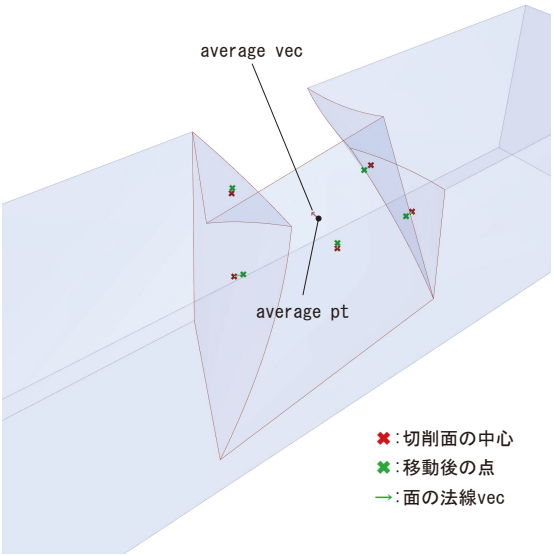
・各種値

面の数 : 切削面の数

面の法線 vec : 切削面中心の法線ベクトル

average vec : 切削面法線ベクトルの平均

average pt : 切削面中心点の平均座標



6.1.1　削る部分の 3D モデルを調整

④削られる部分をオフセット

Roughing では大まかに全体形を削っていくので、削られる部分をオフセットしてできた形状に対して切削経路を作成していく。

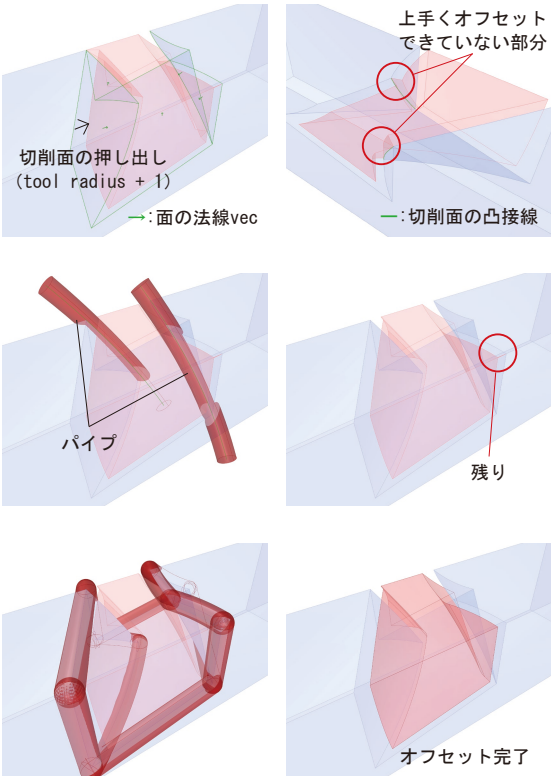
単純に削られる部分をオフセットすると右図の上面や側面にあたる部分が余計にオフセットされ経路作成に悪影響が出る。

そこでまず切削面を法線ベクトル方向にオフセット距離分（tool radius + 1）押し出したものを削られる部分から引き算する。しかしこの操作では切削面同士の接線で凸になっている部分が上手くオフセットされない。

次にこの凸接線部分を引き算することを考える。凸接線を始点、終点の両方向に延長し、その曲線でパイプを作成、それらを上の操作で得られた部分から引き算する。

最後に少し残ってしまった髭のような部分を消すために切削面の凸接線以外のエッジでパイプを作成し、一つ上の操作で得られた形状から引き算する。

以上の操作によって削られる部分のオフセットが完了した。



6.1.2　削られる部分のスライス

⑤調整した削られる部分をスライス

3D モデルをスライス（コンター化）するにあったって以下の三つのパラメータが必要になる。

・contour direction

　コンター化する方向のベクトル。②で判定した削り方（x 正と負の両方向から削るか動か）に応じて、True なら（1,0,0）と（-1,0,0）の二方向、False なら③で求めた average vec を用いる。

・step down

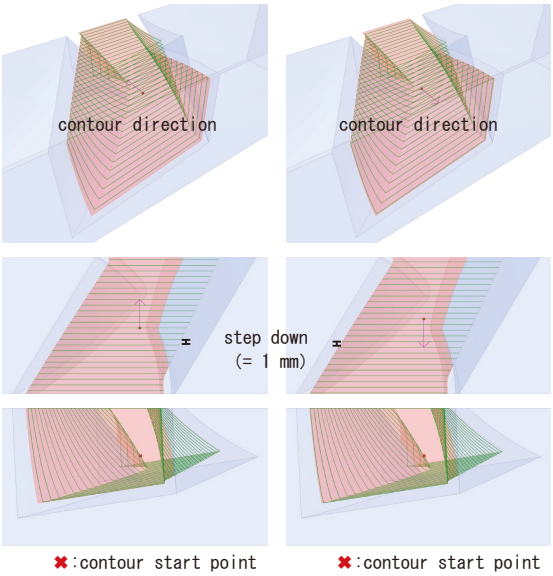
　スライス（コンター）幅。Finishing 時に刃になるべく負荷をかけないために tool radius と同程度にするのが好ましい。

・contour start point

　コンター化のスタート（基準）位置。average pt を基準に contour direction の何%分か動かした（各自設定）点とする。

・Roughing vec

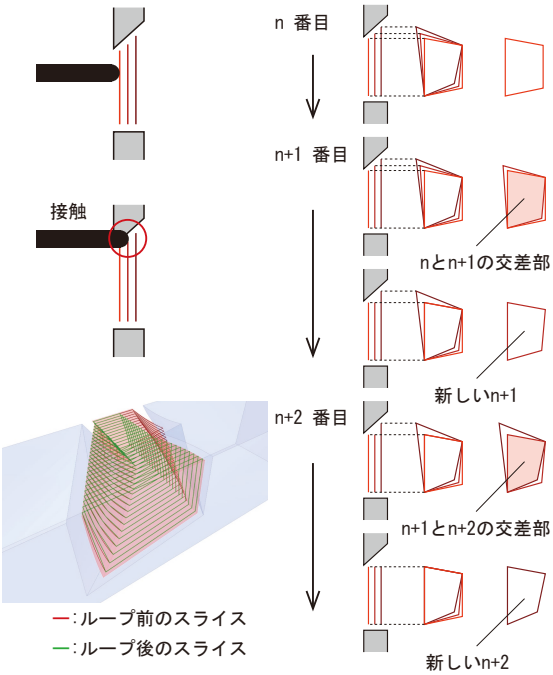
　contour direction を反転させたもの。ドリルの向きを決定する時に使用。



⑥スライスしてできた各層のはみ出し修正

ドリルで切削していく時、各層を順番に削っていくが、基本的に前の層よりも処理中の層が飛び出ていた場合、その部分をトリムしなければドリルと材料が接触する。そこで「n 番目と n-1 番目の層で交差する範囲を新たな n 番目の層とする」という操作を n = 1,2,3,…, 層数に対して行うループ処理が必要になる。

ループ処理をするにあたって計算を軽くするため、スライスされた層は閉じた同一平面曲線であったが、全てポリラインに変換する。



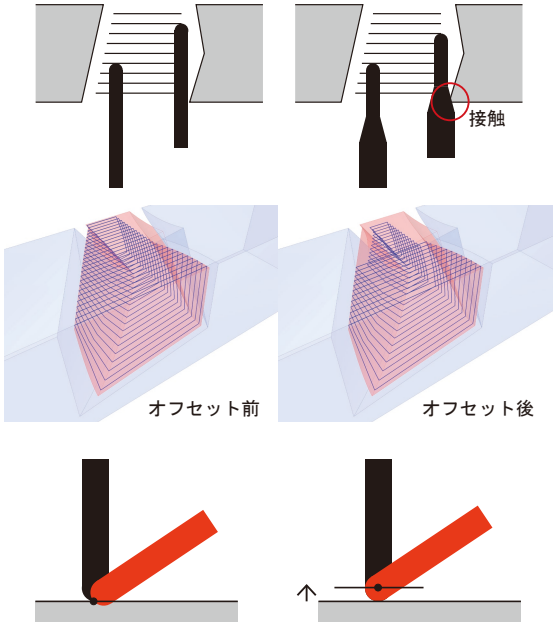
⑦ドリルの形状による各層の調整

・ドリルがテーパーしている

ドリルの形状が途中でテーパーしている場合、切削中に材料と接触する可能性がある。そのため接触する層以降はさらにオフセットする必要がある。

・ドリル先端が半球状

ドリルの先端形状が半球である場合、ドリル先端を中心にドリルの角度を調整すると余計な部分を削ってしまう可能性がある。それを回避するため、半球の半径（tool radius）分 contour direction の方向に層全体を移動させる必要がある。

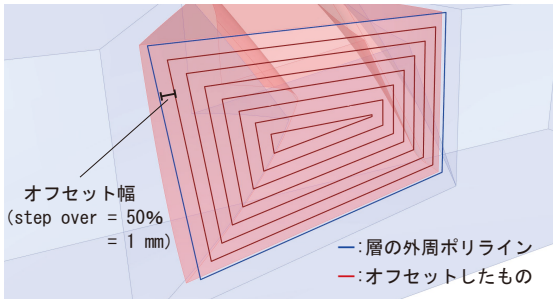


6.1.3　スライスして得た各層での経路設計

各層を削っていくための経路を考える。

⑧全ての層で面内オフセット

右図のように各層の外周ポリラインの内側にオフセットできなくなるまで繰り返す。オフセットの幅は tool diameter（= φ 刃径）の何%か（= step over）で指定する。



⑨極小ポリラインの削除と
ポリラインのシーム点指定

同じ層内のポリラインを一本の経路にするため、閉じたポリラインを開いたポリラインにする必要がある。ポリラインを開くにはポリライン上のある一点の上で極小円を作成し、円内の部分をトリムする。このために極小円よりも小さなポリラインはトリムできないので削除し、その後ポリライン上の一点（シーム点）を定めていく。

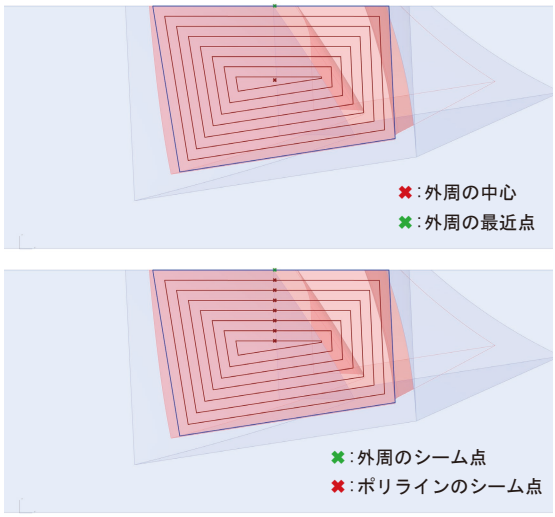
・シーム点の定め方

外周ポリラインの中心を求める。

→中心に最も近い外周上の点を求める。

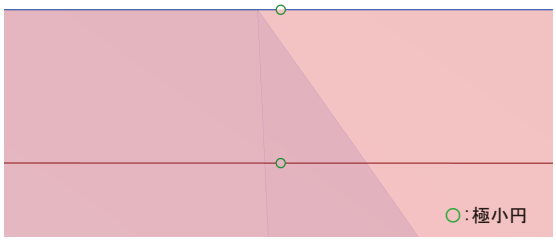
→その点に最も近いポリライン上の点

→シーム点



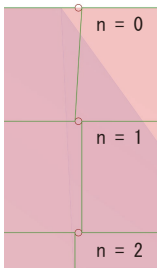
⑩閉じたポリラインから開いたポリラインに

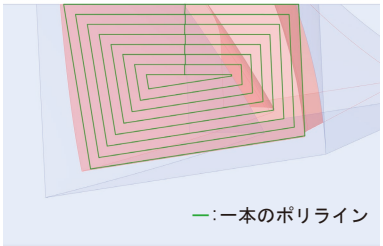
⑨で指定したシーム点に極小円（半径 0,03 mm）を配置し、その円の内部をトリムすることで閉じたポリラインを開いたポリラインにする。



⑪層のポリラインを一本の経路に

⑩で求めた開いたポリラインを層ごとに一本のポリラインに結合する。外側のポリラインから n = 0,1,2,…とすると n = 0 の終点と n = 1 の始点間で直線を引き、n = 1 の終点と n = 2 の始点間で直線を引き…とその層の最後のポリラインまで操作を行い、最後に一本のポリラインになるよう結合する。



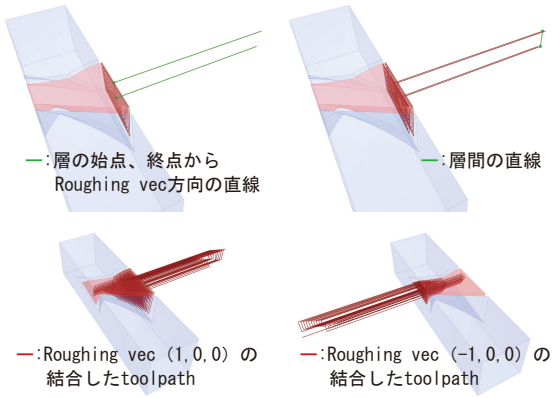


6.1.4　一本の経路（toolpath）にまとめる

⑫一本の経路（toolpath）にまとめる

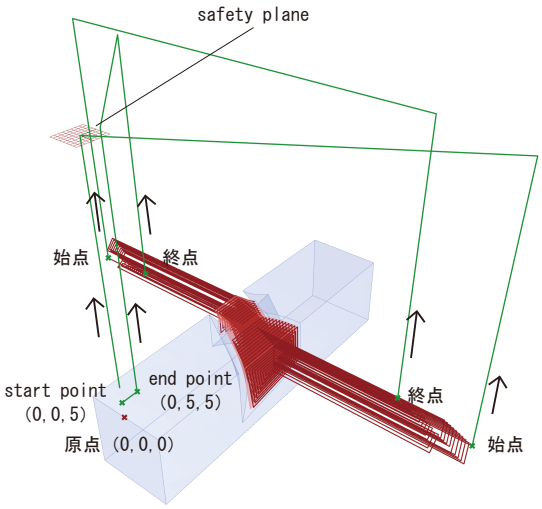
・各層間の経路

各層のポリラインの始点と終点から Roughing vec の方向に長さ 60 mm（ドリルと材料が絶対に接触しない長さに調整）の直線を作成し結合する。続いて第 n 層の終点と第 n+1 層の始点を繋ぐ直線を作成し結合する。右図では二方向から削る（Roughing vec が二つ）ので 2 本のポリライン（toolpath）が生成された。



・Roughing vec ごとの toolpath と切削開始点、終点を繋ぐ

切削開始時と終了時のドリルの先端の位置を決める（右図では（0,0,4）（0,5,4））。⑦で述べたように toolpath ではドリル先端ではなく先端半球の中心の動きを考えるので z 軸方向に tool radius 分足した（0,0,5）と（0,5,5）を start, end point とする。各 toolpath の始点、終点と start, end point から safety plane（ヘッド移動時に材料と緩衝しないような高さの XY 平面）に向かって直線を生成し、それらを結合して一本の toolpath にする。

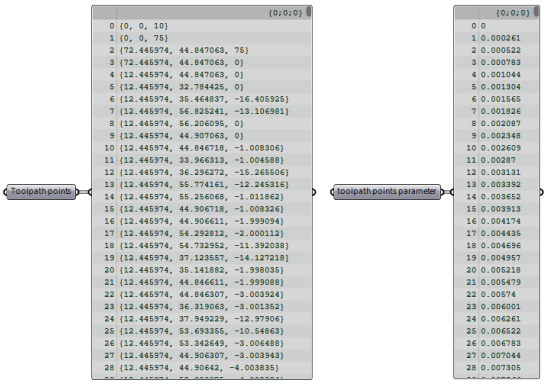


6.1.5　toolpath 上のドリル制御平面設定

⑬ toolpath points とパラメータ表示

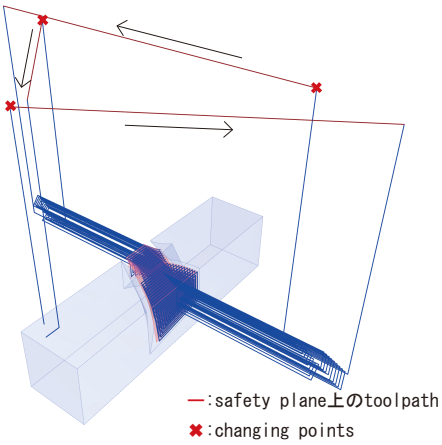
・toolpath points / parameter

toolpath は曲線データで g-code にするには適さないのでこれを点データにする必要がある。Roughing では toolpath がポリラインで構成されるため、ポリラインの頂点を toolpath points として使用する。また今後の操作のため toolpath 全体の長さを 1 とした時の始点から各 toolpath points までの長さを toolpath points parameter とする。



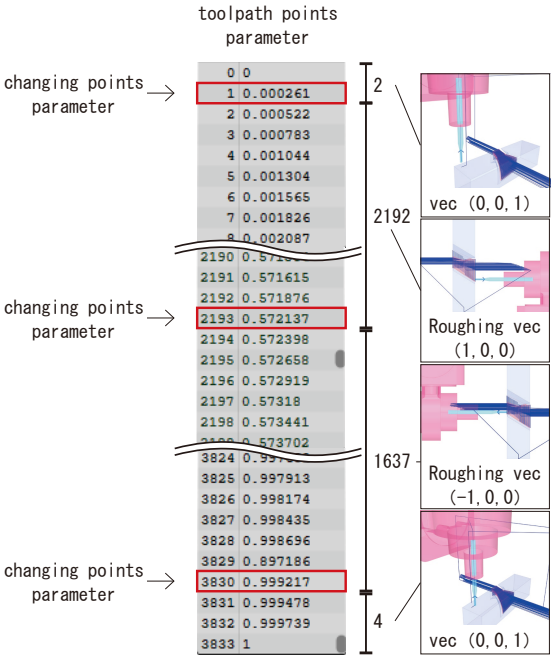
・changing points / parameter

切削時にドリルの向きが変わる、toolpath 上の点を changing points とする。ドリルの向きを変える際に材料と緩衝しないようにするため、changing points は safety plane 上の点とする。またヘッドは次の点へと動きながらドリルの向きを変えるため、safety plane 上の toolpath のうちパラメータが小さいほうを changing points とする。そのパラメータも求める。



⑭ toolpath points でのドリルの向きを設定

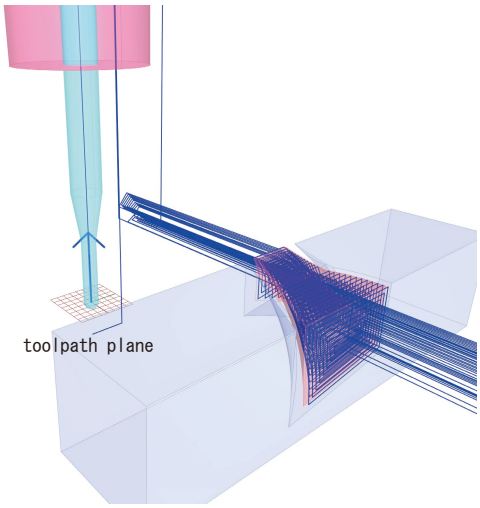
toolpath points parameter と changing points parameter を用いて toolpath points をグループ分けする。これにより切削中のドリルの向きで toolpath points を分けることができる。分けた toolpath points それぞれに対応するベクトルデータ（最初と最後のグループはドリルが下向きになるので（0,0,1）を入力、間は Roughing vec）を紐づけ、点データとベクトルデータを元に plane データを作成する。これを toolpath planes とする。



⑮ toolpath planes の整列、接触確認

⑭の toolpath planes では plane データとして座標と plane の z 軸方向ベクトルしか指定していないため、今後の操作で誤作動が起こらないように plane の x 軸方向を絶対座標の x 軸に揃える。

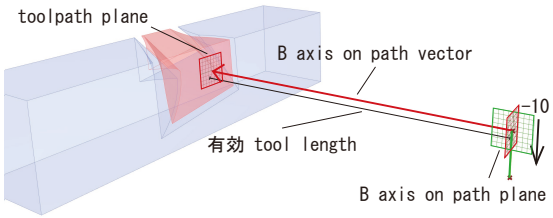
toolpath planes を作ることができたので、全ての平面に対して tool geometry を origin tool tip plane を起点に移動させ、geometry と tool geometry の衝突判定を行う。ここで衝突が確認された場合は⑭で toolpath planes を設定する際の Roughing vec を調整する。



6.1.6 toolpath planes から座標 X, Y, Z、回転角 B, C を算出

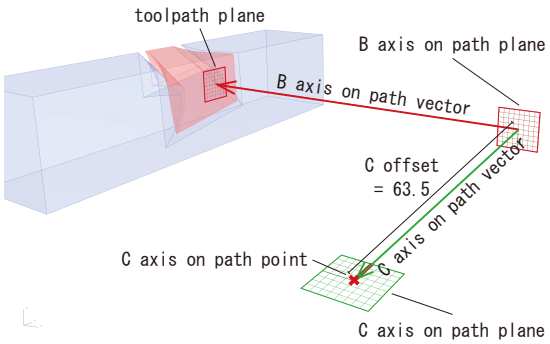
⑯ B axis に関して

toolpath plane をその z 軸方向に有効 tool length 分移動させる。その平面中心から toolpath plane の中心に向かうベクトルを B axis on path vector とする。B axis on path vector の始点を中心に、その点を全体座標 z 軸方向に -10 移動した点、toolpath plane の平面中心点の三点から平面を作成し、B axis on path plane とする。この平面がヘッド B 中心位置で C axis の起点になる。



