

Trelis関連リンク

- Trelis ホームページ

<http://www.trelis.jp>

- M&T Support

<http://support.jpmandt.com/>

様々な操作方法の事例をアーカイブしています。

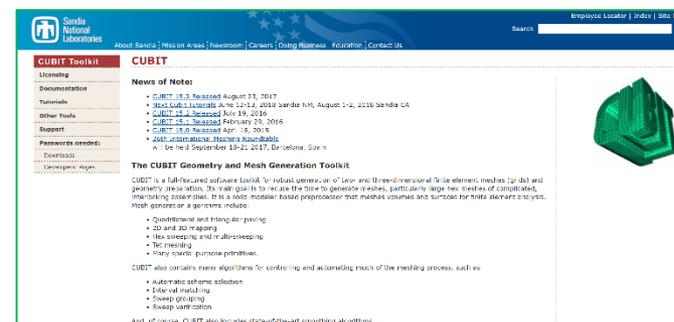
- Sandia CUBIT

<https://cubit.sandia.gov>

CUBITのドキュメント、チュートリアル資料、Pythonスクリプトのサンプルなどをご覧いただけます。

- Google検索

Trelis Cubit Sandia mesh etc.



目次

受講の心得

Trelis (CUBIT) にできること

イントロダクション

チュートリアル ゲットिंगスタート

マウス操作

ユーザインターフェースの基礎

チュートリアル CUBITの作業工程

基本操作演習① Boolean

基本操作演習② ショートカット

ジオメトリツール

"Mesh Vol All"

メッシュコントロール

基本操作演習③ コマンドライン

基本操作演習④ カーブバイアス

Pythonを使ってみよう

受講の心得

・ 知ること

- 何ができて、何ができないか
- どんな機能が、どこにあるのか
- 自分のやりたいことをするために、どのように調べればよいのか

・ 体験すること

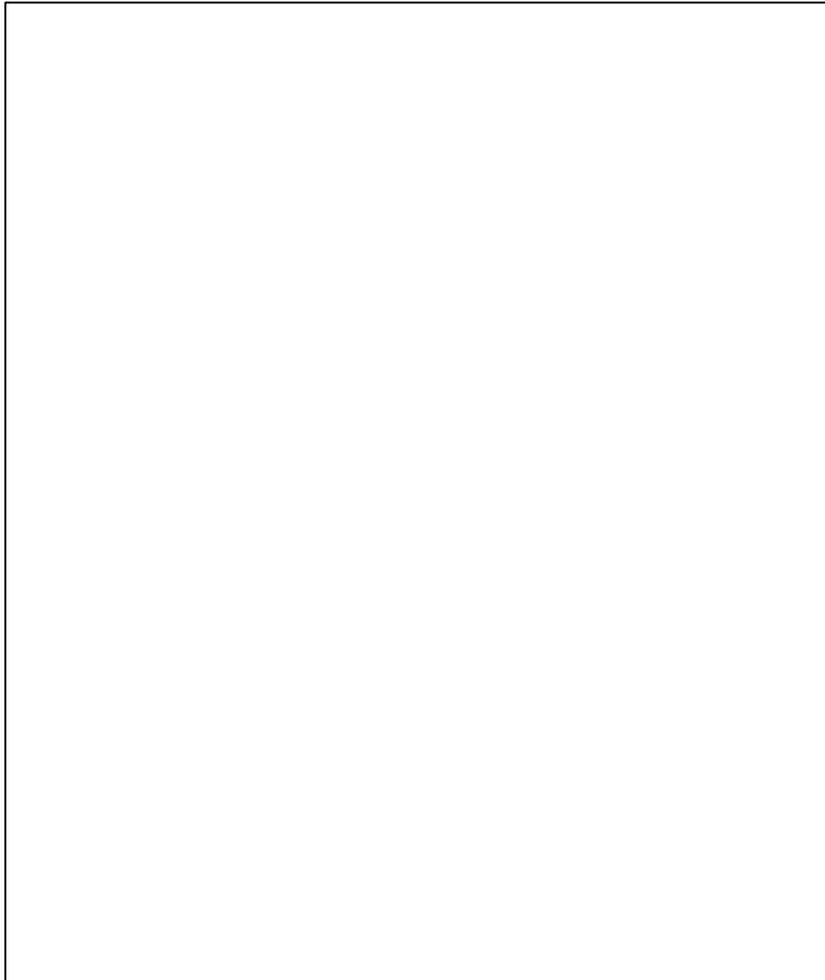
- ソフトウェアの性質の理解を心がける
 - 仕組みを理解
 - 操作方法に慣れる
 - 車の運転方法を覚えるのと同じで、ソフトウェアには、癖があり、慣れるにはある程度時間を要します
 - 得意、不得意を理解し、業務に効率よく利用する
 - Cubitが得意な業務で活用してください

受講の心得

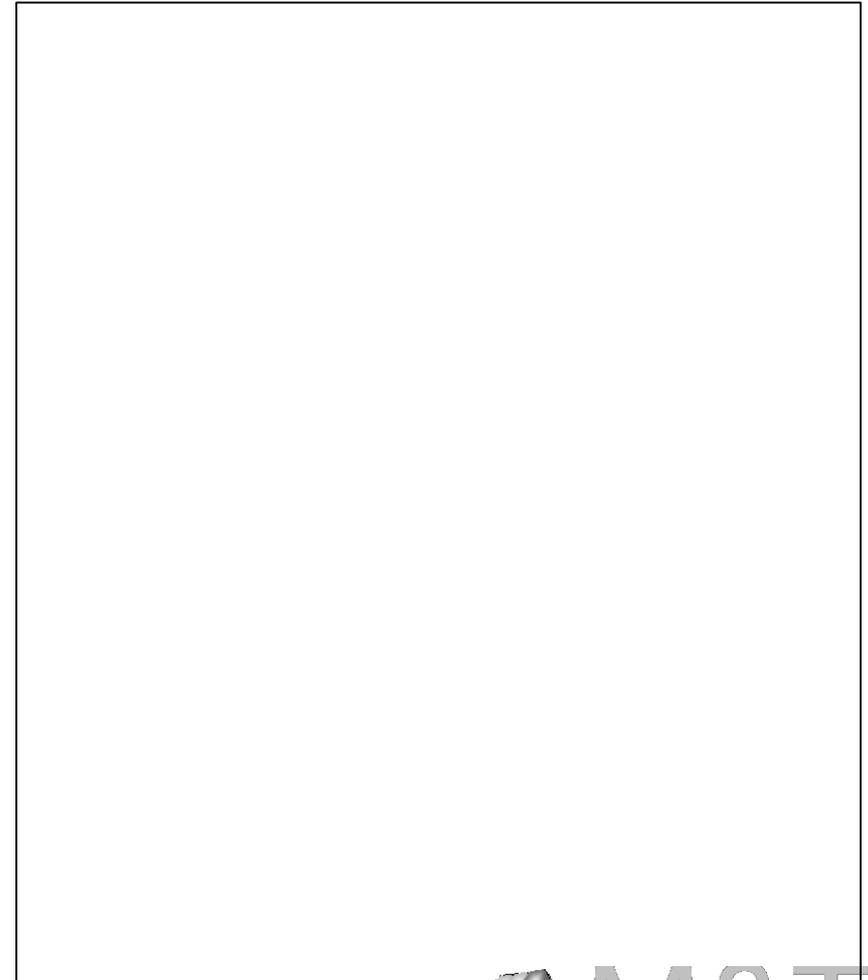
- **製品のコンセプト・思想・ゴールを理解すること**
 - 開発者の気持ちになる
 - モデル／メッシュのプログラミング
 - Cubitは、モデリング／メッシング用 超高級言語
 - エンティティのプロパティ
 - mesh vol all
- **操作手順を覚えな**
 - 手順は人それぞれ、しかし良い手順は存在する
- **自分がやりたいことをイメージする**
 - **イメージを一度、紙に整理してみる**
 - モデルの描き方の手順
 - メッシュの仕上がり
- **最後に、イメージを達成できる手順をCubitに命令するだけ**
 - Cubitはあなたの命令に従います

イメージを描いてみよう

モデルイメージ



メッシュイメージ



良いメッシュを作成するのに必要なスキル

実現象を高精度に再現するためには
目的のヘキサメッシュをイメージできる必要があります

- ユーザーは、物理に精通している必要があります
- メッシュを作成する前に、流れのイメージが必要です
- 物理現象をイメージできないと、現象を高精度に再現するためのメッシュを作成できません

目的のヘキサメッシュイメージに対してヘキサメッシングを行うには

- 空間認識能力が必須です
- 数学パズル、ブロック遊び、テトリスに類似のゲームスキル
- 逆に言えば、ヘキサメッシングされた対象物のイメージさえあれば、予備知識は不要です
- 極論を言えば、パズル同様、利用可能なピース(スキーム)を当てはめる(領域分割する)だけなので、子供から大人まで誰でもできます

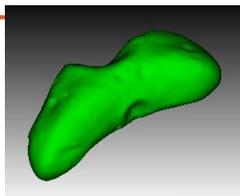
Trelis (Cubit) に出来ること



		Trelis	備考	
インポート／エクスポート※	CAD	○		
	メッシュ	○		
ソリッドモデリング※	作成	○		
	編集／修正	○		
メッシュ作成	ヘキサ (Hex)	Auto	☆☆☆	Bolt
		Semi-Auto	☆☆☆	
		Manual	☆☆☆	
	混合 (Hybrid)	Auto	☆☆☆	
		Semi-Auto	☆☆☆	
		Manual	☆☆☆	
	テトラ (Tet)	Auto	☆☆☆	
		Semi-Auto	☆☆☆	
		Manual	☆☆☆	
		スムージング	○	
自動化・組込	ジャーナル・API	○	Python/C++	

※CADカーネルにACISを採用

メッシュ作成プロセスとTrelisの機能対応状況



メッシュ形式で
インポート
(ポリゴン)

		メッシュ形式で インポート (ポリゴン)	CADあり	CADなし
形状 (ACIS)	作成	○	○	○ ※
	追加	○	○	○ ※
	編集／修正 (Modify)	△ ※	○ ※	○
	修復 (Healing)	×	○ ※	○
トポロジ	編集	○	○	○
メッシュ作成	ヘキサ (Hex)	△	◎	◎
	混合 (Hybrid)	○	◎	◎
	テトラ (Tet)	◎	◎	◎
スムージング		○	◎	◎

・ 形状作成には、ソリッドモデラーのエンジンを利用

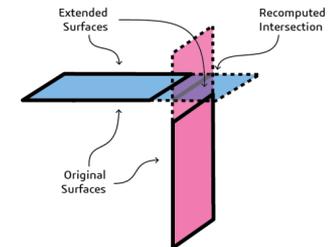
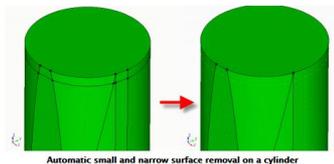
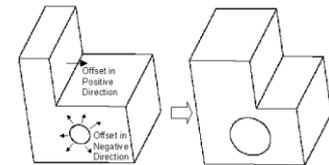
※の作業で主に時間が掛かります。

- ・ CADの処理には強い
- ・ CADの中間ファイルの利用には互換性の注意が必要
 - ✓ CATIAからの出力には要注意
- ・ CGの処理には弱い(ファセット、ポリゴン、STL)

Cleanup & Defeaturing

メッシングのための、ジオメトリ・トポロジの編集・修復

ジオメトリの編集	Removing Geometric Features	サーフェス／パーテックスの除去
	Tweaking Geometry	オフセット、置き換えなどによる編集
	Trimming and Extending Curves	カーブをトリム・延長
トポロジの編集	Auto Clean	ジオメトリの自動クリーンアップ
	Regularizing Geometry	不要なトポロジーを除去
修復	Healing	不正確なACISモデルを修復
	Debugging Geometry	トポロジーの矛盾を検査
	Finding Surface Overlap (Geometry Accuracy)	重なるサーフェスを検知 (ジオメトリ精度の変更が可能)
	Stitching Sheet Bodies	シートボディを縫合
	Validating Geometry	ジオメトリを検証

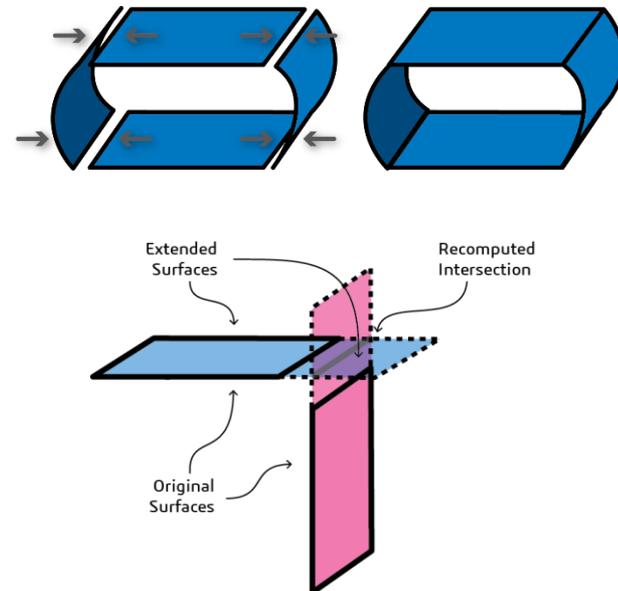
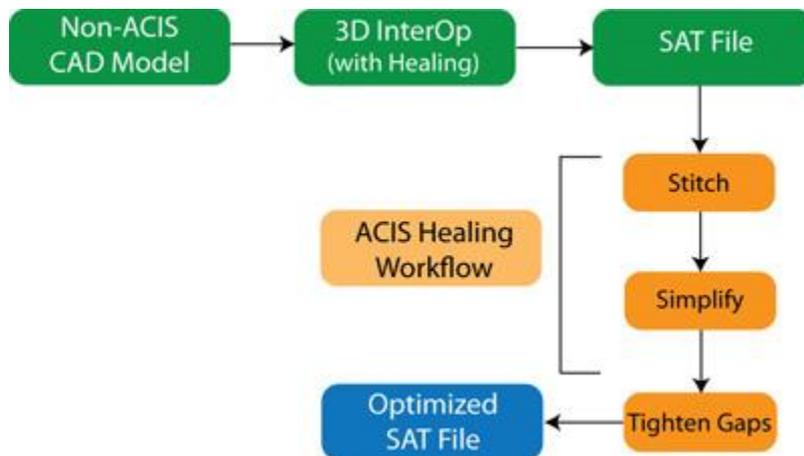


ACIS Healing Workflow

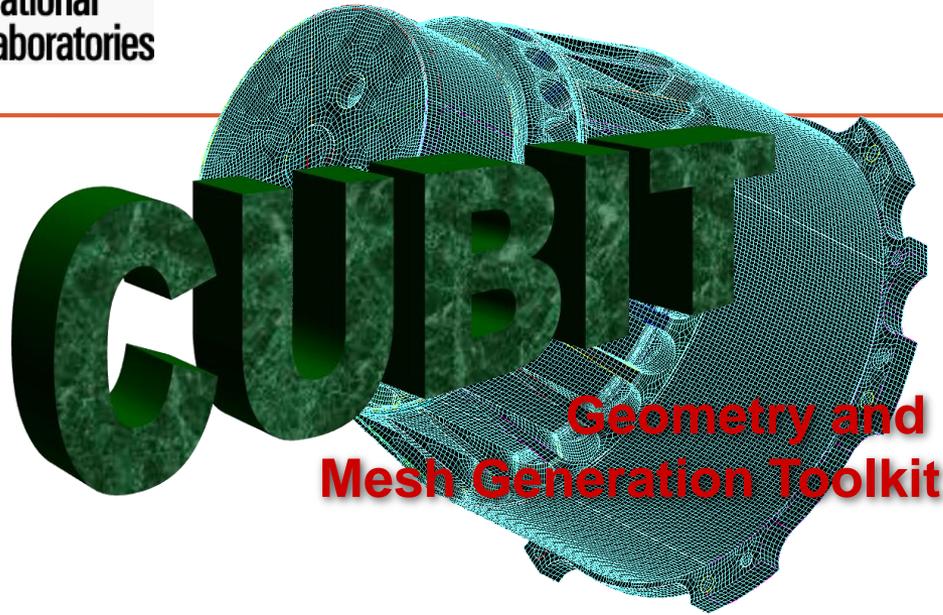
ジオメトリをACISフォーマットに変換すると、度々問題を生じます。

- デフォルトのトレランスだと、CATIA(0.1), Creo, I-DEAS(1e-2), Solidworks(1e-5)は ACIS(1e-6) より不正確です。
- 隙間、オーバーラップ、内部の不一致はよく起こります。

Trelisでは、他のフォーマットから変換したACISモデルに対して自動でヒーリング(Healing)を行います。



トレーニング時間は限られていますので、
適宜ソフトを触りながら講義を聞いてください。



CUBIT ファーストスタートチュートリアル イントロダクション

ここでは

- ・Cubitのヘキサメッシングに対する考え方
 - ・ヘキサメッシングを実現するための方法
 - ・ヘキサメッシングを行うために必要なツール
- を紹介します。

製品のコンセプト・思想・ゴールを理解すること



CUBITは、メッシュ作成にかかる**作業時間を減らすこと**を目的として開発が続けられています。

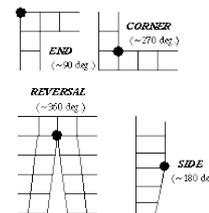
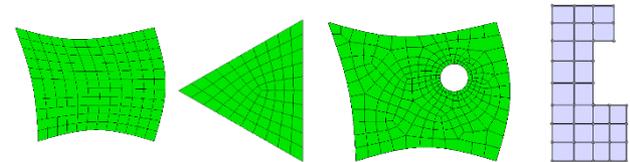
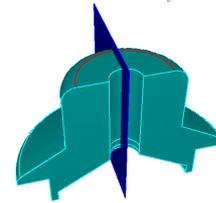
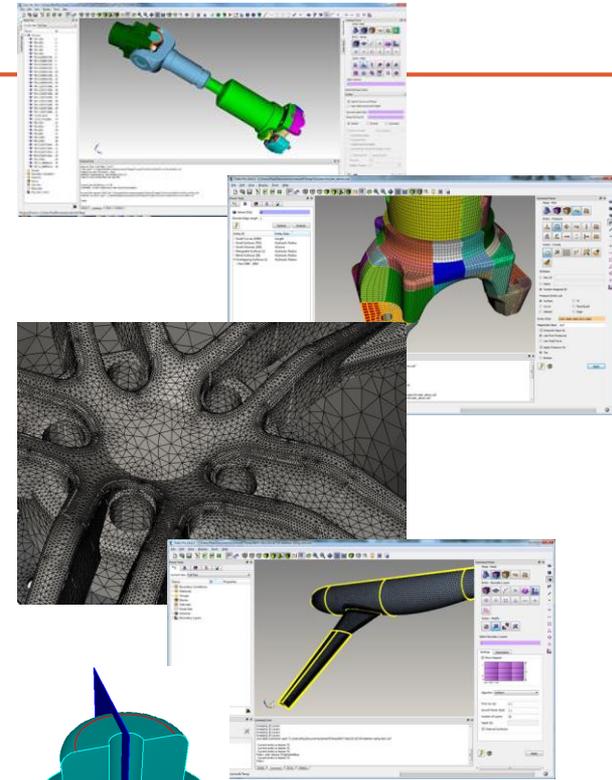
- ・ソリッドモデラで形状を作成
- ・直接ソリッドモデルに大規模自動ヘキサメッシング

歴史的には

- ・形状作成にソリッドモデラーを採用(世界初!)
- ・全自動ヘキサメッシュアルゴリズムの研究開発を目指す

現実的には

- ・全自動ヘキサメッシュは非常に困難
- ・ヘキサメッシング(pave & sweep)可能なモデルのための**分割ツール(Webcut)**の開発
- ・ロバストな大規模**ヘキサメッシングスキーム**の開発
- ・スキームの自動選択機能、インターバルマッチング機能
 - その他、**テトラ**／**ハイブリッド**／**境界層** etcの開発



ユーザインタフェース

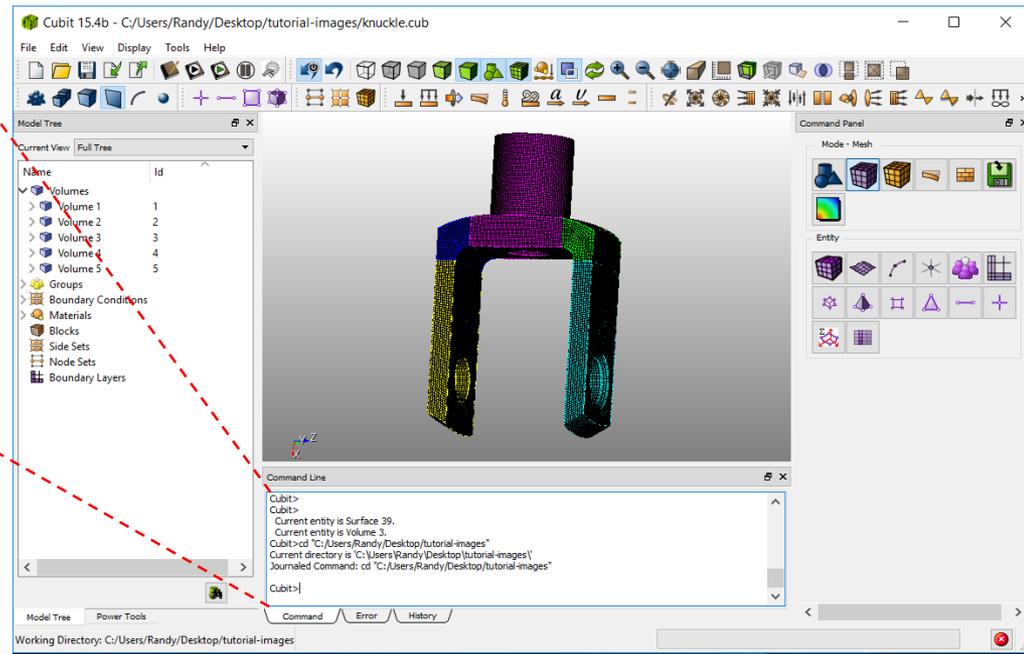
コマンドライン

```
Command Line
Cubit>
Cubit>
Current entity is Surface 39.
Current entity is Volume 3.
Cubit>cd "C:/Users/Randy/Desktop/tutorial-images"
Current directory is 'C:/Users/Randy/Desktop/tutorial-images\'
Journaled Command: cd "C:/Users/Randy/Desktop/tutorial-images"
Cubit>|
```

Command Error History

- Cubit command
- Python
- パフォーマンスの向上
- パワーユーザ向け
- 習得が難しい

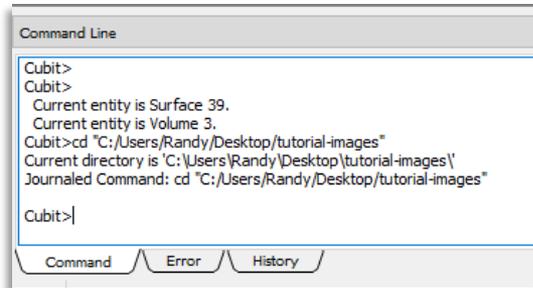
GUI



- 相互的、直観的な操作
- 様々なツールを利用できる
- 習得が容易

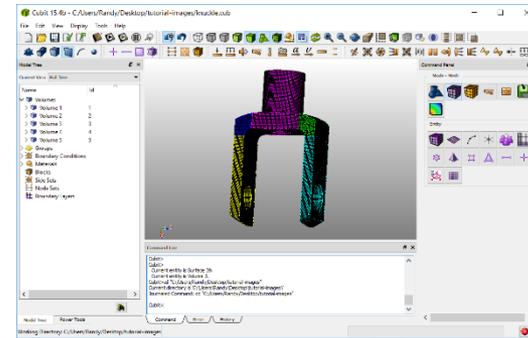
ユーザインタフェース

コマンドライン



```
Command Line
Cubit>
Cubit>
Current entity is Surface 39.
Current entity is Volume 3.
Cubit>cd "C:/Users/Randy/Desktop/tutorial-images"
Current directory is 'C:/Users/Randy/Desktop/tutorial-images/'
Journaled Command: cd "C:/Users/Randy/Desktop/tutorial-images"
Cubit>
```

GUI



Cubitは、コマンドラインベースのソフトウェアです。

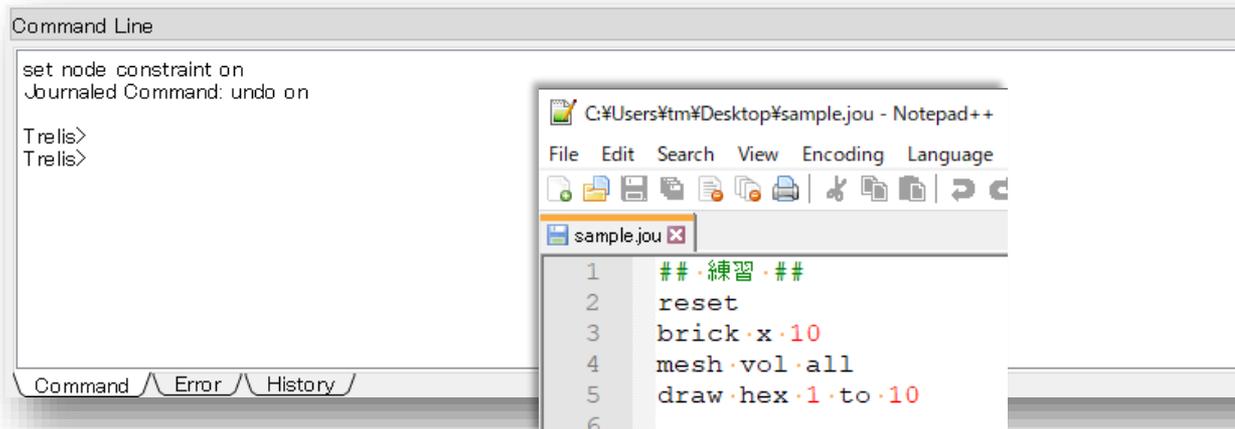
Cubit のGUI は、コマンドラインにコマンドを発行するためのツールですので、コマンドを利用したいシチュエーションで使えるようにGUI上にレイアウトされています。

例: 右クリックメニュー, Power Tools, モデルツリー, コマンドパネル, ツールバー

ほぼ全てのコマンドは、コマンドラインから利用可能です。

コマンドライン

- 全てのコマンドはコマンドウィンドウから入力できます。
- 一部の単語は短縮できます。
- IDの範囲を指定できます。
 - draw curve 1 to 5 except 4
 - draw curve in volume 2



推奨！

好みのテキストエディターでジャーナルファイルを作成し、Copy&Pasteまたは一括で読み込んでコマンドを実行することもできます。

```

## 練習 ##
reset
brick x 10
mesh vol all
draw hex 1 to 10

```

##メッシュサイズ指定##

```

reset vol all
vol all size 0.25
mesh vol all
draw hex 1 to 10
compress
draw hex 1 to 10

```

##メッシュスキーム指定##

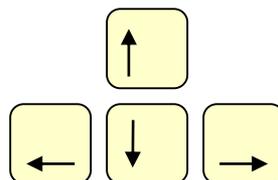
```

reset vol all
vol all scheme tet
mesh vol all
draw tet 1 to 10

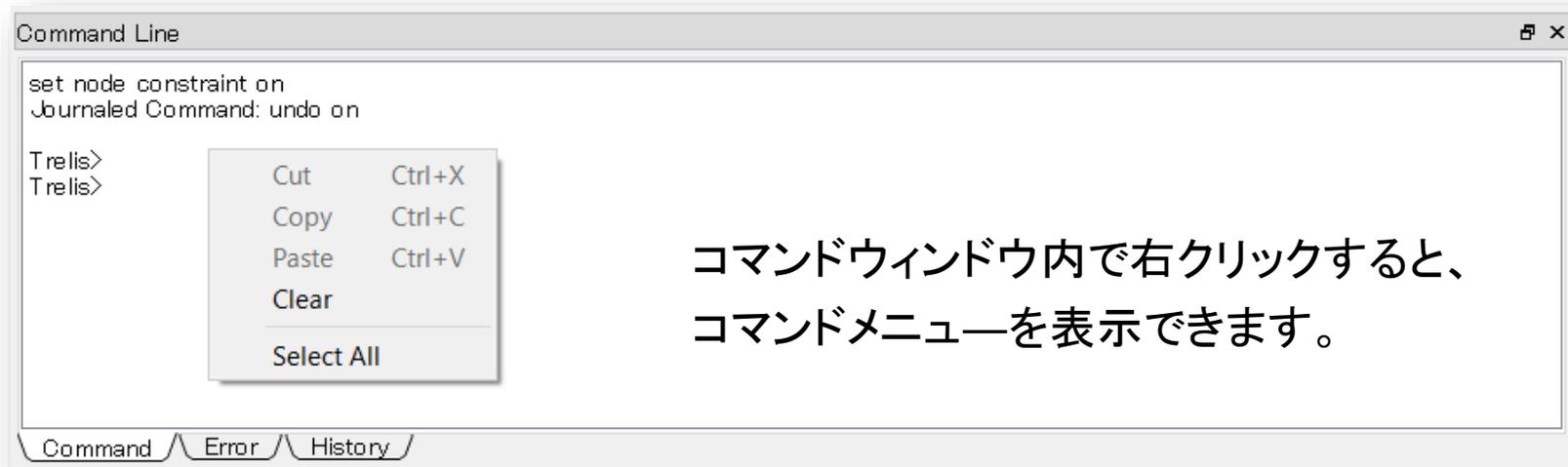
```

コマンドライン

キーボードの↑と↓キーを使って以前実行したコマンドを参照できます。



以前のコマンドを選択、右クリックして、copy/paste を選択することもできます。



コマンドウィンドウ内で右クリックすると、
コマンドメニューを表示できます。

コマンドライン ヘルプ

- | コマンドの一部を打ち込んだ後、‘?’ とタイプします。
入力した部分から始まるコマンドの一覧をアルファベット順で表示します。

```
CUBIT> create?
```

- | `help` の後に続いてキーワードを入力します。
語順に関係なく、入力したキーワードを使用するコマンドを表示します。

```
CUBIT> help create vertex
```

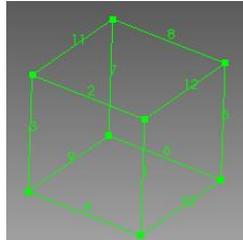
- | コマンドシンタックスの意味

- { } - 中括弧内のうち少なくとも1つがコマンドに含まれる。
- [] - 角括弧内のアイテムはオプションです。
- <> - 山形括弧内にはパラメータが入ります。
- | - ORを意味します; 括弧内での条件設定に使用します。

コマンドライン IDの取得

- グラフィクスウィンドウに、各エンティティのIDを表示できます。

- `CUBIT> label curve on`



カーブのIDラベルを表示した例

- エンティティを選択することで、コマンドライン上でIDを取得できます。

- IDが必要になるまでコマンドを入力します。

- `CUBIT> draw surface`

- エンティティをマウスで選択します。

(この時、選択したエンティティのIDがコマンドラインに表示されます。)

Current entity is Surface 34.

- `CUBIT> draw surface`

- e キーを押すと、コマンドラインにIDが挿入されます。

- `CUBIT> draw surface 34`

コマンドライン エラータブ



```
Command Line
set node constraint on
Journalized Command: undo on

Trelis>
Trelis>brick x10
ERROR:
  Unrecognized Identifier: 'x10'
ERROR:
  syntax error (<stdin>, line 7)
Trelis>
```

Command Error History

エラーが発生すると、このアイコンが表示されます。



```
Command Line
Command: brick x10
ERROR:
  Unrecognized Identifier: 'x10'
ERROR:
  syntax error (<stdin>, line 7)
```

Command Error History

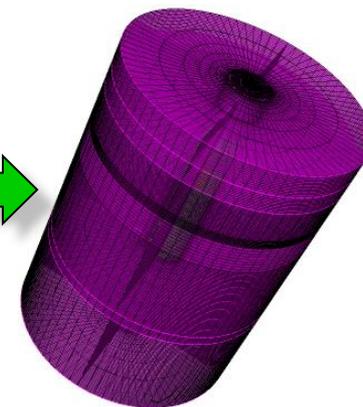
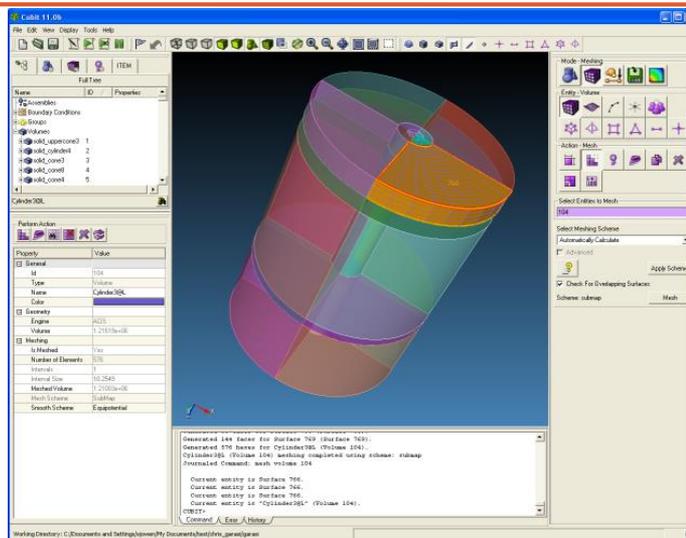
エラータブをクリックすることで、
エラーの詳細がわかります。

コマンドライン ヒストリータブ



ヒストリータブをクリックすると、コマンド履歴を参照することができます。

形状読み込みからメッシュ出力までの機能紹介



CAD モデル

- ACIS
- STEP
- IGES
- Facets
- STL
- Exodus II
- Cubit

CUBIT

- ジオメトリ作成
- ジオメトリ編集・修正・修復
- ジオメトリ分解
- メッシングツール
- メッシュスムージング
- 境界条件
- ジャーナル機能
- Python スクリプト
- 自動化 (Python API)

メッシュ

- Exodus II
- Abaqus
- IDEAS-Universal
- NASTRAN-BDF
- Patran
- LS-Dyna
- Fluent
- OpenFoam

CUBIT ジャーナル機能

ジャーナルの
作成・編集

ジャーナルファイル
の再生

CUBIT ジャーナルファイルには以下が含まれます:

- 一連のCUBIT コマンドのリスト
- コメント
- APREPRO スクリプト

```
*C:\Users\%tm\Desktop\%sample.jou - Notepad++
File Edit Search View Encoding Language Settings Too
sample.jou
1 ##練習##
2 reset
3 #{abc=1}
4 brick.x.{abc}.y.{abc*2}.z.4
```

完成したジャーナルファイルを再生すれば、
複雑なモデルでも簡単に再構築できます。

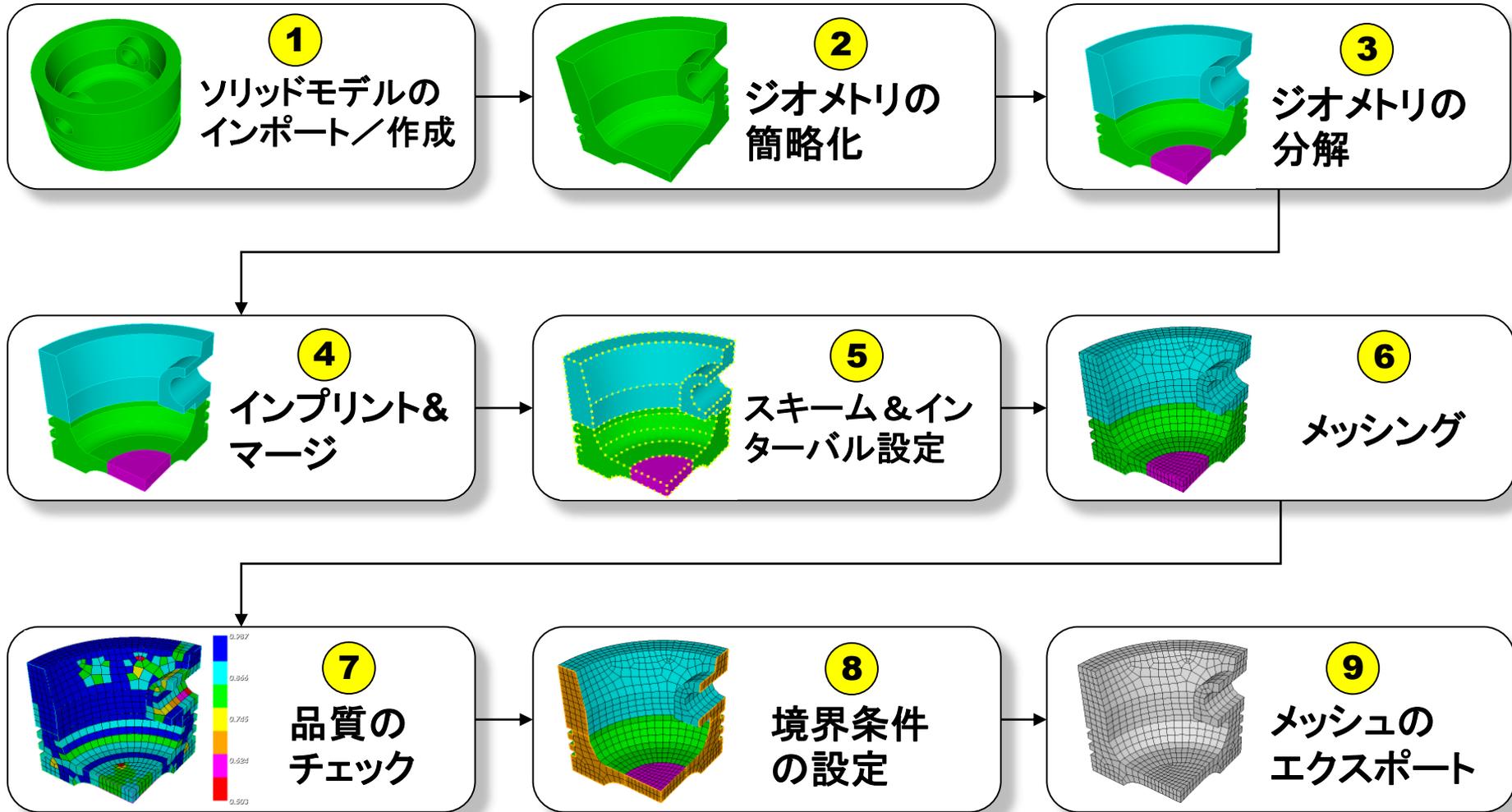
ジャーナルファイルの自動作成

CUBIT起動時に自動で新しいジャーナル
ファイルが作成され、セッション中で実行
されたコマンドは全てそのジャーナルファ
イルに記録されます。

(デフォルトでは、cubitXX.jou もしくは
historyXX.jouという名前ですインストール
ディレクトリに保存されます。)

ジャーナルは、Script/Command/History Tab
及びCubitファイルから抽出可能

Cubitの基本的なメッシングプロセス

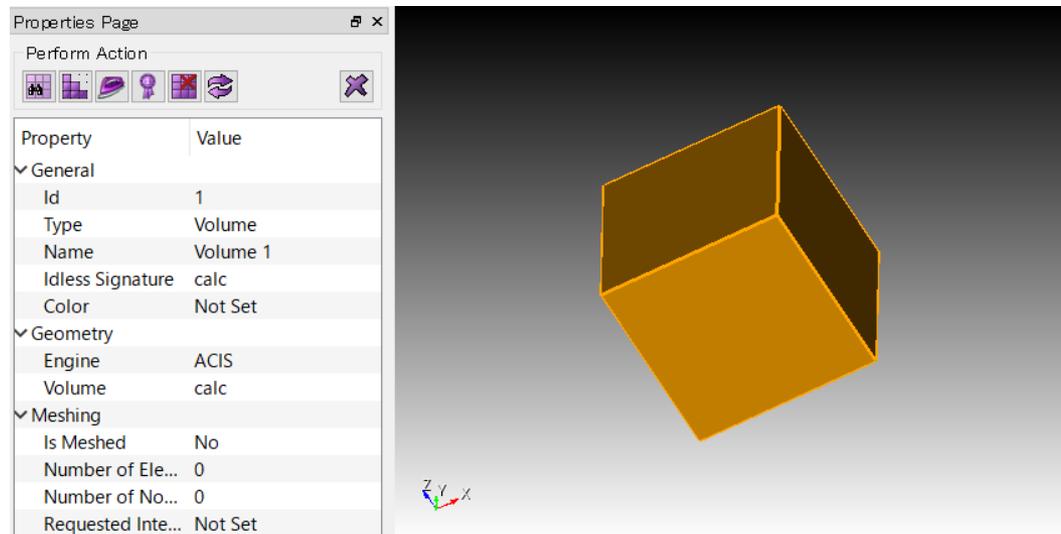


復習

エンティティ (Entity)

CUBITにおけるエンティティとは、プログラム上で扱うモノの総称です。ほとんどのエンティティは、編集可能なプロパティを持っています。モデル内の全てのエンティティは、モデルツリー上で確認できます。

エンティティはタイプとIDで管理されますが、ユーザが設定した任意の名前によって管理することもできます。



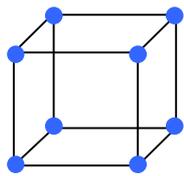
復習

CUBITのエンティティタイプ

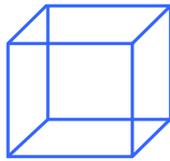
```
# コマンド例
reset
brick x 10
draw surface 2 to 4
```

ジオメトリ エンティティ (CADのエンティティ)

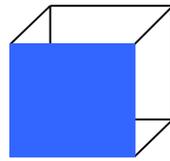
コピーして、コマンドラインに入力してみましょう



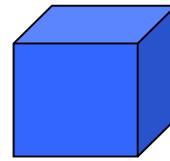
Vertex



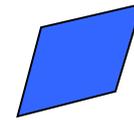
Curve



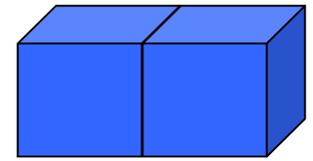
Surface



Volume



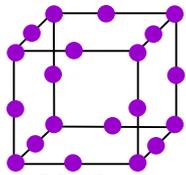
Sheet Body



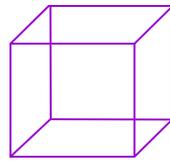
Body

メッシュ エンティティ

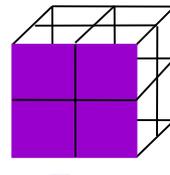
(同次元のジオメトリエンティティの近似表現)



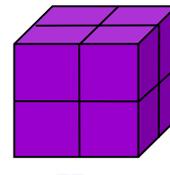
Node



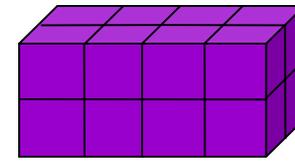
Edge



Face



Hex



Hex

```
# コマンド例
reset
brick x 10
mesh vol all
draw hex 5 to 10
```



Tri



Tet



Pyramid



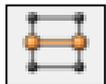
Wedge

CUBIT はまず **Vertex** をメッシングし、次に **Curve**、**Surfaces**、**Volumes** の順にメッシングします。
(Advancing Front パラダイム)

復習

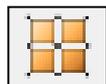
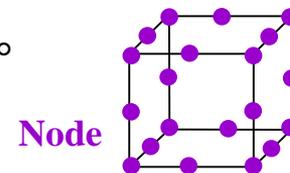
境界条件の設定

境界条件(メッシュ要素)のグルーピングには、3つの種類があります。



Nodeset: モデル中のノードへの負荷設定に使用します。

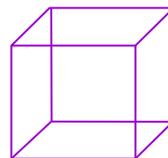
例: Temperature, Force



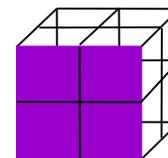
Sideset: モデル中の境界要素(エッジ、フェース)への負荷設定に使用します。

例: Pressure

Edge

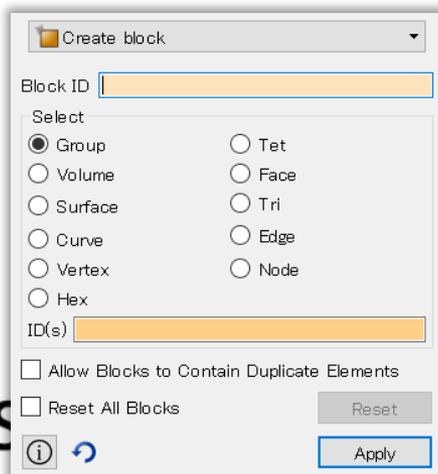
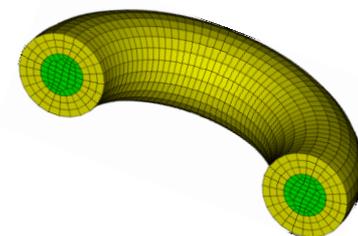


Face



Block: メッシュ要素をグルーピングして、マテリアル(CFDでは、流体/固体/ポアラス領域)の設定に使用します。

1~3次元のメッシュ要素を使用して定義できます。



メッシュ要素のグルーピングは以下の情報を使って定義できます。

- ジオメトリ
- メッシュ
- グループ

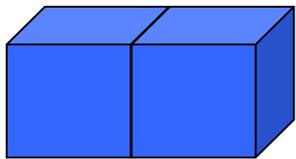
例えば、メッシュ要素単位で Sideset を定義すれば、サーフェスを分割する手間が省けます。



復習

CUBITのエンティティタイプ(補足1)

ジオメトリエンティティ補足



Body

Multi-Volume Bodyの例

reset

brick x 1 y 1 z 10

Volume all copy move x 2 repeat 4

Volume all copy move y 2 repeat 4

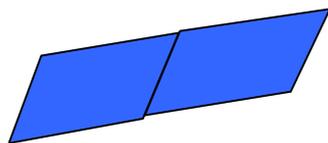
unite body all

ボディ(Body): 

ボリウムとフリーサーフェス(ボリウムに属していないサーフェス)で構成されるエンティティです。

原則として、各ボリウム(またはフリーサーフェス)に対し、ボディが一つずつ自動で定義されます。通常、ボリウムとボディのIDは一致します。

Unite コマンドを利用することで、複数のボリウムまたはフリーサーフェスを1つのボディ(Multi-Volume Body)にまとめることもできます。



Sheet Body

シートボディ(Sheet Body): 

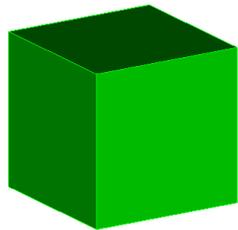
フリーサーフェスのみで構成されるボディはシートボディとも呼ばれます。シートボディとそれに属するフリーサーフェスのIDは必ずしも一致しません。ツリーの Volume List では、厚みのないボリウムとして扱われます。

復習

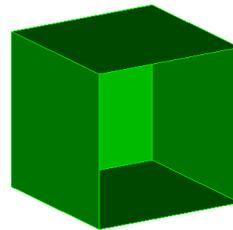
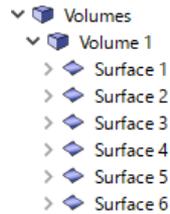
CUBITのエンティティタイプ(補足2)

Sheet BodyとBodyの例

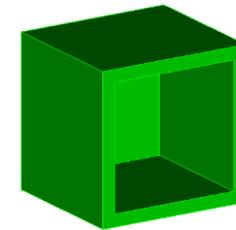
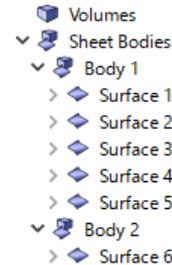
※必要に応じて Refresh Tree を実行してください



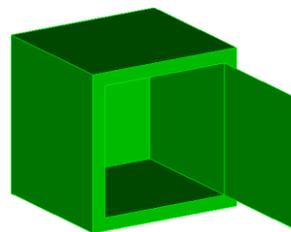
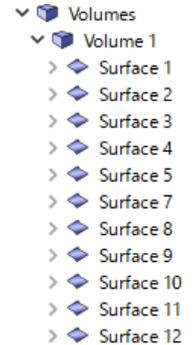
Volume 1



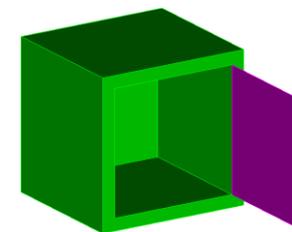
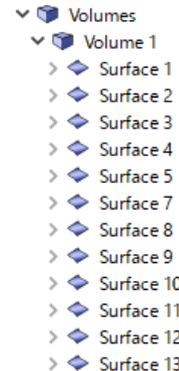
Body 1
(Sheet Body)



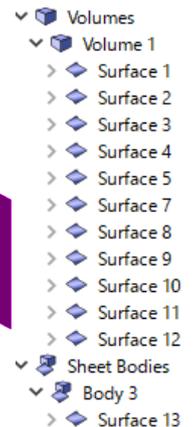
Volume 1



Volume 1
(Body 1)



Volume 1
+ Body 3
(Sheet Body)



```

reset
# Volume (Body) の作成
brick x 10
# Volumeを分解しSheet Bodyを作成します
separate surface 6
delete Surface 6
# Sheet Bodyに厚み(1)をもたせてVolumeにします
thicken volume in surface all depth 1
# multi-volume Body の作成
Surface 8 copy move x 10
unite body all
  
```

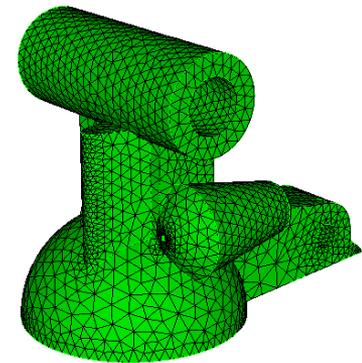
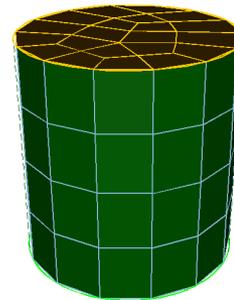
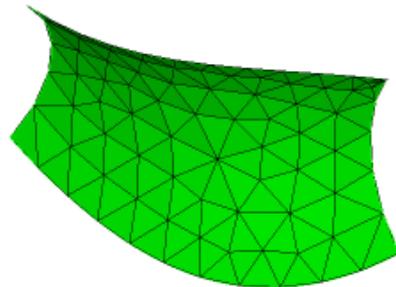
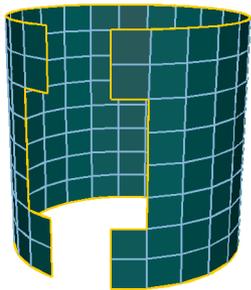
復習

メッシュ(Mesh)

メッシュ(計算格子)

数値解析において、数学モデルを離散化した有限個の部分領域をメッシュ(計算格子)と呼びます。

良い解析結果を得るには、「良いメッシュ」の作成が必要不可欠です。



メッシュ(Mesh)

