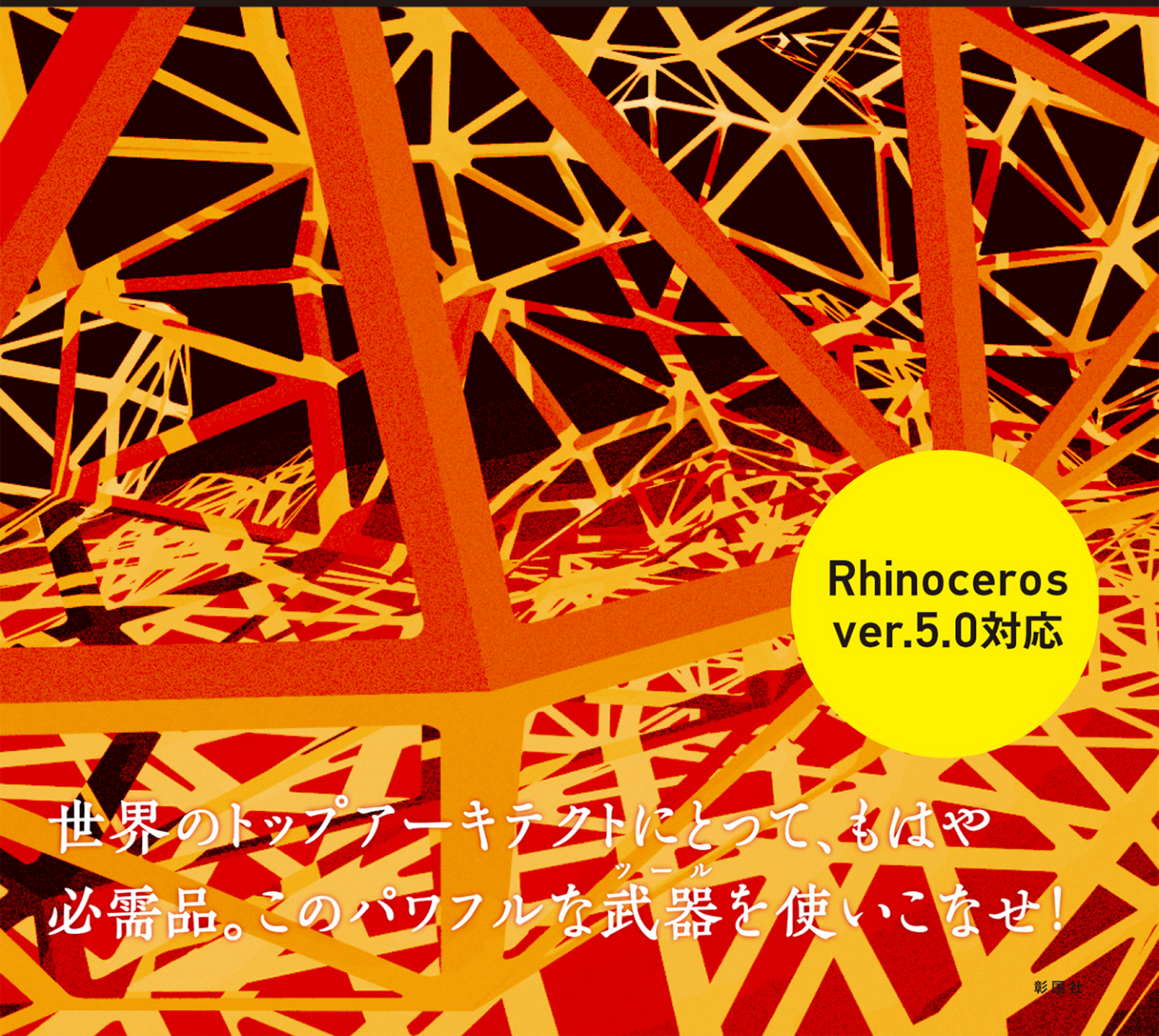


4

RHINOCEROS GRASSHOPPER

Rhinoceros+
Grasshopper
建築デザイン
実践ハンドブック
第2版 ノイズ・アーキテクト 編著



Rhinoceros
ver.5.0対応

世界のトップアーキテクトにとって、もはや
必需品。このパワフルな^{ツール}武器を使いこなせ!



Rhinoceros+
Grasshopper
建築デザイン
実践ハンドブック
第2版 ノイズ・アーキテクト 編著

はじめに

ノイズ・アーキテクト

コンピューターショナル・デザイン (Computational Design) や、デジタル・ファブリケーション (Digital Fabrication) という言葉が建築でも使われ始めている。こうした分野は全くの黎明期でもあり、言葉の定義自体がまだ明確ではない部分も多い。建築におけるコンピューターションは多岐多様な方向性、可能性を持っていて、今世界中で広く研究、実践されているとてもホットな分野なのだが、なぜか日本では意外なほどに認識されていない。