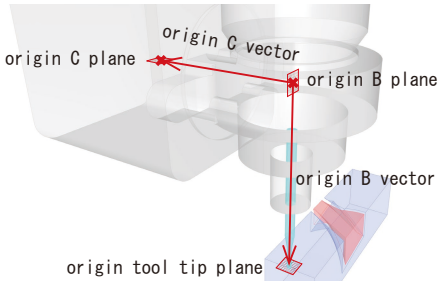
⑰ C axis に関して

B axis on path plane の z 軸方向に C offset（＝本研究の機械では 63.5 固定）分移動した点がヘッド C の中心になる（C axis on path point）。ただし z 軸の正負は各 toolpath plane におけるヘッド C と木材の当たり判定で決定する。B axis on path plane の中心から C axis on path point へのベクトルを C axis on path vector、C axis on path point 上に生成した XY 平面の x 軸を C axis on path vector の反転したものに合わせた平面を C axis on path plane とする。



⑱ヘッド描写の基準点

ドリルが真下を向いて先端が（0,0,0）にあるヘッドの状態を初期状態とする。この初期状態を基準にシミュレーションを行うため、⑯⑰で設定したBやCの平面に対応するものを設定する必要がある。



⑲ヘッド描写シミュレーション

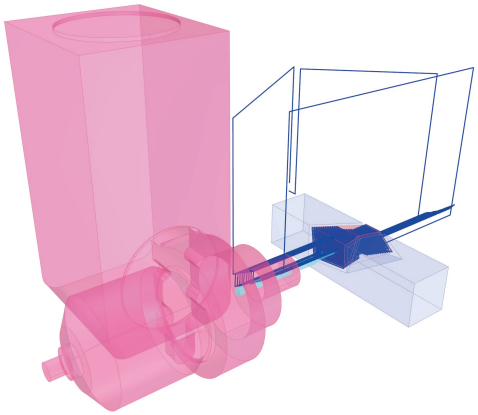
⑱での基準と対応する⑯⑰の平面を示す。

origin tool tip plane → toolpath plane

origin B plane → B axis on path plane

origin C plane → C axis on path plane

tool geometryやヘッドのB、Cをこれらの対応関係を元に初期状態から移動させることで各toolpath planeごとのシミュレーションを行うことができる。目視でもヘッドやドリルの材料との緩衝を確認する。



6.1.7　g-code生成及びデータの受け渡し

⑳ g-code生成及びデータの受け渡し

・g-code作成

ヘッドCの中心座標をg-codeで命令するので⑰で求めたC axis on path pointsと⑱で求めたorigin C planeの差を求める。これがg-codeに入力するXYZになる。

回転角Bは初期状態のorigin B vectorとB axis on path vectorのなす角をB axis on path plane上で計算することで求めることができる。

回転角Cも同様にorigin C vectorとC axis on path vectorのなす角をC axis on path plane（XY平面）上で計算し求める。

回転角B、Cに関してはヘッドの回転動作が最小になるように角度が-180°から180°の値になるように変換を行う（-270°などは避けるということ）。

・データの受け渡し

Finishingへ受け渡すデータを右に示す。g-codeはg-code generatorに受け渡す。

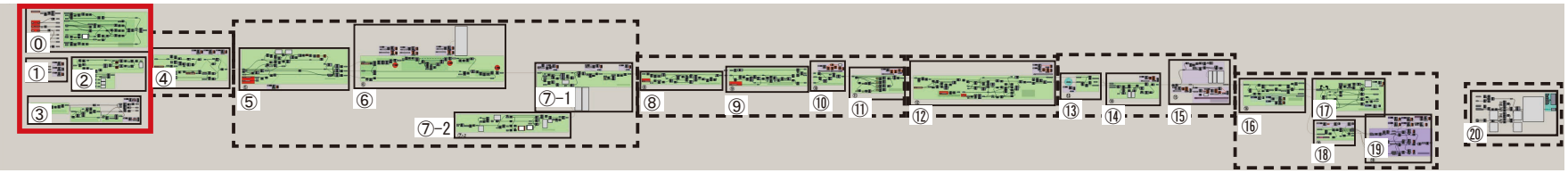
```
0 G1 X0.0000 Y0.0000 Z9.0000 A0.0000 B0.0000 C0.0000
1 G1 X0.0000 Y0.0000 Z74.0000 A0.0000 B0.0000 C0.0000
2 G1 X215.7460 Y-18.6529 Z-5.8000 A0.0000 B-90.0000 C-90.0000
3 G1 X215.7460 Y-18.6529 Z-80.8000 A0.0000 B-90.0000 C-90.0000
4 G1 X155.7460 Y-18.6529 Z-80.8000 A0.0000 B-90.0000 C-90.0000
5 G1 X155.7460 Y-30.7156 Z-80.8000 A0.0000 B-90.0000 C-90.0000
6 G1 X155.7460 Y-28.0352 Z-97.2059 A0.0000 B-90.0000 C-90.0000
7 G1 X155.7460 Y-6.6748 Z-93.9070 A0.0000 B-90.0000 C-90.0000
8 G1 X155.7460 Y-7.2939 Z-80.8000 A0.0000 B-90.0000 C-90.0000
9 G1 X155.7460 Y-18.5929 Z-80.8000 A0.0000 B-90.0000 C-90.0000
10 G1 X155.7460 Y-18.6533 Z-81.8083 A0.0000 B-90.0000 C-90.0000
11 G1 X155.7460 Y-29.5337 Z-81.8046 A0.0000 B-90.0000 C-90.0000
12 G1 X155.7460 Y-27.2037 Z-96.0655 A0.0000 B-90.0000 C-90.0000
13 G1 X155.7460 Y-7.7258 Z-93.0453 A0.0000 B-90.0000 C-90.0000
14 G1 X155.7460 Y-8.2439 Z-81.8119 A0.0000 B-90.0000 C-90.0000
15 G1 X155.7460 Y-18.5933 Z-81.8083 A0.0000 B-90.0000 C-90.0000
16 G1 X155.7460 Y-18.5934 Z-82.7991 A0.0000 B-90.0000 C-90.0000
17 G1 X155.7460 Y-9.2072 Z-82.8001 A0.0000 B-90.0000 C-90.0000
18 G1 X155.7460 Y-8.7670 Z-92.1920 A0.0000 B-90.0000 C-90.0000
19 G1 X155.7460 Y-26.3764 Z-94.9272 A0.0000 B-90.0000 C-90.0000
```

生成したg-codeの一部（5軸加工機なので回転角Aは常に0）

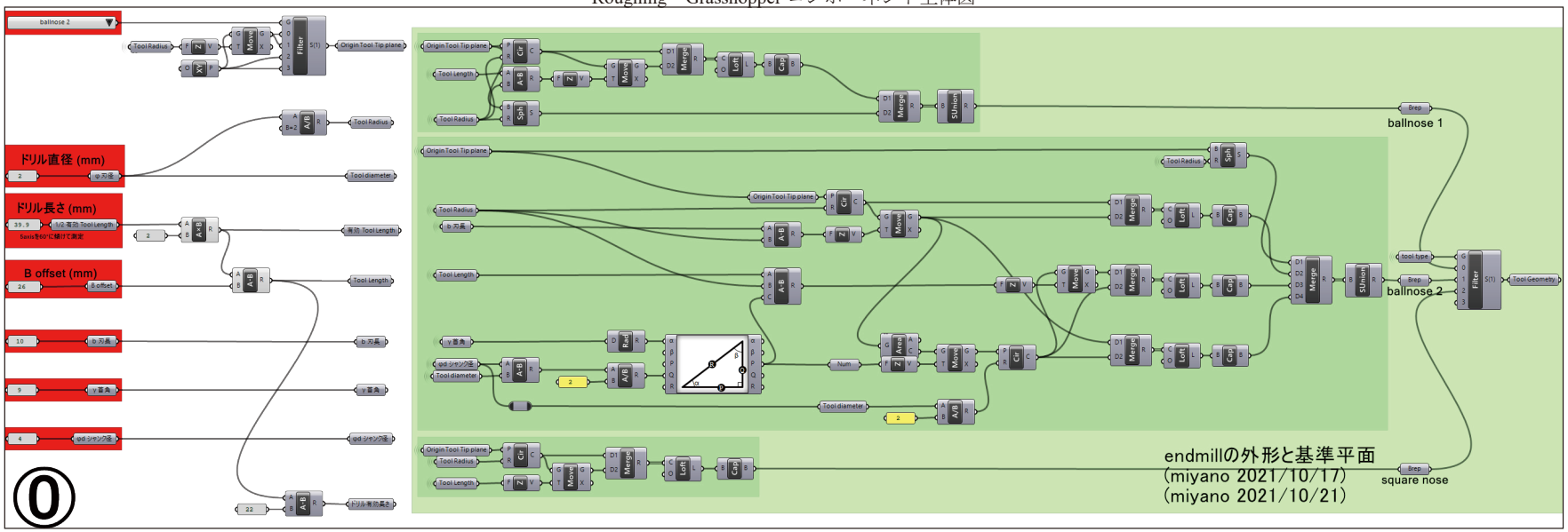
Finishing 1、2に受け渡すデータ

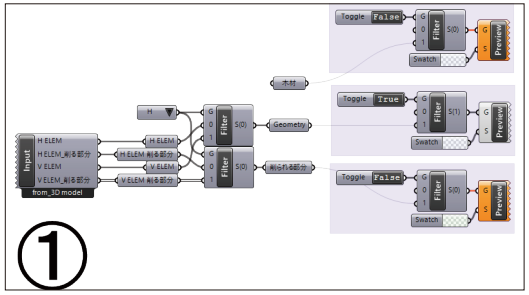
木材当たり判定
endmill type
start point
Geometry
切削面
面の法線vec
safety plane
origin tool tip plane
tool radius
有効tool length
γ首角
tool geometry
面の数

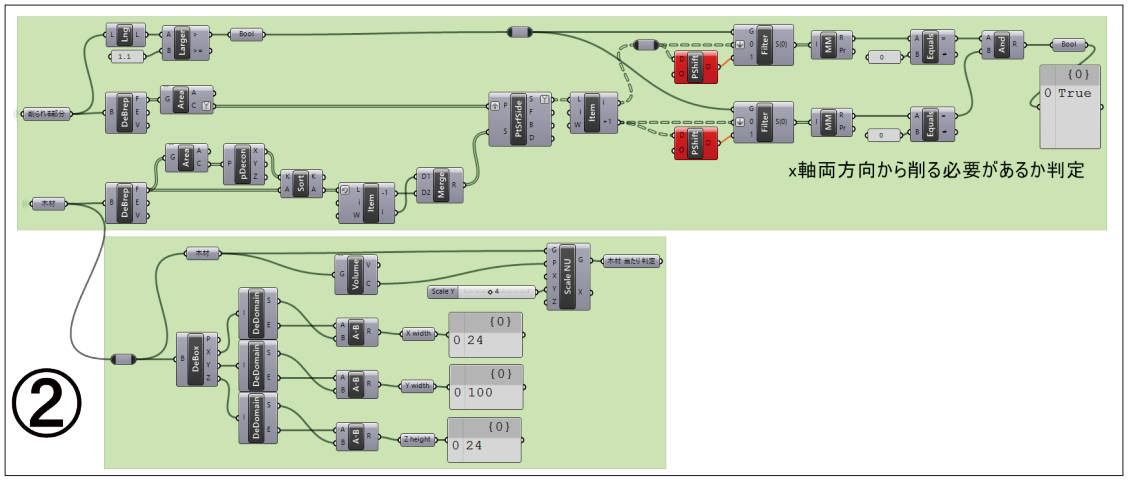
0. 各種入力値の設定

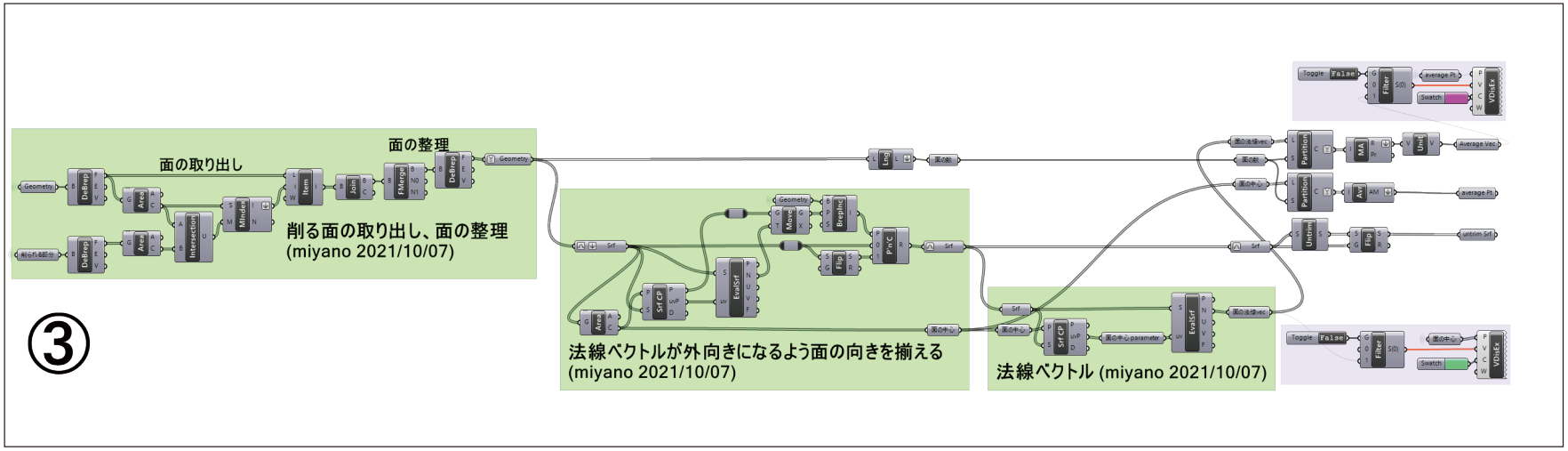


Roughing　Grasshopper コンポーネント全体図

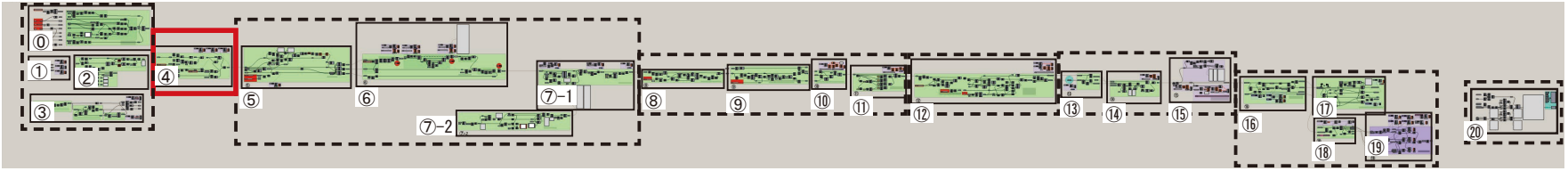




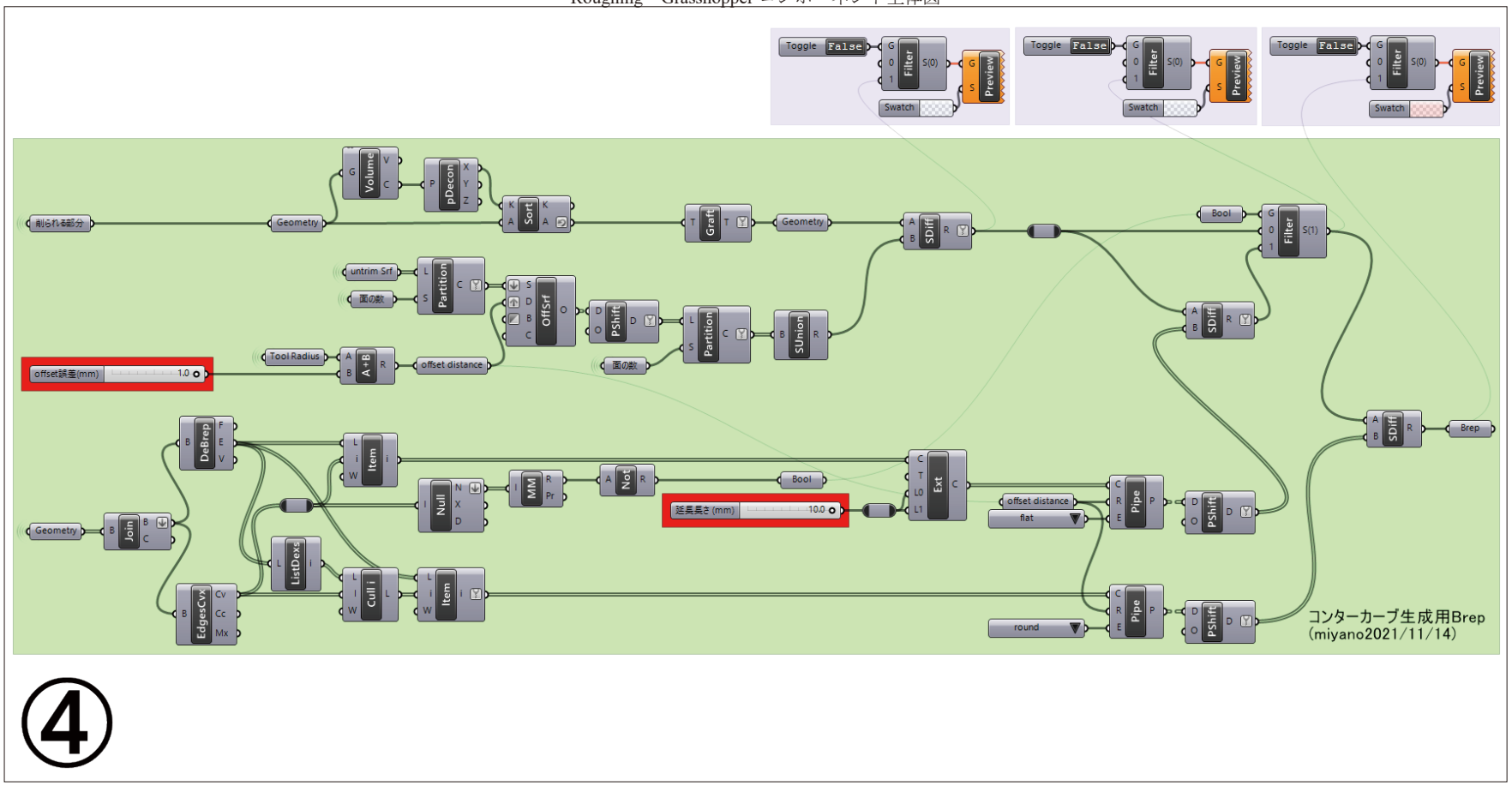

②
x軸両方向から削る必要があるか判定


③
面の取り出し
面の整理
削る面の取り出し、面の整理
(miyano 2021/10/07)
法線ベクトルが外向きになるよう面の向きを揃える
(miyano 2021/10/07)
法線ベクトル (miyano 2021/10/07)