「良いメッシュ」とは

良い解析の条件:

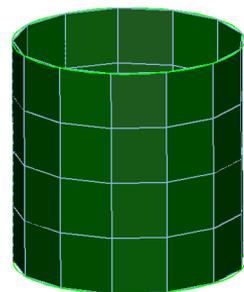
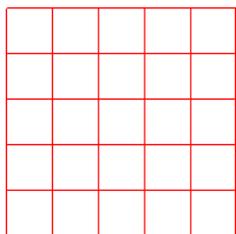
- 目的に応じた解析精度 → メッシュの切り方に依存(ユーザ依存)
- 計算の安定性 → メッシュ品質に依存(メッシャー依存)
- 最小の解析コスト(メッシュ作成・計算時間) → エンジニアに依存

それを実現する「**良いメッシュ**」の条件:

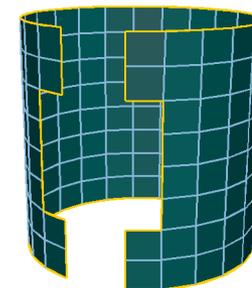
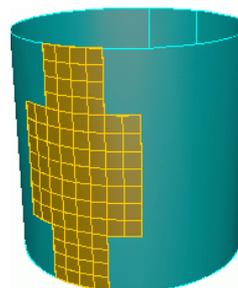
- 適切な要素タイプ(Hex / Tetra / Hybrid)
- 良い要素品質
- 最低限の要素数

サーフェスメッシュ

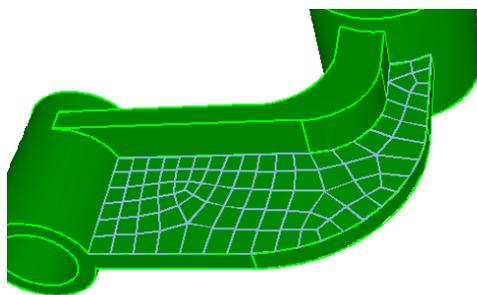
代表的なサーフェスメッシュ



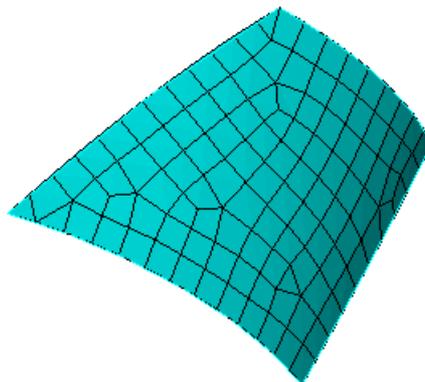
Mapped



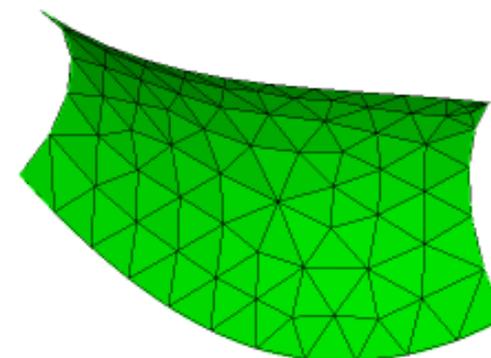
Sub-map



Pave



QuadDominant



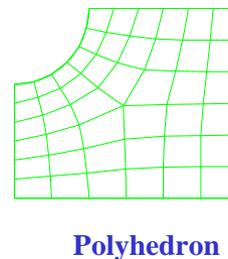
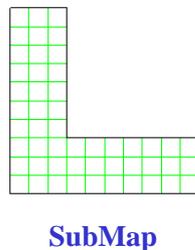
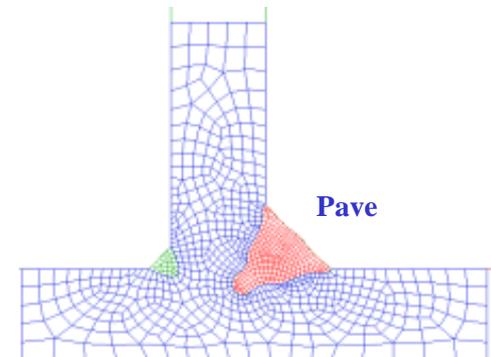
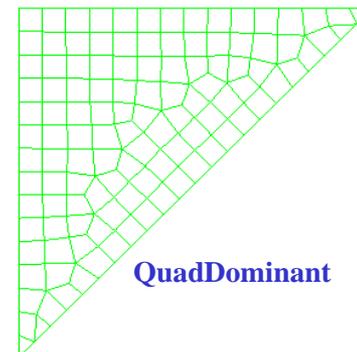
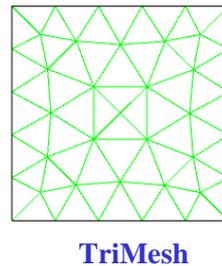
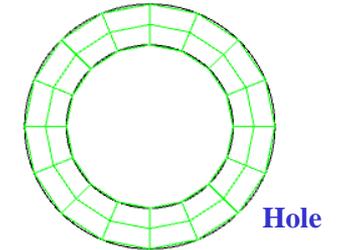
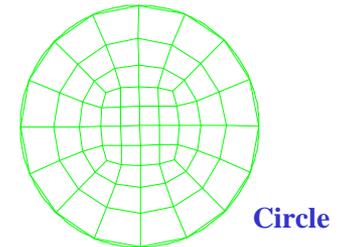
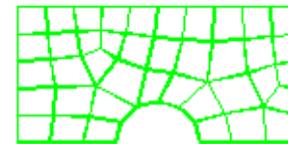
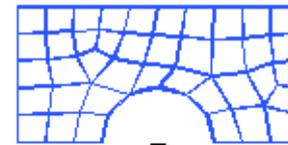
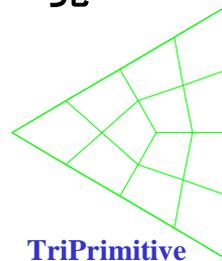
Trimesh

サーフェスメッシュスキーム

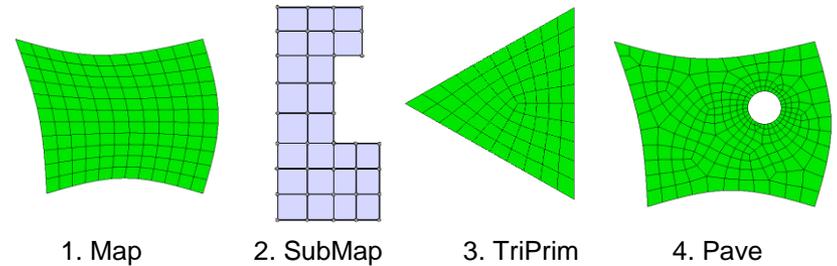
CUBITでは数多くのサーフェスメッシュスキームが利用可能です。サーフェスの形や構成するカーブの数に応じて、最適なメッシュスキームを選択しましょう。また、作成したいボリュームメッシュのタイプによって、サーフェスメッシュのタイプを指定する必要があります。

• サーフェスメッシュスキームの一覧

- Map
- SubMap
- Pave
- QuadDominant
- TriMesh
- TriPrimitive
- Circle
- Hole
- Mirror
- Polyhedron
- TriAdvance
- TriDelaunay



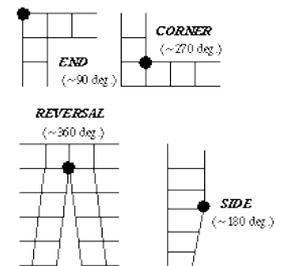
サーフェス自動メッシュスキーム



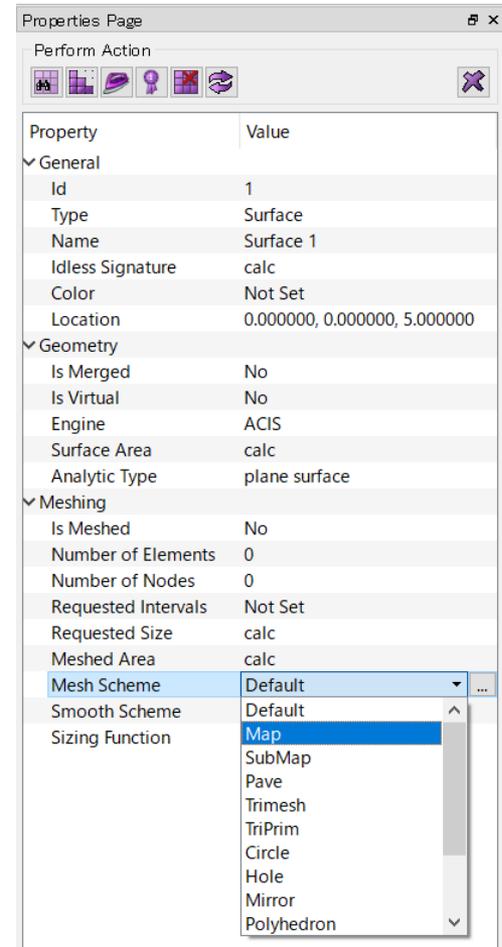
autoスキームでのサーフェスマッシュの優先順位

デフォルトでは、全てのボリュームとサーフェスは**Auto**スキームに設定されています。

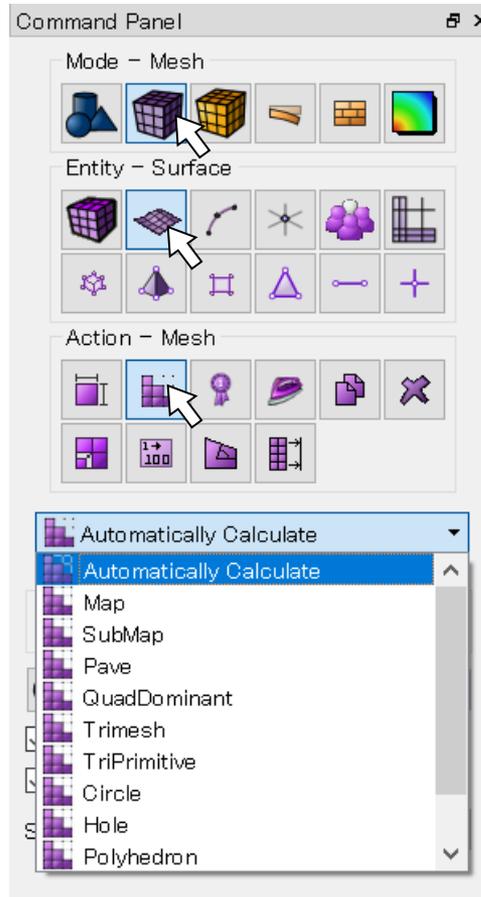
Automatically Calculate を選択する事で、CUBIT はそのサーフェスに最も適したスキームを選択します。スキームは、ジオメトリの特徴に基づいて設定されます。



スキームは、プロパティページやコマンドパネルから手動で設定することもできます。



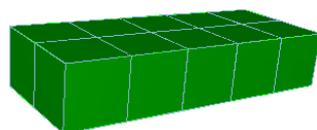
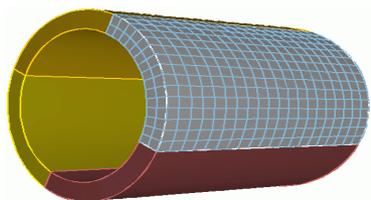
CUBIT プロパティページ



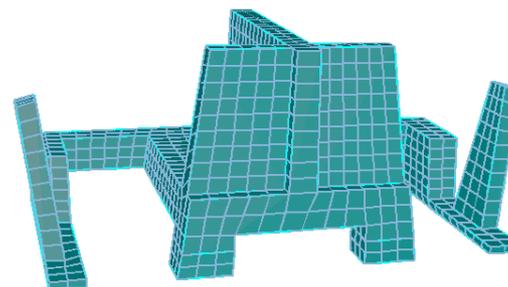
CUBIT サーフェスマッシュ
コマンドパネル

メッシュの種類

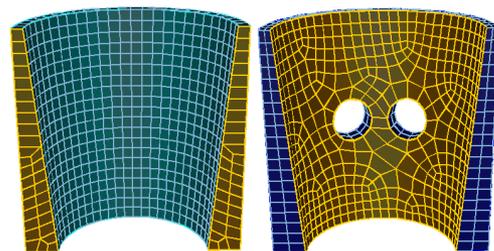
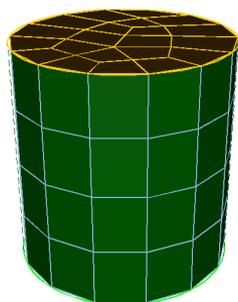
ボリウムメッシュ



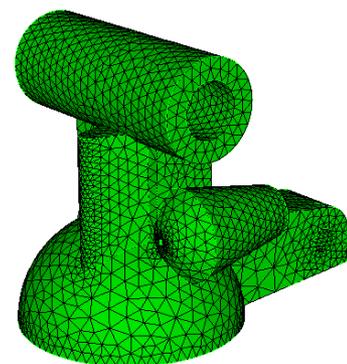
Mapped



Sub-map



Sweep



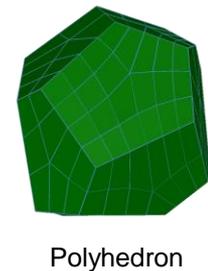
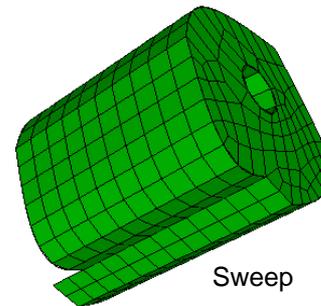
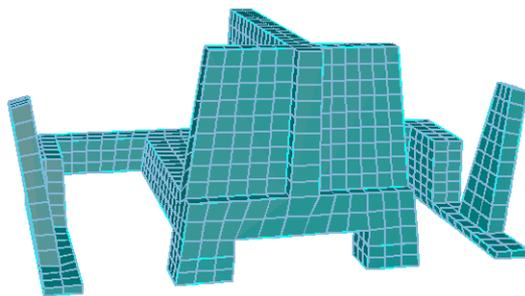
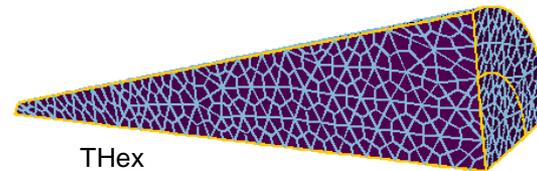
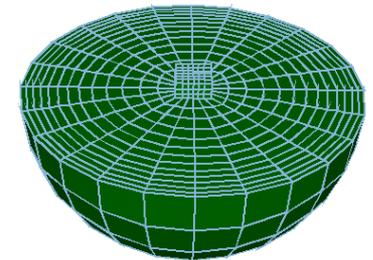
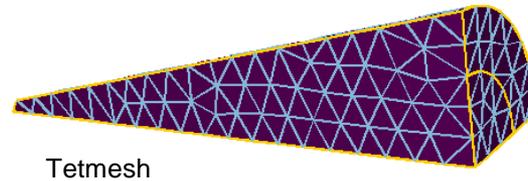
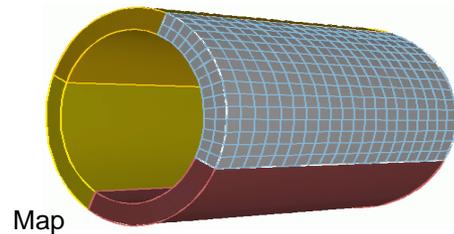
Tetmesh

Hybrid (Hex+Tet+Pyramid+Wedge)

ボリウムメッシュスキーム

CUBITは数多くのボリウムメッシュスキームを提供します。サーフェスマッシュと同様、ボリウムメッシュスキームの選択は、ジオメトリの形状に基づいて行います。

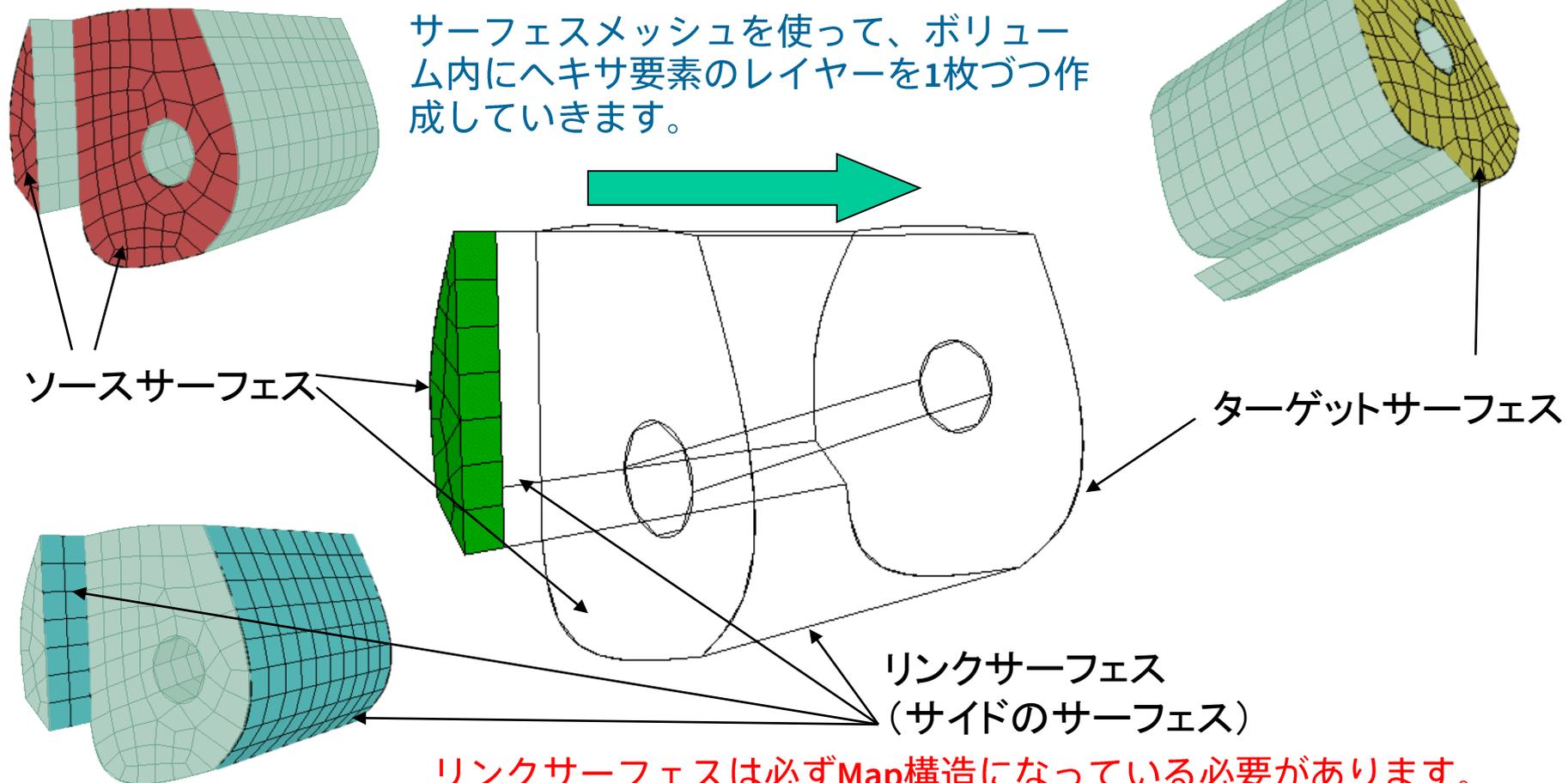
ボリウムメッシュスキームの例



スweepスキーム

最も一般的でよく使われている ボリュームメッシュスキームです。

サーフェスマッシュを使って、ボリューム内にヘキサ要素のレイヤーを1枚ずつ作成していきます。



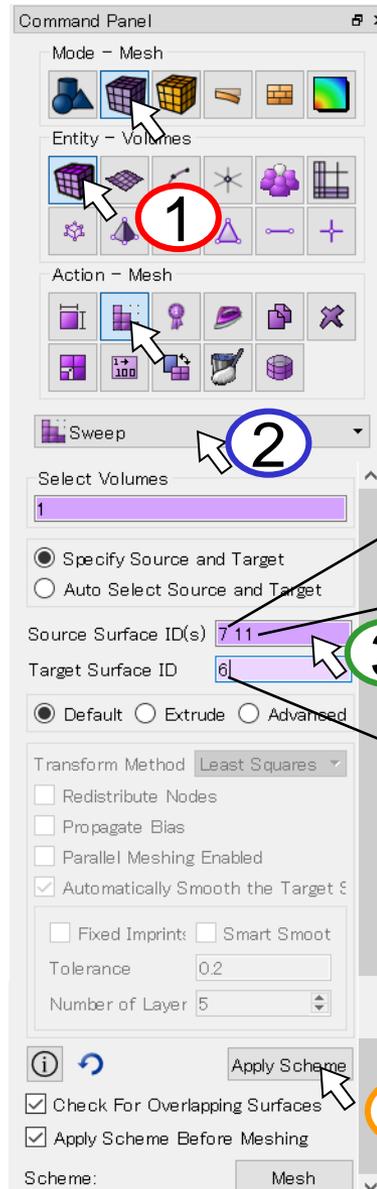
リンクサーフェスは必ずMap構造になっている必要があります。

注意：リンクサーフェスのメッシュスキームがMap以外に設定されている場合は、手動でMapまたはSubMapに切り替えてください。

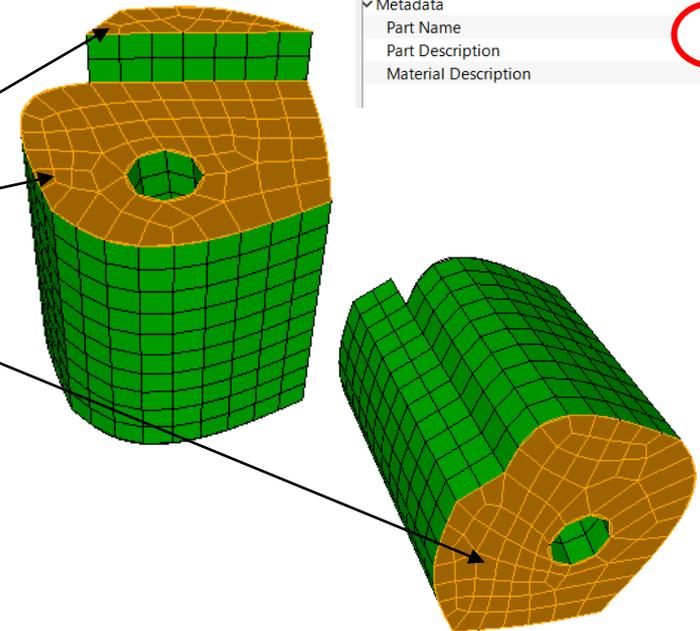
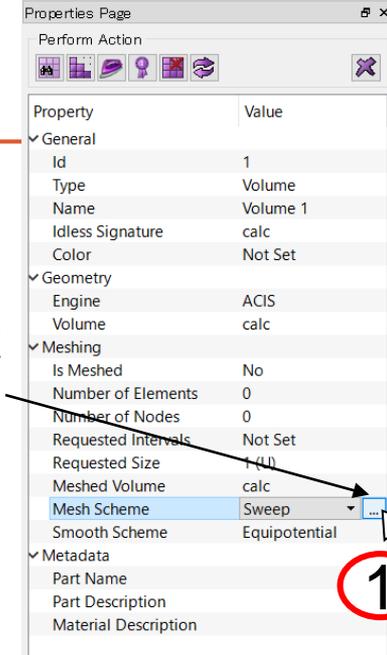
スイープスキーム

ソースサーフェスと
ターゲットサーフェスの設定

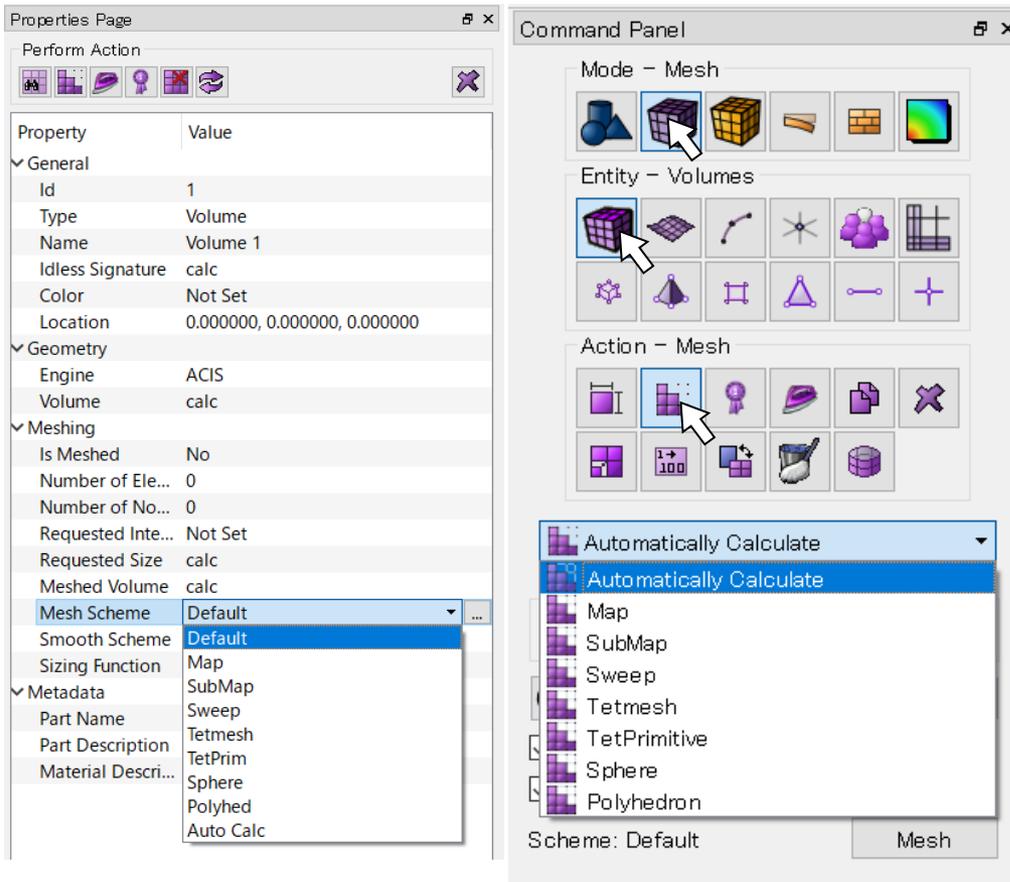
- ① スキーム選択コマンドパネルを選択
- ② Sweep スキームを選択
- ③ ソースとターゲットを選択
- ④ Apply Schemeをクリック



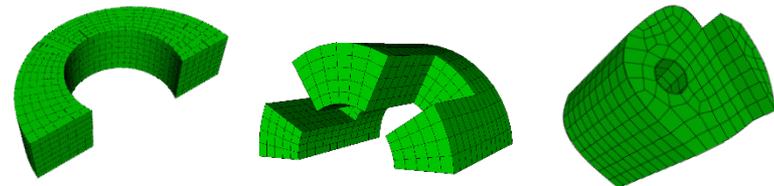
プロパティパネル
からスキーム選択
へのショートカット



ボリューム自動メッシュスキーム選択



CUBIT プロパティページ

CUBIT ボリュームメッシュ
コマンドパネル

1. Map

2. SubMap

3. Sweep

autoスキームでのボリュームメッシュの優先順位

Automatically Calculateを選択することで、Cubitはそのサーフェスに最適なメッシュスキームを選択します。

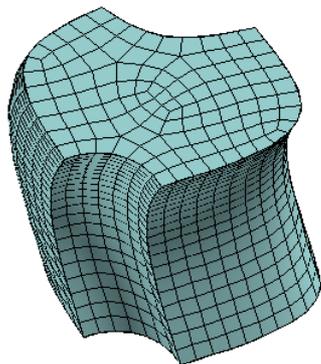
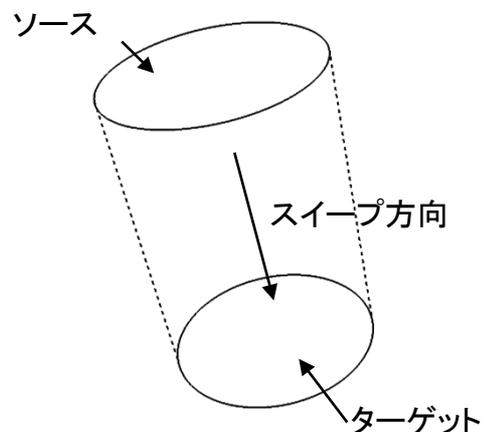
スキームはジオメトリの特徴に基づいて自動的に設定されます。

ボリュームスキームはプロパティページやCUBITサーフェスメッシュコマンドパネルから手動で設定することもできます。

Autoスキームで作成しても、ソース・ターゲットの手動選択が必要になることがあります。

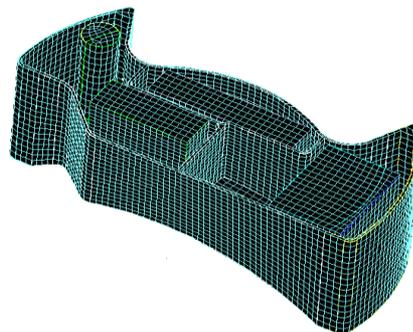
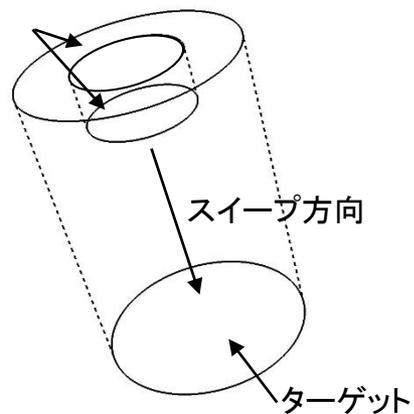
スイープスキーム

One-to-one



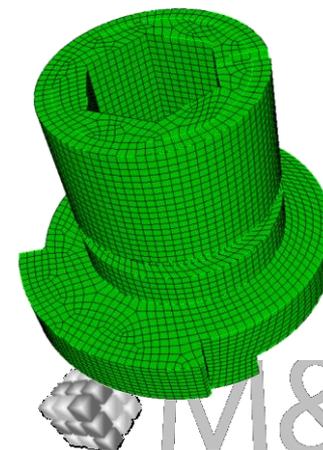
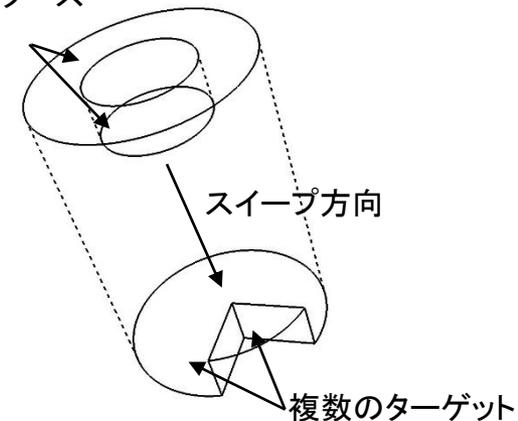
Many-to-one

複数のソース



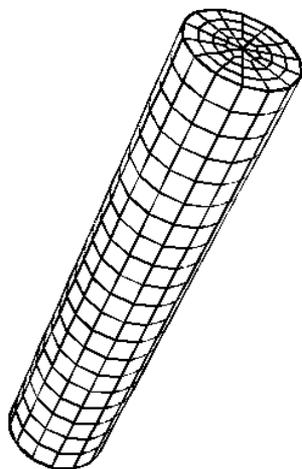
Many-to-many

複数のソース

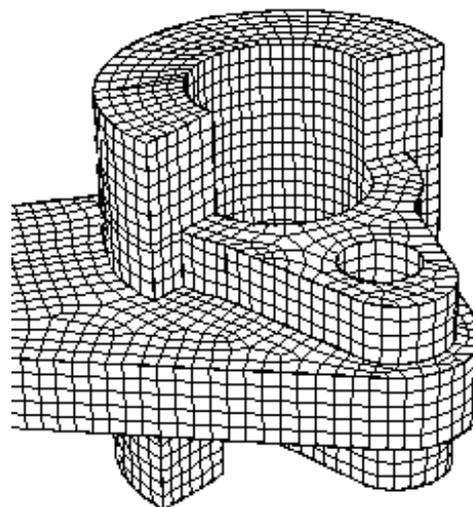


スイープスキーム

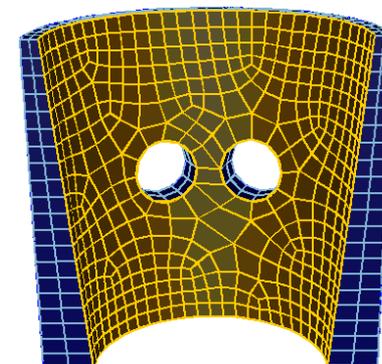
それぞれのメッシュにはどんなタイプのスイープスキームが使われているのでしょうか？



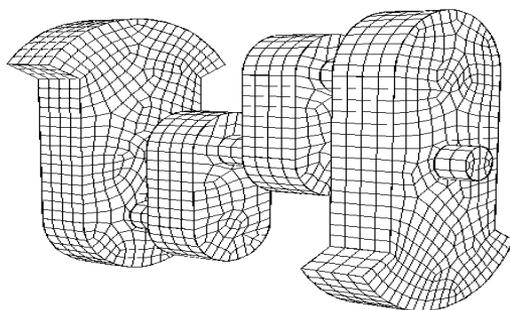
One-to-one



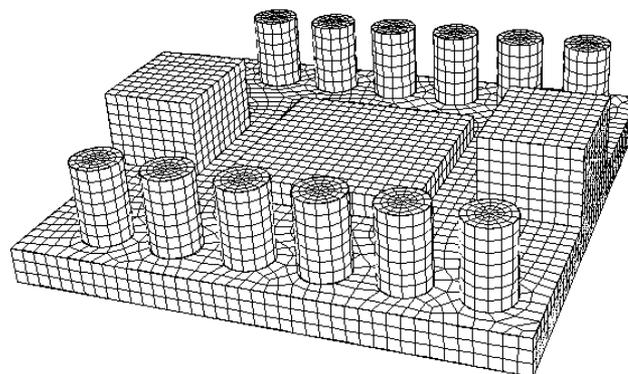
Many-to-many



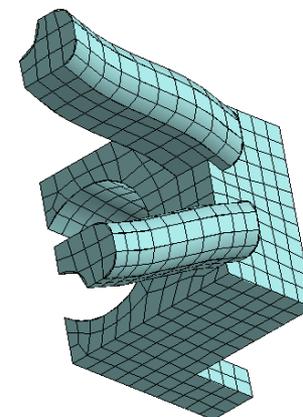
One-to-one



Many-to-many



Many-to-one

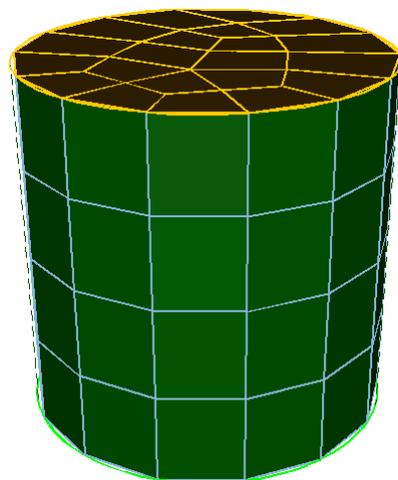


Many-to-many

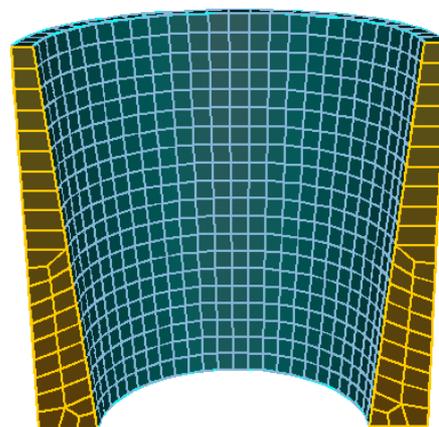
スイープスキーム

典型的なone-to-one スイープ

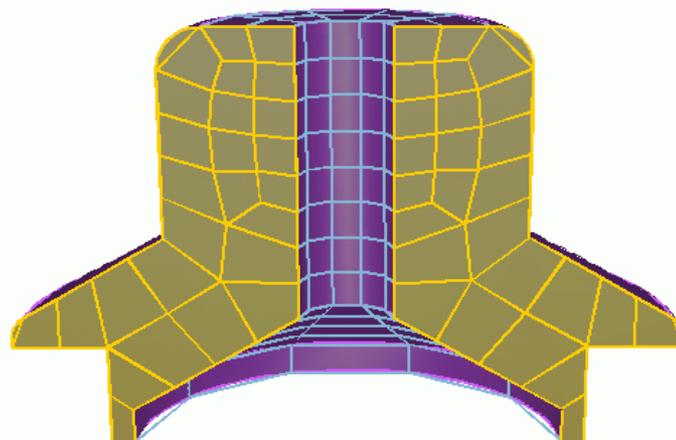
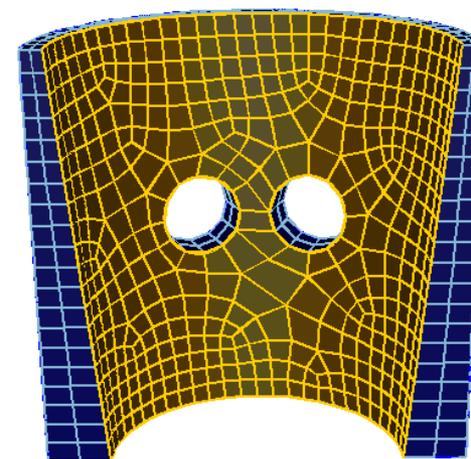
移動



回転



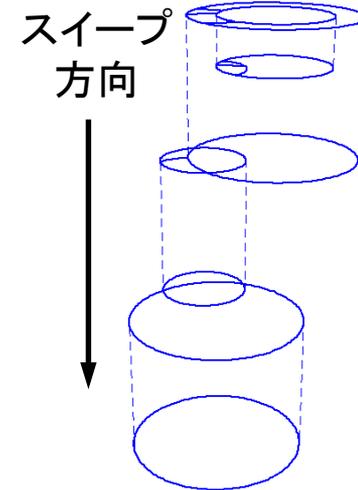
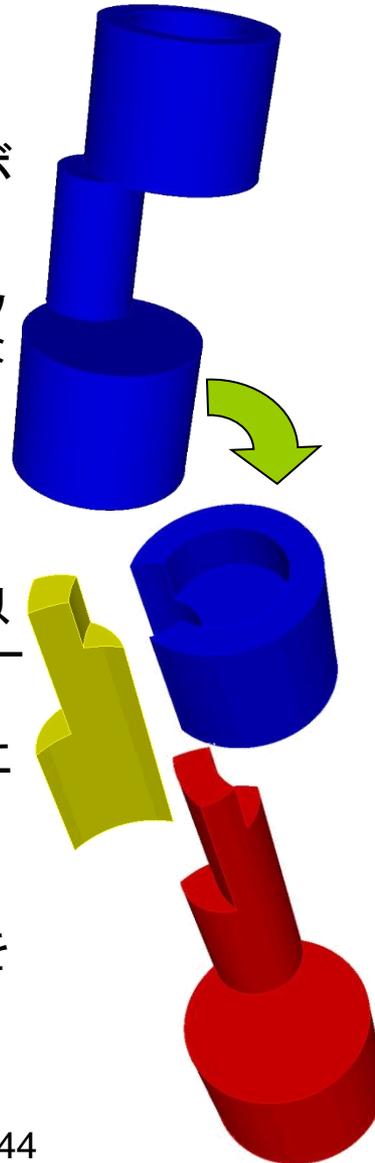
内側から外側へ



Many-to-many スweepの手法

- “multisweeping” (マルチスイープ) として知られています。
- 1つ以上のターゲットサーフェスを含むボリウムをスイープする際の手法です。
 - 自動的にターゲットサーフェスのトポロジーとソースサーフェスを合わせて新しいソースサーフェスを生成し、スイープ可能なボリウムを確保します。
- コマンドを使うには


```
set developer commands on
set multisweep on
```
- スweepするボリウムにおいて、1つ以上のターゲットを指定するとマルチスイープが有効になります
- すでにメッシングされているサーフェスに影響されます。
 - マルチスイープボリウムのメッシングを最初に行いましょう。
- 可能なら1つのターゲットへのスイープを実行します。



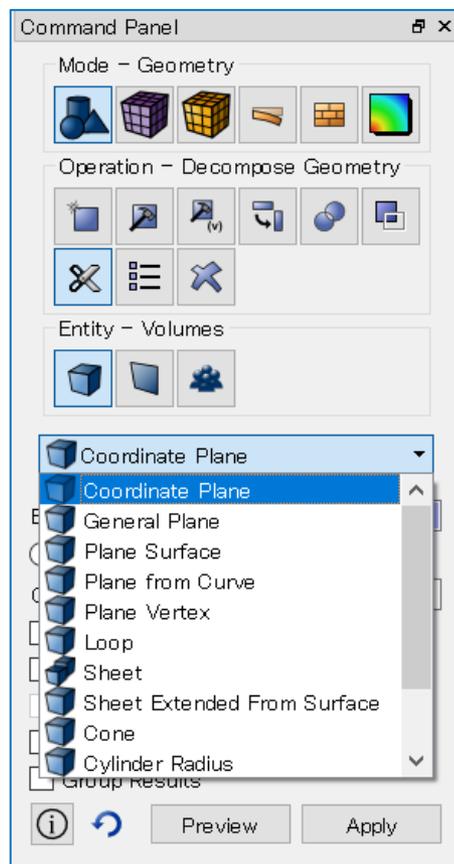
マルチスイープは自動で内部分解を行い、複数のMany-to-oneタイプのボリウムを作成します。

練習

```
reset
set dev on
set multisweep on
create Cylinder height 2 radius 1
volume 1 copy move x 1 y 1 z 1
Unite vol all
mesh vol all
```

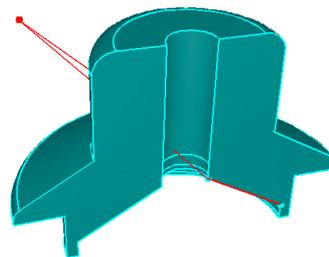
Decomposition (分解) ツール

Cubitは、様々な分解ツールを備えています。

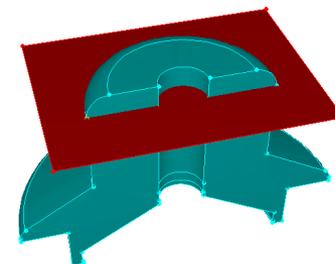


Webcut コマンドパネル

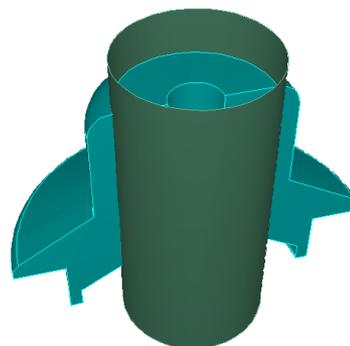
バーテックスで定義
(Plane From Vertices)



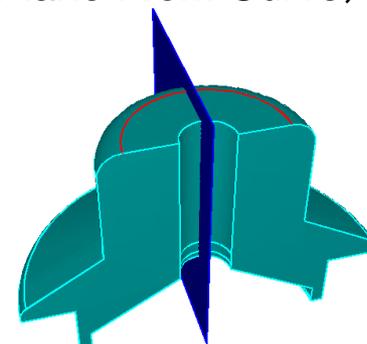
座標平面
(Coordinate Plane)



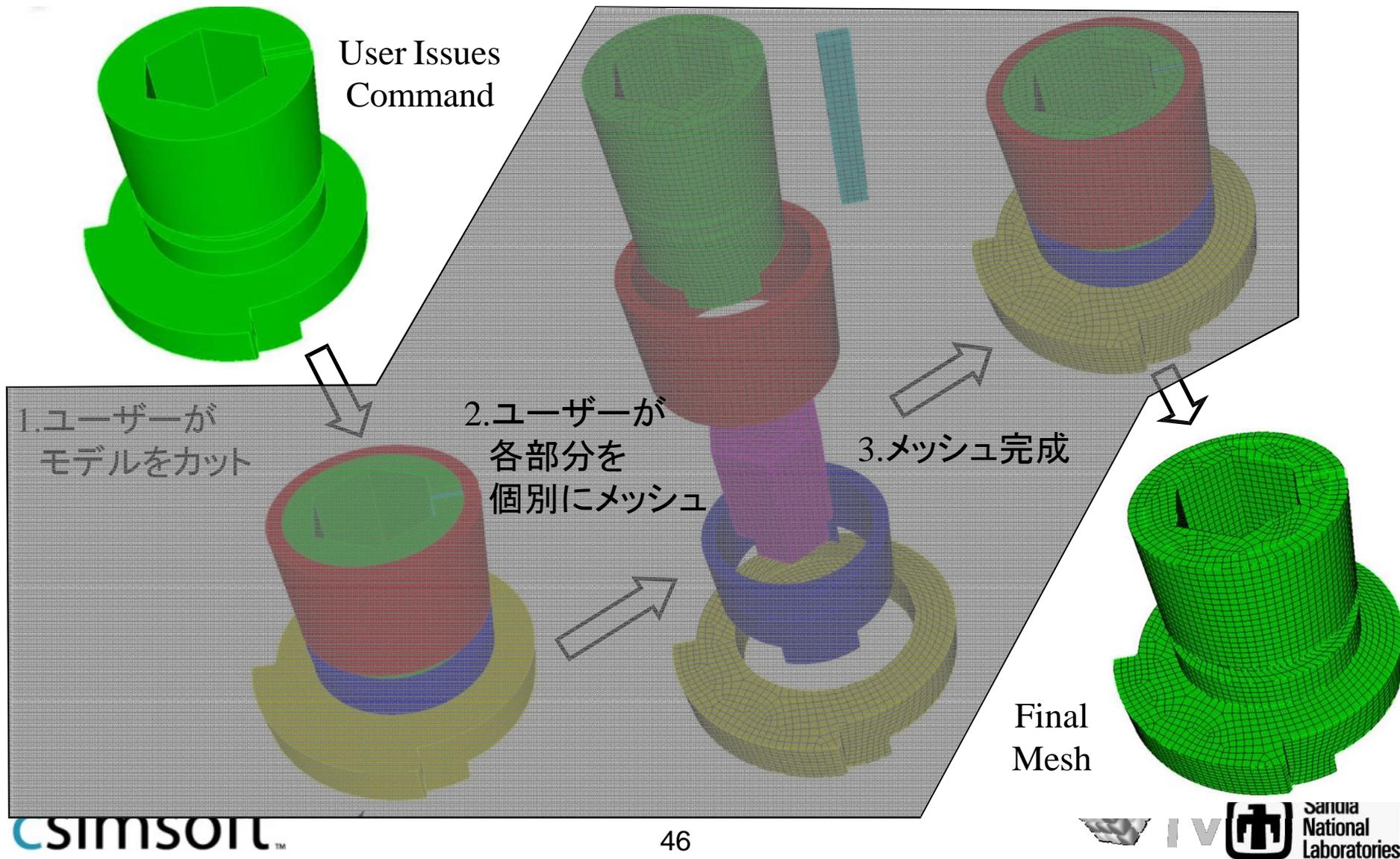
円筒面
(Cylinder)



カーブに垂直な平面
(Plane From Curve)

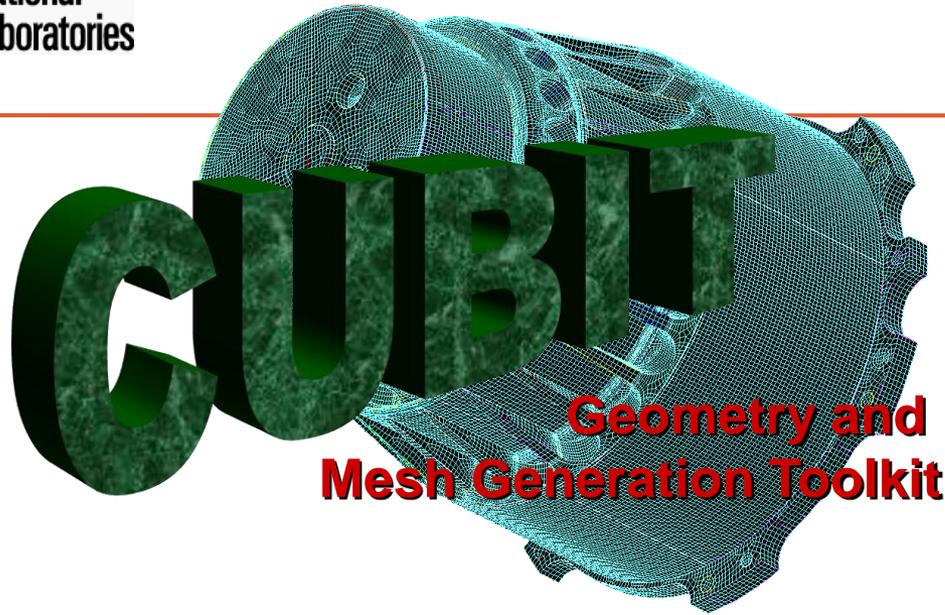


Many-to-Many スイープ: Reducing User Input for Decomposition



最重要

- Cubitは、ユーザがイメージするヘキサメッシングを行うための **ツール**を提供します。
 - メッシングスキーム
 - サーフェス: Map, Submap, Pave, etc.
 - ボリューム: Map, Submap, Sweep, etc.
 - ジオメトリ分解ツール
 - Webcut
- **ユーザの主な仕事は、**
ユーザがイメージするヘキサメッシングを実現するために、**解析対象のモデル**(一般的にはMany-to-Manyのボリューム)をMany-to-Manyから**Many-to-One/One-to-One**に分解することです。



CUBIT ファストスタートチュートリアル ゲッティングスタート

**ゲッティングスタートエクササイズ
概要を説明致します。
その後、自習してください**

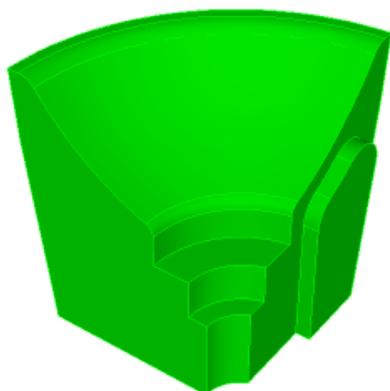
このチュートリアルで学ぶ機能

- **CADファイルの読み込み**
 - ヒーリングオプション
- **プロパティページの使い方**
 - プロパティの編集
 - メッシュの作成
 - メッシュの削除
 - プロパティの初期化
- **メッシュ作成**