本書では、コンピューターショナル・デザインの汎用ソフトウェアとして、とくに欧米の建築界で普及しつつある Rhinoceros (以下ライノ) と、その公式プラグインである Grasshopper (以下GH) を、建築の実務で使うことを念頭に必要なトピック、知識をハンドブックとしてまとめた。一般的なソフトウェアの解説書では広範な基礎的内容が辞書的に網羅され、ページを追って順に読み進めるのが難しいのに対し、実務でよく使う知識や技術には偏りがあり、かつその分そこだけ深かったりもする。この本では、我々ノイズ・アーキテクトが建築の実務でライノとGHを使ってきた経験の中で便利かつ有効だと感じた技術、知識をまとめるという、あくまで「勝手な」視点を貫かせていただいた。もちろん一般に重要な設定や手法は客観的にひと通り押さえているが、経験上重要ではない部分は大胆に省いて、逆に使える機能はかなり応用的なところまでピンポイントに掘り下げる構成、アンバランスさを重視した。触れたかった面白い機能や発展的可能性、理論的な部分はもちろんまだまだたくさんあるが、そうした可能性への理解は、いくつかの具体的内容で一定のレベルを突き抜ければ必ずから芽づる式に広がっていくものだと理解している。

最近は大学等でライノやGH、それらを応用したデザインを教える機会もかなり増えてきた。今後もその動きは広がると思われることから、本書は大学の基礎レベルの授業で教科書として使えることを念頭に、本書を構成した。同時に、実務として興味のある個人でも自習に使えるような、もしくは授業や仕事の合間に気軽にひろい読みしても理解が蓄積していくような気軽な体裁も心がけた。初版の刊行から2年半を経て、ありがたいことに持続的に売り上げも伸びている。この間にライノが ver.5.0、GH も ver.1.0 となり、機能やインターフェイスも細かいところで大幅に変わったこともあり、この度第2版をまとめた。ライノが市販の NURBS 3D モデラーであり、GH がライノ上で動く無料プラグインという基本的な構成は変わっていない。変更部分は一見目立たないかもしれないが、細かなコマンドや表記の変化1つひとつに関して、今回全面的に新しいバージョンに対応させた。新しい機能にもすでに実務で不可欠となっている興味深いものも多いから、一度慣れた人でもぜひ最新版を繰ぎらえて、新たな発見のきっかけにしてほしい。

2014年3月

本書中の表記について

本書では、Rhinoceros[®]英語版を基準に解説していますが、各種コマンドや一部の用語については日本語版で使われている表記に準拠しています。

Rhinocerosは日本語版として設定することも可能ですが、Grasshopper[®]は現時点で英語版のみです。また、後にも触れるように世界中のユーザーが参加するコミュニティサイトなどを有効に使うことがより発展的に使いこなす鍵になるので、ノイズ・アーキテクトとしては最初多少無理をしても、英語版としてインストールすることを推奨しています。

●RhinocerosとGrasshopperは、Robert McNeel & Associatesの登録商標です。

●RhinocerosはWindows、Macintoshともに対応していますが、本書では、Windowsの画面を解説に使用しています。

●本書の解説は、Rhinoceros ver.5.0、Grasshopper ver.1.0をもとにしています。2014年3月に得られた情報に基づいて編集しているため、本書に登場するアプリケーションとプラグインのバージョン、URL、製品スペックなどの情報は、今後変更される可能性があることを、予めご了承ください。

●Rhinocerosや各種プラグインの動作に必要な環境は、今後更新される可能性があるため、ダウンロード、インストールする際に確認してください。

目次

はじめに ノイズ・アーキテック	3
すぐわかるコンピューテーション年表	6
よくわかるアプリケーション相関図	8

Chap.1

Rhinoceros	10
ユーザーインターフェイスを知る	12
モデリングのキホン	
1 プロパティ/レイヤの設定	15
2 ビューポートの設定	16
3 オブジェクトの表示モードの設定	17
4 ファイル形式の設定	18
5 エイリアスの設定	19
6 オブジェクトを選ぶ	20
7 ビューを動かす	21
8 スナップさせる	22
9 モデリング補助機能を使う	24
モデリングのジッセン	
1 コマンドを入力する	29
2 線を引く	30
番外編 快適さが格段に上がるマウスの設定	33
3 プレゼンシートを作る	34
4 立体を作る	38
5 作ったものをいじる	42
6 複数のものを関係づける	46
7 3Dオブジェクトを2Dにする	48
8 モデルを整理する	52
9 作ったものを評価する	56
10 作ったものを出力する	60

Column

時間短縮テクニックを身につけて早く帰ろう	51
UV座標の考え方を理解する	72
さらに深い世界へ 動物王国へようこそ	92

Project

1 【loft】で椅子の断面をつなぐ	41
2 【make2d】【unrollsrf】で組立図と展開図を作る	49
3 【emap】で天井の曲面モデルを確認する	59
4 ネジレンガで柱をつくる	101
5 膨大な点群をラベリングする	103

Chap.2

Grasshopper	68
大図解! これがグラスホッパーだ!	70
グラスホッピングのキホン	
1 グラスホッピングしてみよう!	74
2 コンポーネントのしくみを知る	76
3 データの型を知る	78
4 キャンバスを整理する	79
5 ベクトルの考え方を知る	80
6 データの取り扱い方を知る	82
グラスホッピングのジッセン	
1 アトラクターを使う	94
2 サーフェスから垂直に線を出す	96
3 決まったデータだけ選び出す	98
4 大量のオブジェクトを管理する	102
5 数値の振れ幅をコントロールする	104
6 スクリプティングに挑戦する	106
練習問題	95
練習問題のこたえ	129
グラスホッピングのソザイ	
よく使うコンポーネント	111
よく使うコンポーネント・セット	126

