粒子分散系直接数値計算シミュレータ
KAPSEL
Version 5.12

ユーザーズマニュアル

KAPSEL Development Team

kapsel.dev@gmail.com

2024年2月1日

目次

第1章

はじめに

1.1 KAPSELとは

KAPSEL[*1]とは，粒子分散系 (流体中を粒子が自由に動き回る系) のシミュレーションに特化したソフトウェアである．流体中に分散した粒子の運動をシミュレーションする場合，粒子間に働く間の直接的な相互作用のみならず，流体の運動を介して分散粒子間に働く「流体力学相互作用」と呼ばれる間接的な相互作用が重要となる．したがって，流体に分散した固体粒子の複雑な動きを再現するためには，粒子間に存在する流体の運動を考慮する必要が生じる．これを実現するための代表的な手法が，粒子のサイズより小さいスケールの流体運動をNavier–Stokes方程式に基づいて正確に捉えながら，各時刻において得られた流体の局所応力から分散粒子間に働く相互作用を求める「直接数値計算 (Direct Numerical Simulation: DNS)」と呼ばれる方法である． DNSは一般に計算負荷が大きいので，Navier–Stokes方程式の数値計算を回避する方法の開発も行われている．例えば，運動量保存則を満たす粒子系で流体をモデル化するDissipative Particle Dynamics (DPD) やMulti-Particle Collision Dynamics (MPC) ，流体運動を陽に解かずに粒子間に働く流体力学相互作用を定式化するBrownian Dynamics (BD) やStokesian Dynamics (SD) などの方法が提案されている．これらの簡便な方法とは異なり，流体運動を基礎方程式に基づいて正確に解くことがDNSのもっとも大きな特長であり，それによって単純なNewton流体だけではなく複雑流体への拡張も可能となる．

DNSでは，粒子を点ではなく有限体積をもつ物体として表現し，固液境界面を通して両者の間で正しい運動量交換を実現する．流体中を自由に動く任意形状の固体と，その周りの流体との連成問題を数値計算で取り扱うための最も標準的な方法は，有限要素法 (FEM) と呼ばれる固体形状に沿った境界適合格子 (不規則格子) を時々刻々再生成する方法である．しかし，多数の固体粒子を含む分散系の場合，FEMでは計算ステップ毎に膨大な数の格子点を再構成する必要が生じ，計算コストが莫大になる．この問題を回避しつつ多粒子分散系のDNSを実現する方法として，我々は分散粒子と流体の間のシャープな境界面を有限の厚みを持ったプロファイル関数で置き換えるSmoothed Profile (SP) 法を独自に開発し，計算精度と計算効率の両立に成功した． KAPSEL (Kyoto Advanced Particle Simulator for ELectrohydrodynamics)とは，SP法を実装して粒子分散系のDNSを行うために我々自身が開発したシミュレーターの呼称である．本マニュアルでは，KAPSELの基本原理やインストール方法に関する解説に加え，具体的なサンプル用いた計算事例を紹介する. SP法の詳細やその背景については総説[1]をご覧いただきたい.

[*1] 京都大学大学院工学研究科化学工学専攻のソフトマター工学研究室で開発されたソフトウェアの名称である．version4まではフリーウェアとしてソースコードを公開し，License Agreementの下で自由な仕様を許可してきたが，version5以降は商用ソフトウェアとして有償で実行ファイルのみを提供する．ソースコードの閲覧・改変は別途秘密保持契約(NDA)を締結し行うものとする．現在までのすべてのversionについて，KAPSEL本体の開発責任者は山本量一教授，著作権保有者は京都大学と山本量一教授である．

1.2　KAPSELの機能概要

KAPSELの各種機能は，後述する入力UDF (input.udf) ファイルを編集することによって制御できる．以下に機能の概要を説明する．

1.2.1　Newton 流体に分散した粒子のシミュレーション

constitutive_eqとしてNavier_Stokesを選んで，Newton 流体中に分散した粒子のシミュレーションを行うことができる．この機能により，重力沈降する任意形状粒子の運動の予測や，ブラウン運動による分散粒子の熱拡散や凝集挙動の評価が可能となる．詳しくは第3章で解説する．constitutive_eqとしてNavier_Stokes_FDMを選んだ場合も同様であるが，この場合は流体ソルバーとして有限差分法(FDM)が使用される．

1.2.2　せん断流下の粒子分散系のシミュレーション

constitutive_eqとしてShear_Navier_Stokes_Lees_Edwardsを選んで，せん断流下のニュートン流体中に分散した粒子のシミュレーションを行うことができる．この機能により，せん断流下における任意形状粒子の運動の予測や，粒子分散系のレオロジー特性の評価が可能となる．詳しくは第4章で解説する．constitutive_eqとしてShear_Navier_Stokes_Lees_Edwards_FDMを選んだ場合も同様であるが，この場合は流体ソルバーとして有限差分法(FDM)が使用される．

1.2.3　電解質溶液に分散した荷電コロイド粒子のシミュレーション

constitutive_eqとしてElectrolyteを選んで，電解質溶液に分散した荷電コロイド粒子のシミュレーションを行うことができる．この機能により，荷電コロイド系の安定構造の予測や，定常(DC)電場/振動(AC)電場下における荷電コロイド粒子の電気泳動特性の評価が可能となる．詳しくは第5章で解説する．

1.2.4　二成分相分離流体に分散した粒子のシミュレーション

constitutive_eqとしてNavier_Stokes_Cahn_Hilliard_FDMを選んで，二成分相分離流体中に分散した粒子のシミュレーションを行うことができる．この機能により，流体の各成分や界面と粒子との親和性を変化させた場合の粒子分布や，流体の相分離構造の予測が可能となる．また，constitutive_eqとしてShear_NS_LE_CH_FDMを選んで，せん断流下の二成分相分離流体中に分散した粒子のシミュレーションを行うことができる．この機能により，せん断流下における相分離構造の予測や，二成分相分離流体中に粒子が分散した系のレオロジー特性の評価が可能となる．詳しくは第6章で解説する．

1.2.5　マイクロスイマーのシミュレーション

constitutive_eqとしてNavier_Stokesを選んで，ニュートン流体中に分散したマイクロスイマー粒子のシミュレーションを行うことができる．この機能により，多数のマイクロスイマー粒子が干渉して生じる集団運動の予測などが可能となる．詳しくは第7章で解説する．constitutive_eqとしてNavier_Stokes_FDMを選んだ場合も同様であるが，この場合は流体ソルバーとして有限差分法(FDM)が使用される． constitutive_eqとしてNavier_Stokes_Cahn_Hilliard_FDMを選ぶと，二成分相分離流体中に分散したマイクロスイマー粒子のシミュレーションを行うことができる．

1.2.6　クインケローラーのシミュレーション

constitutive_eqとしてNavier_Stokesを選んで，Newton流体中の平板上を転がって自己推進するクインケローラーのシミュレーションを行うことができる．この機能により，多数のクインケローラーが干渉して生じる集団運動の予測などが可能となる．詳しくは第8章で解説する. constitutive_eqとしてNavier_Stokes_FDMを選んだ場合も同様であるが，この場合は流体ソルバーとして有限差分法(FDM)が使用される．

1.3　本書の構成

KAPSELの実行に必要なソフトウェアやライブラリのインストール・ビルド方法を第2章で詳しく説明する．第3章以降の各章では，その章で取扱う具体的な問題を計算するための「理論的背景と基礎方程式」について詳しく述べた後，シミュレーションに必要なパラメータの設定を行う「入力UDF」について必要な情報に絞って説明し，最後に具体的な「計算事例」を紹介する．入力UDFファイルに関する網羅的な解説は付録Aで行う．

第2章

KAPSELのインストールと実行

2.1 KAPSELのインストール手順および実行方法

2.1.1 KAPSELの動作環境

KAPSELの動作環境について，Windows，Linux，Macにおける注意点を以下に示す．

Linuxの場合

Linux上でKAPSELを利用する場合，2.1.2節へと移動.

Windowsの場合

KAPSEL をWindowsシステムで使用するには，Windows Subsystem for Linux（WSL）[*1]をインストールする必要がある． PowerShell または Windows コマンドプロンプトを右クリックして管理者モードで開く．以下のコマンドを実行した後にPCを再起動したら2.1.2節へと移動.

```
$ wsl --install
```

Macの場合

MacOS上でKAPSELを利用する場合，Xcodeとcommand line tools のインストールが必須である．インストール後に2.1.2節へと移動.

2.1.2 OCTAのインストール

KAPSELは, ソフトマテリアルに対する統合的なシミュレータとして開発されたOCTA内部にあるGourmetとよばれるユーザインターフェースと連携し入力パラメータの管理や出力データの可視化をおこなっている. そこで, KAPSELを使用するためにはOCTAがインストールされていることが前提となっている.

インストールの手順として，まずOCTAホームページ `http://octa.jp/` にアクセスし，適当なインストーラをダウンロードする．その後，インストーラを実行し，OCTAをインストールする[*2]．以降，`OCTA8.#` が `/usr/local/OCTA8#` (Linux/Mac) または `C:\OCTA8.#` (Windows) にインストールされているとして手順を説明する[*3].

2.1.3 KAPSELのインストール

最新バージョンのKAPSELパッケージ `kapsel#.#.zip`[*4] をダウンロードし，圧縮ファイルを解凍する.

[*1] https://learn.microsoft.com/ja-jp/windows/wsl
[*2] OCTA/GOURMETに関する質問は，OCTA–BBSのメンバーになることで可能である.
[*3] インストールしたOCTAのバージョンに応じて，”#” に適当な数字を入力する.
[*4] バージョンに応じて ”#” に適当な数字を入力する

```
$ unzip kapsel#.#.zip
$ cd kapsel#.#
```

解凍後のディレクトリ構成は以下の通りである．

./bin/　各種プラットフォーム用のKAPSELの実行ファイルがある．
./Documents/　ユーザーマニュアル（本書）がある．
./Examples/　KAPSELの各種機能を用いた例題がある．
./UDF/　KAPSEL最新版のすべての機能を使用するための define.udf/input.udf がある．

./bin/ ディレクトリにある各種実行ファイルのうち，実行環境に適切なものをKAPSELのインストールディレクトリにシンボリックリンクする．

Linuxの場合

```
$ ln -s ./bin/linux/* .
```

Windowsの場合 Ubuntuのコンソールウインド（Windowsのコマンドラインではない！）にて以下のコマンドを実行する．

```
$ ln -s ./bin/linux/* .
```

Macの場合(Intel or Arm CPU)

```
$ ln -s ./bin/macOS/* .
```

2.1.4　Validating your KAPSEL license

商用のKAPSELバイナリを実行するには有効なライセンスが必要である．ライセンスには、ノードロックライセンスとクラスターライセンスの2種類があり，どちらも offline / online環境をサポートしている．ノードロックライセンスはマシンごとに発行され，アクティベートするためにはマシン固有のMachine Fingerprintが必要である．クラスタライセンスでは，実行を許可するUser Groupsのリストとともに，クラスター固有のEnviron Fingerprintが必要である．これらの識別子は，ライセンスが未設定の状態でkeyコマンドを実行すると以下のように表示される．

```
$ ./key
[INFO] System name : ...
[INFO] Machine     : ...
[INFO] Node name   : ...
[INFO] Release     : ...
[INFO] Version     : ...
[INFO] Network interface (en0) : ...
[INFO] Network interface (en1) : ...
...
[INFO] Host domain : ...
[INFO] Machine UUID: ...
# [Machine Fingerprint]
8bzm9b3xae00898539e754a3adba83bb11cdcea441d06401f5f3df350fcc7705
# [Environ Fingerprint]
22cca0e1fc66357d0aa8d238e7gg675a6d7ebcf323fdbbcef7e09c46c6259f17
# [User Groups]  user_group_1 user_group_2 ...
[ERROR] Environment variable KAPSEL_PUBLIC_KEY is missing
```

ノードロックライセンスの場合，KAPSELを実行するすべてのマシンについてMACHINE FINGERPRINTをライセンス供与者に提出し，ライセンスキー（文字列）とマシン固有のライセンスファイルを取得する必要がある．クラ

スターライセンスの場合(Linixのみ)，ENVIRON FINGERPRINTとクラスターでの実行を許可するユーザーグループ (e.g., user_group_1, user_group_2)のリストをライセンス供与者に提出し，ライセンスキー（文字列）とクラスター固有のライセンスファイルを取得する必要がある．必要なENVIRON FINGERPRINTを知るためには，ログインノード上ではなく，必ず計算ノード上でkeyコマンドを実行すること（KAPSELの実行と同じように，計算ノードにジョブを投入してkeyコマンドを実行しなければいけない)．どちらの場合でも，KAPSEL は起動後最初に有効なonlineライセンスを探し，見つからなければライセンスファイルの中で有効なofflineライセンスを探す．必ずライセンスキー（環境変数に設定する文字列）とライセンスファイルの両方が必要であることに注意する．ライセンスキーは，KAPSEL を実行する際に使用する個人のパスワードのようなもであり，他人と共有すべきではない．

KAPSEL 起動時にライセンス認証を行うには，環境変数を適切に設定する必要がある．この設定は，./binディレクトリにある setvars.shファイルを環境に合わせて編集した後にソースすることで行う．

```
    $ source ./bin/setvars.sh
```

上記操作で KAPSEL_ACCOUNT_ID， KAPSEL_PUBLIC_KEY， KAPSEL_LICENSE_KEY， KAPSEL_LICENSE_PATHが設定される．実行前に setvars.shファイルを編集し，特定のマシンIDに対して発行されたライセンスキーをKAPSEL_LICENSE_KEYに設定し，ライセンスファイルのパスをKAPSEL_LICENSE_PATHで指定する必要がある．ライセンスキーのフォーマットは key/your_kapsel_license_keyとなっている．ライセンスファイルは，以下のような暗号化された文字列を含むテキストファイルとして提供される．

```
-----BEGIN MACHINE FILE-----
eyJlbmMiOiJsSTc4N0QwcGZua1RvRDVOSjFpRXlaU093Q09QQ0NOdktZHpC
...
TlJWWGp6Ym5DRkF6V3lOU3NUeG9xZm9MV2FlWlhITEZnR21Ub2VBdz09Iiwi
YWxnIjoiYWVzLTI1Ni1nY20rZWQyNTUxOSJ9
-----END MACHINE FILE-----
```

KAPSEL_ACCOUNT_IDと KAPSEL_PUBLIC_KEY，およびライセンスファイルは勝手に変更するとライセンス認証ができなくなるので絶対に変更しないこと．

スクリプトで設定する代わりに，以下のようにコマンドラインで直接設定することもできる．

```
    $ export KAPSEL_ACCOUNT_ID="85e8531e-915a-4795-a287-2bec0d90ae5e"
    $ export
↪   KAPSEL_PUBLIC_KEY="5fae6bb532c12ef70e1ce5f69b33ecd4fea8a522658b1204a6203ee5f101f608"
    $ export KAPSEL_LICENSE_KEY="key/your_kapsel_licence_key"
    $ export KAPSEL_LICENSE_PATH="./license.lic"
```

環境変数が適切に設定され，有効なライセンスが存在すれば，KAPSEL を起動できる．ライセンスとマシンの認証に成功すると，以下のメッセージが表示される．

```
    $ ./key
    # [KAPSEL LICENSE VERIFICATION]
    # [Machine UUID] 02687FF3-FD77-51E9-9757-47A4E2AB0521
    # [Machine Fingerprint]
↪   8f85c6968218b225df7794d95c515f00f4c4eaa49d5b2ae5c6b98da850fa1505
    ...
    # [OK] Signature is valid!
    # [OK] License key is valid!  (code = "VALID")
    # [License ID] f8cc491f-a19f-424b-99a6-a0c9f093d69f
    #
    OK!
# [KAPSEL LICENSE VERIFICATION]
```

2.1.5　KAPSELのテスト

KAPSELを実行し，以下に示すような出力がコマンドライン上に表示された場合，KAPSELは正しくインストールされている．

```
$ cd UDF
$ ../kapsel -Iinput.udf -Ooutput.udf -Ddefine.udf -Rrestart.udf
#
# OMP RUNTIME :
# Number of processors        : 0
# Number of threads           : 8
# Max OMP threads             : 8
# Dynamic thread enabled?     : 0
# Nested parallelism enabled? : 1
#
#using input.udf as input
#using output.udf as output
#using define.udf as definition
#using restart.udf as restart
# [KAPSEL LICENSE VERIFICATION]
# [Machine Fingerprint]
↪ 8f85c6968218b225df7794d95c515f00f4c4eaa49d5b2ae5c6b98da850fa1505
# [OK] Signature is valid!
# [OK] License key is valid!  (code = "VALID")
# [License ID] f8cc491f-a19f-424b-99a6-a0c9f093d69f
#
    ...

#output.udf end.
#restart.udf end.
#Simulation has ended!
#Total Running Time (s):      24.77
#                   (m):       0.41
#                   (h):       0.01
#Average Step Time  (s):       0.02
#                   (m):       0.00
#                   (h):       0.00
```

上記のサンプルで利用したinput.udfは，5個の重い粒子と5個の軽い粒子がNewton流体中を沈降する現象を，$32 \times 64 \times 32$ のCFD[*5]計算格子上で解くための入力ファイルである．1分以内に完了するシミュレーションであり，正しく実行されていればoutput.udfが作成されているはずである．

マルチコア実行ファイルを使用する場合，KAPSELの実行前に以下の環境変数を設定することでコア数（下の例では8）を指定することができる．

```
$ export OMP_NUM_THREADS=8
```

2.1.6　シミュレーションデータの可視化方法

GOURMETでPythonスクリプトを用いることで，さまざまなシミュレーションデータを可視化することができる．以下に，簡単な例を示す．

- GROUMETを起動する
  - Windowsの場合: (OCTA install dir)\GOURMET\gourmet
  - Linuxの場合 : /usr/local/OCTA8#/GOURMET/gourmet

[*5] Computational Fluid Dynamics

– Macの場合：/usr/local/OCTA8#/GOURMET/gourmet

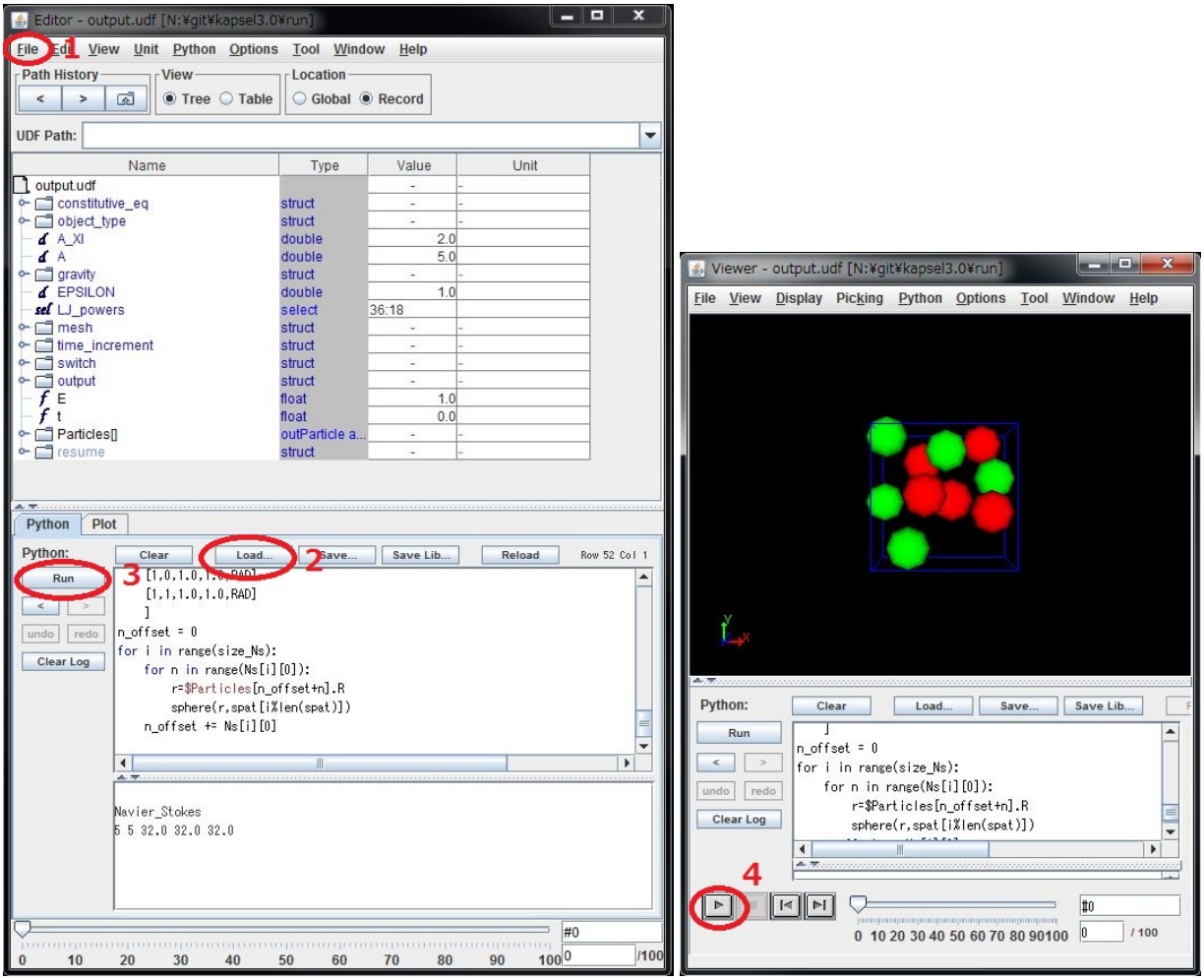
・output.udfを開く

File -> Open -> output.udf を開く [1]

各時間ステップにおける全粒子の位置および速度は，変数Particle[]内に格納されている．GROUMETビューアウィンドウの1番下にあるスライドバーを操作することで，異なる時刻における変数を読み取ることができる．

・GROUMET上でアニメーションを作成する

1. GROUMETビューアウィンドウ下部のPythonパネル内のLoadボタンをクリックする． [2]
2. particleshow.pyファイルを開き，Runボタンをクリックする． [3]
3. 新しいウィンドウが開かれ．そのウィンドウ上の再生ボタンをクリックする． [4]