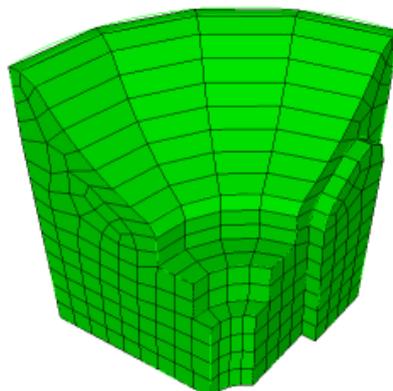
課題:

プロパティのリセットとメッシュ削除の違いを理解しましょう

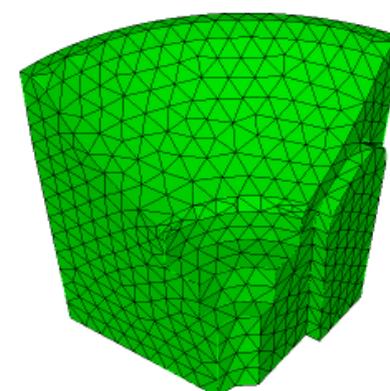
ゲッティングスタート エクササイズ



ソリッドモデル



ヘキサメッシュ



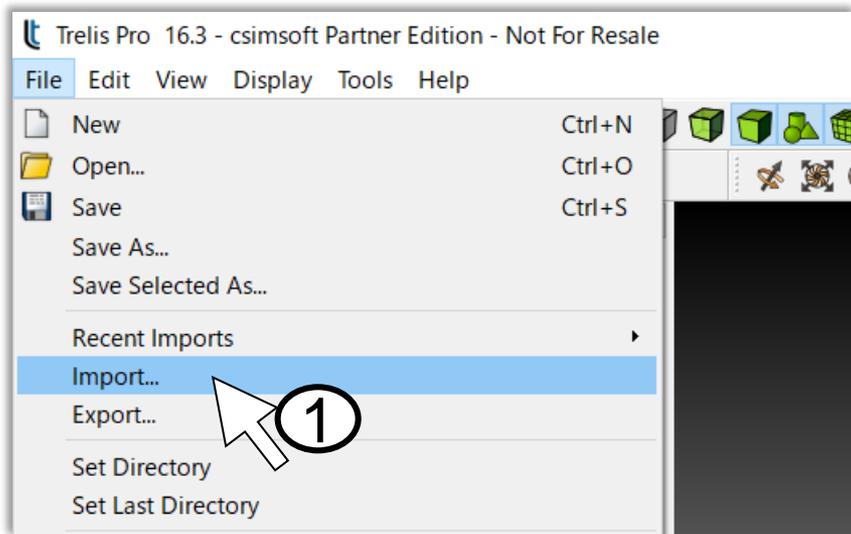
テトラメッシュ

- ・プロパティの編集
- ・mesh vol all

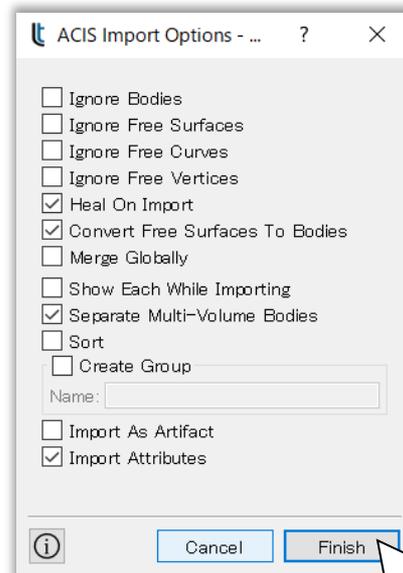
1. ファイルからジオメトリをインポートします。
2. メッシュサイズを0.5に設定します。
3. メッシュスキームをsweepに設定します。
4. ヘキサメッシュを生成します。
5. メッシュを**消去**します。
6. メッシュスキームをtetmeshに設定します。
7. テトラメッシュを生成します。

- ・プロパティの編集
- ・mesh vol all

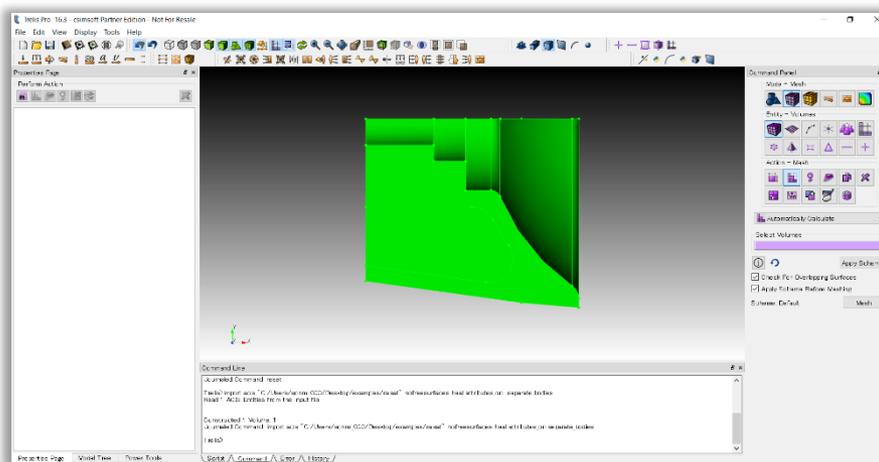
ソリッドモデルのインポート



① File メニューからImport を選択して、“ra.sat”をインポートします。

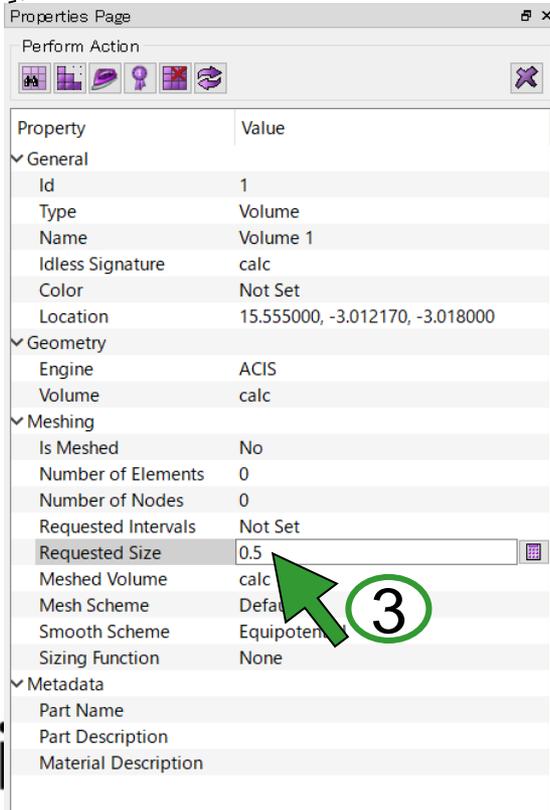
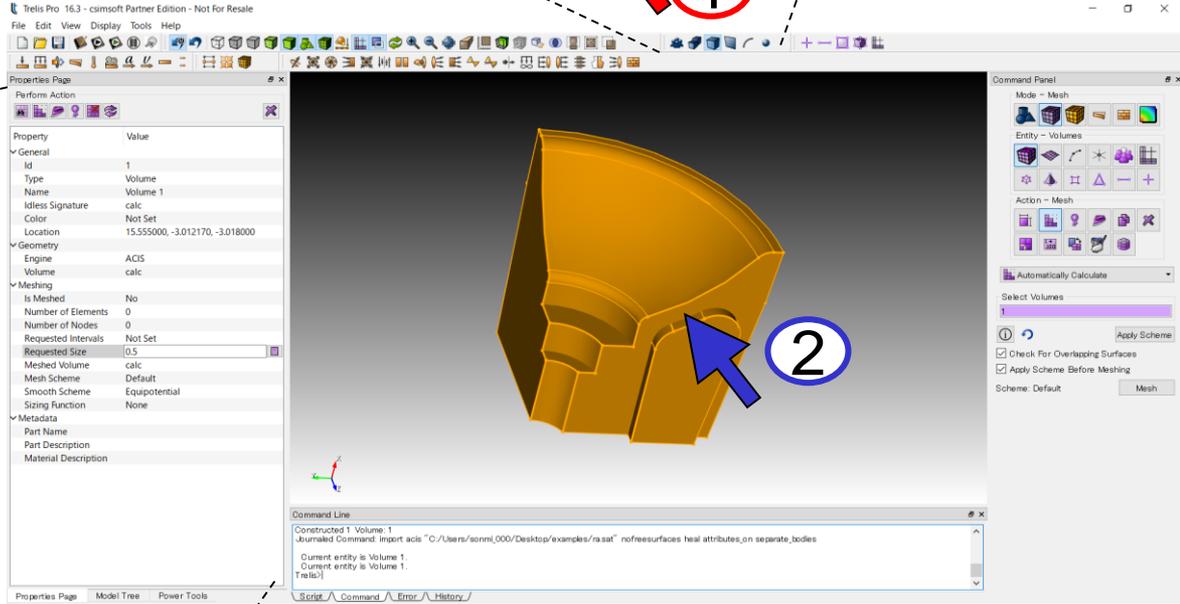
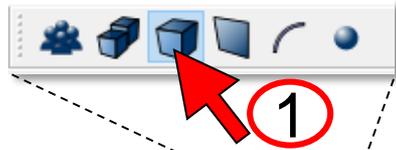


② Finishをクリックします。



メッシュサイズの設定

- ① ボリューム選択フィルターのアイコンをクリックします。
- ② パーツをダブルクリックします。



- ③ プロパティパネル上で Requested Size を0.5に設定して、Enterキーを押します。

- ・プロパティの編集
- ・mesh vol all

ヘキサメッシュの生成

① Mesh Scheme のドロップダウンメニューから Sweep スキームを選択します。

② メッシュボタン  を押します。

- ・プロパティの編集
- ・mesh vol all

テトラメッシュの生成

The screenshot shows the CUBIT software interface. On the left, the 'Properties Page' for 'Volume 1' is displayed. It has three numbered annotations: a red circle '1' pointing to the 'Reset' button in the 'Perform Action' toolbar; a blue circle '2' pointing to the 'Tetmesh' option in the 'Mesh Scheme' dropdown menu; and a green circle '3' pointing to the 'Mesh' button in the 'Perform Action' toolbar. On the right, the main 3D view shows a green tetrahedral mesh on a curved surface. Below the 3D view, the 'Command Line' shows the command: 'Volume 1 (Volume 1) meshing completed using scheme: tetmesh'. At the bottom, a red-bordered box contains the command sequence: '##コマンド練習##', 'reset vol all', 'vol all size 0.1', and 'mesh vol all'.

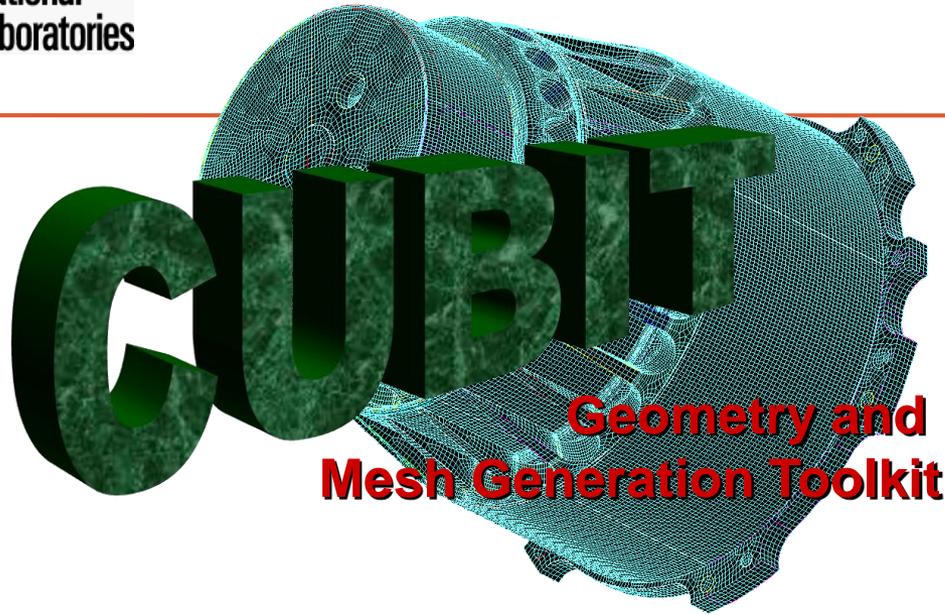
- ① リセットボタンを押して、メッシュの消去とボリユームのリセットを行います。
- ② Mesh Schemeのドロップダウンメニューから *Tetmesh* スキームを選択します。
- ③ メッシュボタンをクリックします。

```
##コマンド練習##
reset vol all
vol all size 0.1
mesh vol all
```

このチュートリアルで学んだ機能

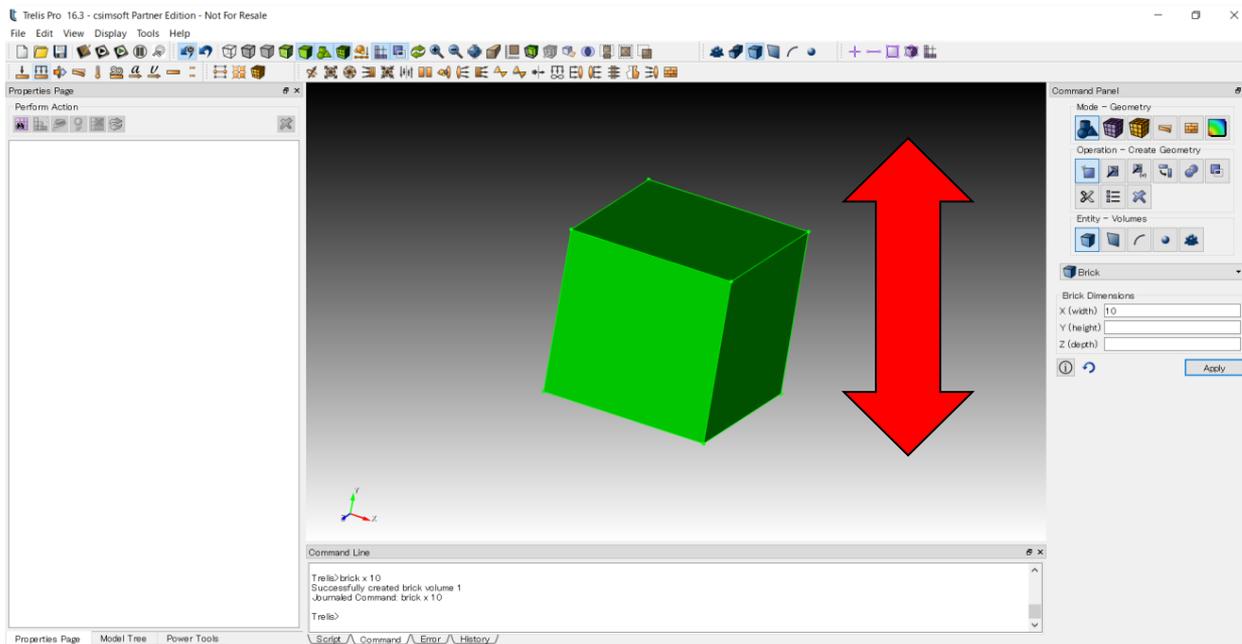
- エンティティのプロパティのリセット
 - メッシュ情報の削除
 - プロパティ設定 (メッシュスキームなど) の初期化
 - 下位のエンティティのプロパティ設定も初期化されます
- メッシュ削除
 - メッシュ情報の削除
 - プロパティ設定はそのまま
 - 下位のエンティティのプロパティ設定もそのまま
 - **メッシュコマンドを再度実行すると、同じメッシュができる**
- Volume (上位) と Surface (下位) のプロパティの連動
 - Default, Map, Submap, Sweep は下位を変更しない
 - Tetmesh は下位 Surface の Scheme を Trimesh に変更

スキーム設定等でエラーが起きた場合は、プロパティのリセットによってスキームを初期化することをお勧めします。

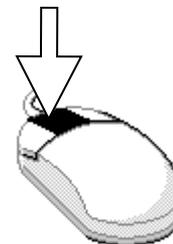


CUBIT チュートリアル マウス操作

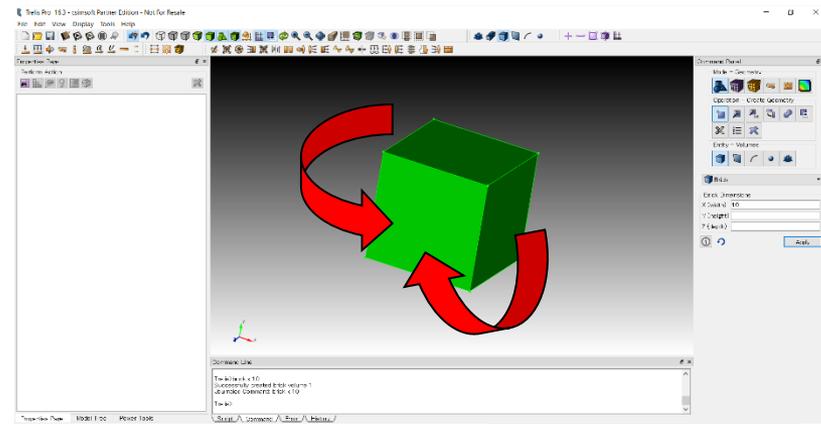
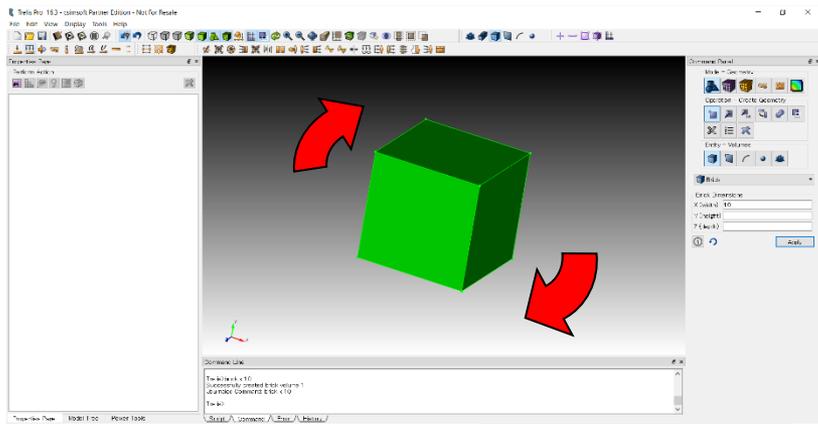
グラフィックウィンドウ ズーム (Zoom)



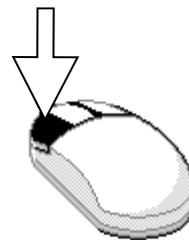
ズームをするには、マウスをグラフィックウィンドウに持ってゆき、マウスのミドルボタンをスクロールします。



グラフィックウィンドウ 回転 (Rotate)

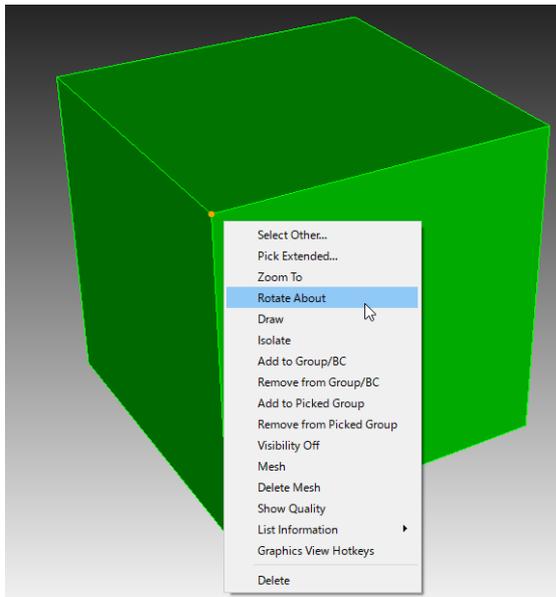


モデルをスクリーンに対して軸対象に回転させるには、マウスをグラフィックウィンドウの角の近くに持ってゆき、マウスの左ボタンを押してポインターをウィンドウの角に沿って回すように動かします。

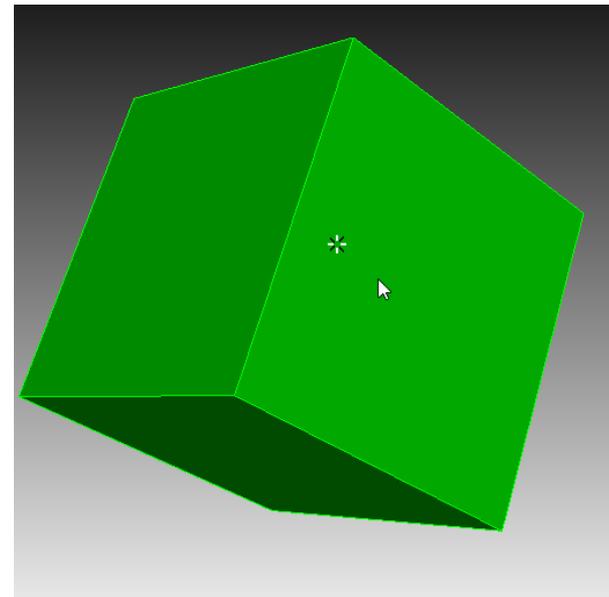


モデルを中央で回転させるようにするには、マウスの左ボタンを押しながら、グラフィックウィンドウの中央付近で動かします。

グラフィックウィンドウ 回転中心の切り替え

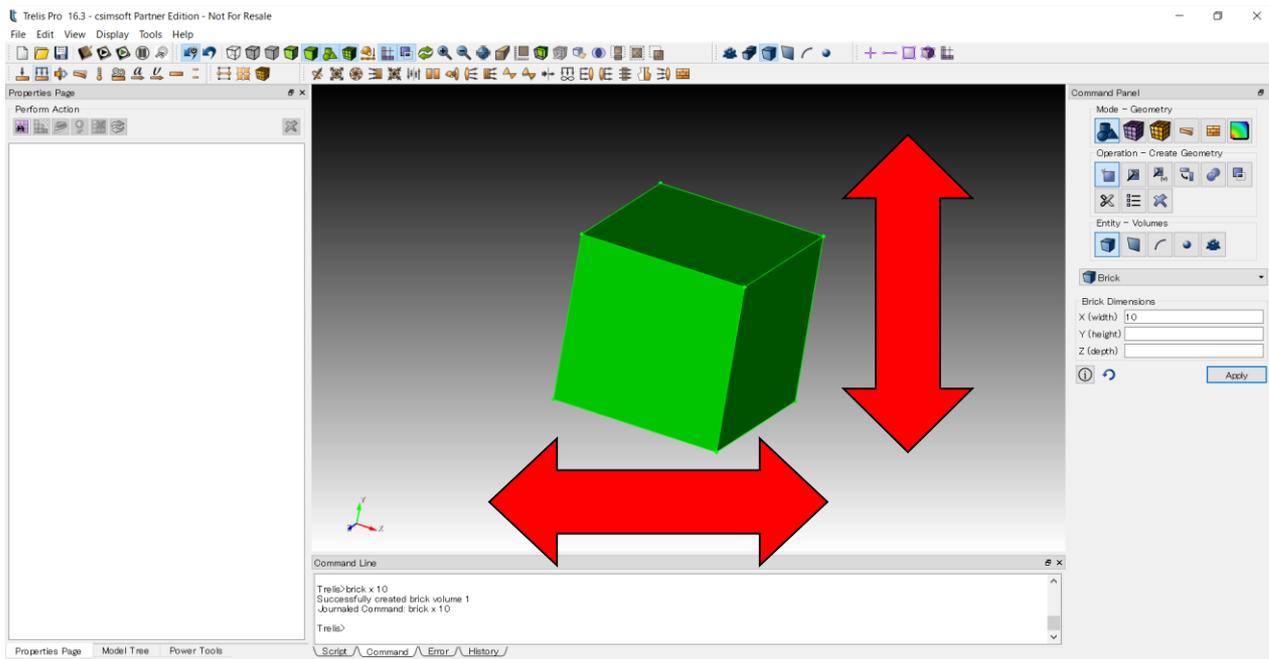


任意のエンティティの重心まわりに回転するには、エンティティを選択後、右クリックしてライトクリックコマンドパネルを開き、Rotate About を選択します。

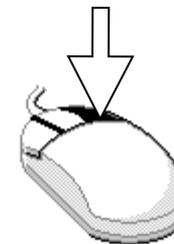


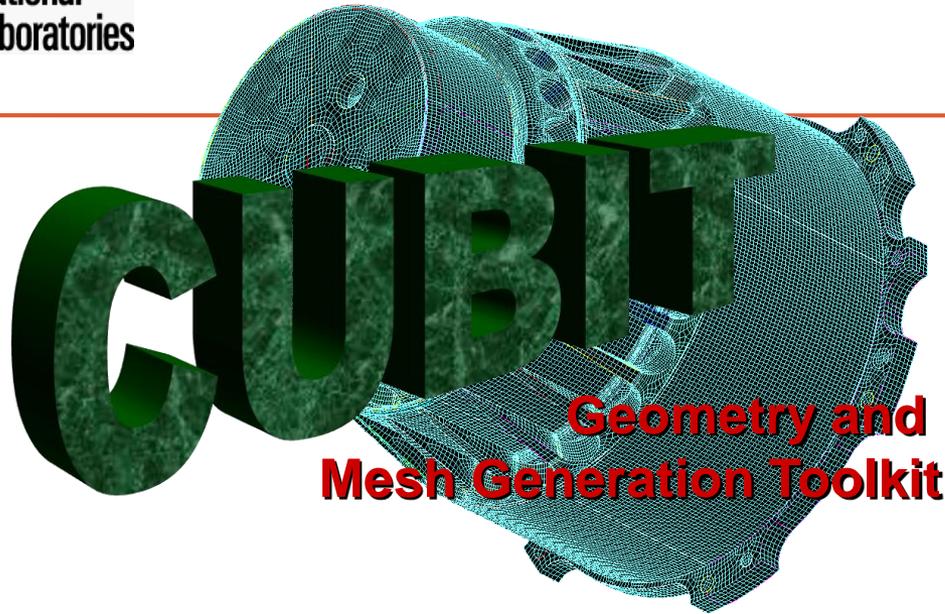
エンティティ上の任意の点まわりに回転するには、エンティティ上の任意の点をShift キーを押しながらクリックし、ドラッグします。

グラフィックウィンドウ パン(Pan)



パンするには、グラフィックウィンドウでマウスの右ボタンを押しながら、上下・左右に動かします。





**CUBIT ファーストスタートチュートリアル
ユーザーインターフェースの基礎**

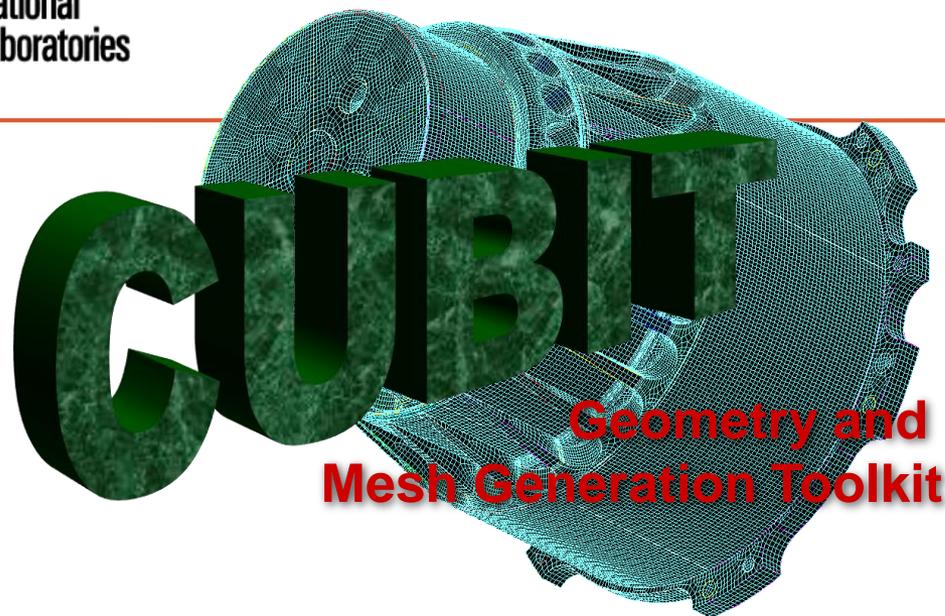
- ・どんな機能が、どこにあるのか
- ・自分のやりたいことをするために、何を調べればよいのか

GUIの構成を説明します。 詳細は参考資料を御覧ください。

独学で全機能を把握するのは
時間がかかります。

とても重要な説明ですので集中して聞いてください。

どんな機能がどこにあるのか？
ほんの少し覚えてください。



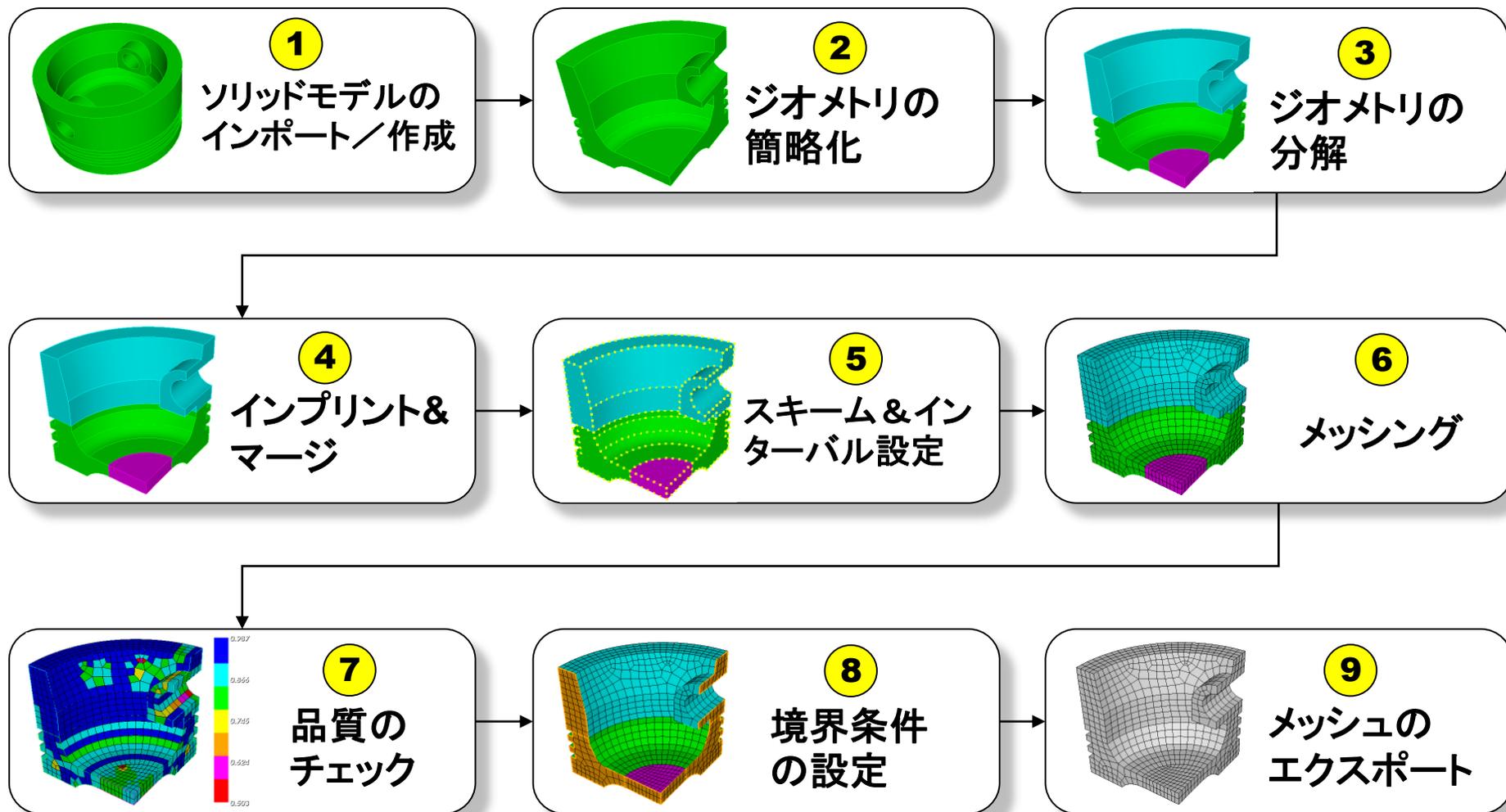
CUBIT ファーストスタートチュートリアル
CUBITの基本的な作業工程

**基本的なCubitの流れ
概要を説明致します。
その後、自習してください**

このチュートリアルで使う機能

- ジオメトリの簡略化
 - Webcut
 - 座標面で分割
 - Delete
 - 不要なジオメトリを削除
- ジオメトリの分割
 - Webcut
 - シリンダーで分割
 - サーフェスを延長して分割
- インプリント／マージ
- メッシュ作成
 - サイズとスキーム設定
- 条件設定
- メッシュ出力

Cubitの基本的なメッシングプロセス



- Webcut
- プロパティの編集
- mesh vol all



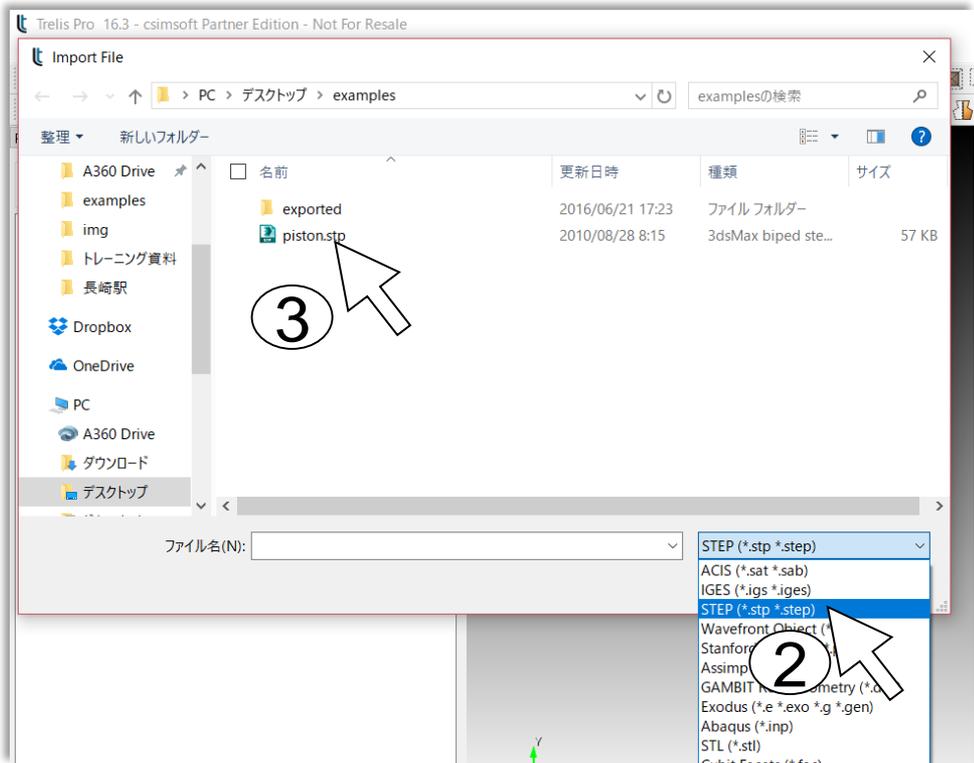
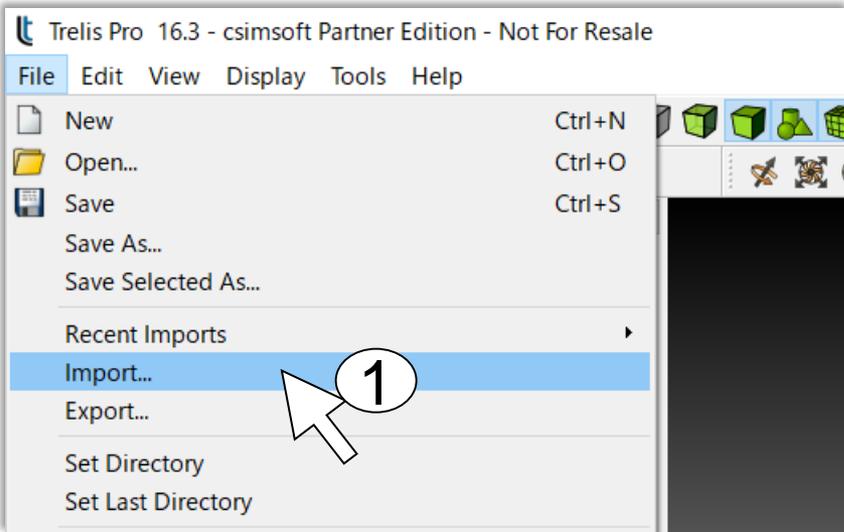
1

ソリッドモデルの インポート／作成

STEP file “piston.stp”
をインポートします。

① メニューの *File* → *Import* を選択

③ “piston.stp” を選択



② ファイルフィルタータイプ
を STEP (*.stp) に設定

- Webcut
- プロパティの編集
- mesh vol all

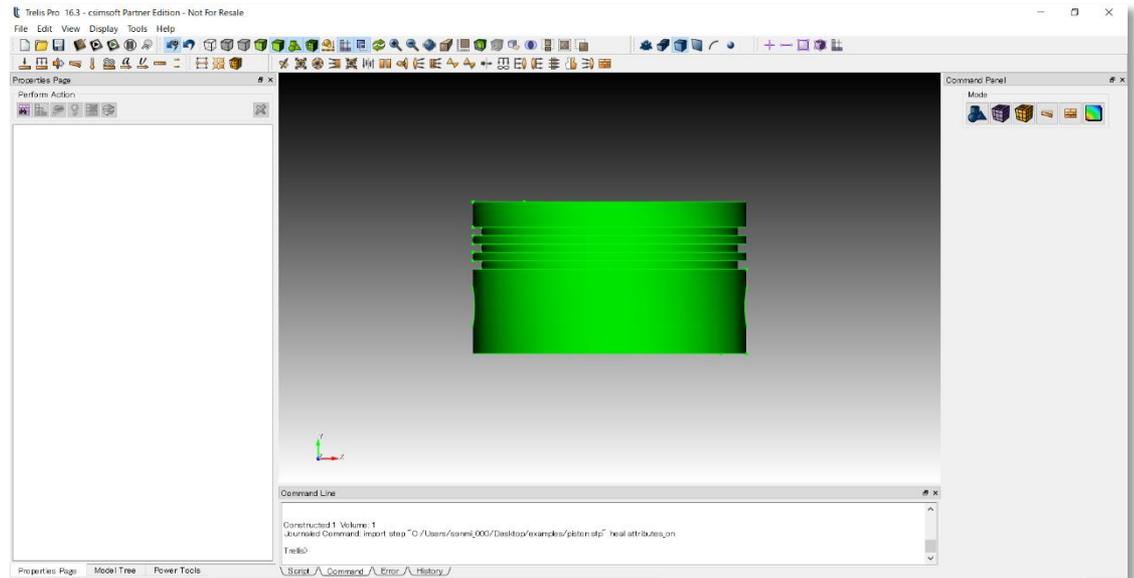
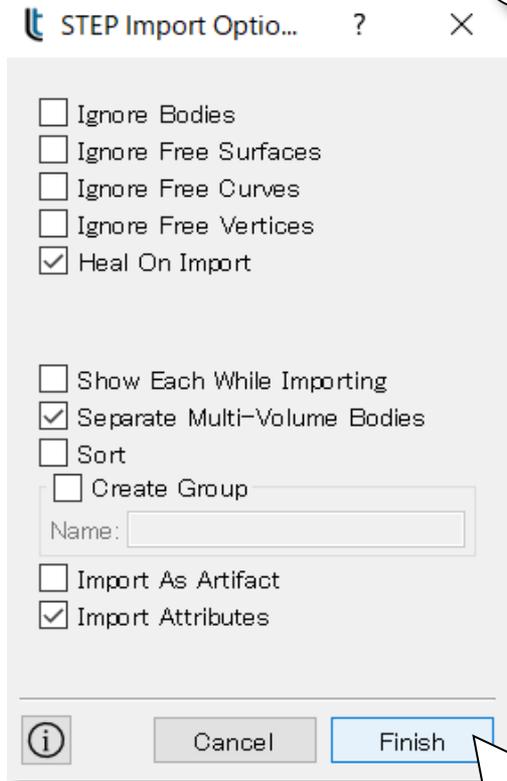
CUBIT テキスト



1

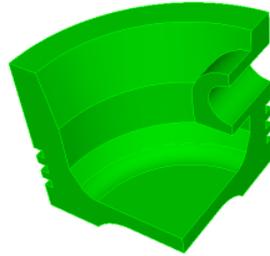
ソリッドモデルの インポート／作成

STEP file “piston.stp”
をインポートします。



4 Finish をクリック

- ・Webcut
- ・プロパティの編集
- ・mesh vol all



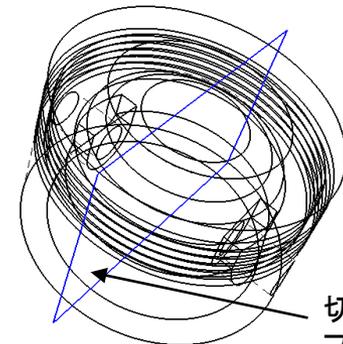
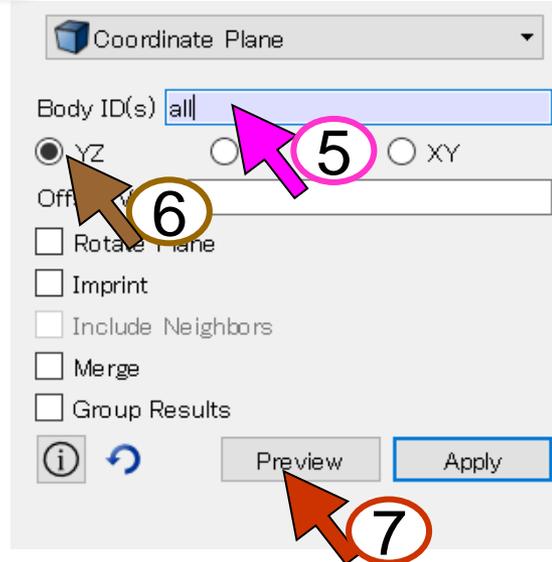
2

ジオメトリ の簡略化

図形の対称性を利用
するために、モデルを
四等分します。

パーツの切断を設定し、
プレビューを見ます。

- ① Mode → Geometry を
クリック
- ② Operation → Decompose
を選択
- ③ Entity → Volumes をクリック
- ④ ドロップダウンメニューから
Coordinate Plane を選択
- ⑤ ボリュームを選択 (また
はID 欄にall と入力)
- ⑥ YZ Plane を選択
- ⑦ Preview をクリックして切
断面のプレビューを表示



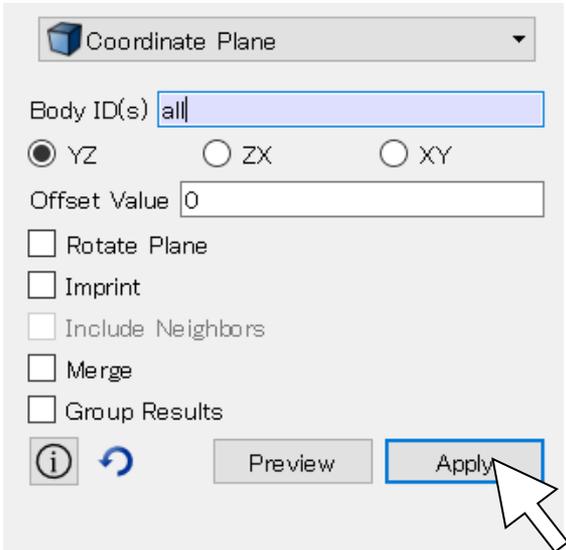
切断面の
プレビュー

- Webcut
- プロパティの編集
- mesh vol all

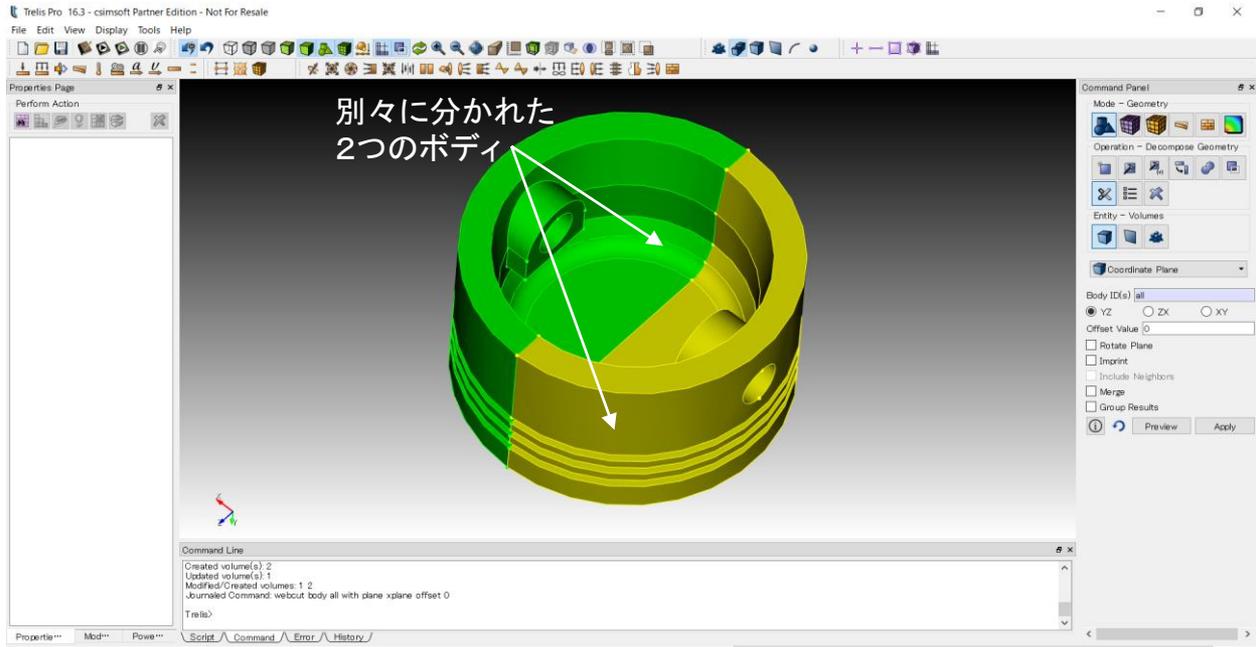


ジオメトリ の簡略化

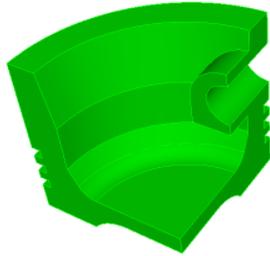
図形の対称性を利用するために、モデルを四等分します。



⑧ Apply を押して webcut コマンドを実行



- ・Webcut
- ・プロパティの編集
- ・mesh vol all

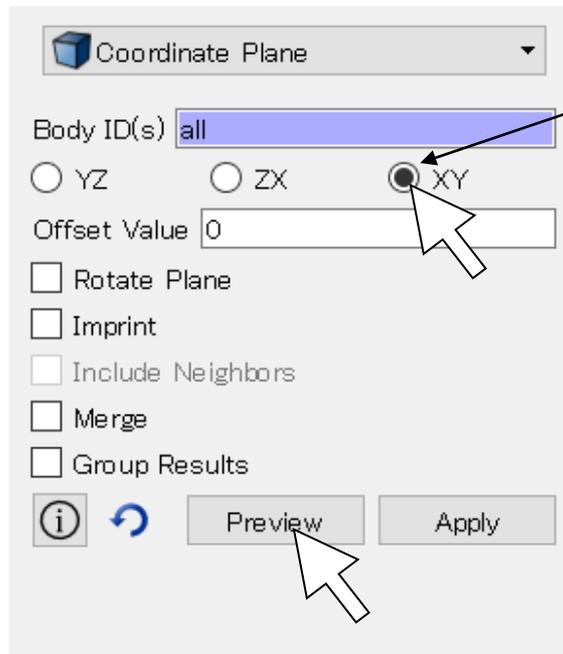


2

ジオメトリ の簡略化

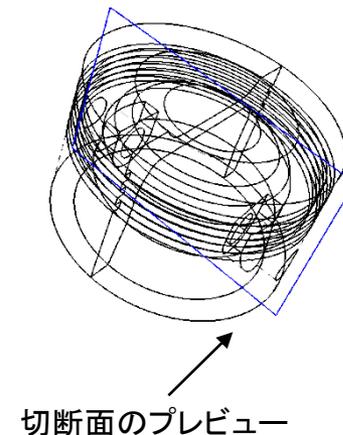
図形の対称性を利用するために、モデルを四等分します。

モデルへの2回目のwebcutを設定してプレビューをみます。



⑨ 切断面を *XY Plane* に変更

⑩ 切断面のプレビューを表示

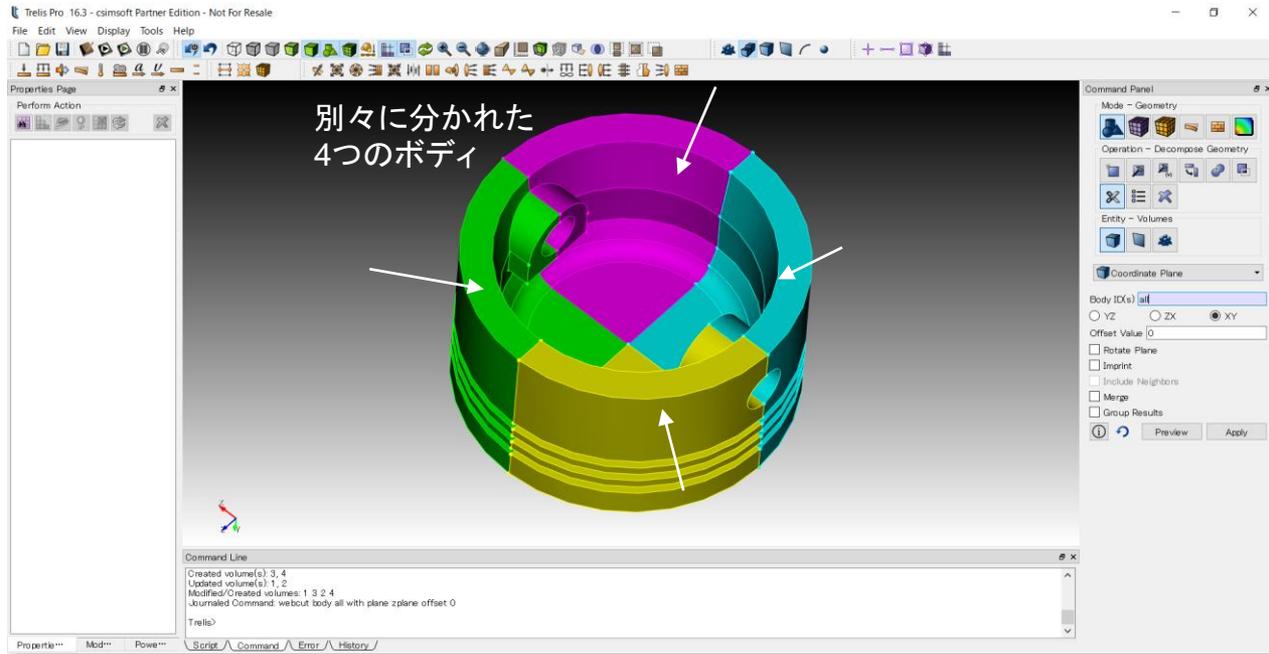
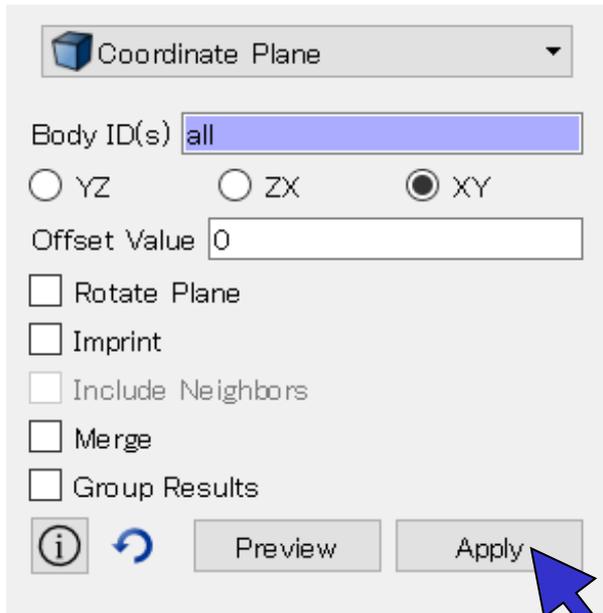


