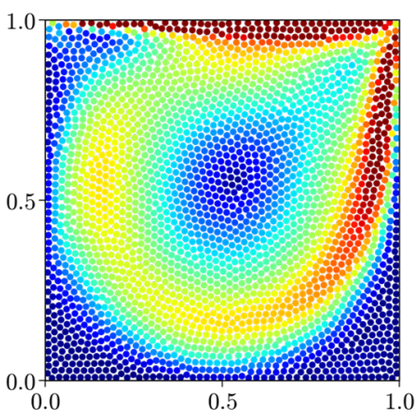
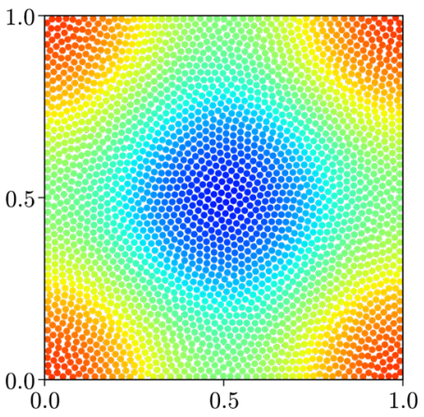
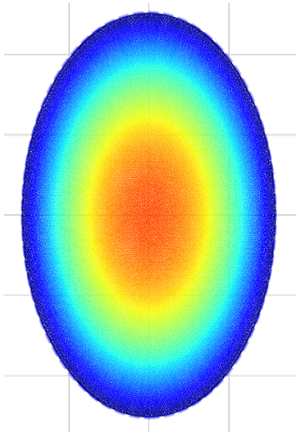
LS-SPH Fluid Simulator
User Manual

菖蒲迫健介・藤川浩希・久賀凌・吉田茂生







第1版 (2025/6/13)

本書は LS-SPH-Fluid-Simulator のユーザーマニュアルです.

はじめに

本プログラムは筆者らの SPH 法を用いた流体計算に関する研究成果をまとめたもので，本マニュアルはその内容および実装に関する覚書です．SPH 法 [1,2] とは，計算点が空間的に固定されない粒子法 (メッシュフリー法) の代表例であり，連続体の合体・分裂，自由表面を含む混相流，複雑なジオメトリーを持つ流体計算の取り扱いに長けています．LS-SPH 法 [3] とは，筆者らが開発した SPH 法の一般化モデルであり，最小二乗法に基づいて定式化された高精度な SPH 法のことを言います．本プログラムでは，LS-SPH 法 (あるいは従来の SPH 法) を用いた流体計算の標準的な例題や発展的な応用例を提供します．本マニュアルは，本プログラムの構成や使用方法，入力ファイルの設定，解析結果の可視化などについて解説します．

本マニュアルは，LS-SPH 法（あるいは SPH 法）の導入を検討している研究者・技術者を主な対象とし，数値計算や流体解析に関する基礎的な知識を有する読者を想定しています．一部の例題を除き，一般的な PC 環境での実行を前提としており，SPH 法に初めて触れる方にも配慮した内容となっています．今後，より複雑な流体解析に取り組む際の出発点として，本マニュアルが有用な資料となることを願っております．

＊＊＊

筆者らの研究遂行にあたっては，多くの方々より温かいご指導とご支援を賜りました．特に，東京大学の川田佳史さん，神戸大学の中島涼輔さんには多大なご協力を頂きました．川田さんには研究の細部に至るまで丁寧なご助言を頂き，困難な課題に直面した際には根本から問題を見直し，多くの有益な解決策をご提案して頂きました．中島さんには，研究遂行に不可欠であった数理的テクニックの数々を賜り，また解析的計算の細部に渡るチェックをして頂きました．