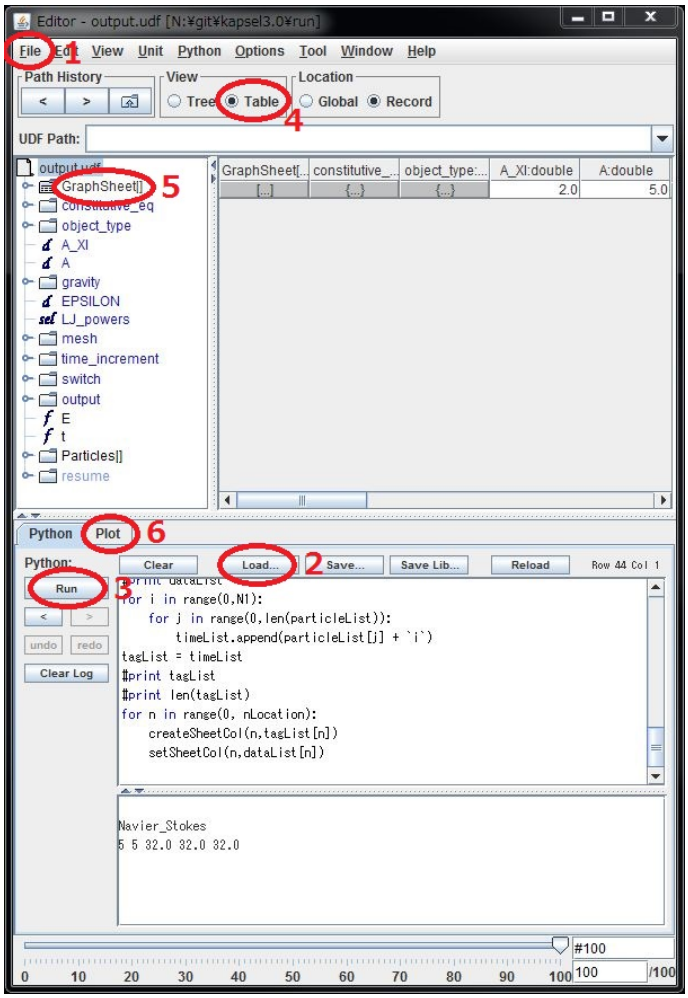


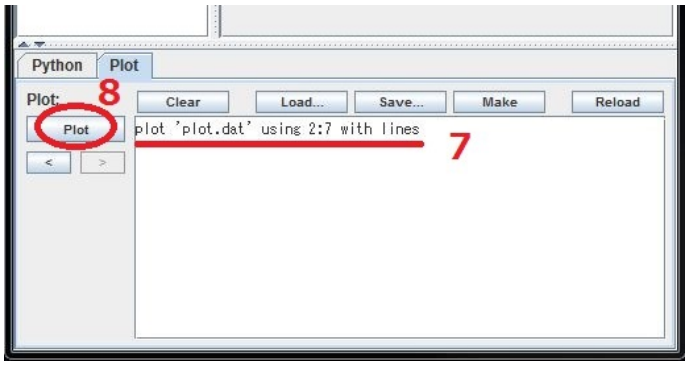
・GROUMET上でデータをプロットする (gnuplot)

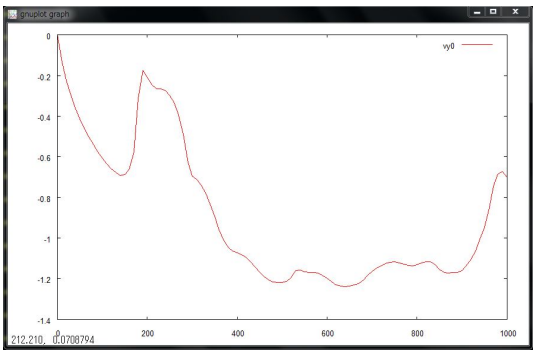
1. GROUMETビューアウィンドウ下部のPythonパネル内のLoadボタンをクリックする． [2]
2. plot.pyファイルを開き，Runボタンをクリックする． [3]
3. ウィンドウ上部のViewボックスからTableをクリックする． [4]
4. ウィンドウ左部のGraph Sheet[]を選択する． [5]
5. ウィンドウ下部のPlotパネルを選択し，以下のコマンドをコマンドボックスに入力する． [6,7]

6. Plotボタンをクリックすると，粒子 (粒子番号1) のVxについて時間発展を図示できる． [8]

```
plot 'plot.dat' using 2:7 with lines
```







2.1.7　シミュレーションデータの解析方法

シミュレーションを実行したときの計算データ (各時刻における粒子座標および速度) は output.udf に保存される． output.udf 内のデータには，以下に示すいずれかの方法でアクセスすることができる．

- Pythonコードで解析する場合

  Pythonコードを用いる場合，UDFファイルのデータにアクセスするためのUDFManager.py[*6] をインポートできる．添付されている sk.py は， output.udf 内に保存されている粒子位置の時系列情報から，静的構造因子 $S(k)$ を計算するPythonスクリプトである

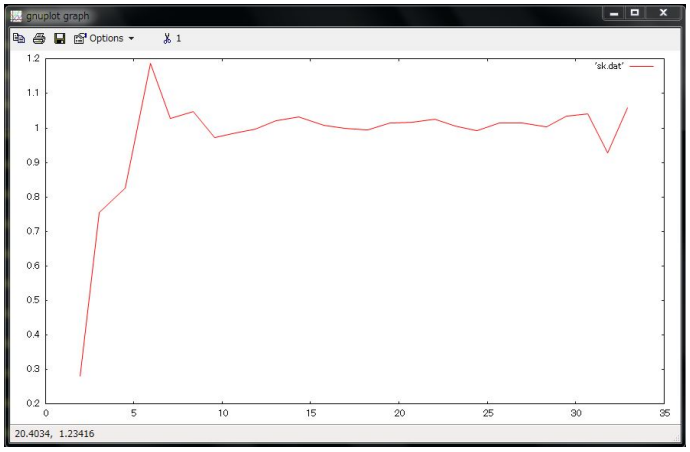
  本スクリプトを利用するには numpy パッケージの用意する必要がある．目的に応じて，本スクリプトの内容を編集しシミュレーションデータを解析することが可能である．

  本スクリプト sk.py を利用するには，まずGROUMETを起動し，

  - Windowsの場合: Start Menu > All Programs > OCTA > StartGourmetTerm
  - Linux/Macの場合: /usr/local/OCTA8#/GOURMET/gourmetterm を実行

  以下のコマンドを実行する．

```
python sk.py
gnuplot
>> plot 'sk.dat' w line
```



- Jupyter Notebook で解析する場合

  以下のように，Jupyter Notebook[*7] を利用して，上記と同様の解析を行うことができる．

```
$ . $PF_FILES/bin/gourmet_profile.sh
$ . $PF_FILES/bin/platform_env.sh
$ jupyter notebook sk.ipynb
```

- FortranまたはCで解析する場合

  FortranまたはCを用いる場合，UDFファイルのデータにアクセスするための libplatform[*8] ライブラリを利用できる．

KAPSELでは，いくつかの重要なデータ (各時刻における圧力等) が stderr に返される．通常の場合，それらデータはコマンドライン上で表示されるが，以下のようにして stderr をファイルにリダイレクトすることが可能である．

[*6] 詳細については「OCTA ソフトマテリアルのための統合化シミュレータ GOURMET PYTHON スクリプトリファレンスマニュアル」を参照.

[*7] Anaconda等で配布されている

[*8] 詳細については「OCTA ソフトマテリアルのための統合化シミュレータ プラットフォームインターフェースライブラリ libplatform リファレンスマニュアル」を参照.

- cshまたはtcshの場合

```
$ ../kapsel  -Iinput.udf -Ooutput.udf -Ddefine.udf -Rrestart.udf >& out
```

- sh，bash，Windowsコマンドプロンプトの場合

```
$ ../kapsel  -Iinput.udf -Ooutput.udf -Ddefine.udf -Rrestart.udf 2> out
```

2.2　入力UDFと定義UDFについて

入力UDF（input.udf）を用いて目的のシミュレーションを行うための設定を行う． input.udfの構造は定義UDF（define.udf）によって定義されており，異なるバージョンのinput.udfとdefine.udfを混在させることはできない． input.udfは次節以降の各設定項目で構成されている．より詳細な説明は付録Aを参照頂きたい．

2.2.1　流体の設定

constitutive_eqの設定ごとに内容が異なるので，それぞれ3章以降の「入力UDFについて」の節で説明する．

2.2.2　オブジェクト(粒子)の設定

object_type.typeでは，spherical_particle(球状粒子)，chain(フレキシブル鎖)，rigid(剛体)を選択できる．

粒子間の相互作用としてレナードジョーンズポテンシャルを導入できる．このポテンシャルは，

$$U_{LJ}(r) = \begin{cases} 4\epsilon\left[\left(\frac{\sigma}{r}\right)^{2n} - \left(\frac{\sigma}{r}\right)^{n}\right] + \epsilon & (r < r_{\mathrm{cut}}), \\ 0 & (r \geq r_{\mathrm{cut}}) \end{cases} \tag{2.1}$$

であり，$\sigma = d = 2a, \epsilon, r$は，それぞれ粒子直径，相互作用の強度，$i$番目と$j$番目の粒子間の距離をあらわしている．$n = 6, 12, 18$のいずれかを選択することができ，$n$が大きいほうが鋭いポテンシャルになる．排除体積による斥力相互作用のみの導入は，swithch.LJ_truncate をONにすることで可能であり，その場合はポテンシャルの打ち切り距離が$r_{\mathrm{cut}} = 2^{1/n}\sigma$に設定される．もし粒子間に引力相互作用を導入したい場合は，swithch.LJ_truncate をOFFにすることでポテンシャルの打ち切り距離が$r_{\mathrm{cut}} = 2.5\sigma$となり，通常のレナードジョーンズポテンシャルの計算が可能となる．

LJポテンシャルのべき指数の設定においてDLVOを選択すると，原子スケールに比べて大きくそれぞれ価数z_iとz_jを持つi番目とj番目の球状粒子間に働く相互作用ポテンシャルとして，以下のDLVOポテンシャルを選択できる[2].

$$V_{\mathrm{DLVO}}(r) = \begin{cases} V_{\mathrm{Coul}}(r) + V_{\mathrm{vdW}}(r) & (r \geq r^*) \\ V_{\mathrm{Coul}}(r) + V_{\mathrm{excv}}(r) & (r < r^*) \end{cases} \tag{2.2}$$

ここで

$$V_{\mathrm{Coul}}(r) = \frac{e^2}{4\pi\varepsilon_r\varepsilon_0} z_i z_j \left(\frac{\exp(\kappa a)}{\kappa a + 1}\right)^2 \frac{1}{r}\exp(-\kappa r) \tag{2.3}$$

は荷電粒子間に働く遮蔽クーロンポテンシャルである．

$$V_{\mathrm{vdW}}(r) = -\frac{A}{12}\left[\frac{d^2}{r^2 - d^2} + \frac{d^2}{r^2} + 2\ln\left(\frac{r^2 - d^2}{r^2}\right)\right] \tag{2.4}$$

は原子スケールに比べて大きい球状粒子間に働くvan der Waalsポテンシャル，

$$V_{\mathrm{excv}}(r) = \frac{K}{2}(r - d)^2 - V_m \tag{2.5}$$

は粒子の重なりを避けるための排除体積ポテンシャルであるが，$r = r^*(> d \equiv 2a)$において両者を切り替えて使用する．その際に両者の接続をなめらかに行う必要があるため，3つの係数A，K，V_mのうちAを決めると，残りの2つは以下の2式で自動的に与えられる．

$$K = \frac{A}{6}\frac{r^* d^2}{r^* - d}\left(\frac{1}{(r^{*2} - d^2)^2} + \frac{1}{r^{*4}} + \frac{r^{*2} - d^2}{r^{*6}}\right) \tag{2.6}$$

$$V_m = \frac{K}{2}(r^* - d)^2 + \frac{A}{12}\left[\frac{d^2}{r^{*2} - d^2} + \frac{d^2}{r^{*2}} + 2\ln\left(\frac{r^{*2} - d^2}{r^{*2}}\right)\right] \tag{2.7}$$

各ポテンシャルの概形を図2.1に示す．z_i，r^*，κa，$C \equiv \frac{e^2}{4\pi\varepsilon_r\varepsilon_0}$，$A$の値については，`input.udf`内に定義された以下の変数を通じて与える．

- `object_type.spherical_particle.Particle_spec[i].Surface_charge` = z_i
- `DLVO.n` = r^*/d
- `DLVO.kappa_a` = κa
- `DLVO.vdw_coeff` = A
- `DLVO.coulomb_coeff` = C

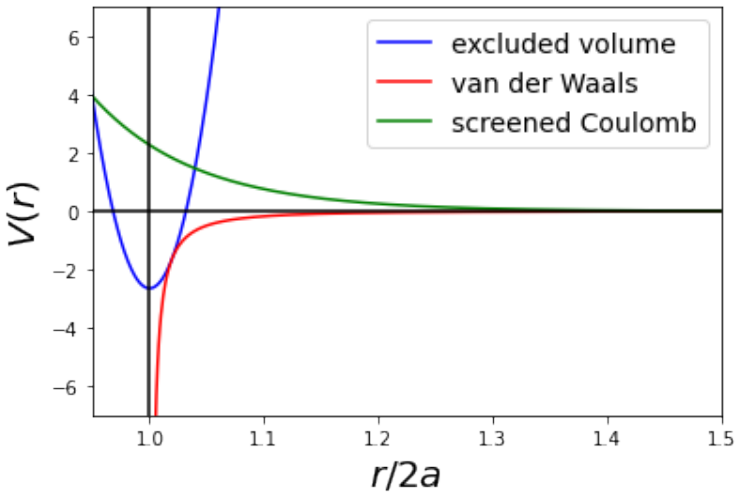


図2.1: DLVOポテンシャルを構成する各ポテンシャル項の概形．緑線は遮蔽クーロンポテンシャル$V_{\mathrm{Coul}}(r)$，赤線はvan der Waals(Hamarker)ポテンシャル$V_{\mathrm{vdW}}(r)$，青線は排除体積ポテンシャル$V_{\mathrm{excv}}(r)$を表す．この図では$d = 1$，$r^* = 1.02d$，$\kappa a = 5$，$\frac{e^2}{4\pi\varepsilon_r\varepsilon_0} z_i z_j \left(\frac{\exp(\kappa a)}{\kappa a + 1}\right)^2 = 50000$，$A = 1$を用いた．

LJポテンシャルのべき指数の設定において`electro_osmotic_flow`を選択すると，平板近傍の荷電粒子で重要となる電気浸透流による相互作用ポテンシャルを考慮できる．これは第 8章のクインケローラーでのみ使用する．

フレキシブル鎖と剛体は，複数の球状1次粒子の集合体として実現されている．したがってそれぞれの初期配置は，`switch.INIT_distribution=user_specify`を選択し，`input.udf`内で与える必要がある．フレキシブル鎖を選択した場合は，鎖内の隣接ビーズ間はFENE (finitely extensible non-linear elastic)ポテンシャル

$$U_F(r) = -\frac{1}{2}k_c R_0^2 \ln\{1 - (r/R_0)^2\}, \tag{2.8}$$

で結合される．ここで$k_c = 30\epsilon/\sigma^2$, $R_0 = 1.5\sigma$, r は隣接ビーズ間の距離をあらわしている．剛体を選択した場合は，拘束力により初期の粒子構造が保持される．

◯ object_type.spherical_particle.Particle_spec[]以下には球状粒子のタイプの情報を入れる．

- object_type.spherical_particle.Particle_spec[].Particle_number…粒子数.

- object_type.spherical_particle.Particle_spec[].MASS_RATIO...粒子の流体に対する密度比.
- object_type.spherical_particle.Particle_spec[].Surface_charge...総表面電荷:z (二成分相分離流体シミュレーションでは非対応).
- object_type.spherical_particle.Particle_spec[].janus_axis...粒子に固定された座標系におけるJanus軸の方向の指定.
- object_type.spherical_particle.Particle_spec[].janus_propulsion...Janus粒子の推進運動の指定. OFF(無効), TUMBLER(一定の外力で推進軸方向に推進する粒子), SQUIRMER(Slip境界条件により推進軸方向に推進するスクワマー粒子), OBSTACLE(固定された障害物粒子)から選択する.
- object_type.spherical_particle.Particle_spec[].janus_force.x...推進力のx成分の指定.
- object_type.spherical_particle.Particle_spec[].janus_force.y...推進力のy成分の指定.
- object_type.spherical_particle.Particle_spec[].janus_force.z...推進力のz成分の指定.
- object_type.spherical_particle.Particle_spec[].janus_torque.x...推進トルクのx成分の指定.
- object_type.spherical_particle.Particle_spec[].janus_torque.y...推進トルクのy成分の指定.
- object_type.spherical_particle.Particle_spec[].janus_torque.z...推進トルクのz成分の指定.
- object_type.spherical_particle.Particle_spec[].janus_slip_vel...自己推進粒子の表面滑り速度を決めるパラメータB_1.
- object_type.spherical_particle.Particle_spec[].janus_slip_mode...スクワマー粒子の泳動形態（Pusher/Neutral/Puller）を決定するパラメータB_2/B_1[*9].
- object_type.spherical_particle.Particle_spec[].janus_rotlet_C1...推進軸周りの回転（単極子）を表すパラメータC_1.
- object_type.spherical_particle.Particle_spec[].janus_rotlet_dipole_C2...推進軸周りの回転（双極子）パラメータC_2.

○ object_type.chain.Chain_spec[]ではフレキシブル鎖の情報を入れる.

- object_type.chain.Chain_spec[].Beads_number...一本の鎖に属するビーズの数.
- object_type.chain.Chain_spec[].Chain_number...鎖の本数.
- object_type.chain.Chain_spec[].MASS_RATIO...ビーズの流体に対する密度比.
- object_type.chain.Chain_spec[].Surface_charge...表面電荷(二成分相分離流体シミュレーションでは非対応).
- object_type.chain.Chain_spec[].janus_axis...ビーズに固定された座標系におけるJanus軸の方向の指定.

○ object_type.rigid.Rigid_spec[]では剛体の情報を入れる.

- object_type.rigid.Rigid_spec[].Beads_number...剛体を構成するビーズ数.
- object_type.rigid.Rigid_spec[].Rigid_number...剛体数.
- object_type.rigid.Rigid_spec[].MASS_RATIO...ビーズの流体に対する密度比. 複数のビーズが重なった部分の密度は二重にカウントしない.
- object_type.rigid.Rigid_spec[].Surface_charge...表面電荷(二成分相分離流体シミュレーションでは非対応).
- object_type.rigid.Rigid_spec[].Rigid_motion...剛体の運動の種類を, free(自由運動)とfix(並進・回転速度が固定された運動）のどちらかで選択する. fixでは、速度は以下で定義されるRigid_velocityとRigid_omega変数によって指定される通りとなる. デフォルトではこれは6つの自由度がすべて指定された値(ラボフレーム)に固定されることに注意すること. 並進・回転の自由度を個々に拘束するためには，後述のswitch.free_rigidオプションを使用する.

[*9] このパラメータが0より小さいときPusher型，0より大きいときPuller型のマイクロスイマーとなる[3, 4].

- object_type.rigid.Rigid_spec[].Rigid_velocity.x...剛体の速度のx成分の指定(ラボフレーム).
- object_type.rigid.Rigid_spec[].Rigid_velocity.y...剛体の速度のy成分の指定(ラボフレーム).
- object_type.rigid.Rigid_spec[].Rigid_velocity.z...剛体の速度のz成分の指定(ラボフレーム).
- object_type.rigid.Rigid_spec[].Rigid_omega.x...剛体の角速度のx成分の指定(ラボフレーム).
- object_type.rigid.Rigid_spec[].Rigid_omega.y...剛体の角速度のy成分の指定(ラボフレーム).
- object_type.rigid.Rigid_spec[].Rigid_omega.z...剛体の角速度のz成分の指定(ラボフレーム).

2.2.3　共通のシミュレーション設定

粒子半径，界面厚さは以下に情報を入れる.

- A_XI...界面の厚さξ [*10].
- A...粒子半径.

gravity以下で重力に関する情報を入れる.

- gravity.G...重力加速度.
- gravity.G_direction...重力を加える方向を選ぶ.

粒子間力のタイプを決める.

- EPSILON...Lennard-Jonesポテンシャルのエネルギーの単位を決める.
- LJ_powers...粒子間ポテンシャルを，`12:6/24:12/36:18` (Lennard-Jonesポテンシャル)，`DLVO` (DLVOポテンシャル)，あるいは，`electro_osmotic_flow` (クインケローラーのみ使用)から選ぶ.

mesh以下でシミュレーションサイズを決める.

- mesh.NPX...x方向のサイズは$L_x = 2^{\texttt{NPX}}$.
- mesh.NPY...y方向のサイズは$L_y = 2^{\texttt{NPY}}$.
- mesh.NPZ...z方向のサイズは$L_z = 2^{\texttt{NPZ}}$.

time_increment以下で時間刻みを決める.

- time_increment...`auto`を選べば時間刻みの上限である$T_{step} = \rho/\eta k_{max}^2$を時間刻みとする．ここで$k_{max}$は格子幅$\Delta$で決まる最大波数である． `manual`を選べば手動で値を設定できる.
- time_increment.auto.factor...入力する値`factor`を使って時間刻みは$\Delta t = T_{step} \times$ `factor`と決まる.
- time_increment.manual.delta_t...手動でΔtの値を設定する.

2.2.4　選択できる機能の設定

switch以下でシミュレーションの各種条件を決める.

- switch.ROTATION...粒子の回転運動の運動方程式を解く場合は`ON`を選ぶ.
- swithch.LJ_truncate...粒子間に働くLennard-Jonesポテンシャルによる力について，引力項を含む通常の形`OFF`，引力項を含まない斥力のみ`ON`，まったく力を加えない`NONE`から選ぶ.
- switch.INIT_distribution...粒子の初期配置を， `uniform_random`(ランダム)，`random_walk`(正方格子上か

[*10] 電解質溶液や二成分相分離流体のシミュレーションの場合，GL自由エネルギーFでϕの勾配を計算する必要があるため，少なくとも$\xi \geq 2$を選ぶことが望ましい.

らランダムにずれている)， FCC(FCC格子上)，BCC(BCC格子上)，user_specify(座標と速度はユーザによって指定される)から選ぶ [*11].