- Webcut
- プロパティの編集
- mesh vol all

CUBIT テキスト

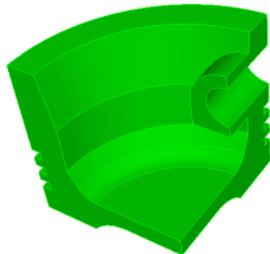


図形の対称性を利用するために、モデルを四等分します。



⑪ Apply を押して webcut を実行

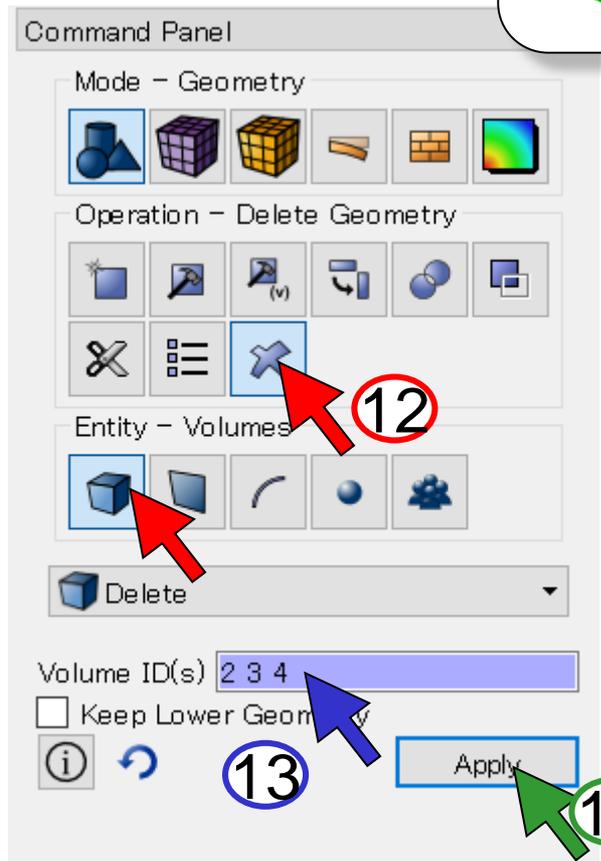
- Webcut
- プロパティの編集
- mesh vol all



2

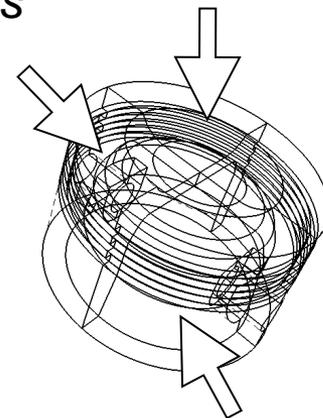
ジオメトリ の簡略化

図形の対称性を利用するために、モデルを四等分します。

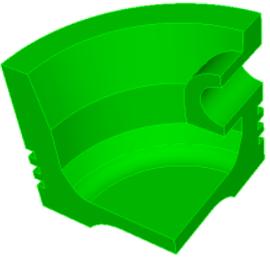


4つのボリユームのうち3つを消去します。

- ⑫ *Operation* → *Delete* メニューを選択し、*Entity* → *Volumes* を指定
- ⑬ グラフィックウィンドウで4つのうち3つのボリユームを選択
- ⑭ *Apply* を押して消去コマンドを実行



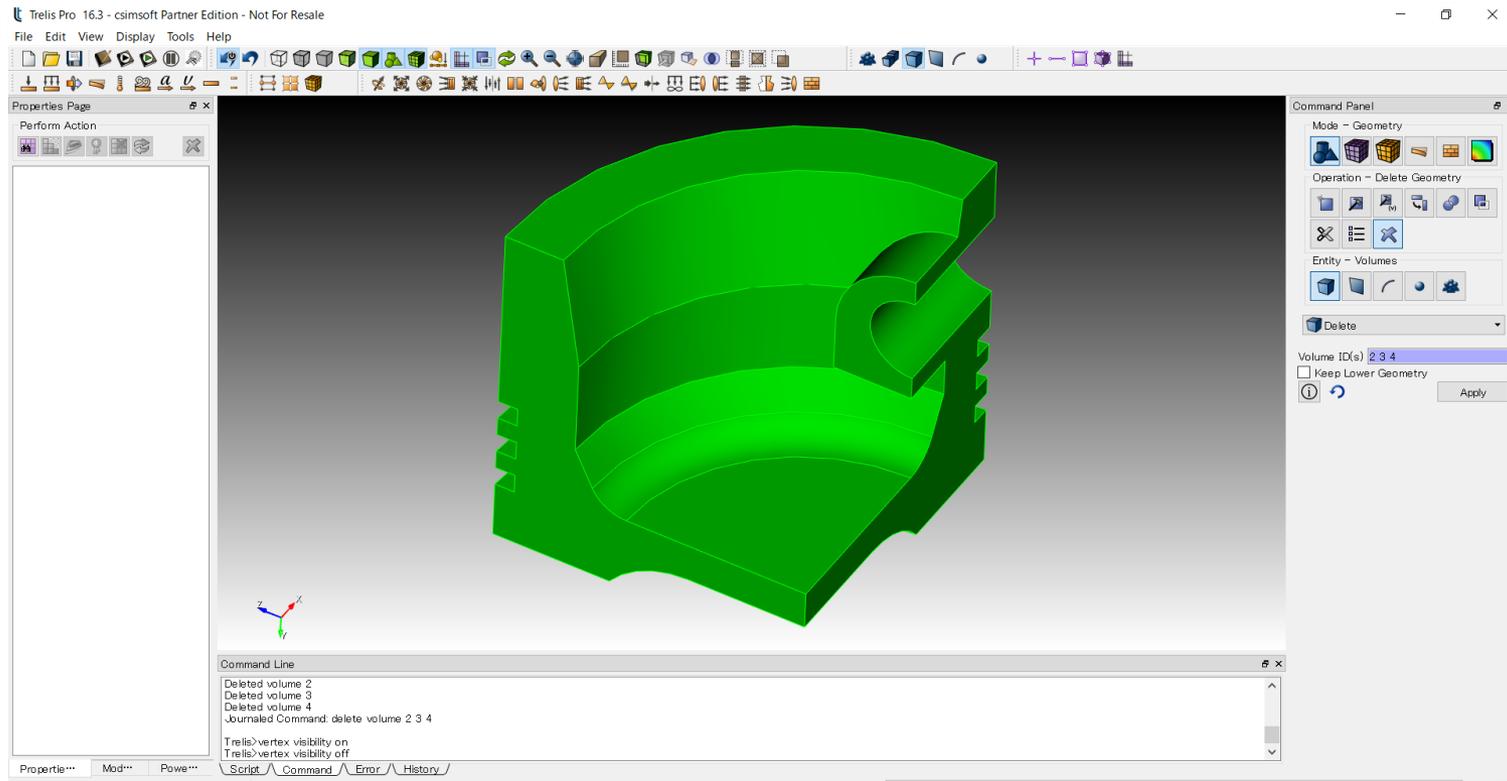
- Webcut
- プロパティの編集
- mesh vol all



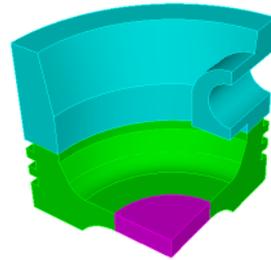
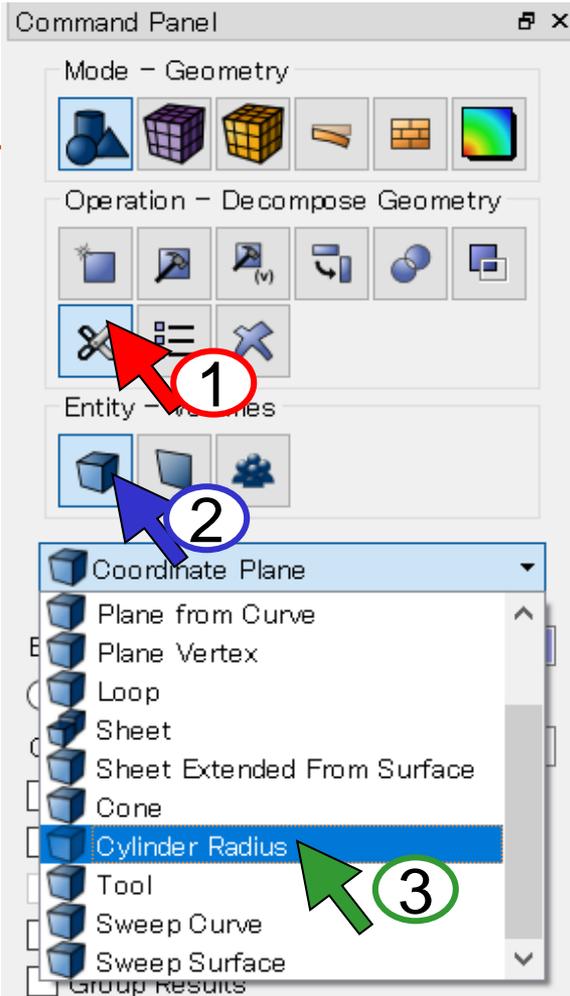
2

ジオメトリ の簡略化

図形の対称性を利用するために、モデルを四等分します。



モデルを簡略化し、分解する準備ができました。



3

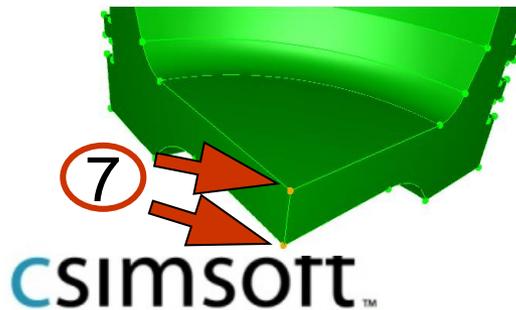
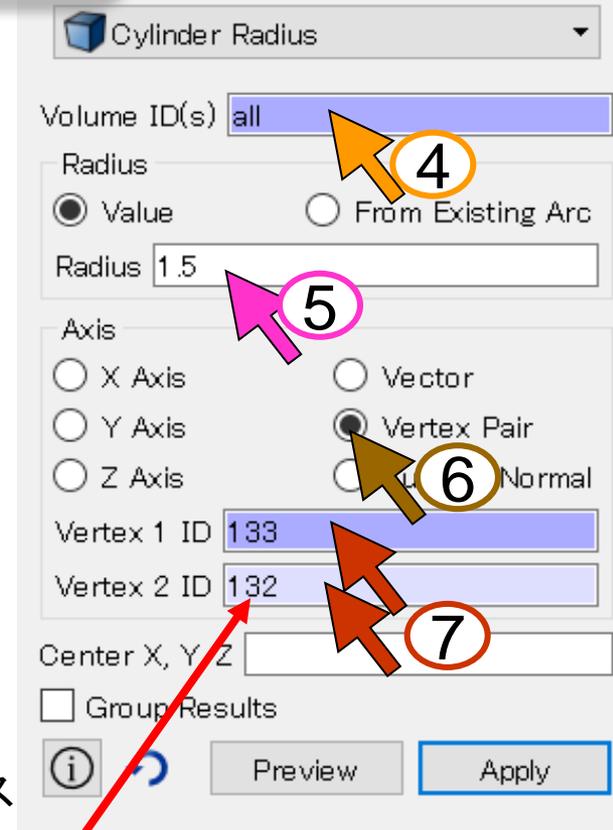
ジオメトリの分解

スイープできるボリュームにモデルをカットします。

円筒形で切断する方法を設定します。

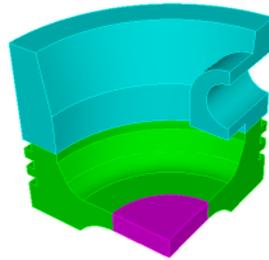
円筒の半径と軸を指定します。

- ① Operation → Decompose をクリック
- ② Entity → Volumes をクリック
- ③ ドロップダウンメニューから Cylinder Radius を選択
- ④ ボリュームを選択 (all)
- ⑤ Radius に 1.5 を入力
- ⑥ Vertex Pair を選択
- ⑦ 左図に示した2つのバーテックスを、グラフィクス上からマウスで選択し、円筒の軸を定義



ID は例と異なる場合があります。左図を参考に、正しいバーテックスをグラフィクス上で選択してください。

- Webcut
- プロパティの編集
- mesh vol all

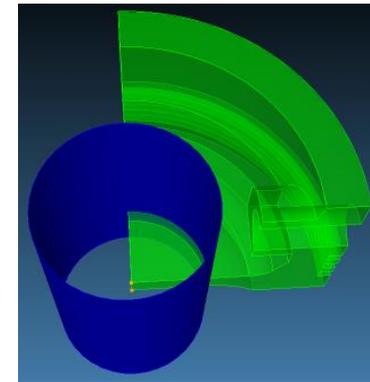


3

ジオメトリの 分解

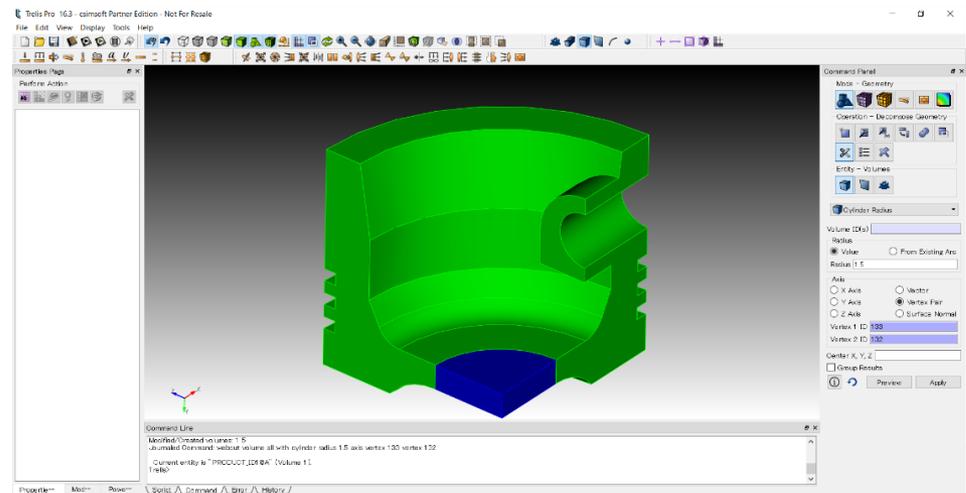
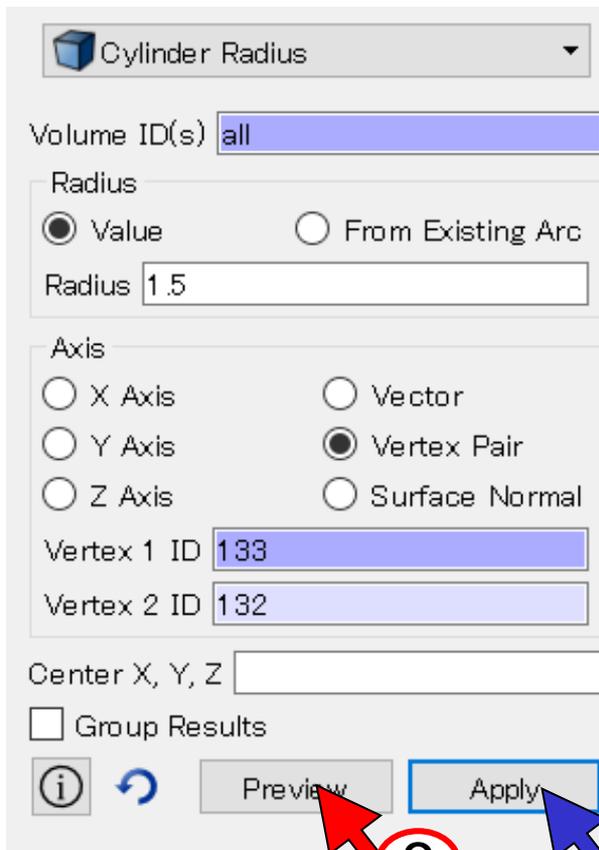
スワイプできるボリューム
にモデルをカットします。

円筒形でのWebcutを
プレビュー、実行します。



Webcut の
プレビュー

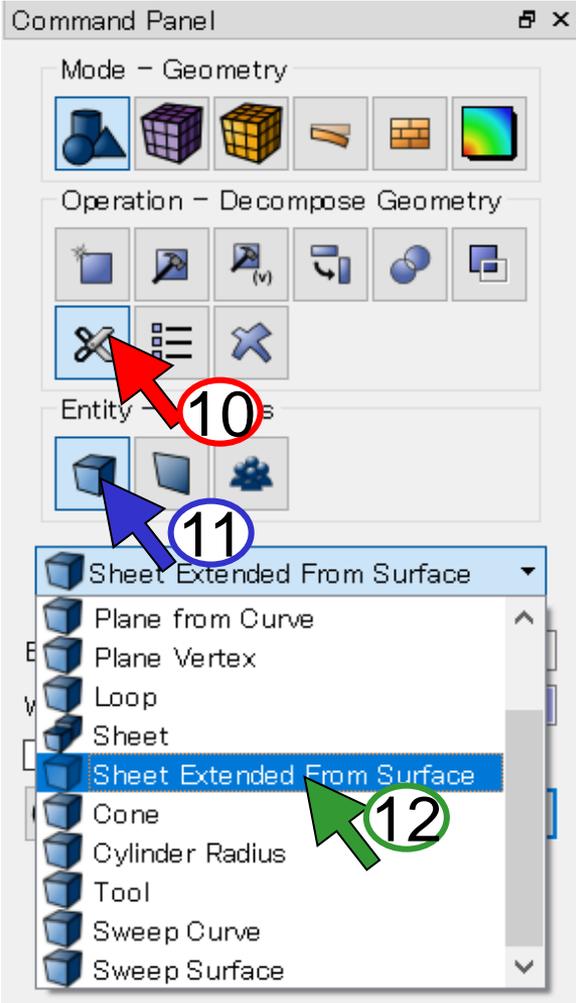
- Preview をクリックして
切断円筒のプレビューを
見ます。
- Apply をクリックしてボ
リュームを切断します。



- Webcut
- プロパティの編集
- mesh vol all

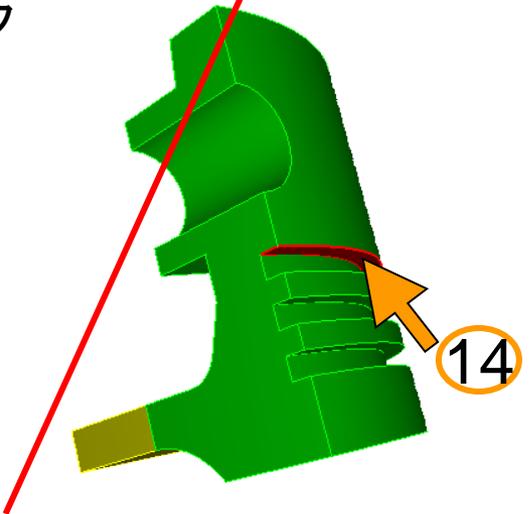
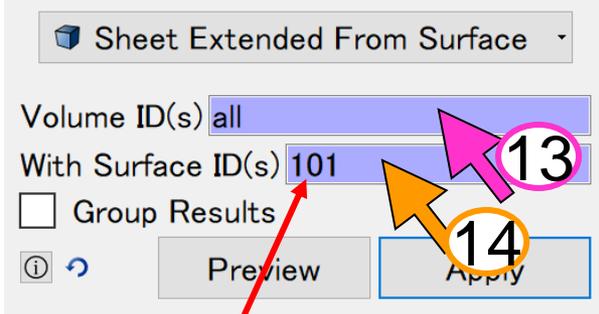
3 ジオメトリの分解

スweepできるボリュームにモデルをカットします。



平面で切断面を定義します。
切断面は、既存の面を延長して定義します。

- ⑩ Operation → Decompose をクリック
- ⑪ Entity → Volumes をクリック
- ⑫ ドロップダウンメニューから *Sheet Extended* を選択
- ⑬ 切断するボリュームを選択 (*all* または *1*)
- ⑭ 図に示したサーフェスをマウスで選択して、切断面を定義

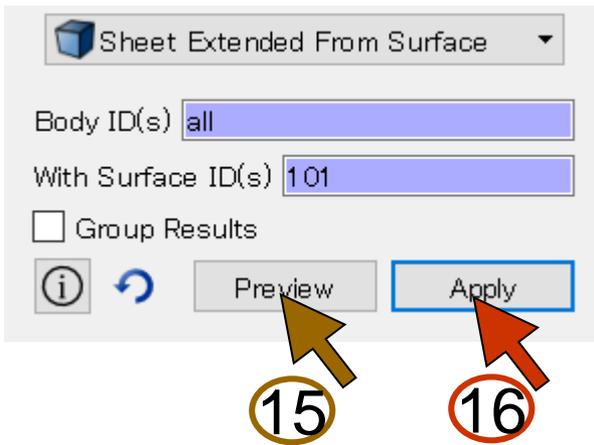


IDは例と異なる場合があります。上図を参考に、正しいサーフェスを選択してください。

- Webcut
- プロパティの編集
- mesh vol all



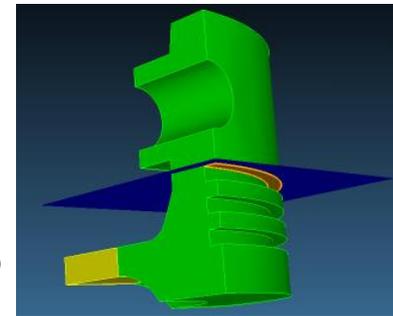
スイープできるボリュームにモデルをカットします。



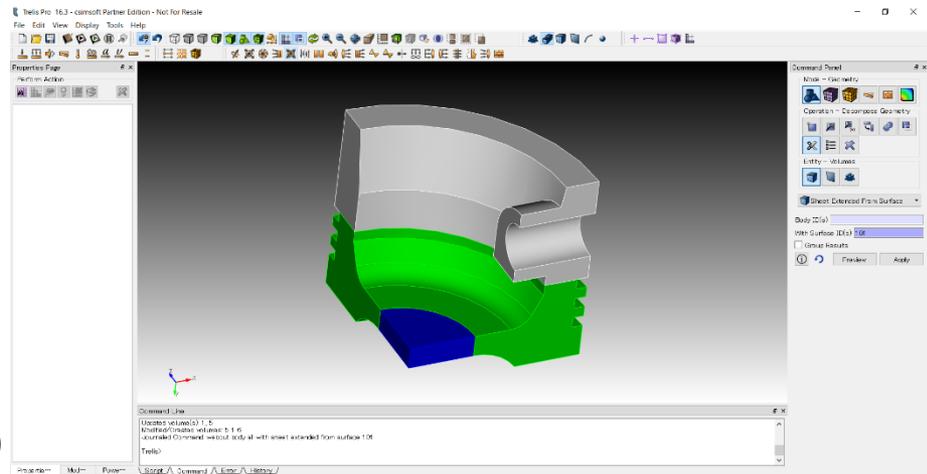
平面でのwebcut をプレビュー、実行します。

15 Preview をクリックして、切断面のプレビューを見る

16 Apply をクリックしてボリュームを切断



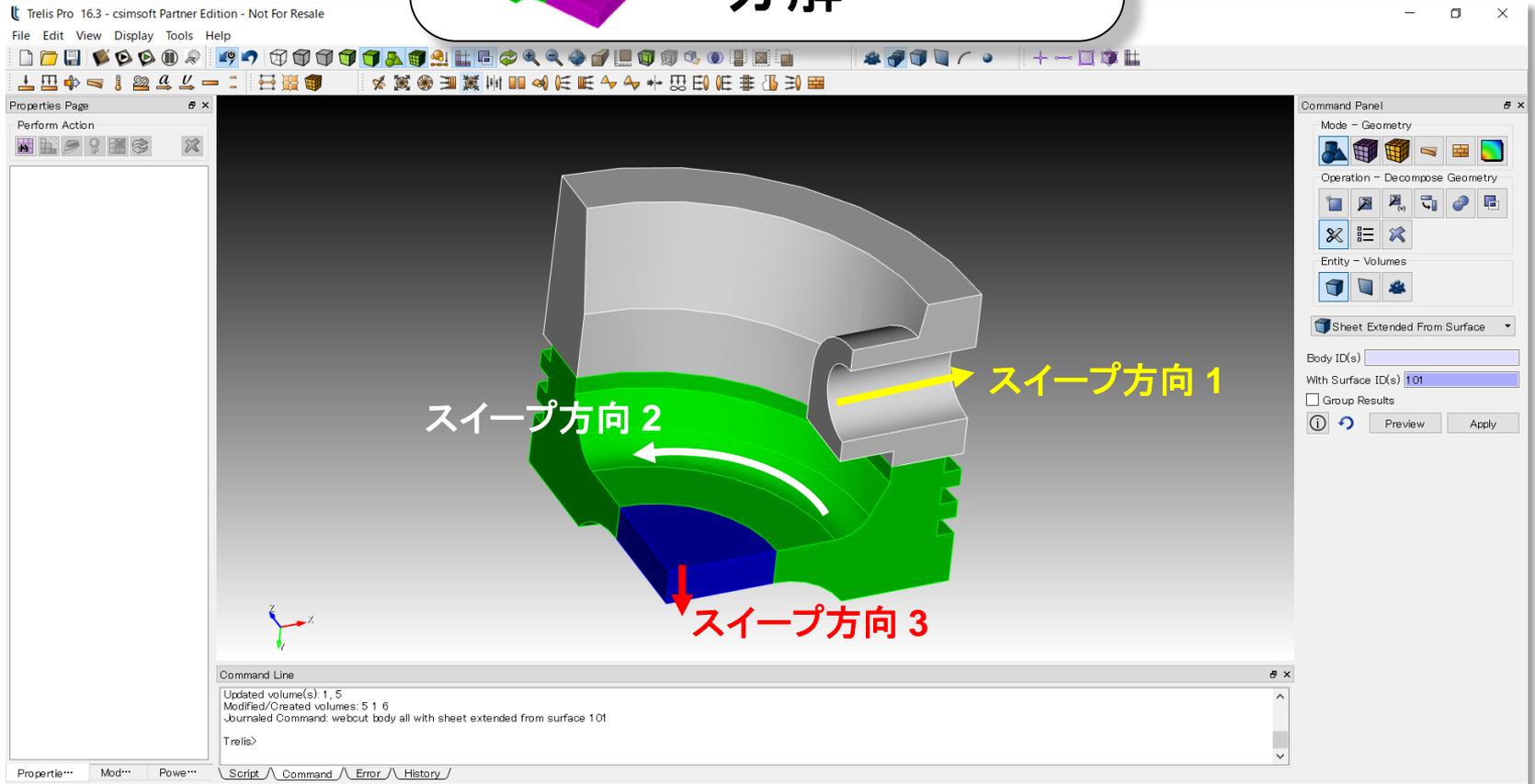
Webcut のプレビュー



- Webcut
- プロパティの編集
- mesh vol all



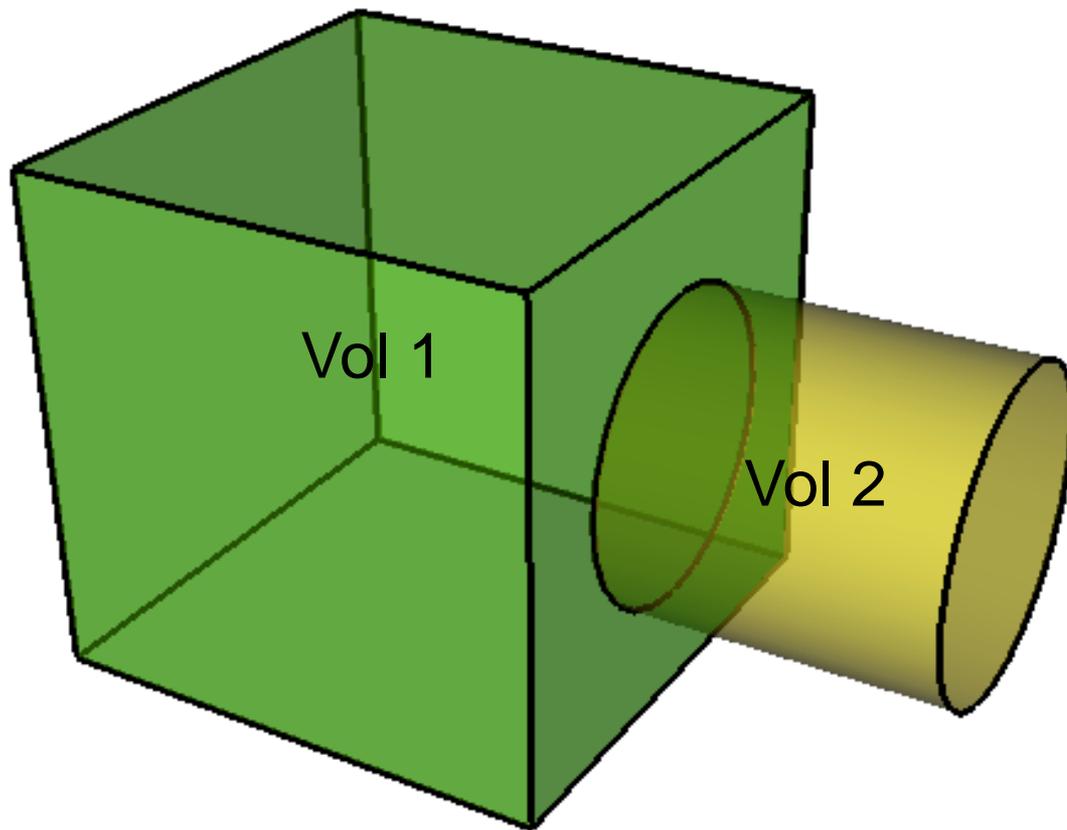
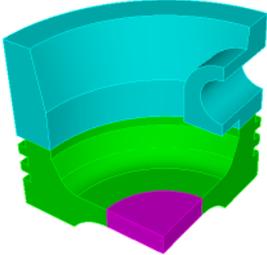
スイープできるボリュームにモデルをカットします。

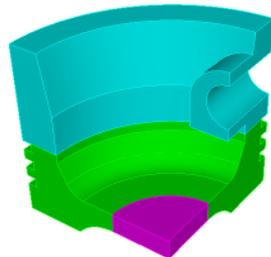


ボリュームは個々にスイープできる状態になりました。

ここまで分割すれば、デフォルトの設定で自動ヘキサメッシングが可能です。

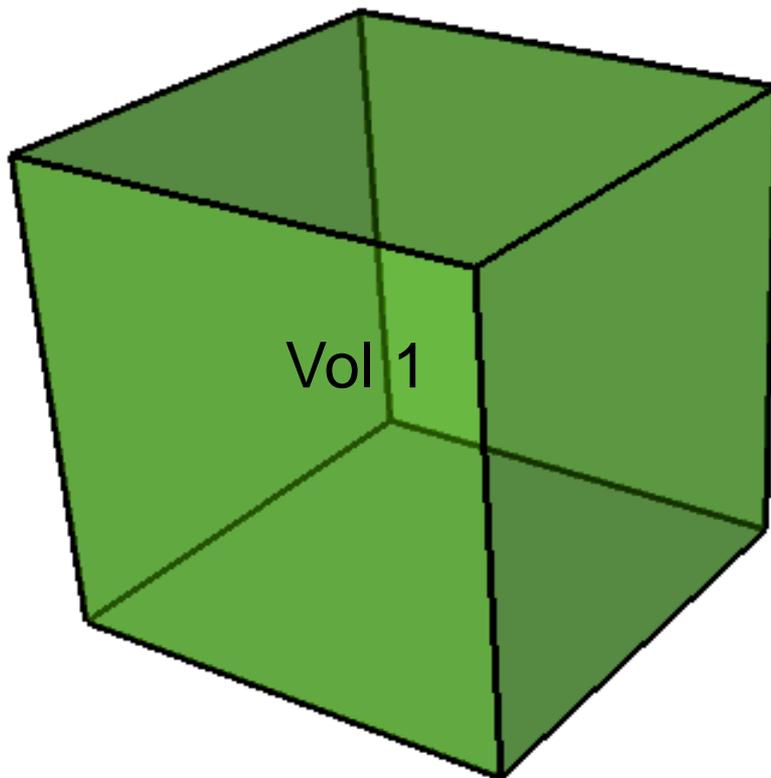
4
インプリント&
マージ



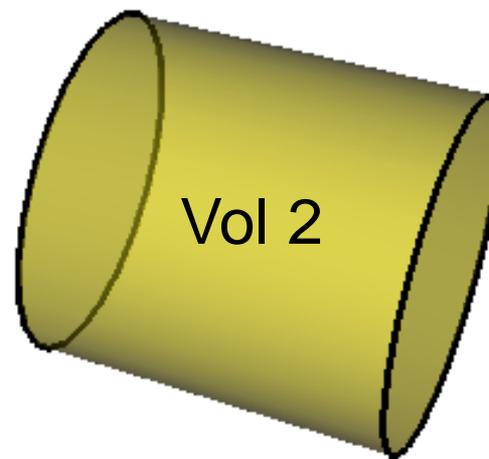


4

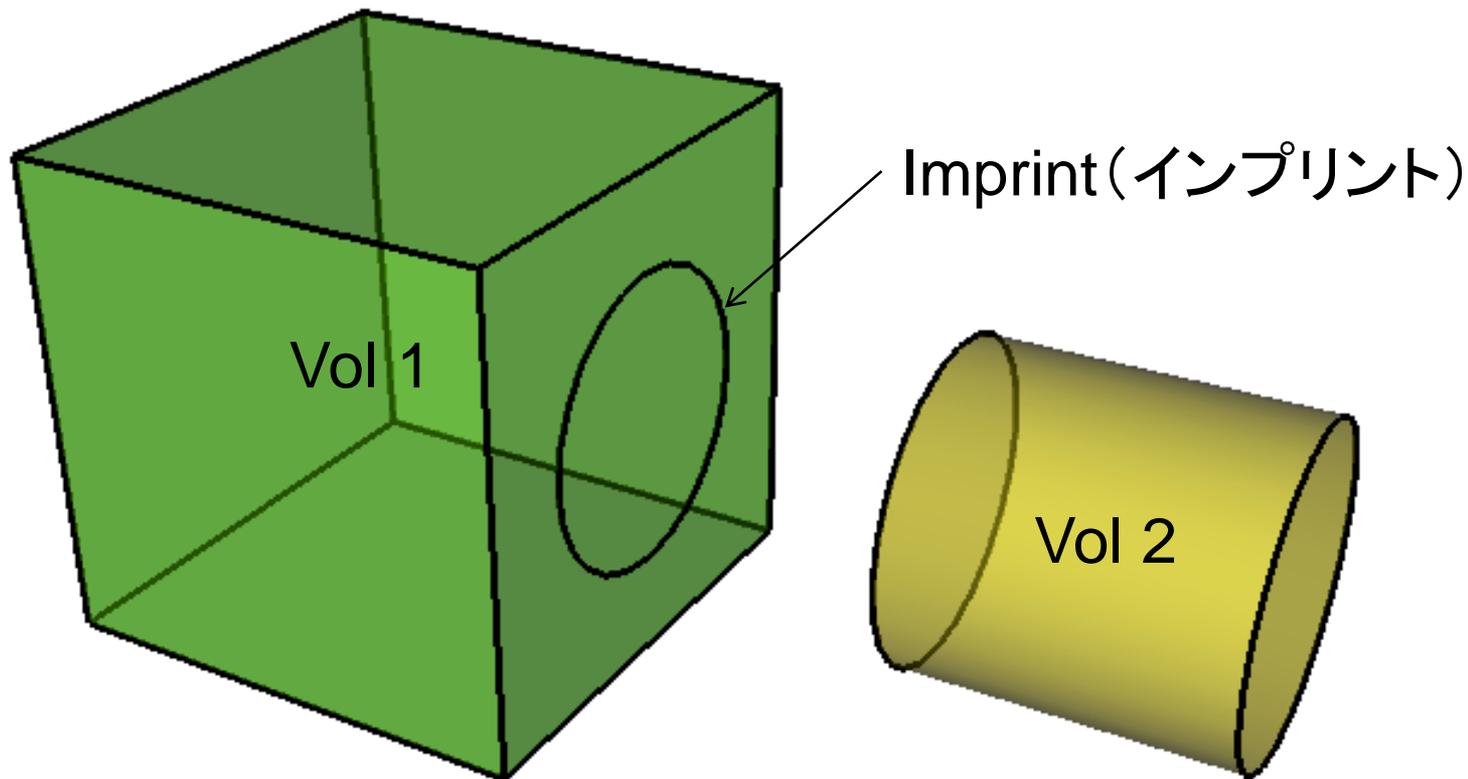
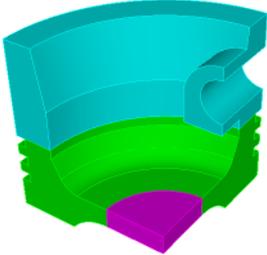
インプリント& マージ



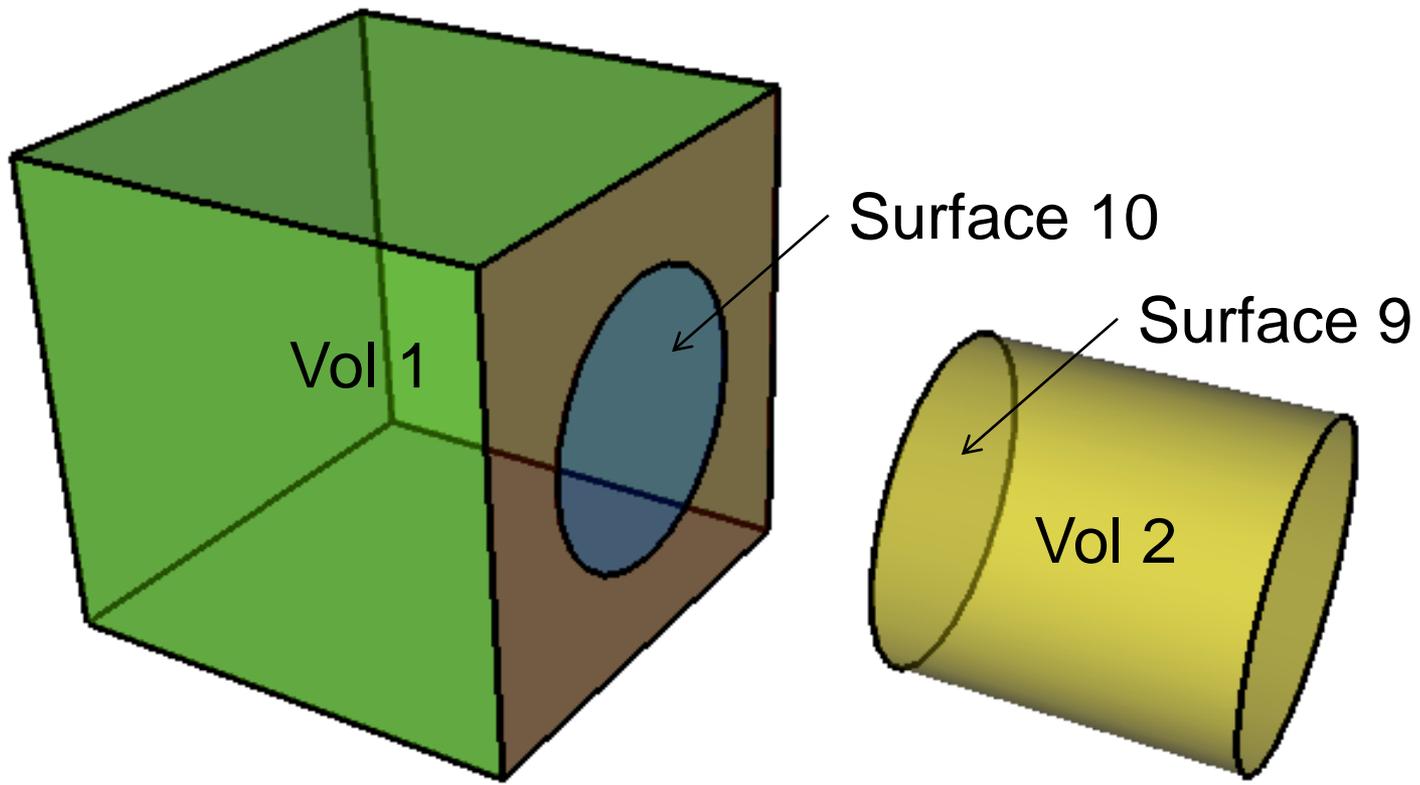
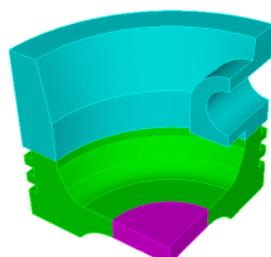
独立したボリューム間では、
互いに接している部分のノードは
共有されません。