1. 削る部分の 3D モデルを調整

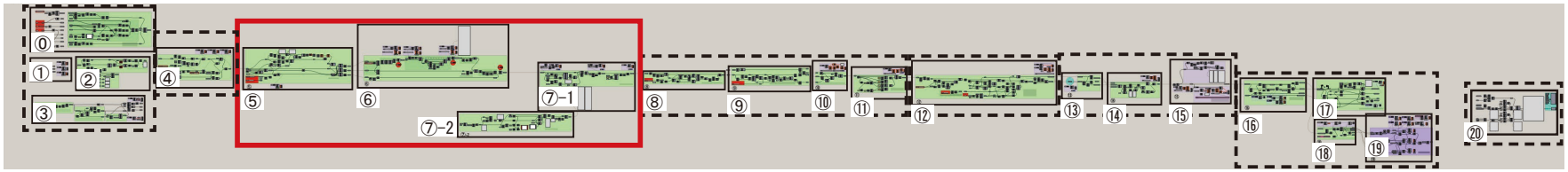


Roughing　Grasshopper コンポーネント全体図

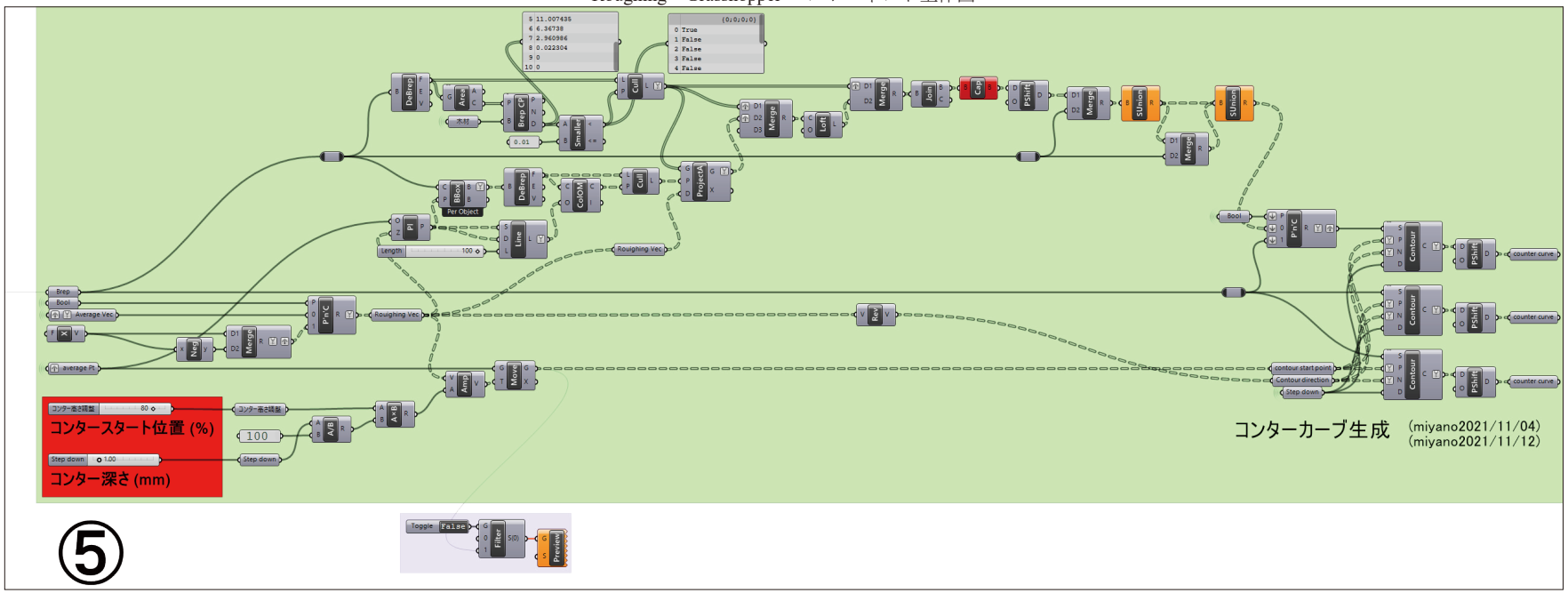


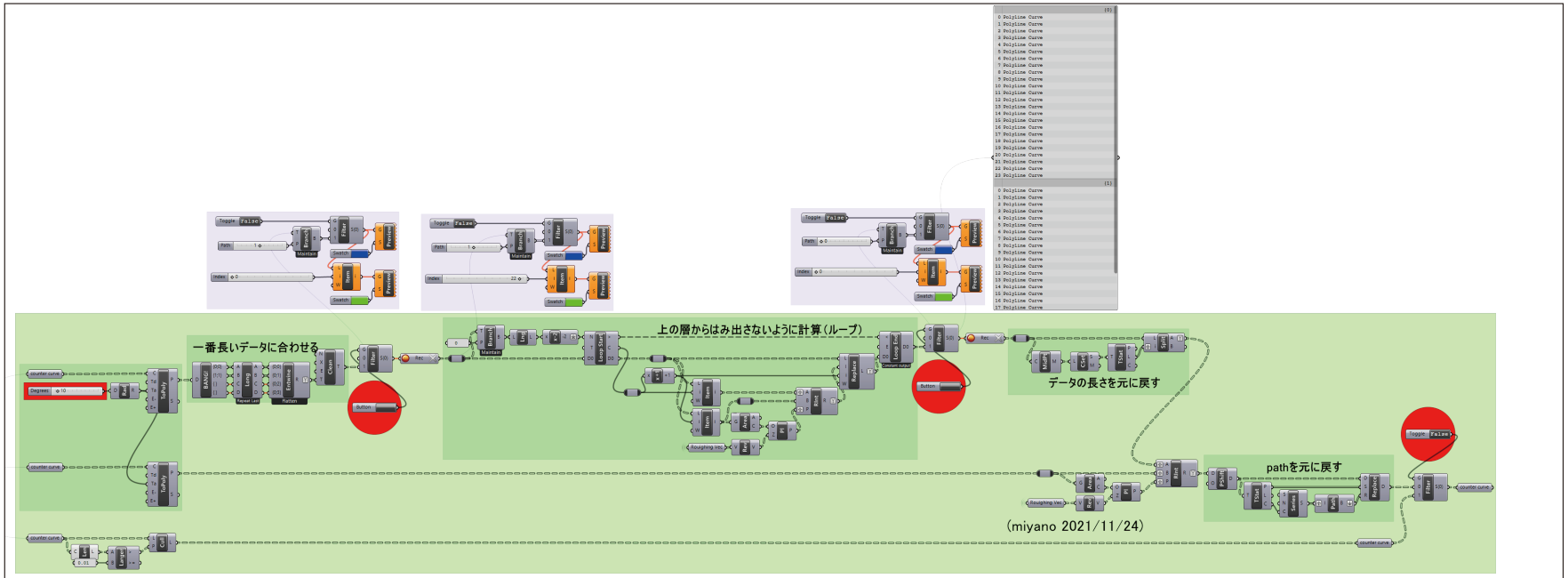
④

2. 削る部分のスライス

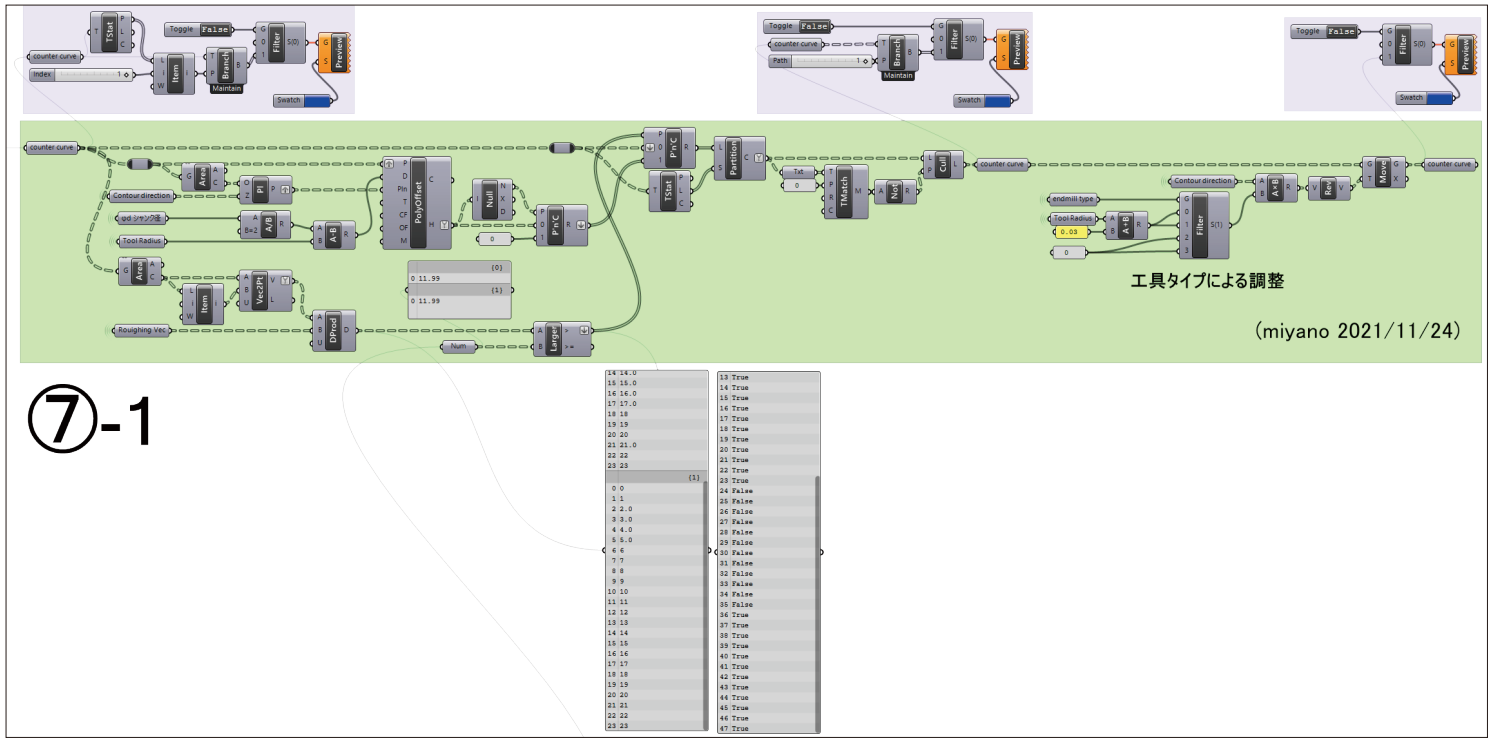


Roughing　Grasshopper コンポーネント全体図

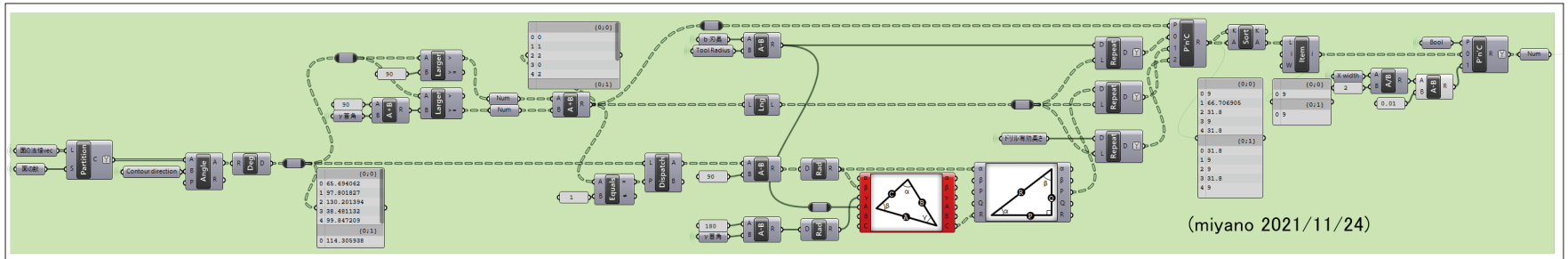



一番長いデータに合わせる
上の層からはみ出さないように計算（ループ）
データの長さを元に戻す
pathを元に戻す
（miyano 2021/11/24）

⑥

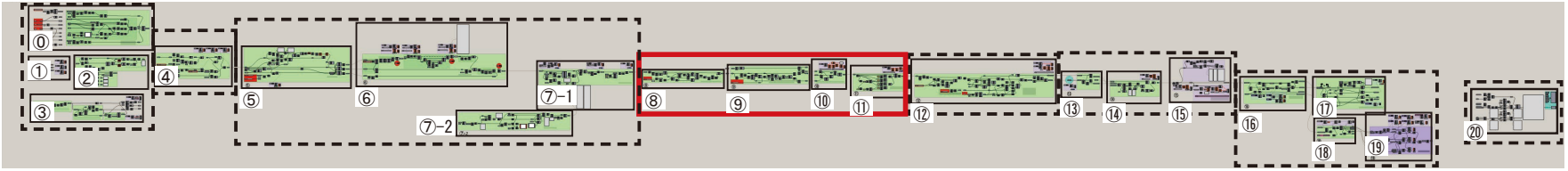


⑦-1

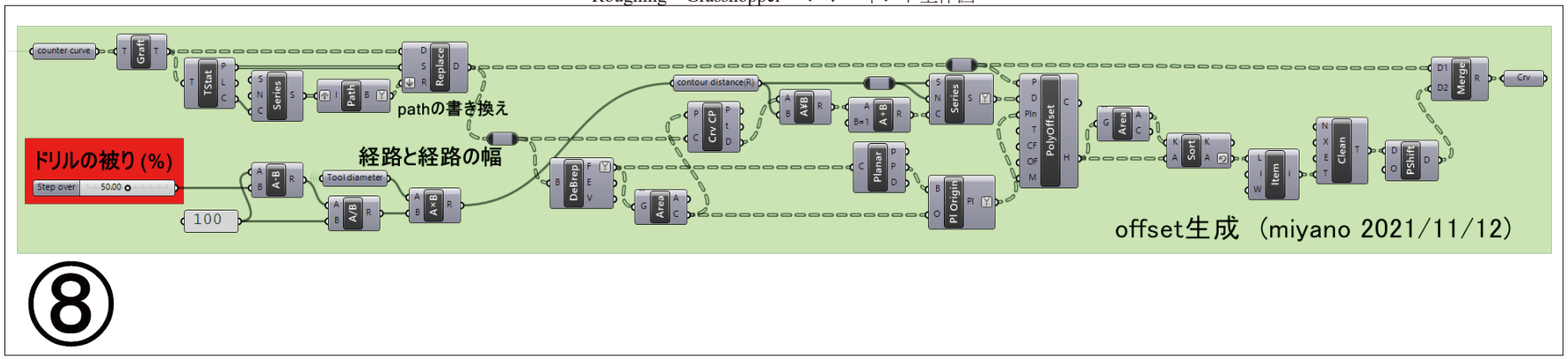


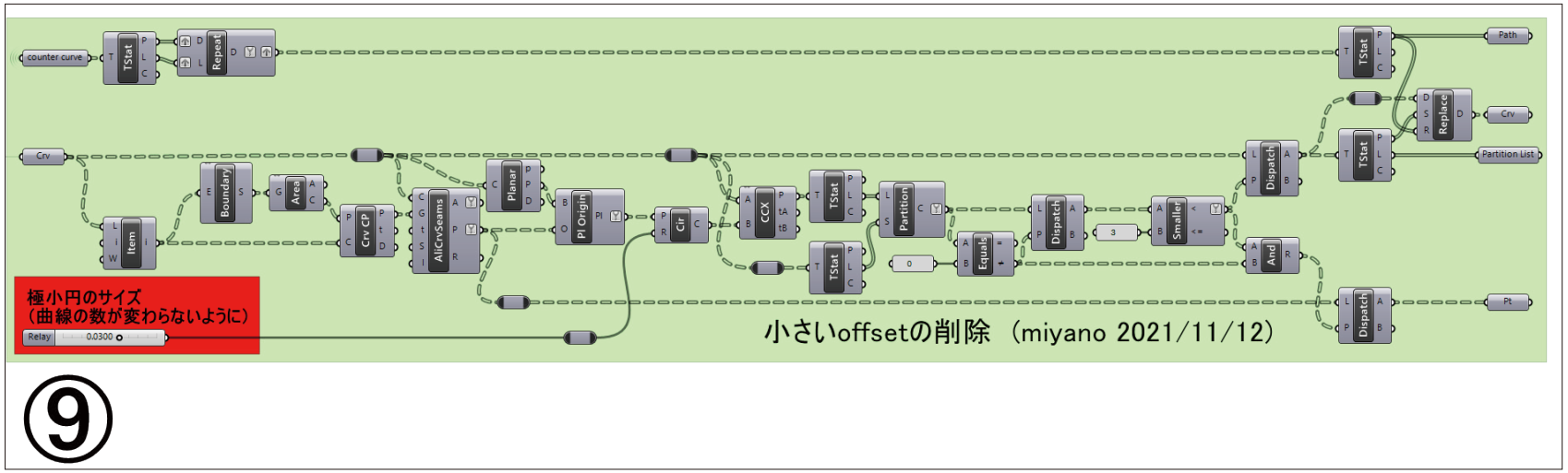
⑦-2

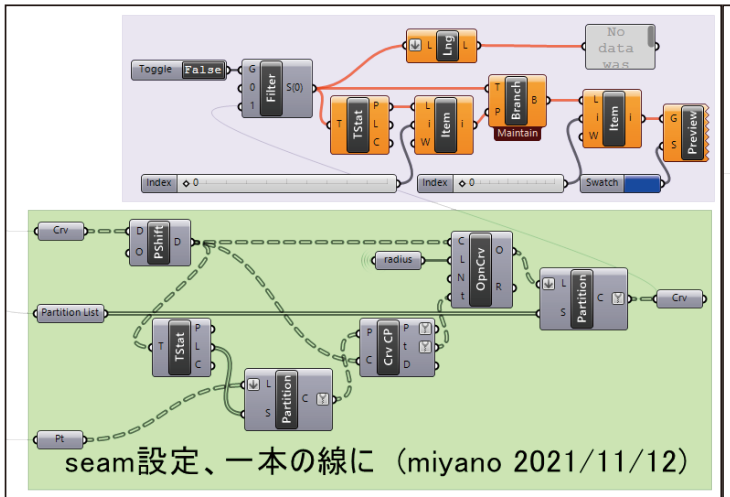
3. スライスして得た各層での経路設計



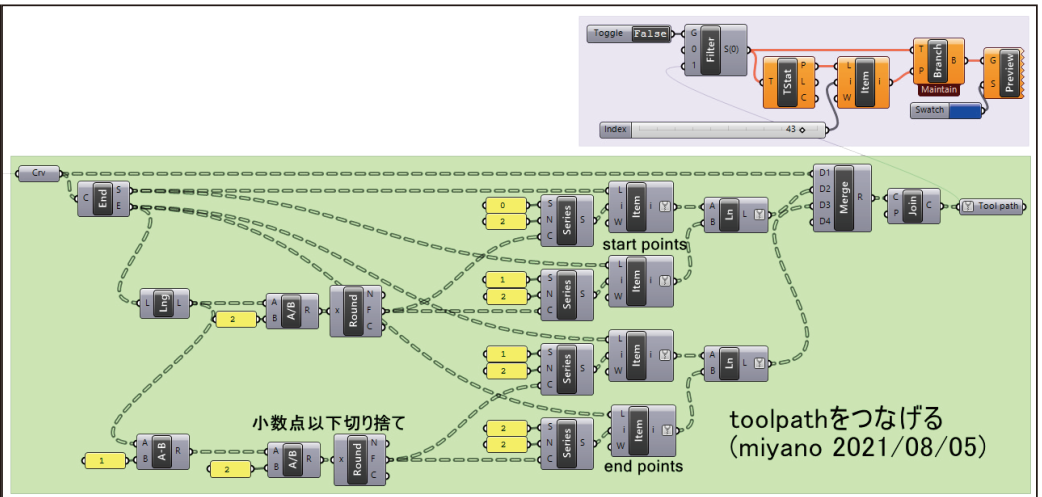
Roughing　Grasshopper コンポーネント全体図





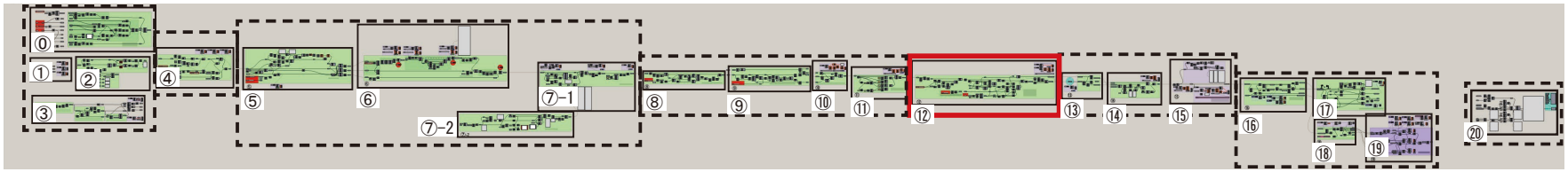

Toggle
False
Filter
Lng
No data was
TStat
Item
Branch
Maintain
Preview
Index
Swatch
Crv
PShift
radius
OpnCrv
Partition List
Partition
Crv CP
Pt
seam設定、一本の線に（miyano 2021/11/12）

⑩

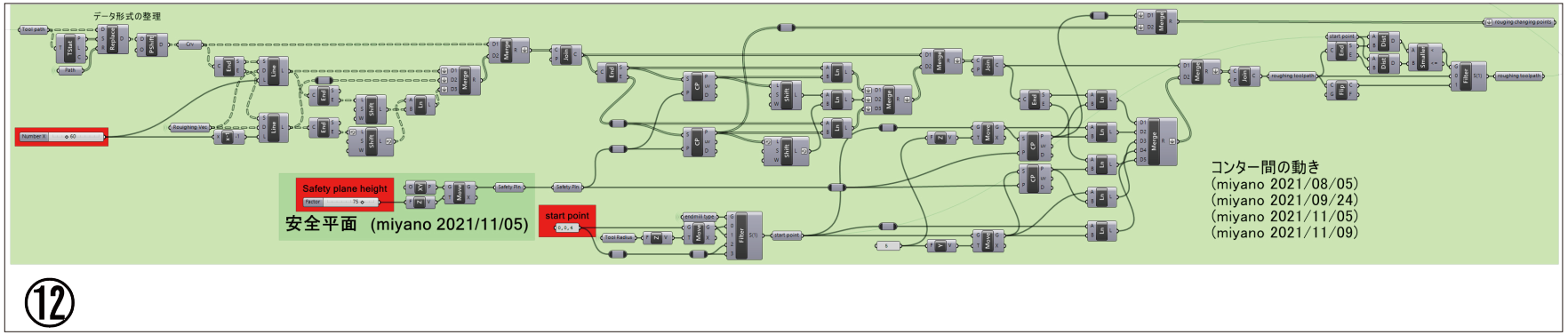

Toggle
False
Filter
TStat
Item
Branch
Maintain
Preview
Swatch
Index
Crv
End
Lng
A/B
Round
Series
start points
end points
小数点以下切り捨て
Ln
Merge
Join
Tool path
toolpathをつなげる
（miyano 2021/08/05）