- switch.INIT_distribution.random_walk.iteration...粒子の初期配置をrandom_walkに設定したときの試行回数を設定する.
- switch.INIT_orientation...粒子の初期配向を，user_specify(座標と速度はユーザによって指定される)，random(ランダム)，space_align(1方向に配向)から選ぶ [*12].
- switch.SLIP_tol...自己推進粒子界面における接戦方向のslip速度を導入する際の，流体流れ場の反復計算の収束判定基準.
- switch.SLIP_iter...自己推進粒子界面における接戦方向のslip速度を導入する際の，流体流れ場の反復計算の最大反復回数.
- switch.FIX_CELL.x...x方向の全速度の直流成分を0とする場合ONにする.
- switch.FIX_CELL.y...y方向の全速度の直流成分を0とする場合ONにする.
- switch.FIX_CELL.z...z方向の全速度の直流成分を0とする場合ONにする.
- switch.pin.type...粒子を固定する場合YESにする.
- switch.pin.YES.pin[]...並進運動させない粒子番号の指定.
- switch.pin.YES.pin_rot[]...回転運動させない粒子番号の指定.
- switch.free_rigid.type...剛体の並進・回転の自由度を個別に設定する．このオプションは，6つの自由度のすべてを指定された並進/回転速度に固定する object_type.rigid.Rigid_spec[].Rigid_motion オプションと同時に使用することを意図している． 6つの自由度を剛体粒子ごとに独立に設定するには，このオプションを YES に設定し，後述の vel.x|y|z と omega.x|y|z を使用して該当する自由度を選択的に解放(YES)，または固定 (NO)する．各速度成分の固定値は，object_type.rigid.Rigid_spec[] オプションの Rigid_velocityとRigid_omega 変数で指定された値である．例えば，vel.y と omega.z を固定した剛体粒子を指定する場合は，以下のように free_rigid オプションを指定すればよい．つまりここではどの自由度を自由に設定するかを示せばよい.

  vel.x=YES, vel.y=NO, vel.z=YES, omega.x=YES, omega.y=YES, omega.z=NO
- switch.free_rigid.YES.DOF[].spec_id...設定する剛体番号の指定.
- switch.free_rigid.YES.DOF[].vel.x...剛体のx方向の並進運動の自由度の設定．YESなら自由度を持つが，NOなら固定される.
- switch.free_rigid.YES.DOF[].vel.y...剛体のy方向の並進運動の自由度の設定．YESなら自由度を持つが，NOなら固定される.
- switch.free_rigid.YES.DOF[].vel.z...剛体のz方向の並進運動の自由度の設定．YESなら自由度を持つが，NOなら固定される.
- switch.free_rigid.YES.DOF[].omega.x...剛体のx方向の回転運動の自由度の設定．YESなら自由度を持つが，NOなら固定される.
- switch.free_rigid.YES.DOF[].omega.y...剛体のy方向の回転運動の自由度の設定．YESなら自由度を持つが，NOなら固定される.
- switch.free_rigid.YES.DOF[].omega.z...剛体のz方向の回転運動の自由度の設定．YESなら自由度を持つが，NOなら固定される.
- ns_solver.OBL_INT...せん断流下のシミュレーションにおける座標系変換時の近似関数をlinear(線形近似)とspline(スプライン近似)から選ぶ.

[*11] user_specifyを選んだ場合，初期の粒子位置と速度をそれぞれuser_specify.Particles[].Rとuser_specify.Particles[].vに設定する．入力するリストの数がParticle_numberで指定した数より小さければ， GOURMET上でEdit->Add an array Elementとしてuser_specify.Particles[]を増やすか直接UDFファイルを編集する.

[*12] user_specifyを選んだ場合，粒子の初期配向をuser_specify.Particles[].qに設定する.

- switch.wall.type...FLATを選ぶと，平面壁を設定できる．
- switch.wall.FLAT.axis...平面壁の法線方向をX，Y，Zで指定する．
- switch.wall.FLAT.DH...平面壁の厚みを格子点の数で指定する．
- switch.wall.FLAT.LJ_Params...平面壁に働くポテンシャルの設定．AUTOを選ぶと粒子間ポテンシャルと同じポテンシャルが設定される．MANUALを選ぶと，switch.wall.FLAT.MANUALで個別に設定できる．
- switch.wall.FLAT.MANUAL.truncate...平面壁での引力の有無を設定する．ONにすると斥力のみが働くポテンシャルとなり，OFFにすると引力も働くポテンシャルとなる．
- switch.wall.FLAT.MANUAL.powers...Lennard-Jonesポテンシャルのべき指数を12:6/24:12/36:18から選ぶ．
- switch.wall.FLAT.MANUAL.EPSILON...平面壁のLennard-Jonesポテンシャルの強さを決める．
- switch.quincke.type...クインケローラーのシミュレーションをする場合，ONに設定する．
- switch.quincke.ON.e_dir...外部電場を印加する方向をX，Y，Zで指定する．
- switch.quincke.ON.w_dir...クインケ効果による回転の角速度ベクトルの方向をX，Y，Zで指定する．
- switch.quincke.ON.torque_amp...回転トルクの大きさ．
- switch.multipole.type...Ewald法によるシミュレーションをする場合，ONに設定する．
- switch.multipole.ON.Dipole...クインケ粒子の双極子を扱う場合，ONに設定する．
- switch.multipole.ON.Dipole.ON.magnitude...双極子モーメントの大きさ．
- switch.multipole.ON.Dipole.ON.type...双極子の種類を，FIXED(固定)とQUINCKE(クインケ回転)から選ぶ．
- switch.multipole.ON.Dipole.ON.FIXED.dir...双極子を固定する場合，その方向をX，Y，Zで指定する．
- switch.multipole.ON.EwaldParams.alpha...Ewald法における遮蔽パラメータ．
- switch.multipole.ON.EwaldParams.delta...Ewald法における$k_{\max}$を決定するための収束判定基準．
- switch.multipole.ON.EwaldParams.converge...Ewald法における収束パラメータ．
- switch.multipole.ON.EwaldParams.epsilon...Ewald法における境界での誘電率．

2.2.5　データ出力設定

output以下にデータ出力の情報を設定する．

- output.GTS...データ出力のインターバルのステップ数．
- output.Num_snap...データ出力の回数，つまり全ステップ数はGTS × Num_snapで決まる．
- output.AVS...AVS形式のデータを出力する場合はONを選ぶ．https://ja.overleaf.com/project/5fe7389e29fad9c51e06569e
- output.AVS.ON.Out_dir...AVS形式のデータを出力するディレクトリを指定する [*13]．
- output.AVS.ON.Out_name...AVS形式の出力データのファイル名の接頭語を指定する．
- output.AVS.ON.FileType...AVSデータファイルのフォーマットをBinary，ASCII，EXTENDEDから選ぶ．
- output.ON.EXTENDED.Driver.Format...AVS拡張データフォーマットを選ぶ．現行のプログラムでは，HDF5のみ選択できる．
- output.ON.EXTENDED.Print_field.Crop...YESを選ぶと，フィールドデータを間引いて出力する．
- output.ON.EXTENDED.Print_field.YES.Slab_x.start...x方向において，出力する最初の格子点番号を指定する．
- output.ON.EXTENDED.Print_field.YES.Slab_x.count...x方向において，出力する格子点数を指定する．
- output.ON.EXTENDED.Print_field.YES.Slab_x.stride...x方向において，出力する格子点間隔を指定する．

[*13] たとえばdataと指定するなら，事前に./dataおよび./data/avs/のディレクトリを作成しておく必要がある． AVSのfieldファイルは./data/にdata.fldというファイル名で出力される．データファイルは./data/avs/にdata_*.datというファイル名で出力される． *にはステップ数が入る．

- output.ON.EXTENDED.Print_field.YES.Slab_y.start...y方向において，出力する最初の格子点番号を指定する．
- output.ON.EXTENDED.Print_field.YES.Slab_y.count...y方向において，出力する格子点数を指定する．
- output.ON.EXTENDED.Print_field.YES.Slab_y.stride...y方向において，出力する格子点間隔を指定する．
- output.ON.EXTENDED.Print_field.YES.Slab_z.start...z方向において，出力する最初の格子点番号を指定する．
- output.ON.EXTENDED.Print_field.YES.Slab_z.count...z方向において，出力する格子点数を指定する．
- output.ON.EXTENDED.Print_field.YES.Slab_z.stride...z方向において，出力する格子点間隔を指定する．
- output.ON.EXTENDED.Print_field.Vel...YESを選ぶと，速度場を出力する．
- output.ON.EXTENDED.Print_field.Phi...YESを選ぶと，SP関数ϕを出力する．
- output.ON.EXTENDED.Print_field.Charge...YESを選ぶと，電荷密度分布を出力する(二成分相分離流体のシミュレーションでは非対応).
- output.ON.EXTENDED.Print_field.Pressure...YESを選ぶと，圧力場を出力する(未実装).
- output.ON.EXTENDED.Print_field.Tau...YESを選ぶと，応力テンソルを出力する．
- output.UDF...UDFを出力する場合はONを選ぶ [*14].

2.2.6　リスタートの設定

resume以下で中断した計算を再計算するリスタートUDFについて指定する．

- resume.caclucation...NEWを選ぶと新規の計算ができる． CONTINUE/CONTINUE_FDM/CONTINUE_FDM_PHASE_SEPARATIONはいずれも前回計算終了したときの情報を読み込んで計算を再開するための設定であり，constitutive_eqの設定によって使い分ける必要がある． CONTINUEは，スペクトル法を用いるシミュレーション [*15] を再開する設定である． CONTINUE_FDMは，差分法による一成分流体のシミュレーション [*16] を再開する設定である． CONTINUE_FDM_PHASE_SEPARATIONは，差分法による二成分流体のシミュレーション [*17] を再開する設定である [*18].

2.2.7　GOURMETでの表示設定

Unit_Parameter以下は，GOURMETでの表示に関する設定である．シミュレーション結果への影響はない．

- Unit_Parameter.Name...UDFファイルの名前を設定できる．
- Unit_Parameter.Comment...UDFファイルのコメントを設定できる．
- Unit_Parameter.Temperature...GOURMETでシミュレーションパラメータを実次元表示するための基準となる単位温度を指定する．
- Unit_Parameter.Length...GOURMETでシミュレーションパラメータを実次元表示するための基準となる単位長さを指定する．
- Unit_Parameter.rho...GOURMETでシミュレーションパラメータを実次元表示するための基準となる単位密度を指定する．

[*14] 出力UDFのParticles[]以下には各粒子の座標と速度がoutput.Num_snapで指定した数のレコードデータに保存される．
[*15] constitutive_eqにおいて，Navier_Stokes/Shear_Navier_Stokes/Shear_Navier_Stokes_Lees_Edwards/Electrolyteを選択している場合である．
[*16] Navier_Stokes_FDM/Shear_Navier_Stokes_Lees_Edwards_FDMを選択している場合である．
[*17] Navier_Stokes_Cahn_Hilliard_FDM/Shear_NS_LE_CH_FDMを選択している場合である．
[*18] リスタートUDFファイルのresume.CONTINUE.Saved_Data以下に結果が保存される．そのため，前回終了時にできるリスタートUDFを入力UDFとして計算を再開することができる．

2.3　出力UDFと再開UDFについて

出力UDF（output.udf）にはシミュレーション途中のデータが保存される．シミュレーションのステップ数とデータの保存頻度は，input.udfで以下の変数を用いて設定する．

- output.GTS... データ出力のインターバルのステップ数.
- output.Num_snap... データ出力の回数. つまり全ステップ数=GTS×Num_snap

再開UDF（restart.udf）にはシミュレーション終了時の全データが保存される． restart.udfを使って計算を再開させたい場合は以下の手順でおこなえばよい.

1. restart.udfをinput2.udfにリネームする.
2. Gourmetでinput2.udfを開き, resume.CalculationでCONTINUE/CONTINUE FDM/CONTINUE FDM PHASE SEPARATIONを適切に選ぶ.
3. さらにoutput.Num_stepを増やしトータルで計算させたいステップ数（=前回計算したステップ数+今回計算したいステップ数）を指定して保存する.
4. 各種パラメータの値や粒子数なども変更してリスタートすることもできる．粒子を追加/変更する場合は，object_typeブロックを適切に設定し，resumeブロックのParticlesに追加/変更を適用すること. switchブロックのINIT_distribution/INIT_orientationはリスタートでは無視される.
5. 再び
   % kapsel -Iinput2.udf -Ooutput.udf -Ddefine.udf -Rrestart.udf
   とすればinput2.udfからデータを読み込み計算を再開する.
6. AVSデータ, output.udfともに前回計算したデータをそのままに, 再開した計算の出力データを追加することができる.
7. restart.udfには再び計算終了時におけるデータが格納されるので, この手順を繰り返せばさらに計算を続行することが可能である.

第3章

Newton流体に分散した粒子のシミュレーション

微粒子が液体(溶媒)中に分散した系は微粒子懸濁液と呼ばれ，食品・塗料・顔料・化粧品・スラリーなど我々の日常生活に数多く存在している．このような微粒子懸濁液を理解する上で，微粒子および溶媒の両者の運動を同時に把握することは非常に重要である．

3.1　理論的背景と基礎方程式

KAPSELでは，さまざまな粒子分散系のシミュレーションを行うために，Smoothed Profiled (SP) 法と呼ばれる直接数値計算法を採用している[5, 6]．SP法とは，流体と固体粒子の両方を考慮した運動方程式を同時に解くという手法であり．Newton流体だけでなく，任意のReynolds数における複雑流体を溶媒とした系に対しても適用が可能である．

3.1.1　粒子分散系における基礎方程式

SP法を用いると，連続体中に分散した固体粒子の運動を解くことが可能となる[6]． SP法は原理上，任意の形状の固体粒子に適用可能である．また，単純液体だけでなく，多相複雑流体のような複雑な系に対しても，それに応じた構成方程式を自由に用いることで，直接計算することができる．

粒子と流れの運動がSP法とどのように結びつくかを示すために，本節では，Newton流体中を自由に動く円盤状の形をした剛体（固体）の運動を考える[7]．固体は図3.1に示すように球状ビーズの集合体として表現される．以後，このような固体を単に粒子と呼ぶ． 非圧縮性のNewton流体は，連続の式とNavier–Stokes方程式で支配される．

$$\nabla \cdot \boldsymbol{u}_f = 0, \tag{3.1}$$

$$\left(\partial_t + \boldsymbol{u}_f \cdot \nabla\right)\boldsymbol{u}_f = \rho_f^{-1}\nabla \cdot \boldsymbol{\sigma}, \tag{3.2}$$

$$\boldsymbol{\sigma} = -p\boldsymbol{I} + \eta\left[\nabla \boldsymbol{u}_f + \left(\nabla \boldsymbol{u}_f\right)^t\right] \tag{3.3}$$

ここで，$\boldsymbol{u}_f, \rho_f, \eta, \boldsymbol{\sigma}$ はそれぞれ流体の速度，密度，粘度，および応力（テンソル）場を表し，p は圧力，$\boldsymbol{I}$ は単位テンソル，$(\cdots)^t$ は転置を表す．特に明記しない限り，以後，粒子と流体の密度は等しい($\rho_f = \rho_p = \rho$)と仮定する．また，固体–流体界面では滑りなしの境界条件を適用する(すなわち，固体表面では $u_f = u_p$ が成立する)． u_p は固体表面での粒子速度である． I 番目の粒子($I = 1, \cdots, N$)の質量を M_I，慣性モーメントを I_I としたとき，N 個の剛体球から成る固体粒子の運動は，Newton–Euler方程式

$$\dot{\boldsymbol{R}}_I = \boldsymbol{V}_I, \quad \dot{\boldsymbol{Q}}_I = \mathrm{skew}(\boldsymbol{\Omega}_I) \cdot \boldsymbol{Q}_I, \tag{3.4}$$

$$M_I\dot{\boldsymbol{V}}_I = \boldsymbol{F}_I^H + \boldsymbol{F}_I^C + \boldsymbol{F}_I^E + \boldsymbol{G}_I^V, \tag{3.5}$$

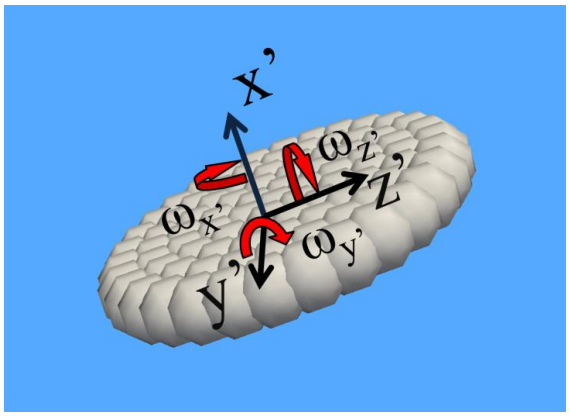


図3.1: 3次元空間を運動する円盤状の形をした固体の概略．運動は3つの並進自由度と3つの回転自由度により特徴づけられる．

$$\dot{\boldsymbol{J}}_I = \boldsymbol{N}_I^H + \boldsymbol{N}_I^C + \boldsymbol{N}_I^E + \boldsymbol{G}_I^\Omega, \tag{3.6}$$

で記述される[8]．ここで，$\boldsymbol{R}_I, \boldsymbol{V}_I$ はそれぞれ重心座標と速度，$\boldsymbol{Q}_I, \boldsymbol{\Omega}_I$ は対応する方向行列と角速度，$\boldsymbol{J}_I = \boldsymbol{I}_I \cdot \boldsymbol{\Omega}_I$ は角運動量である．方向行列により，固定直交座標系(実験座標系) $\{\boldsymbol{e}_i\}$ $(i = 1,2,3)$ は，粒子とともに回転する時間依存した座標系 $\{\tilde{\boldsymbol{e}}_i\}$ $(i = 1,2,3)$ と，$\tilde{\boldsymbol{e}}_i = \sum_{j=1}^{3} \boldsymbol{e}_j Q_{ji}$ のように関係づけられる．回転行列の時間発展は(歪対称)角速度行列により決定され，角速度行列は角速度ベクトル $[\mathrm{skew}(\boldsymbol{\Omega})]_{ij} = -\sum_{k=1}^{3} \epsilon_{ijk}\Omega^k$ から導かれる(ϵ_{ijk} はLevi–Civita記号).

$$\mathrm{skew}(\boldsymbol{\Omega}) = \begin{pmatrix} 0 & -\Omega^z & \Omega^y \\ \Omega^z & 0 & -\Omega^x \\ -\Omega^y & \Omega^x & 0 \end{pmatrix}. \tag{3.7}$$

粒子には，周りからの力 $\boldsymbol{F}$ およびトルク $\boldsymbol{N}$ が作用する．ここで力(トルク)は，溶媒からの寄与，直接的な粒子間相互作用からの寄与 $\boldsymbol{F}^C$，重力など外場からの寄与 $\boldsymbol{F}^E$ の3つに分けられる．溶媒からの寄与は，流体力学的相互作用による寄与 $\boldsymbol{F}^H$ と熱揺らぎによる寄与 $\boldsymbol{F}^V$ から構成される．注意点として，力・トルクの各寄与についての正確な表式は対象となる系に依存し，それぞれ個別に議論される．また，$\boldsymbol{G}^V, \boldsymbol{G}^\Omega$ は，熱揺らぎにより粒子に働くランダム力およびトルクであり，白色雑音として導入されている．すなわち，$\langle \boldsymbol{G}_I^V \rangle = \langle \boldsymbol{G}_I^\Omega \rangle = 0$ かつ

$$\langle \boldsymbol{G}_I^V(t) \cdot \boldsymbol{G}_J^V(0) \rangle = 3k_B T \alpha^V \delta(t)\delta_{IJ}, \tag{3.8}$$
$$\langle \boldsymbol{G}_I^\Omega(t) \cdot \boldsymbol{G}_J^\Omega(0) \rangle = 3k_B T \alpha^\Omega \delta(t)\delta_{IJ}, \tag{3.9}$$

である．ここで $\langle\rangle$ は統計平均を表しており，α^V および α^Ω は，粒子温度 T を正確に設定値に一致させるための調整パラメータである．粒子温度 T は，溶媒中で揺らいでいる 1 粒子のシミュレーションを予め行い，並進拡散係数 D^V および回転拡散係数 D^Ω が理論値と一致するように試行錯誤で決定する．

ここからは，上記の方程式で表される系をSP法を用いて解く方法について説明する． SP法では，溶媒部分と粒子内部の固体部分の両方を含む全空間領域($\boldsymbol{x}$)で速度場

$$\boldsymbol{u}(\boldsymbol{x},t) = (1-\phi)\,\boldsymbol{u}_f(\boldsymbol{x},t) + \phi \boldsymbol{u}_p(\boldsymbol{x},t), \tag{3.10}$$

を定義する．ここで

$$\phi \boldsymbol{u}_p(\boldsymbol{x},t) = \sum_{I=1}^{N} \phi_I \left[\boldsymbol{V}_I + \boldsymbol{\Omega}_I \times \boldsymbol{r}_I\right], \tag{3.11}$$

は固体粒子の剛体運動を速度場で表現したものであり， $\boldsymbol{r}_I = \boldsymbol{x} - \boldsymbol{R}_I$ は粒子 I の重心からの距離を，$\phi_I = \phi_I(\boldsymbol{x}; \boldsymbol{R}_I, \boldsymbol{Q}_I) \in [0,1]$ は粒子内部の固体部分 $(\phi \simeq 1)$ と外部の流体部分 $(\simeq 0)$ を区別するPhase field関数である．

SP法では，本来のシャープな固液界面を有限の厚みを持ったぼやけた界面領域に置き換えるために，界面厚み ξ を ϕ （Phase Field）関数に導入している．これを本稿ではSmoothed Profile (SP)関数と呼ぶ．このような界面に有限の幅を持たせる方法は，数値計算をする上で2つの大きな利点がある． 1つ目は限られた数の格子点でも界面の位置を追跡できるということ，2つ目は疎水性や親水性など固体–流体間の相互作用を目的に応じて正しく設定できるということである．

3.1.2　SP法の導入

流体部分と固体（粒子）部分からなる全系の速度場$\boldsymbol{u}(\boldsymbol{r})$は，連続の式とNavier–Stokes方程式により，

$$\nabla \cdot \boldsymbol{u} = 0, \tag{3.12}$$

$$(\partial_t + \boldsymbol{u} \cdot \nabla)\, \boldsymbol{u} = \rho^{-1} \nabla \cdot \boldsymbol{\sigma} + \phi \boldsymbol{f}_p, \tag{3.13}$$