さらに，九州大学工学府の浅井光輝さんには，SPH 法を中心とした数値流体解析に関する最先端の知見を惜しみなくご教授頂きました．また，浅井研究室の門下である辻勲平さん，藤岡秀二郎さんには，SPH 法に関する未発表の知見やコード開発に関する基礎的な情報をご提供頂きました．特に藤岡さんには，筆者の菖蒲迫を浅井研究室にご紹介いただき，SPH 法を用いた一連の研究における大きな転機を与えて下さいました．加えて，九州大学理学府の中村草平さんには，数多くの解析的計算を手伝って頂き，物理学の観点に基づく貴重なご助言を賜りました．

筆者らの研究に際して，多くの先生方，友人，そして家族に様々な形で応援して頂きました．この場をお借りして，深く感謝申し上げます．

■ 改訂歴

- 第 0 版 (2024/03/19)：プログラムとマニュアルの作成を開始
- 第 1 版 (2025/06/13)：第〇部〜第△部まで作成

■ 著者と担当

著者	所属 * と担当箇所	作成版
菖蒲迫健介	九州大学大学院理学府地球惑星科学専攻 博士 3 年	1
	プロジェクト統括 / マニュアル第〇部執筆 / プログラム全体の設計および実装	
藤川浩希	九州大学大学院理学府地球惑星科学専攻 修士 1 年	1
	第〇部執筆 / △実装	
久賀凌	九州大学理学部地球惑星科学科 学部 4 年	1
	第〇部執筆 / △実装	
吉田茂生	九州大学大学院理学研究院地球惑星科学部門 准教授	1
	監修 (指導教員)	1

* 作成当時の所属.

■ 著作権

目次

第 1 部
プログラムの概要と利用方法

第 1 部では，本プログラムの概要および利用方法について説明します．内容には，本プログラムの基本的な構造，ファイルの命名規則や依存関係などが含まれます．

1　概要

本プログラムは，粒子法 (メッシュフリー法) の一種である Smoothed Particle Hydrodynamics (SPH) 法 [1,2] および，その高精度版である Least Squares SPH (LS-SPH) 法 [3] に基づいた流体解析コードです．本プログラムの特色は以下の通りです．

本プログラムの特色

- Fortran による SPH 法あるいは LS-SPH 法の流体計算プログラムを提供
- Python による作図テンプレートを提供
- 固定壁条件 / ~~周期的境界条件~~ (工事中) / ~~自由境界条件~~ (工事中) を利用可能
- バックグラウンドセルを用いた近傍粒子探索アルゴリズム (e.g., [4]) を実装
- 複数の離散化モデル (例：Standard SPH, Corrected SPH [5–7], LS-SPH) を選択可能
- OpenMP を利用した共有メモリ並列計算に対応

1.1　実装されている計算例

本プログラムに実装されている計算例を表 1.1 に示します．各セットアップや代表的な結果などの詳細については，第 4 部をご参照ください．新しい計算例は今後のアップデートにて順次追加していく予定です．

表 1.1: 本プログラムに実装されている流体計算例.

計算例	説明	ディレクトリ
Taylor–Green vortex	非圧縮性粘性流体の運動エネルギーが粘性によって散逸する問題	2d_fixed_wall
Lid-driven cavity flow	上壁境界に一定の速度を与える問題	2d_fixed_wall
Boussinesq convection	熱膨張のみによって浮力が生じる対流問題 (ブシネスク対流)	2d_fixed_wall
~~Oscillating drop~~	~~自由表面を有する液滴が中心力によって振動する問題~~ (工事中)	2d_free_surface

2　利用方法

2.1　実行環境

本プログラムは **Linux 環境下での実行を想定しており，流体計算を Fortran が担い，解析や作図を Python が担います**．Fortran のコンパイラは Intel Fortran で，プログラム中では Intel MKL (Math Kernel Library) を使用します．また，Fortran コードのビルドには make を利用します．

なお，SPH 法による流体計算は計算負荷が高いため，OpenMP を用いた共有メモリ並列計算の使用を推奨します (デフォルト設定)．Python のバージョンは 3.10.12 を使用しており，解析には `matplotlib` や `scipy` などのライブラリを利用します．その他，MP4 形式の動画作成のために `ffmpeg` を利用します．