Essay

多焦点であること	26
どんな変化が起こりつつあるのか	64
技術と職業	90
ゆるさの理由	108

索引	130
革命とかではなくて あとがきにかえて	133
略歴・写真クレジット	136

いよいよラインを使って3Dモデルを作成していく。「モデリングのジッセン」では、頻出するコマンドをカテゴリに分けて紹介するほか、実務で役に立つ情報や設定を勝手に紹介するコラムも加え、3Dモデリングを解説していく。ラインのコマンドは膨大にあるため、ここですべてを紹介しきれないが、使ったことのないコマンドや機能は、ヘルプ機能(P.12)で簡単に調べることができる。

Modeling Practices

モデリングの ジッセン

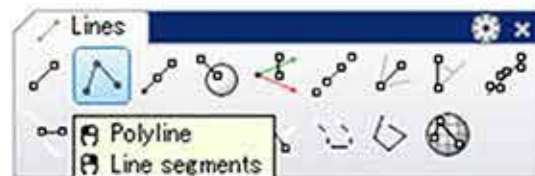
1 コマンドを入力する

ラインで3Dモデルを実際に行うには、対応するコマンドを入力していく必要がある。そこで、まずはコマンドの入力方法を紹介しよう。実際にラインのコマンドを使って作図するには、下記の4通りの方法があり、そのときの状況に応じて使い分けるようにする。

方法1. ツールボタンをクリック

関連するコマンドを発見しやすい

ツールボタンのアイコンは、どれもとてもわかりやすいデザインなので、感覚的にコマンドを選ぶことができる。また、関連するコマンドがグループになっているので、使用したことのないコマンドを発見しやすいというメリットも。



ツールボタンから [polyline] を選択 (左クリック)。アイコンの上にカーソルをのせると、左クリックと右クリックとでオプションが異なることが確認できる。

方法2. コマンドエリアに入力

コマンドのつづりを十分知っている人向け

コマンドのつづりを覚えていれば、直接コマンドエリアにタイプできる。コマンドを途中まで入力すれば候補を予測し表示される。コマンド入力後、Enterキーを押すか右クリックすると実行される。頻繁に使うコマンドは自然と限られてくるので、慣れてきたらこの方法が一番楽。

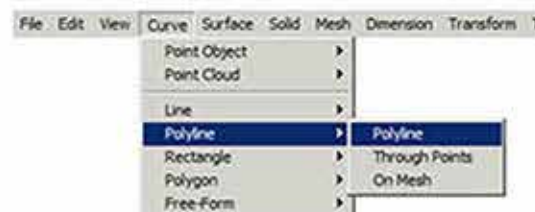


コマンドエリアにコマンド [polyline] を入力。後の何文字か入力すると、コマンドの候補が予測される。

方法3. メニューバーから選ぶ

コマンドのアイコンもつづりもわからないときに

コマンドのつづりもアイコンも知らないけれどもこういう操作をしたい、なんていうときには便利。メニューバーからやりたい操作を絞り込んでいけば、目的の機能に辿り着く。頭の片隅に置いておけば、いざというときに役に立つかも。



メニューバーからコマンド [polyline] を選択

方法4. コマンドを繰り返す

同じコマンドを複数回繰り返すときに

たとえば線を連続でたくさん引く際、上記3つのどの方法であっても、同じコマンドを何度も入力しなければならず面倒だ。そんなとき、マウスの右ボタンをクリックするか、キーボードのスペースキーかEnterキーを押すと、直前に使用したコマンドを繰り返し入力できる。



マウスやキーボードを使ってコマンドを繰り返す

全方法共通. サブコマンドからオプションを選ぶ

コマンドごとの細かいオプションをコマンドエリアから入力

上記いずれの方法でコマンドを開始しても、その後の細かい設定はコマンドエリアの中で行う。コマンドを進めていくと、各段階でオプションを示すサブコマンドが表示され

る。対応したサブコマンドをクリックするか、下線の引いてあるアルファベットをタイプすることで、サブコマンドを変更できる。

Next point of polyline. Press Enter when done (PersistentClose=No Close Mode=Line Helpers=No Length Undo):

上記の赤い地の部分がサブコマンド。クリックするか、下部のアルファベットを入力してオプションを変更

2

線を引く

描画精度が高く、製図用のコマンドも多いラインは、2Dの製図アプリケーションとしても十分使うことができる。実際に、ラインはプロダクトデザインや機械工学などの分野でも、各種設計図を描くために用いられることが多い。当然、建築の分野でも、ラインで図面を引いて、そのままパース用のモデリングを始めるなんてこともできる。3D的なモデリングを始める前に、まずは直線や曲線を自由自在に描くことをマスターしておく、ラインを使う用途がぐっと広がることになるだろう。

[line]
直線を引く
おすすめのエイリアス: l

通常は始点と終点を結ぶ線を描くコマンドだが、サブコマンドの中からその他の線の描き方を選ぶことができる。サブコマンドには、「Normal (法線)」「Perpendicular (直交)」「Tangent (接点)」などがある。