と表される．ここで，$\phi \boldsymbol{f}_p$ は粒子領域に剛体性の制約を課すための拘束力として作用する．また局所応力テンソルは，流体部分の速度 $\boldsymbol{u}_f$ を用いた式(3.3)で定義されるが，SP法では固体粒子内部も含めた全系の速度 $\boldsymbol{u}$ を用いて定義される． $\phi \boldsymbol{f}_p, \boldsymbol{F}_I^H, \boldsymbol{N}_I^H$ の各項については，以下の式で示すように t に関する離散化($n \to n+1$)を経て定義する． $\boldsymbol{u}^n$ を n 番目の時間ステップ $t_n = nh$ (h は時間刻み幅)での速度場として表す．

まず始めに，式(3.13)と(3.4)を項 $\phi \boldsymbol{f}_p$ がないものとして積分($t_n \to t_n + h$)を行うと，粒子の位置と方向は以下のように変化する．

$$\boldsymbol{u}^* = \boldsymbol{u}^n + \int_{t_n}^{t_n+h} \mathrm{d}s \nabla \cdot \left[\rho^{-1} \boldsymbol{\sigma} - \boldsymbol{u}\boldsymbol{u} \right], \tag{3.14}$$

$$\boldsymbol{R}_I^{n+1} = \boldsymbol{R}_I^n + \int_{t_n}^{t_n+h} \mathrm{d}s \boldsymbol{V}_I, \tag{3.15}$$

$$\boldsymbol{Q}_I^{n+1} = \boldsymbol{Q}_I^n + \int_{t_n}^{t_n+h} \mathrm{d}s\, \mathrm{skew}(\boldsymbol{\Omega}) \cdot \boldsymbol{Q}_I. \tag{3.16}$$

この時点で，たとえ粒子速度が変化していなくても，粒子の速度場の更新を行う必要がある． $\phi \boldsymbol{f}_p$がない場合での速度 $\boldsymbol{u}^*$ に対して運動量保存を仮定すると，粒子に加わる流体力学的な力およびトルクは，流体が粒子から受け取る運動量とバランスして

$$\left[\int_{t_n}^{t_n+h} \mathrm{d}s \boldsymbol{F}_I^H \right] = \int \mathrm{d}\boldsymbol{x} \rho \phi_I^{n+1} \left(\boldsymbol{u}^* - \boldsymbol{u}_p^n \right), \tag{3.17}$$

$$\left[\int_{t_n}^{t_n+h} \mathrm{d}s \boldsymbol{N}_I^H \right] = \int \mathrm{d}\boldsymbol{x} \left[\boldsymbol{r}_I^{n+1} \times \rho \phi_I^{n+1} \left(\boldsymbol{u}^* - \boldsymbol{u}_p^n \right) \right]. \tag{3.18}$$

と表される．その結果，粒子速度は $\boldsymbol{V}_I^{n+1}$ と $\boldsymbol{\Omega}_I^{n+1}$ へと更新される．この更新により，式(3.11)から最終的な粒子速度場 $\phi^{n+1}\boldsymbol{u}_p^{n+1}$ ，すなわち ϕ_I^{n+1} が得られる．

最後に，粒子の剛体性を保持するために導入した$\phi \boldsymbol{f}_p$を考慮することにより，$n+1$ステップの全系の速度が求まる．

$$\boldsymbol{u}^{n+1} = \boldsymbol{u}^* + \left[\int_{t_n}^{t_n+h} \mathrm{d}s \phi \boldsymbol{f}_p \right], \tag{3.19}$$

$$\left[\int_{t_n}^{t_n+h} \mathrm{d}s \phi \boldsymbol{f}_p \right] = \phi^{n+1} \left(\boldsymbol{u}_p^{n+1} - \boldsymbol{u}^* \right) - \frac{h}{\rho} \nabla p_p, \tag{3.20}$$

ここで，剛体性による圧力は連続の式 $\nabla \cdot \boldsymbol{u}^{n+1} = 0$ から得られる．全領域にわたって粘性応力が適用されるため，関連のある粒子表面での滑りなし条件は有限界面幅ξの範囲内で満たされる．

3.1.3　Smoothed Profiled (SP)関数

本節では，任意の形状を持つ粒子に対して，どのようにSP関数が作成されるかを説明する．SP関数は，有限の界面幅 ξ の領域内で，粒子内部領域 $\phi = 1$ と流体領域 $\phi = 0$ が滑らかに分離されるような関数である．例えば球状粒子の場合，いくつかのSP関数の解析的表式が提案されている[5]．その一例として，

$$\phi_I = \frac{1}{2}\left[\tanh\left(\frac{a - |\boldsymbol{r}_I|}{\xi}\right) + 1\right], \tag{3.21}$$

があり[9, 10]，ここで a は球径を表す．界面幅 ξ は自由に指定できるパラメータであるが，数値計算の安定性のため，一般的に格子間隔の1倍または2倍に設定することが望ましい．界面幅 ξ を用いて粒子領域と流体領域がどのように分離されるかを例示するために，図3.2に粒子断面の概略を示す．

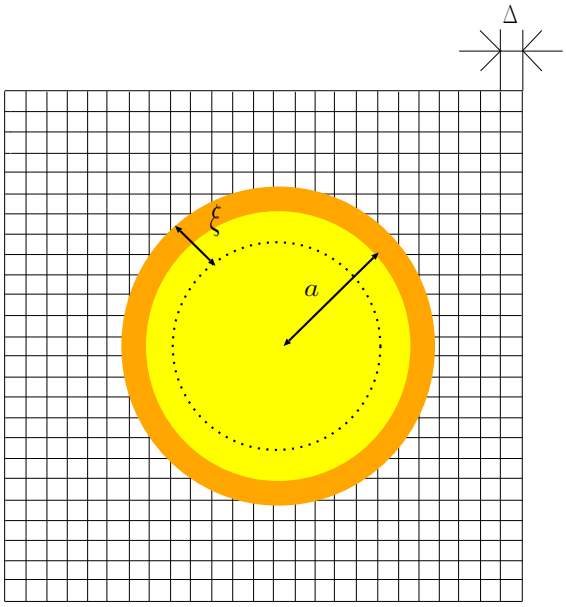


図3.2: 球状SP関数の断面図の概略．Δ は格子間隔，a は粒子径，xi は界面幅．ここで，粒子表面の有限体積 $\sim \pi a^2 \xi$. Reproduced from Phys. Rev. E 71, 036707[5], Copyright 2005, with permission from the American Physical Society.

非球状粒子については，その複雑な形状を球状ビーズの集合体により表現する．構成要素となるビーズ同士が重なり合うことは可能であるため，任意の粒子を球状ビーズの集合によって表現できる．I 番目の粒子が n_I 個の球状ビーズで構成されるとおくと，そのようなビーズの集合体に対するSP関数は，

$$\Phi_I(\boldsymbol{x}, t) = \sum_{i=1}^{n_I} \phi_{I,i}(\boldsymbol{x}, t), \tag{3.22}$$

$$\phi_I(\boldsymbol{x}, t) = \frac{\Phi_I}{\max(\Phi_I, 1)}, \tag{3.23}$$

と表される．ここで $\phi_{I,i}$ は I 番目粒子に属する i 番目の球状ビーズのSP関数である．

複雑な形状の粒子においても，SP関数が定義されると，粒子に対する流体力学的な力積およびそれに対応した流体に対する運動量の力積は式(3.17), (3.18), (3.20)で表される．

流体の慣性は上述のSPのアルゴリズムで解かれるため，付加的な質量効果は式(3.17)中の流体力学的な力積で説明される．その慣性効果を明らかにするため，式(3.17)に示した流体力学的な力積の被積分関数は

$$\rho\phi_I^{n+1}\left(\boldsymbol{u}^* - \boldsymbol{u}_p^n\right) = -\rho\phi_I^{n+1}\left(\boldsymbol{u}_p^{n+1} - \boldsymbol{u}^*\right) + \rho\phi_I^{n+1}\left(\boldsymbol{u}_p^{n+1} - \boldsymbol{u}_p^n\right)$$

$$= -\rho \int_{t_n}^{t_n+h} \mathrm{d}s\phi_I \boldsymbol{f}_p - h\nabla p_p + \rho\phi_I^{n+1}\left[\boldsymbol{V}_i^{n+1} - \boldsymbol{V}_i^n + \left(\boldsymbol{\Omega}_I^{n+1} - \boldsymbol{\Omega}_I^n\right) \times \boldsymbol{r}_I^{n+1}\right], \tag{3.24}$$

のように分解できる．$\int \mathrm{d}\boldsymbol{x}\phi_I \boldsymbol{r} = 0$ を用いてこの式を空間積分すると，

$$\boldsymbol{F}_I^H = -\int \mathrm{d}\boldsymbol{x}\rho\phi_I \boldsymbol{f}_p - \int \mathrm{d}\boldsymbol{x}\phi_I \nabla p_p + M_I \frac{\rho}{\rho_{p,I}} \dot{\boldsymbol{V}}_I, \tag{3.25}$$

ここで，$\rho_{p,I}$ は I 番目粒子の質量密度である．この式より，式(3.17)中の流体力学的な力は，剛体制約の反作用と付加的な質量効果から構成されることがわかる．

SP法の計算精度は，時間の刻み幅 h と界面幅 ξ の両方に依存する． Luoら[11]の報告によれば，h を小さくすることで単調に計算精度が向上するわけでなく， h と $\xi^2\rho/\eta$（運動量が粘性拡散により距離 ξ 移動するのに必要な時間）が同じオーダーであるときに最適値となる．この関係は，剛体制約の力積(式(3.20))から生じる粘性ストークス層を考慮するためには，それに見合った界面幅が必要であるということを反映している．

3.1.4　粒子温度の決定手順

1. ある値にα^Vとα^Ωをそれぞれ固定して 1 粒子系の熱平衡計算を行う．
2. シミュレーションから 1 粒子の並進拡散係数 D_{sim}^Vや回転拡散係数D_{sim}^Ωを平均二乗変位等を通してそれぞれ求める．
3. 得られた拡散係数$D_{sim}^V, D_{sim}^\Omega$を無限希釈系の 1 粒子系の拡散係数の解析解(並進運動に対しては，$D_0^V = k_B T^V/6\pi\eta a$および，回転運動に対しては，$D_0^\Omega = k_B T^\Omega/8\pi\eta a^3$)と比較することにより，それぞれの並進および回転運動に対する温度T^V, T^Ωを決定することができる．
4. $T^\Omega \neq T^V$の場合，$T^\Omega = T^V(= T)$を満足するようにα^Vまたはα^Ωを調整する．

このアプローチによる温度決定方法の詳細については文献[12–14]を参照されたい．

3.2　入力UDF について

3.2.1　流体の設定

constitutive_eqとしてNavier_Stokesを選ぶと Newton 流体中のコロイド粒子の運動のシミュレーションが行える．以下, 入力U¡DFファイルで指定する変数について解説する．

constitutive_eq.Navier_Stokes以下で溶媒の情報を指定する．

- constitutive_eq.Shear_Navier_Stokes.DX…長さの単位量である格子幅Δ.
- constitutive_eq.Shear_Navier_Stokes.RHO…溶媒の密度.
- constitutive_eq.Shear_Navier_Stokes.ETA…溶媒の粘度.
- constitutive_eq.Shear_Navier_Stokes.kBT…粒子温度.
- constitutive_eq.Shear_Navier_Stokes.alpha_v…粒子の並進温度に対する補正項[*1].
- constitutive_eq.Shear_Navier_Stokes.alpha_o…粒子の回転温度に対する補正項.

3.2.2　オブジェクト(粒子)の設定

object_type.typeでは，spherical_particle(球状粒子)，chain(フレキシブル鎖)，rigid(剛体)を選択できる．

3.2.3　空間単位と時間単位

長さの単位としてを格子幅Δを採用している. 時間の単位τ_0ついて流体の密度ρと粘性率ηと格子幅で決まる$\tau_0 = \rho\Delta^2/\eta$を採用している．

- Navier–Stokes方程式で$\rho = \eta = \Delta = 1$となるような単位系を採用している．
- 仮に入力udfファイルでRHO= A, ETA= B, DX= Cと入力した場合, 最大波数k_{max}はCを用いて与えられ, 時間刻みの上限は運動量の拡散時間より$T_{step} = (A/B)/k_{max}^2$ と与えられる. 時間刻みΔtは$T_{step} \times$ factorで調整する．
- 現実との対応を考察する．考察したい空間スケールとして，グリッドサイズを1 μmの長さを考える．また溶媒として水($\eta = 1 \times 10^{-3}$ Pa s，$\rho = 1 \times 10^3$ kg m^{-3})を対象とした場合，時間の単位は$\tau_0 = 1 \times 10^{-6}$ sとなる．

[*1] 微粒子の熱拡散を導入する場合は，温度を入力する．熱平衡での粒子の拡散運動から温度は見積もられるが，界面の厚さ等の数値的な理由のために正しい温度を返さない可能性がある．この場合，もし測定された温度が設定値の1/n倍であると見積もられた場合，このファクターをnと設定することで既定の温度を実現することができる．回転に対しても同様である．半径5または4，かつ$\xi = 2$に対しては，alpha_v=alpha_o=1でほぼ設定温度を再現できているようである．

3.3　計算事例

3.3.1　粒子の沈降

入力UDFとして，Examples/02/フォルダの gravity.udfおよびgravity100000.udfを用いればよい．gravity.udfおよびgravity100000.udfでは，それぞれ3204粒子，100000粒子の沈降を計算することができる．具体的には./avs_g1/と./avs_g1/avs/ が作成されていることを確認した上で以下を実行する[*2]．

```
$ ../../kapsel -Igravity.udf -Ooutput.udf -Ddefine.udf -Rrestart.udf
$ ../../kapsel -Igravity100000.udf -Ooutput.udf -Ddefine.udf -Rrestart.udf
```

この例(Fig. 3.3)ではメッシュサイズ256 × 256 × 512で計算している． gravity.udfでは，粒子数$N_p = 3204$，粒子直径$D = 10$，界面厚さ$\xi = 2$であり，このときの粒子体積分率は$\varphi = 0.05$である．図3.3には，重力下で沈降している3204粒子の懸濁液のスナップショットを示している．

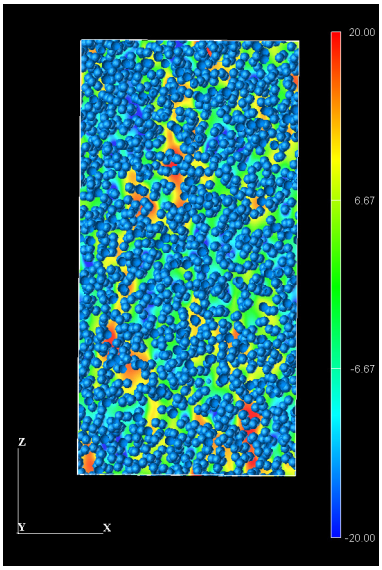


図3.3: 重力下での3204粒子の懸濁液の沈降．重力は，z方向下向きに働いている．流体の色は，z方向の流体場の速度を示している．

一方，gravity100000.udfでは，粒子数$N_p = 100000$，粒子直径$D = 4$，界面厚さ$\xi = 2$であり，このときの粒子体積分率は$\varphi = 0.10$である．図3.4には，重力下で沈降している100000粒子の懸濁液のスナップショットを示している．

3.3.2　粒子の拡散

入力UDFとして，Examples/03/フォルダの repulsive.udfを用いればよい． repulsive.udfでは，粒子間に反発力のみの相互作用をもつBrown粒子系の拡散現象を計算することができる．

```
$ ../../kapsel -Irepulsive.udf -Ooutput.udf -Ddefine.udf -Rrestart.udf
```

[*2] AVSデータが必要でない場合は， AVSのスイッチをOFFに切り替え，./avs_g1/と./avs_g1/avs/を作成する必要はない．これはその他のサンプルファイルに対しても同様である．

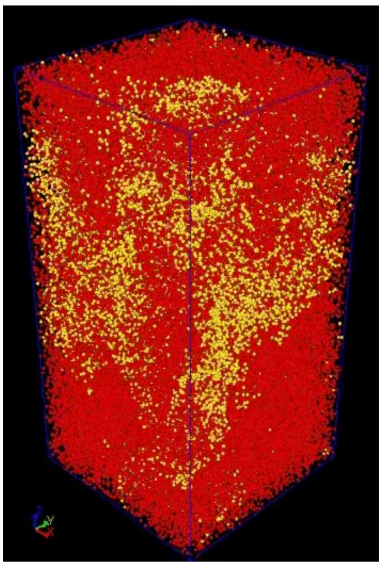


図3.4: 重力下での100000粒子の懸濁液の沈降．重力は，z方向下向きに働いている．粒子は全て等価であり，可視化のために赤と黄色に着色されている．

この例(Fig. 3.5)ではメッシュサイズ$128 \times 128 \times 128$で計算している．このとき粒子体積分率は$\varphi = 0.064$であり，粒子数$N_p = 500$, 粒子直径 $D = 8$, 界面厚さ $\xi = 2$，温度$k_BT = 5$である．溶媒は，密度$\rho = 1$，粘性係数$\eta = 1$である．図3.5には，反発相互作用を持った懸濁液の拡散のスナップショットを示している．

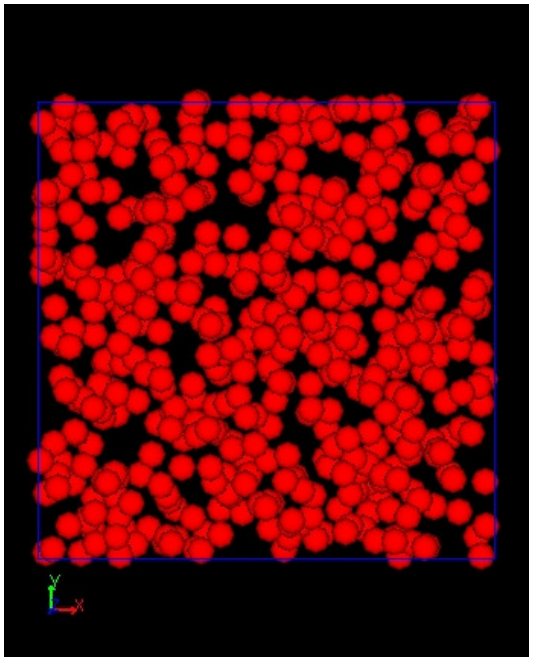


図3.5: 反発相互作用のみの粒子間相互作用をもつ懸濁液のスナップショットを示している．

3.3.3　粒子の凝集

入力UDFとして，`Examples/05/`フォルダの `aggregate.udf`，`aggregate_LE.udf`および`aggregate_LE_2.udf`を用いればよい. `aggregate.udf`では，粒子間に引力相互作用が働いた系の凝集・拡散現象を計算することがで

きる．

```
$ ../../kapsel -Iaggregate.udf -Ooutput.udf -Ddefine.udf -Rrestart.udf
```

この例(Fig. 3.6)ではメッシュサイズ$128 \times 128 \times 128$で計算している．このとき粒子体積分率は$\varphi = 0.064$であり，粒子数$N_p = 500$, 粒子直径 $D = 8$, 界面厚さ $\xi = 2$，温度$k_BT = 5$である．溶媒は，密度$\rho = 1$，粘性係数$\eta = 1$ である．図3.6には，粒子間に引力相互作用をもった懸濁液の凝集・拡散の様子を示している．

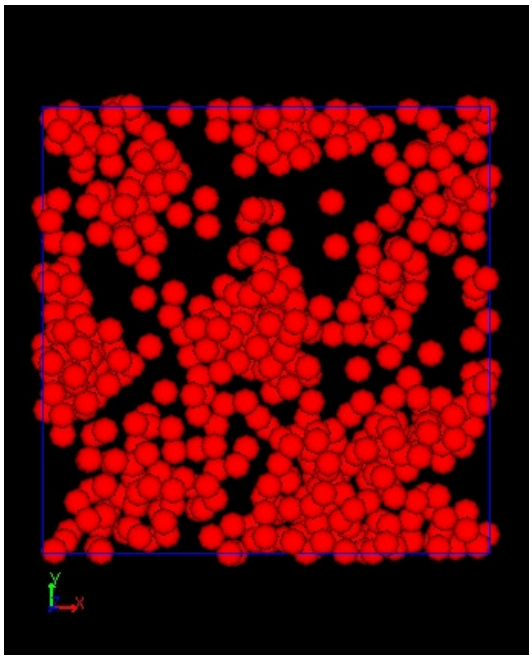


図3.6: 粒子間相互作用に引力相互作用をもつ懸濁液の凝集・拡散の様子のスナップショットを示している．

aggregate_LE.udfおよびaggregate_LE_2.udfでは，せん断流下での粒子間に引力相互作用が働いた系の凝集・拡散現象を計算することができる．

```
$ ../../kapsel -Iaggregate_LE.udf -Ooutput.udf -Ddefine.udf -Rrestart.udf
$ ../../kapsel -Iaggregate_LE_2.udf -Ooutput.udf -Ddefine.udf -Rrestart.udf
```

この例(Fig. 3.7)ではメッシュサイズ$128 \times 128 \times 128$で計算している．このとき粒子体積分率は$\varphi = 0.064$であり，粒子数$N_p = 500$, 粒子直径 $D = 8$, 界面厚さ $\xi = 2$，温度$k_BT = 5$である．溶媒は，密度$\rho = 1$，粘性係数$\eta = 1$ である．aggregate_LE.udfでは，せん断速度$\dot{\gamma} = 0.005$，aggregate_LE_2.udfでは，せん断速度$\dot{\gamma} = 0.05$である．図3.7には，せん斷流下での粒子間に引力相互作用をもった懸濁液の凝集・拡散の様子を示している．図3.7(左)には，弱いせん断流下($\dot{\gamma} = 0.005$)でのスナップショットを示している．また，図3.7(右)には，強いせん断流下($\dot{\gamma} = 0.05$)でのスナップショットを示している．

3.3.4　粒子鎖の運動

入力UDFとして，Examples/06/フォルダの s000_t010_free.udf，s001_t010_free.udfおよびs001_t000_free.udf を用いればよい．s000_t010_free.udfでは，静止流体中の1本の粒子鎖を計算することができる．一方，s001_t010_free.udfおよびs001_t000_free.udfでは，ジグザグせん断流下の1本の粒子鎖を計算することができる．

