2021/04/29

スパイダーアルゴリズムの検証

大野泰治郎

目次

1 目的

IS さんの作成されたスパイダーアルゴリズム（蜘蛛の巣メッシュ作成）の検証と、使用するときの注意点をまとめること。

2 スパイダーアルゴリズム

３次元での直方体（拡張対象領域）からそれを拡張した（内部に含むという意味）直方体（拡張後領域）への変換アルゴリズムである。このとき、拡張しない領域（不拡張領域）を指定できること、メッシュの相対的な構造（つまり、隣接メッシュの関係および点の数）を保つことが要求され、できるだけメッシュとしての形状を壊さないことが望まれる。

次の定数を使用する。() 内は対応する python プログラムの変数

$$
\begin{cases}
p_0(p_0) \quad \text{拡張の原点 (拡張対象領域の境界を含めてその内部にとる) ３次元} \\
C_u(cuboid\_len) \text{ 正規化係数（各成分は正値）　３次元} \\
\text{以下は、拡張係数を計算するための定数（スカラー値）} \\
\beta_1(\text{これは実数値 } 1.) \\
\beta_2(beta) > \beta_1 \\
\gamma_1(gamma1) \geq 0. \\
\gamma_2(gamma2) > \gamma_1 \text{および} \quad \gamma_2 \leq 1.
\end{cases}
$$

拡張対象領域内の点 p に対して拡張係数$\alpha(p)$ を与える計算は次である。

$$\alpha(p) = f(|(p - p0)/C_u|_{max}) \tag{1}$$

ここで、$|v|_{max}$は３次元の点 v の各成分の絶対値の最大値すなわち $max(|v_x|, |v_y|, |v_z|)$ のことである。
f(x)=

$$\begin{cases} \beta_1 = 1. & on(0. \leq x < \gamma_1) \\ \frac{\beta_2 - \beta_1}{\gamma_2 - \gamma_1}(x - \gamma_1) + \beta 1 & on(\gamma_1 \leq x \leq \gamma_2) \\ \beta_2 & on(\gamma_2 < x \leq 1.) \end{cases}$$

このときに、スパイダーアルゴリズムとは次の変換のことである。$\hat{p} = alpha(p)(\frac{p - p_0}{C_u}) * C_u + p_0 = alpha(p)(p - p_0) + p_0$

3　拡張されない領域および拡張後の領域

アルゴリズムに従って、不拡張（拡張されない）領域は次のように計算できる。

$$|\frac{p - p_0}{C_u}|_{max} \leq \gamma_1 \text{従って、} |p - p_0|_{max} \leq \gamma_1 * C_u \tag{2}$$

アルゴリズムに従って、拡張後の領域は次のように評価できる。

$$\hat{p} \leq \beta_2(p - p_0) + p_0 \tag{3}$$

4　変換後メッシュを構成するセルはいかに変形するか？

これをメインにするつもりでしたが、別のことをしはじめてやれてませんでした。
申し訳ないです。。。つまり、検証にはなってませんです。

5　このスパイダーアルゴリズムを使用にあたって注意する点

- 注意１　拡張は、拡張の原点 p_0 と拡張する前の点 p を結んだ線分の p 側の線分を延長した上にある。
- 注意２　上の延長の倍数が最大 β_2 で与えられる。
- 注意３　このことは、拡張したセルの長さが最大 β_2 で与えられるということではない。セルの長さはそれよりもむしろ途中段階でさらに大きく拡張する。（なぜならば、前のセルを構成する点は β_2 以下で拡張されるからだ。）
- 注意４　アルゴリズムからわかる重要なことは正規化された図形では、拡張前の拡張の原点を中心とした立方体は立方体に移るということである。（なぜならば $||_{max}$ の値がその立方体の境界上で変わらないからである。）おそらくそのことが、このアルゴリズムの正当性を与えている。
- 注意５　上は直方体では成立しない。つまり、一般には直方体は変換によって歪む。
- 注意６　ここで、正規化後の形状が直方体とか立方体とか指摘しているのは、C_u だけでなく、変換前のメッシュを構成するセルの形状（元セルと言うことにする）にも依存することに注意すべきである。
- 注意７　$||_{max}$を用いる代わりに例えば、$|v|_{abc} = max|v_x| * a, |v_y| * b, |v_z| * c$ を用いると、この場合は直方体 (1/a,1/b,1/c) が、変換で不変であるが、これは、もとの図形から考えると C_u を変更するのと同じことである。

- 注意８　C_u のとり方は、もともと拡張の原点から、元の図形での x,y,z 方向の（セルの個数）長さを設定しているようである。（名前のつけかたから想定して）
- 注意９　上の注意８は、元セルが立方体であることを前提とすると、注意８で設定された値が（x，y，z）方向で l、m，n とすると、各セルは、1/l,1/m,1/n に縮小されることになる。つまり、l=m=n でない限り、正規化後は、セルは立方体でなくなる。つまり、拡大後、同一平面上に並ぶ保証はなくなる。（この部分ずいぶんはしょって説明しております。）
- 注意１０　つまり。もっとも望ましいのは、拡大原点を中心にして、拡大対象領域および不拡大領域および、元セルが立方体である場合である。