サブコマンド例

- ① コマンドのアイコン
- ② コマンド ([]内に表記)
- ③ おすすめエイリアス (P.19)
- ④ コマンド解説
- ⑤ 各段階の画面
- ⑥ 各段階の解説
- ⑦ コマンドエリア内の表示
- ⑧ サブコマンド (P.29)

サブコマンド内に数値が表示される部分について、ここでは*を示した

① 始点を設定
直線の始点を設定する。
Start of line (BothSides Normal Angled Vertical FourPo* Bisector Perpendicular Tangent Extension);

② 終点を設定
直線の終点を設定し、始点と終点を結ぶ線が描ける。
End of line (Bot*);

Tangent: 曲線に対する接線を引く

【凡例】

[line]
直線を引く
おすすめのエイリアス: l

通常は始点と終点を結ぶ線を描くコマンドだが、サブコマンドの中からその他の線の描き方を選ぶことができる。サブコマンドには、「Normal (法線)」「Perpendicular (直交)」「Tangent (接点)」などがある。

サブコマンド例

- Normal: 選択したサーフェスからの法線を引く
- Tangent: 曲線に対する接線を引く

① 始点を設定
直線の始点を設定する。
Start of line (BothSides Normal Angled Vertical FourPoint Bisector Perpendicular Tangent Extension);

② 終点を設定
直線の終点を設定する。始点と終点を結ぶ線が描ける。
End of line (BothSides);

[polyline]
ポリラインを引く
おすすめのエイリアス: pl

直線、または円弧の組み合わせた線 (ポリライン) を描く。意外と知られていないけれど、ポリラインを描いている途中でサブコマンドから「Mode (モード)」を選択すると円弧モードになる。最後にポリラインを閉じたい場合は「Close (閉じる)」を選択する。

① 始点を設定
ポリラインの始点を設定する。
Start of polyline (PersistentClose=No);

② 頂点を設定
ポリラインの頂点を設定していく。「Mode」を入力すると、円弧にできる。
Next point of polyline. Press Enter when done (PersistentClose=No Mode=Line Helpers=No Undo);

③ 終了を宣言
Enterキーを押すとポリラインが終了する。頂点が3つ以上ある場合は「Close」でポリラインを閉じることができる。

[curve]
カーブを引く
おすすめのエイリアス: cu

ラインで描く曲線は、NURBS (Non-Uniform Rational B-Spline) 曲線と呼ばれており、次数 (degree) と呼ばれる値と、制御点を設定して描く。NURBS曲線に関しては高度な説明が必要なのでここでは割愛。要するに、次数が高いと滑らかな曲線が描けて、低いと粗い曲線になるということ。あとは体で覚えましょう。

① 始点を設定
NURBS曲線の始点を設定する。カーブの次数はこの時点で設定する。
Start of curve (Degree=3 PersistentClose=No);

② コントロールポイントを設定
NURBS曲線のコントロールポイント (制御点) を設定していく。
Next point (Degree=3 PersistentClose=No Undo);

③ カーブの終了を宣言
Enterキーを押してNURBS曲線の終了を宣言する。
Next point. Press Enter when done (Degree=3 PersistentClose=No Close Sharp=No Undo);

[circle]
円を描く
おすすめのエイリアス: ci

中心の位置と半径、もしくは直径を設定し、円を描く。サブコマンドから、その他の円の作図方法を設定できる。

① 中心点を設定
円の中心を設定する。サブコマンドで3点で円を定義したり、接線から円を定義することもできる。
Center of circle (Deformable Vertical 2Point 3Point Tangent AroundCurve FitPoints);

② 半径 (直径) を設定
半径 (radius) を入力して円を定義する。直径 (diameter) による定義も可能。
Radius <=> (Diameter Orientation Circumference Area);

③ 円が生成
Enterキーを押すと円が生成される。

[join]
オブジェクトを結合する
おすすめのエイリアス: j

ばらばらの線や曲線が、複数接しているとき、それらを結合してポリカーブにすることができる。ただし、始点、終点が接していないとダメ。また、サーフェスのエッジ同士が接していればポリサーフェスに結合できる。使い勝手のよいコマンド。

① 結合するオブジェクトを選択
結合したいオブジェクトを選択する。
Select object for join;

② 結合するオブジェクトを選択
引き続き結合したいオブジェクトを選択していく。オブジェクトの端部同士が接しているか注意。
Select curve to join. Press Enter when done;

③ 結合
結合されて1つのオブジェクトになる。
4 curves joined into one open curve;

[explode]
オブジェクトを分解する
おすすめのエイリアス: exp

[join]と対になっているコマンド。ポリカーブやポリサーフェスなど、複数のセグメントが結合できているオブジェクトを、セグメントごとに分解することができる。なお、グループは分解できないので注意。予想以上に大量のパーツに分かれることもあるので、そのあたりにも注意。

① 分解するオブジェクトを選択
ポリカーブやポリサーフェスなどを選択。
Select object for explode;

② 分解
セグメント単位に分解される。
Exploded a curve into 9 segments;