⑪

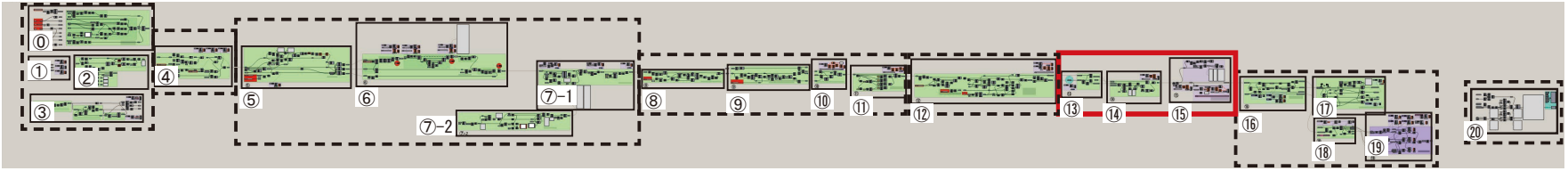
4. 層ごとの経路をまとめる



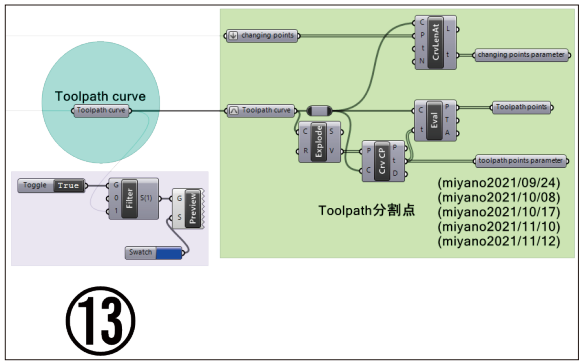
Roughing　Grasshopper コンポーネント全体図

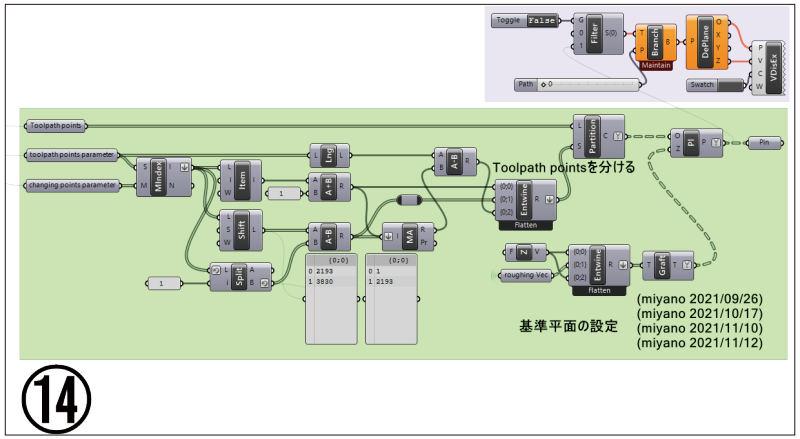


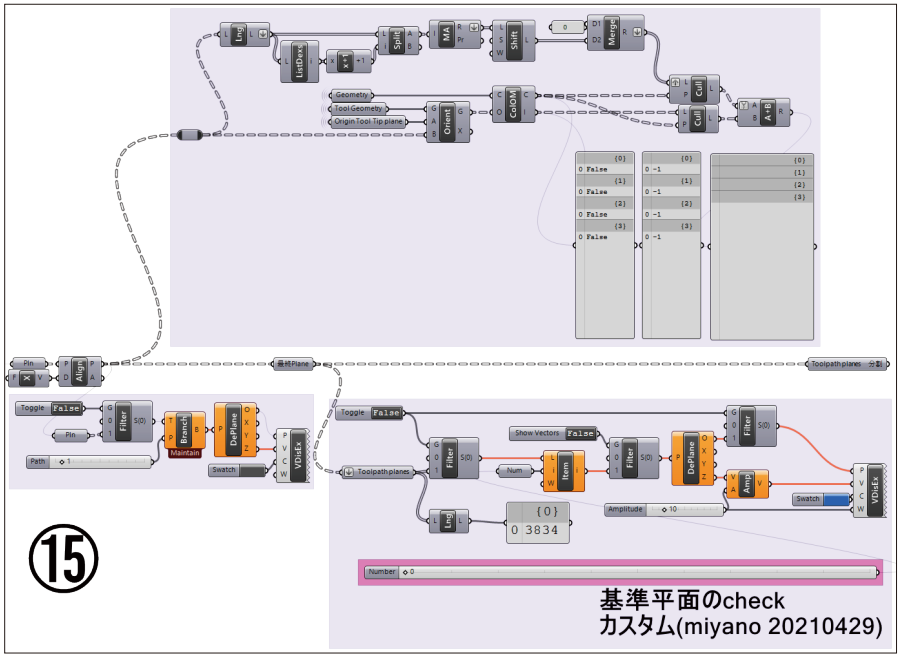
5. 経路上のドリル制御平面の設定



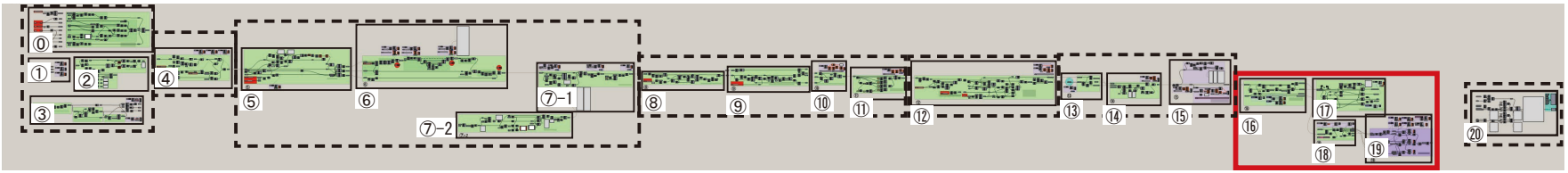
Roughing　Grasshopper コンポーネント全体図



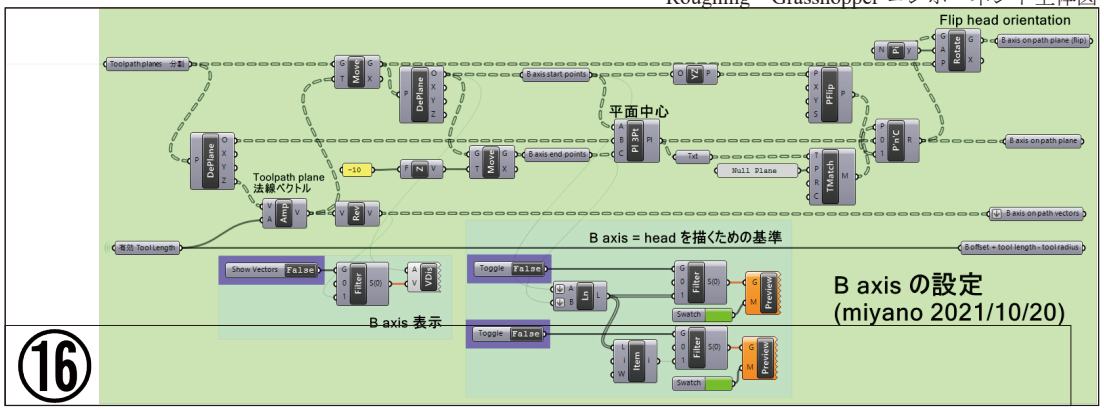


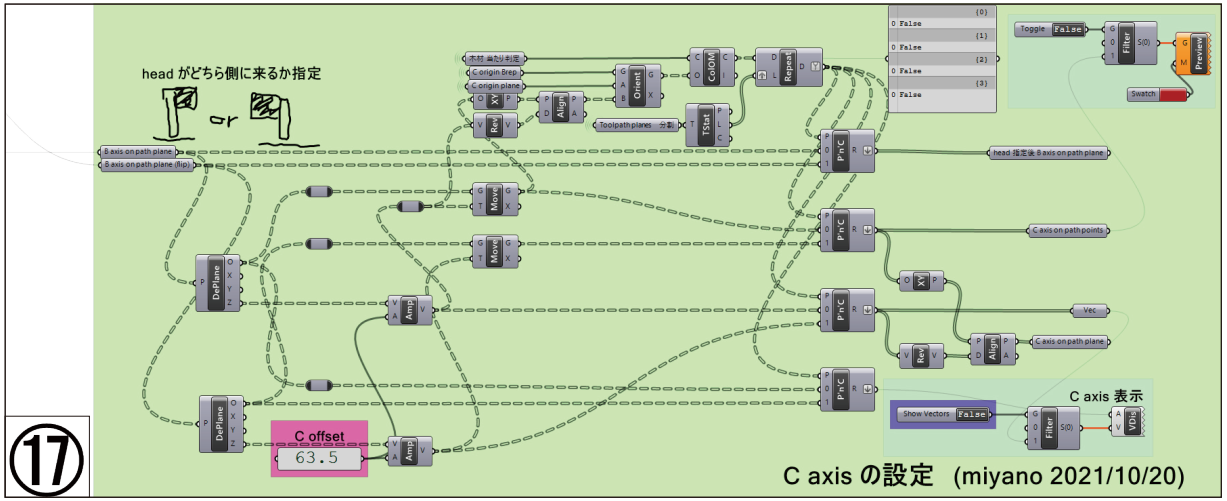


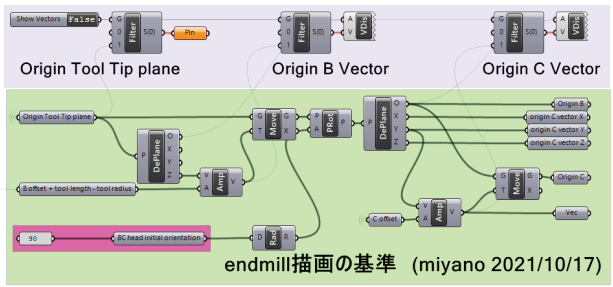
6. 各ドリル制御平面から座標 X, Y, Z 及び回転角 B, C を算出



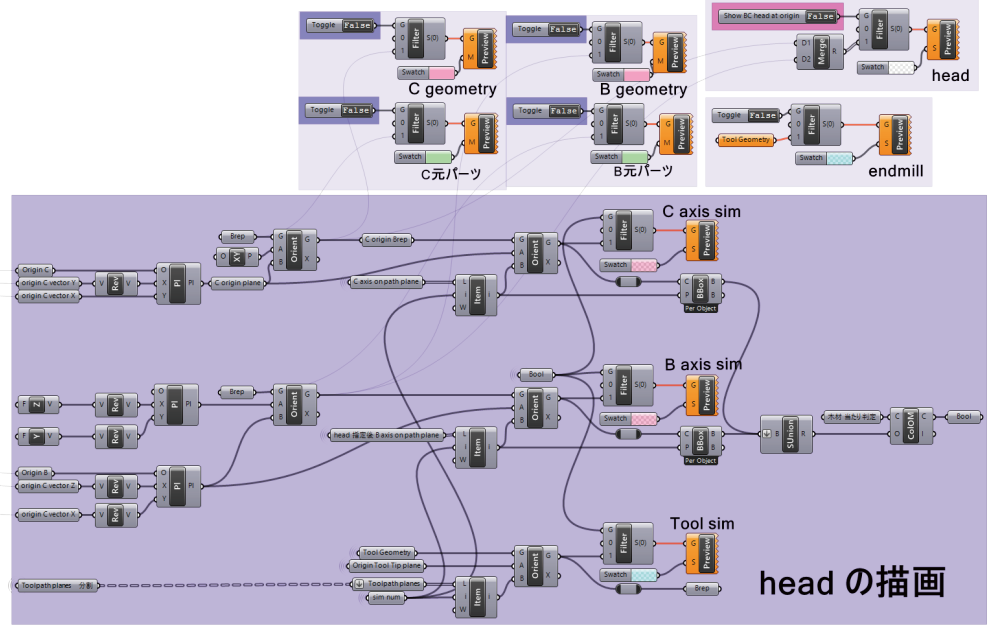
Roughing　Grasshopper コンポーネント全体図





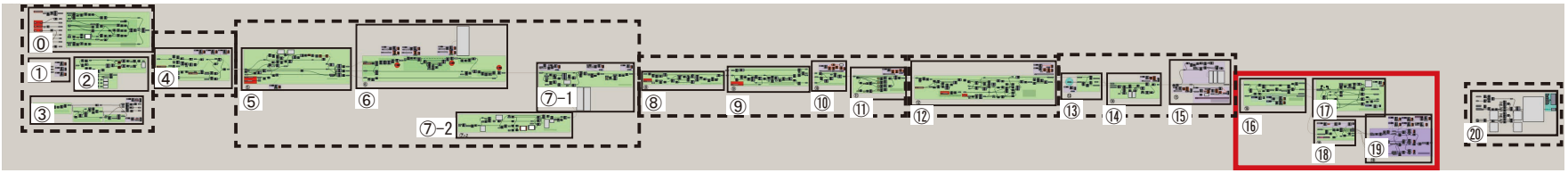

Origin Tool Tip plane
Origin B Vector
Origin C Vector
endmill描画の基準 (miyano 2021/10/17)

⑱

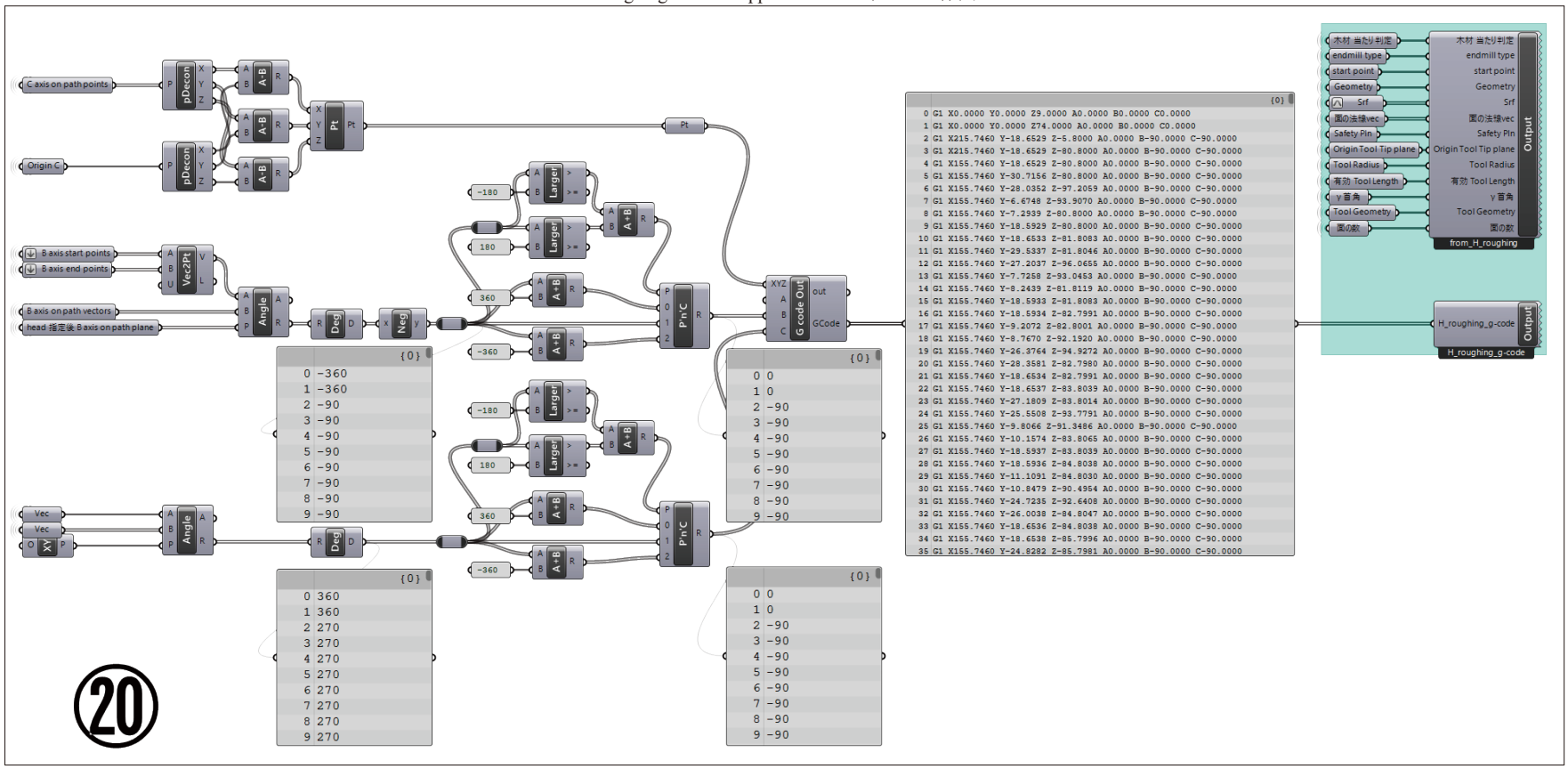

C geometry
B geometry
head
C元パーツ
B元パーツ
endmill
C axis sim
B axis sim
Tool sim
head の描画

⑲

7. X, Y, Z, B, C を元に g-code 生成及びデータの受け渡し



Roughing　Grasshopper コンポーネント全体図



6.2 Finishing 1

2.4.3 で大まかに示したが、Finishing 1 のアルゴリズムの流れを再び以下に示す。

0. 各種入力値の設定
1. 削る面の処理
2. 面ごとの経路作成
3. 経路を一本の Toolpath に
4.Toolpath plane の作成

（5.toolpath planes から座標 X,Y,Z、回転角 B,C を算出）
（6.g-code 生成及びデータの受け渡し）
（5、6 は Roughing と共通なので省略）

これら各段階におけるアルゴリズムの詳細を以下に示す。また実際に作成した Grasshopper ファイルの画像を掲載する。

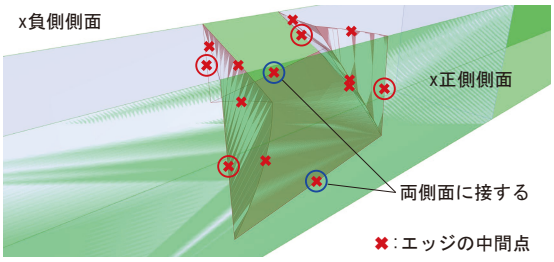
6.2.0 各種入力値の設定

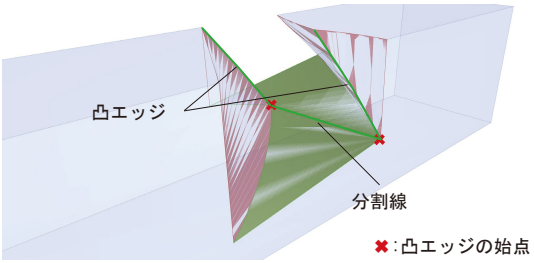
⓪データの受け取り

Roughing の⑳で受け渡しと記したデータを受け取る。

①削る面の分割を行うかどうか判定

削る面が材料の x 軸正側側面と負側側面の両方に接している時、以降の操作で不具合を起こす可能性があるので、各面が材料の両側面に接するか判定を行い、接する場合はその面を分割する。判定方法は各面のエッジの中点をとり、その点が材料側面上に存在するか判定し、正側に存在する点と負側に存在する点を両方持つ面を両側面に接する面とした。





面の分割では削る面同士の接線の中で convex curve（凸エッジ）を取り出し、その曲線の始点（z 座標が小さいほう）間を繋ぐ直線で行った。

・両側面接触判定
　各削る面ごとの接触判定（True or False）
・面分割判定
　分割面の有無判定（True or False）

②各種値の設定

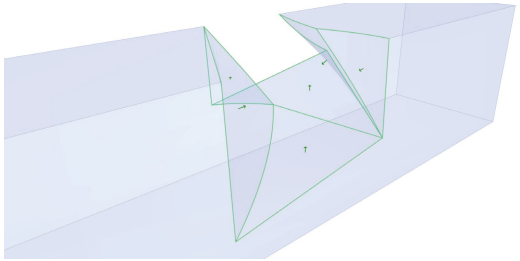
①の面分割判定を用いて、分割前の削る面を使用するか分割後の削る面を使用するか判定。また同様に削る面数も判定を行う。

判定した削る面から以下のパラメータを取り出す。

面の中心点

面の中心点パラメータ

各面中心での法線ベクトル



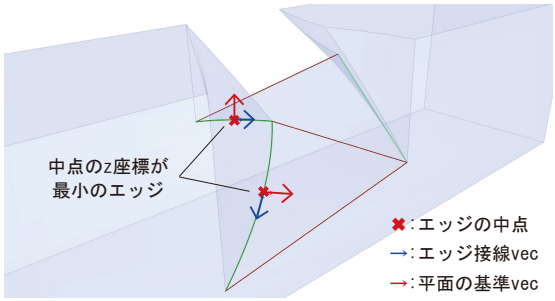
6.2.1　削る面の処理

③削るコンター方向 vector

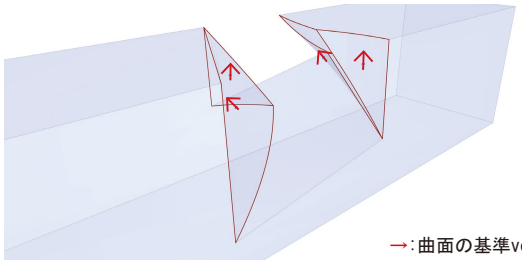
面を削っていく時まず面をコンター化（スライス）し、生成される曲線をつなげて toolpath とする。その時の下準備としてそのコンター方向を定めるベクトル（基準 vector）を各面ごとに求める。

まず各面が平面かどうかの判定を行う。

面が平面の場合、材料側面上にないエッジの内、中点の z 座標が最も小さいエッジの中点での接線方向ベクトルを求め、それを面の法線ベクトルを軸として 90° 回転させたベクトルを基準 vector とする。

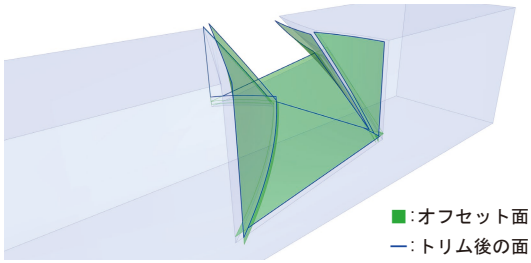


曲面の場合、面の中心点での曲率が最も大きい方向を基準 vector とする。

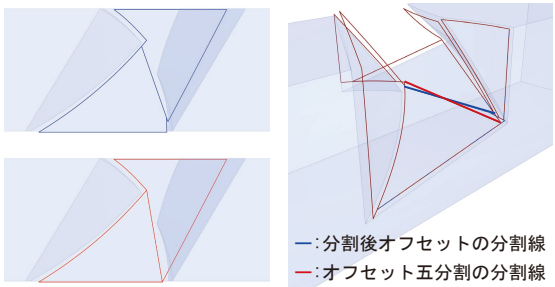


④削る面のオフセット

ドリルの先端形状が半球であることを考慮すると、toolpath は削る面を半球の半径分オフセットした面上に作らなければならない。オフセットした削る面同士は交差するので、はみ出た分はトリムする。