以上の必要な環境は，各自でご用意ください．筆者らは WSL2 (Windows Subsystem for Linux 2) を用いてこれらの環境構築を行いました[*1]．その際の構築手順を Qiita 記事として公開しております．なお，`ffmpeg` も忘れずにインストールしておいてください．

- WSL2 のインストールとアンインストール[*2]
  (https://qiita.com/zakoken/items/61141df6aeae9e3f8e36)
- WSL2 による gfortran と intel fortran の環境構築[*3]
  (https://qiita.com/zakoken/items/2a5e629020ce68f3efe1)
- WSL2 による Python3 の環境構築
  (https://qiita.com/zakoken/items/8ddfda7267e7d95b3c46)

2.2　実行手順

Taylor–Green vortex を例として，その実行手順を説明します．なお，以下の「編集」箇所は，配布時のデフォルト状態で予め設定済みです．その他の計算例のパラメータ設定については，第 4 部をご参照ください．

実行手順

手順 (1)　該当ディレクトリに移動します (本例では `sph_code` → `2d_fixed_wall`)

手順 (2)　`source_code` ディレクトリに移動し，`makefile` を編集して計算系を選択します

手順 (3)　続けて，`input.f90` を編集し，系のパラメータや物性値を設定します

手順 (4)　ターミナルから `make` を実行し，Fortran コードをビルドします
以下のような出力があればビルド成功です：

```
[message] compilation has finished
[message] input ./start_calculation
```

手順 (5)　続けて，`./start_calculation` を実行すると，計算が始まります

手順 (6)　計算終了後，同じディレクトリにある `TG_main.py` を実行して解析を行います

[*1] WSL2 とは Windows 上で仮想的に Linux を動作させることができるシステム環境です．
[*2] WSL2 では make がデフォルトで利用可能です．
[*3] この方法でインストールした場合，MKL がデフォルトで利用可能です．

3　プログラムの構造

3.1　ディレクトリ構成

全ての計算および解析コードは，`sph_code` ディレクトリ配下に格納されています．この親ディレクトリの下には，空間次元 (例：2D, 3D) や境界条件 (例：固定壁，自由表面境界) に基づいて分類された子ディレクトリが配置されています．各ディレクトリに対応する計算例の一覧は，表 1.1 をご参照ください．

- sph_code
  - 2d_fixed_wall　　（2 次元・固定壁境界）
  - 2d_free_surface　（2 次元・自由表面境界）
  - 2d_multi_boundary（2 次元・固定壁＋自由表面境界）

3.2　各ディレクトリの構造

`2d_fixed_wall` ディレクトリの構造を例として説明します．各ソースコードの詳細は第 3 部をご参照ください．続く，3.3 節および 3.4 節で，各ファイルの命名規則と依存関係を述べます．

- 2d_fixed_wall
  - initialize_local.py　（配布時の状態にリセットする python コード）
  - fig　（図の格納場所 ← プログラム実行時に作成される）
  - output　（計算結果の格納場所 ← プログラム実行時に作成される）
  - reference_solution　（参照解の格納場所）
  - source_code　（ソースコードの格納場所）
    - ana_○○.py　（解析コード（Python）で用いるモジュール）
    - BC_○○.py　（Boussinesq Convection（BC）用の解析コード）
    - CF_○○.py　（Lid-driven Cavity FLow（CF）用の解析コード）
    - TG_○○.py　（Taylor--Green vortex（TG）用の解析コード）
    - cal_○○.f90　（計算を行うサブルーチン）
    - check_compile_options.f90　（コンパイルオプションの確認コード）
    - global_variables.f90　（グローバル変数を格納したファイル）
    - input.f90　（入力ファイル）
    - lib_○○.f90　（計算で用いるモジュール）
    - main.f90　（メインプログラム）
    - Makefile　（ビルドを行う make ファイル）
    - out_○○.f90　（書き出しに関するサブルーチン）
    - set_○○.f90　（計算の準備を行うコード）