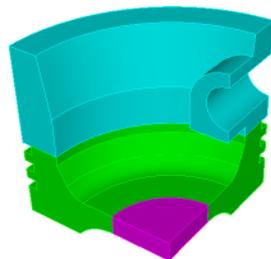


4
インプリント&
マージ



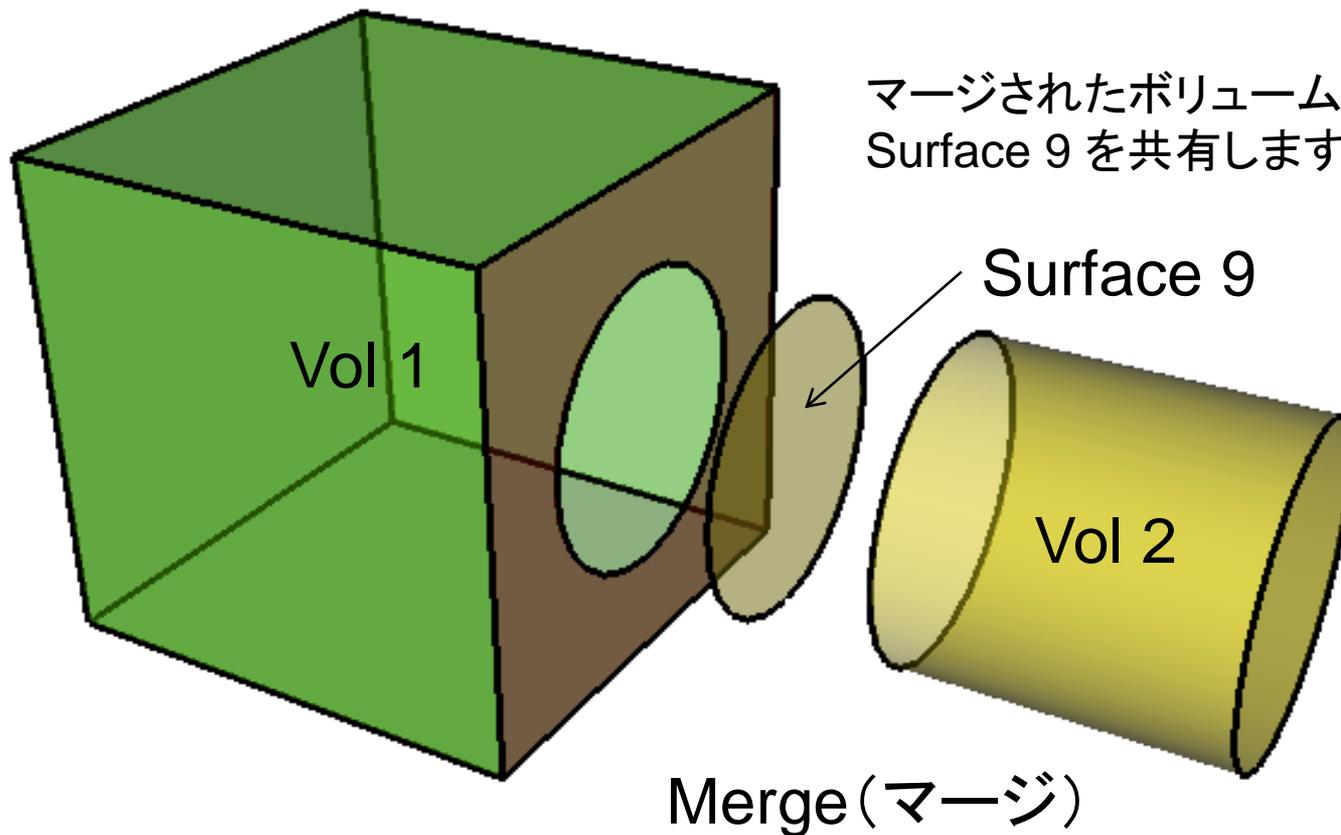
4
インプリント&
マージ

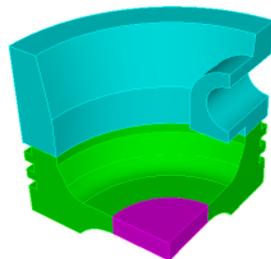




4

インプリント& マージ

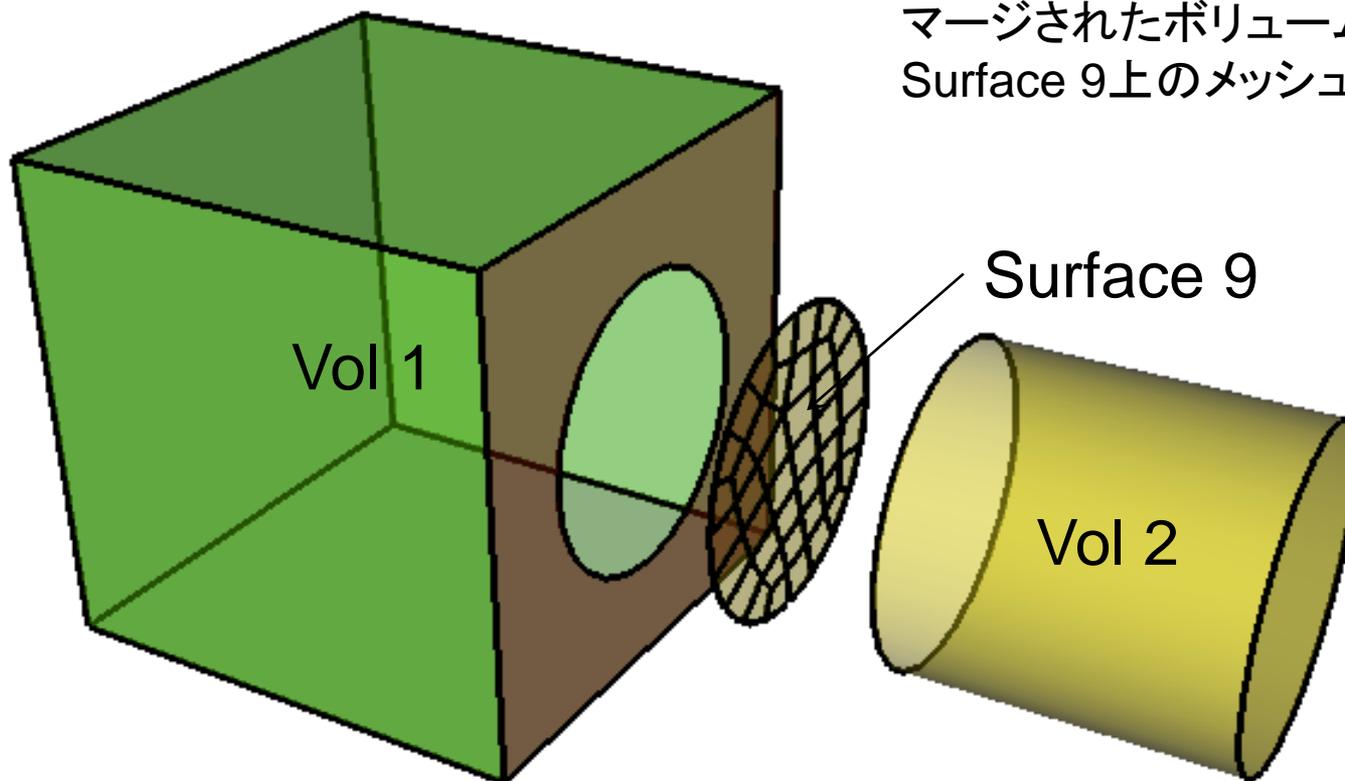


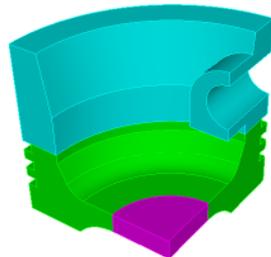


4

インプリント& マージ

マージされたボリューム1と2は、
Surface 9上のメッシュを共有します。

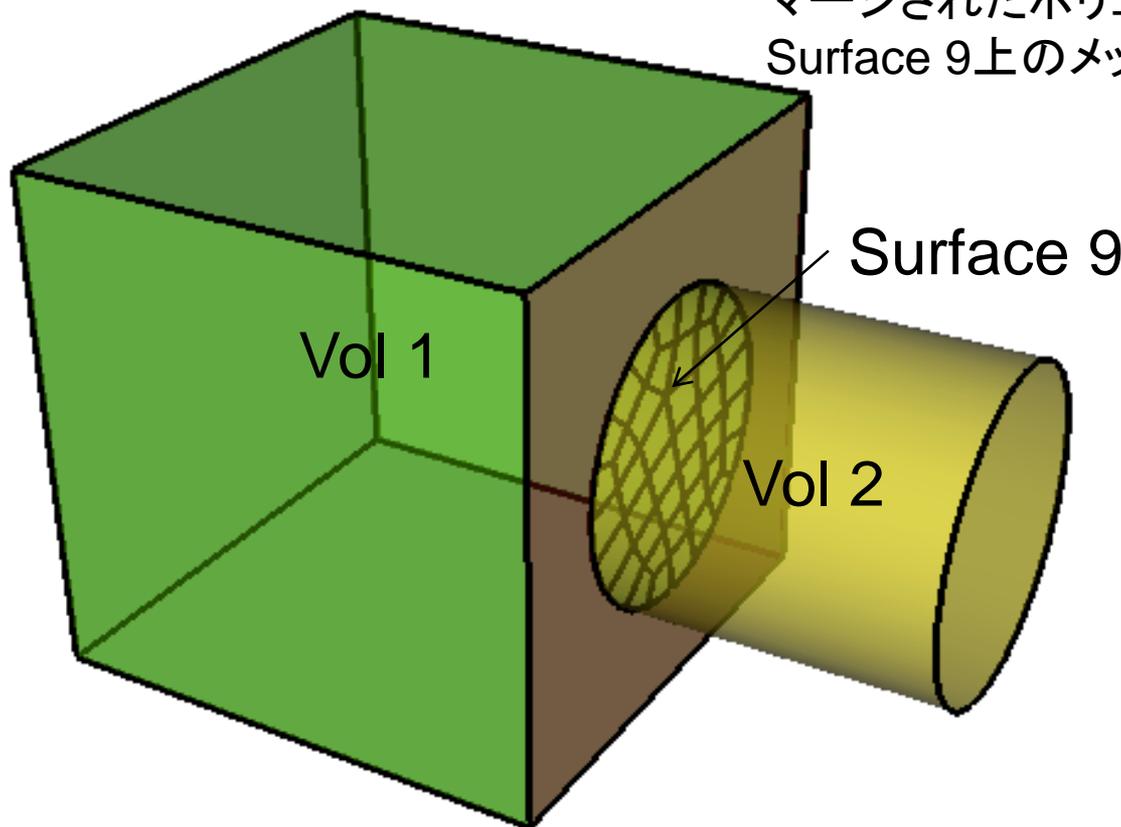


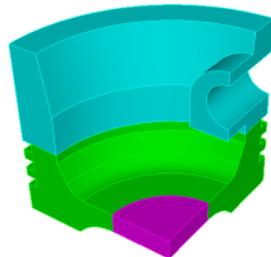


4

インプリント& マージ

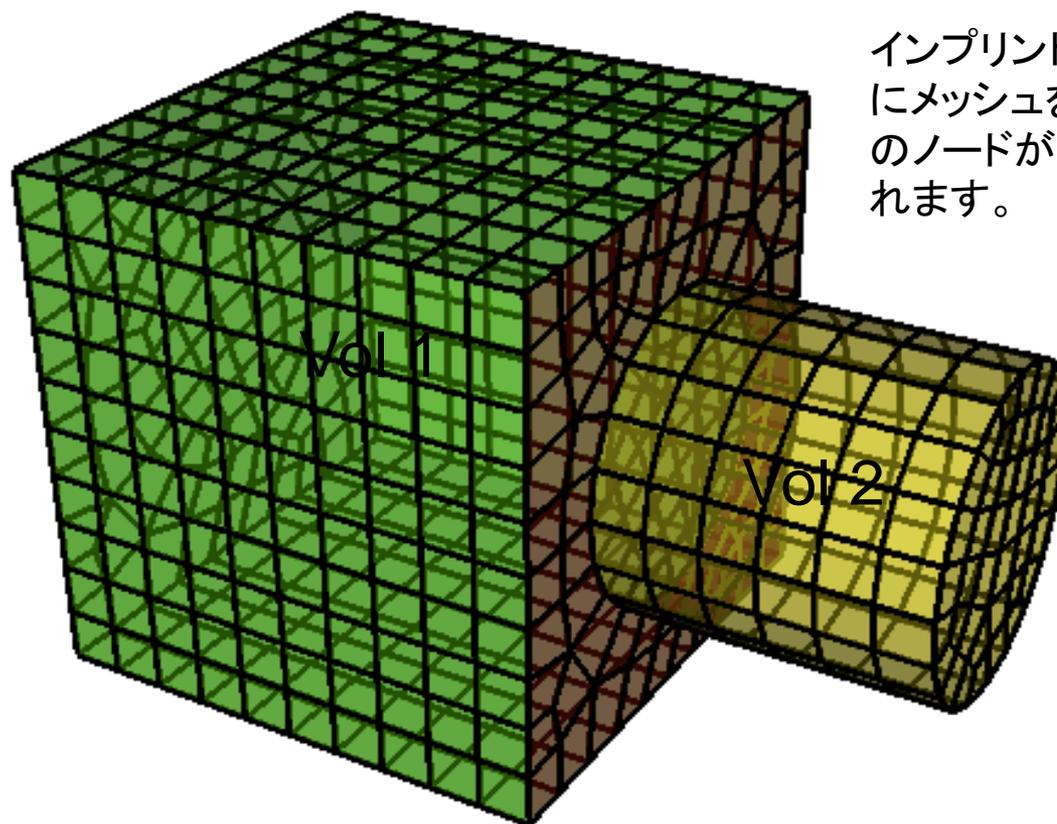
マージされたボリューム1と2は、
Surface 9上のメッシュを共有します。





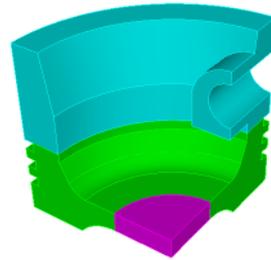
4

インプリント& マージ



インプリント・マージしたボリュームにメッシュを生成すると、境界面上のノードがボリューム間で共有されます。

- Webcut
- プロパティの編集
- mesh vol all



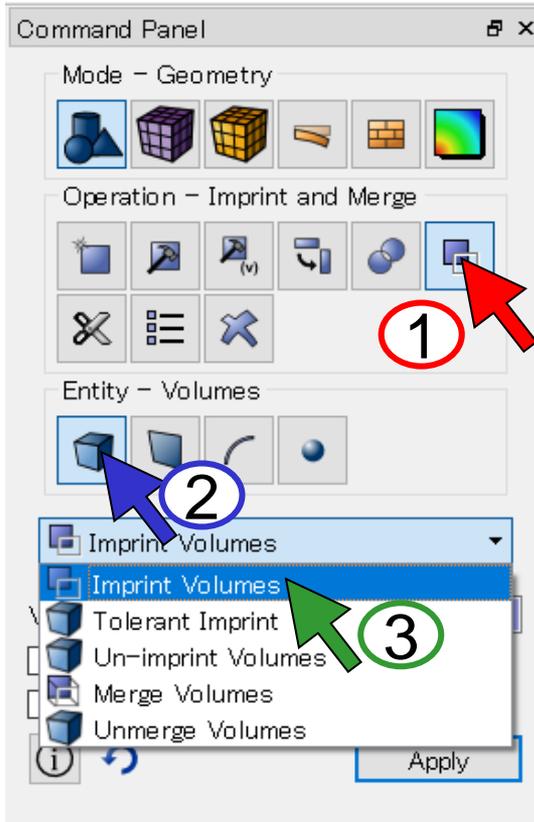
4

インプリント& マージ

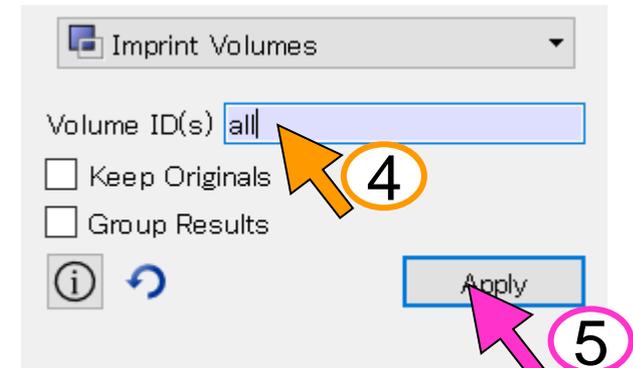
ボリュームを接続し、
連続したFEAメッシュ
が生成できるよう
します。

練習のため、今回はインプリントとマージを別々に行います。
(最新バージョンでは、まとめて行うことも出来ます)

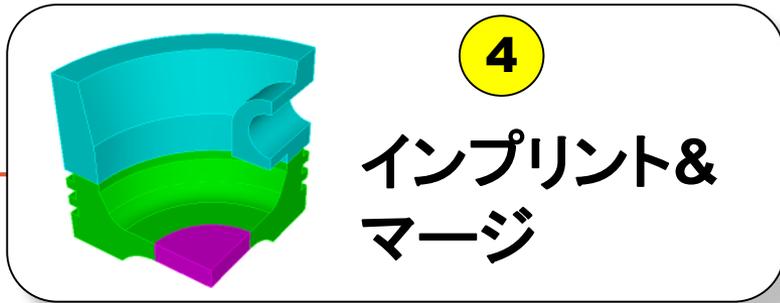
ボリュームのインプリントを行います。



- ① Operation → *Imprint and Merge* をクリック
- ② Entity → *Volumes* を選択
- ③ ドロップダウンメニューから *Imprint Volumes* を選択
- ④ ボリュームを選択
- ⑤ *Apply* をクリック



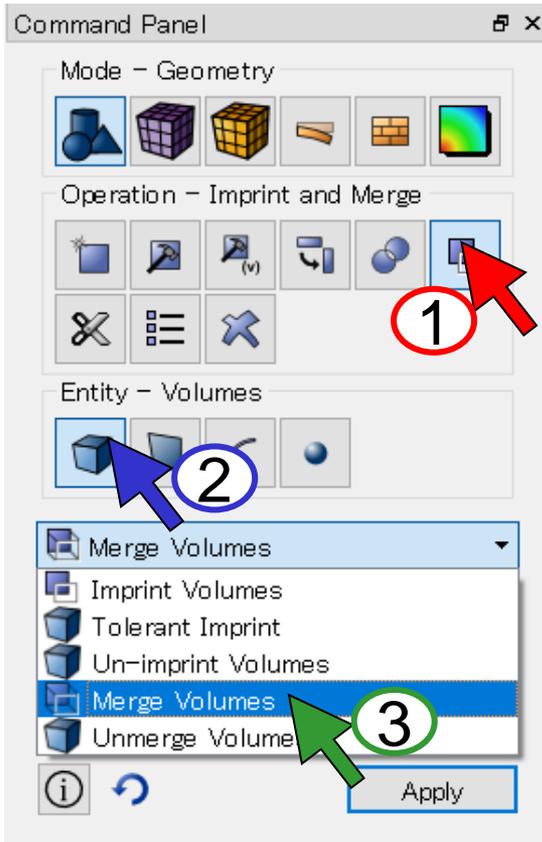
- Webcut
- プロパティの編集
- mesh vol all



インプリント&マージ

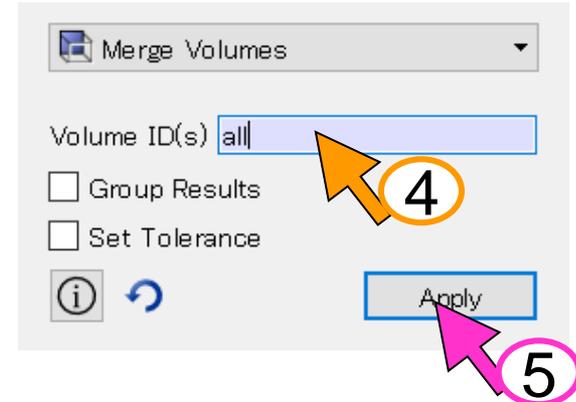
ボリウムを接続し、連続したFEAメッシュが生成できるようになります。

練習のため、今回はインプリントとマージを別々に行います。
(最新バージョンでは、まとめて行うことも出来ます)



ボリウムをマージします。

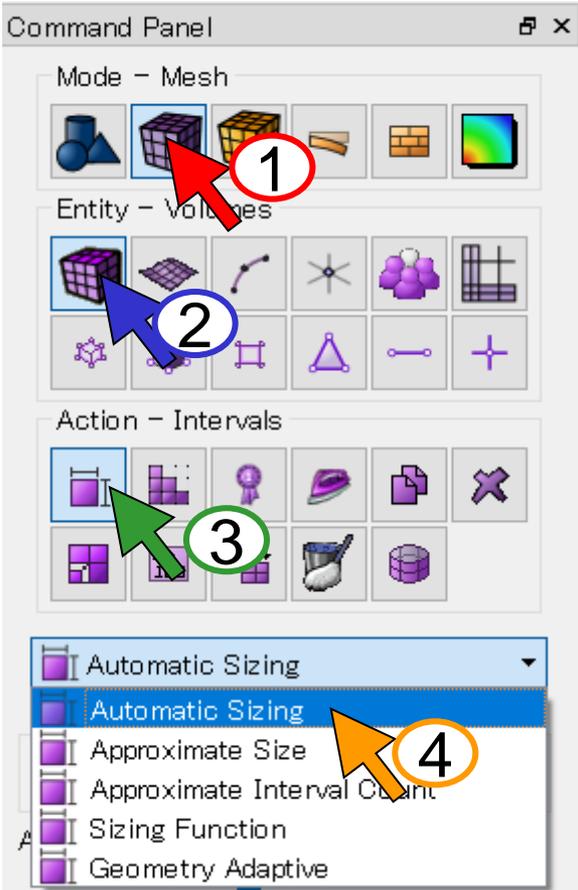
- ① Operation → *Imprint and Merge* をクリック
- ② Entity → *Volumes* を選択
- ③ ドロップダウンメニューから *Merge Volumes* を選択
- ④ ボリウムを選択
- ⑤ *Apply* をクリック



- Webcut
- プロパティの編集
- mesh vol all

5 スキーム & インターバル設定

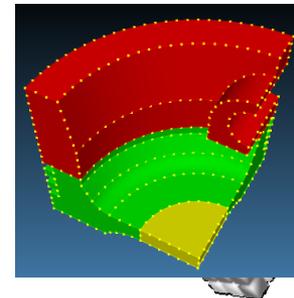
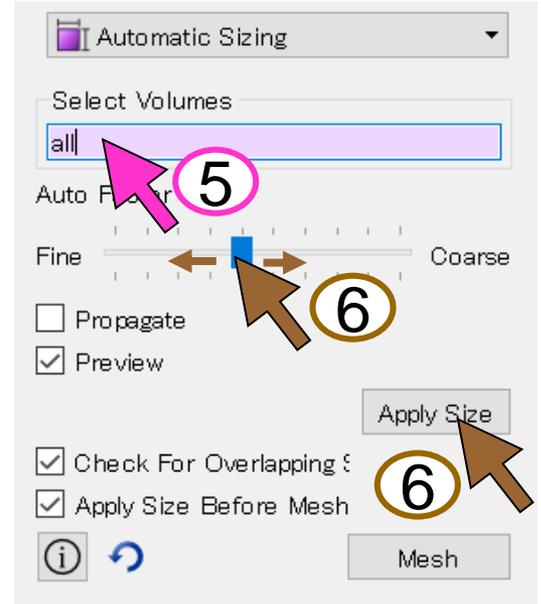
使用するメッシュスキームとジオメトリのメッシュサイズを設定します。



メッシュサイズのプレビューと設定

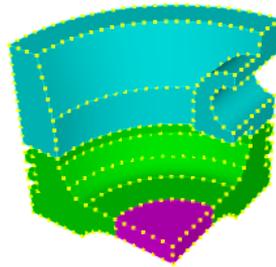
- ① Mode → Mesh をクリック
- ② Entity → Volumes をクリック
- ③ Action → Intervals をクリック
- ④ ドロップダウンメニューから Automatic Sizing を選択
- ⑤ ボリュームを全て選択
- ⑥ 適切な解像度になるまでスライダを動かして調節し、Apply をクリック

(グラフィックウィンドウでプレビューを確認)



カーブ上のノードのプレビュー

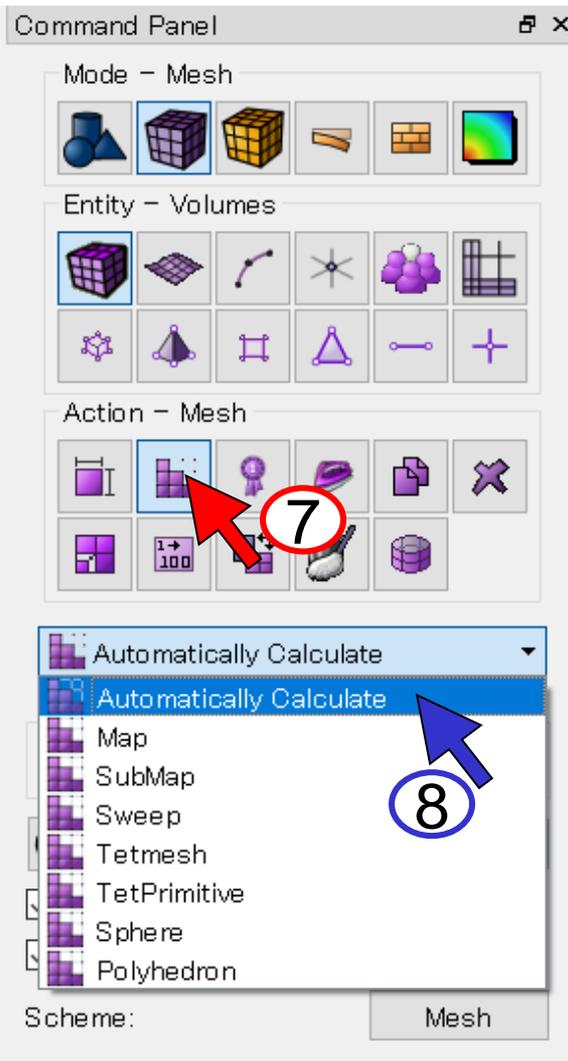
- Webcut
- プロパティの編集
- mesh vol all



5

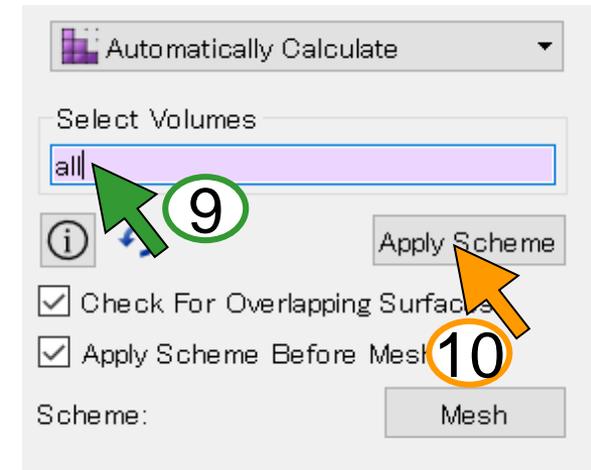
スキーム & インターバル設定

使用するメッシュスキームとジオメトリのメッシュサイズを設定します。

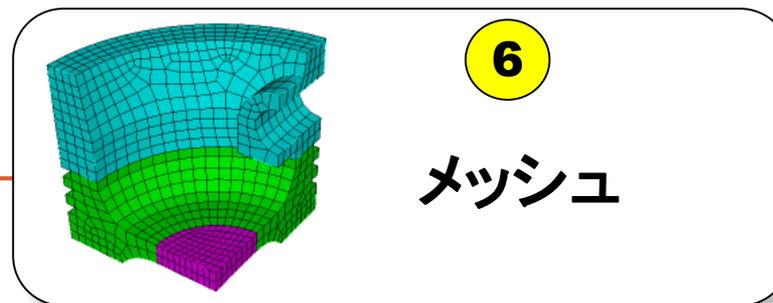


メッシュスキームをAutomatically Calculateに設定します。

- ⑦ Action→Mesh をクリック
- ⑧ ドロップダウンメニューからAutomatically Calculate をクリック
- ⑨ 全てのボリウムを選択 (all)
- ⑩ Apply Scheme をクリック



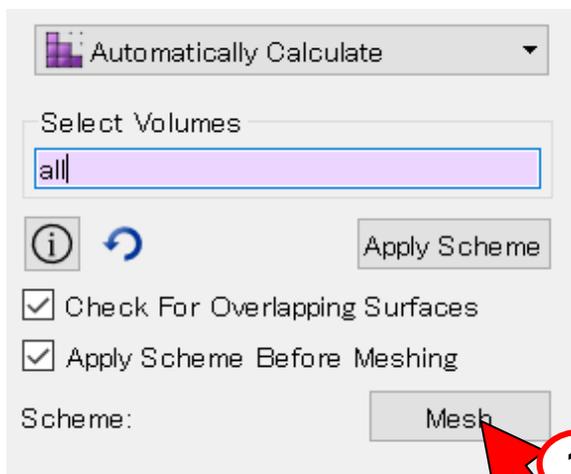
- Webcut
- プロパティの編集
- mesh vol all



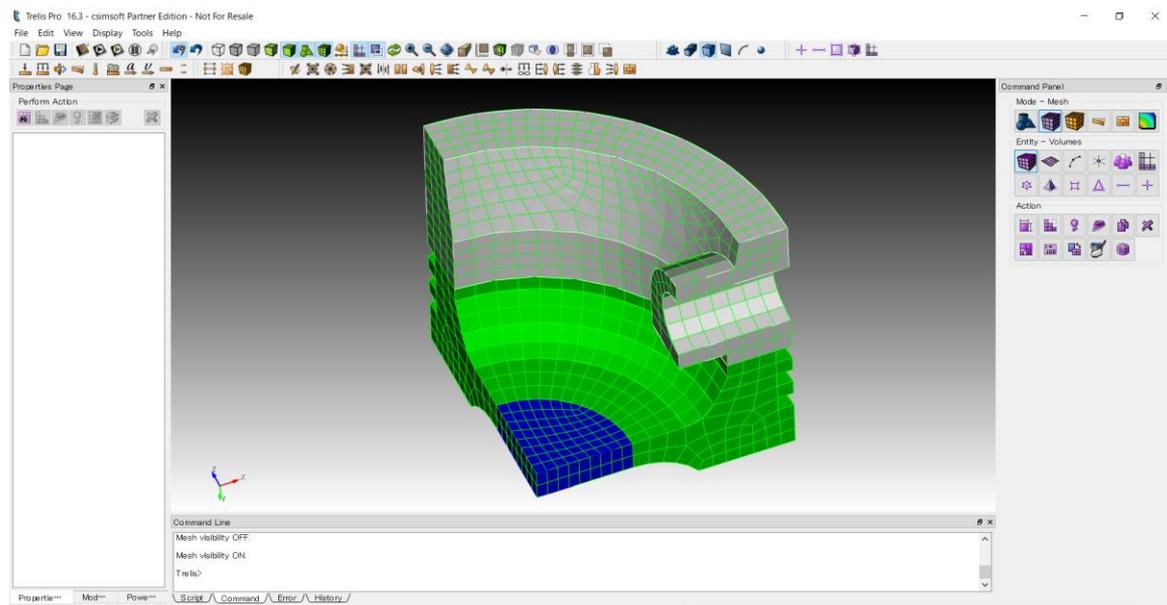
メッシュ

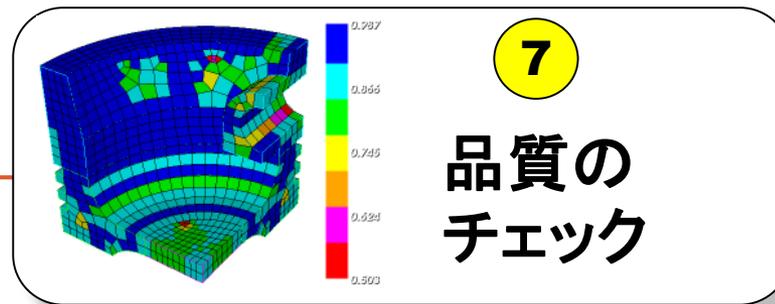
ボリウムにヘキサ
メッシュを作成します。

ボリウムのメッシング

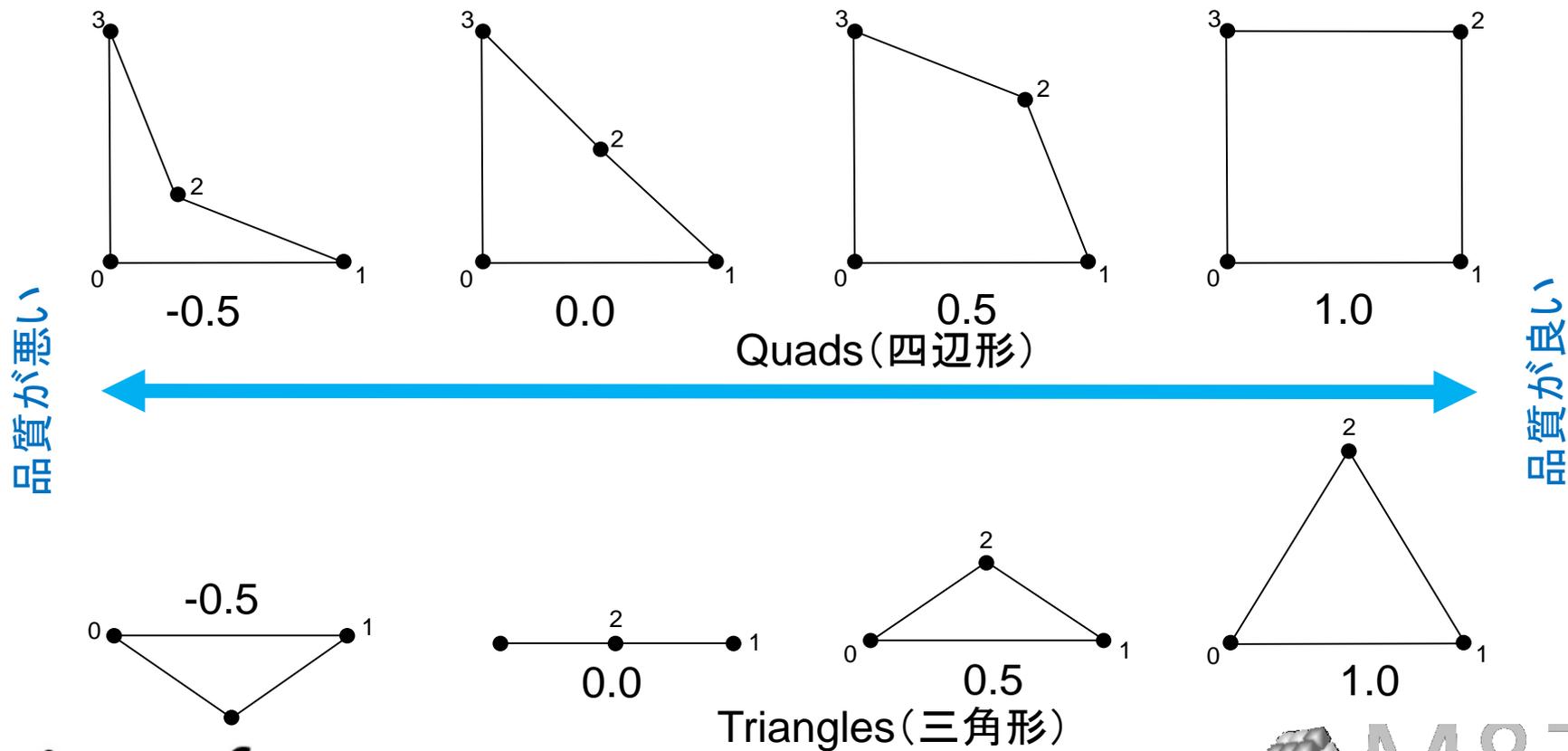


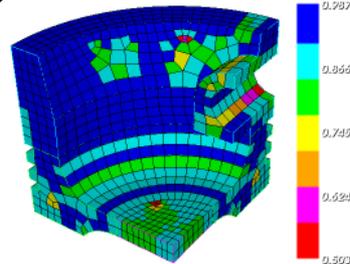
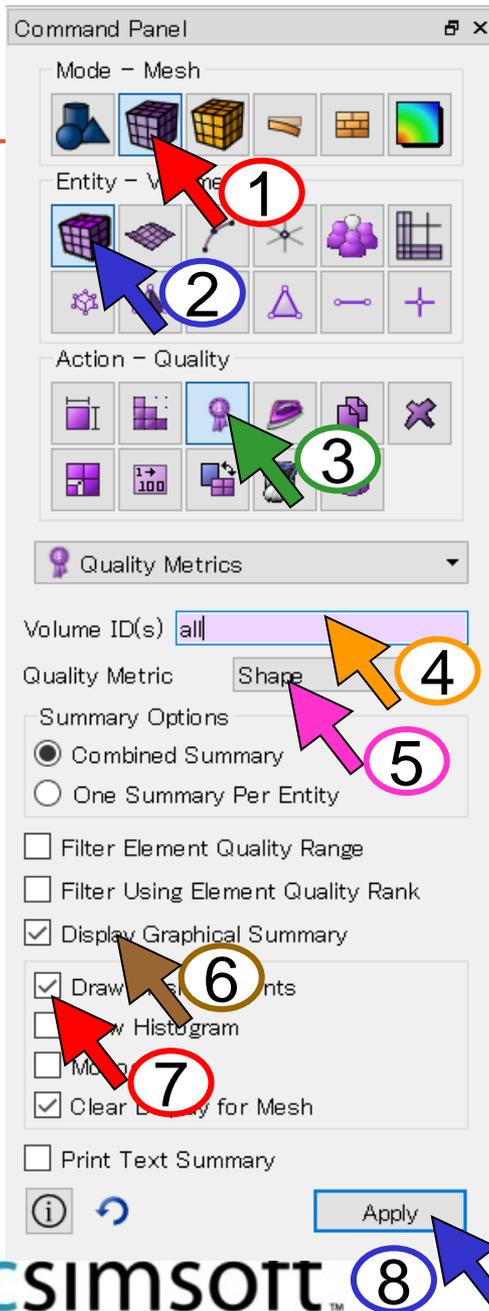
① Mesh ボタンをクリック
します。





Shape 品質メトリクス





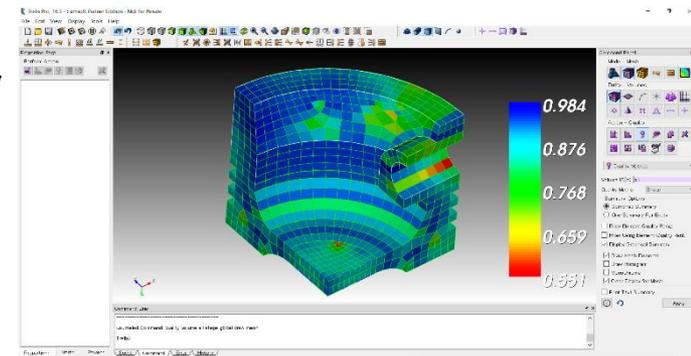
7 品質の チェック

要素の品質が解析に適しているかどうか
チェックします。

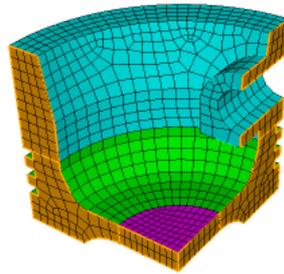
品質メトリクス: *Shape* (要素形状) に基づいた要素
品質を色表示します。

- ① *Mode* → *Mesh* をクリック
- ② *Entity* → *Volumes* をクリック
- ③ *Action* → *Quality* をクリック
- ④ 全てのボリュームを選択
- ⑤ ドロップダウンメニューから *Shape* 品質メトリクスを選択
- ⑥ *Display Graphical Summary* にチェックを入れる
- ⑦ *Draw Mesh Elements* にチェックを入れる
- ⑧ *Apply* をクリック

結果を確認した後は、F5で画面をリフレッシュできます。



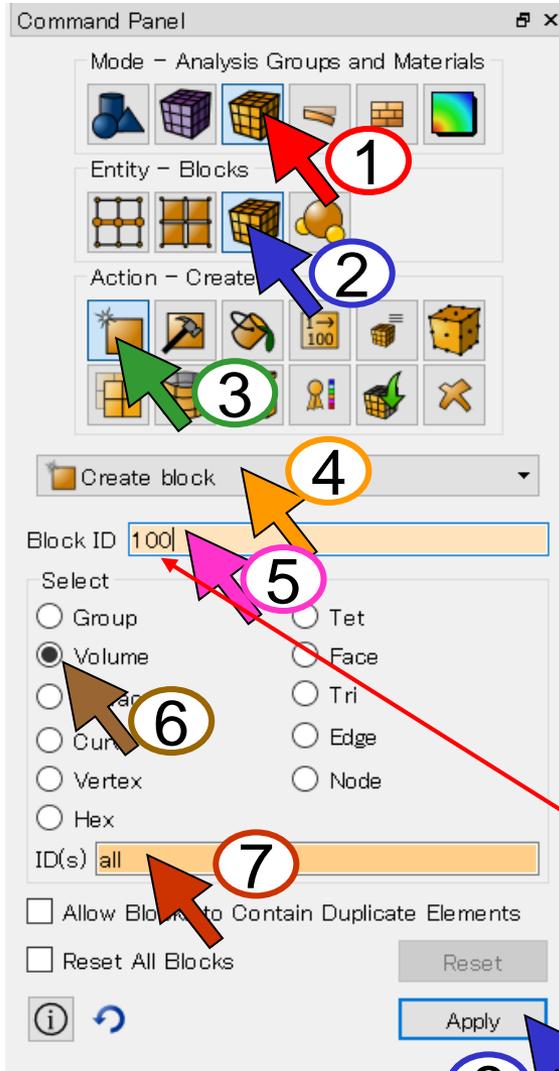
- Webcut
- プロパティの編集
- mesh vol all



8

境界条件 の設定

モデルに対して、マテリアルと境界条件の設定を行います。



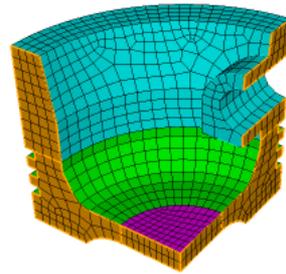
マテリアルのブロックを設定します。

ボリュームを指定して、セル要素をブロックにまとめます。
(直接セルを指定することも可能です)

- ① Mode → Analysis Groups and Materials をクリック
- ② Entity → Blocks をクリック
- ③ Action → Create をクリック
- ④ ドロップダウンメニューから Create block を選択
- ⑤ Block ID に 100 と入力
- ⑥ Volume を選択
- ⑦ ブロックに追加するボリュームを選択 (all)
- ⑧ Apply をクリックして、新しいブロック 100 を作成

Block ID は任意で構いません

- Webcut
- プロパティの編集
- mesh vol all



境界条件 の設定

モデルに対して、マテリアルと境界条件の設定を行います。

追加したブロックの要素タイプを *HEX8* に設定します。

① Action→Element Types をクリック

② Block ID に 100 を入力

さきほど設定したBlock ID を入力します

③ Volumes を選択

④ Hex8 を選択

FEMでは必要に応じて、より高次の要素タイプを指定します。(CFDではその必要はありません)

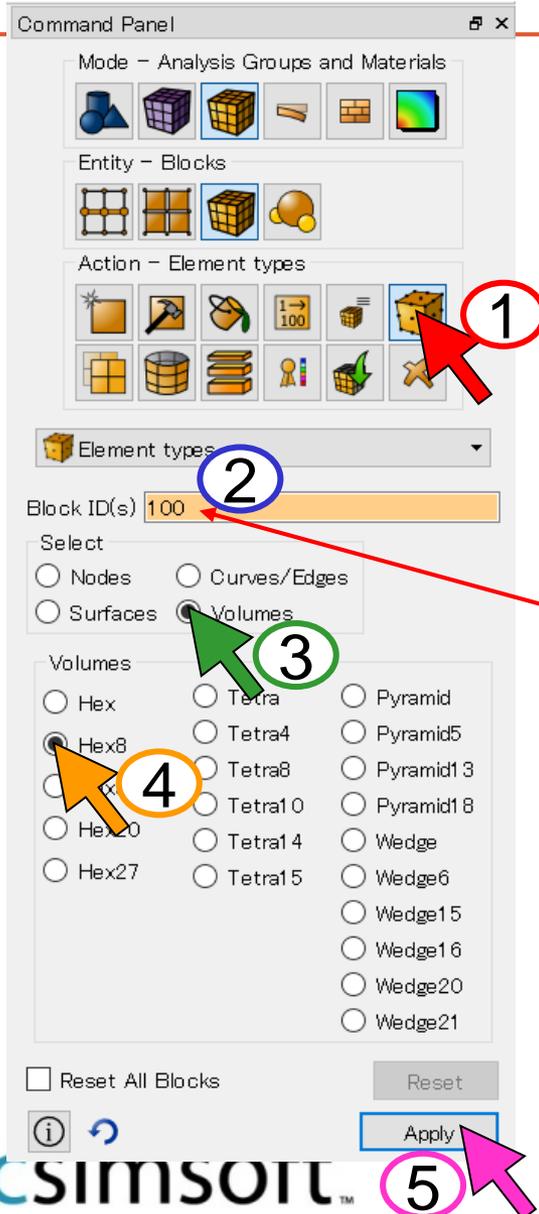
⑤ Apply をクリック

CUBITの要素タイプは、ExodusII フォーマットの要素タイプに対応しています。

詳細は

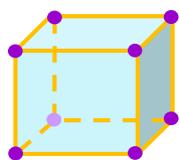
http://csimsoft.com/help/finite_element_model/export/exporting_finite_element_model.htm

をご覧ください。

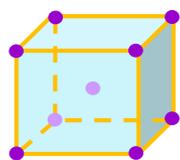


CUBITの要素タイプ

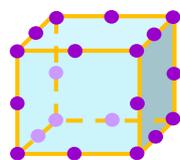
要素タイプ例(3次元要素)



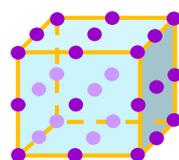
HEX8



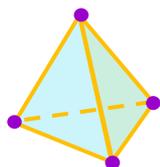
HEX9



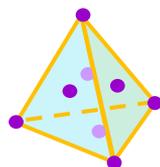
HEX20



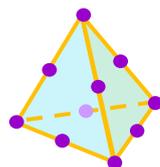
HEX27



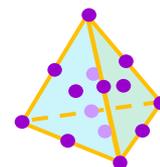
TETRA4



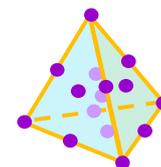
TETRA8



TETRA10



TETRA14

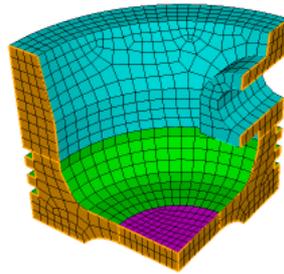


TETRA15

CUBITの要素タイプは、ExodusII フォーマットの要素タイプに対応しています。

詳細は http://csimsoft.com/help/finite_element_model/export/exporting_finite_element_model.htm をご覧ください。

- Webcut
- プロパティの編集
- mesh vol all



8

境界条件の設定

モデルに対して、マテリアルと境界条件の設定を行います。

サイドセットを作成します。

サーフェスを指定することで、face要素をサイドセットにまとめます。
(直接face要素を指定することも可能です)

① Entity→Sidesets をクリック

② Action→Create sideset をクリック

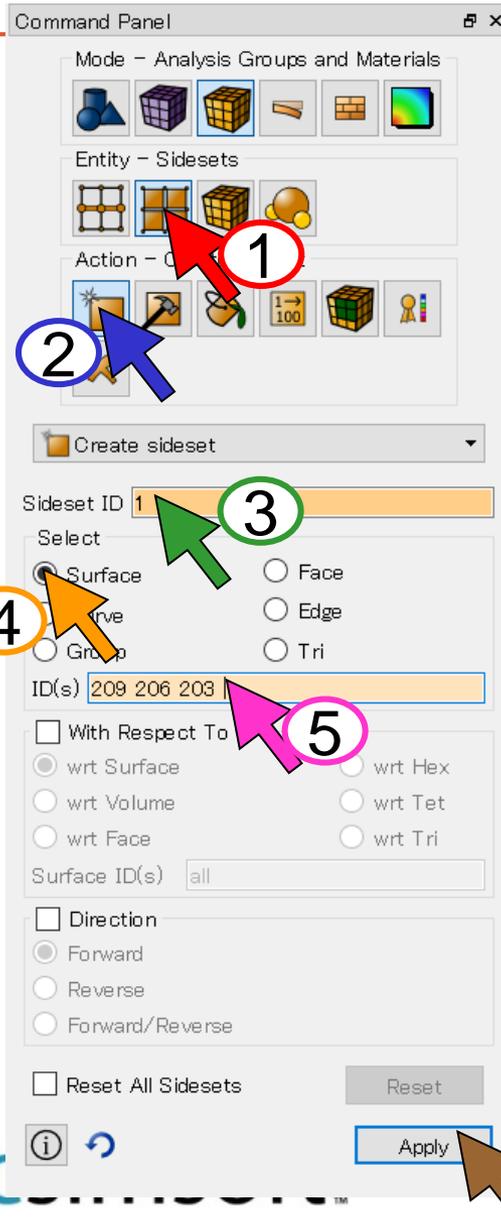
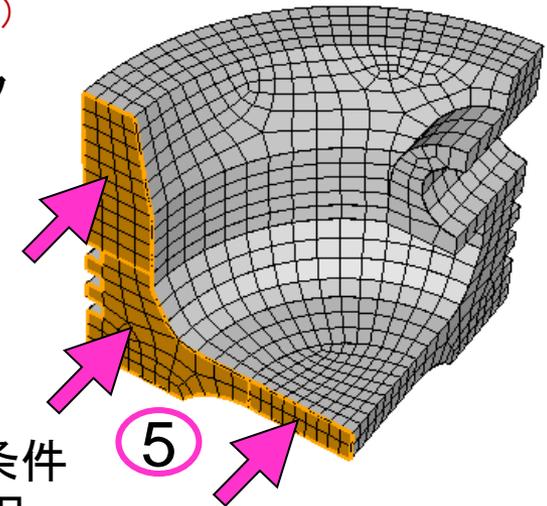
③ Sideset ID に1を入力

④ Surface を選択

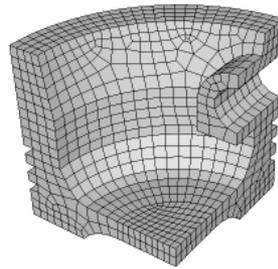
⑤ 右図に示したように、境界条件を適用するサーフェスを選択

Ctrlキーを押しながら複数のエンティティを選択します

⑥ Apply をクリック



- Webcut
- プロパティの編集
- mesh vol all



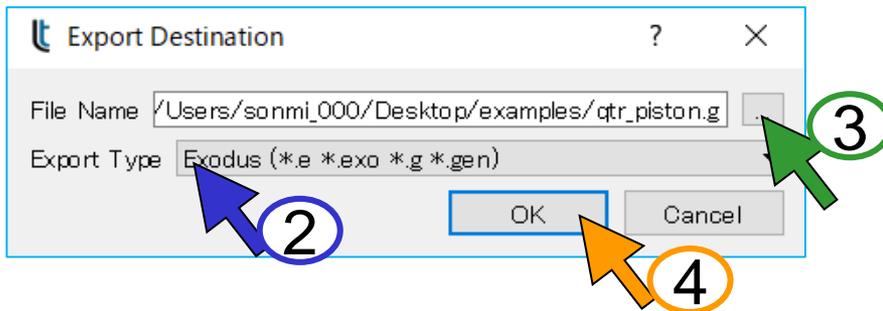
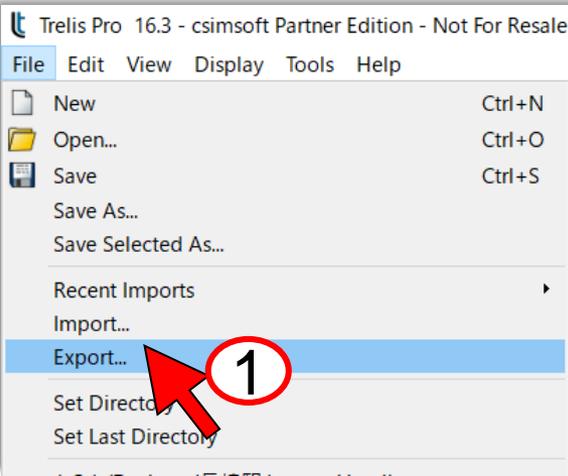
9

メッシュのエクスポート

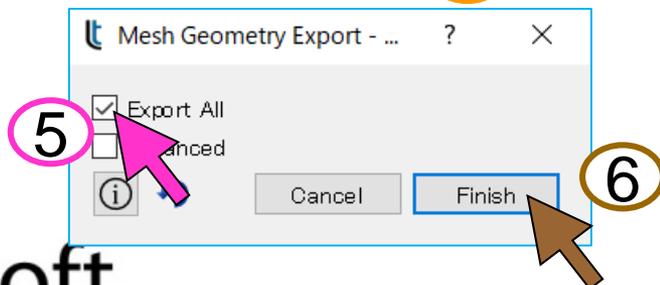
解析に使用するFEAメッシュファイルを作成します。

メッシュ・ブロック・サイドセットを含む Genesis file をエクスポートします。

- ① メニューの *File* → *Export* をクリック
- ② ドロップダウンメニューから *Exodus* を選択



- ③ *FileName* 欄の横のボタンをクリックし、保存するディレクトリにファイル名“qtr_piston.g”を入力
- ④ *OK* をクリック
- ⑤ *Export All* をチェック
- ⑥ *Finish* をクリック



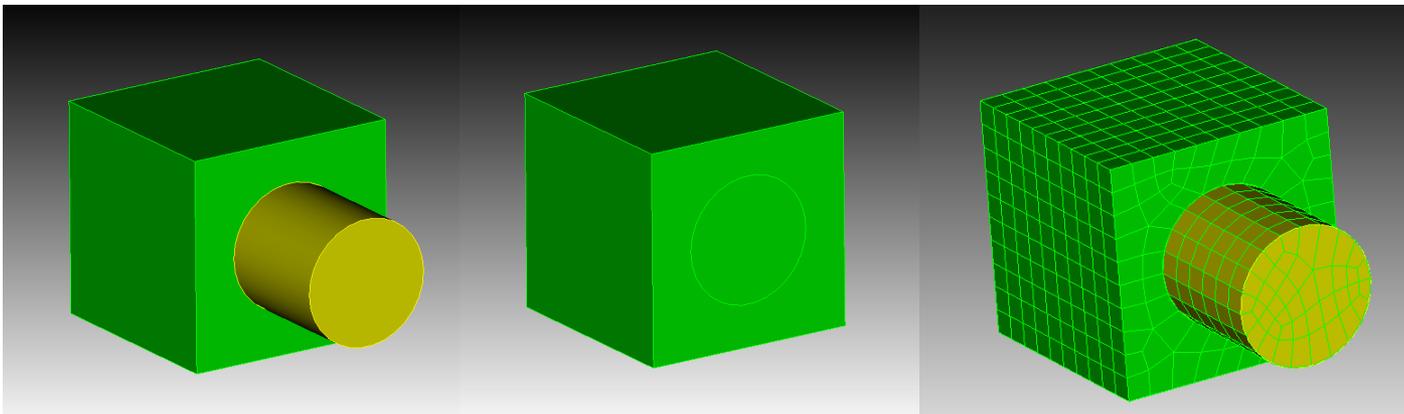
復習

インプリント&マージ (Imprint & Merge)

複数のボリウム(ボディ)で構成されるモデルを扱う場合、メッシュを生成する前にインプリント&マージを実行します。

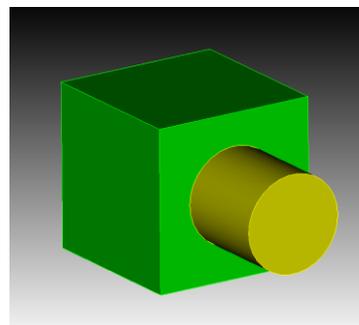
- ① インプリント: 接触しているボリウムの交線から共有面を作成します。
- ② マージ: 共有面を結合します。

これにより、複数のボリウム(ボディ)にまたがって連続したメッシュを生成することが可能になります。



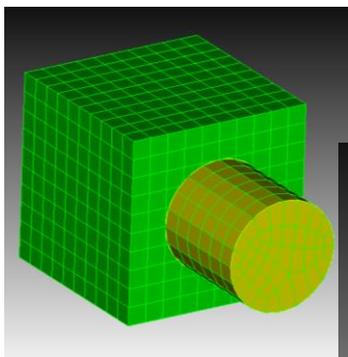
復習

インプリント&マージ

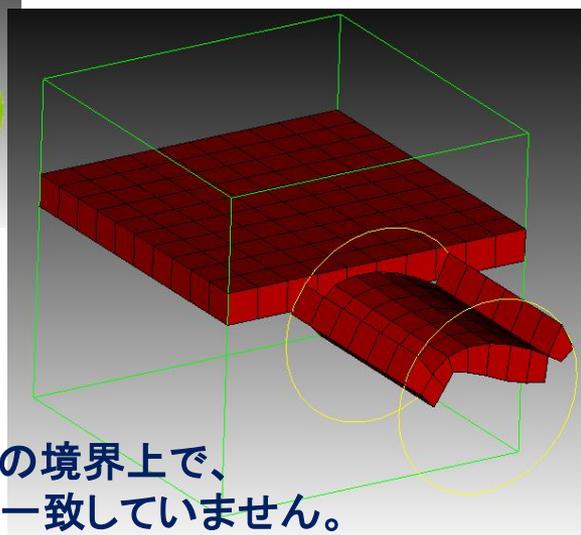


reset
brick x 10
create Cylinder height 6 radius 3
move Volume 2 z 8 include_merged