ただし、①で分割をした面はオフセット後頂点で分割されるかわからないので分割前の面をオフセット＆トリムし作られた面を新たに分割する。

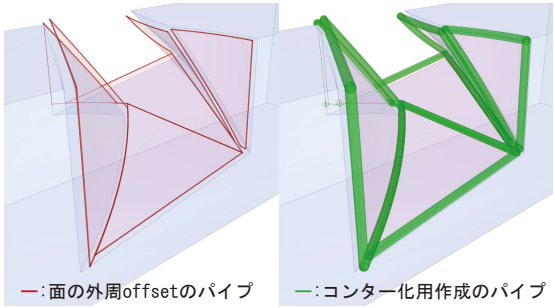


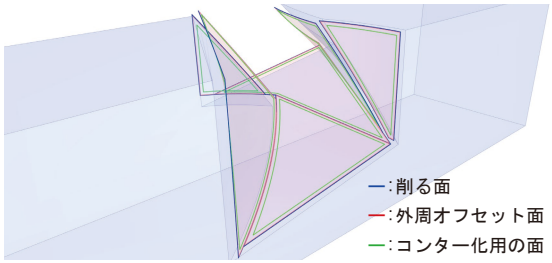
⑤削る面外周の面内オフセット

toolpath は作成していく上で、toolpath が自己交差又は同じ曲線上を数回通るように設定すると誤作動を起こす可能性がある。

それらを防ぐため、面の外周は隣の面と接しているのでほんの少しだけ面内オフセット（右図では面の外周 offset = 0.075 mm）させる必要がある。また面をコンター化してできる曲線は外周と交差するので、コンター化するための面は外周から tool radius 分面内オフセットしたものを使用する。

面内オフセットの仕方としては、各面のエッジで指定したオフセット分の太さのパイプを作成し、面との交差部を取り出す。

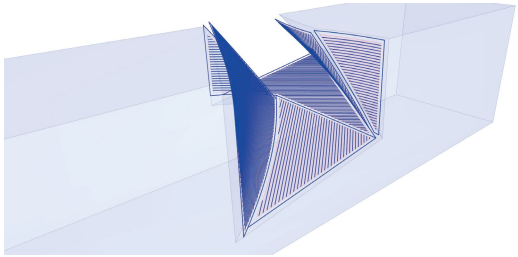




6.2.2　面ごとの経路作成

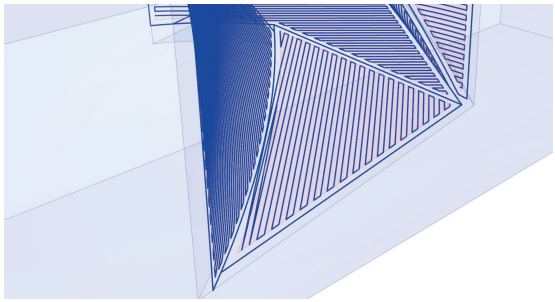
⑥コンター化（スライス）

⑤で作成したコンター化用の削る面に、ドリルの被り（tool radius × 2 の%、右図では 80%）を指定してコンター幅（右図で 0.2 mm）を設定し、③で求めた基準 vector を用いてコンター化を行う。



⑦コンター化の曲線を一本の曲線に結合

⑥で得られた面ごとの曲線群に対して、番号を振り、偶数番目の終点から奇数番目の終点へ、奇数番目の始点から偶数番目の始点へ測地線を引き、曲線軍と結合させる。この操作により、各面の曲線群は一本の toolpath になる。



⑧ toolpath の始点を調整

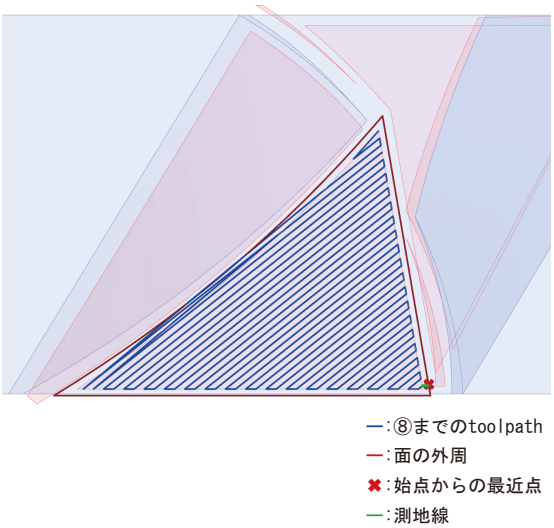
各面の toolpath に対して始点と終点の z 座標を比較し、z 座標が大きいほうを始点として統一する。

⑨各 toolpath と面の外周の結合

⑤で求めた面の外周と⑧までで作成した各面の toolpath を結合することで各面の toolpath が完成する。

⑧までの toolpath の終点から最も近い面の外周上の点を求め、その点で面の外周に対して極小円でのトリムを行う。これにより面の外周は閉じた曲線から開いた曲線になり、その始点と⑧までの toolpath 終点を測地線で結ぶことで一本の toolpath となる。

また、開いた面の外周の始点と終点は後に toolpath planes を指定する時に使用する。



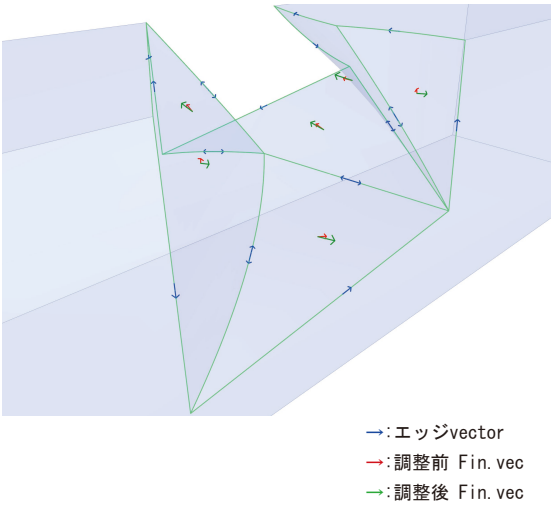
6.2.3　経路を一本の Toolpath に

⑩各面切削後のドリルが抜ける方向 vector

各面の toolpath を結合し一本の toolpath を作成するには、面の toolpath 間の移動でのドリルの抜ける方向（Finishing vec）を定める必要がある。

分割後の各面（オフセット前）のエッジ中点での法線ベクトルを求め、エッジの長さに応じて重みづけをし（全てのエッジの中で最大のものの長さで割ったもの）、各面ごとで法線ベクトルとエッジベクトルを合成したものを Finishing vec とする。

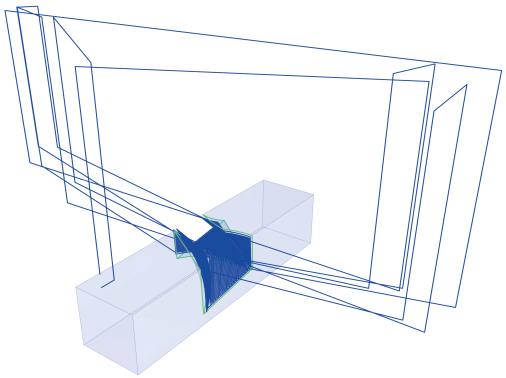
しかしこの Finishing vec でドリルと材料が緩衝する場合があるので、その場合はこの Finishing vec を法線ベクトルを軸に回転させ調整をする。



⑪一本の経路（toolpath）にまとめる

Roughing の⑫と同様に safety plane を設定し、Roughing vec の代わりに Finishing vec を用いて各面の toolpath 間を繋ぐ。

また Roughing の⑫と同様に changing points も求める。



6.2.4 Toolpath plane の作成

⑫ toolpath points とパラメータ表示

g-code は移動の時座標間を直線補完で動くため曲線の toolpath はポリラインに変更する必要がある。より滑らかに切削を行うため、ポリライン化するときの許容角度（今回は 2°）を設定し toolpath の曲線を滑らかにポリラインに置換する。

また Roughing の ⑬ と同様に toolpath points、toolpath points parameter、changing points parameter を求め、加えて⑨で求めた面の外周の始点と終点もパラメータ化する。

⑬ toolpath points でのドリルの向きを設定

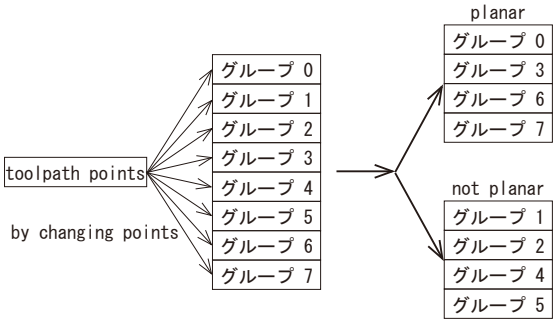
まずは Roughing の ⑭ 同様に changing points parameter によって toolpath points をグループ分けする。

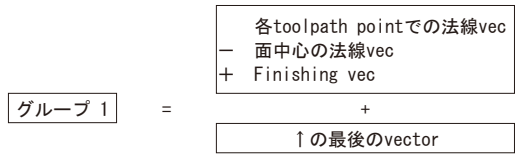
次に各削る面が平面か曲面かによってそれらを場合分けする。

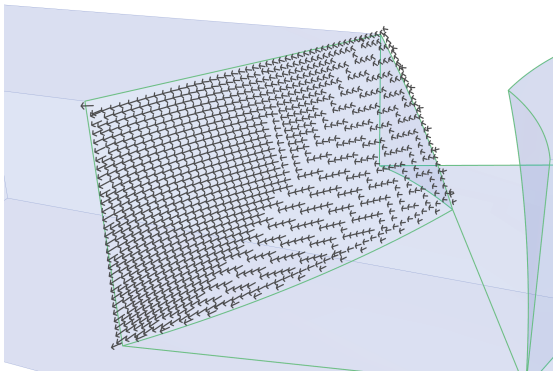
平面の場合、toolpath planes の z 軸となるベクトルはその面に対応する Finishing vec を用いる。

曲面の場合、各 toolpath point での法線ベクトルを求め、そのベクトルに Finishing vec から面の中心法線ベクトルを引いたものを足す。これによって各 toolpath point に対応するベクトルはその点での法線ベクトルに合わせて少しずつ変化させることができ、より滑らかな切削を行うことができる。

また、曲面の面の外周始点終点間の toolpath plane ベクトルは、面の外周始点のひとつ前の toolpath point でのベクトルを用いる。

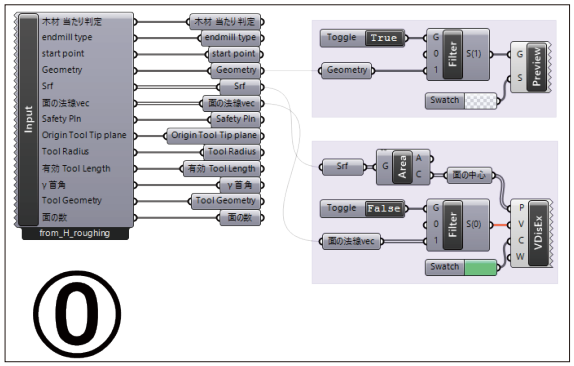


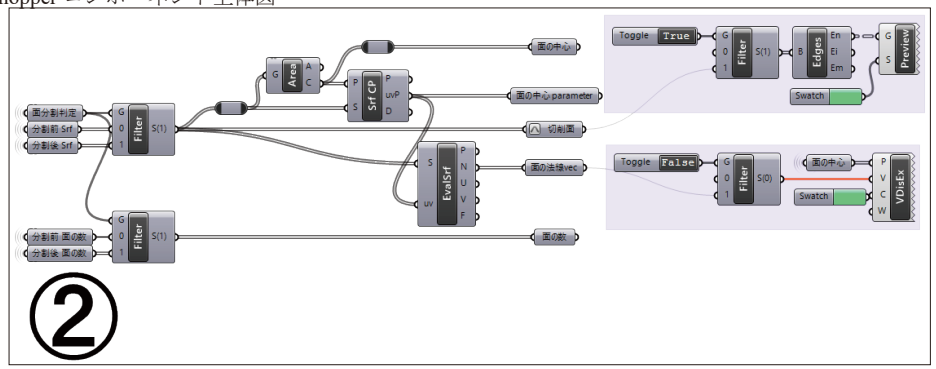


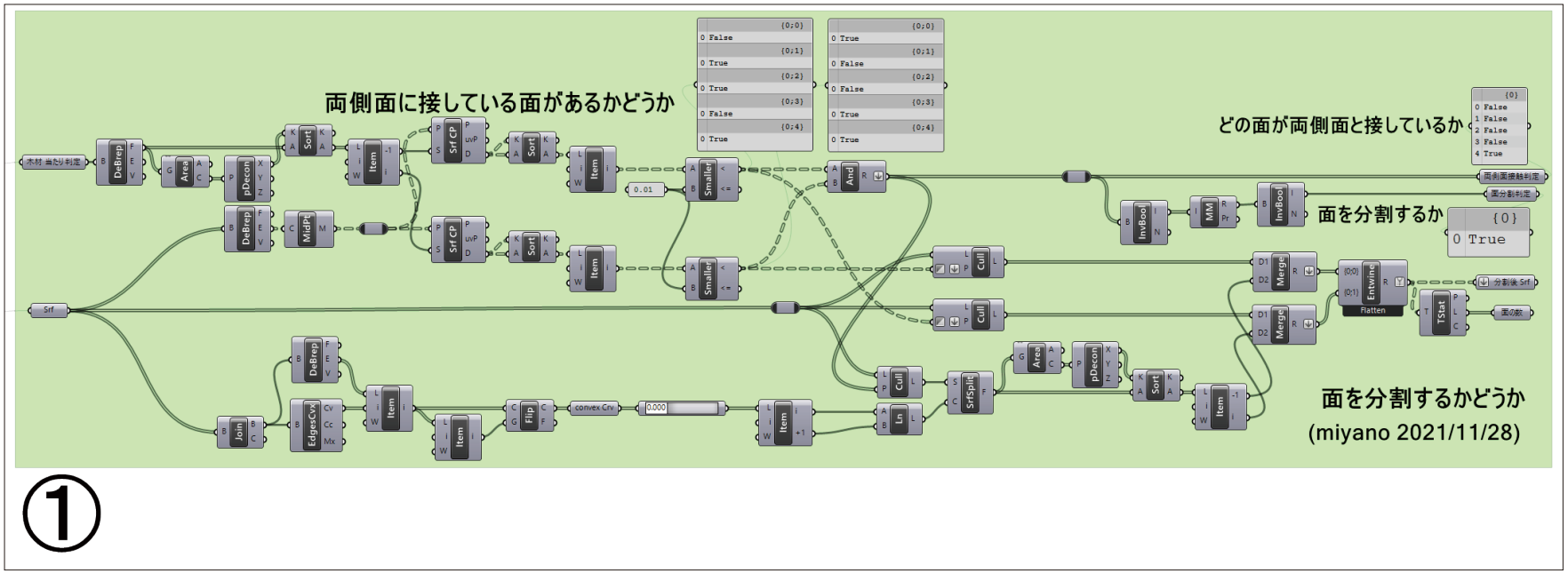


0. 各種入力値の設定

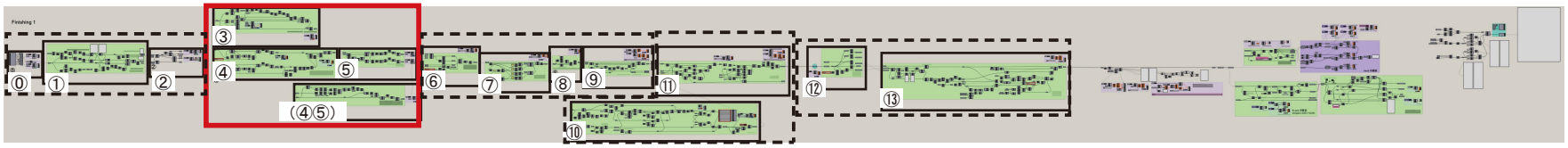
Finishing 1　Grasshopper コンポーネント全体図



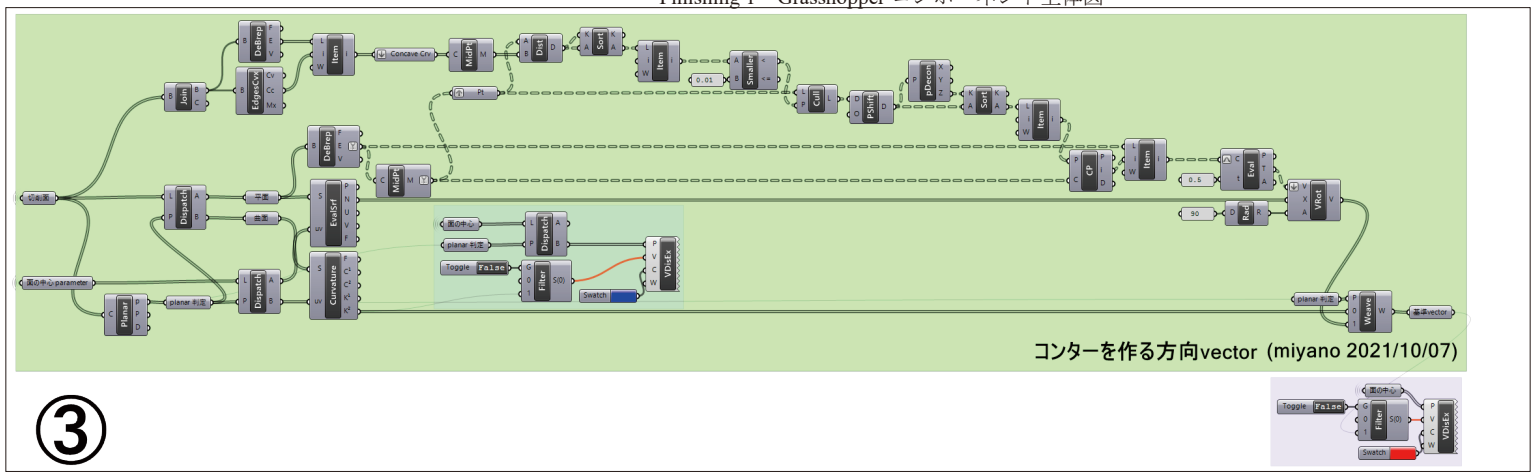


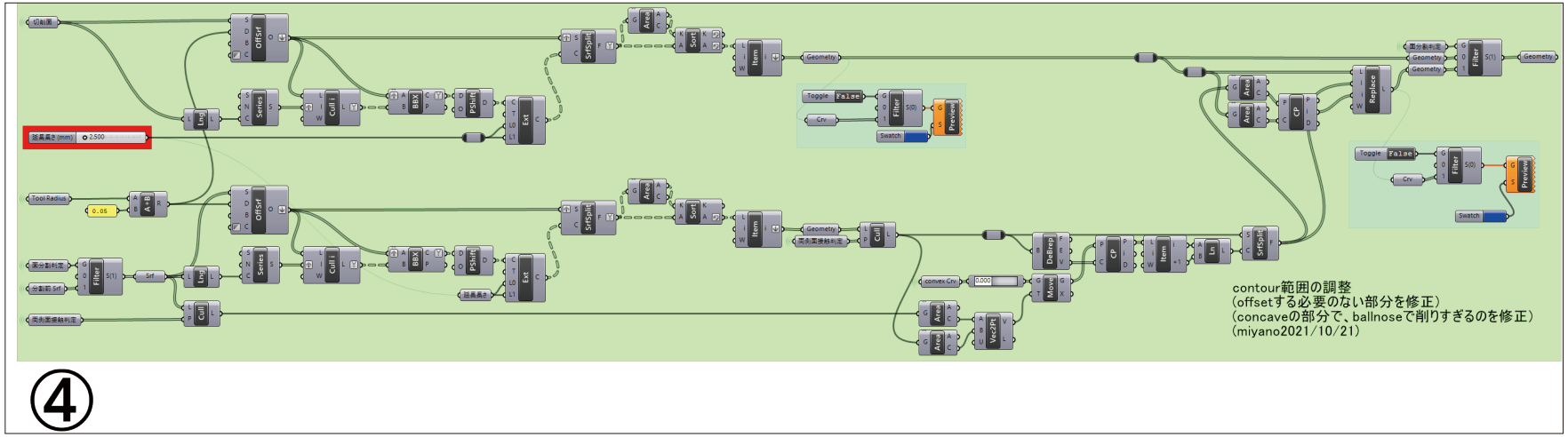


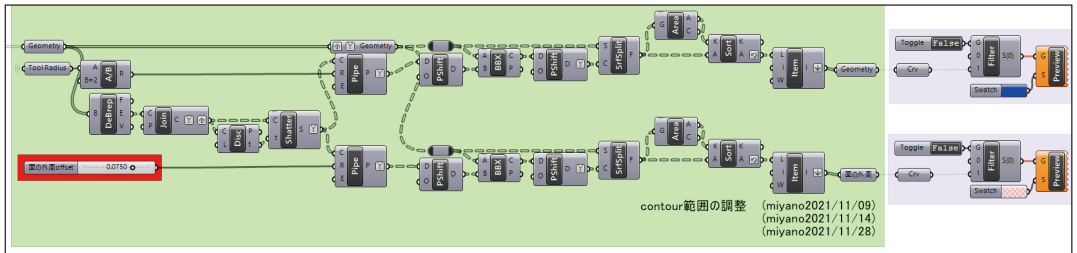
1. 削る面の処理



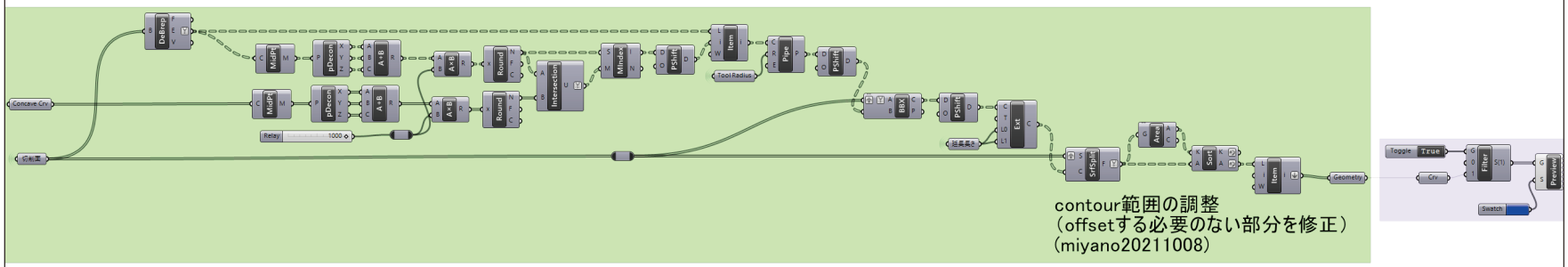
Finishing 1　Grasshopper コンポーネント全体図






contour範囲の調整
(miyano2021/11/09)
(miyano2021/11/14)
(miyano2021/11/28)