4

立体を作る

3D空間の中で線を引けるようになったら、次はよいよその線を立体にしていく。ラインはサーフェス・モデラーと呼ばれ、立体をサーフェスの集合として作成していく。サーフェス・モデラーに対してソリッド・モデラーと呼ばれるアプリケーションでは、一般に閉じた立体はすべて中身が詰まったものとして扱われる。ラインでも閉じた立体を便宜的に「ソリッド」と呼んでいるが、ソリッドとして作成した球や立方体も、実は中身は空っぽなのだ。

[loft]

断面カーブを結ぶ曲面を作る
おすすめエイリアス: l

断面となるカーブをいくつか指定して、そのカーブをすべて通る面を生成する。

曲面を作る上で最もポピュラーで使い勝手のよい面生成コマンドの1つ。断面のカーブのコントロールポイントの数を [rebuild] コマンド (P.57) などで揃えておくと、スムーズな面になる。



1. 複数のカーブを選択

選択状態で面のでき方が違う。

Select curves to loft. Press Enter when done (Point).

Drag seam point to adjust. Press Enter when done

(Flip Automatic Natural)

2. オプションを設定

カーブを選択しEnterキーを押すと、面を生成する直前にダイアログボックスが出てくる。面の精度などを微調整できる。プレビューも表示できる。

3. 面が生成される

それぞれのカーブ上の [loft] の始点を動かして、できるだけ平面的に近い位置に揃えておくようにすると滑らかな曲面になる。

[extrudecrv]*

カーブを押し出して立体を作る
おすすめエイリアス: ext

カーブを1方向へ押し出して立体を作る。

曲面のない図形モデリングであれば、これ1つで事足りてしまうほど汎用性が高い。平面図から壁や床を立ち上げるときなどに使用する。押し出す距離は数値で設定できる。

*crv: curve (カーブ) のこと。ラインでは、直線と曲線ともに「カーブ」として扱う。



1. カーブを選択

原型となるカーブを選択。ソリッドにする場合、カーブが閉じていて同一平面上にあることを確認する。

Select curves to Extrude. Press Enter when done;

2. 面が生成される

初期設定ではZ軸方向だが、サブコマンドの「Direction (方向)」で押し出す方向を変更できる。

Extrusion distance=>
(Direction BothSides=No Solid=No DeleteInput=No ToBoundary SplitAtTangents=No SetBasePoint)

その他の

「押し出し (Extrude)」

直線だけでなく、曲線に沿ったり点まで押し出すなど、様々なオプションがある。モデリングの局面によってはかなり使えるものもあるので、色々試してみよう。

[planarsrf]*

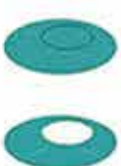
カーブ内に面を作る
おすすめエイリアス: ps

同一平面上にある閉じたカーブを選択し、その内側に面を張る。

カーブの中にさらに閉じたカーブがあれば、そこに穴を開けた状態で面を張ってくれる。

カーブが同一平面上にない場合は、[project] コマンド (P.47) などを使ってカーブを平面に作り直すといよい。

*srf: surface (サーフェス) のこと。



1. カーブを選択

面を張りたいカーブを選ぶ。

Select curves to build surface. Press Enter when done;

2. 面が生成される

穴の開いた面を作りたいときには、カーブを複数選択する。

[planarsrf] でできないパターン
エッジカーブが閉じていないか、同一平面上にない。[planarsrf] を実行できない。

Curve must be closed and planar.



[sweep1]

1本のレールと断面カーブから曲面を作る
おすすめエイリアス: s1

まずレールとなるカーブを1つ指定し、次に基準となるカーブを指定すると、断面カーブがレールに沿って動いて面を生成する。

レールと断面カーブは接していなくてもよい。パイプ状のものなど、特定の断面を曲線に沿わせてできるオブジェクトを作りたいときなどによく用いられる。



1. レールと基準カーブを選択
レールカーブを1つ選択し、サーフェスが通る際に断面カーブを選択する。
Select rail (ChainEdges);
Select cross section curves (Point);

2. オプションを設定
面の滑らかさなどの微調整は、面の生成直前に出てくるダイアログボックスにて設定できる。

3. 面が生成される



[sweep2]

2本のレールと基準カーブから曲面を作る
おすすめエイリアス: s2

[sweep1] コマンドと原理は同じだが、こちらはレールとなるカーブを2つ指定できるため、より複雑なオブジェクトを作ることができる。



1. 第1、第2と断面カーブを選択
断面カーブは曲面の入口と出口の2つを選択することもできる。
Select first rail (ChainEdges);
Select second rail;
Select cross section curves.
Press Enter when done (Point);

2. オプションを設定
面の滑らかさなどの微調整は、面の生成直前に出てくるダイアログボックスにて設定できる。

3. 面が生成される



[networksrf]

複数の断面カーブを通る曲面を作る
おすすめエイリアス: net

縦方向の断面カーブと、横方向の断面カーブをすべて選択し、それらを通る曲面を生成する。複雑な曲面を生成するときに用いる。

縦方向カーブはすべての横方向カーブと交差していなければならない。横方向カーブはすべての縦方向カーブと交差していなくてはならないことに注意。



1. 縦・横方向のカーブを選択
左記の注意事項を守って、すべてのカーブを選択しよう。
Select curves in network.
Press Enter when done (NoAutoSort);

2. オプションを設定
面の滑らかさなどの微調整は、面の生成直前に出てくるダイアログボックスにて設定できる。

3. 面が生成される



[revolve]