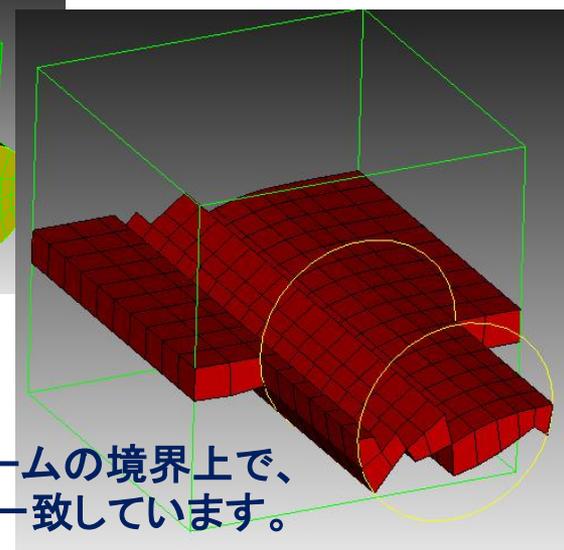
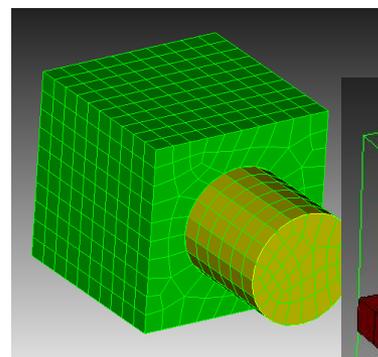
mesh vol all



imprint body all
merge body all
mesh vol all



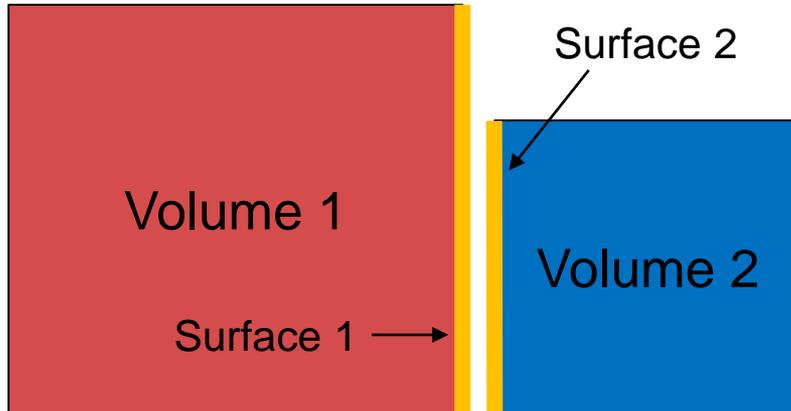
2つのボリュームの境界上で、
メッシュの配置が一致していません。



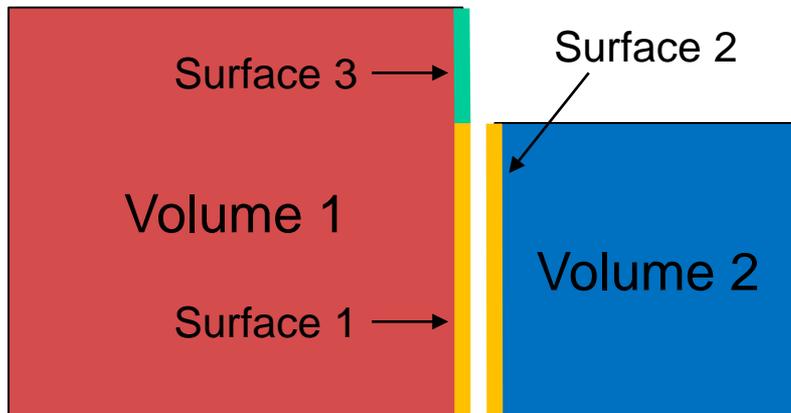
マージしたボリュームの境界上で、
メッシュの配置が一致しています。

復習

インプリント & マージ



※2つのボリュームは接しています。



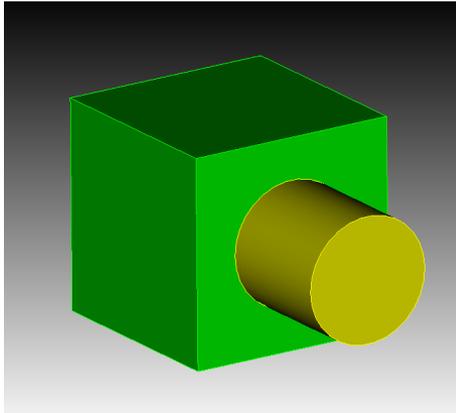
インプリント (Imprint)

インプリント操作は、カーブ、バーテックスなどのフィーチャーとの交線(交点)をサーフェス上に焼き付け(imprint)し、ボリューム間の共有面を作成します。

インプリントするフィーチャーは、ユーザが任意に指定することもできます。

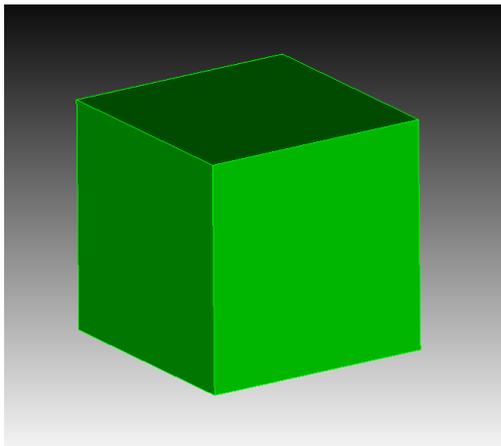
復習

インプリント&マージ

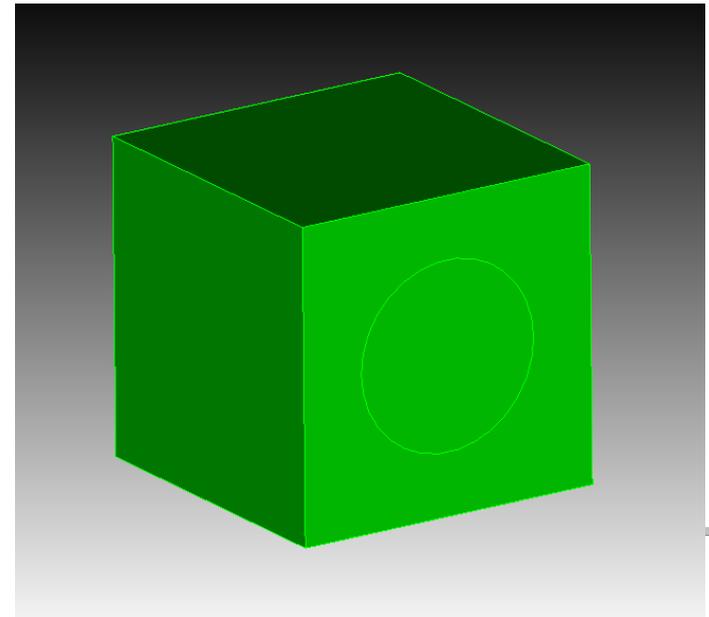


```
reset  
brick x 10  
create Cylinder height 6 radius 3  
move Volume 2 z 8 include_merged
```

draw vol 1



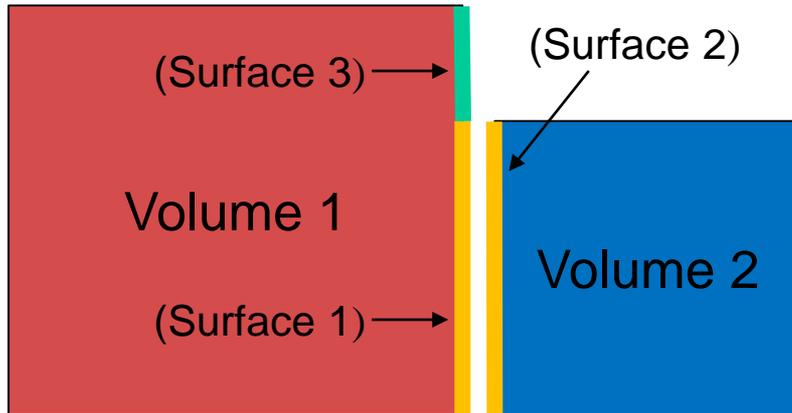
imprint body all



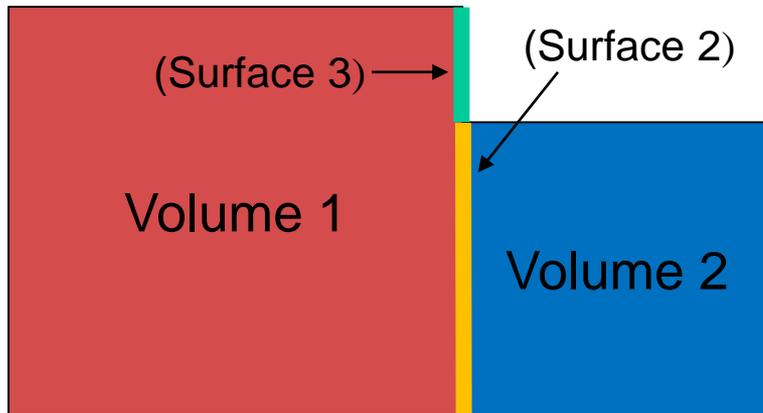
ボリューム1とボリューム2の共有面
が作成されました。

復習

インプリント & マージ

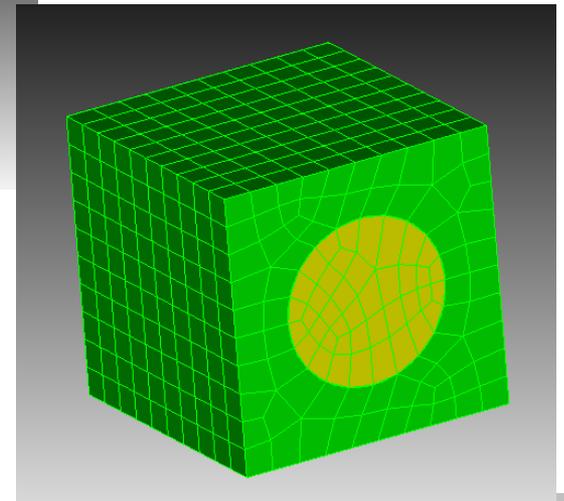
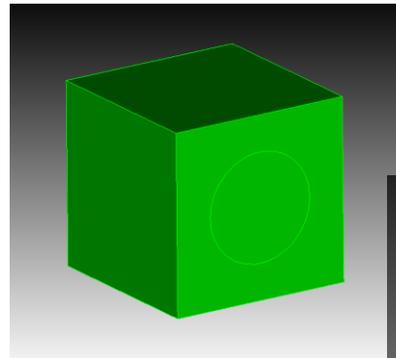


※2つのボリュームは接しています。



マージ (Merge)

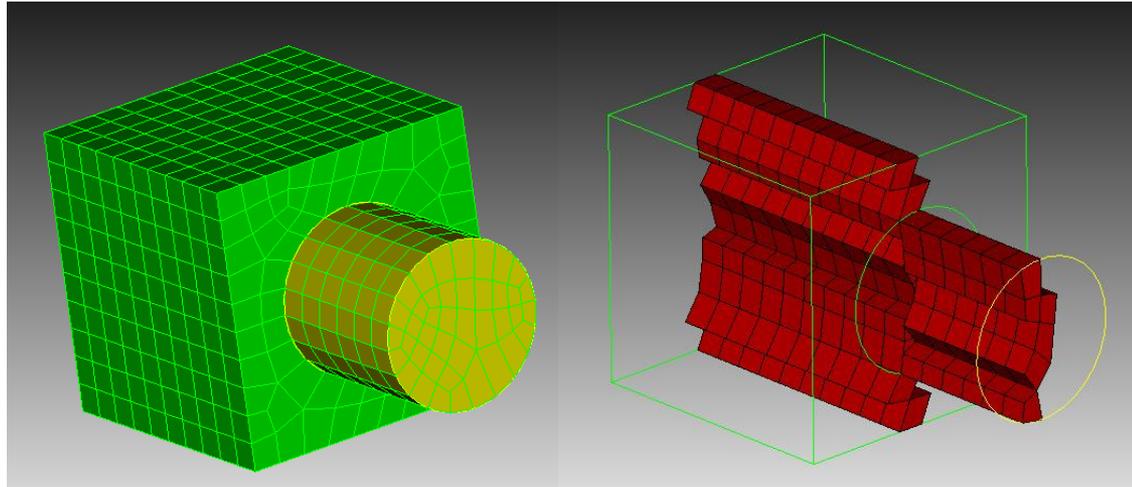
マージ操作では、隣接したボリューム間の共有面を結合し、連続したメッシュの生成を可能にします。



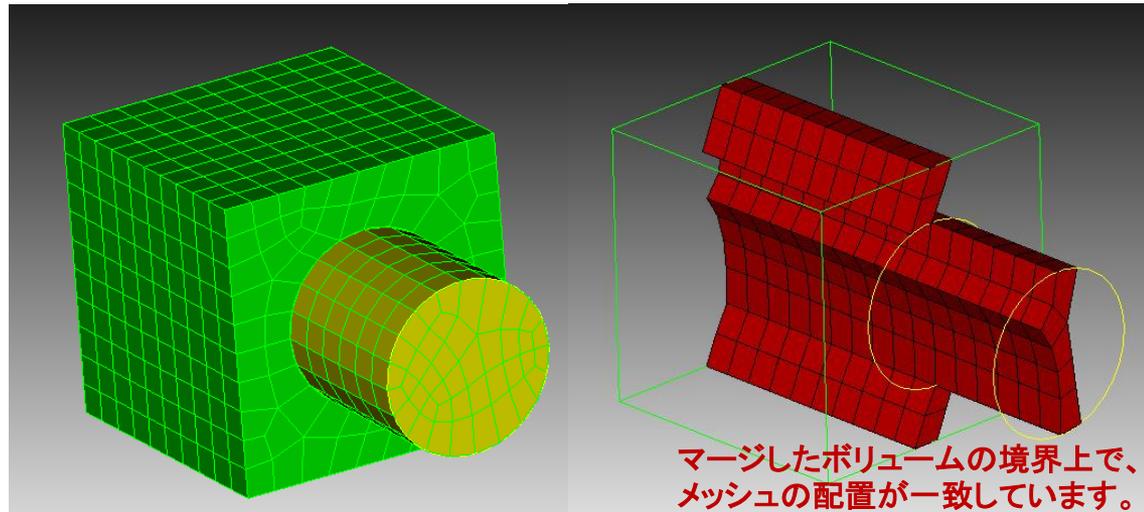
復習

インプリント & マージ

インプリント後
マージなし: mesh vol all



マージあり: merge body all
mesh vol all



マージしたボリュームの境界上で、
メッシュの配置が一致しています。

このチュートリアルで学んだこと

ヘキサメッシングのために

- Webcutツールで
 - Many-to-ManyのVolumeをMany-to-One / One-to-Oneに分解
 - 分解された各Volumeに対してスイープメッシング
- Imprintで
 - 隣接するVolume間の共有面を作成した
- Mergeで
 - 共有面を結合した

解析のために

- 要素のグルーピング機能で
 - 境界用に、Face要素をまとめてサイドセットを作成した
 - 領域用に、Hex要素をまとめてブロックを作成した

基本操作演習(別冊子) p.1

基本操作演習① Boolean

最重要！

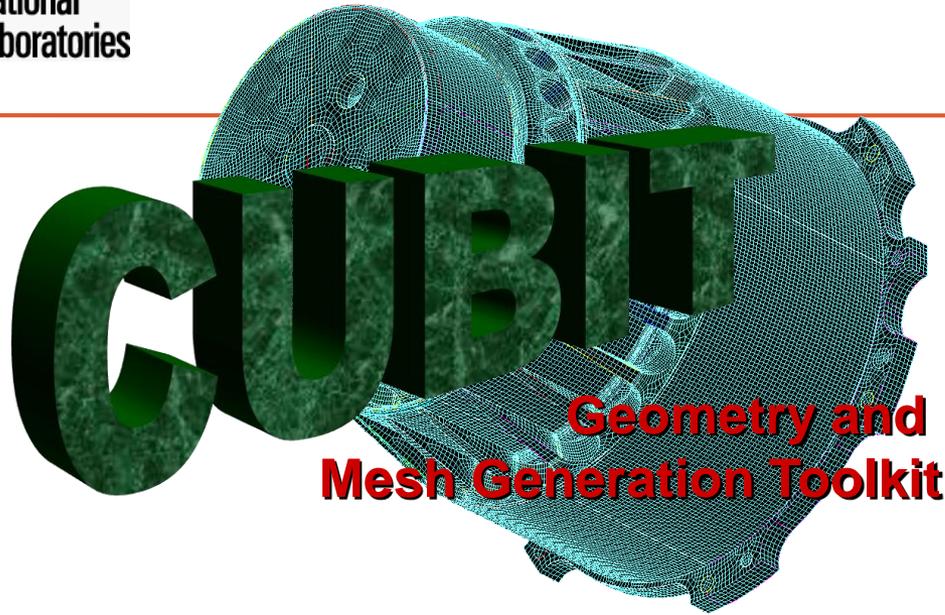
マウス操作／エンティティ選択方法は
理解するより体で覚えてください！

基本操作演習(別冊子) p.5

基本操作演習② ショートカット

最重要！

マウス操作／エンティティ選択方法は
理解するより体で覚えてください！



CUBIT ファーストスタートチュートリアル
6. ジオメトリツール

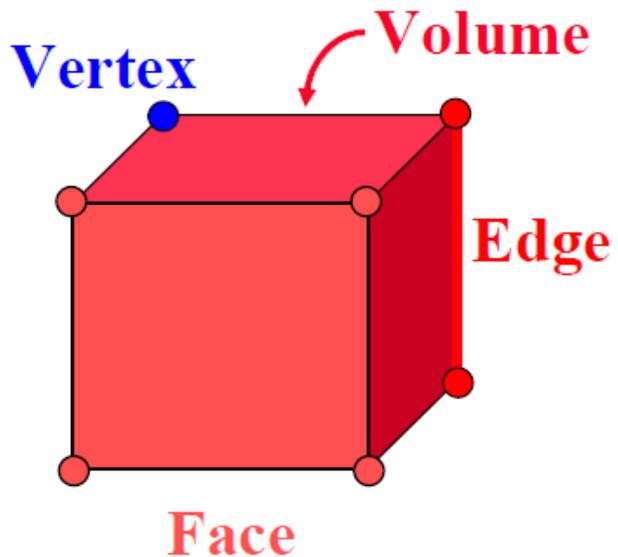
もしあなたがCADの開発者なら

モデルを作成するために
どのような操作／機能が必要か考えてみましょう。

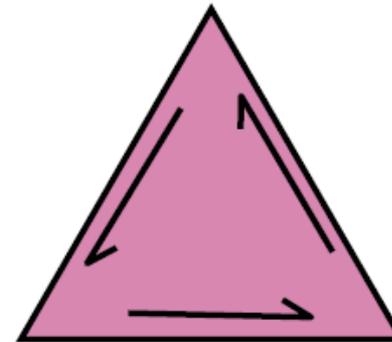
Topological Model

“Standard” BREP Entities

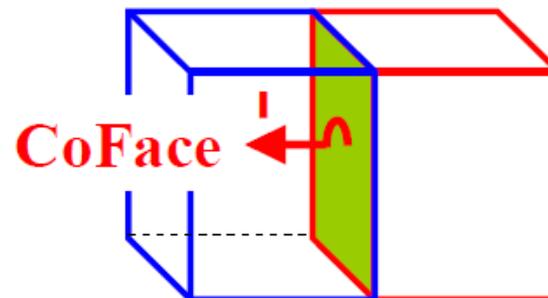
Basic:



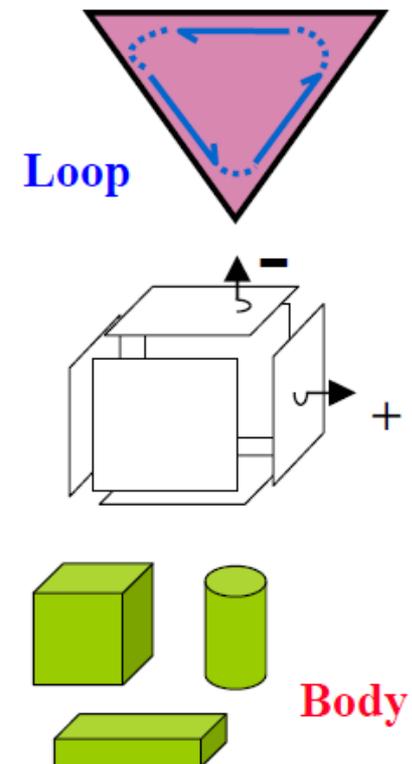
Sense:



CoEdge



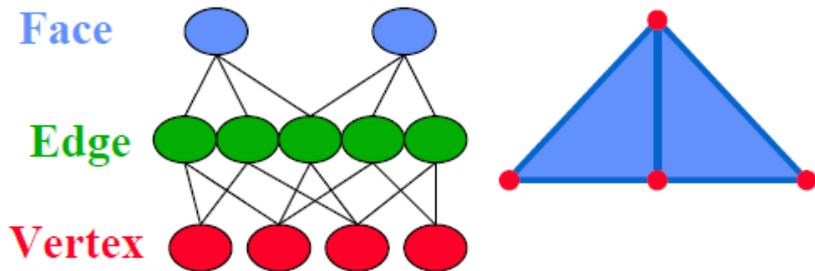
Grouping:



ジオメトリモデルの表現方法

BREP(境界表現)

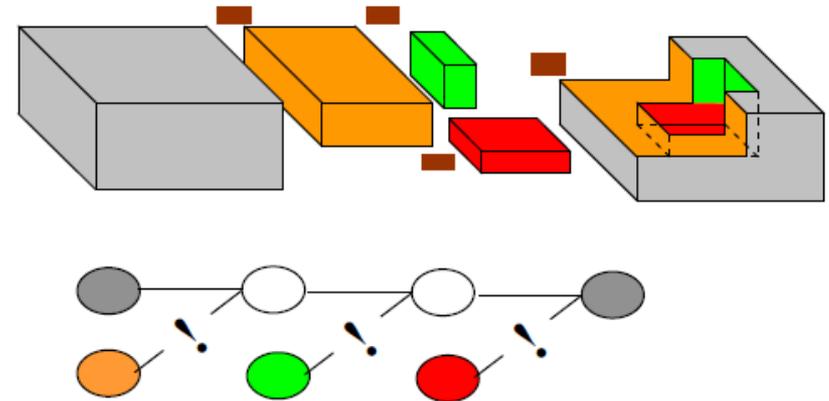
- モデルのトポロジはグラフで表現される。
- 構築履歴は保持されない。



- ACIS, Parasolid

CSG/Feature-Based Model

- 構築履歴がグラフに保持される。
- 実用上、モデルのBREPも保持される。



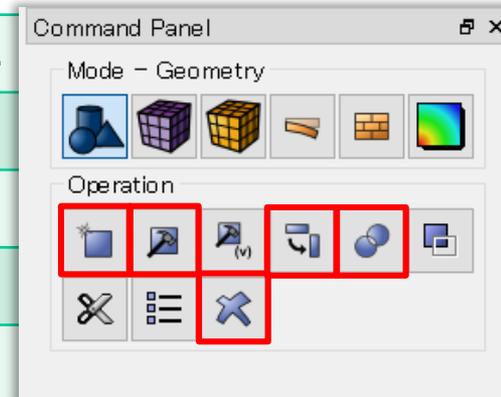
- Creo, SolidWorks

Trelis でモデルを作成する

TrelisのCADモデリング:

5つの基本操作 × 4つのエンティティ

① 作成 (Create)	プリミティブ, スweep, コピー, etc.
② 修正/編集 (Modify)	フィーチャーの除去, etc.
③ 座標変換 (Transform)	移動, 回転, スケール, etc.
④ ブーリアン (Boolean)	交差, 結合, 減算
⑤ 削除 (Delete)	



他のCADソフトも、基本的な操作は同じです。

※メッシュ生成・コントロールのためのジオメトリ機能:

⑥分解 (Decomposition)

⑦バーチャルジオメトリ (Virtual Geometry)

×



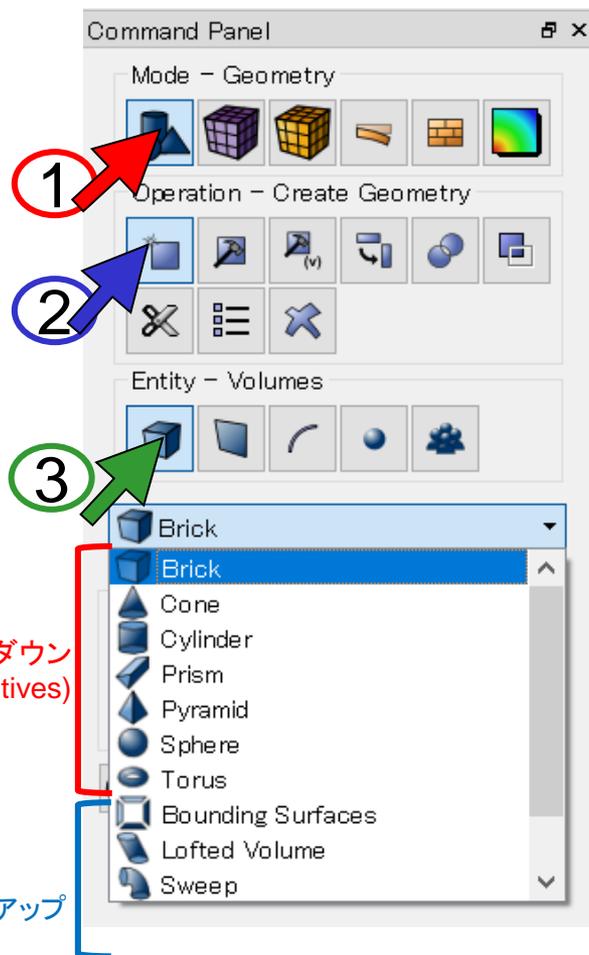
①ボリューム

②サーフェス

③カーブ

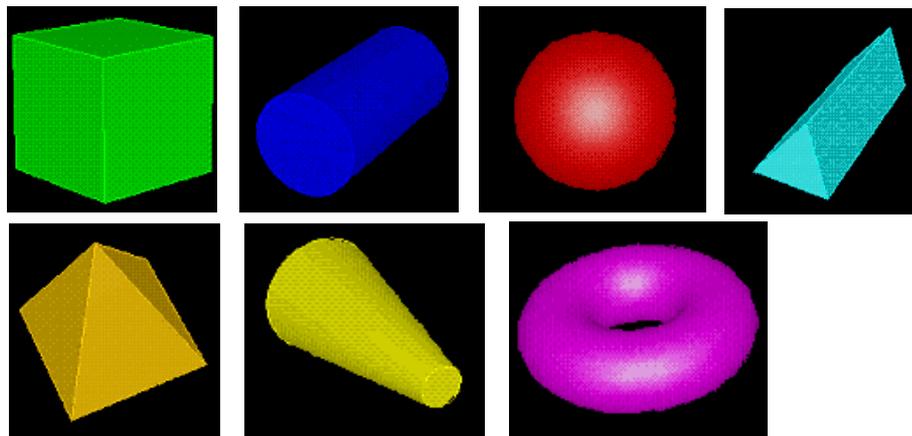
④バーテクス

①ジオメトリの作成(Create)



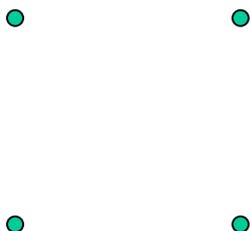
- ① *Mode*→*Geometry*をクリック
- ② *Operation*→*Create Geometry* をクリック
- ③ *Entity*→*Volumes* をクリック

ジオメトリの Primitives

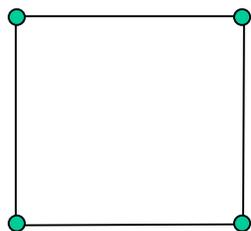


作成されるプリミティブの初期位置は原点(0, 0, 0)です。
transform を使って位置や方向を変更しましょう。

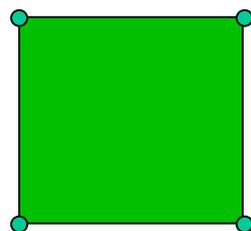
ボトムアップでのジオメトリ作成



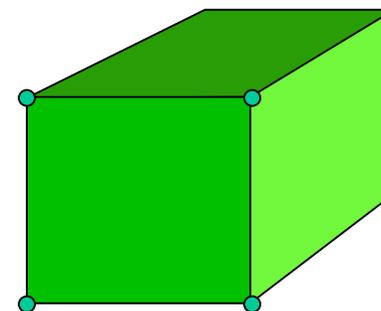
① まずバーテックスを作成します。



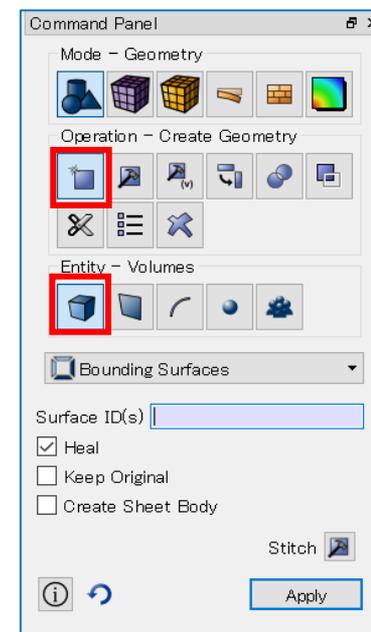
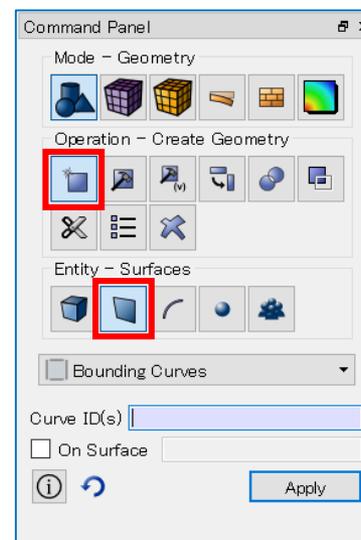
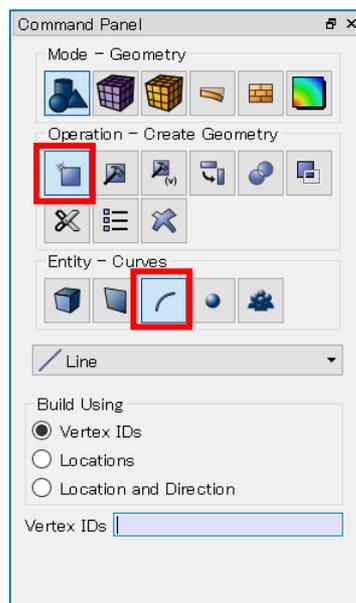
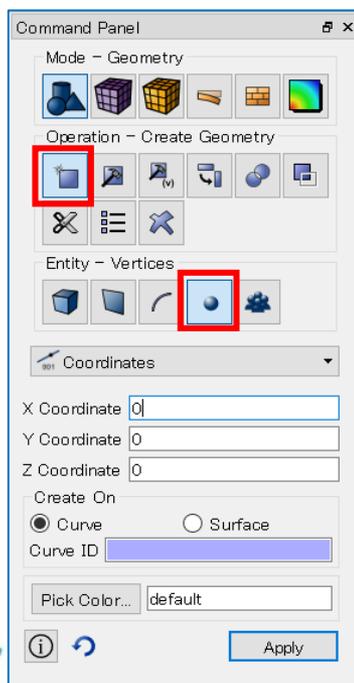
② バーテックスをカーブでつなぎます。



③ カーブの輪からサーフェスを作成します。

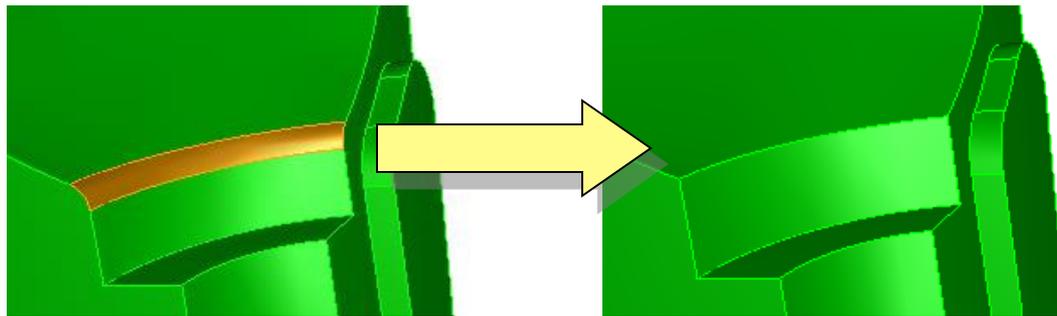


④ 閉じたサーフェスの集合からボリュームを作成します。



②ジオメトリの修正／編集 (Modify)

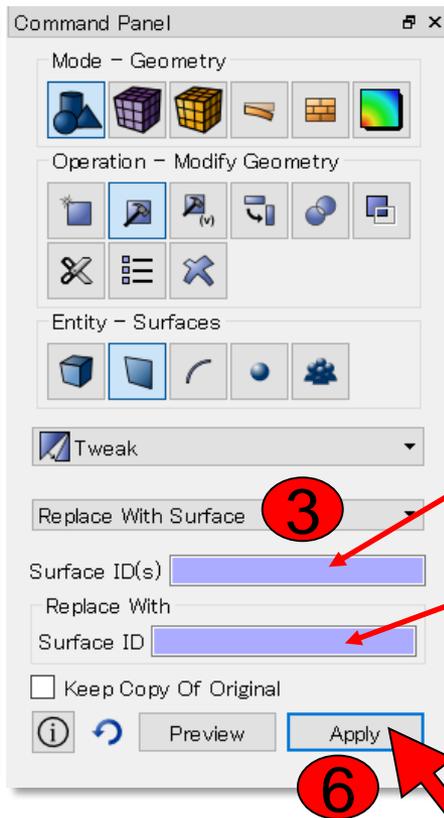
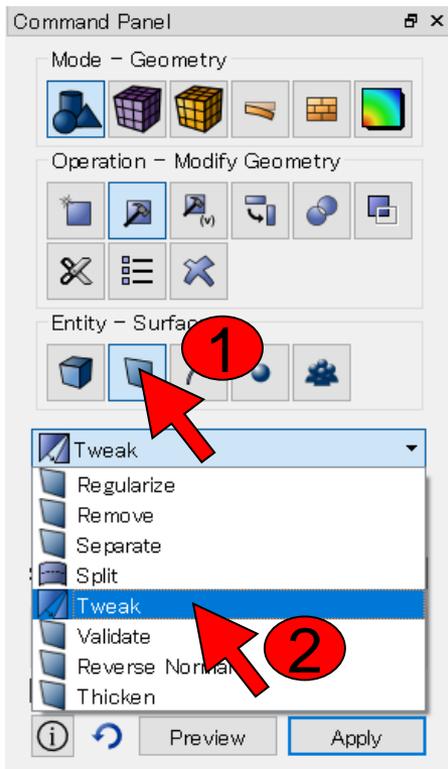
- サーフェスの移動・取り換えができます。
- 隣接したジオメトリは必要に応じて修正されます。
- こんな場合に便利：
 - 隙間などの問題の修正
 - フィーチャーの除去
 - スイープ可能にする
- “リアル”な操作 (バーチャルでない)
 - CADモデルに直接変更を加えます



例：フィレットの除去

サーフェスの除去 (Replace)

① Geometry→Modify→Surface



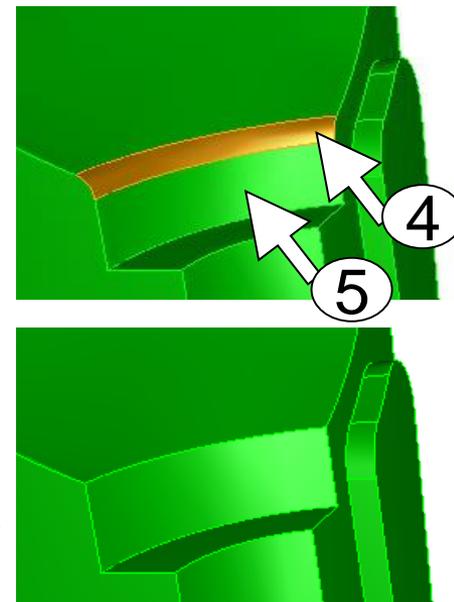
② Tweakを選択

③ Replace With Surface を選択

④ 除去するサーフェスを選択

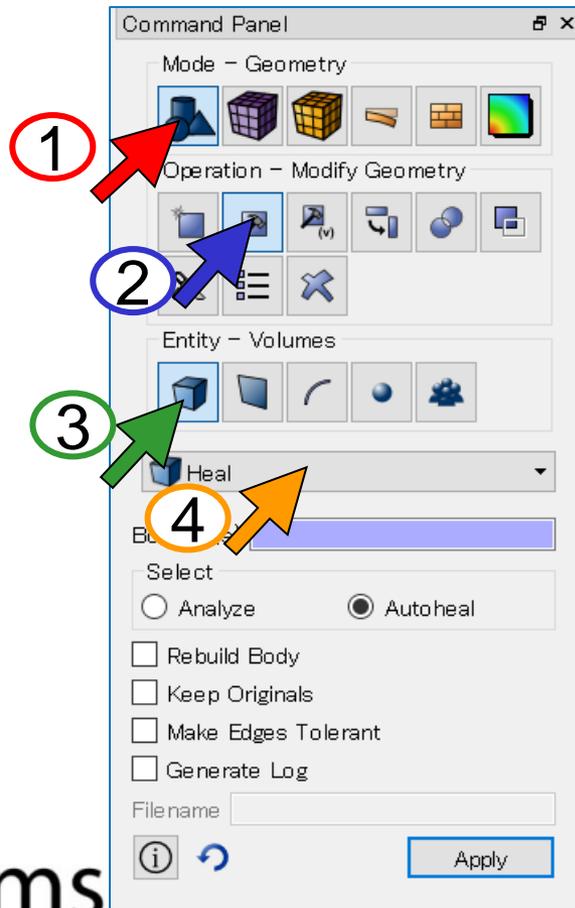
⑤ 隣接するサーフェスを選択

⑥ Applyをクリック



ヒーリング

- ジオメトリのACISフォーマットへの変換は度々問題を生じます。
 - デフォルトのトレランスだと、Creo は ACIS より不正確です。
 - 隙間、オーバーラップ、内部不一致はよく起こります。
- ACIS Healing Huskなら多くの問題を修正できるでしょう。



① *Mode*→*Geometry* をクリック

② *Operation*→*Modify* をクリック

③ *Entity*→*Volumes* をクリック

④ ドロップダウンメニューから *Heal* を選択

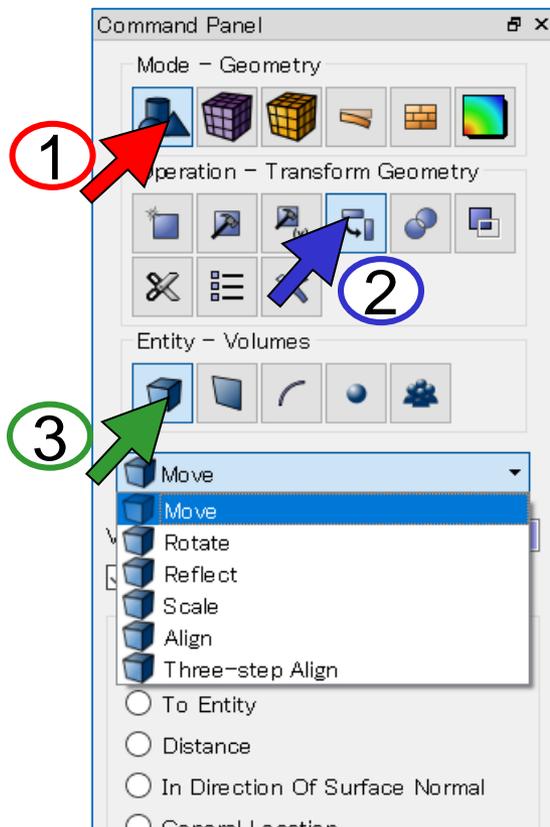
Analyze

選択したボリユームの問題点の一覧を表示します。

AutoHeal

問題の修正を試みます。

③ジオメトリの座標変換 (Transform)



- ① *Mode*→*Geometry* をクリック
- ② *Operation*→*Transform* をクリック
- ③ *Entity*→*Volumes* をクリック

- Align** 他のエンティティと整列させます。
- Move** 指定した方向に平行移動させます。
- Reflect** 指定した平面に対して対称の位置・姿勢となるよう動かします。
- Rotate** 指定した軸を中心に回転させます。
- Scale** x、y、z軸のファクターで拡大・縮小します。

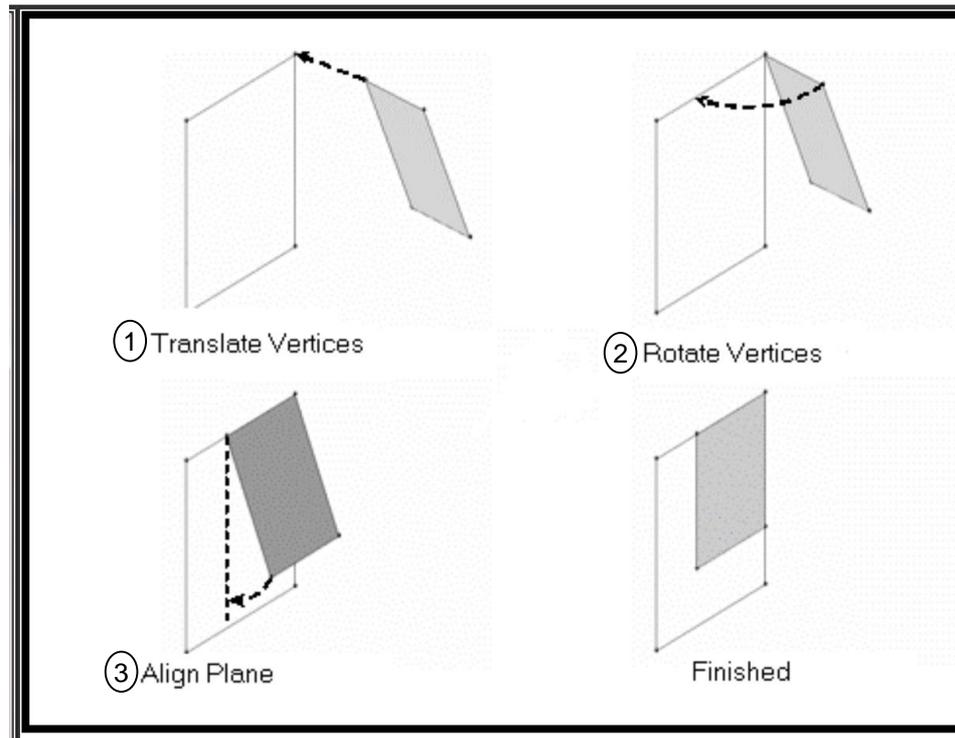
CUBIT は無単位です。

ヒント:

Scale を使って単位を変更します。

Align / 3-step Align

移動と回転の組み合わせにより、エンティティの位置や姿勢を、他のエンティティと揃えることができます。

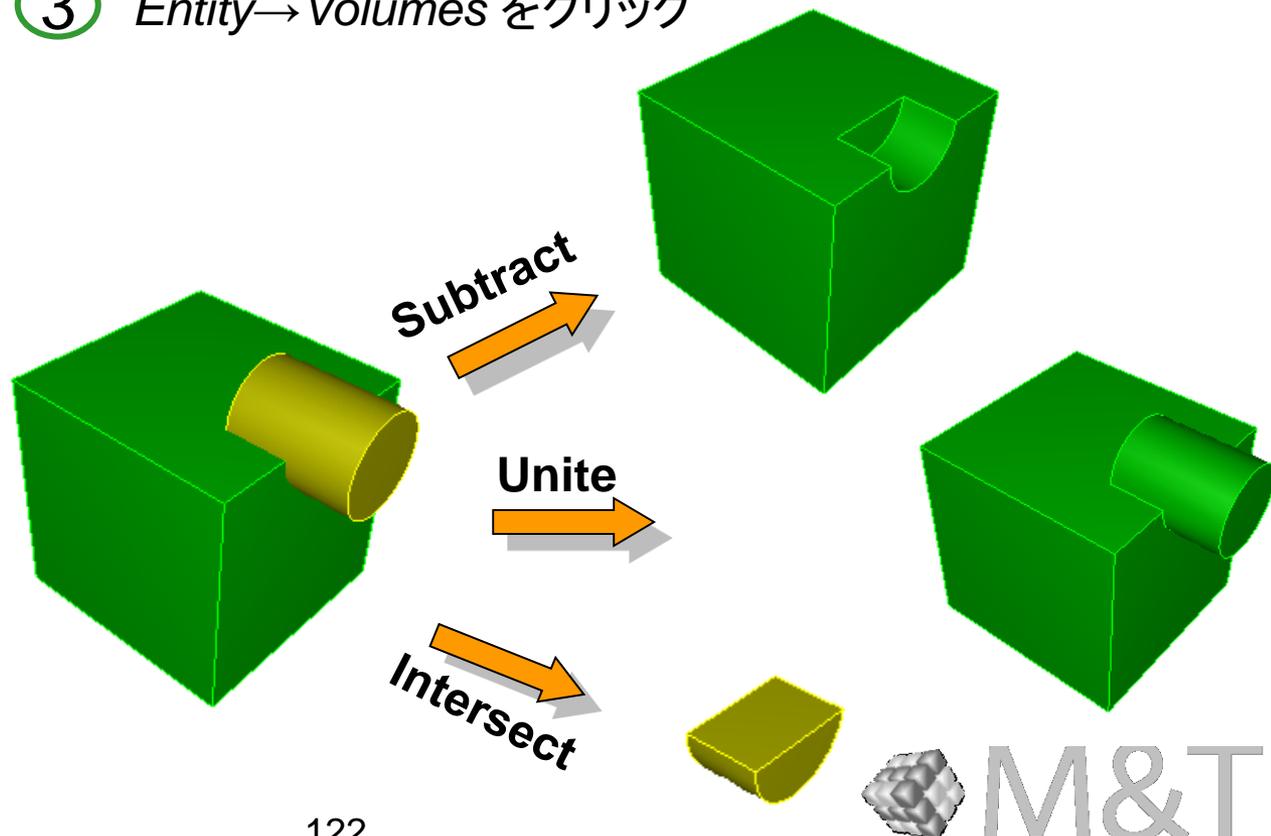
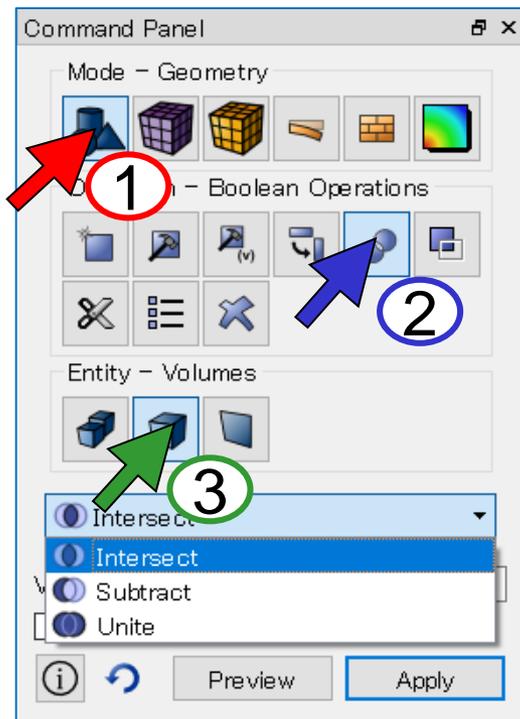


3-step Align: 選択したサーフェスの角と、他のサーフェスの角が揃うように座標変換しています。

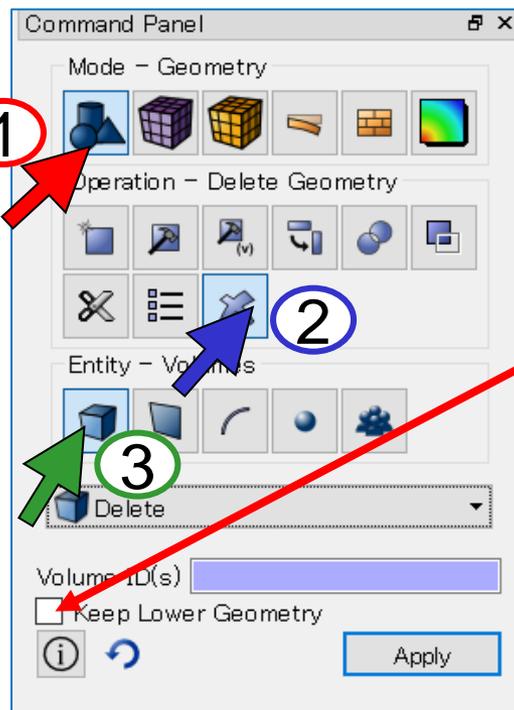
④ブーリアン(Boolean)

ボリウム間でブーリアン操作を実行します。

- ① *Mode*→*Geometry* をクリック
- ② *Operation*→*Boolean* をクリック
- ③ *Entity*→*Volumes* をクリック

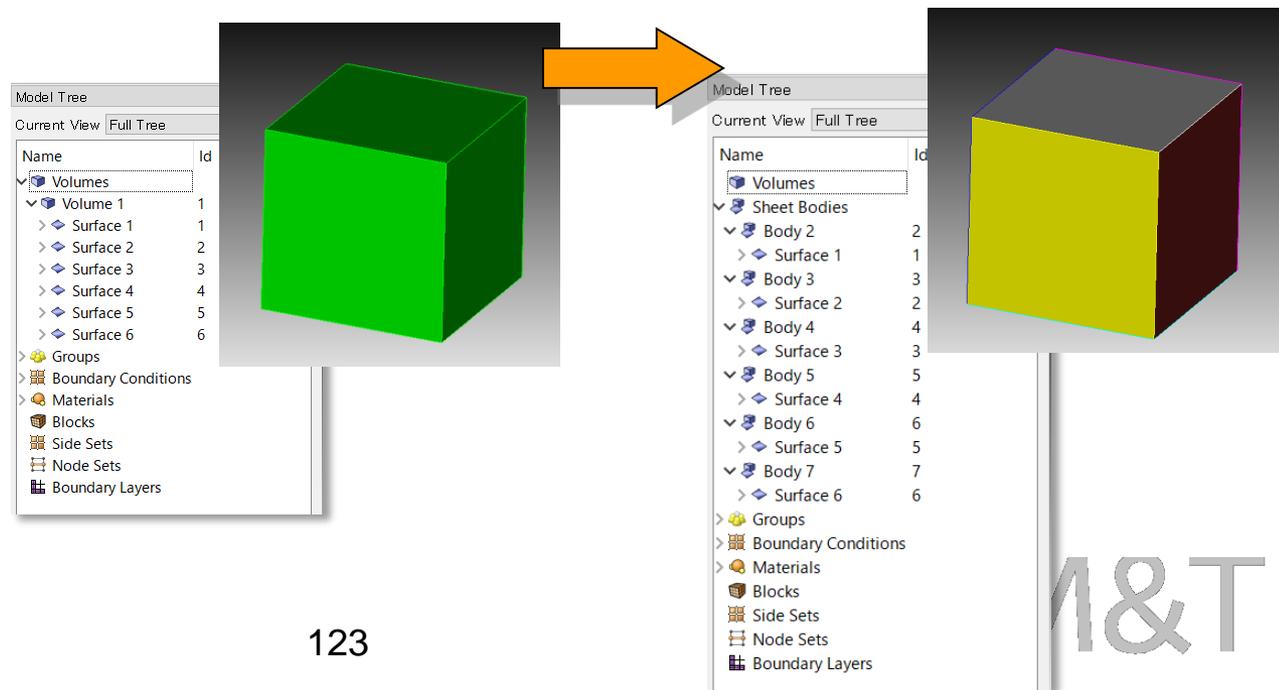


⑤ ジオメトリの削除 (Delete)