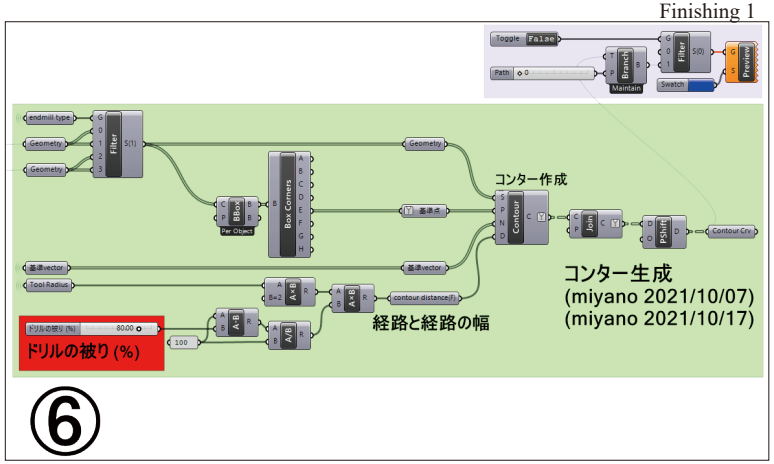
⑤

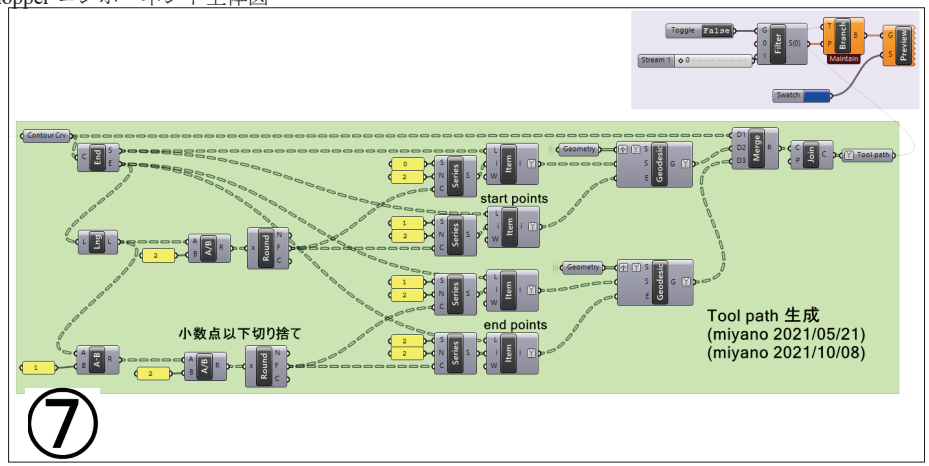

contour範囲の調整
(offsetする必要のない部分を修正)
(miyano20211008)

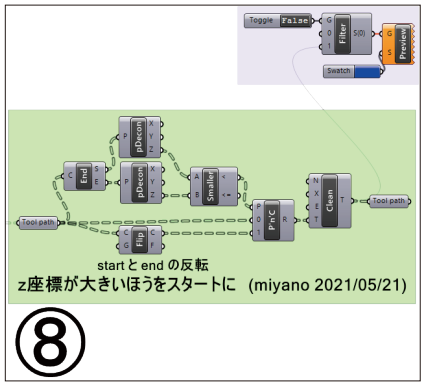
(④⑤)

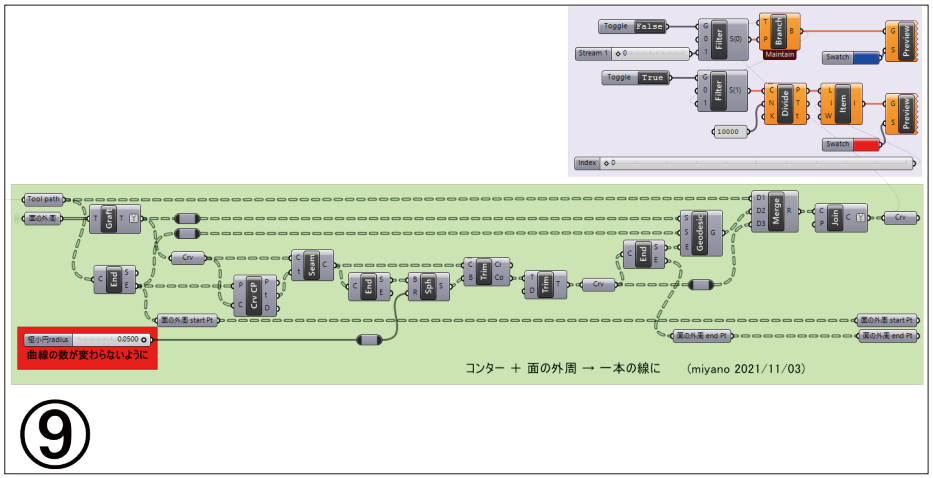
2. 面ごとの経路作成

Finishing 1　Grasshopper コンポーネント全体図

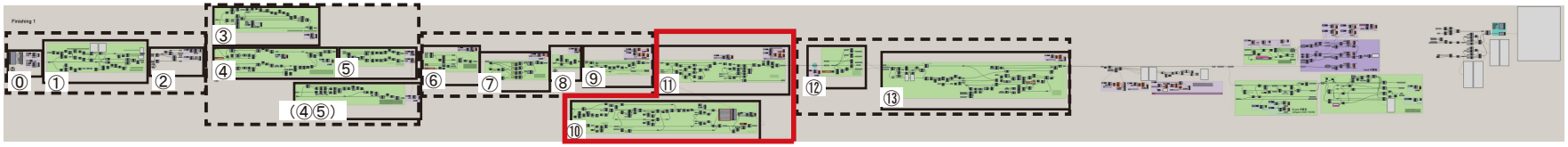




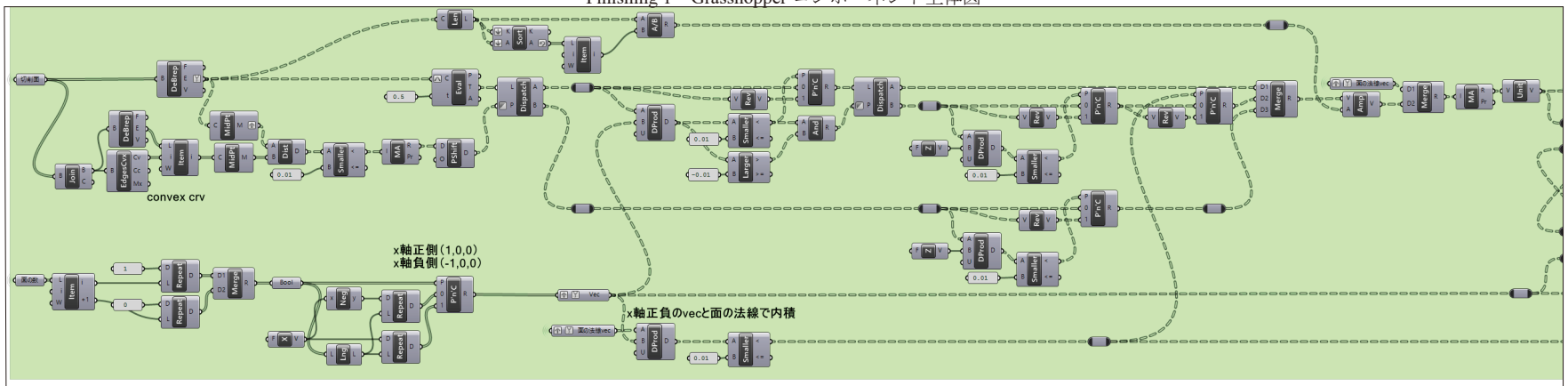


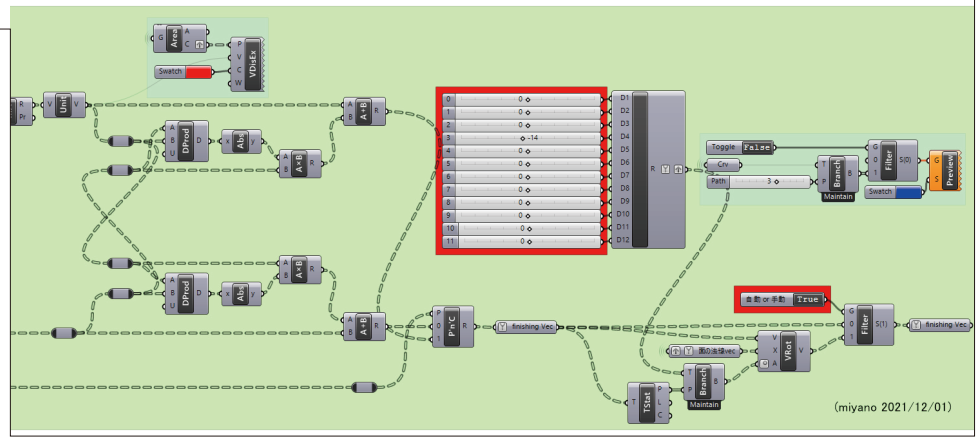


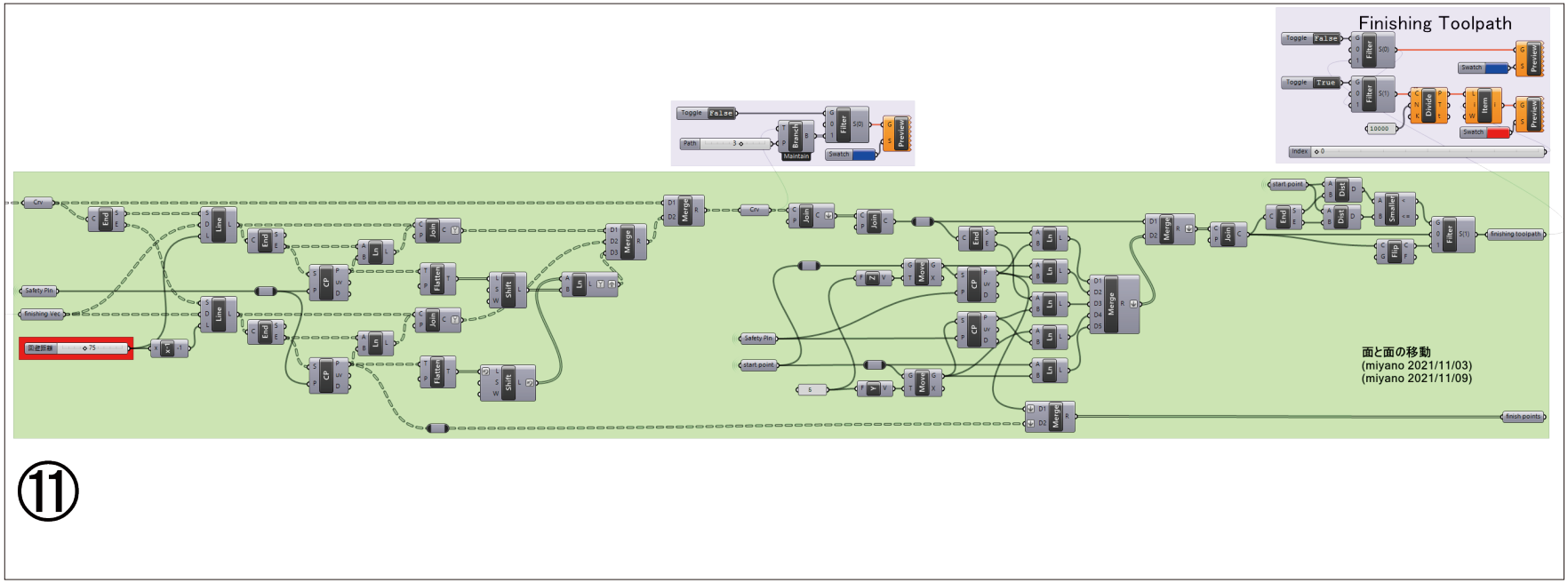
3. 経路を一本の Toolpath に



Finishing 1　Grasshopper コンポーネント全体図

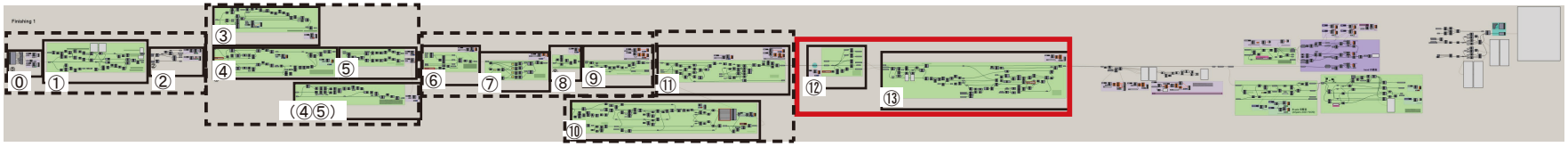




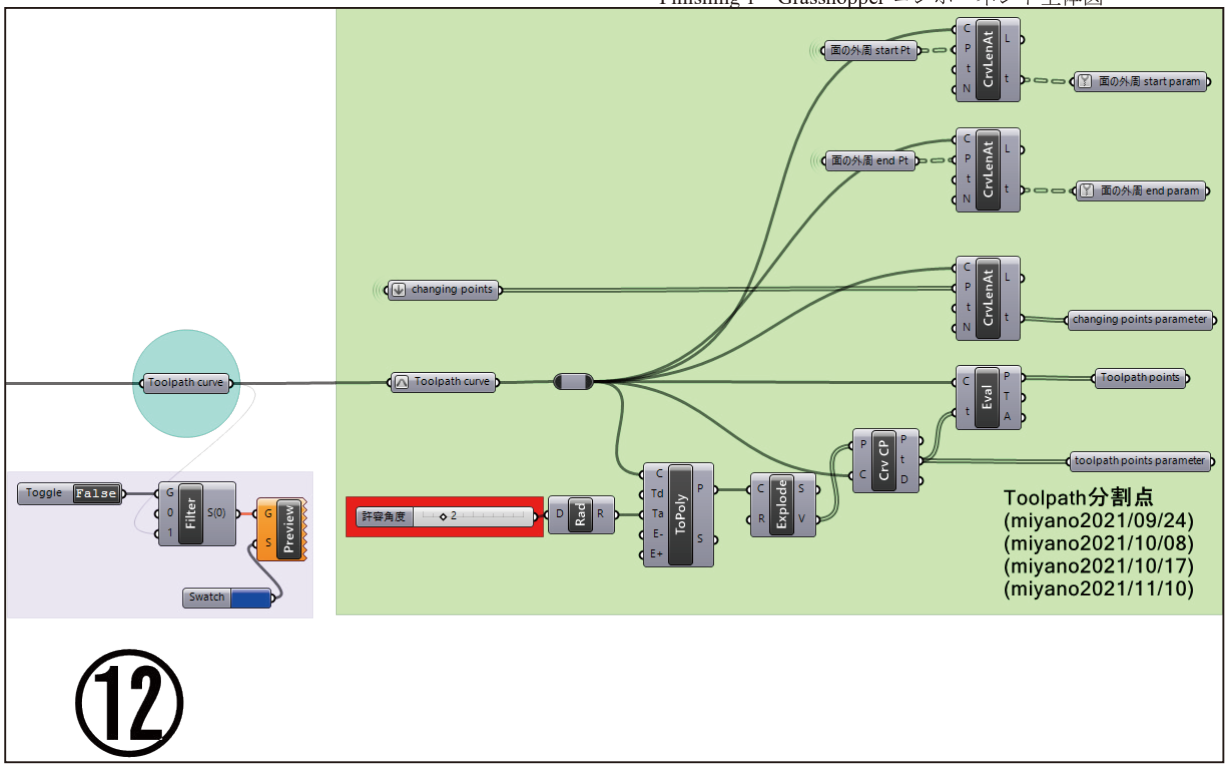

Finishing Toolpath
面と面の移動
(miyano 2021/11/03)
(miyano 2021/11/09)

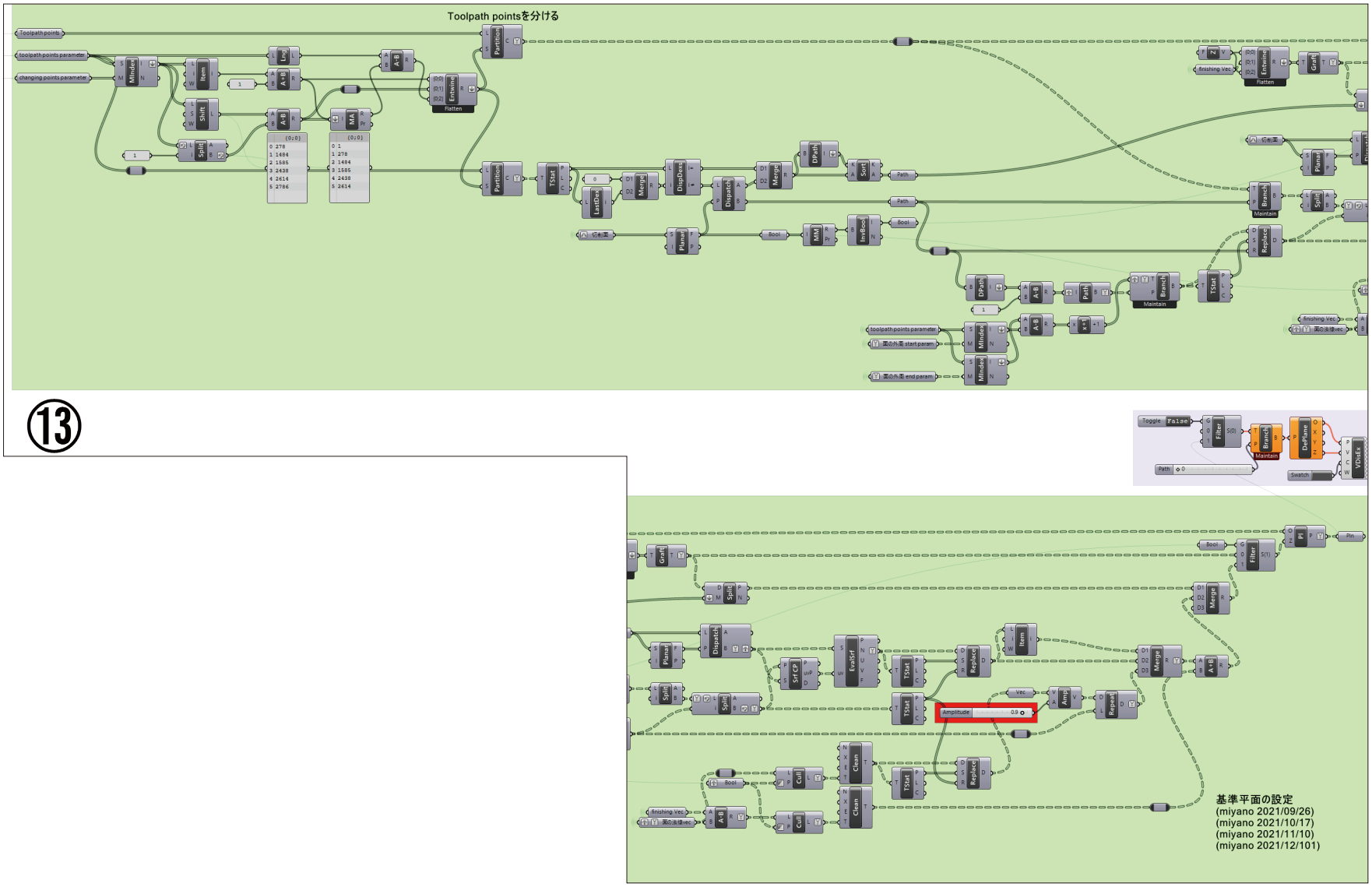
⑪

4. Toolpath plane の作成



Finishing 1　Grasshopper コンポーネント全体図




Toolpath pointsを分ける
Toolpath points
toolpath points parameter
changing points parameter
finishing Vec
Path
Bool
Amplitude
0.9
Vec
基準平面の設定
(miyano 2021/09/26)
(miyano 2021/10/17)
(miyano 2021/11/10)
(miyano 2021/12/101)

