2021/09/08

spider2とspider3の比較

大野泰治郎

1　基本的な流れ

1.1　spider2 を用いた蜘蛛の巣メッシュの作成手順

CAD データ作成ー＞ CAD データをポリゴンメッシュ化ー＞そのポリゴンメッシュを覆うメッシュデータ (土台、ここでは CAD データを覆うコア部分と蜘蛛の巣を張るために用いる拡張部分を合わせた領域) を blockMesh により作成ー＞必要があれば、snappyHexMesh を行いメッシュデータのコア部分から CAD データをくり抜くー＞くり抜いた土台に spider2 をかけて、拡張部分を蜘蛛の巣化したメッシュを作成する。[*1]

1.2　spider3 を用いた蜘蛛の巣メッシュの作成手順

コア部分を想定して、spider3.paramters ファイルを作成する。snappyHexMesh によってのみメッシュを作成する場合には、空の blockMeshDict を用意する。spider3 を用いてコア部分および蜘蛛の巣を追加する blockMeshDict を作成する。blockMesh を行い、コア部分をもつ蜘蛛の巣形状のメッシュデータを作成する。そのメッシュデータを土台にして、必要であれば、コア部分に snappyHexMesh を行い CAD データをくり抜いた蜘蛛の巣メッシュを作成する。[*2]
本来、空の blockMeshDict ではなく、拡張したい blockMeshDict に対して、拡大処理を行えるプログラムであるが、まだ試験が不十分なので、空の blockMeshDict に今の所限定している。

[*1] 私自身は openfoam を用いたシミュレーションをしたことがないので、この作成手順は推定です。間違えている場合はご容赦ください。 spider3 も同様

[*2] snappyHexMesh によりコアの領域外のメッシュに影響を与えないことが前提である。逆にいうと、影響を与えないようにコア領域を作らなければいけない。

2 spider2 と spider3 の比較表

	spider2.py	*spider*2.1[0.1]	spider3.03
開発言語	python2	java 11	java 11
入力ファイル	points	points	拡大対象となる *bockMeshDict*、ない場合には *blockMeshDictEmpty*[1.1]
出力ファイル	spider_points	spider_points.2.1	blockMeshDict.out.3.03
必要なファイル[1]	parameters	parameters	parameters.spider3
アルゴリズム[2]	非線形点拡大 α (p)(p-p0)+p0	非線形拡大	線形点拡大 A(p-p0)+p0
蜘蛛の巣を張るメッシュ数[3]	$7 \times l \times m \times n$	$7 \times l \times m \times n$	$2 \times (l \times m + m \times n + n \times l)$[2.1] × *divED*
対象図形[4]	直方体（制約あり）[4.1]	直方体（制約あり）[4.2]	直方体（制約なし）
拡大の原点	中心または境界面の中心[5.1]	コア領域内部[5.2]	コア領域内部
拡大後の図形[6]	非相似形（制約あり）[6.1]	非相似形（制約あり）	相似形
拡張メッシュ並びの調整	できない。	できない。	*multigrading*[6.2] を使用することで可能

0.1 spider2.py の java 版

1 spider2.py のパラメータファイル `http://3.83.39.176/kyutech/2021.07.07/parameters.pdf` と spider2.01 のパラメータファイルの形式は同じだが、spider2.py では p0（拡大の中心）cuboid（正規化パラメータ) gamma1 gamma2 beta1 beta2 しかみていないのに比べて、spider2.01 では minX minY minZ maxX maxY maxZ も見ている。divX,divY,divZ は入力チェックはするが内部で使用していない。spider3.03 のパラメータ `http://3.83.39.176/kyutech/2021.07.07/parameters.spider3.pdf` はそれとは異なり、p0 beta2 coreMinX coreMinY coreMimZ coreMaxX coreMaxY coreMaxZ divX divY divZ divED multigrading を見ている。

1.1 現バージョンは、blockMeshDict の入力はしていないが、本来は拡張したい blockMeshDict を入力ファイルとして設定するようにすべきである。空の blockMeshDict ファイル.. 本来はプログラム内に埋め込むべきもの。

2 p は点 (point) の座標　p0 は拡大の原点、α $(p) =$ β $(|(p - p0)/cuboid|_{max})$ はスカラー値であるが、p 依存である。即ち、全体では線形ではない。ここで、$(p - p0)/cuboid$ は x, y, z 成分ごとの割り算、$||_{max}$は各座標値の絶対値の最大値つまり、$|v|_{max} = max\{|v_x|, |v_y|, |v_z|\}$ を表す。点拡大の意味は p0 と p の直線上に拡大された点がくることを意味する。線形拡大 A とは単なる行列による拡大である。

2.1 拡大の中心がコア内部にある場合。境界上にある場合は、それよりも減る。

3 (l,m,n) はコア領域の (x,y,z)-方向メッシュ数、spider2 では伸びしろの点数をコアの 2 倍にとって計算、

spider3 の divED とは伸びしろ方向の分割数 (parameters.spider3 で与えられる入力パラメータ)

4.1 パラメータ cuboid の与え方は、2 通りある。1 つ目は、対象図形が立法体でない場合でも、cuboid:x,y,z の値 x,y,z を等しくすること。このとき、x_len（拡大の原点 (p0) からの x_長）として、x=y=z=min(x_len.y_len,z_len)
と与えると、拡張後の対角線でのメッシュでの整合性があう。ただし、直方体の最小方向以外では、拡大幅が一定 (β) のメッシュが残りの部分を埋めることになる。http://3.83.39.176/kyutech/2021.07.07/png/70.60.80.spider2.pdf subsection1.1 subsection1.2 これは python 版も java 版も同じ結果である。
x=y=z=max(x_len,y_len,z_len) にとると、拡大幅一定のメッシュがくることはないが、全体が直方体にはならずに、シルクハット図形 (極端な例だけれど、http://3.83.39.176/kyutech/2021.07.07/png/70.60.80.spider2.pdf の subsection 1.4) となる。これを修正したのが、spider2.1 java 版である。(subsection 1.3)
2 つ目の方法は、x-len,y-len,z-len の比率が、例えば 4:1:1 など整数倍になった場合である。（2021 年の４年のみなさんの課題 ただし、メッシュは大幅に荒いはず）http://3.83.39.176/kyutech/2021.07.07/png/140.70.70.spider2.pdf

4.2 python 版 spider2 に x=y=z=max(x_len,y_len,z_len) にとった場合の修正版である。(作成したのは 2021.07.07 頃) http://3.83.39.176/kyutech/2021.07.07/png/70.60.80.spider2.pdfsubsection 1.3
この例では、あまり効果がないように思うかもしれないが、おそらく役に立つ場合がある。

6.1 これは、試験が不十分なので、やめておく。http://3.83.39.176/kyutech/2021.07.07/png/70.70.70.xCoreRatio0.5.png