- ① Mode → Geometry をクリック
- ② Operation → Delete Geometry をクリック
- ③ Entity → Volumes をクリック

Keep Lower Geometry オプションを選択すると、元のエンティティを構成していた低次元エンティティのみを残すことができます。



⑥ ジオメトリの分解 (Decomposition)

メッシュを生成・コントロールするために、ジオメトリの分解が必要になることがあります。

CUBITでは、ウェブカット操作とスプリット操作によって、ジオメトリを分解することができます。

ウェブカット (Webcut) 操作: 

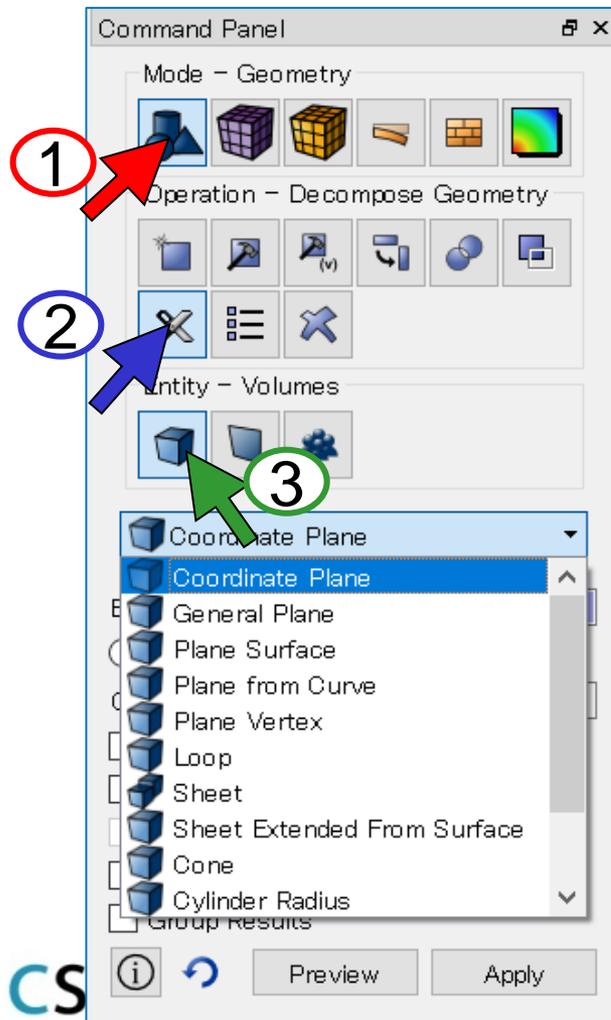
品質の良いヘキサメッシュを生成するために、ボディ(ボリューム・シートボディ)を分解します。

スプリット (Split) 操作: 

メッシュ形状をコントロールするために、サーフェス・カーブを分解します。新たなカーブ・バーテックスを作成することなく、インプリントと同様の操作を可能にします。

ウェブカット (Webcut)

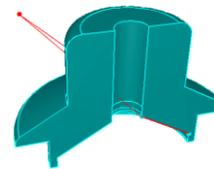
既存のジオメトリや、コマンド内で定義する切断面を用いて、ボリューム（またはシートボディ）を分解します。



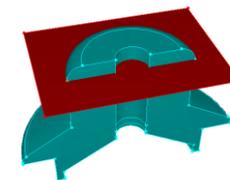
- ① *Mode*→*Geometry* をクリック
- ② *Operation*→*Decompose Geometry* をクリック
- ③ *Entity*→*Volumes* をクリック

切断面の定義方法(例):

バーテックスで定義
(Plane from Vertex)



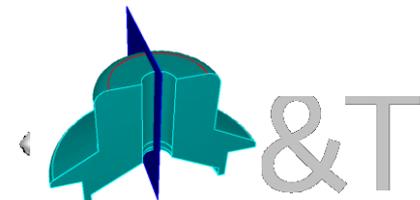
座標平面
(Coordinate Plane)



円筒面
(Cylinder Radius)



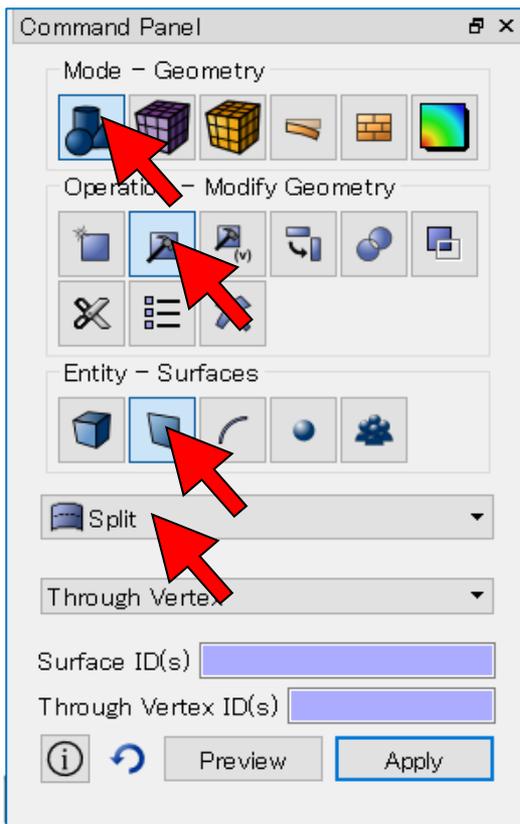
カーブに垂直な平面
(Plane from Curve)



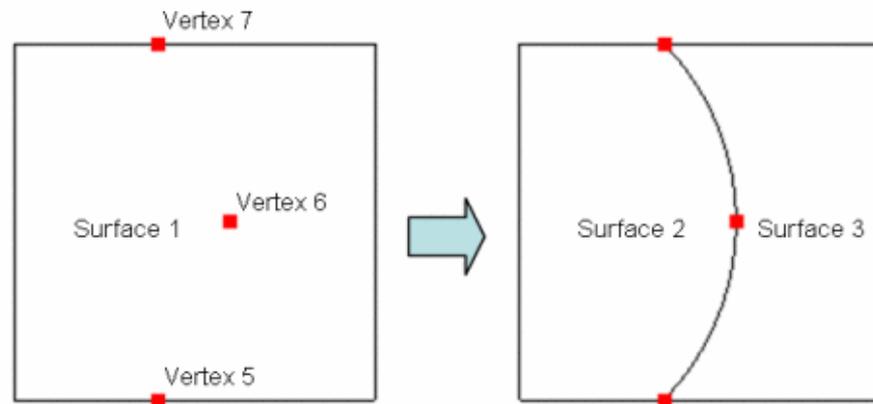
スプリット (Split)

サーフェス・カーブを分割することで、メッシュのコントロールを行います。

- | | |
|-----------------------|-------------------|
| ・ カーブ | … Entity- Curve |
| ・ 周期サーフェス (円筒/トーラス/球) | … Entity- Volume |
| ・ サーフェス | … Entity- Surface |



Geometry-Modify-Surfaces/Volumes の後に *Split/Split Periodic* をクリックします。



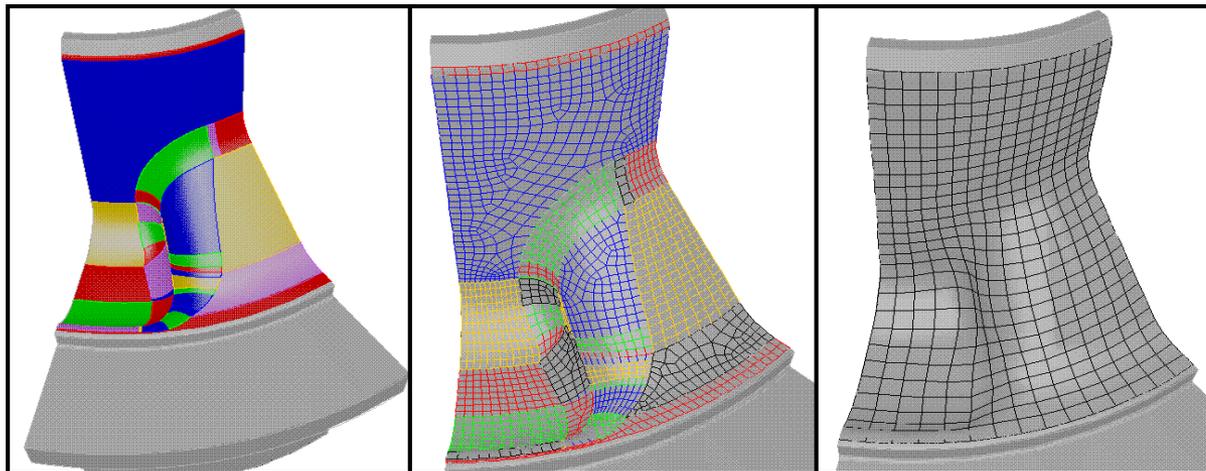
⑦バーチャルジオメトリ (Virtual Geometry)

バーチャルジオメトリは、CADのカーネルが定義できない形状を、強制的・仮想的に定義する機能です。

バーチャルジオメトリは、メッシュ作成時のメッシュコントロールにご利用ください。モデリングにおける形状定義には推奨しません。

コンポジット(Composite)操作:

- ・サーフェス同士を**組合**せます。
- ・メッシュは境界のカーブに拘束されません。
- ・メッシュ品質の改善に使用します。
- ・元のCADモデルには変更を加えません。



元のCADモデル

メッシュ(コンポジットなし)

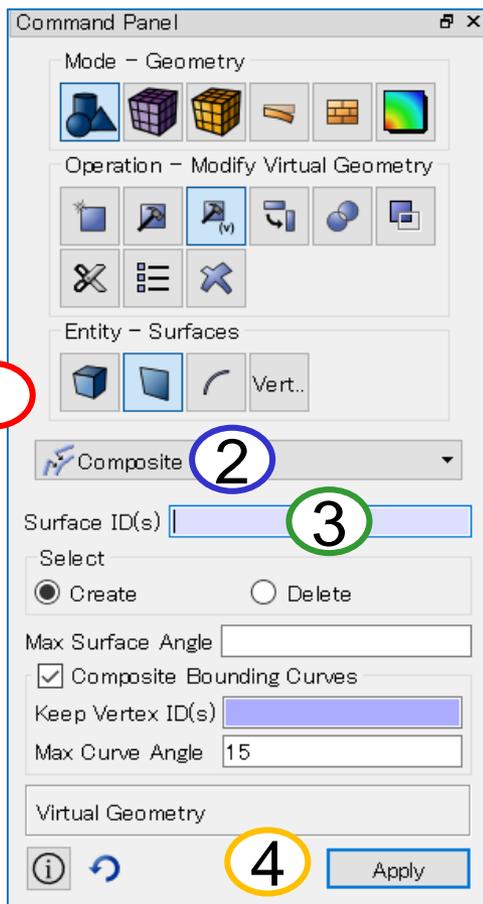
メッシュ(コンポジット後)

・“リアル”: ソリッドモデラー
カーネルによるCAD操作

・“バーチャル”: CUBIT CGM
による、トポロジーの変更

コンポジットサーフェスの作成

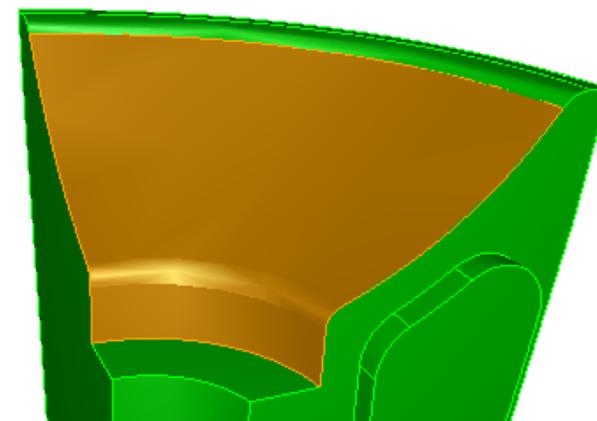
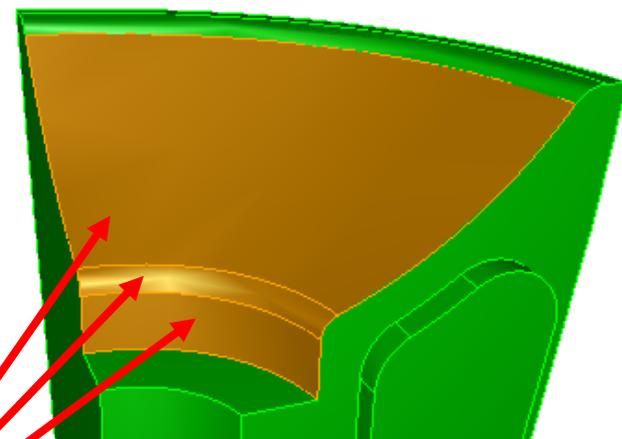
① *Geometry*→*Modify Virtual*→*Surfaces*

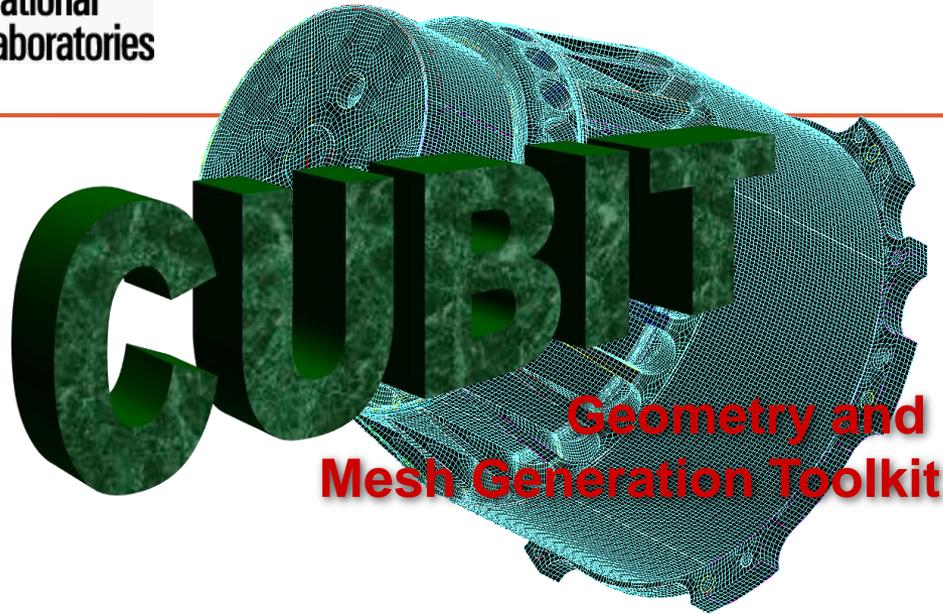


② *Composite*を選択

③ サーフェスを選択

④ *Apply*をクリック





**CUBIT ファーストスタートチュートリアル
“Mesh Vol All”**

Cubitを使いこなす上での ポイントを説明します。

コンセプトの理解が重要です

コマンド: *mesh vol all* を使うために セミオートメッシャーの活用

CUBITでは、*mesh vol all* コマンドでメッシュ作成を行うために、モデルを編集します。

1. エンティティのプロパティ

- ①プロパティの設定
- ②プロパティの編集
- ③プロパティを元にメッシュ作成 (*mesh vol all*)

Cubitは、デフォルトのプロパティ情報を参照してメッシュ生成を試みます。

- プロパティ情報が**適切**であれば、**成功**します。
- 失敗した場合、適切なプロパティ情報を指定する必要があります。

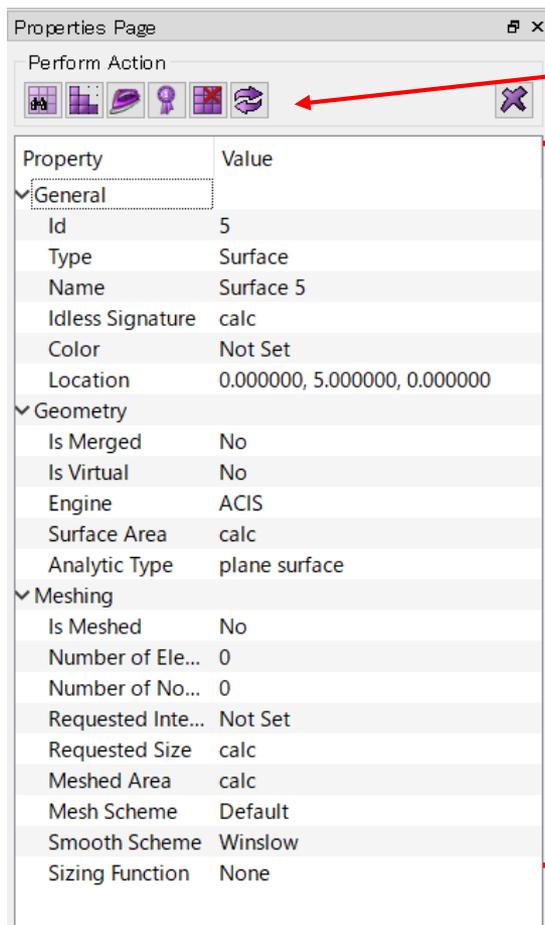
ユーザーが行うことは、

- **適切なプロパティを設定すること**
- **利用可能なメッシングスキームのためにボリウムを定義／分割すること**

に尽きます！！

コマンド: *mesh vol all* を使うために セミオートメッシャーの活用

・プロパティページ



Action ボタン

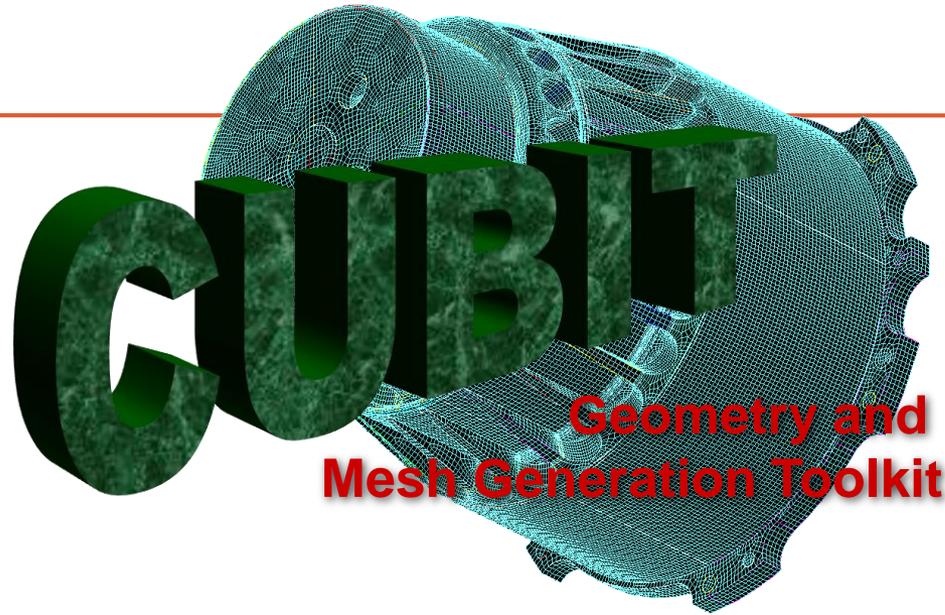
-  Preview (プレビュー)
-  Mesh (メッシュ作成)
-  Smooth (スムージング)
-  Quality (品質チェック)
-  Delete Mesh (メッシュを削除)
-  Reset Entity (エンティティをリセット)
-  Delete Entity (エンティティを削除)

エンティティのプロパティ
一部のプロパティはここから変更できます。

課題:モデル作成(30分)

ここまでの講義内容を踏まえて、ご自分の課題のモデルを作成しましょう。

テトラメッシュを扱う方は、作成したモデルへのメッシングも行ってみましょう。



CUBIT ファーストスタートチュートリアル

12. メッシュコントロール

メッシュコントロール

Trelisにおけるメッシュコントロールの方法

- ユーザによるメッシュ制御

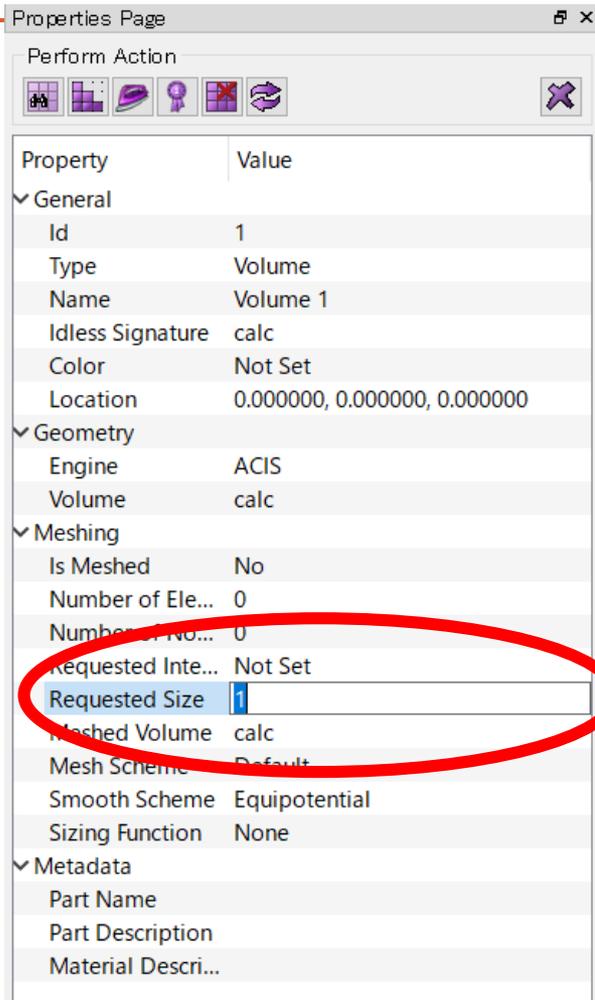
メッシュサイズやバイアスの指定、メッシュのリファインなど

- スムージング

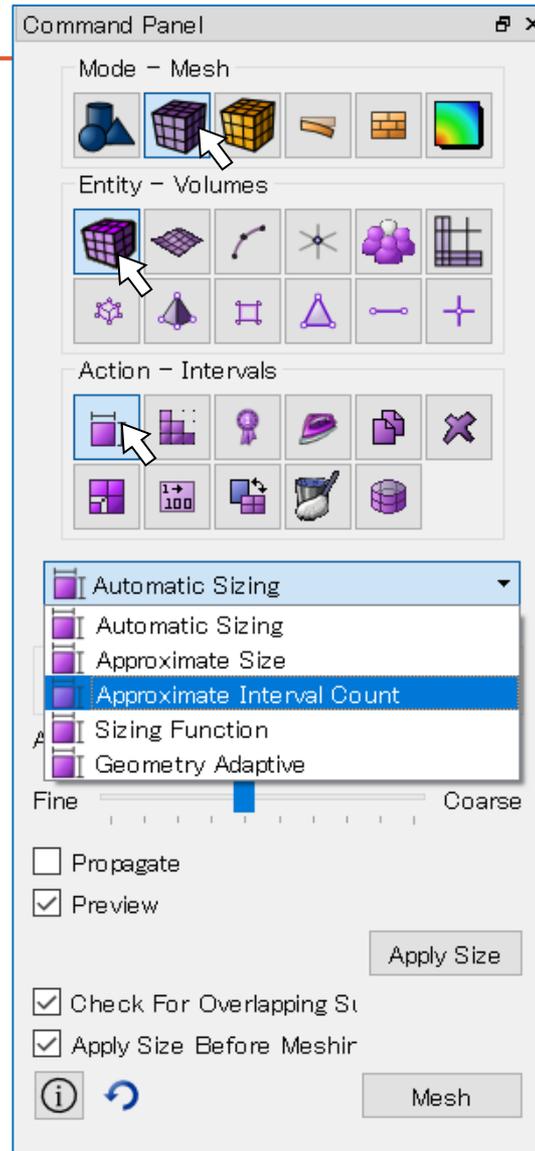
様々なスキームによるメッシュ品質の向上

メッシュ品質は、メッシュ品質確認ツールで調べます。

メッシュサイズ



メッシュサイズはプロパティパネルからジオメトリエンティティに設定できます。



メッシュサイズはジオメトリの属性です。エンティティを選択して、属性を設定します。

Approximate Size: エッジの長さを一定に保つようにメッシュ生成します。

Interval Count: カーブ上のインターバル数を規定します。

Automatic: 相対値

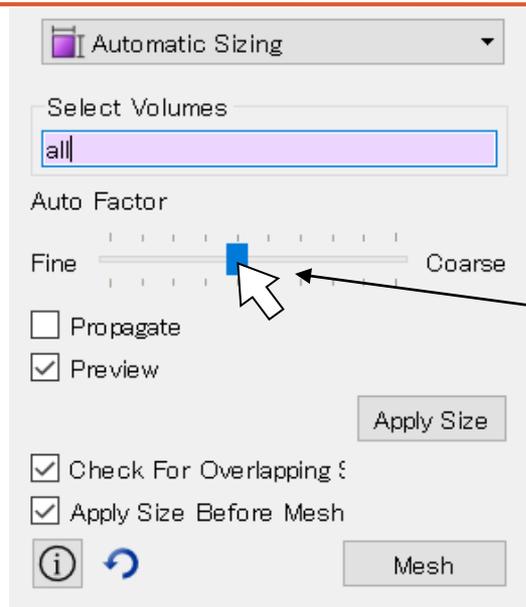
1 = 精細メッシュ

10 = 粗いメッシュ

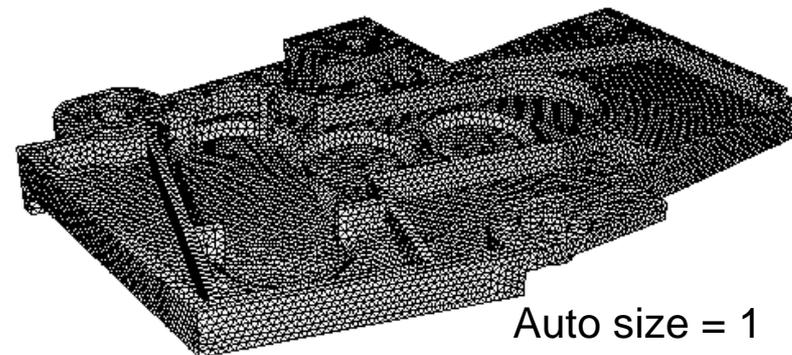
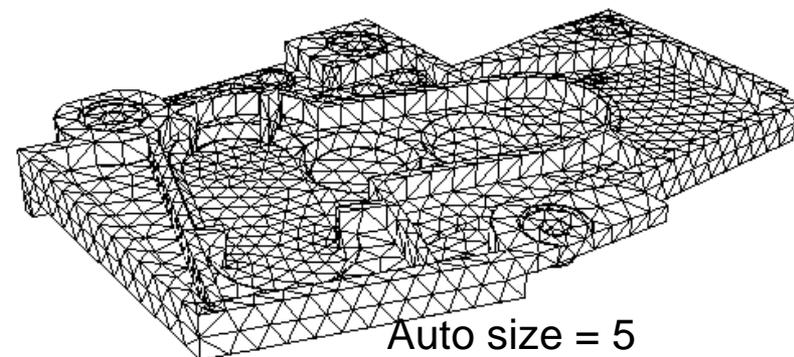
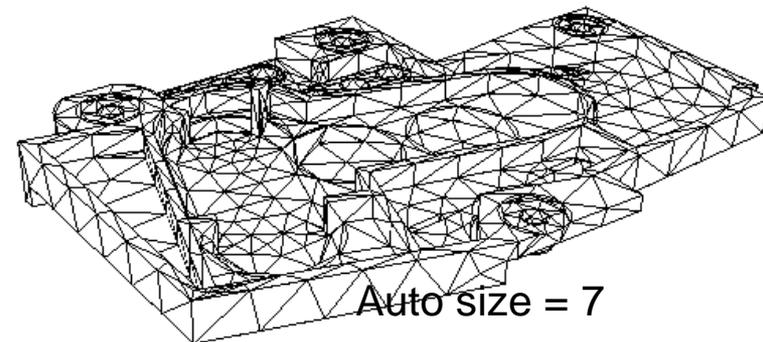
プレビュー機能でメッシュサイズの確認ができます。

Geometry Adaptive: ジオメトリの特徴にあわせて要素サイズを決めます。

自動メッシュサイズ



スライダーを移動させて
カーブ上のメッシュのプ
レビューを見ます。



要素の正確なサイズを知る必要
はありません。

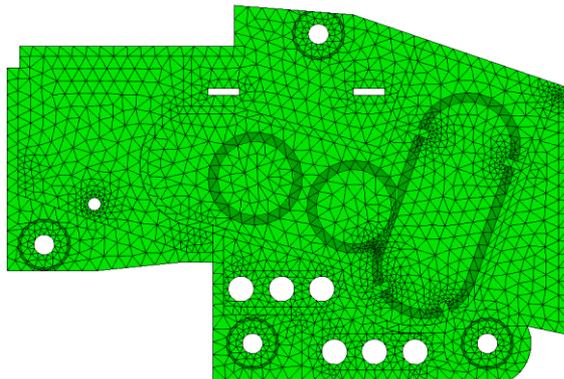
試行錯誤を行って要素のサイズ
を求めます。

サイズを設定しない場合、
CUBITが自動的に”Auto mesh
size=5”と設定します。

ジオメトリアダプティブ メッシュコントロール エクササイズ 1

- ① newを選んでcad6.sat をインポートします。
- ② テトラメッシュでボリウムをメッシングします。
- ③ いくつかのテトラ要素が生成されましたか？
- ④ 品質を検査します。
- ⑤ 最も品質の悪い要素を特定してズームします。

なぜ品質が悪いのでしょうか？
改善方法を考えてみましょう。



Vol 1 scheme tetmesh
Mesh vol 1

List totals

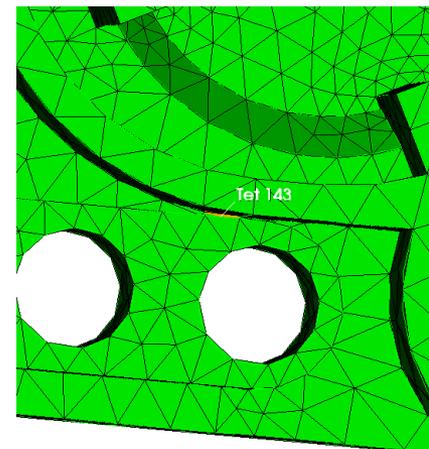
Quality Vol 1

Shape 値が最小な tet 要素
の id を入力します。

Highlight tet _____

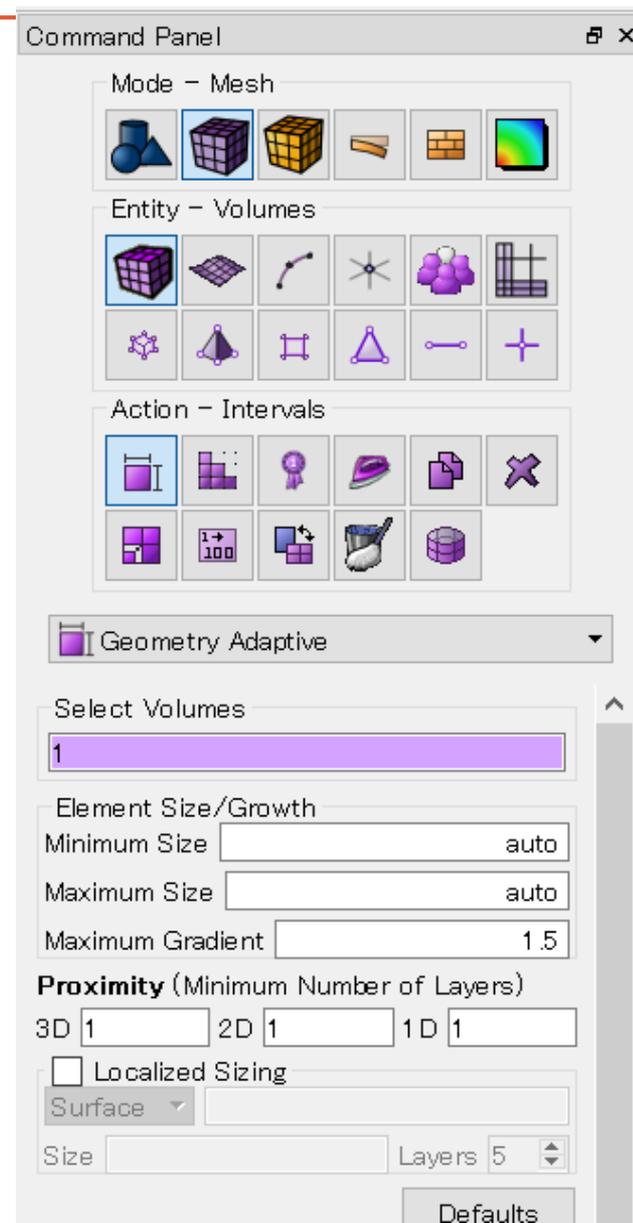
Locate tet _____

Zoom tet _____

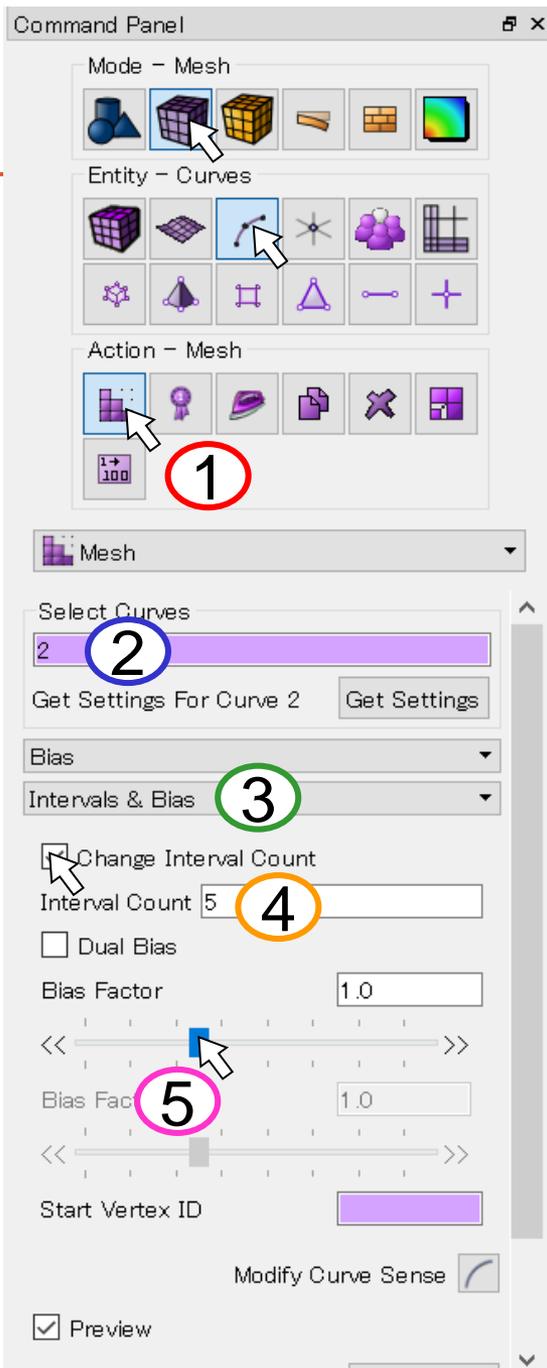


ジオメトリアダプティブ メッシュコントロール エクササイズ 1

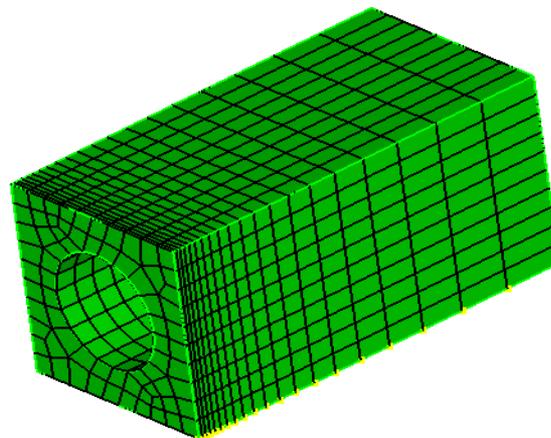
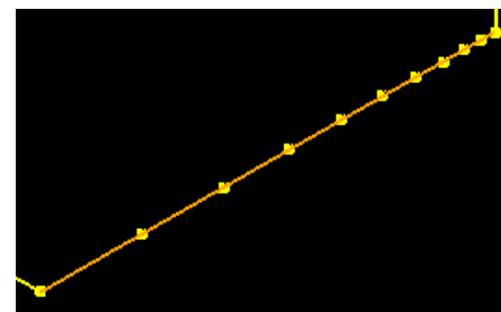
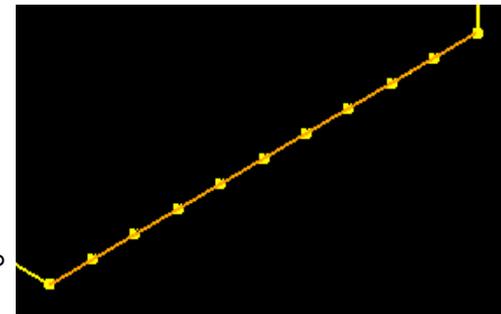
- ⑥ メッシュを削除します。
- ⑦ Geometry-adaptiveサイズ機能を適用します。
- ⑧ もう一度テトラメッシュでボリュームをメッシングします。
- ⑨ いくつかのテトラ要素が生成されましたか？
- ⑩ メッシュ品質をチェックして、前回のメッシングで品質の悪い要素があった箇所を調べます。
- ⑪ このモデルのジオメトリのアダプティブサイズへの別のオプションを試してみましょう。
- ⑫ 時間があれば、**driver.sat** のモデルで同じ手順を試してみましょう。



カーブバイアス



- ① カーブバイアスパネルを開きます。
- ② カーブを選択します。
- ③ Bias スキームを選択します。
- ④ インターバルの数や開始点などを設定します。
- ⑤ スライダーを動かして、バイアスメッシュのプレビューをみます。



Propagate Curve Bias

GUI パネルでBiasスキームから選択します。

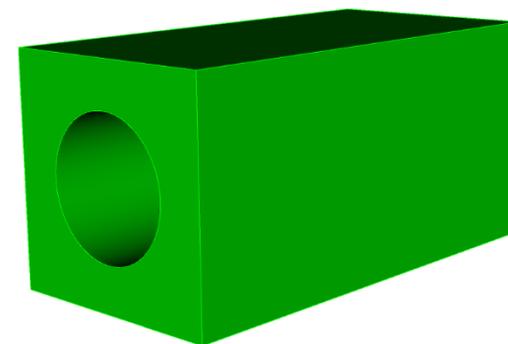
③



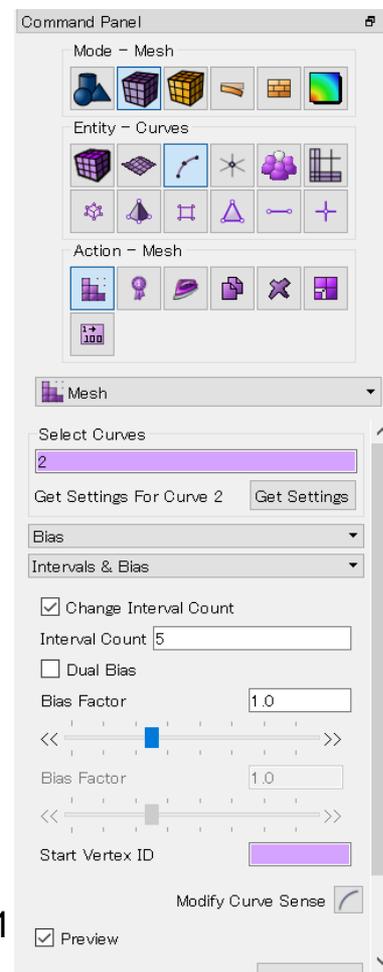
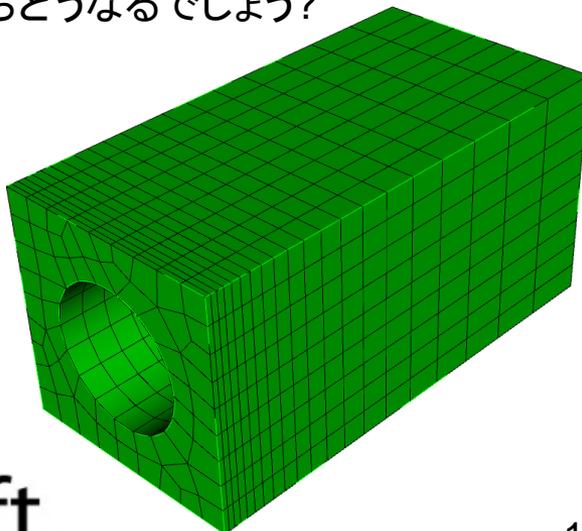
カーブバイアス エクササイズ

- ① GUI かコマンドラインを使って、図に示した簡単なモデルを作成します。
- ② 前のスライドに示した手順で、バイアスメッシュを生成します。
- ③ カーブバイアスの別のオプションを試してみましょう。

Brick x 10 y 10 z 20
Create cylinder height 25 rad 3
Subtract vol 2 from vol 1

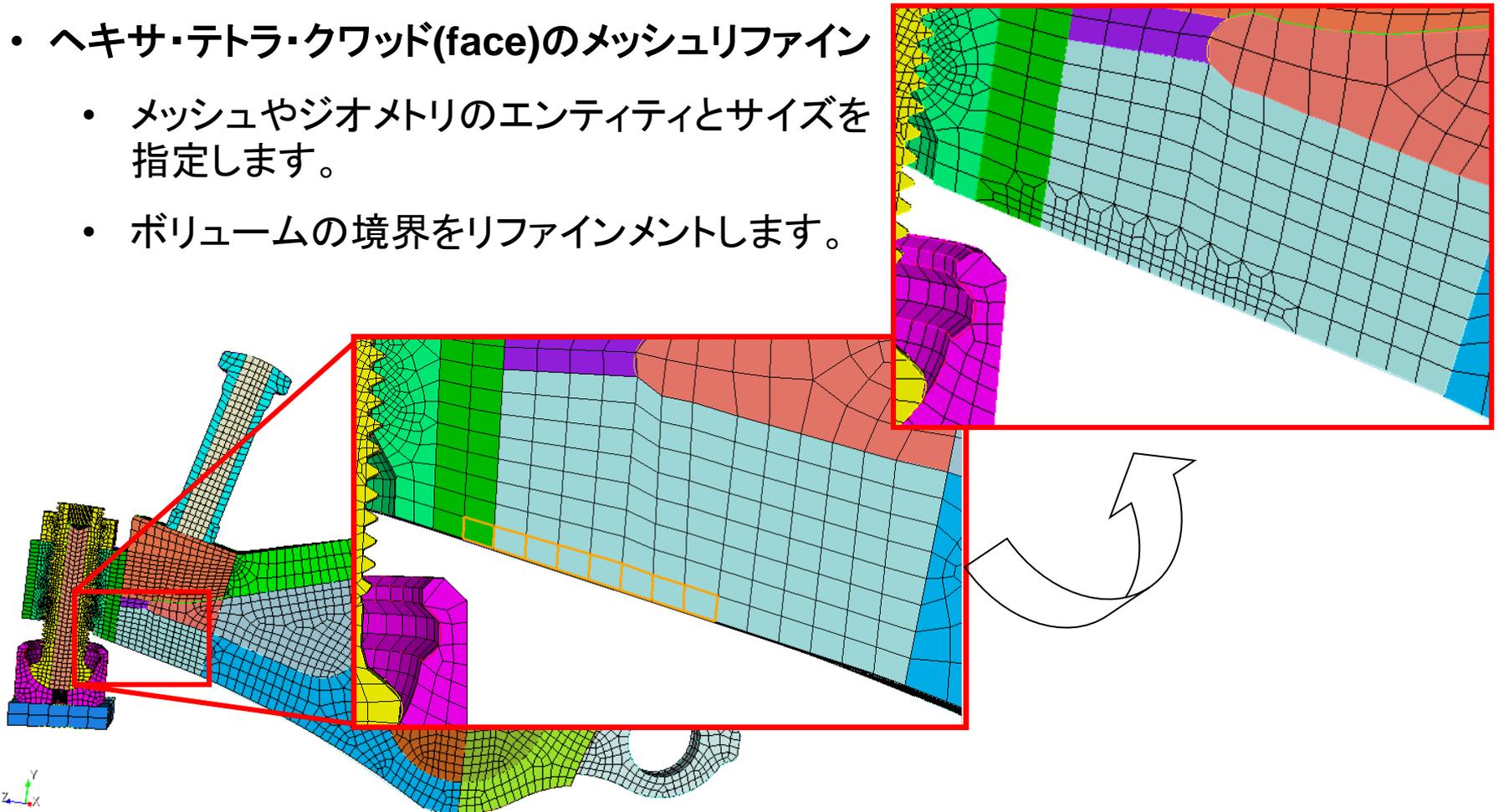


もしpropagate curve bias が適用されなかったらどうなるでしょう？

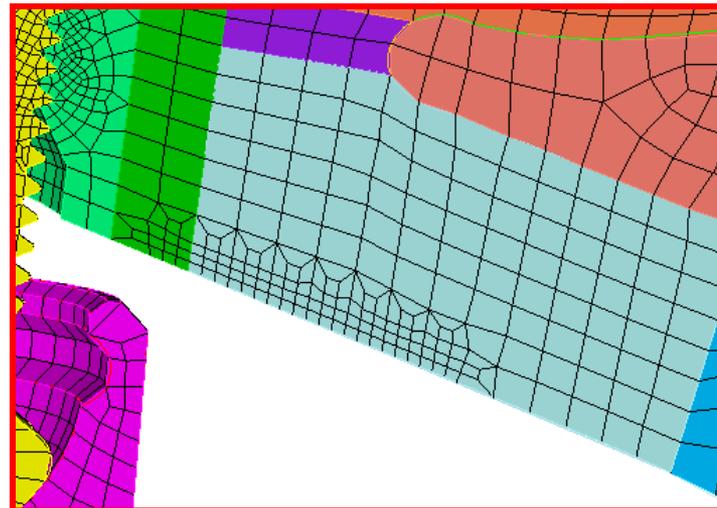
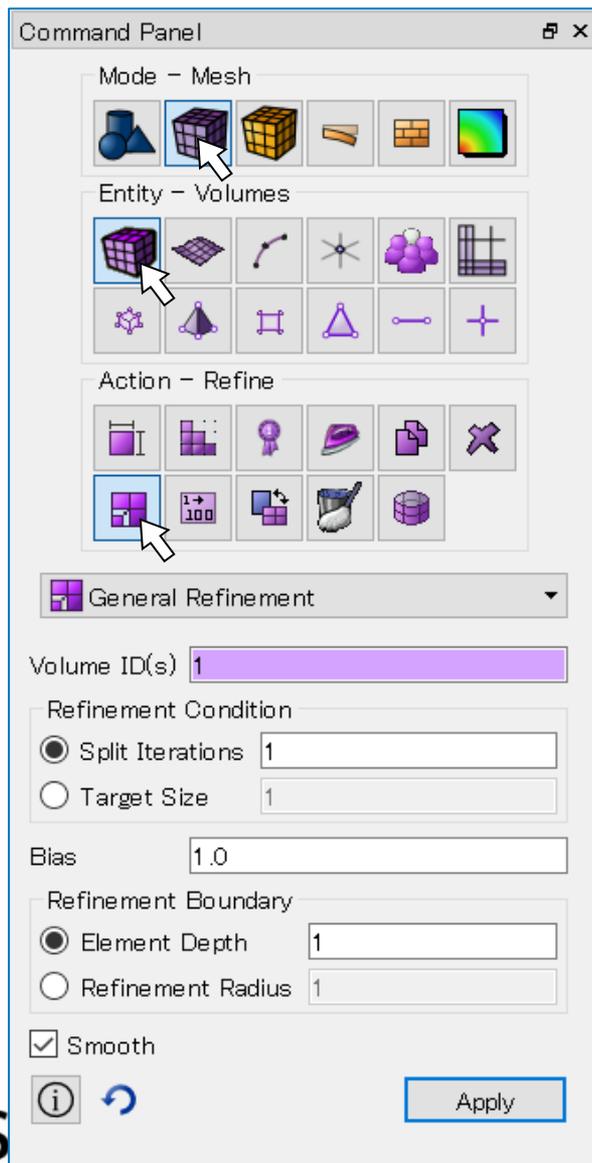


メッシュのリファイン

- ヘキサ・テトラ・クワッド(face)のメッシュリファイン
 - メッシュやジオメトリのエンティティとサイズを指定します。
 - ボリュームの境界をリファインメントします。



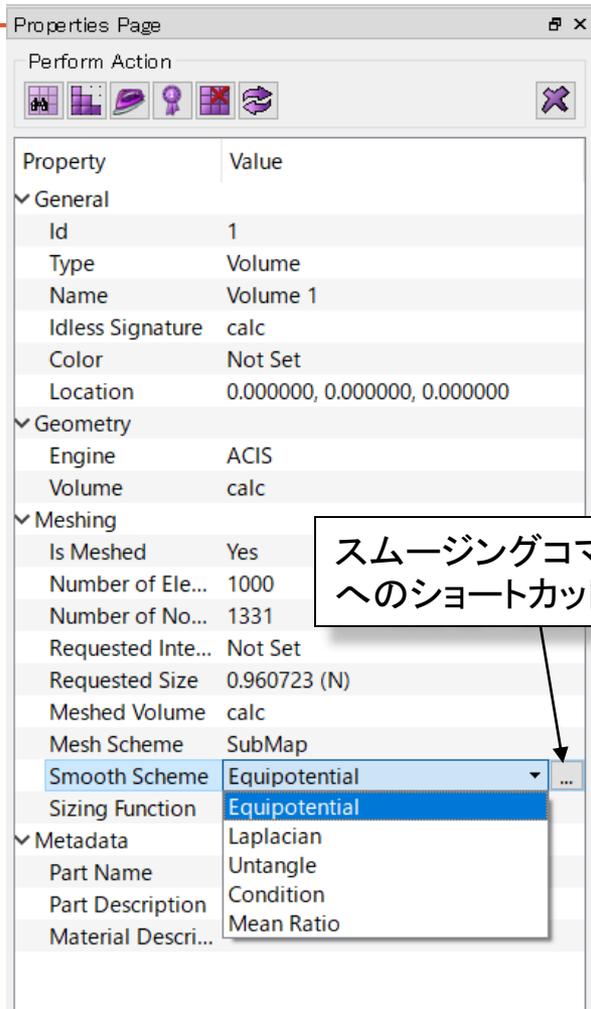
メッシュのリファイン



```
Refine {Node|Edge|Tri|Face|Tet|Hex}
<range>      [NumSplit <int = 1>|Size
<double> [Bias <double>]]      [Depth
<int>|Radius <double>] [Sizing_Function]
[no_smooth]
```