3.3　命名規則

source_code ディレクトリ内のソースコードは，以下の規則に基づいて命名されています．

ファイルの命名規則

- 解析コード (Python ファイル) は，各計算例の頭文字などを接頭辞として使用
  (例：Taylor–Green vortex → TG_〇〇.py)
- ただし，全ての解析コードで共通なソースコードには ana_〇〇.py と命名
  (例：カーネル関数や Fortran による書き出しファイルを読み込む関数など)
- 計算コード (Fortran ファイル) は以下に準拠
  ・計算を行うサブルーチンの接頭辞には cal を付与・・・SPH の相互作用計算など
  ・計算で用いるモジュールの接頭辞には lib を付与・・・カーネル関数や行列演算など
  ・書き出しサブルーチン　の接頭辞には out を付与・・・書き出しなど
  ・初期設定サブルーチン　の接頭辞には set を付与・・・配列の設定など
  ・それ以外の Fortran ファイルの説明は 3.2 節を参照

3.4　依存関係

計算コード (Fortran ファイル) には明確な依存関係が存在し，これは Makefile に記述されたコンパイル順と一致しています．最上位には，多くのファイルから参照される input.f90 および global_variables.f90 モジュールが配置されます．次に，関数モジュール lib_〇〇.f90 が続き，その後に具体的な処理を行う set_〇〇.f90，cal_〇〇.f90，out_〇〇.f90 が並びます．最後に，メインプログラム main.f90 がこれらすべてを呼び出す構成となっています．

なお，set_〇〇.f90，cal_〇〇.f90，out_〇〇.f90 は基本的に相互に独立していますが，一部において依存関係が存在します．こうした依存関係は main.f90 における呼び出し順に対応しています．

解析コード（Python ファイル）にも依存関係はありますが，Python では明示的なコンパイルが不要なため，本マニュアルでは詳細な説明を省略します．

第 2 部
アルゴリズムの概要

第 2 部では，本プログラムに実装されている基本的なアルゴリズムについて概説します．内容には，基礎方程式や離散化手法などを含む数値計算法，近傍粒子探索のアルゴリズム，および全体の計算フローが含まれます．

1　数値計算法

圧縮性が小さな流れを陽解法を用いて計算[*4]

1.1　基礎方程式

1.2　離散化手法

1.3　時間積分法

1.4　境界処理法

1.5　安定化手法

2　近傍粒子探索アルゴリズム

3　計算フロー

[*4] マッハ数が小さい流れを弱圧縮性近似 [8] を用いて計算する weakly compressible SPH (WCSPH) 法 [9, 10] を採用.

第 3 部
ソースコードの詳細

第 3 部では，ソースコードの中身を解説します．最も基礎的な `2d_fixed_wall` というディレクトリを対象として，主要なソースコードや，重要な変数・配列の構造について説明した後，各ソースコードを詳細に解説します．最後に，`sph_code` 直下に存在する他のディレクトリにおける `Makefile` および `input.f90` のパラメータについても説明します．

第 4 部
実装済みの計算例

第 4 では本プログラムに実装されている数値計算例について，各セットアップの概要および得られた代表的な結果を簡潔にご紹介します．

1 Taylor–Green vortex

1.1 問題設定

lid-driven cavity flow は，Ghia et al. (1982) によって考案されました．このテストでは，正方形領域の上壁に一定速度を与え，その定常解をベンチマークとして使用します (図 2.1)．

■ 基礎方程式

■ 境界条件

■ パラメータ

1.2 解析例

2 Lid-driven cavity flow

2.1 問題設定

lid-driven cavity flow は，Ghia et al. (1982) によって考案されました．このテストでは，正方形領域の上壁に一定速度を与え，その定常解をベンチマークとして使用します (図 2.1)．