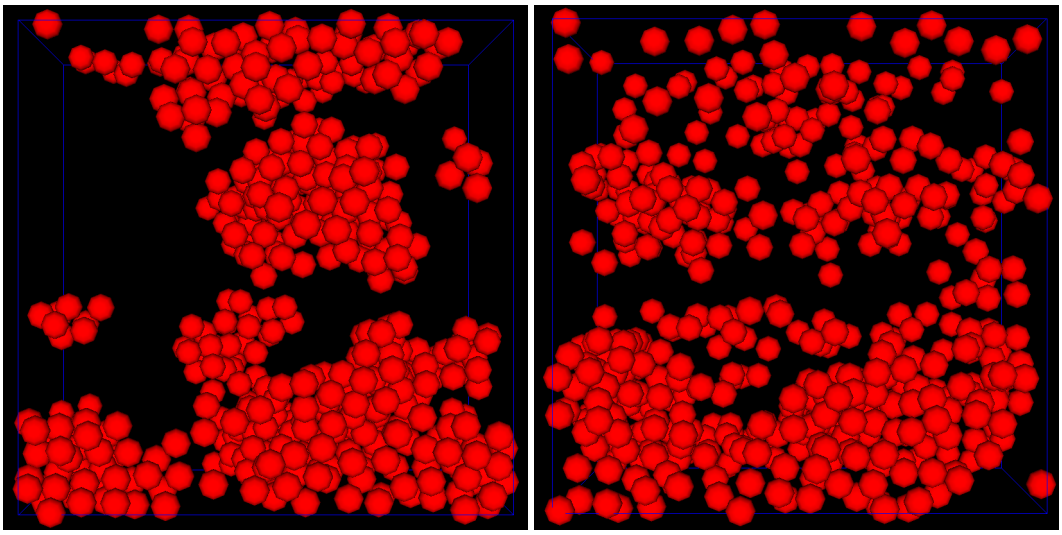


図3.7: せん断流下での粒子間相互作用に引力相互作用をもつ懸濁液の凝集・拡散の様子のスナップショットを示している．左図は弱いせん断流下でのスナップショットを示している．右図は強いせん断流下でのスナップショットを示している．

```
$ ../../kapsel -Is000_t010_free.udf -Ooutput.udf -Ddefine.udf -Rrestart.udf
$ ../../kapsel -Is010_t010_free.udf -Ooutput.udf -Ddefine.udf -Rrestart.udf
$ ../../kapsel -Is010_t000_free.udf -Ooutput.udf -Ddefine.udf -Rrestart.udf
```

この例(Fig. 3.8)ではメッシュサイズ$32 \times 64 \times 16$で計算している．このとき粒子体積分率は$\varphi = 0.005$であり，1本の鎖の粒子数$N_p = 5$，粒子直径$D = 4$，界面厚さ$\xi = 2$である．`s000_t010_free.udf`および`s001_t010_free.udf`では，温度$k_BT = 1.0$，`s001_t000_free.udf`では，温度$k_BT = 0.0$である．`s001_t010_free.udf`および`s001_t000_free.udf`でのせん断速度$s = 0.01$である．図3.8には，1本の柔軟な粒子鎖のスナップショットを示している．図3.8(左)には，温度$k_BT = 1.0$での静止流体中での粒子鎖のスナップショットを示している．図3.8(中央)には，温度 $k_BT = 1.0$でのジグザグせん断流下の粒子鎖のスナップショットを示している．図3.8(右)には，温度$k_BT = 0.0$でのジグザグせん断流下の粒子鎖のスナップショットを示している．

3.3.5　任意形状剛体粒子の運動

入力UDFとして，`Examples/08/`フォルダの`N30k3.0p52.0_vy.udf`，`N30k3.0p52.0_omegay.udf`，`N30k3.0p52.0_gravity.udf`，`helixes_gravity.udf`，`helixes_shear0005.udf`を用いればよい．`N30k3.0p52.0_vy.udf`では，一定速度で並進運動する螺旋粒子鎖の計算をすることができる．

```
$ ../../kapsel -IN30k3.0p52.0_vy.udf -Ooutput.udf -Ddefine.udf -Rrestart.udf
```

この例ではメッシュサイズ$256 \times 256 \times 256$で計算している．このとき粒子体積分率は$\varphi = 0.00048$であり，1本の鎖の粒子数$N_p = 30$，粒子直径$D = 4$，界面厚さ$\xi = 1$，速度$V = -0.005$である．

`N30k3.0p52.0_omegay.udf`では，一定の角速度で回転した螺旋粒子鎖の計算をすることができる．

```
$ ../../kapsel -IN30k3.0p52.0_omegay.udf -Ooutput.udf -Ddefine.udf -Rrestart.udf
```

この例ではメッシュサイズ$256 \times 256 \times 256$で計算している．このとき粒子体積分率は$\varphi = 0.00048$であり，1本の鎖の粒子数$N_p = 30$，粒子直径$D = 4$，界面厚さ$\xi = 1$，角速度$\Omega = 0.00041$である．

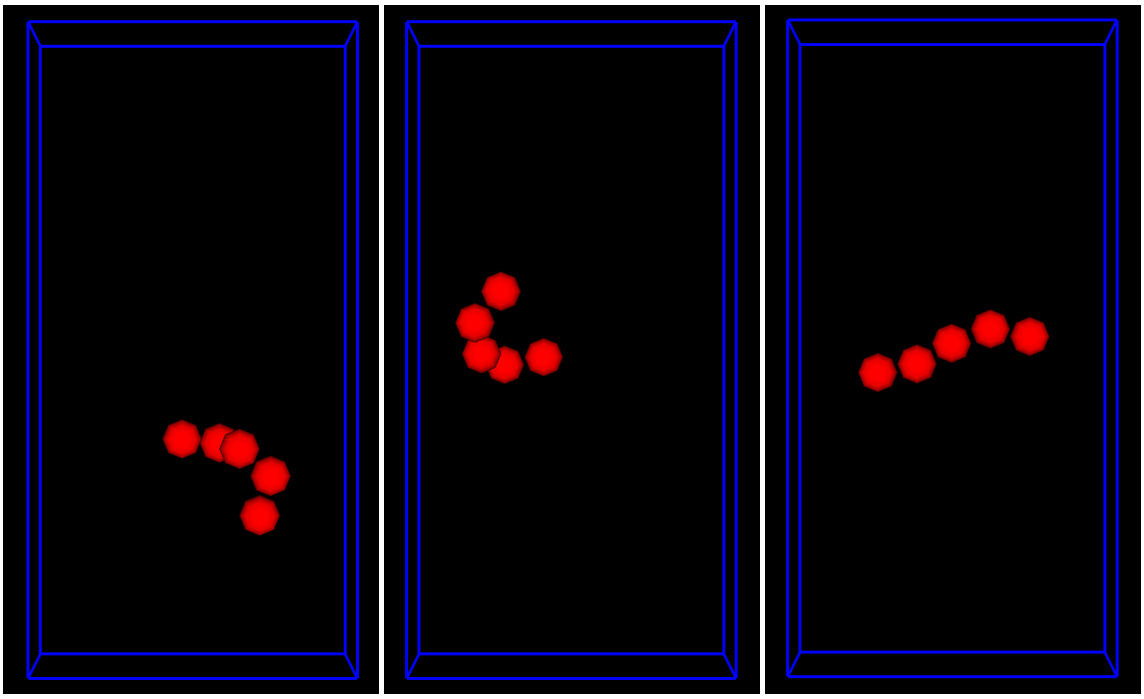


図3.8: 左図は温度$k_BT = 1.0$での静止流体中の1本の柔軟な粒子鎖のスナップショットを示している．中央図は温度$k_BT = 1.0$でのジグザグせん断流下のスナップショットを示している．右図は温度$k_BT = 0.0$でのジグザグせん断流下でのスナップショットを示している．

N30k3.0p52.0_gravity.udfでは，一定の外力（重力）の下で沈降する螺旋粒子鎖の計算をすることができる．

```
$ ../../kapsel -IN30k3.0p52.0_gravity.udf -Ooutput.udf -Ddefine.udf -Rrestart.udf
```

この例ではメッシュサイズ$256 \times 256 \times 256$で計算している．このとき粒子体積分率は$\varphi = 0.00048$であり，1本の鎖の粒子数$N_p = 30$，粒子直径$D = 4$，界面厚さ$\xi = 1$，重力$g = -0.01$である．

helixes_gravity.udfでは，重力下での3本の螺旋粒子鎖の沈降の計算をすることができる．

```
$ ../../kapsel -Ihelixes_gravity.udf -Ooutput.udf -Ddefine.udf -Rrestart.udf
```

この例(Fig. 3.9)ではメッシュサイズ$256 \times 256 \times 256$で計算している．このとき粒子体積分率は$\varphi = 0.0016$であり，1本の鎖の粒子数$N_p = 100$，粒子直径$D = 4$，界面厚さ$\xi = 1$，重力$g = -0.01$である．図3.9には，重力下での3本の螺旋粒子鎖の沈降のスナップショットを示している．

helixes_shear0005.udfでは，せん断流下の3本の螺旋粒子鎖の計算をすることができる．

```
$ ../../kapsel -Ihelixes_shear0005.udf -Ooutput.udf -Ddefine.udf -Rrestart.udf
```

この例ではメッシュサイズ$256 \times 256 \times 256$で計算している．このとき粒子体積分率は$\varphi = 0.0016$であり，1本の鎖の粒子数$N_p = 100$，粒子直径$D = 4$，界面厚さ$\xi = 1$，せん断速度$\dot{\gamma} = 0.005$である．

3.3.6　任意形状剛体粒子の沈降

入力UDFとして，Examples/10/フォルダの input_rigid_free1.udfを用いればよい．input_rigid_free1.udfでは，$Re = 800$で落下する物体の計算をすることができる．

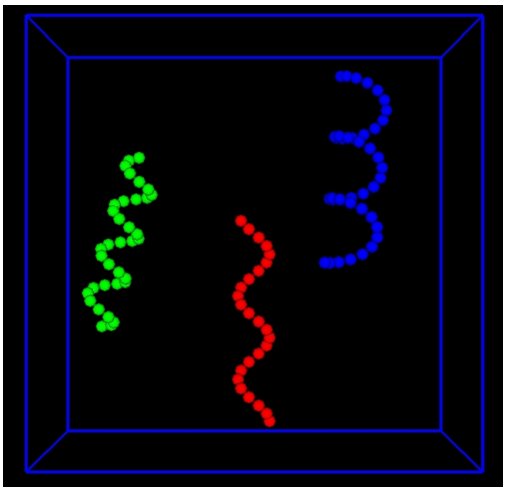


図3.9: 重力下での3本の螺旋粒子鎖の沈降のスナップショット

```
$ ../../kapsel -Iinput_rigid_free1.udf -Ooutput.udf -Ddefine.udf -Rrestart.udf
```

この例(Fig. 3.10)ではメッシュサイズ$64 \times 128 \times 64$で計算している．落下する物体の密度は流体の密度の1.5倍であり，重力$g = -50$である．図3.10には，$Re \simeq 800$で落下する物体のスナップショットを示す．

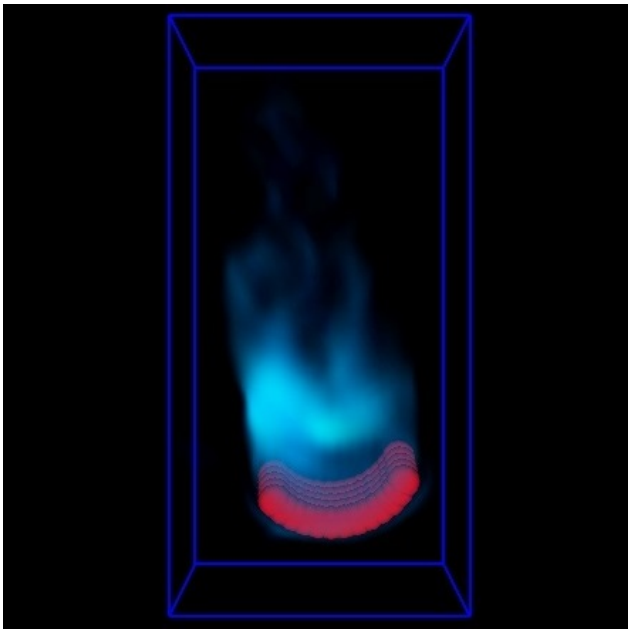


図3.10: $Re = 800$で落下する物体のスナップショット．流体中の水色のレンダリングは流速の大きさを表す．

第4章

せん断流下の粒子分散系のシミュレーション

4.1 理論的背景と基礎方程式

微粒子が液体（溶媒）中に分散している微粒子懸濁液は，化学工学や機械工学，物理学など工学・理学分野の広範囲にわたって重要であり，活発に研究が行われている．このような系は，食品・塗料・顔料・化粧品・スラリーなどに見出すことができ，その用途および産業的な製造・加工プロセスにおいてその流動挙動の把握が重要である．しかしながら，微粒子懸濁液の動的挙動は微粒子間相互作用や熱揺らぎなどに影響され大変複雑な様相を示し，その流動挙動の把握は容易ではない．微粒子懸濁液は，異なる状況のもとで多種多様な挙動を示す．例えば，微粒子懸濁液の輸送現象においては粘度のような微粒子懸濁液の流動特性が，分散した粒子の濃度や流動の強さ，粒子界面の性質などに強く依存する．このような巨視的な流動特性と微視的なメカニズムの関係を明らかにすることは，微粒子懸濁液の基礎研究における主要なテーマの一つとなっている．

上述のように懸濁液のレオロジーは応用上重要であるが，それについて理論的に一般論を構築するにはあまりに複雑である (例えばテキスト [15]や比較的最近のレビュー(日本語) [16, 17]を参照.)．溶媒の粘度をηとすると，粒子濃度（体積分率φ）が希薄な領域においては懸濁液の粘度η^{app}は

$$\frac{\eta^{app}}{\eta} = 1 + \frac{5}{2}\varphi, \tag{4.1}$$

のようにφに対して線形に変化するという理論的な結果が得られている(例えば [15]を参照)．これは有名な Einstein の粘度式である．しかし Einstein の粘度式が成立するのは体積分率で$\varphi \ll 10\%$程度である．粒子濃度が高くなると，粒子間相互作用の影響とそれによる粒子構造変化、さらにガラス化・結晶化などが起こる．そのため懸濁液のレオロジーを基本法則から理論的に予測することは途端に難しくなる．以上のような難しさはあるものの，その重要性から実験的な研究は数多くなされ，それを整理する形で懸濁液の粘度についての経験式・半経験式がいくつか提案されている (Doughherty–Krieger 式[18],文献 [17, 19] およびその引用文献参照).

粒子濃度が高くなっていくと，コロイド体積分率以外の多様な効果が現象に関与するため，統一的な理論を作るよりも現象個別に対する解析が必要になってくる． SP法は溶媒によるコロイド粒子間多体相互作用を直接計算できるシミュレーション法であり，これを用いて多様な種類の懸濁液のレオロジーを解析することが可能である．レオロジー解析を可能とする本直接数値計算手法の主な特徴は，「粒子間の流体力学的相互作用」，「粒子の熱揺らぎ」，そして「せん断流」が考慮されている点である． KAPSELでは，Newton流体中の単分散系についての単純および振動せん断流下のシミュレーションおよびレオロジー測定(定常流および動的粘弾性測定)が行える．

KAPSELでは，均一なせん断流を実現するためにLees–Edwards周期境界条件を採用している． Lees–Edwards周期境界条件の利点として，壁などの特殊な境界を用いることなくせん断速度を正確に与えることが可能で，かつジグザク流れのように系内に速度勾配の符号が変わる特異点がないことが挙げられる．本周期境界条件では，せん断流れによって時間の経過に伴い格子点が移流する斜交座標上で構成方程式を解く． Lees–Edwards周期境界条件を採用した場合，せん断流れの実現に伴う座標変換がどのように行われるかの概要を図4.1に示す．時刻 $t = t_0$ においてせん断流れがない場合，溶媒中に分散した固体粒子は矩形格子上で離散化される(図4.1 (a))．せん断流

れが生じる($t > t_0$)と，格子点と粒子の両方が流れにより移流するが，計算格子はせん断により変形する一方で，粒子の形は変化しない．すなわち計算格子は，固定座標系(図4.1 (b))とせん断流れに伴って歪む斜交座標系(図4.1 (c))の2通りで表され，後者の斜交座標系では粒子が歪んだ形となる．この論述は，非平衡系MDシミュレーションにおいて発散なしの均一な流れをモデル化するために用いられるSLLODアルゴリズムと同等であるが，格子ベースの系においてこのアルゴリズムを実装する際は注意が必要である[20–22]．時間の経過とともにせん断流れによって変形する斜交座標系上での速度を $\boldsymbol{\xi} = \boldsymbol{u} - \boldsymbol{U}$ とすると，速度 $\boldsymbol{\xi}$ に対するNavier–Stokes方程式の適切な反変形(contravariant form)は，基底ベクトル($\hat{\boldsymbol{E}}_1 = \boldsymbol{e}_x + \gamma(t)\boldsymbol{e}_y, \hat{\boldsymbol{E}}_2 = \boldsymbol{e}_2, \hat{\boldsymbol{E}}_3 = \boldsymbol{e}_3$)を用いて

$$\rho\left(\partial_{\hat{t}} + \hat{\xi}^j\hat{\nabla}_j\right)\hat{\xi}^i = \hat{\nabla}_j\hat{\sigma}^{ji} + \hat{\phi}\hat{f}^i - 2\dot{\gamma}(\hat{t})\hat{\xi}^2\delta^{i,1} \tag{4.2}$$

$$\hat{\nabla}_i\hat{u}^i = \hat{\nabla}_i\hat{\xi}^i = 0 \tag{4.3}$$

で与えらる．ここで，$(\hat{\cdot})$は斜交座標系上でのテンソル成分を表す．上記の流れに関する方程式を解いた後，固定座標系(実験室系)において粒子–流体間の相互作用を計算し粒子ダイナミクスを解く．

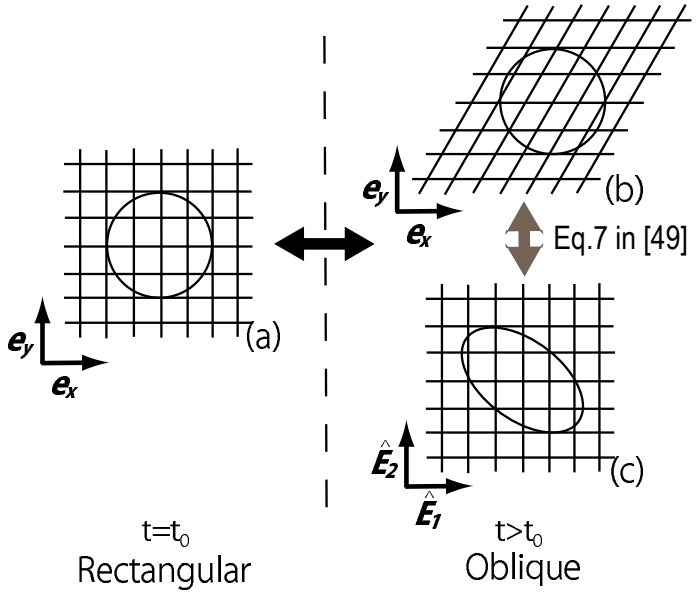


図4.1: せん断流れを実現するためのLees–Edwards周期境界条件における座標変換の概要． (a)時刻 $t = t_0$ において固体粒子は矩形格子上で離散化される．せん断流れが生じる($t > t_0$)ことで粒子と溶媒(格子点)の両方が移流するが，計算格子は変形する一方で粒子の形状は変わらない．つまり，固定実験系(b)では計算格子が変形する一方で，斜交座標系(c)では粒子がせん断流れに応じて歪む． (b)と(c)間の座標の変換については文献[22]式(7)を参照されたい． Reproduced from J. Chem. Phys 134, 064110[22], Copyright 2011, with the permission of AIP Publishing.