回転体を作る
おすすめエイリアス: なし

断面となるカーブを選択して、回転軸と回転角を入力し、軸の周りにカーブが回転した面を張る。軸は直線である必要がある。

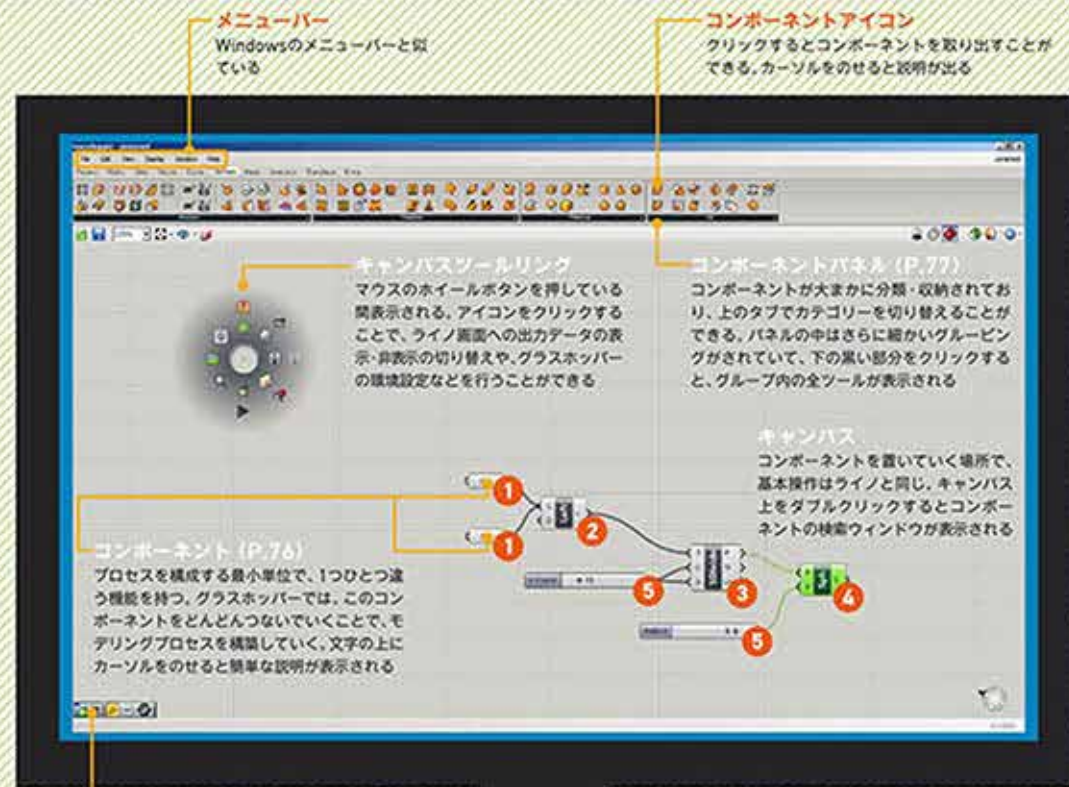


1. 断面カーブ、軸を設定
断面カーブを1つ選択し、回転軸 (直線) の始点と終点を設定する。
Select curves to revolve. Press Enter when done;
Start of revolve axis;
End of revolve axis (Press Enter to use CPlane z-axis direction);

2. 回転角度を指定すると、面が生成される
回転角度を度数で入力する。
Start angle <0>
(DeleteInput=No Deformable=No FullCircle AskForStartAngle=Yes SplitAtTangents=No);
Revolution angle <360>
(DeleteInput=No Deformable=No FullCircle SplitAtTangents=No);

大図解！これがグラスホッパーだ！

グラスホッパーはライノセラスのプラグインの1つだ。インストール後、ライノで [grasshopper] とコマンド入力すると、下の操作画面が起動する。グラスホッパーの操作画面は、ライノでの操作画面と必ず対応しているの、ここでは右のライノ画面と比較しながら説明する。