⑬

6.3 Finishing 2

Finishing 2 は Finishing 1 の③のみ異なるため、その部分だけアルゴリズムを示す。また Grasshopper ファイルの画像を掲載する。

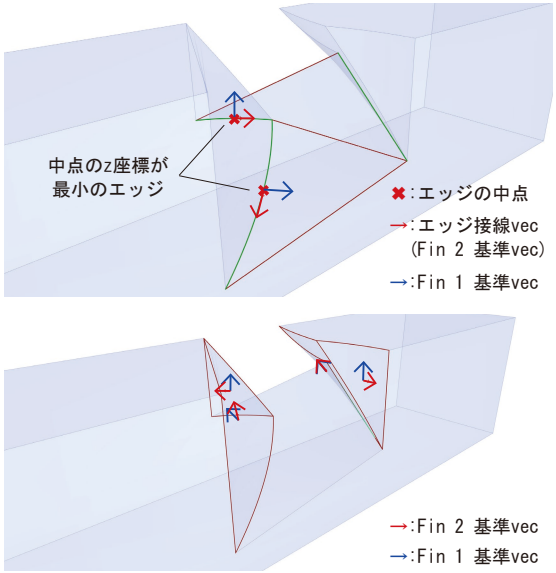
③削るコンター方向 vector

Finishing 1 の③のように基準 vector を求める。より面を滑らかに切削するため、基準 vector の方向が Finishing 1 と直交するような方向が必要になる。

まず各面が平面かどうかの判定を行う。

面が平面の場合、材料側面上にないエッジの内、中点の z 座標が最も小さいエッジの中点での接線方向ベクトルを求め、それを基準 vector とする。

曲面の場合、面の中心点での曲率が最も小さい方向を基準 vector とする。

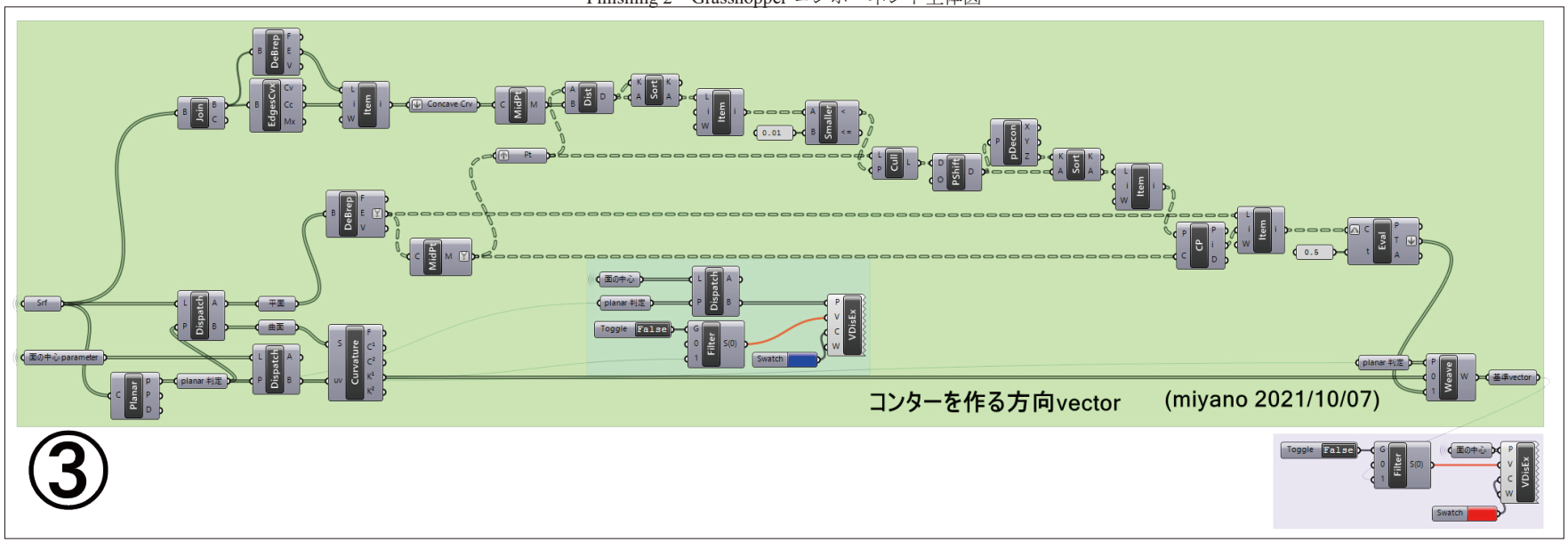


6.4 g-code generator

g-code generator では Roughing、Finishing 1、2 で生成した g-code を順番に結合し、その直前にドリルの回転速度、移動策度などを指定する g-code を入力し一つのデータにする。

実際の g-code の一部と Grasshopper ファイルの画像を掲載する。

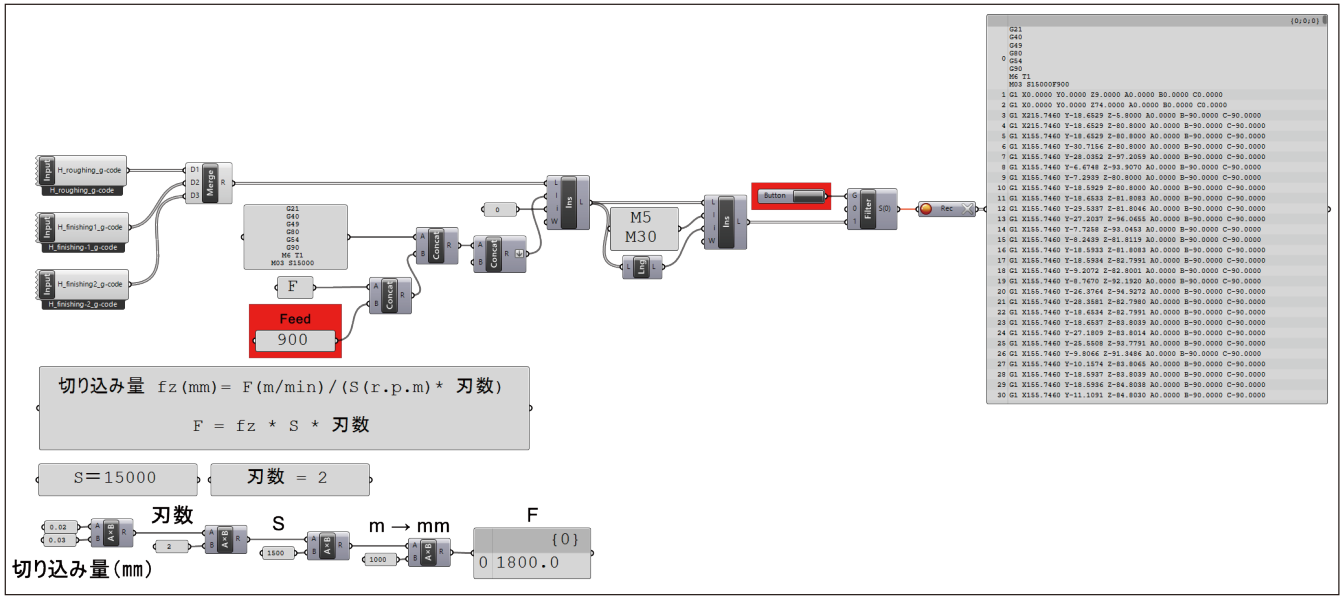
Finishing 2　Grasshopper コンポーネント全体図



g-code generator

```
                                                                    {0;0;0}
    G21
    G40
    G49
  0 G80
    G54
    G90
    M6 T1
    M03 S15000F900
  1 G1 X0.0000 Y0.0000 Z9.0000 A0.0000 B0.0000 C0.0000
  2 G1 X0.0000 Y0.0000 Z74.0000 A0.0000 B0.0000 C0.0000
  3 G1 X215.7460 Y-18.6529 Z-5.8000 A0.0000 B-90.0000 C-90.0000
  4 G1 X215.7460 Y-18.6529 Z-80.8000 A0.0000 B-90.0000 C-90.0000
  5 G1 X155.7460 Y-18.6529 Z-80.8000 A0.0000 B-90.0000 C-90.0000
  6 G1 X155.7460 Y-30.7156 Z-80.8000 A0.0000 B-90.0000 C-90.0000
  7 G1 X155.7460 Y-28.0352 Z-97.2059 A0.0000 B-90.0000 C-90.0000
  8 G1 X155.7460 Y-6.6748 Z-93.9070 A0.0000 B-90.0000 C-90.0000
  9 G1 X155.7460 Y-7.2939 Z-80.8000 A0.0000 B-90.0000 C-90.0000
 10 G1 X155.7460 Y-18.5929 Z-80.8000 A0.0000 B-90.0000 C-90.0000
```

g-code の一部



6.5 Edge finishing

2.4.6 で大まかに示したが、Edge finishing のアルゴリズムの流れを再び以下に示す。

0. 各種入力値の設定
1. 凹エッジを取り出す
2. 経路を一本の Toolpath に
3.Toolpath plane の設定

（4.toolpath planes から座標 X,Y,Z、回転角 B,C を算出）
（5.g-code 生成）
（4、5 は Roughing と共通なので省略）

これら各段階におけるアルゴリズムの詳細を以下に示す。また実際に作成した Grasshopper ファイルの画像を掲載する。

6.5.0 各種入力値の設定

⓪データの受け取り

Roughing から以下のデータを受け取る。
木材当たり判定
start point
geometry
削る面
safety plane

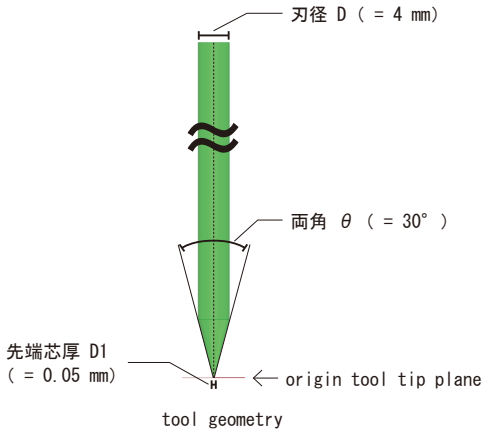
①ドリルの設定

Edge finishing ではドリルの先端が尖った V 溝 / 面取り用エンドミルを用いる。

・origin tool tip plane
　ドリルの先端が尖っているため、先端を基準にドリルを回転させても問題がない。よって今回は原点での XY 平面を設定する。

・刃径 D、先端芯厚 D1、両角 θ
　ドリル形状を決める値。右図参照。

・tool length
　5Axismaker でドリルを下向きで材料原点に接触させた時とドリルを 60° に傾けて接触させた時の Z 座標の差の二倍。

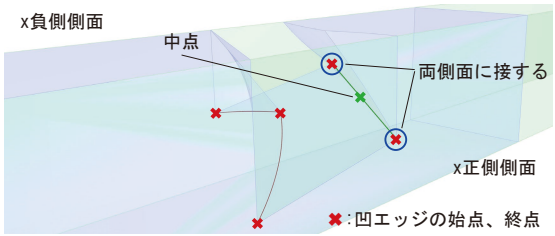


6.5.1　凹エッジを取り出す

②削るエッジの抽出と調整

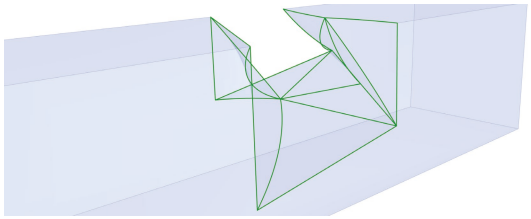
Edge finishing では、削る面同士の接線の中で凹になっているエッジのみを削る。

またできるだけドリルと材料の余計な接触を避けるため、始点と終点両方が材料の側面上に存在するエッジはその中点で半分に分けて扱う。



③面の分割

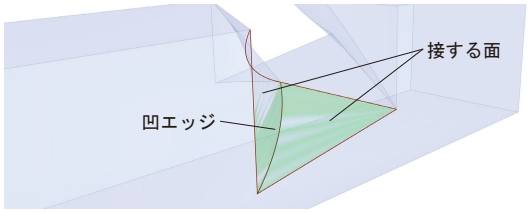
すべての面を三角形（曲面があるので正確には 3 頂点の閉じた曲線）に分割する。



④凹エッジと接する面

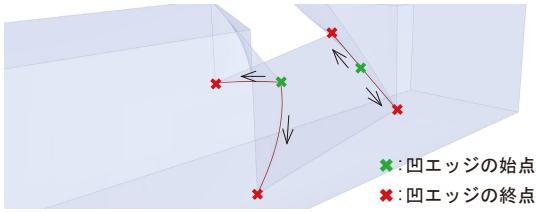
②で求めた凹エッジに接する③で求めた面を求める。

凹エッジの中点と分割面のエッジ中点が一致するかを判定し、接する面を求める。



⑤凹エッジの始点調整

すべての凹エッジが材料の側面から遠い方の点を始点となるようにする。



6.5.2　経路を一本の Toolpath に

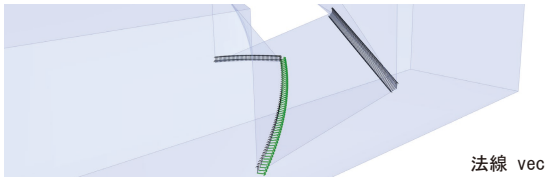
⑥凹エッジの分割、各点でのベクトル

toolpath points となる、凹エッジの分割点を決める。それぞれのエッジを長さが均一になるように分割する（edge 分割数 = 50）。

また、この各分割点でのベクトルを三種類導出する。

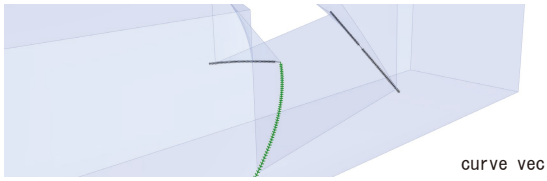
・法線 vec

凹エッジに接する面の法線ベクトルで、分割点ごとに求める。



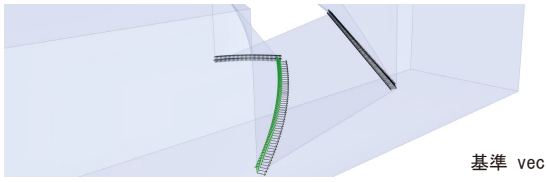
・curve vec

凹エッジ分割点での接線方向ベクトル。



・基準 vec

分割点での curve vec と法線 vec の外積（curve vec × 法線 vec）。



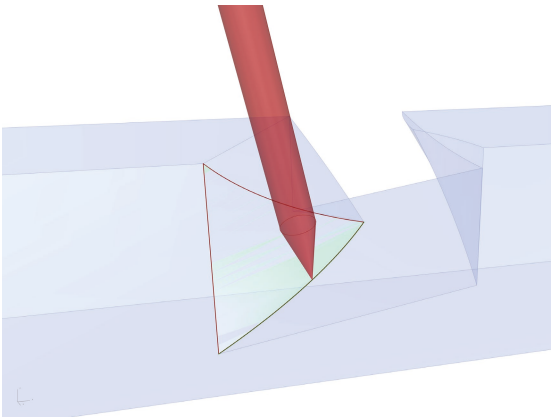
⑦ドリルの角度を決める

すべての分割点でドリルが意図しない場所で材料と緩衝しないように角度を決める。

まず法線 vec を回転軸として基準 vec を回転させる。この時、凹エッジ始点での角度と終点での角度を調整し（手動）、始点と終点の間の分割点での角度は、始点での角度から終点での角度に等間隔で変化するよう設定する。

次に法線 vec を x 軸、角度調整したベクトルを y 軸とする平面を各分割点で作成する。その平面の z 軸ベクトルを回転軸にして、角度調整したベクトルをさらに回転させる（手動で調整）。

こうして得られたベクトルを toolpath vec とする。また凹エッジごとの toolpath vec の最初と最後のベクトルをそれぞれ start vec、end vec とする。

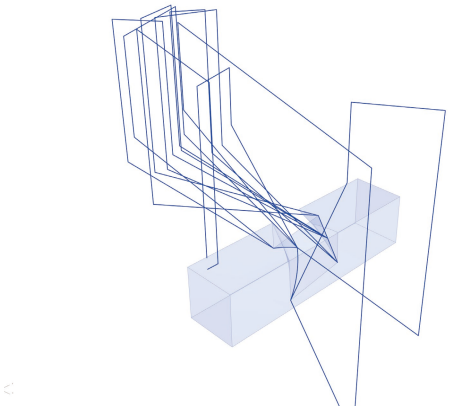


⑧一本の経路（toolpath）にまとめる

Roughing ⑫での Roughing vec、Finishing 1 ⑪での Finishing vec のかわりに⑦で生成した start vec、end vec を用いて凹エッジごとの toolpath を作る。

その後 Roughing や Finishing 1 と同様に safety plane を用いて凹エッジ toolpath 間の動きを設定し、一本の toolpath を生成する。

また changing points も同様に設定する。



6.5.3　Toolpath plane の設定

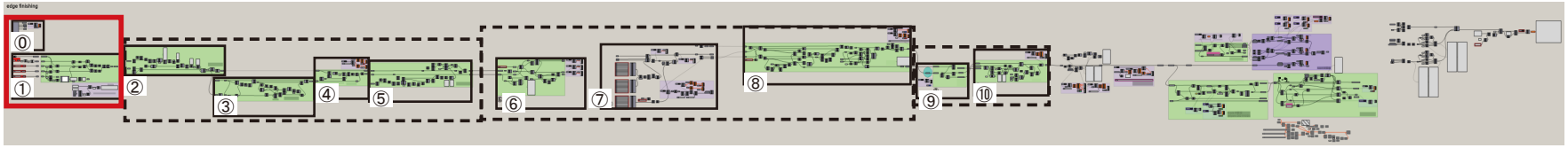
⑨ toolpath points とパラメータ表示

Roughing の⑬と同様に toolpath points、toolpath points parameter、changing points parameter を作成する。

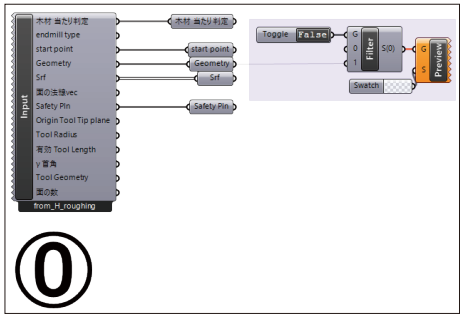
⑩ toolpath points でのドリルの向きを設定

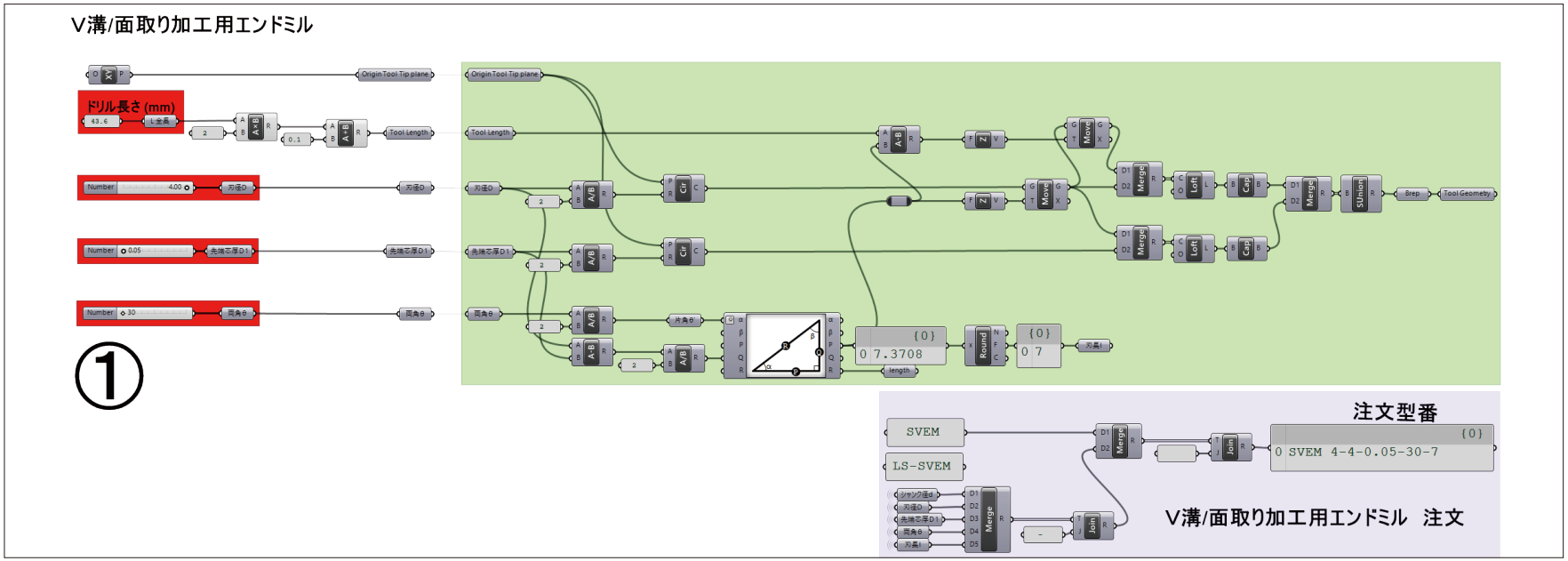
Roughing の⑭と同じ操作を行う。ただし、Roufhing vec の部分は toolpth vec になる。