6　適用

対象モデルを次の図のようなものとする。　ーーーー図はまだ描いていない。元セルは立方体であると仮定する。

- ステップ１　対象領域を切り出す。
- ステップ２　拡張原点をきめる。
- ステップ３ 拡張対象領域と非拡張対象領域を決める。このとき、拡張原点を中心として、拡張対象領域と非拡張対象領域が立方体であることがベスト、そうでない場合でも、縦、横、高さが１対２対２のような（形状（横、縦、高）（つまり、２個あつめると立方体になる）ような形状がのぞましい。
- ステップ４　正規係数　C_u を決める。このとき。C_u で割った値が拡張対象領域を長さ１の立方体にするように決める。
- ステップ５　拡張後領域と非拡張領域と拡張対象領域からパラメータ $\beta_2\gamma_1, \gamma_2$ を決める。

7　(

例)

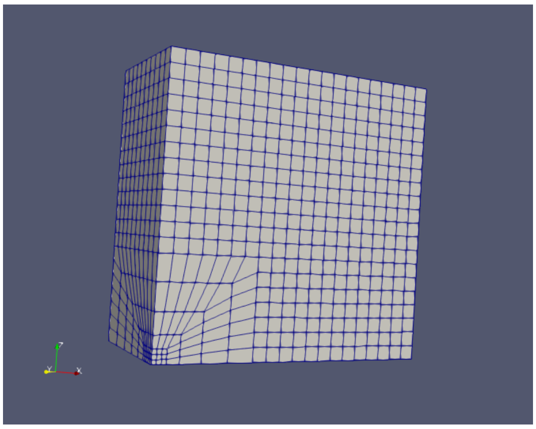

図１　l/3 m n/3

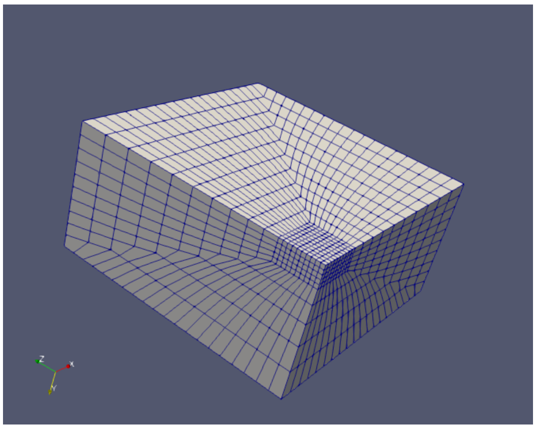

図 2　l m n

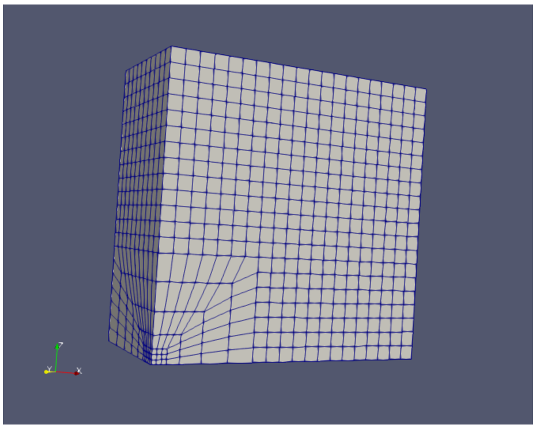

図 3

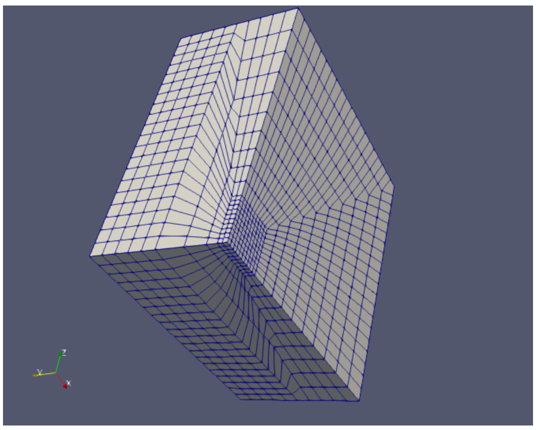

図 4　l m/2 n

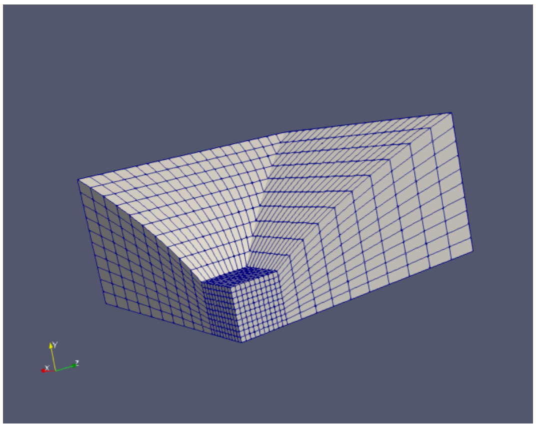

図 5　l 2*m n

8　(

おまけ)

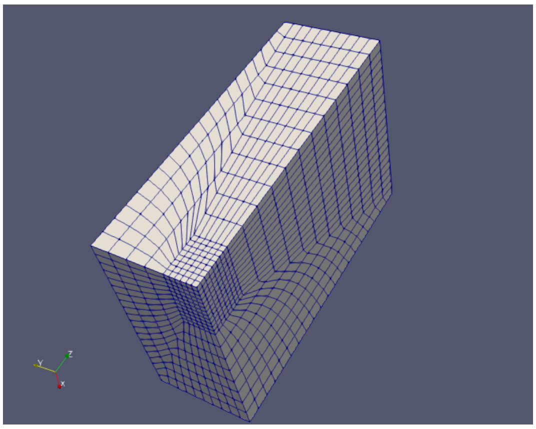

図 6　omake