画面 1 グラスホッパー
右の画面2 (ライノセラス) と対応している

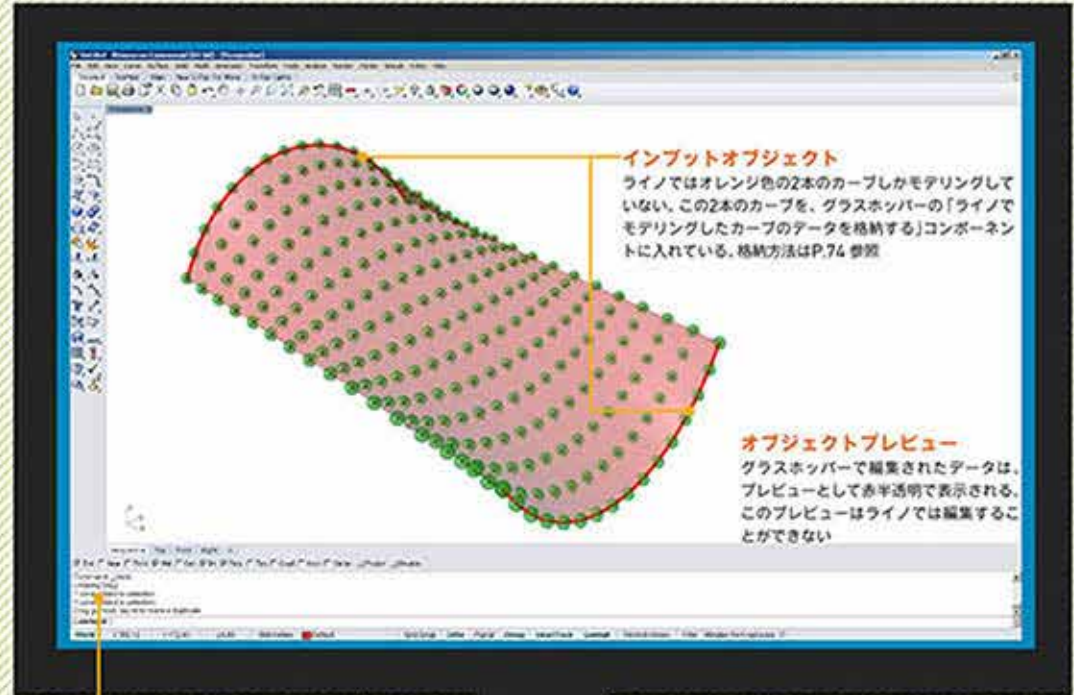
コマンドを組み合わせてオリジナルツールを作る

グラスホッパーでモデリングプロセスを作るといことは、言い換えればライノで使うことのできる自分だけのオリジナルツールを作るといことだ。たとえば下の概念図では、「選択した2本のカーブの間に面を張り、その面のUVグリッド上に球を出す」ツールをグラスホッパーで作ったことになる。さらにそのグリッドの細かさや球の半径はグラスホッパー内で自由に変更することができる。ツールの実行結果を後で編集できる点が、グラスホッパーの最大の特徴である。



ライノセラスとの対応を確認しよう！

グラスホッパーのプロセスの起点となるコンポーネントに、ライノでモデリングしたオブジェクトのデータを格納して、グラスホッパーとライノを同期する。以後、グラスホッパーで作るモデリングのプロセスが、ライノの画面に反映されていく。下の例では、ライノでは2本のカーブしかモデリングしていない。



画面 2 ライノセラス
左の画面1 (グラスホッパー) と対応している

モデリングの工程表を作成する

グラスホッパーは、それぞれ異なる機能を持ったコンポーネントを左から右へつないで、モデリングの工程表を作成するプラグインだ。コンポーネントにデータが入力されると、ライノで作成されたオブジェクトは工程表で指示された順番通りに自動的に加工され、目的のモデルとなって出力される。グラスホッパーのファイルを開読するときは、原則として左から右へと読んでいく。

ライノで作成したオブジェクトをプロセスの起点にする

コンポーネントにはライノで作成されたオブジェクトのデータを格納できるものがある (格納の方法はP.74参照)。左の画面1では、「ライノでモデリングしたカーブのデータを格納した」コンポーネント①を「カーブを入れるとロフトされた面が出る」コンポーネント②につなぎ、それを「面を入れるとグリッドの交点が出る」コンポーネント③に、さらにそれを「入力した点を中心とした球が出る」コンポーネント④につなぐことで、上記のライノ画面にサーフェスとグリッドにのった球をプレビューしている。オブジェクトプレビューの球の半径と密度は、「入力する数値を変更することができる」スライダー状のコンポーネント⑤のつまみを動かすと変更される。ライノでモデリングされているのは2本のエッジカーブだけで、面や球はいずれもグラスホッパーでモデリングされているのだ。そのため、グラスホッパーの操作画面を開くと、ライノの画面上では面と球が消えてカーブだけが残る。

1 アトラクターを使う

アトラクター(attractor)とはもともと複雑系理論などで用いられている概念だが、「引きつけるもの」という邦訳の通り、ここでは「空間や場に影響を及ぼす点やモノ」という意味で使っている。グラスホッパーの特性を利用すると、アトラクターによって影響を受けるモデルを作成できる。これを応用すると力学的なシミュレーションや、自然界に存在する複雑な形状をラインでモデリングできるようになる。

ボロノイ・スカイスクレイパーのレシピ

用意するもの:

ライン・オブジェクト: ボロノイ図を作成する点群、アトラクター点1個

グラスホッパー・コンポーネント: [Point] [Box] [Distance] [Extrude] [Unit Z] [Voronoi] [Cap Holes]

1. 点群とアトラクター点をセットする

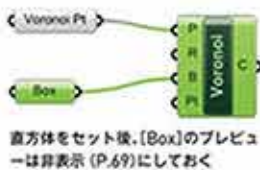
点群を1つ目の[Point] (P.111)に、アトラクターを2つ目の[Point]にセットする。複数の点をまとめてセットするときは、「右クリック>Set Multiple Points」で選択。右クリックメニューでコンポーネントをそれぞれ[Voronoi Pt] [Attractor Pt]とする。



セット内容に応じてコンポーネントの名前を変更

2. ボロノイ図*を作成する

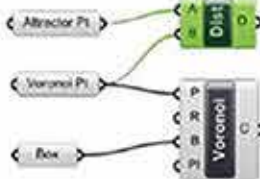
[Voronoi] (P.123)のP入力に[Voronoi Pt]を接続する。また、[Box]にボロノイ図の領域を指定する直方体をセット(右クリック>Set one Box)して[Voronoi]のB入力に接続すると、C出力から各セルのカーブデータが出力される。