せん断流下において，分散系全体の瞬間的な応力を空間平均したものは，粒子の拘束力 $\rho\phi\boldsymbol{f}_p$ を用いて

$$\boldsymbol{\Sigma} = \frac{1}{V}\int \mathrm{d}\boldsymbol{x}\left[\boldsymbol{\sigma} - \boldsymbol{x}\rho\phi\boldsymbol{f}_p + \boldsymbol{x}\boldsymbol{u}\cdot\boldsymbol{\nabla}(\rho\boldsymbol{u})\right], \tag{4.4}$$

として定義される．また，ホスト流体による応力への寄与の平均を引くことにで，応力に対する粒子の寄与 $\boldsymbol{s} = \boldsymbol{\Sigma} - \langle\boldsymbol{\sigma}\rangle$ が得られる．

4.1.1　定常せん断流の場合

懸濁液の粘度 η_{app}，および特性粘度 $[\eta]$ は，それぞれ

$$\eta_{app} = \frac{\langle\Sigma^{xy}\rangle}{\dot{\gamma}} = \frac{\langle\sigma^{xy}\rangle}{\dot{\gamma}} + \frac{\langle s^{xy}\rangle}{\dot{\gamma}} = \eta + \frac{\langle s^{xy}\rangle}{\dot{\gamma}} \tag{4.5}$$

$$[\eta] = \frac{\eta_{app} - \eta}{\eta\varphi} = \frac{\langle s^{xy} \rangle}{\eta\varphi\dot{\gamma}} \tag{4.6}$$

で求めることができる．

4.1.2　振動せん断流の場合

振動せん断流場を系に導入することによって，懸濁液の動的粘弾性を測定することが可能である． KAPSELでは，系に時間的に変化するせん断流を形成するために，制御パラメータとして，ずり速度$\dot{\gamma}$を次のように与えている．

$$\dot{\gamma}(t) = \dot{\gamma}_0 \cos(\omega t). \tag{4.7}$$

ここで，$\dot{\gamma}_0$はずり速度の振幅である．また、ひずみの最大振幅γ_0は，

$$\gamma_0 = \int_0^{\pi/2\omega} \dot{\gamma}(s) ds = \frac{\dot{\gamma}_0}{\omega} \tag{4.8}$$

によって評価される．ひずみが小さい場合は線形粘弾性の領域に相当し，一方大きい場合は非線形粘弾性領域に相当する．

振動せん断流れ$\dot{\gamma}(t)$に対する懸濁液の応力の時間変化$\Sigma^{xy}(t)$)は，線形応答の範囲内で

$$\Sigma^{xy}(t) = \sigma_0 \cos(\omega t - \delta) \tag{4.9}$$

と書ける．ここで，σ_0は応力の振幅であり，δはずり速度$\dot{\gamma}$とずり応力Σ^{xy}との間の位相差をあわらしている．この時，動的粘弾性関数である貯蔵弾性率G'および，損失弾性率G''は，

$$G'(\omega) = \frac{\omega\sigma_0 \sin\delta}{\dot{\gamma}_0},\ G''(\omega) = \frac{\omega\sigma_0 \cos\delta}{\dot{\gamma}_0}. \tag{4.10}$$

と書ける．G'は懸濁液の弾性的な挙動を特徴づけ，G''は粘性的な挙動を特徴づける． KAPSELにおいては，式(4.7))のずり速度$\dot{\gamma}(t)$が入力であり，式(4.9)の懸濁液のずり応力$\Sigma^{xy}(t)$がその応答（出力）となっている．ωを変えてシミュレーションを行い，それぞれの条件下で変数の時間変化を上式にフィッティングすることにより$G'(\omega), G''(\omega)$を求めることが可能である．これらの関係式の導出については，レオロジーに関する教科書を参照して頂きたい．

4.2　入力UDF について

4.2.1　流体の設定

constitutive_eqとしてShear_Navier_Stokes_Lees_Edwardsを選ぶとシア流下における Newton 流体中のコロイド粒子の運動のシミュレーションが行える．以下, 入力UDFファイルで指定する変数について解説する．

constitutive_eq.Shear_Navier_Stokes以下で溶媒の情報を指定する．

- constitutive_eq.Shear_Navier_Stokes.DX…長さの単位量である格子幅Δ.
- constitutive_eq.Shear_Navier_Stokes.RHO…溶媒の密度.
- constitutive_eq.Shear_Navier_Stokes.ETA…溶媒の粘度.
- constitutive_eq.Shear_Navier_Stokes.kBT…粒子温度.
- constitutive_eq.Shear_Navier_Stokes.alpha_v…粒子の並進温度に対する補正項 [*1].
- constitutive_eq.Shear_Navier_Stokes.alpha_o…粒子の回転温度に対する補正項.
- constitutive_eq.Shear_Navier_Stokes.External_field.type…DC (定常シア流れ）または AC (振動シア流れ)を選択する.
- constitutive_eq.Shear_Navier_Stokes.External_field.DC.Shear_rate…ずり速度 $\dot{\gamma}$.
- constitutive_eq.Shear_Navier_Stokes.External_field.AC.Shear_rate…振動ずり速度の振幅 $\dot{\gamma}_0$
- constitutive_eq.Shear_Navier_Stokes.External_field.AC.Frequency…ずり速度の周波数 ω

4.2.2　オブジェクト（粒子）の設定

object_type.typeでは，spherical_particle(球状粒子)，chain(フレキシブル鎖)，rigid(剛体)を選択できる．

[*1] 熱平衡での粒子の拡散運動から温度は見積もられるが，界面の厚さ等の数値的な理由のために正しい温度を返さない可能性がある．この場合，もし測定された温度が1.2倍と低い温度が見積もられた場合，このファクターを1.2と設定することで既定の温度を実現することができる．回転に対しても同様である．半径5または4，かつ$\xi = 2$に対しては，alpha_v=alpha_o=1で既定の温度を計算できる．

4.3　計算事例

4.3.1　定常せん断流下の懸濁液のレオロジー

入力UDFとして，Examples/07/フォルダの colloid_LE_v31_t01.udfおよびcolloid_LE_v55_t50.udfを用いればよい．

```
$ ../../kapsel -Icolloid_LE_v31_t01.udf -Ooutput.udf -Ddefine.udf -Rrestart.udf
$ ../../kapsel -Icolloid_LE_v55_t50.udf -Ooutput.udf -Ddefine.udf -Rrestart.udf
```

この例(Fig. 4.2)ではメッシュサイズ$64 \times 64 \times 64$でせん断速度$\dot{\gamma} = 0.001$, 粒子半径$a = 4$, 界面厚さ$\xi = 2$ で計算している． colloid_LE_v31_t01.udfでは，温度$k_BT = 0.1$，粒子数$N_p = 300$であり，このとき粒子体積分率は$\varphi = 0.307\%$である．一方，colloid_LE_v55_t50.udfでは，温度$k_BT = 5.0$，粒子数$N_p = 540$であり，このとき粒子体積分率は$\varphi = 0.552\%$である．

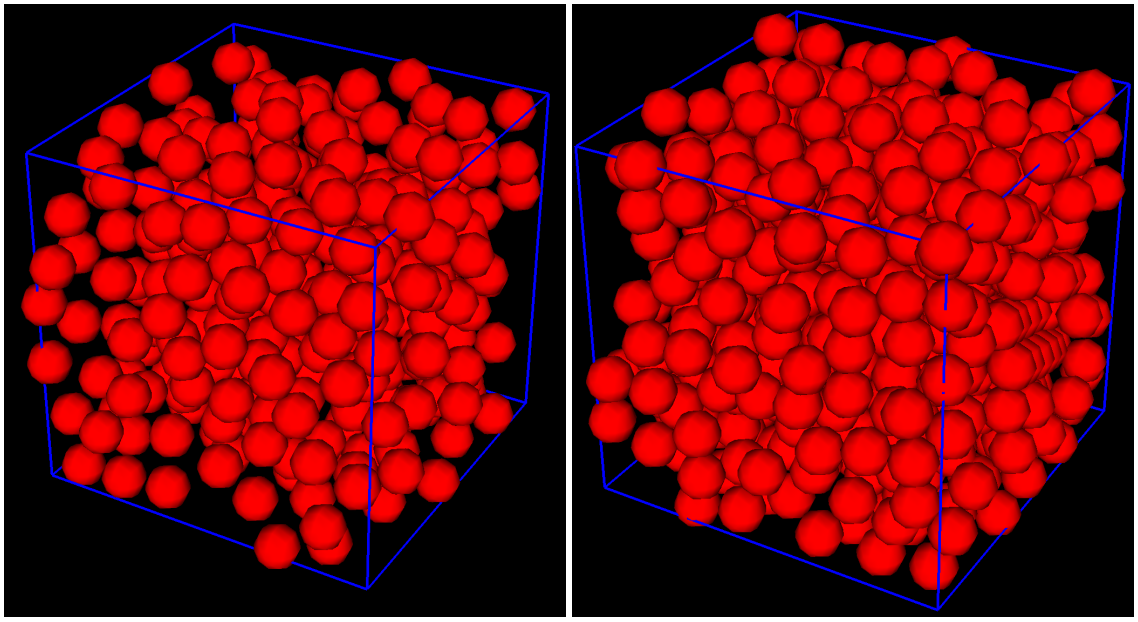


図4.2: Lees-Edwards周期境界条件を用いた定常せん断流下の懸濁液の流動．左図は温度$k_BT = 0.1$，粒子数$N_p = 300$のスナップショット，右図は温度$k_BT = 5.0$，粒子数$N_p = 540$のスナップショットを示している．

constitutive_eqとしてShear_Navier_Stokes_Lees_Edwardsを選び， Externa_fieldとしてDCを選択すると，コマンドライン（標準エラー出力）に

```
#1:time 2:shear_rate 3:degree_oblique 4:shear_strain_temporal 5:lj_dev_stress_temporal 6:
0.0748173 -0.00588609 -0.000440381  ...
```

⋮

のようなデータが出力される．この時系列出力の意味は以下のとおり。

- 1:time...経過時間
- 2:shear_rate...せん断速度

- `3:degree_oblique`...斜交座標系の歪み [*2]
- `4:shear_strain_temporal`...印加された歪み($\dot{\gamma}t$)
- `5:lj_dev_stress_temporal`...粒子間ポテンシャル由来のせん断応力
- `6:shear_stress_temporal_old`...粒子–流体間相互作用由来のせん断応力
- `7:shear_stress_temporal_new`...粒子–流体間相互作用由来のせん断応力 [*3]
- `8:reynolds_stress`...レイノルズせん断応力
- `9:fluid_stress`...バルク流体由来のせん断応力
- `10:interfacial_stress`...流体–流体界面由来のせん断応力
- `11:apparent_stress`...懸濁液のせん断応力 [*4]
- `12:viscosity`...懸濁液の粘度 [*5]

Fig. 4.3は，懸濁液の応力の時間発展を示している．また，Fig. 4.4は，粘度とひずみの関係を示している．

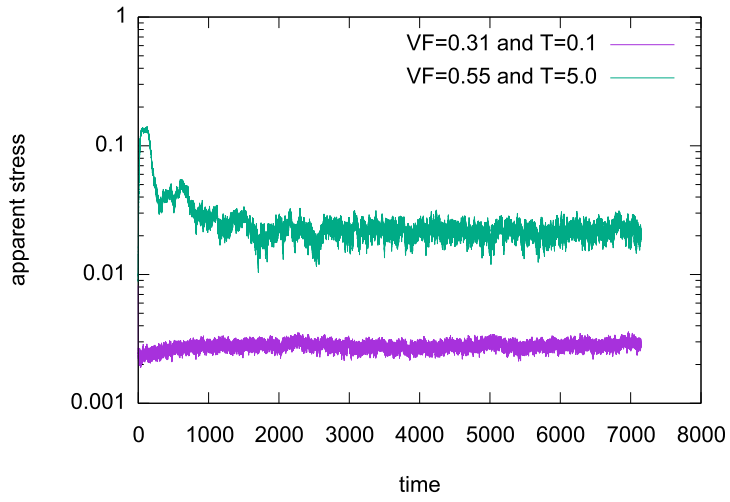


図4.3: 定常せん断流下の懸濁液の応力の時間発展. 1列目と11列目のプロットを表示している.

4.3.2　振動せん断流下の懸濁液のレオロジー

入力UDFとして，`Examples/07/`フォルダの`colloid_LE_v31_t01_AC.udf`および`colloid_LE_v55_t50_AC.udf`を用いればよい.

```
$ ../../kapsel -Icolloid_LE_v31_t01_AC.udf -Ooutput.udf -Ddefine.udf -Rrestart.udf
$ ../../kapsel -Icolloid_LE_v55_t50_AC.udf -Ooutput.udf -Ddefine.udf -Rrestart.udf
```

この例(Fig. 4.5)ではメッシュサイズ$64 \times 64 \times 64$でせん断速度$\dot{\gamma} = 0.001$, 粒子半径$a = 4$, 界面厚さ$\xi = 2$ で計算している．`colloid_LE_v31_t01_AC.udf`では，温度$k_BT = 0.1$，粒子数$N_p = 300$であり，このとき粒子体積分率

[*2] KAPSELでは，一様なせん断流動を実現するため，時間と共にせん断方向に歪む斜交座標系上で計算を行っている．その際，計算安定性の観点から，斜交座標系の歪みがある一定値に到達したとき歪みをリセットし，もとの直交座標系へ値をリマップする操作を行っている． `3:degree_oblique`で表示される値は，このリマップを考慮した斜交座標系の歪みであり，系に印加された歪み($\dot{\gamma}t$)とは異なることに注意する．詳細は文献[23]に記されている.

[*3] 粒子–流体間相互作用由来のせん断応力は，oldとnewの2種類の計算方法が実装されている．通常はnewを用いればよい.

[*4] `5:lj_dev_stress_temporal`，`7:shear_stress_temporal_new`，`8:reynolds_stress`，`9:fluid_stress`，`10:interfacial_stress`の総和である.

[*5] `11:apparent_stress`を`2:shear_rate`で割った値である.

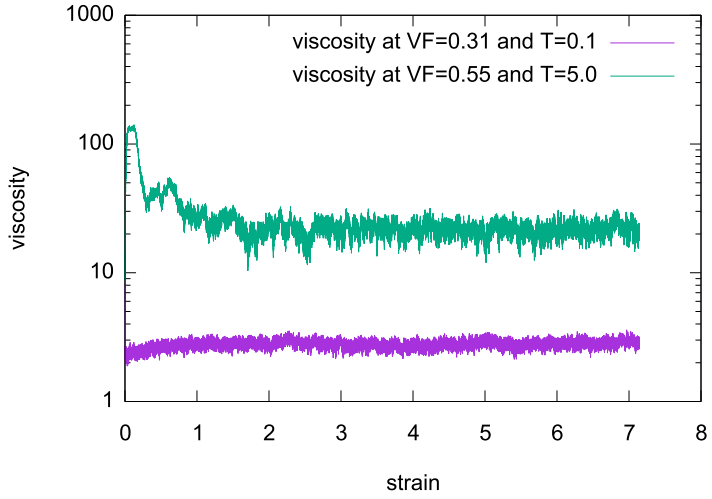


図4.4: 定常せん断流下の懸濁液のひずみと粘度の関係. 4列目と12列目のプロットを表示している.

は$\varphi = 0.307\%$である．一方，colloid_LE_v55_t50_AC.udfでは，温度$k_BT = 5.0$，粒子数$N_p = 540$であり，このとき粒子体積分率は$\varphi = 0.552\%$である．
constitutive_eqとしてShear_Navier_Stokes_Lees_Edwardsを選び， Externa_fieldとしてACを選択すると標準エラー出力に

```
#1:time 2:shear_rate 3:degree_oblique 4:shear_strain_temporal 5:lj_dev_stress_temporal 6:
0.0748173 -0.00588609 -0.000440381  ...
```

⋮

のようなデータが出力される．この時系列出力の意味は以下のとおり。

- 1:time...経過時間
- 2:shear_rate...せん断速度
- 3:degree_oblique...斜交座標系の歪み [*6]
- 4:shear_strain_temporal...印加された歪み($\dot{\gamma}t$)
- 5:lj_dev_stress_temporal...粒子間ポテンシャル由来のせん断応力
- 6:shear_stress_temporal_old...粒子–流体間相互作用由来のせん断応力
- 7:shear_stress_temporal_new...粒子–流体間相互作用由来のせん断応力 [*7]
- 8:reynolds_stress...レイノルズせん断応力
- 9:fluid_stress...バルク流体由来のせん断応力
- 10:interfacial_stress...流体–流体界面由来のせん断応力
- 11:apparent_stress...懸濁液のせん断応力 [*8]
- 12:viscosity...懸濁液の粘度 [*9]

[*6] KAPSELでは，一様なせん断流動を実現するため，時間と共にせん断方向に歪む斜交座標系上で計算を行っている．その際，計算安定性の観点から，斜交座標系の歪みがある一定値に到達したとき歪みをリセットし，もとの直交座標系へ値をリマップする操作を行っている． 3:degree_obliqueで表示される値は，このリマップを考慮した斜交座標系の歪みであり，系に印加された歪み($\dot{\gamma}t$)とは異なることに注意する．詳細は文献[23]に記されている．

[*7] 粒子–流体間相互作用由来のせん断応力は，oldとnewの2種類の計算方法が実装されている．通常はnewを用いればよい．

[*8] 5:lj_dev_stress_temporal，7:shear_stress_temporal_new，8:reynolds_stress，9:fluid_stress，10:interfacial_stressの総和である．

[*9] 11:apparent_stressを2:shear_rateで割った値である．

Fig. 4.5は，懸濁液の応力の時間発展を示している．

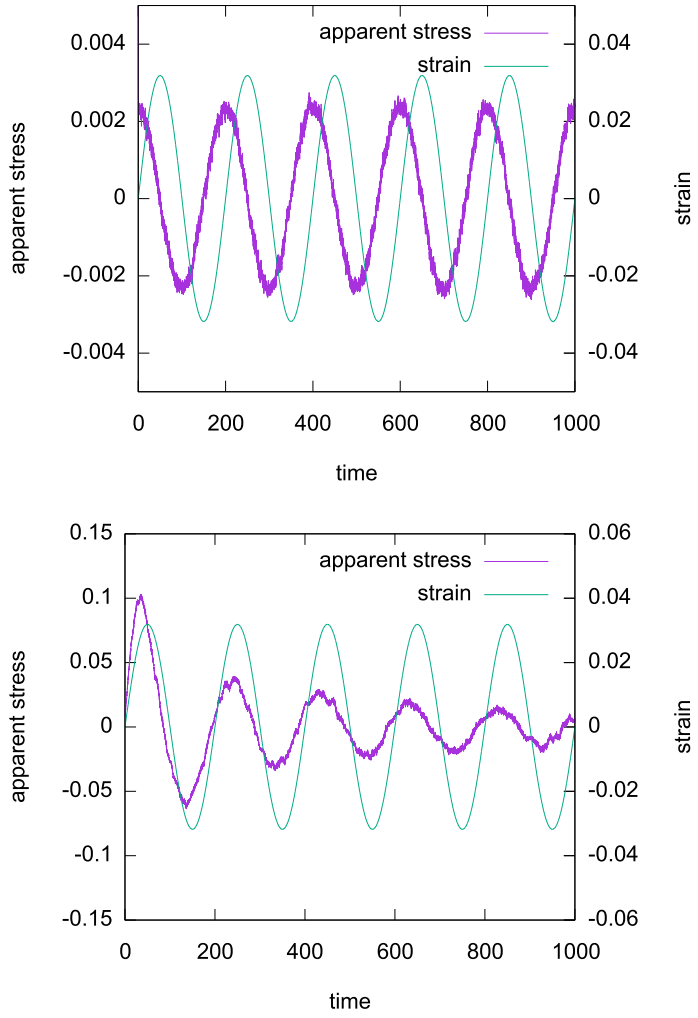


図4.5: 振動せん断流下の懸濁液のせん断応力の時間発展. 1列目と11列目のプロットを表示している．上図は粒子体積分率$\varphi = 0.307$，温度$k_BT = 0.1$での懸濁液の応力の時間発展を示している．下図は粒子体積分率$\varphi = 0.552$，温度$k_BT = 5.0$での懸濁液の応力の時間発展を示している．

第5章

電解質溶液に分散した荷電粒子のシミュレーション

5.1 荷電コロイド粒子の電気泳動

水のような誘電率が非常に大きい溶媒にコロイド粒子が分散しているとき, コロイド表面にある解離基からイオンが放出されて粒子表面は電荷を帯びる. 放出されたイオンは粒子表面に静電的に引き寄せられるが, 同時に熱ゆらぎによって拡散され, コロイド粒子周りに電気二重層とよばれる雲のようなイオン雰囲気を形成する. 平衡状態のコロイド分散系の性質はPoisson–Boltzmann方程式によって記述され, これを線型化したDebye–Hückel近似を用いて見通しの良い理解を得ることができる. これに対して電気泳動をはじめとした界面動電現象とよばれる現象では, 粒子とイオン分布の挙動は流体力学相互作用と静電相互作用の競合によって決定される. 両者の競合の結果としてイオン分布は粒子の運動に追随できず, 電気二重層は球対称から歪み平衡状態とは異なる状況が起こる. このような複雑な状況を解析するには, 理論的には簡単な近似を導入する他なく, さらに計算機シミュレーションによっても正しく再現されたことはこれまでほとんどなかった.

SP法ではコロイドは粒子描像のまま扱う． その一方で, 溶媒およびイオンについては連続体として粗視化された密度場として与える． この計算の実現のために，滑らかな界面を与えるSP関数を用いてコロイド粒子(運動方程式)・イオン分布(移流拡散方程式)・溶媒速度場(Navier–Stokes 方程式)の3つの自由度とそれらのカップリングをコンシステントに解くためのフォーマリズムを構成した [5, 6, 24, 25]. これにより, 定量的な電気泳動シミュレーションにも世界で初めて成功した [25]. 以下, KAPSEL内部で考慮されている基本方程式を説明し, 同時に背景にある理論について簡単に触れる.

5.2 基本方程式

文献 [6, 25]にしたがいシミュレーションに必要な電気流体力学の基本方程式を概観する. 電解質溶媒中に分散している半径aの球状コロイド粒子N個を考える. また溶媒の誘電率ϵは, コロイド粒子内部も含めて空間的に一様であるとする. SP法では粒子と溶媒の界面を有限な厚さξをもつ滑らかな関数$\phi(\boldsymbol{r}) \in [0, 1]$によってあらわす. 固定直交格子上で粒子領域を $\phi = 1$, 溶媒領域を$\phi = 0$, 界面領域を$0 < \phi < 1$で表現している. この様な直交格子状で$\phi(\boldsymbol{r})$を用いた方法としては, 他に Tanaka–Araki [26]やKajishimaら [27]の方法がある. 界面関数を導入することによって, 非構造格子を用いた有限要素法などと比べた場合に圧倒的な計算効率の向上を実現できる. コロイド粒子の表面は一様に帯電していると仮定し, 1粒子あたりの帯電量はZeであるとする. 通常の連続体描像では, 粒子表面電荷分布はデルタ関数を用いて表される. そのため有限要素法などでは, 適切な境界適合格子が用いられ計算効率向上の大きな障害となっている. それに対して, SP法ではこの粒子表面電荷分布$eq(\boldsymbol{r})$ についてもSP関数ϕの 1階微分を用いて

$$eq(\boldsymbol{r}) = \frac{Ze|\nabla\phi(\boldsymbol{r})|}{4\pi a^2}, \tag{5.1}$$

のように滑らかに表現する. SP関数$\phi(\boldsymbol{r})$が$\xi \to 0$でステップ関数に帰着するように構成されているのと同様に, $q(\boldsymbol{r})$も$\xi \to 0$とすればデルタ関数に帰着するように構成されている.

5.2.1　移流拡散方程式

価数z_αをもつα種イオンの密度分布C_αを

$$C_\alpha(\boldsymbol{r}, t) = (1 - \phi(\boldsymbol{r}, t))C^*_\alpha(\boldsymbol{r}, t) \tag{5.2}$$

のように計算領域全体で定義する. イオンが存在する領域は$(1 - \phi)$によって表現されている. $C^*_\alpha(\boldsymbol{r}, t)$ は計算のための補助変数であり, 計算領域全体で滑らかに定義されている. コロイドの領域($\phi = 1$)におけるC^*_α に物理的な意味はない. 粒子表面電荷分布を含めた全電荷分布は

$$\rho_e(\boldsymbol{r}) = e \sum_\alpha z_\alpha C_\alpha(\boldsymbol{r}) + eq(\boldsymbol{r}) \tag{5.3}$$

である. このとき初期分布は電気的中性条件 $\int \rho_e \mathrm{d}\boldsymbol{r} = 0$ を満たすようにとられる.

