

流体解析における ChatGPT を用いた条件設定について

日大生産工(院) ○吉沢 昇馬 日大生産工 沖田 浩平

1. 緒言

近年、OpenAI社が開発したChatGPT-4に代表される言語生成AIは、文章の要約等だけでなく、プログラム言語を扱うことができるため、指定した条件に対してプログラミングコードを生成することも可能になってきている。また、生成AIを利用した三次元形状の生成も研究されるなど、言葉だけで適切な形状を作成し、その評価を行うAIの登場が期待されている。

一方、流体解析においては、計算格子の生成や境界条件等の各種設定は非常に煩雑なものとなっている。特に、オープンソースの流体解析開発環境であるOpenFOAMの利用に際しては、Fig.1に示すようなテキスト形式で各種設定を行うため、初学者が解析を行うにあたってのハードルが高いものとなっている。また、複雑化した流れ解析の条件をテキストで設定する際に時間を要する。

そこで、本研究では、流れ解析におけるテキストベースの条件設定ファイルの生成に、言語生成AIであるChatGPT-4を用いて解析を行う手法について検証した。

2. 流体解析用設定ファイルの作成

流体解析開発環境であるOpenFOAMにおける設定ファイルの作成について述べる。

2.1. 流体解析の設定項目

流れ解析における設定項目は、通常、以下のようものが挙げられる。

- ・計算領域と流路等の物体形状の設定
- ・支配方程式（物理モデル）とソルバーの選定
- ・境界条件
- ・初期条件

OpenFOAMでは、これらの設定のために複数のテキストファイルを準備する必要がある。

2.2. 格子生成

計算領域の空間的な離散化である格子生成には、ベースとなる格子生成にblockMeshを、形状データの境界適合の非構造格子生成にsnappyHexMeshを用いた。これらの格子生成の設定にはそれぞれテキストベースのDictフ

```

1 /*-----*/
2 // Field | OpenFOAM: The Open Source CFD Toolbox
3 // Operation | Website: https://openfoam.org
4 // And | Version: 11
5 // Manipulation |
6 //-----*/
7 FoamFile
8 {
9     format      ascii;
10    class       dictionary;
11    object      blockMeshDict;
12 }
13 //-----*/
14 //
15
16 convertToMeters 1;
17
18 vertices #codeStream
19 {
20     codeInclude
21     #include "pointField.H"
22 }
23 #}
24
25 code
26 #f
27 pointField points(19);
28 points[0] = point(0, 0, -0.5);
29 points[1] = point(1, 0, -0.5);
30 points[2] = point(2, 0, -0.5);
31 points[3] = point(0, 0.707107, -0.5);
32 points[4] = point(0.707107, 0.707107, -0.5);
33 points[5] = point(0.353553, 0.353553, -0.5);
34 points[6] = point(2, 2, -0.5);
35 points[7] = point(0.707107, 2, -0.5);
36 points[8] = point(0, 2, -0.5);
37 points[9] = point(0, 1, -0.5);
38 points[10] = point(0, 0.5, -0.5);
39 points[11] = point(-0.5, 0, -0.5);
40 points[12] = point(-1, 0, -0.5);
41 points[13] = point(-2, 0, -0.5);

```

Fig. 1 blockMeshDict の一部分

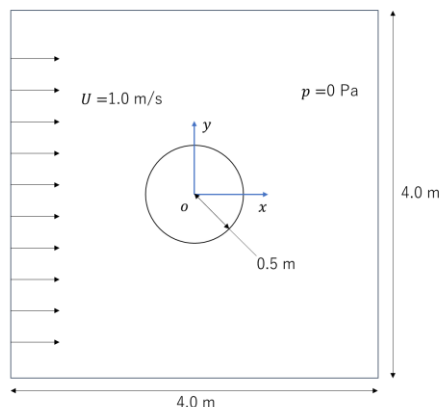


Fig. 2 解析条件 ②

ファイルを準備する必要がある。Fig.1はblockMeshDictの一部を示しており、ベース格子生成の代表的な格子点座標の設定項目部分である。

3. 流れの解析条件

今回は、検証のために、設定ファイルが比較的少ないポテンシャル流れソルバーであるpotentialFoamを用いて、Fig.2のような円柱周りの流れを対象に解析を行った。半径0.5mの円柱を中心に、流れ方向をx方向として、x, y方向に4m、z方向に1mの計算領域を設定した。境界条件は、左面を流入境界として流入速度一定 $U = (1,0,0)$ m/s、右面を流出境界とし