0. 各種入力値の設定

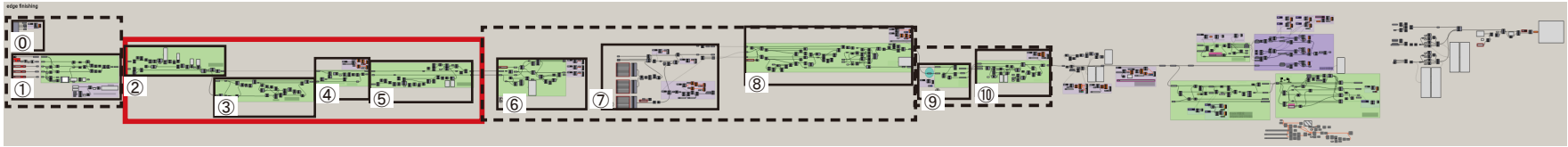


Edge finishing　Grasshopper コンポーネント全体図

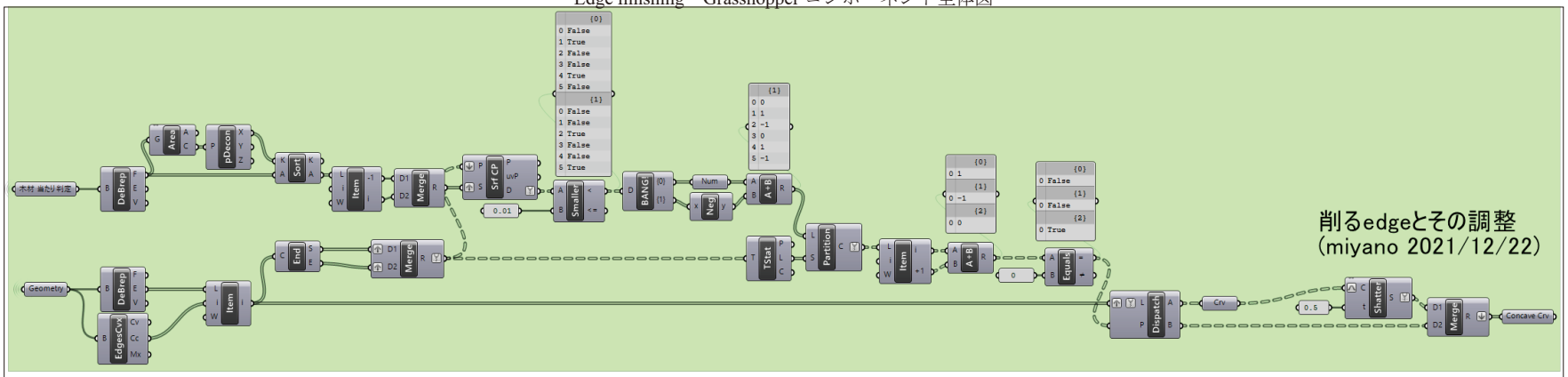




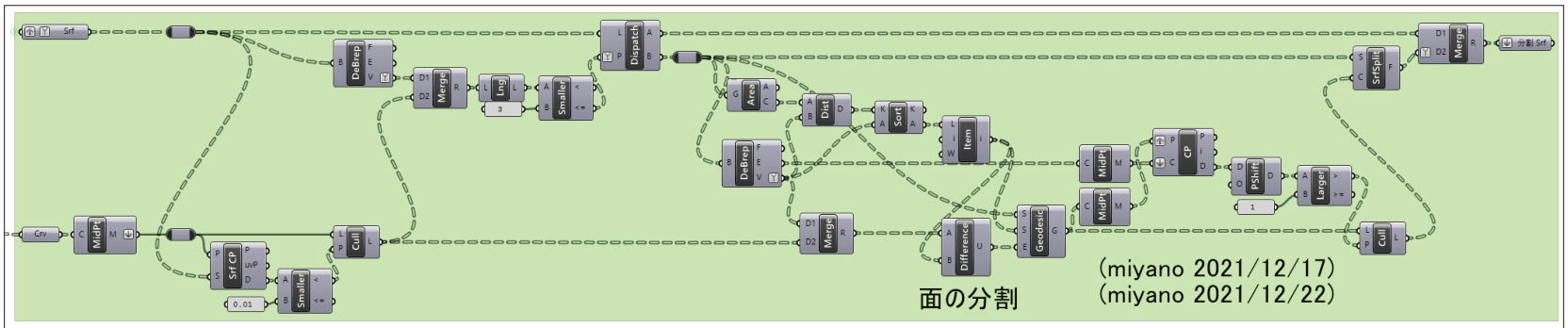
1. 凹エッジを取り出す



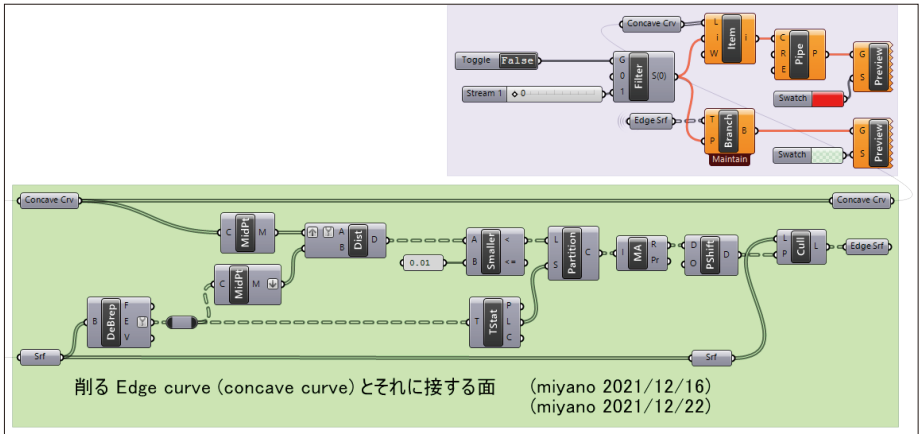
Edge finishing　Grasshopper コンポーネント全体図



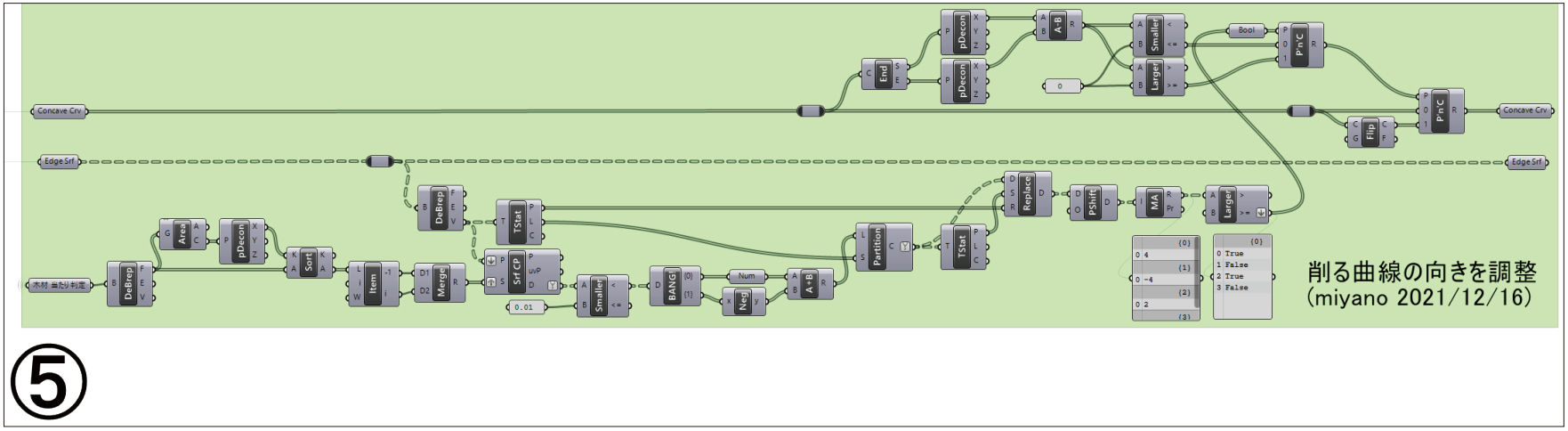
②



③

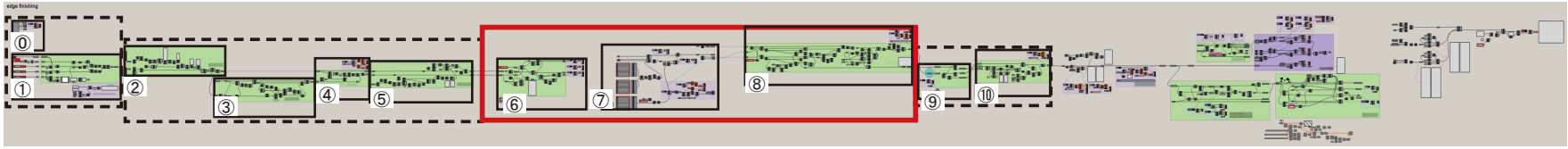

Concave Crv
Toggle
False
Stream 1
Filter
Item
Pipe
Preview
Swatch
Edge Srf
Branch
Maintain
MidPt
Dist
0.01
Smaller
Partition
MA
PShift
Cull
DeBrep
TStat
Srf
削る Edge curve (concave curve) とそれに接する面
(miyano 2021/12/16)
(miyano 2021/12/22)

④

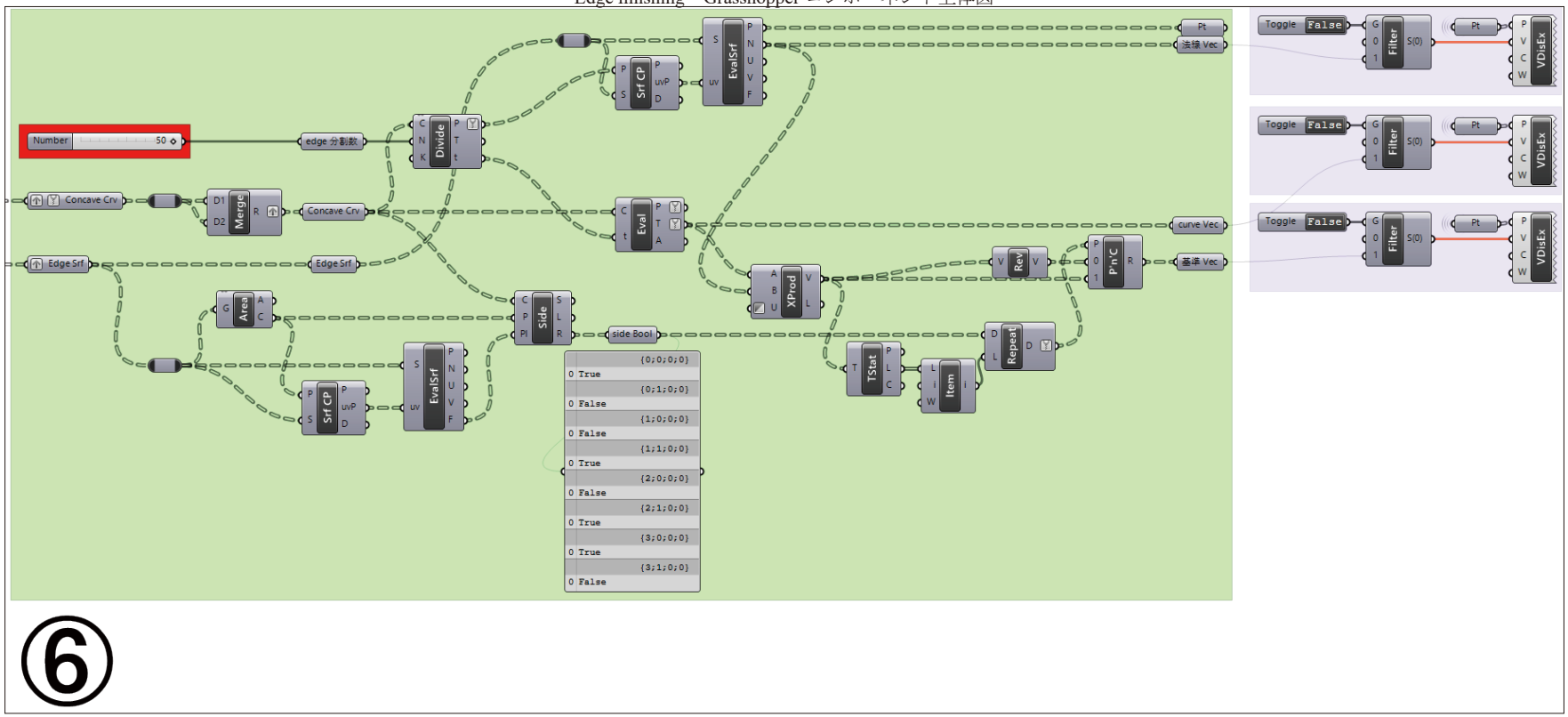

Concave Crv
Edge Srf
End
pDecon
A+B
Smaller
Larger
Bool
P+C
Flip
DeBrep
Area
Sort
Item
Merge
TStat
Srf CP
0.01
BANG
Num
Neg
Partition
Replace
PShift
MA
True
False
True
False
削る曲線の向きを調整
(miyano 2021/12/16)

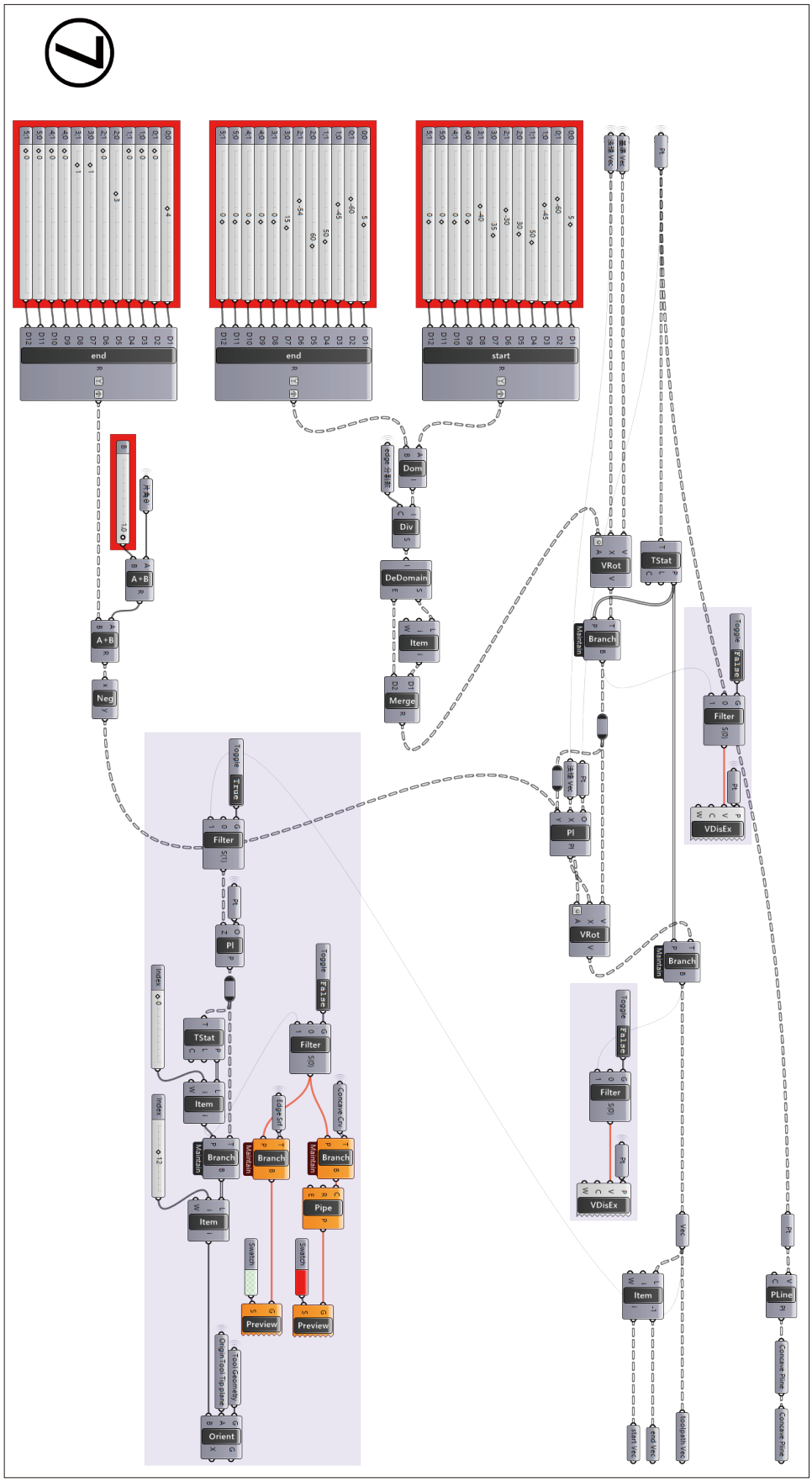
⑤

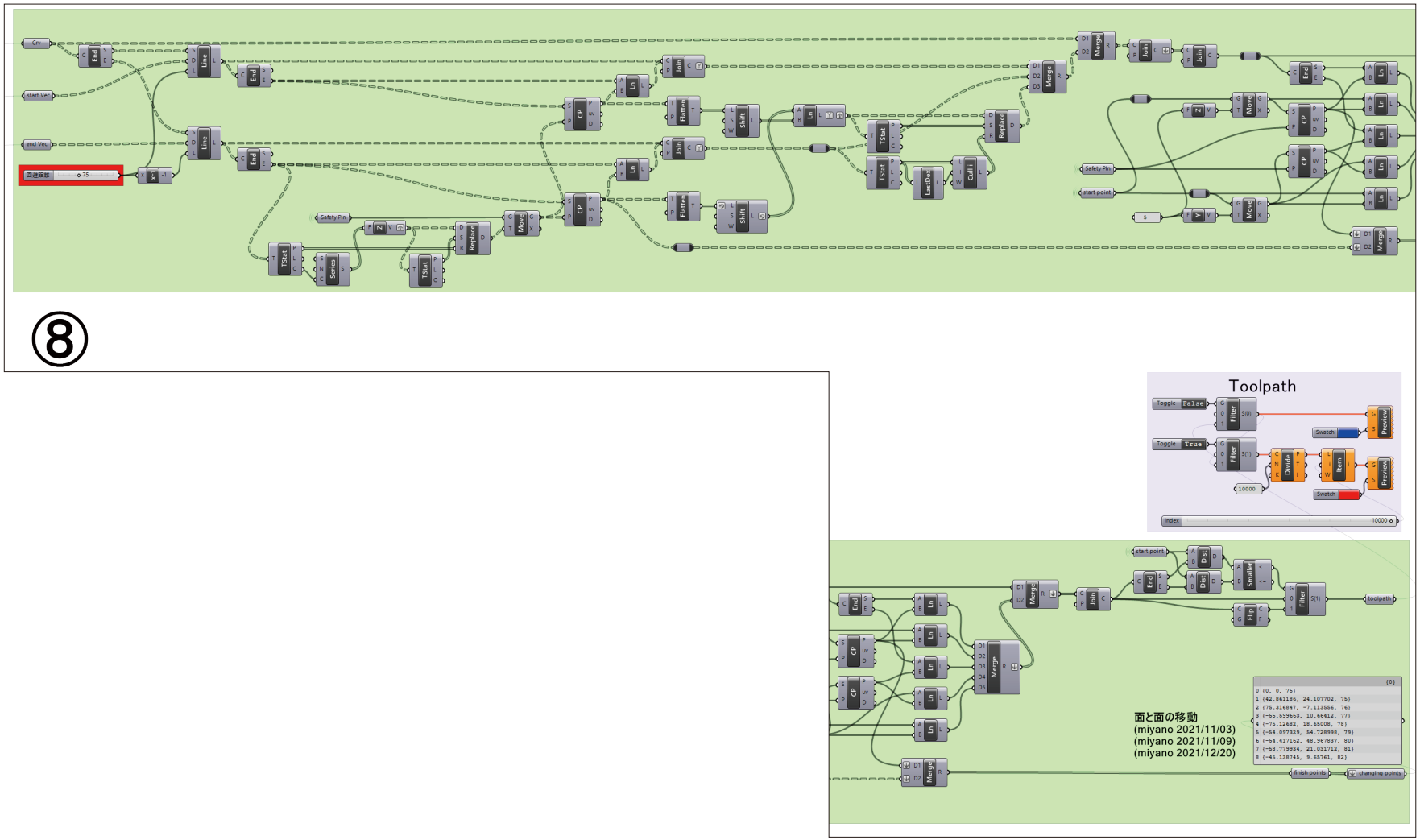
2. 経路を一本の Toolpath に



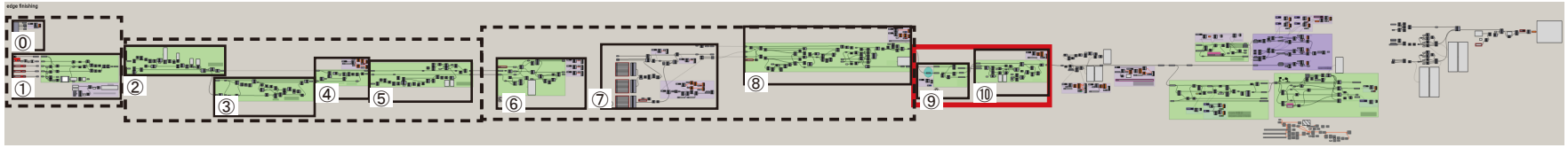
Edge finishing　Grasshopper コンポーネント全体図




start
end
end
Dom
Div
DeDomain
Item
Merge
A+B
A+B
Neg
VRot
Branch
TStat
Toggle
Filter
VDisEx
PI
VRot
Branch
Toggle
Filter
VDisEx
Item
start Vec
end Vec
toolpath Vec
PLine
Concave Pline
Toggle
Filter
PI
TStat
Item
Branch
Item
Toggle
Filter
Branch
Pipe
Switch
Preview
Switch
Preview
Orient


⑧
Toolpath
面と面の移動
(miyano 2021/11/03)
(miyano 2021/11/09)
(miyano 2021/12/20)

3. Toolpath plane の設定



Edge finishing　Grasshopper コンポーネント全体図