補助イオン密度C^*_αの時間発展は移流拡散方程式:

$$\partial_t C^*_\alpha = -\nabla \cdot C^*_\alpha \boldsymbol{v} + \Gamma_\alpha \nabla \cdot (C^*_\alpha \nabla \mu_\alpha), \tag{5.4}$$

にしたがうとする. これは溶媒速度場$\boldsymbol{v}$による移流と, 化学ポテンシャルμ_αの勾配による拡散の2つの項からなる. C^*_αは移流拡散方程式にしたがうので、 $\int \mathrm{d}\boldsymbol{r} C^*_\alpha$は保存する。

イオンがコロイド粒子内部に浸透しないことは, イオン拡散流速の界面法線方向成分が0となる条件として考慮される。すなわち, $\boldsymbol{n} \cdot \nabla \mu_\alpha = 0$ である [6, 25]. ここで $\boldsymbol{n}$は粒子表面における(外向き)法線方向ベクトルを表す. SP関数を用いると$\boldsymbol{n} = -\nabla\phi/|\nabla\phi|$と表すことができる. また Γ_αはα種イオンのOnsager輸送係数であり, イオンの摩擦係数および拡散係数とは$f_\alpha = 1/\Gamma_\alpha, D_\alpha = k_B T \Gamma_\alpha$のような関係にある. イオンの化学ポテンシャルは

$$\mu_\alpha = k_B T \ln C^*_\alpha + z_\alpha e(\Psi - \boldsymbol{E} \cdot \boldsymbol{r}) \tag{5.5}$$

とする [28]. $\boldsymbol{E}$は外部電場をあらわし, 静電ポテンシャル$\Psi(\boldsymbol{r})$は Poisson方程式:

$$\epsilon \nabla^2 \Psi = -\rho_e \tag{5.6}$$

を解いて得られる. この化学ポテンシャルにしたがうイオンは，平衡状態において Poisson–Boltzmann分布となる.

5.2.2　Navier–Stokes方程式

溶媒流動は非圧縮性の流れ($\nabla \cdot \boldsymbol{u} = 0$) である. 溶媒の速度場$\boldsymbol{u}$は Navier–Stokes方程式:

$$\rho(\partial_t + \boldsymbol{u} \cdot \nabla)\boldsymbol{u} = -\nabla p + \eta \nabla^2 \boldsymbol{u} - \rho_e(\nabla\Psi - \boldsymbol{E}) + \phi \boldsymbol{f}_p \tag{5.7}$$

にしたがう. ρ, η, pはそれぞれ溶媒の密度, 粘性係数, 圧力をあらわす. ここで流体には静電気力$-\rho_e(\nabla\Psi - \boldsymbol{E})$が働いていることに注意されたい. 外力項$\phi \boldsymbol{f}_p$は, これは粒子の剛体性を保証するための強制力をあらわしている. すなわち$\phi \boldsymbol{f}_p$によって粒子表面の粘着境界条件が考慮される. 詳しくは文献 [5, 6]を参照のこと.

5.2.3　運動方程式

質量M_pをもつi番目コロイド粒子の位置および速度$\{\boldsymbol{R}_i, \boldsymbol{V}_i\}$ は運動方程式:

$$\dot{\boldsymbol{R}}_i = \boldsymbol{V}_i, \tag{5.8}$$

$$M_p \dot{\boldsymbol{V}}_i = \boldsymbol{F}^H_i + \boldsymbol{F}^{other}_i \tag{5.9}$$

によって時間発展する. ここで$\boldsymbol{F}^H_i$は流体から受ける力をあらわし, 固液間での運動量収支を表している [6]. また$\boldsymbol{F}^{other}_i$は Lennard–Jonesポテンシャルなど粒子間のポテンシャルによる力である. ここでは省略したが粒子の回転運動も同様に考慮されている [6]. 以上がシミュレーションに用いられる基礎方程式である.

5.3　電気二重層の性質

電気二重層の構造を定量的に扱うための基礎は Poisson–Boltzmann方程式である.

5.3.1　Poisson–Boltzmann方程式

外部電場がない($\boldsymbol{E}=0$)とき, 式(5.5)のもとでイオンの平衡分布を求める. 化学ポテンシャルが一様, $\mu_\alpha = \mathrm{c}st.$, になったとき, 平衡イオン分布として

$$C^*_\alpha(\boldsymbol{r}) = \bar{C}_\alpha \exp\left(-\frac{z_\alpha e\Psi(\boldsymbol{r})}{k_BT}\right) \tag{5.10}$$

を得る. これは静電ポテンシャルΨのもとでの Boltzmann分布である. これは式(5.6)とあわせてPoisson–Boltzmann方程式とよばれる.

5.3.2　Debye–Hückel近似とDebye遮蔽長

$z:z$対称電解質溶媒中にある1個の球状コロイド粒子を考える. Poisson–Boltzmann方程式は

$$\nabla^2\Psi(\boldsymbol{r}) = \frac{2ze\bar{C}}{\epsilon}\sinh\left(\frac{ze\Psi(\boldsymbol{r})}{k_BT}\right), \tag{5.11}$$

となる [15, 29]. 無限遠方での境界条件は$\Psi|_{\boldsymbol{r}=\infty}=0$ および$C^*|_{\boldsymbol{r}=\infty}=\bar{C}$である. 粒子表面における境界条件は

$$\nabla\Psi|_{\mathrm{surface}} = -\frac{\sigma e}{\epsilon}, \tag{5.12}$$

であり, ここで表面電荷密度$\sigma e = Ze/4\pi a^2$である. すなわち表面電荷一定の境界条件を考えている. $ze\Psi/k_BT \ll 1$を仮定し式(5.11)を線形化する近似は Debye–Hückel近似と呼ばれる.

$$\nabla^2\Psi(\boldsymbol{r}) = \frac{2z^2e^2\bar{C}}{k_BT\epsilon}\Psi = \kappa^2\Psi, \tag{5.13}$$

となる. このとき、長さの次元をもつ定数

$$\kappa^{-1} = \frac{1}{\sqrt{8\pi\lambda_B z^2\bar{C}}}, \tag{5.14}$$

はDebye 遮蔽長とよばれる. ここで$\lambda_B = e^2/4\pi k_BT\epsilon$はビヨルン(Bjerrum)長である. 一般の電解質の場合は,

$$\kappa^{-1} = \frac{1}{\sqrt{4\pi\lambda_B\sum_\alpha z_\alpha^2\bar{C}_\alpha}}, \tag{5.15}$$

である. 今, 系は球対称なので動径方向$r=|\boldsymbol{r}|$についてのみ考えればよく,

$$\frac{d^2\Psi}{dr^2} + \frac{2}{r}\frac{d\Psi}{dr} = \kappa^2\Psi \tag{5.16}$$

という式に帰着する. この一般解はYukawa型ポテンシャル

$$\Psi(r) = \Psi_0\frac{a}{r}\exp[-\kappa(r-a)], \tag{5.17}$$

である. コロイド電荷による静電力はκ^{-1}の程度で遮蔽される. κ^{-1}は, 表面電荷に異符号イオンが引き寄せられるクーロン力の効果と, イオンの局在をかき消そうとする熱拡散のつり合う距離と考えることができる. したがってDebye遮蔽長κ^{-1}は電気二重層の厚さとみなすことが出来る. 温度が高く熱エネルギーk_BTが大きい場合κは大きくなる. またイオン強度($\sum_\alpha z_\alpha^2\bar{C}_\alpha/2$)が大きいほど, イオンによる遮蔽効果が大きくなり κは小さくなる. た

とえば$z:z$対称電解質溶媒中をとして, バルク塩濃度$\bar{C}$を考えるとする. 媒質を25 °Cの水とするとビヨルン長は$\lambda_B = 0.72$nmとなり, 実際の数値を入れると式(5.14)は

$$\kappa^{-1} = \frac{0.3}{z\sqrt{\bar{C}}}\ (\mathrm{nm}) \tag{5.18}$$

となる. $z = 1, \bar{C} = 0.1$Mでは$\kappa^{-1} = 1$nm, $z = 1, \bar{C} = 0.001$Mでは$\kappa^{-1} = 10$nm となっている.

表面電位$\Psi(r = a) = \Psi_0$は境界条件$d\Psi/dr(r = a) = -\sigma e/\epsilon$から定まる. Ψと表面電荷密度σeの関係は,

$$\sigma e = \epsilon\kappa\Psi_0(1 + (\kappa a)^{-1}), \tag{5.19}$$

となる. すなわち表面電荷の増加に対して、 表面電位が線型に増加する関係を与える. 表面電位が大きいところでは Debye–Hückel近似は適用できない. 実際は表面電荷の増加に対して表面電位はしだいに増加しにくくなる. このときの関係式は Poisson–Boltzmann方程式を解くことによって得られるが, 1:1電解質溶媒に対する近似式が Loeb–Overbeek–Wiersema [15]あるいは Ohshima–Healy–White [29, 30] によって提案されている.

5.4　電気泳動の原理

外部から電場Eをかけると, Zeに帯電したコロイド粒子は静電気力ZeEを受けて動き出す. 粒子は静電気力によって加速するが流体から粘性抵抗力を受け, 両者がつり合ったところで定常速度Vに達し等速運動をする. 粘性抵抗力を半径aの球状粒子に対する$6\pi\eta aV$のStokes抵抗と仮定すると, 力のつりあいは,

$$ZeE = 6\pi\eta aV \tag{5.20}$$

と書け, 電気泳動易動度は

$$\frac{V}{E} = \frac{Ze}{6\pi\eta a} \tag{5.21}$$

となる. しかしこれは適当ではなく, 実際の電気泳動易動度はこの値よりも小さくなる. これはコロイド周囲のイオン雰囲気にも静電力が働くことで発生するイオン雰囲気の運動および静電力が考慮されていないことによる. コロイドの泳動はイオン雰囲気の運動も伴うので、 泳動速度はその分遅くなる. さらに電気二重層が球対称から歪めばそれによる力も考慮しなくてはならなく, 理論的な解析は極めて複雑なものとなる. そこで簡単なモデルについて電気泳動度に対する解析がいくつかなされている [15, 29].

5.4.1　Smoluchowskiの式

粒子半径aが電気二重層の厚さκ^{-1}よりも非常に大きい$\kappa a \gg 1$のときSmoluchowskiの式を適用することができる. この場合, 電気二重層厚さが無限に薄い極限となるので, 粒子表面の曲率を無視して, 粒子表面を平板とみなすことができる. 平板に平行な(x方向とする)外部電場E_xが加わっているとする. 粒子に固定した座標系を考えて, 粒子の上の乗って液体の動きを見ると, 粒子から無限に離れたところの流体は$-V$の速度をもつことになる. 粘性抵抗力と静電気力のつりあいを考えると,

$$\eta\frac{\partial^2 v_x}{\partial y^2} + \sum_\alpha eC_\alpha E_x = 0 \tag{5.22}$$

となる. Poisson方程式によってイオン分布は静電ポテンシャルの2階微分と関係しているので,

$$\eta\frac{\partial^2 v_x}{\partial y^2} = \epsilon\frac{\partial^2\Psi}{\partial y^2}E_x \tag{5.23}$$

となる. 無限遠方で速度勾配, ポテンシャル勾配, ポテンシャルそれぞれ0という境界条件のもと積分して

$$\eta[v_x(y) + V] - \epsilon E\Psi(y) = 0 \tag{5.24}$$

となる. すると粒子表面($y = 0$)で速度0であるので,

$$\frac{V}{E} = \frac{\epsilon\zeta}{\eta} \tag{5.25}$$

というSmoluchowskiの式を得ることができる. ここでゼータ電位ζは本来はすべり面での電位として定義されるのだが粒子表面における電位$\Psi(0)$と置き換えられている.

5.4.2　Hückelの式

Smoluchowski の状況とは逆に, 粒子が電気二重層の厚さに対して非常に小さい $\kappa a \ll 1$の極限を考える. つまりZeが点電荷の極限の場合は Hückelの式を適用することができる. 式(5.21)において, 粒子表面のポテンシャルをクーロンポテンシャル

$$\zeta = \frac{Ze}{4\pi\epsilon a} \tag{5.26}$$

で与えると

$$\frac{V}{E} = \frac{2}{3}\frac{\epsilon\zeta}{\eta} \tag{5.27}$$

というHückelの式が得られる.

5.4.3　Henryの式およびO'Brien–Whiteによる解析

式(5.25)と式(5.27)を結び付けるものとして一般のκaに対してHenryの式

$$\frac{V}{E} = f(\kappa a)\frac{\epsilon\zeta}{\eta} \tag{5.28}$$

が知られている. $f(\kappa a)$はHenry係数といわれ

$$f(\kappa a) = 1 - 5\exp(\kappa a)E_7(\kappa a) + 2\exp(\kappa a)E_5(\kappa a) \tag{5.29}$$

$$= \frac{2}{3} + \frac{(\kappa a)^2}{24} - \frac{5(\kappa a)^3}{72} - \frac{(\kappa a)^4}{144} + \frac{(\kappa a)^5}{144} + \left[\frac{(\kappa a)^4}{12} - \frac{(\kappa a)^6}{144}\right]\exp(\kappa a)E_1(\kappa a) \tag{5.30}$$

で定義される. ここで$E_n(\kappa a)$はn次積分指数関数である. Smoluchowskiの式は$f = 1(\kappa a \to \infty)$, Hückelの式は$f = 2/3(\kappa a \to 0)$に対応する(Fig. 5.1).

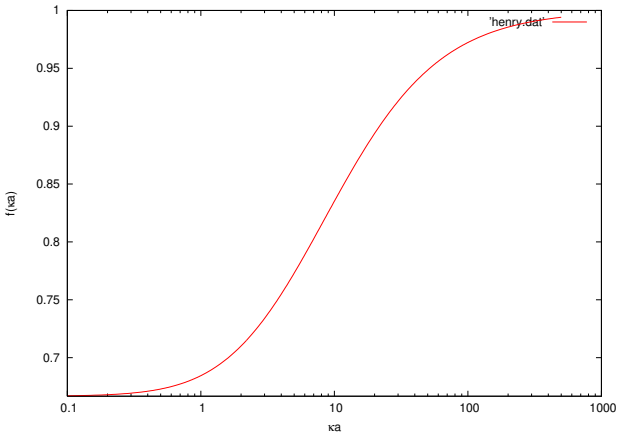


図5.1: Henry係数$f(\kappa a)$.

Henryの式(5.28)はゼータ電位に対して線形な関係式であり, ゼータ電位が低い場合にのみ適用できる. ゼータ電位が高くなると電気二重層の変形の効果(緩和効果とよばれる)を考慮しなければならない. O'Brien–Whiteは一般

のκa, ζに対して電気泳動度とゼータ電位の関係を数値解析によって提案している [31]. その後, Ohshima–Healy–Whiteによる$\kappa a \geq 10$に対する解析式が提案されている [32].

5.5　入力UDF について

5.5.1　流体の設定

constitutive_eqとしてElectrolyteを選ぶと荷電コロイドの運動や, 外部電場の下での電気泳動現象をシミュレートすることができる [*1]. 以下, 入力UDFファイルで指定する変数について解説する.

constitutive_eq.Electrolyte以下には溶媒とイオン分布の情報を入れる.

- constitutive_eq.Electrolyte.DX... 長さの単位量である格子幅Δ.
- constitutive_eq.Electrolyte.RHO... 溶媒の密度.
- constitutive_eq.Electrolyte.ETA... 溶媒の粘度.
- constitutive_eq.Electrolyte.kBT... 粒子温度.
- constitutive_eq.Electrolyte.alpha_v... 粒子の熱ゆらぎ（並進運動）に対する補正項.
- constitutive_eq.Electrolyte.alpha_o... 粒子の熱ゆらぎ（回転運動）に対する補正項.
- constitutive_eq.Electrolyte.Dielectric_cst... 溶媒の誘電率.
- constitutive_eq.Electrolyte.Init_profile... イオン分布の初期設定についてUniformかPoisson_Boltzmannを選ぶ. [*2]
- constitutive_eq.Electrolyte.Add_salt.type... saltfree (粒子表面とは反対符号のイオン1種類のイオンを考慮する場合) または salt (正負2種類のイオンを考慮する) を選択する.
  - constitutive_eq.Electrolyte.Add_salt.saltfree.Valency_counterion... 対イオンの価数.
  - constitutive_eq.Electrolyte.Add_salt.saltfree.Onsager_coeff_counterion... 対イオンのOnsager輸送係数.
  - constitutive_eq.Electrolyte.Add_salt.salt.Valency_positive_ion... 正イオンの価数.
  - constitutive_eq.Electrolyte.Add_salt.salt.Valency_negative_ion... 負イオンの価数.
  - constitutive_eq.Electrolyte.Add_salt.salt.Onsager_coeff_positive_ion... 正イオンのOnsager輸送係数.
  - constitutive_eq.Electrolyte.Add_salt.salt.Onsager_coeff_negative_ion... 負イオンのOnsager輸送係数.
  - constitutive_eq.Electrolyte.Add_salt.salt.Debye_length... Debye遮蔽長を指定すると対応する塩濃度が設定される.
- constitutive_eq.Electrolyte.Electric_field.type... 外部電場のON, OFFを選択する.
  - constitutive_eq.Electrolyte.Electric_field.ON.type... DC(直流電場)かAC(交流電場)を選択する.
    - constitutive_eq.Electrolyte.Electric_field.ON.DC.Ex... x方向電場の強さ. 交流電場の場合も同じ.
    - constitutive_eq.Electrolyte.Electric_field.ON.DC.Ey... y方向電場の強さ. 交流電場の場合も同じ.
    - constitutive_eq.Electrolyte.Electric_field.ON.DC.Ez... z方向電場の強さ. 交流電場の場合も同じ.
    - constitutive_eq.Electrolyte.Electric_field.ON.AC.Frequency... 交流電場の周波数

5.5.2　オブジェクト（粒子）の設定

object_type.typeでは，spherical_particle(球状粒子)，chain(フレキシブル鎖)，rigid(剛体)を選択できる.

[*1] constitutive_eqの他のswitchとして, Newton流体中のコロイド粒子の運動はNavier_Stokesを, シア流下におけるNewton流体中のコロイド粒子の運動はShear_Navier_Stokesを選べばシミュレートすることができる. 荷電粒子の熱揺らぎは考慮されていない

[*2] Poisson_Boltzmannを選ぶと, 特に多粒子のときに時間がかかることがあるので注意する.

5.5.3　空間単位と時間単位

長さの単位としてを格子幅Δを採用している. 時間の単位τ_0ついて流体の密度ρと粘性率ηと格子幅で決まる$\tau_0 = \rho\Delta^2/\eta$を採用している.

- Navier–Stokes方程式で$\rho = \eta = \Delta = 1$となるような単位系を採用している.
- 仮に入力udfファイルでRHO= A, ETA= B, DX= Cと入力した場合, 最大波数k_{max}はCを用いて与えられ, 時間刻みの上限は運動量の拡散時間より$T_{dump} = (A/B)/k_{max}^2$ と与えられる. 時間刻みΔtは$T_{dump} \times$ `factor`で調整する.
- 現実との対応を考察する．考察したい空間スケールとして，グリッドサイズを$1\,\mu$mの長さを考える．また溶媒として水($\eta = 1 \times 10^{-3}\,\mathrm{Pa\,s}$，$\rho = 1 \times 10^{3}\,\mathrm{kg\,m^{-3}}$)を対象とした場合，時間の単位は$\tau_0 = 1 \times 10^{-6}\,\mathrm{s}$となる.
- `constitutive_eq=Electrolyte`の場合、$(A/B)/k_{max}^2$と$(1/k_BT\Gamma_\alpha)/k_{max}^2$の小さな方を$T_{dump}$とする. 実際のシミュレーションでどちらが用いられたかは，KAPSEL実行時のコマンドライン（標準エラー）出力で確認できる.

5.6　計算事例

5.6.1　1粒子の電気泳動

1粒子の電気泳動シミュレーションの計算例は, 入力UDFとして，`Examples/01/`フォルダの `colloid_1.udf`を用いればよい. 具体的には`./avs_ch/`と`./avs_ch/avs/` が作成されていることを確認した上で以下を実行する.

```
$ ../../kapsel -Icolloid_1.udf -Ooutput.udf -Ddefine.udf -Rrestart.udf
```

この例ではメッシュサイズ$64 \times 64 \times 64$で 1:1電解質溶媒(粘性係数$\eta = 1$, 密度$\rho = 1$)を考え, 半径$a = 5$, 界面厚さ$\xi = 2$, 帯電量$Z = -100$, 電場$E_x = 0.1$, Debye遮蔽長$\kappa^{-1} = 10$で計算している. Fig. 5.2には1粒子電気泳動のスナップショットを示している． Fig. 5.3には泳動速度の時間発展を示した. 静電気力と流体抵抗力がつりあうところで定常速度を実現していることがわかる. Fig. 5.3のプロットには，`plot.py` を用いている.

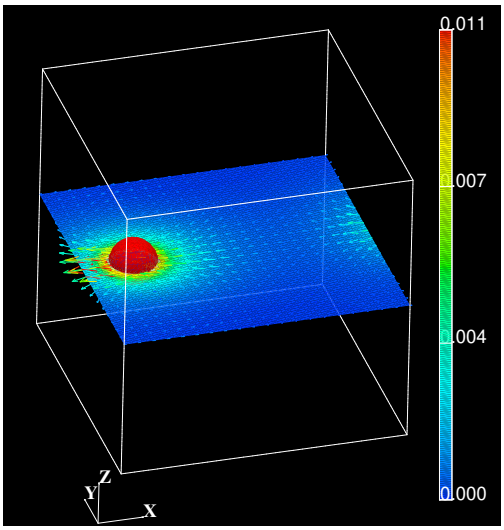


図5.2: $+x$方向に印加された外部電場Eによって電気泳動するコロイド粒子(赤色)の様子. 濃淡は電荷密度分布を, 矢印は溶媒速度場をあらわす.