Table 1 各設定ファイルの生成回数

| 設定ファイル名 | 生成回数 |
|-------------------|------|
| blockMeshDict | 生成失敗 |
| snappyHexMeshDict | 7 |
| controlDict | 1 |
| fvSchemes | 1 |
| fvSolution | 2 |
| U | 2 |
| p | 2 |

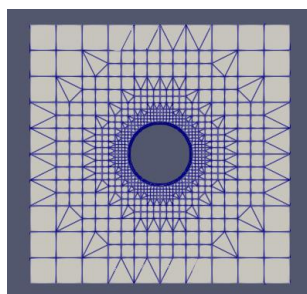


Fig. 3 作成されたメッシュ

て圧力一定 $p = 0 \text{ Pa}$ とした。また、初期条件は $U = 0 \text{ m/s}$, $p = 0 \text{ Pa}$ とした。

4. ChatGPTを用いた設定ファイルの生成と流れ解析結果

本研究では、ChatGPTへ解析条件を指定し、生成された設定ファイルを用いて解析を行った。実行時エラーが発生した場合は、エラー文をそのままChatGPTに送信することで設定ファイルの修正を行った。Table 1に、ChatGPTで生成した設定ファイルと生成に要した生成回数を示す。

まず、blockMeshを用いて、Fig.2のような計算領域に対する境界適合の構造格子を作成するための設定ファイルblockMeshDictの生成を行ったところ、同じエラーが繰り返され、設定ファイルの生成に失敗した。これは、言語生成AIでは、現状2桁の足し算のように数字を扱うこと苦手であり、blockMeshにおける格子生成において、座標の把握まではできたものの、座標からなる面の定義に失敗したことが原因と考えられる。そこで、blockMeshでベースとなる計算格子を生成し、snappyHexMeshで境界適合の非構造格子を生成する方法に変更した。この場合、円柱の形状データが必要となるが、その形状データ作成のためのPythonコードについてもChatGPTで生成した。snappyHexMeshDictの生成に際しては、円柱形状の内側にメッシュを生成するような設定

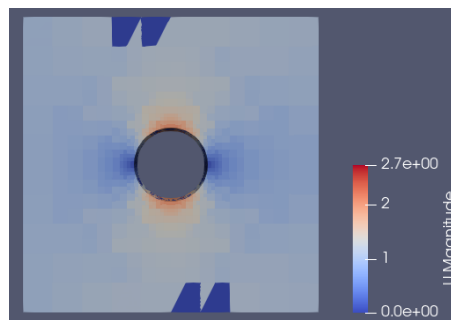


Fig. 3 円柱周りの速度分布

ファイルが初期に生成されたが、誤った設定項目を指摘することで適切な修正を行い、その結果、Fig.3のような円柱周りのメッシュが作成された。また、ソルバーの制御に関する設定ファイルであるcontrolDict、fvSchemes、fvSolution、速度と圧力に対する境界条件設定ファイルU、pの生成については、1~2回とかなり少ないChatGPTとの対話によって設定ファイルの生成が実現できた。以上のChatGPTによって生成した設定ファイルを用いてポテンシャル流れの解析を行った結果をFig.4に示す。円柱の上部と下部で流速が速くなっており、定性的に妥当な結果が得られた。

5. 結言

言語生成AIであるChatGPT-4に流れの解析条件を指定することで、OpenFOAMの設定ファイルを生成した結果、完全ではないものの解析を行うことができた。講演では、ChatGPTと対話内容について具体的に示し、設定ファイルの生成方法について説明する。

参考文献

- 1) RIOS, Thiago; MENZEL, Stefan; SENDHOFF, Bernhard. Large Language and Text-to-3D Models for Engineering Design Optimization. arXiv preprint arXiv:2307.01230, 2023.
- 2) OpenCFD Ltd 2023, 2.2 Flow around a cylinder. OpenFOAM Tutorial Guide. (2023). <https://www.openfoam.com/documentation/tutorial-guide/2-incompressible-flow/2.2-flow-around-a-cylinder/>(参照 2023-10-8)