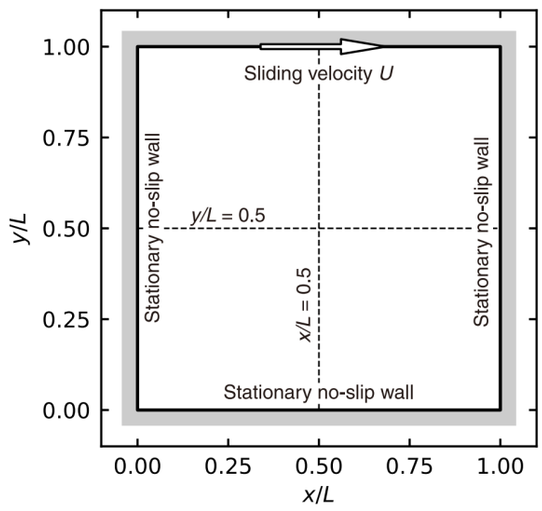


図 2.1: lid-driven cavity flow の概要．図は Matsunaga et al. (2020) z の Fig. 23 を引用．

■ 基礎方程式

■ 境界条件

■ パラメータ

2.2 解析例

参考文献

[1] L. B. Lucy, A numerical approach to the testing of the fission hypothesis, The Astronomical Journal 82 (1977) 1013–1024.
URL https://doi.org/10.1086/112164

[2] R. A. Gingold, J. J. Monaghan, Smoothed particle hydrodynamics: theory and application to non-spherical stars, Monthly Notices of the Royal Astronomical Society 181 (3) (1977) 375–389.
URL https://doi.org/10.1093/mnras/181.3.375

[3] K. Shobuzako, S. Yoshida, Y. Kawada, R. Nakashima, S. Fujioka, M. Asai, A generalized smoothed particle hydrodynamics method based on the moving least squares method and its discretization error estimation, Results in Applied Mathematics 26 (2025) 100594.
URL https://www.sciencedirect.com/science/article/pii/S2590037425000585

[4] C. E. Rhoades, A fast algorithm for calculating particle interactions in smooth particle hydrodynamic simulations, Computer Physics Communications 70 (3) (1992) 478–482.
URL https://www.sciencedirect.com/science/article/pii/001046559290109C

[5] P. Randles, L. Libersky, Smoothed particle hydrodynamics: Some recent improvements and applications, Computer Methods in Applied Mechanics and Engineering 139 (1) (1996) 375–408.
URL https://www.sciencedirect.com/science/article/pii/S0045782596010900

[6] J. K. Chen, J. E. Beraun, T. C. Carney, A corrective smoothed particle method for boundary value problems in heat conduction, International Journal for Numerical Methods in Engineering 46 (2) (1999) 231–252.
URL https://onlinelibrary.wiley.com/doi/abs/10.1002/%28SICI%291097-0207%2819990920%2946%3A2%3C231%3A%3AAID-NME672%3E3.0.CO%3B2-K

[7] J. Bonet, T.-S. Lok, Variational and momentum preservation aspects of smooth particle hydrodynamic formulations, Computer Methods in Applied Mechanics and Engineering 180 (1) (1999) 97–115.
URL https://www.sciencedirect.com/science/article/pii/S0045782599000511

[8] A. J. Chorin, A numerical method for solving incompressible viscous flow problems, Journal of Computational Physics 135 (2) (1997) 118–125.
URL https://www.sciencedirect.com/science/article/pii/S0021999197957168

[9] J. Monaghan, Simulating free surface flows with sph, Journal of Computational Physics 110 (2) (1994) 399–406.
URL https://www.sciencedirect.com/science/article/pii/S0021999184710345

[10] J. P. Morris, P. J. Fox, Y. Zhu, Modeling low reynolds number incompressible flows using sph, Journal of Computational Physics 136 (1) (1997) 214–226.
URL https://www.sciencedirect.com/science/article/pii/S0021999197957764