直方体をセット後、[Box]のプレビューは非表示 (P.69) にしておく

3. 各ボロノイ点とアトラクターとの距離を測る

[Voronoi Pt]と[Attractor Pt]を[Distance] (P.117)のAとB入力に接続する。D出力からはボロノイの基準点からアトラクター点までの距離の数値が出力される。

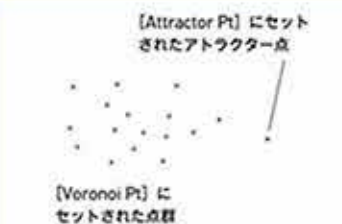


4. 3で求めた距離を各セルの高さに設定する

[Distance]を[Unit Z] (P.118)を介して[Extrude] (P.122)のD入力に、[Voronoi]を[Extrude]のB入力に接続する。3.で求めた数値をZ軸方向の高さとして、対応するそれぞれのボロノイ・セルを立ち上げる。



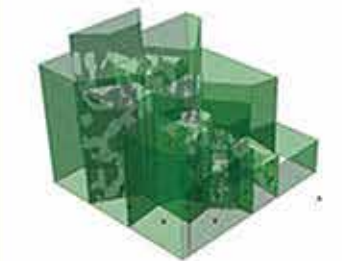
ライン画面でのプレビュー



ボロノイ図の領域を設定する直方体を [Box] にセットする



*ボロノイ図: 任意の位置に配置された複数個の点(母点)に対して、各々の母点同士の二等分線によって領域分けされた図のこと。ボロノイの各領域をセルと呼ぶ



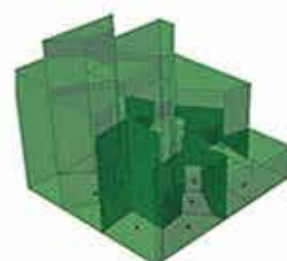
各セルを押し出す高さは、アトラクター点との距離で決まる

5. 立ち上がったオブジェクトをソリッドにする

[Extrude]を[Cap Holes]に接続する。立ち上がったセルが閉じたポリサーフェスになる。この段階で、ライン画面のプレビューに[Extrude]に属しているジオメトリと[Cap Holes]に属しているジオメトリが重なってしまうので、[Extrude]のプレビューを非表示にしておくとうい。



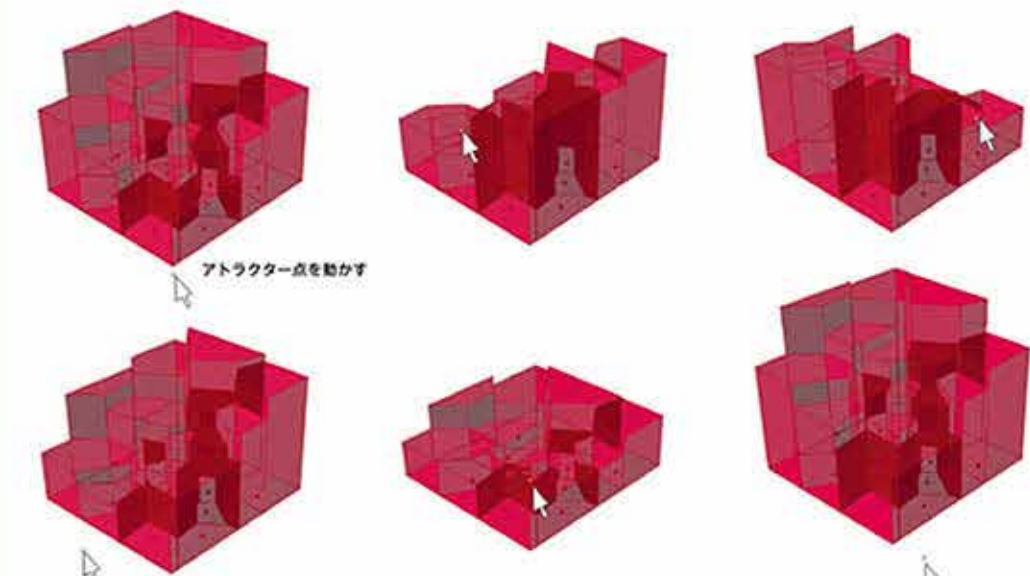
ライン画面でのプレビュー



カーブを立ち上げた形状のオブジェクトに[Cap Holes]が接続され、閉じたポリサーフェス(ソリッド)が作成

6. ラインでアトラクター点を動かしてみる

ラインでアトラクター点の位置を変更すると、ボロノイのセルとの距離も変化するので、それに応じて各セルの立ち上がりの高さが変わる。ここでは、グラスホッパーを使ってアトラクター点とボロノイ立体の「関係性」をデザインしている。



練習問題

1. アトラクター点が近づくとき、それに合わせて大きくなるようなグリッド上の円を作る。(ヒント: [Circle CNR] を使おう)
2. 始点が固定され、終点が常にアトラクター点の方向を向くような長さが一定の線を作る。(ヒント: [Line SDL] を使おう)

(解答はP.129)