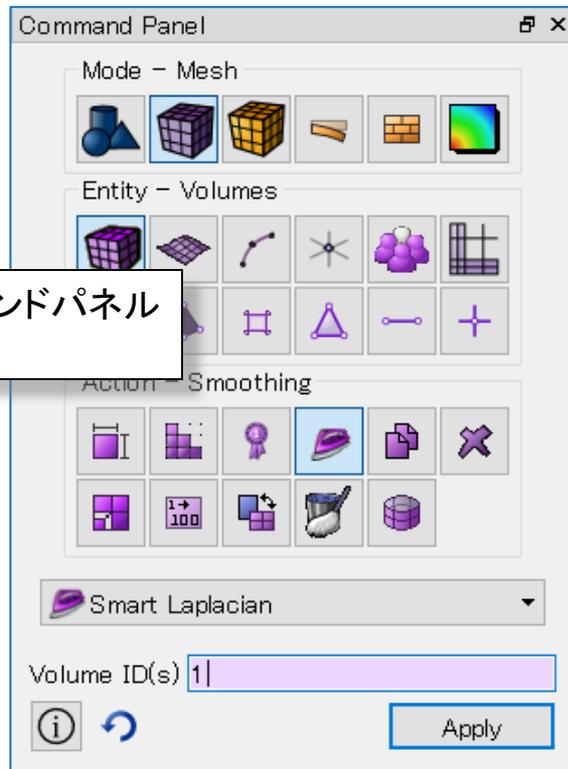
```
Refine {Vertex|Curve|Surface|Volume}
<range>      [NumSplit <int = 1>|Size
<double> [Bias <double>]]      [Depth
<int>|Radius <double>] [Sizing_Function]
[no_smooth]
```

スムージング



スムージングスキームはプロパティパネルから設定します。

スムージングは、要素の接続を変えずに、ノードの位置を調整します。



スムージングスキームは、コマンドパネルからも設定可能です。

数多くの特徴的なスムージングスキームがあります。

通常、反復アルゴリズムによってメッシュの品質を向上させます。

スムージングスキームは面やボリウムにも適用します。

スムージングスキームは、1つの属性として使われます(メッシュスキームに類似)。

サーフェスとボリウムのスキームは独立しています。(free boundary option を選択している場合は除く)

スムージング

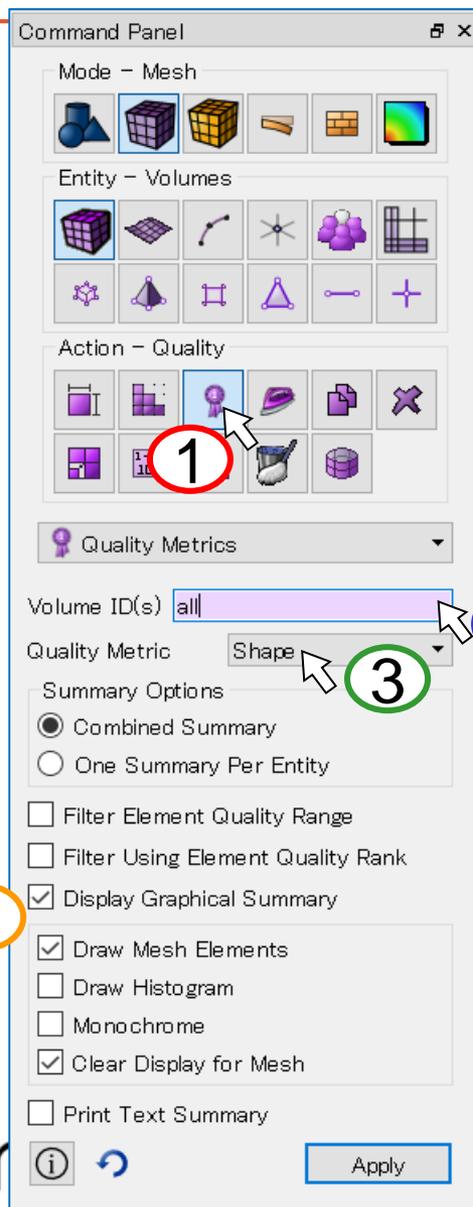
サーフェススムージングスキーム

- **Equipotential**
- **Centroid Area Pull**
- **Optimize Jacobian**
- **Winslow**
 - 長年好まれてきた構造化メッシュへのスキームを、CUBITにおいて非構造化メッシュに拡張
 - 迅速で高品質、最初に試す
- **Laplacian, Centroid Area Pull (Smart Laplacian)**
 - 迅速、くぼんだフィーチャー付近では品質が落ちる
- **Untangle**
 - 反転した要素(負のヤコビアン)の除去
- **Condition number, Mean Ratio**
 - 低品質な要素の改善
 - 要素が反転していないことが必要
- **Edge Length**
 - 各々のエッジの長さを均等にする

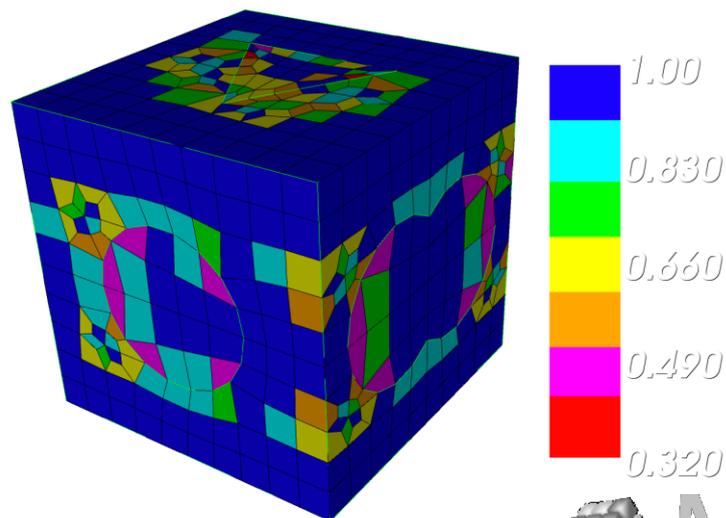
ボリウムスムージングスキーム

- **Laplacian/Smart Laplacian**
 - 迅速、くぼんだフィーチャー付近では品質が落ちる
- **Equipotential**
 - 中くらいの速さと品質
- **Untangle**
 - 反転した要素の除去
 - 時間が掛かるため、時間を制限する:
[cpu <double=10>]
- **Mean Ratio**
 - 要素品質の最適化
 - 時間が掛かるため、時間を制限する:
[cpu <double=10>]
- **Condition number**
 - 同等かそれ以上の品質を保証
 - 低品質な要素の改善
 - 時間が掛かるため、時間を制限する:
[cpu <double=10>]
 - 要素が反転している場合、**Untangle** を先に実行する

メッシュ品質 コマンドパネル



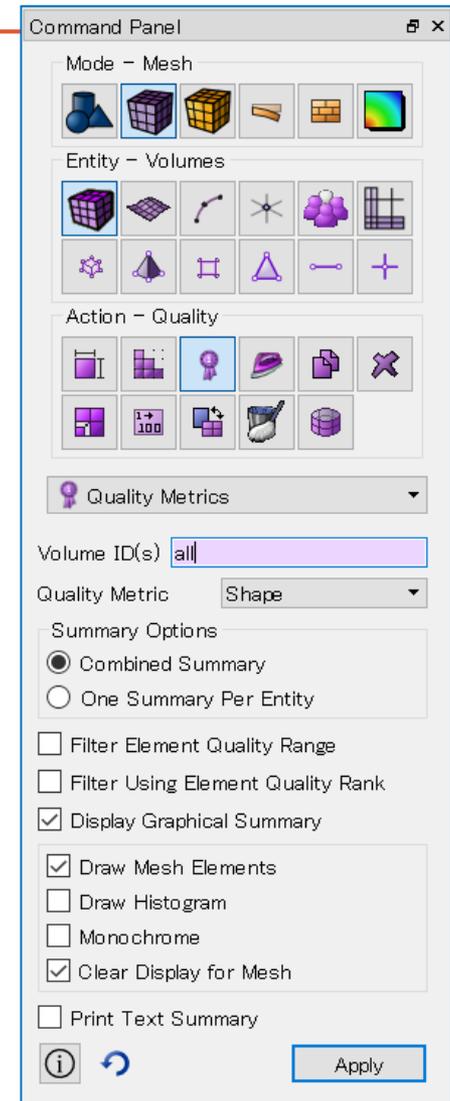
- ① メッシュ品質パネル  を開きます。
- ② 検査するメッシングしたボリュームを選択します。
- ③ 品質メトリクスを選びます。
- ④ メトリクスの表示オプションを選びます。



品質メトリクス

- メトリクスの定義は、オンラインドキュメントに記載されています。
 - qualityページに行って F1 キーを押します。

Function Name	Dimension	Full Range	Acceptable Range	Reference
Aspect Ratio	L^0	1 to inf	1 to 4	[5]
Skew	L^0	0 to 1	0 to 0.5	[5]
Taper	L^0	0 to +inf	0 to 0.4	[5]
Element Volume	L^3	-inf to +inf	None	[5]
Stretch	L^0	0 to 1	0.25 to 1	[6]
Diagonal Ratio	L^0	0 to 1	0.65 to 1	
Dimension	L^1	0 to +inf	None	[5]
Oddy	L^0	0 to +inf	0 to 20	[7][8]
Condition No.	L^0	1 to inf	1 to 8	[8]
Jacobian	L^3	-inf to +inf	None	[8]
Scaled Jacobian	L^0	-1 to +1	0.5 to 1	[8]
Shear	L^0	0 to 1	0.3 to 1	[9]
Shape	L^0	0 to 1	0.3 to 1	[9]
Relative Size	L^0	0 to 1	0.5 to 1	[9]



ポイント

ユーザーにとってより良いメッシュを作成するには、品質確認と調整を繰り返して、最適なメッシュ設定・作成方法を見つけるしかありません。
試行錯誤が重要です。

最適なメッシュはユーザーによって異なりますので、好みのメッシュ分布が得られる方法を研究してください。

最適なメッシュが得られたら、次回からそれをノウハウとして活用願います。
ジャーナルによる制御がとてもパワフルです！
記録も残ります！

基本操作演習(別冊子) p.9

基本操作演習③ コマンドライン

最重要！

マウス操作／エンティティ選択方法は
理解するより体で覚えてください！

基本操作演習(別冊子) p.12

基本操作演習④ カーブバイアス

最重要！

マウス操作／エンティティ選択方法は
理解するより体で覚えてください！

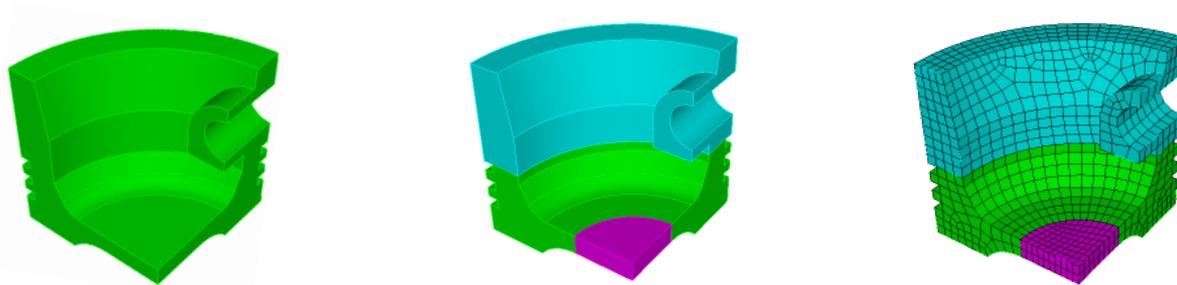
(休憩)

- 以降はヘキサメッシングを行う方向けの補足説明と相談会です

コマンド: *mesh vol all* を使うために

2. 形状の webcut

ヘキサメッシュを作成する場合、webcut コマンドを使ってモデルを分解します。
また、連続なメッシュを作成するために、モデルを分割した後でインプリント・マージを行います。

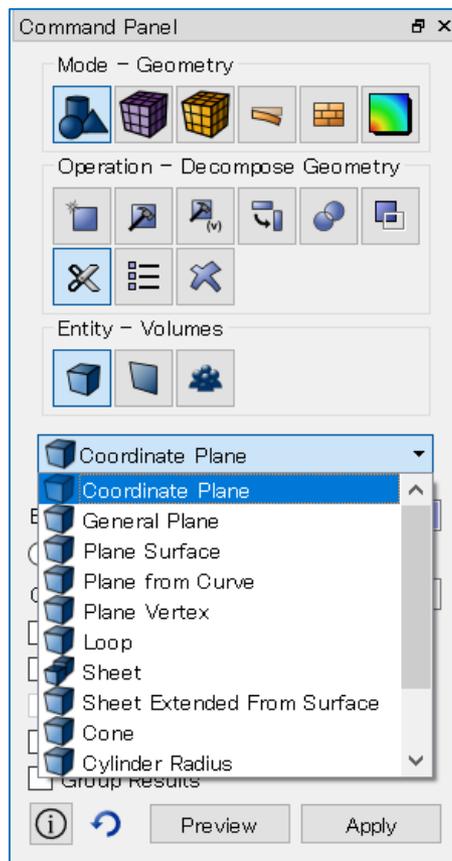


Cubitは、デフォルトのプロパティ情報を参照してメッシュ生成を試みます。

- プロパティ情報が**適切**であれば、**成功**します。
 - 失敗した場合、適切なプロパティ情報を指定する必要があります
- ユーザーが行うことは、
- **適切なプロパティを設定すること**
 - **利用可能なメッシングスキームのためにボリュームを定義／分解すること**
- に尽きます！！

コマンド: *mesh vol all* を使うために

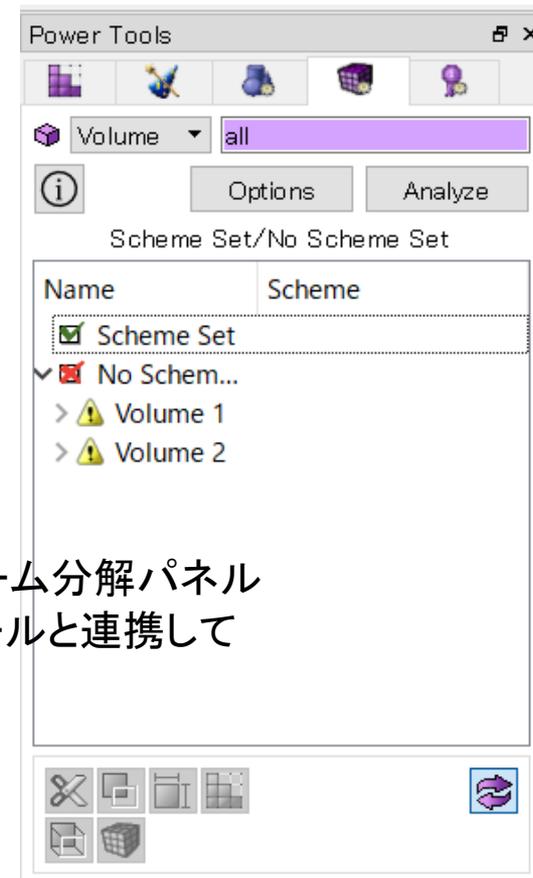
・Web Cut



Webcut コマンドパネル

Web Cut パネルは、様々な分解ツールを備えています。

Web cut は、ITEM ボリューム分解パネルあるいはメッシュパワーツールと連携して使用できます。

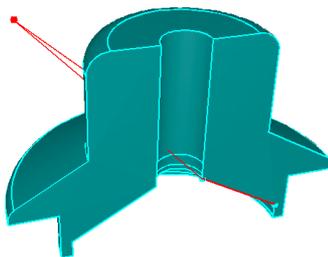


メッシュパワーツール

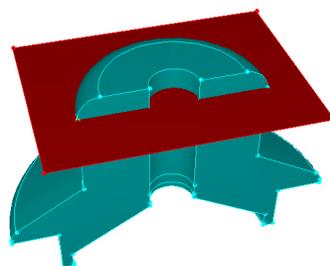
コマンド: *mesh vol all* を使うために

・Web Cut

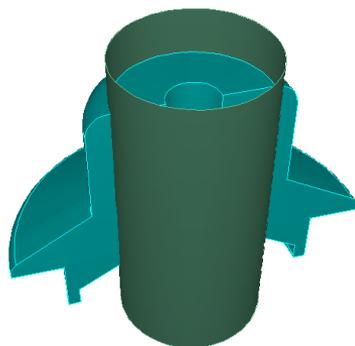
バーテックスで定義
(Plane From Vertices)



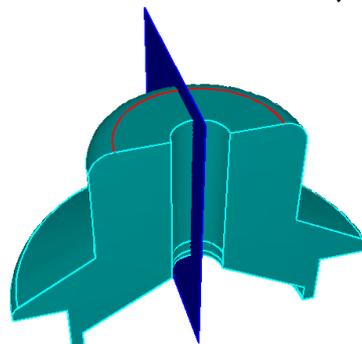
座標平面
(Coordinate Plane)



円筒面
(Cylinder)

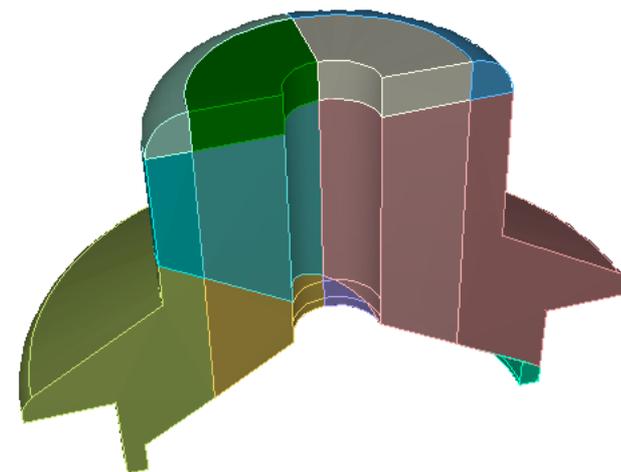


カーブに垂直な平面
(Plane From Curve)



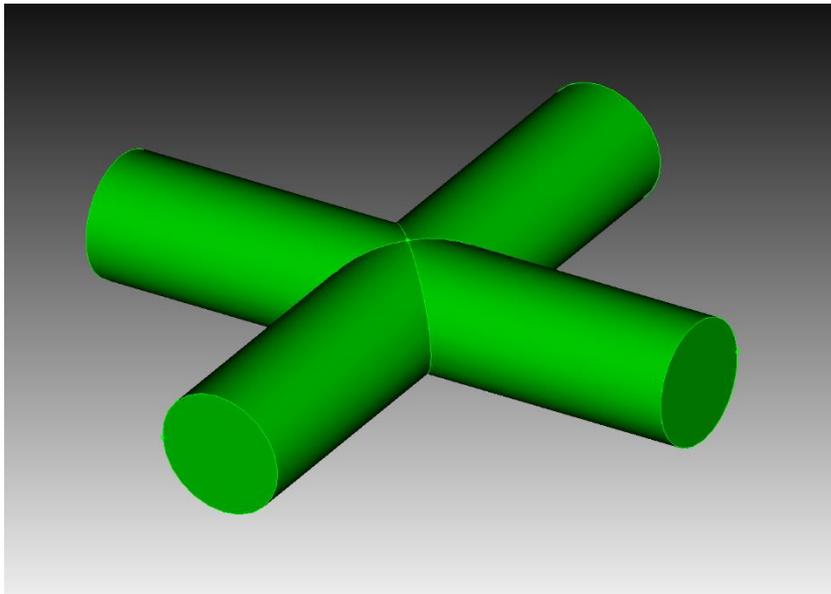
Webcut 例

結果:



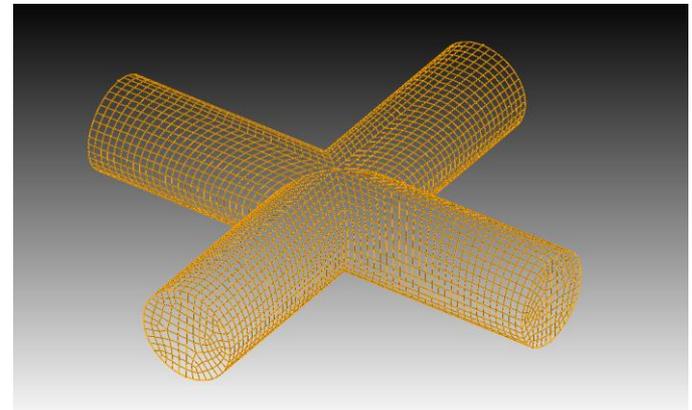
練習 (mesh vol all)

- 以下の形状をmesh vol all でメッシングしてみましょう



```
reset  
create cylinder height 20 radius 2  
volume 1 copy rotate 90 about y  
unite body all
```

どのような分割・プロパティ設定を行いますか？

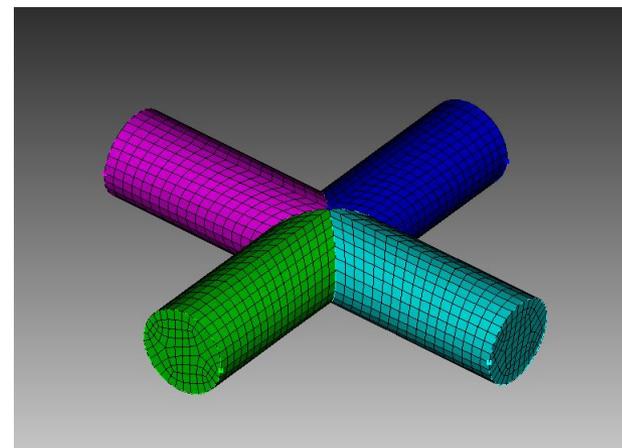


解答例

分割&スイープ設定

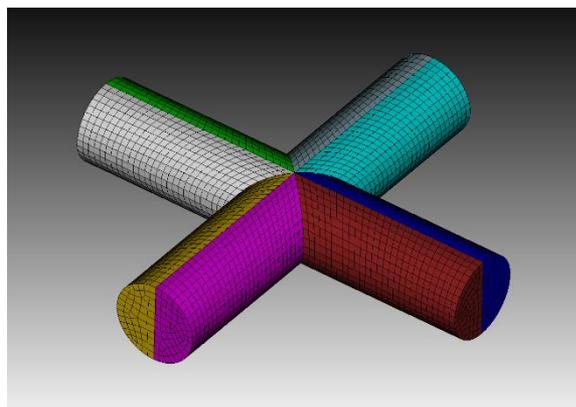
```
webcut volume all with loop curve 5 7
webcut volume all with loop curve 8 6

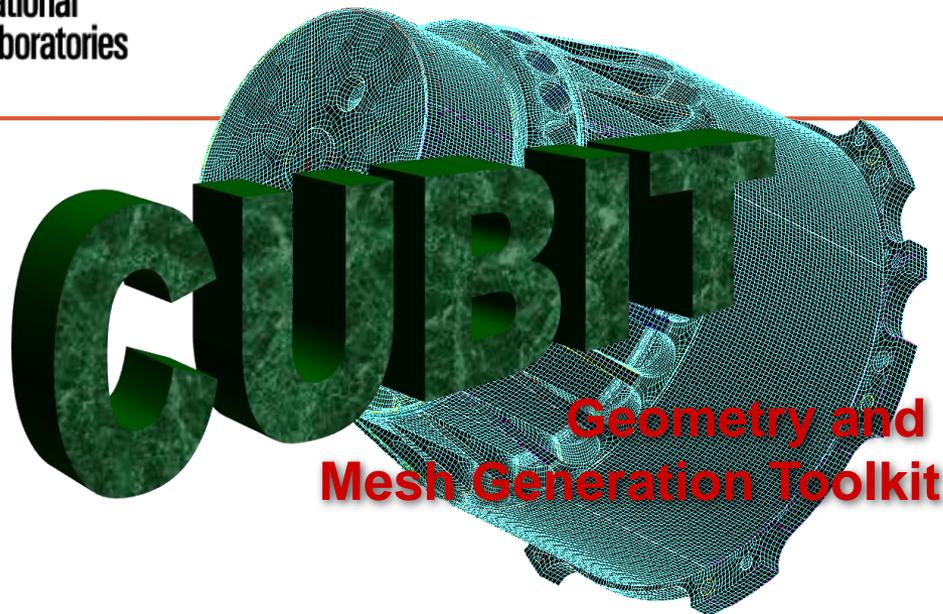
volume 5 scheme Sweep source surface 20 19 target surface 3
volume 4 scheme Sweep source surface 15 16 target surface 6
volume 3 scheme Sweep source surface 18 17 target surface 5
volume 1 scheme Sweep source surface 13 14 target surface 2
mesh vol all
```



分割のみ

```
webcut volume all with loop curve 5 7
webcut volume all with loop curve 8 6
webcut volume all with plane xplane
webcut volume all with plane zplane
mesh vol all
```





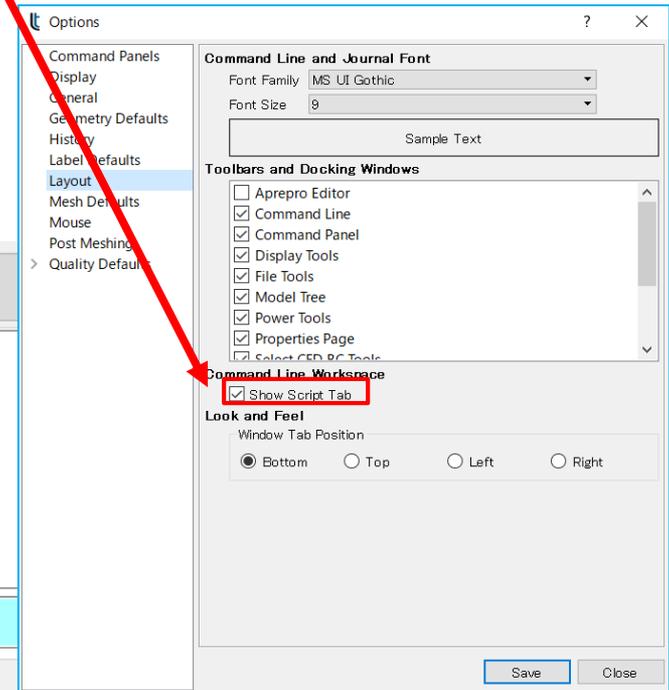
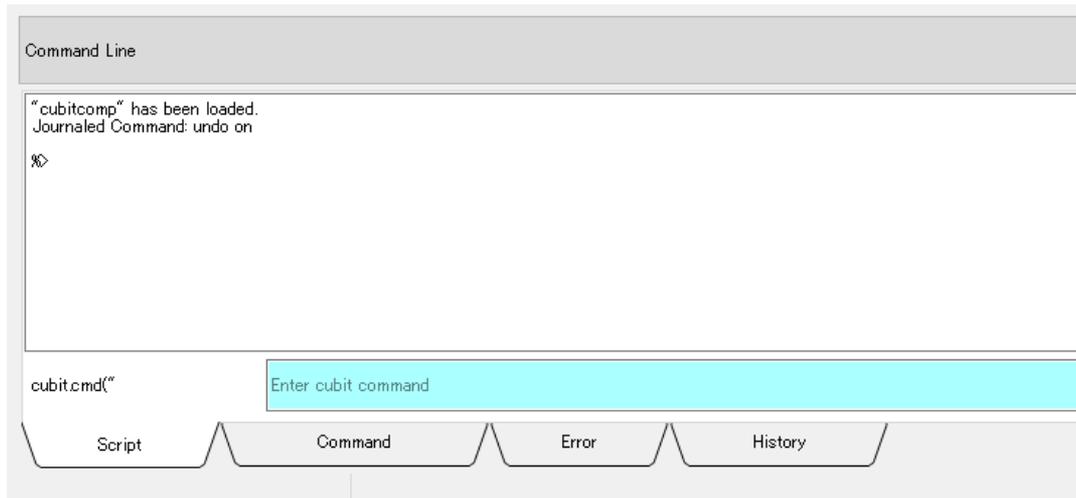
**CUBIT ファーストスタートチュートリアル
Python を使ってみよう！**

Pythonとは

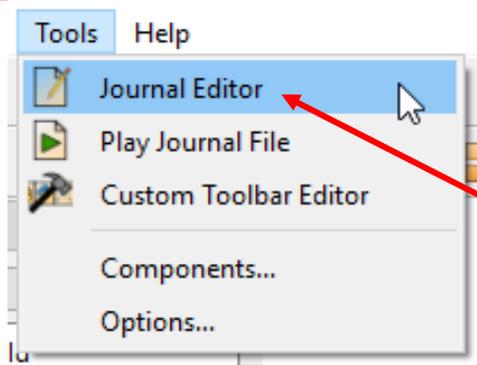
- Python は、実績と人気のある高水準言語です。工学系分野でも広く使われています。
 - Abaqus
 - Paraview
 - PyTrilinos
- Pythonの学習に役立つリンク：
 - 公式サイト: www.python.org
 - Getting started: www.python.org/about/gettingstarted/
 - Tutorial (2.7): docs.python.org/2/tutorial/index.html
 - Reference (2.7): docs.python.org/2/reference/index.html

スクリプトタブの表示

- Tools – Options – Layout – Show Script Tab にチェックを入れる。
- コマンドラインにスクリプトタブが表示され、Pythonコマンドを直接入力できるようになります。



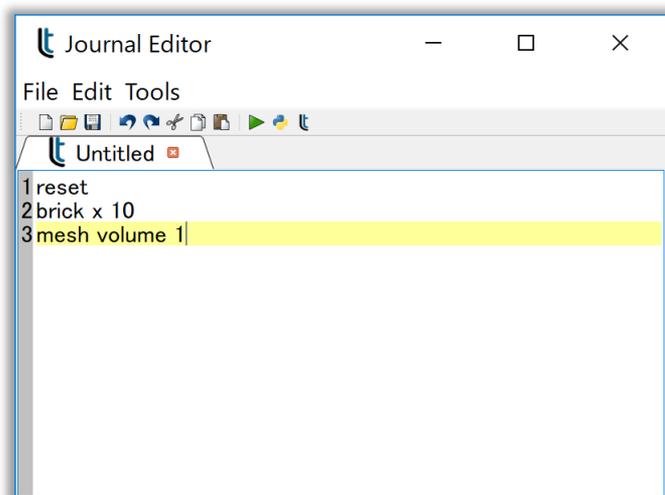
ジャーナルエディタ



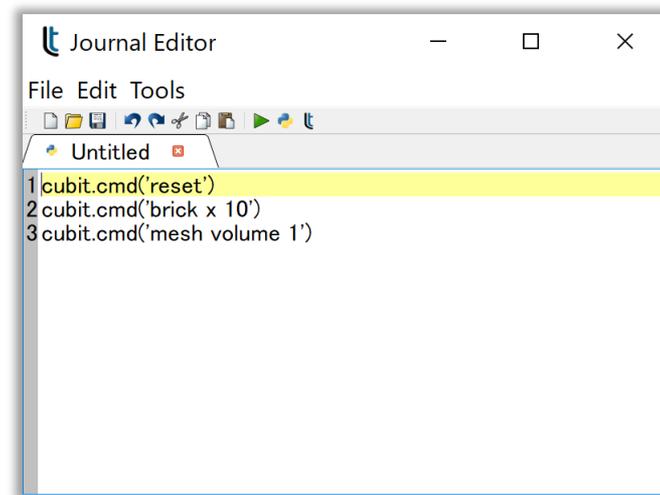
メニューからのアクセス



ツールバーからのアクセス

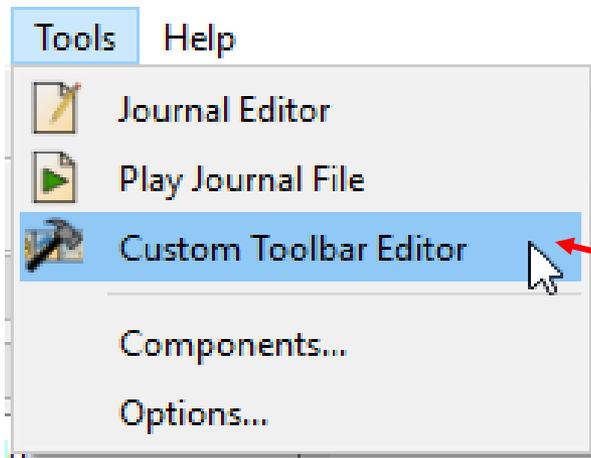


ジャーナルエディタ (Cubitコマンド)



ジャーナルエディタ (Pythonコマンド)

カスタムツールバー



メニューからエディタを開く



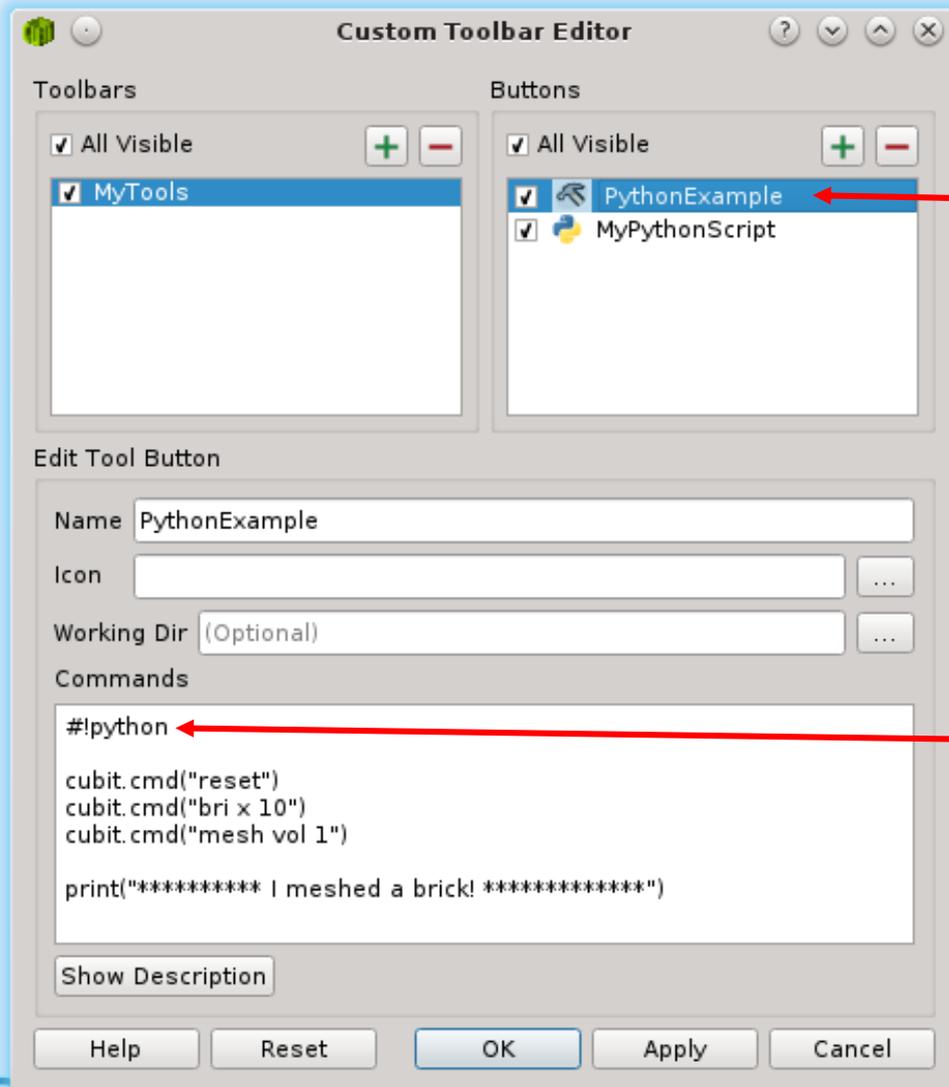
ツールバーからエディタを開く

- 各ボタンをクリックすると、登録したコマンドをまとめて実行できます
- Python スクリプトを登録・実行できます



カスタムツールバーの作成例

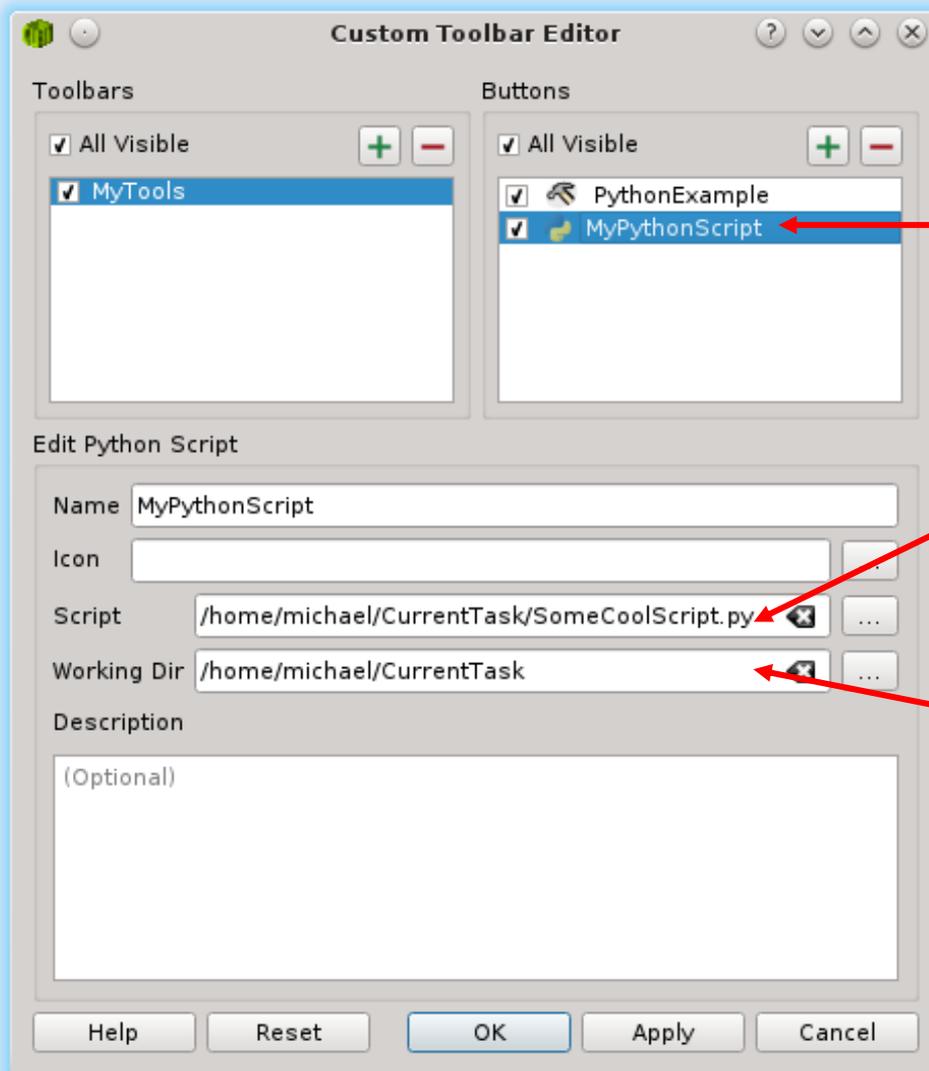
カスタムツールバー



新しいカスタムツールボタンを作成し、コマンド欄にPythonスクリプトを入力してみましょう。

#!python を入れることで、Pythonスクリプトを直接入力することが可能になります。
(necessary for this tool only)

カスタムツールバー



新しいPython スクリプトボタンを作成し、再生するスクリプトを指定しましょう。

再生するPythonスクリプトを指定します

オプションとして、スクリプトのワーキングディレクトリを指定することも出来ます。

Cubit Interface

- 元々はCubit へのクエリインターフェースとして開発されました。

```
- double mesh_size =  
  cubit.get_mesh_size("volume", 22);
```

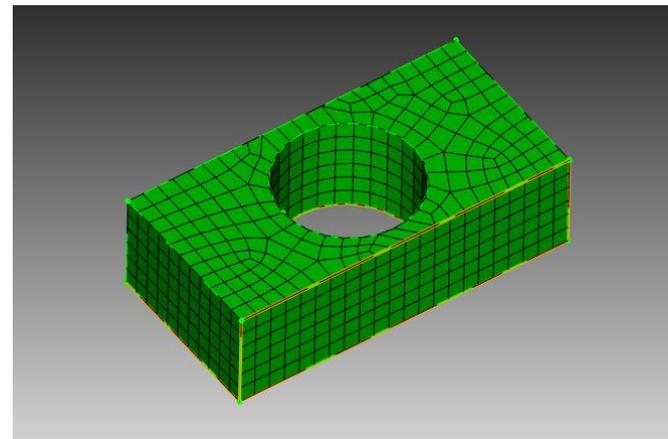
- C++ または Python で使用できます。
- `cubit.cmd(" ... ")` でCubitコマンドを実行します。

```
- import cubit  
- cubit.cmd("create brick x 10 y 10 z 10")  
- cubit.cmd("mesh volume 1,3,5")
```

Cubit Extended Interface

- Cubit でpython オブジェクトを作成
- ID管理のトラブルを減らすことができます

```
bri = cubit.brick(10,5,3)
cyl = cubit.cylinder(12,2,2,2)
vols = cubit.subtract([cyl], [bri])
v = vols[0].volumes()
v[0].mesh()
print dir(v[0])
print v[0].id()
v[0].mesh()
```



Cubitブラックボックス

Python からCubit を実行する方法 (1)

- ・ Cubitはpythonからも実行できます。

- PythonのダウンロードページからPython2.7.12(推奨)をダウンロードします。

(<https://www.python.org/downloads/>)

※Python3系では動作しません

- 環境変数を設定するバッチファイル(pycubit.bat)を作成、保存します。

```
set Path=C:¥Python27
```

```
set PYTHONPATH=C:¥Program Files¥Trelis 16.0¥bin
```

環境変数は、お使いの環境に合わせて調整してください

(バージョン16.1では、

```
set PYTHONPATH=C:¥Program Files¥Trelis 16.1¥bin
```

```
set Path=C:¥Python27;C:¥Program Files¥Trelis 16.1¥bin¥acis¥code¥bin
```

にする必要があります。)

Path: Pythonがインストールされてるディレクトリーへのパスを設定します。

(Trelis16.1では、Acisのコードがインストールされているパスも設定してください。)

PYTHONPATH: TrelisのPython用ライブラリーがインストールされているパスを設定します。

Cubitブラックボックス

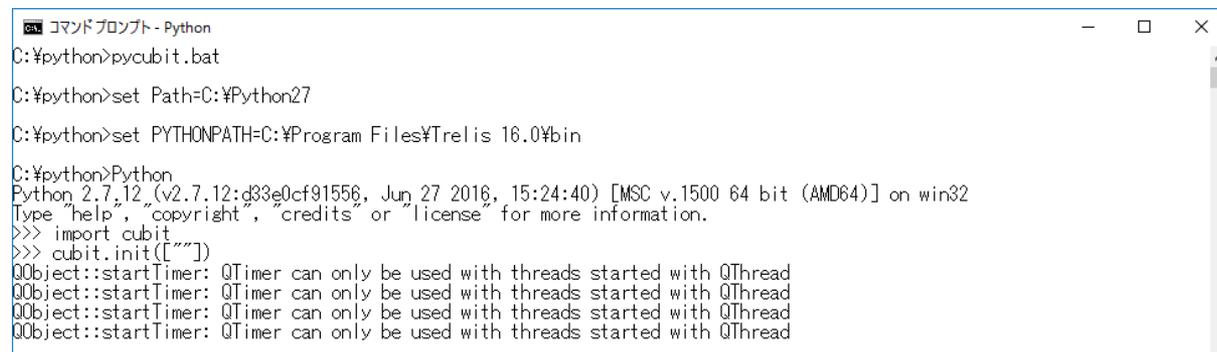
Python からCubit を実行する方法 (2)

- ・ コマンドプロンプトを起動し、バッチファイルを保存したディレクトリまで移動します。

- ・ 以下のコマンドを順に実行します。

```
pycubit.bat          #環境変数の実行
python               #Pythonの実行
import cubit         #PythonによるCubitのインポート
cubit.init([""])     #PythonによるCubitの初期化(必須)
cubit.cmd('brick x 10') #PythonによるCubitコマンドの実行
cubit.cmd('mesh volume 1')
cubit.cmd('export mesh "block.e" overwrite')
```

- ・ Cubitをプログラムで実行し、他のツールとインタラクトさせることができます。



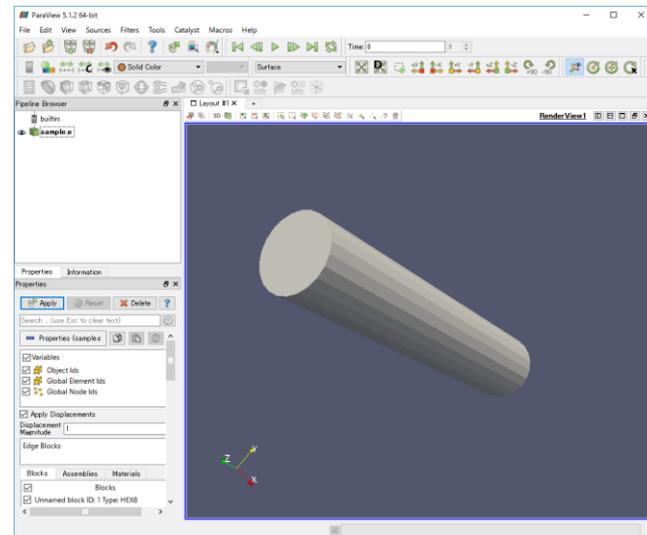
```
コマンドプロンプト - Python
C:\python>pycubit.bat
C:\python>set Path=C:\Python27
C:\python>set PYTHONPATH=C:\Program Files\Trelis 16.0\bin
C:\python>Python
Python 2.7.12 (v2.7.12:d33e0cf91556, Jun 27 2016, 15:24:40) [MSC v.1500 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> import cubit
>>> cubit.init([""])
QObject::startTimer: QTimer can only be used with threads started with QThread
QObject::startTimer: QTimer can only be used with threads started with QThread
QObject::startTimer: QTimer can only be used with threads started with QThread
QObject::startTimer: QTimer can only be used with threads started with QThread
```

Cubitブラックボックス サンプル

- 例えば以下のようなコマンドを実行すると、高さ10半径1の円柱が作成・メッシングされ、e形式・cub形式で保存されます。

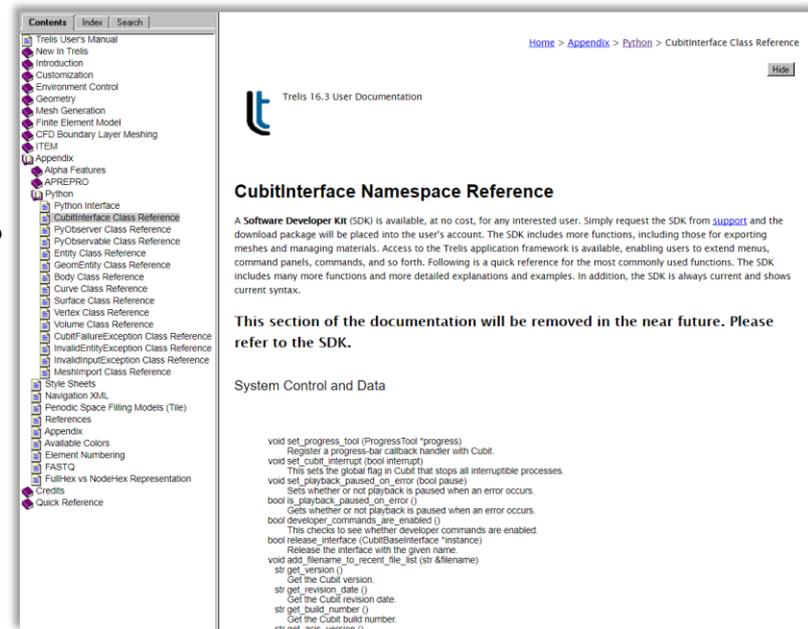
```
cubit.cmd('create Cylinder height 10 radius 1 ')  
cubit.cmd('mesh vol all')  
cubit.cmd('export mesh "sample.e" overwrite ')  
cubit.cmd('save as "sample.cub" overwrite journal')
```

- ParaViewなどのソフトでファイルが正常に保存されていることが確認できます。
- Cubitでcub形式のファイルを開く際には、graphics resetのコマンドを入力する必要があります。



PythonAPIの参照

- Help – Trelis Help からヘルプを参照することができます。
- Appendix – Python からCubit上の様々なデータを取得するためのAPIマニュアルを参照することができます。



The screenshot shows the Trelis 16.3 User Documentation website. The left sidebar contains a navigation menu with categories like 'Trelis Users Manual', 'Introduction', 'Customization', 'Environment Control', 'Geometry', 'Mesh Generation', 'Finite Element Model', 'CFD Boundary Layer Meshing', 'ITEM', 'Appendix', 'Alpha Features', 'APREPRO', 'Python', and 'Python Interface'. Under 'Python Interface', 'CubitInterface Class Reference' is selected. The main content area displays the 'CubitInterface Namespace Reference' section, which includes a notice: 'This section of the documentation will be removed in the near future. Please refer to the SDK.' Below this, there is a 'System Control and Data' section with a list of Python API functions.

```

void set_progress_tool (ProgressTool *progress)
  Register a progress-bar callback handler with Cubit.
void set_cubit_interrupt (bool interrupt)
  This sets the global flag in Cubit that stops all interruptible processes.
void set_playback_paused_on_error (bool pause)
  Sets whether or not playback is paused when an error occurs.
bool is_playback_paused_on_error ()
  Gets whether or not playback is paused when an error occurs.
bool developer_commands_are_enabled ()
  This checks to see whether developer commands are enabled.
bool release_interface (CubitInterface *instance)
  Release the interface with the given name.
void add_filename_to_recent_file_list (str &filename)
str get_version ()
  Get the Cubit version.
str get_revision_date ()
  Get the Cubit revision date.
str get_build_number ()
  Get the Cubit build number.
str get_actx_version ()
  
```

カスタマイズ

- **Cubit** では、**GUIをカスタマイズすることができます**
 - メニューアイテムの追加
 - ダイアログの追加
 - コントロールパネルの追加 (現在は不可)

- **PyQt5**

```
from PyQt5 import QtGui
QtGui.QMessageBox.question(None, "Title", "Hello")
```

PyQt4 との互換性はありません。既にPyQt4 を使っている場合は、PyQt5 にアップグレードする必要があります。