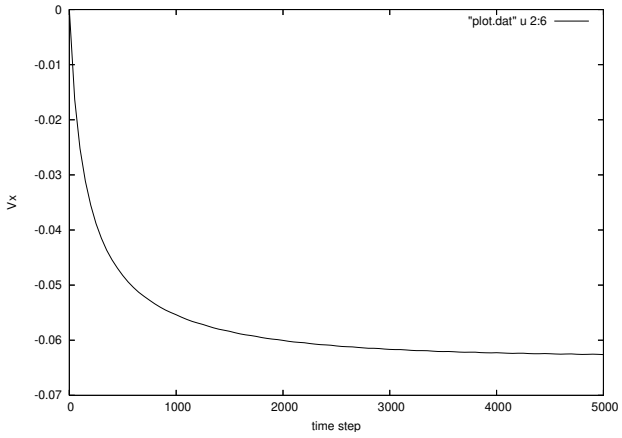


図5.3: 1 個のコロイド粒子の泳動速度の時間発展.

5.6.2　多粒子の電気泳動

粒子数を32個とした電気泳動シミュレーションは，Examples/01/フォルダのcolloid_32.udfで計算できる．初期配置は，ランダムであり，その他のパラメータは1粒子と同じである．具体的には./avs_ch/と./avs_ch/avs/が作成されていることを確認した上で以下を実行する．

```
$ ../../kapsel -Icolloid_32.udf -Ooutput.udf -Ddefine.udf -Rrestart.udf
```

Fig. 5.4には多粒子系での電気泳動のスナップショットを示している． Fig. 5.5には，平均泳動速度の時間発展を示している．この図では，粒子数 $N_P = 16, 32, 64$の場合のデータがプロットされている．粒子数が増加するにつれて粒子の平均泳動速度が減少することがわかる．

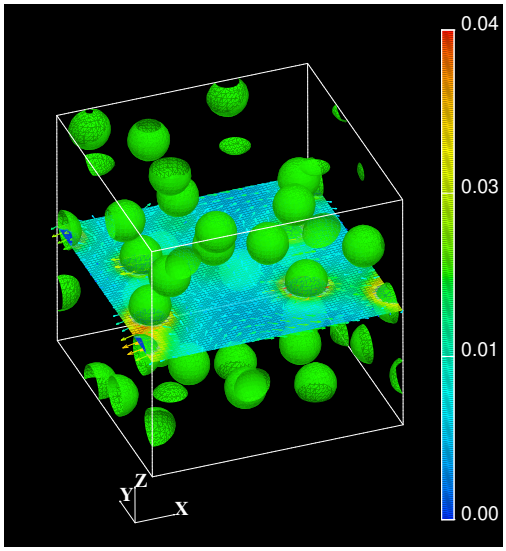


図5.4: $+x$方向に印加された外部電場Eによって電気泳動するコロイド粒子(緑色)の様子. 濃淡は電荷密度分布を, 矢印は溶媒速度場をあらわす.

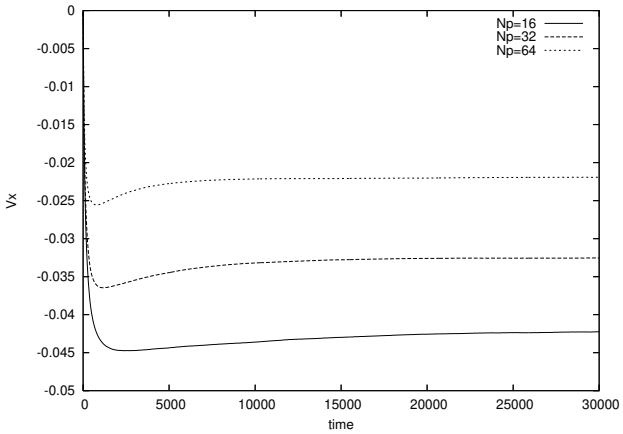


図5.5: 多粒子系での平均泳動速度の時間発展. 上から粒子数 $N_p = 64, 32, 16$のデータである.

5.6.3　正負電荷の粒子混合系の電気泳動

正に帯電したコロイド粒子を32個, 負に帯電したコロイド粒子を32個の混合系の電気泳動は，Examples/01/フォルダの colloid_p32m32.udf で計算できる．その他のパラメータは1粒子と同じである．具体的には./avs_ch/と./avs_ch/avs/ が作成されていることを確認した上で以下を実行する．

```
$ ../../kapsel -Icolloid_p32m32.udf -Ooutput.udf -Ddefine.udf -Rrestart.udf
```

Fig. 5.6には，正に帯電したコロイド粒子を32個, 負に帯電したコロイド粒子を32個の混合系の電気泳動のスナップショットを表している．

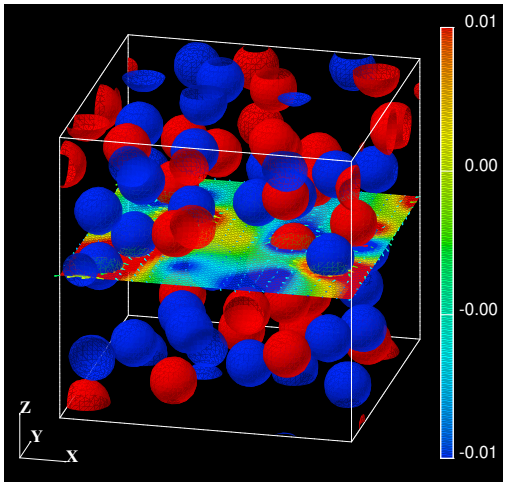


図5.6: $+x$方向に印加された外部電場Eによって電気泳動する正に帯電したコロイド粒子(赤色)と負に帯電したコロイド粒子(青色)の様子. 正符号コロイドは$+x$方向に, 負符号コロイドは$-x$方向に泳動している．

一方，colloid_p5km5k.udfでは，正に帯電したコロイド粒子を5000個，負に帯電したコロイド粒子を5000個の混合系の電気泳動を計算できる．その他のパラメータは1粒子と同じである．

```
$ ../../kapsel -Icolloid_p5km5k.udf -Ooutput.udf -Ddefine.udf -Rrestart.udf
```

Fig. 5.7には，正に帯電したコロイド粒子を5000個，負に帯電したコロイド粒子を5000個の混合系の電気泳動のスナップショットを表している．

5.6.4　AVS/Expressによる可視化

output.AVSをONとしてAVS形式のファイルを出力する．AVS/Express[*3]でavs_charge.vファイルを読み込みRead_fieldモジュールにフィールドファイルdata.fldを指定すると粒子, 溶媒速度場, 電荷分布をそれぞれ可視化することができる．Fig. 5.2, Fig. 5.4, Fig. 5.6は avs_charge.vを用いて可視化している．

5.6.5　Gourmetによる可視化

Gourmetを起動してoutput.udfを読み込み，GourmetのPythonパネルで KAPSELに付属のPythonスクリプトshow_field.pyをLoad してからRun するとグラフィックウインドウが開き, 粒子, 溶媒速度場, 電荷密度分

[*3] http://www.avs.com/

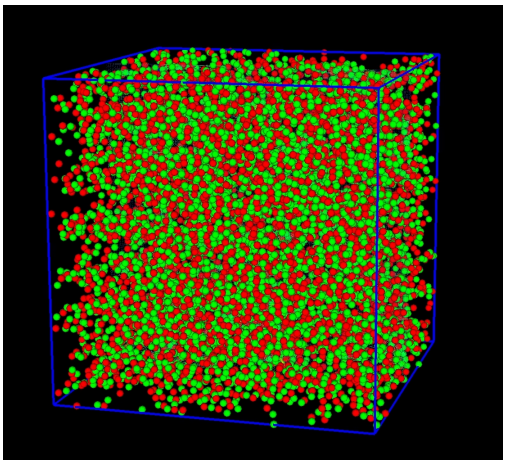


図5.7: $+x$方向に印加された外部電場Eによって電気泳動する正に帯電したコロイド粒子(赤色)と負に帯電したコロイド粒子(緑色)の様子. 正符号コロイドは$+x$方向に, 負符号コロイドは$-x$方向に泳動している.

布の可視化が行われる[*4]. そして, グラフィックウインドウの最下段にある再生ボタンをクリックすればアニメーションがスタートする. 粒子は出力UDFにレコードデータとして出力される粒子座標データをもとに可視化しているので, output.UDFを`ON`とする必要がある. さらに溶媒速度場と電荷密度分布はAVS形式で出力されるBinaryファイルを読み込んでいるので, output.AVSを`ON`としてさらに output.AVS.ON.FileTypeを`Binary`にする必要がある. 粒子の運動のみを表示したい場合は，`particleshow.py`を用い，output.UDFを`ON`とする必要がある.

5.6.6　gnuplotによるグラフ化

UDFに出力されたレコードデータから粒子の位置, 速度の時系列変化を gnuplot[*5]を用いてグラフ化することができる [33]. Gourmetを起動して`output.udf`を読み込み, GourmetのPythonパネルで KAPSELに付属のPythonスクリプト`plot.py`を`Load` してから`Run`すると, 粒子の位置, 速度を格納しているレコードデータからグラフ描画用シートを作成することができる. データをプロットするためには“View”ボックスで`Tree`についているチェックを`Table`に変更し, 変数の一覧に`GraphSheet`があるので選択して`GraphSheet`のデータをブラウズする. そして, GourmetのPlotパネルで`Make`してから`Plot`すると Fig. 5.3のようなグラフを描くことができる. 線種が多すぎて見づらいときはPlotパネルのエディター中に作成されたplotコマンドから不要な部分を削除すればよい. Gourmetからgnuplotを呼び出す方法は文献 [33]の第3章に詳しく記述されている.

[*4] Gourmet_2003以前のバージョンには不具合があり, このスクリプトが正常に動作しないことが確認されている. その場合は代りに付属のPythonスクリプト`particleshow.py`を用いて可視化する. 溶媒速度場や電荷密度分布は表示されず, 粒子のみが表示される.

[*5] http://www.gnuplot.info/

第6章

二成分相分離流体に分散した粒子のシミュレーション

6.1 粒子が添加された二成分相分離流体

非相溶な二成分流体が示す多様な相分離構造の制御は，工業的応用の観点において極めて重要である．例えば，一方の成分がμm以下のオーダーの大きさのドロップレットとして，もう一方の流体内に分散しているエマルションは，親和性の低い二つの流体が共存する性質を生かして，食品や化粧品，石油回収等，幅広い分野で利用されている．しかし，エマルションは元来熱力学的に不安定であることから，その安定化や構造制御が応用上課題となっていた． 1900年代初頭，Ramsden[34]とPickering[35]は，不溶な微粒子を添加することでエマルションを安定化できることを報告した．これは，微粒子が二つの流体間の界面に吸着することで界面活性剤のような働きを示すことによる． RamsdenとPickeringの報告以来，シリカやカーボンブラック，金属錯体といった様々な微粒子をエマルションに添加する研究が盛んに行われてきた[36, 37]．また近年では，共連続構造を示す二成分相分離流体界面を微粒子で安定化することで得られるBijelと呼ばれる新しい複合材料も注目を集めている[38, 39]．このような微粒子添加による相分離構造制御では，流体–流体間の相互作用に加えて，流体–粒子間の相互作用や流体界面–粒子間の親和性が重要な因子となる． KAPSELではそれらの相互作用を考慮した二成分相分離流体に添加された粒子をシミュレーションする機能が実装されており，上記したような粒子添加による相分離構造やレオロジー的性質への影響を調べることができる．

KAPSELで用いられるSmoothed Profile(SP)法で一成分流体中の粒子を扱う場合は，粒子–流体間の界面のみを考慮すればよかったが，二成分流体では新たに流体–流体間界面を表現する必要がある．そこで，二成分相分離流体に分散した粒子のシミュレーションでは，新たに二成分相分離流体を区別する界面関数ψを導入し，その時間発展を記述する(図6.1)． KAPSELでは，二成分相分離流体のダイナミクスを記述するモデルとして良く知られているmodel H[40]をSP法と組み合わせることで，固体粒子を含む二成分相分離流体を扱う．

6.2節に，二成分相分離流体を扱うために拡張されたSP法について概説する． 6.3節では，KAPSELで使用するシミュレーションパラメータについて説明する． 6.4節では，本機能を用いた解析事例として，二成分相分離流体に分散する固体粒子とPickeringエマルションのシミュレーションを紹介する．

6.2 理論的背景と基礎方程式

6.2.1 二成分相分離流体の基礎方程式

model H[40]では，二成分相分離流体の運動方程式は，化学ポテンシャル勾配項を導入したNavier–Stokes(NS)方程式で記述される:

$$\frac{\partial \boldsymbol{u}}{\partial t} + (\boldsymbol{u}\cdot\boldsymbol{\nabla})\boldsymbol{u} = -\frac{1}{\rho}\boldsymbol{\nabla} p + \frac{1}{\rho}\boldsymbol{\nabla}\cdot\eta(\boldsymbol{\nabla}\boldsymbol{u} + \boldsymbol{\nabla}\boldsymbol{u}^{\mathrm{t}}) - \frac{\psi}{\rho}\boldsymbol{\nabla}\mu_{\psi} - \frac{\phi}{\rho}\boldsymbol{\nabla}\mu_{\phi} + \phi\boldsymbol{f}_{\mathrm{p}}. \tag{6.1}$$

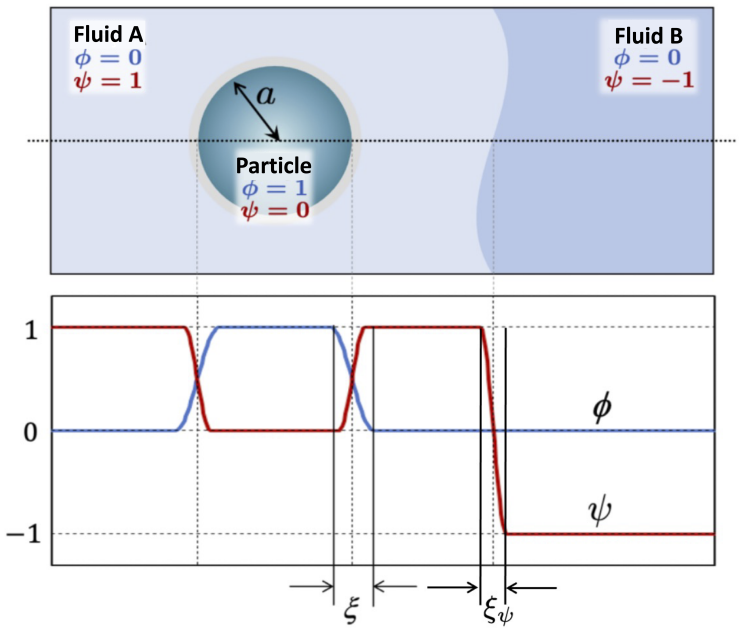


図6.1: 二成分相分離流体に分散した粒子を記述する界面関数．流体と粒子を識別するSP関数ϕに加えて，二成分流体を区別する界面関数ψが導入される．粒子–流体間界面の界面厚みはξ，流体–流体間界面の界面厚みはξ_ψで与えられる．図では，各流体のバルクのψの値が1及び−1として記されているが，実際は二重井戸型ポテンシャル$f(\psi)$の設定に依存する．

ここで，ηは粘度であり，二成分流体では場所に依存する物理量として扱う．また，上付き添字のtは，転置行列を表す．μ_ψとμ_ϕは，それぞれψとϕに対する局所的な化学ポテンシャルであり，Ginzburg–Landau(GL)自由エネルギーFの汎関数微分により与えられる: $\mu_\psi = \delta F/\delta\psi$，$\mu_\phi = \delta F/\delta\phi$．KAPSELで用いる$F$の詳細は後述する．

また，流体界面関数ψの時間発展は，Cahn–Hilliard(CH)方程式によって記述される:

$$\frac{\partial\psi}{\partial t} = -\nabla\cdot\boldsymbol{J}. \tag{6.2}$$

ここで，$\boldsymbol{J}$は物質流束であり，ここでは移流項を含む形式

$$\boldsymbol{J} = \psi\boldsymbol{u} - \kappa\nabla\mu_\psi \tag{6.3}$$

で表される．また，κは二成分流体の易動度である．同じ記号を用いるが，第5章で用いたデバイ長κ^{-1}とは無関係である．

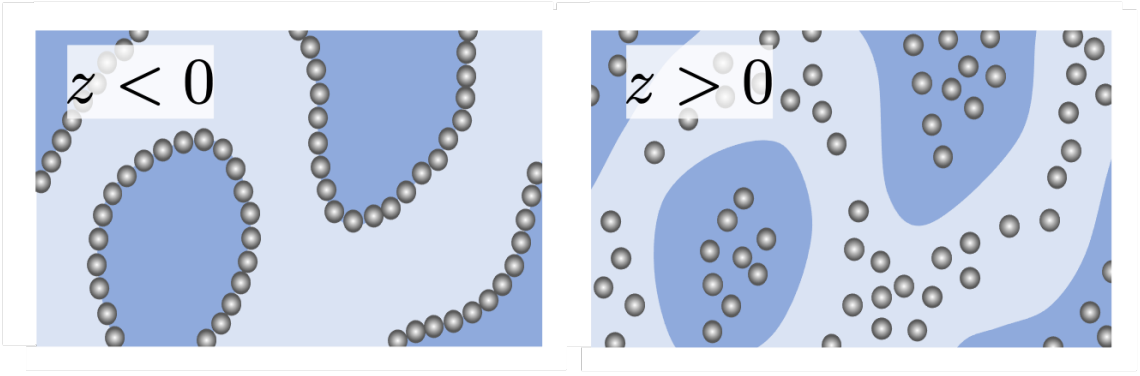


図6.2: パラメータzによる粒子分散の変化.

次に，GL自由エネルギーFについて説明する． KAPSELでは，粒子と相互作用する二成分流体の相分離を扱うため，Fは次のように与えられる:

$$F = \int d\boldsymbol{r} \left[f(\psi) + \frac{\alpha}{2}(\boldsymbol{\nabla}\psi)^2 + w\xi\psi|\boldsymbol{\nabla}\phi|^2 + d(\psi - \bar{\psi})^2\phi + z\xi_\psi\phi(\nabla\psi)^2 \right]. \tag{6.4}$$

関数$f(\psi)$は，流体界面関数ψのローカルなダイナミクスを支配する二重井戸型ポテンシャルである． KAPSELでは，Landau型ポテンシャル

$$f(\psi) = \frac{a}{4}\psi^4 - \frac{b}{2}\psi^2 \tag{6.5}$$

や，Flory–Huggins(FH)型ポテンシャル

$$f(\psi) = k_\mathrm{B}T_0 \left[\frac{\psi}{N_\mathrm{A}} \ln\psi + \frac{(1-\psi)}{N_\mathrm{B}} \ln(1-\psi) + \chi\psi(1-\psi) \right] \tag{6.6}$$

が利用できる．ここで，式(6.5)中のaとbは，各項の大小を表すパラメータである．また，式(6.6)の$k_\mathrm{B}T_0$は基準となる熱エネルギー単位，N_AとN_Bは二成分流体の各成分の重合度であり，χはFloryの相互作用パラメータである．

式(6.4)の第2項は，二成分流体間の界面エネルギー項を表す．また，第3項は粒子–流体間相互作用を表し，第4項は$\bar{\psi} = \frac{a\psi_0^3 - (b-2d)\psi_0}{2d}$を通して粒子内部の$\psi$を任意の値$\psi_0$に固定するための項である[*1]．第5項は粒子と二成分流体間の界面との相互作用を表す項であり，zは係数，ξ_ψは二成分流体間の界面厚さを表す[*2]．図6.2にパラメータzによる粒子分散の変化の模式図を示す． $z < 0$の場合，Fは粒子が流体–流体間界面上に存在するときに最も低くなるため，粒子は流体–流体間界面上へと移動する．一方，$z > 0$の場合，粒子は流体間界面から遠ざかるようになる．また，ξ_ψは関数$f(\psi)$に依存する量であるが，Landauの二重井戸型ポテンシャルの場合，

$$\xi_\psi = \sqrt{\frac{\alpha}{2b}} \tag{6.7}$$

で表される[41].

粒子を含む二成分相分離流体のシミュレーションでは，流体と粒子の運動方程式(式(6.1)， (3.4)–(3.6))とともに，さらに二成分流体を区別する流体界面関数ψの時間発展方程式である式(6.2)を逐次的に解くこととなる．KAPSELでは，これらの方程式を，semi-staggered Arakawa B 格子[42]上で，Marker And Cell (MAC) 法[43, 44]を用いて解いている．また，第4章に示す手法と組み合わせることで，二成分相分離流体に分散した粒子のシミュレーションをせん断流下でも行うことができる．解法の詳細は付録Dに示す．

6.2.2　懸濁液の粘度

単成分流体に粒子が添加された懸濁液のせん断応力と粘度は，式(4.5)，(4.6)で得られる．一方，二成分相分離流体に粒子が添加された系では，流体–流体界面における表面張力項の寄与を考慮する必要がある．この流体界面による応力テンソルのxy成分は，

$$\sigma_\mathrm{int}^{xy} = \frac{1}{V} \int d\boldsymbol{r} \left(-\alpha \frac{\partial\psi}{\partial x} \frac{\partial\psi}{\partial y} \right) \tag{6.8}$$

で表される[45]．従って，この系のみかけの粘度は，式(4.6)に式(6.8)の寄与を加えることで得られる． 6.4.2節で，定常せん断流下での粘度の計算例を示す．

[*1] d=0としてはいけない．この項を無視したければd=0.0001などとする．

[*2] KAPSELでは関数$f(\psi)$においてFlory–Huggins型ポテンシャルを設定するとき，この項は無効化される．

6.3　入力UDFについて

6.3.1　流体の設定

KAPSELでは，二成分相分離流体に分散した粒子のシミュレーションを，初期流れ場なし，あるいはせん断流下で行うことができる．それぞれ，constitutive_eqにおいて，Navier_Stokes_Cahn_Hilliard_FDMとShear_NS_LE_CH_FDMを選ぶとシミュレーションできる．

初期流れ場なしのシミュレーション設定

constitutive_eq.Navier_Stokes_Cahn_Hilliard_FDM以下で二成分相分離流体の情報を入れる．

- constitutive_eq.Navier_Stokes_Cahn_Hilliard_FDM.NS_solver.type…NS方程式の解法をexplicit_scheme(陽解法)あるいはimplicit_scheme (陰解法)から選択する[*3].
- constitutive_eq.Navier_Stokes_Cahn_Hilliard_FDM.NS_solver.implicit_scheme.tolerance…NS方程式の陰解法の収束判定基準.
- constitutive_eq.Navier_Stokes_Cahn_Hilliard_FDM.NS_solver.implicit_scheme.maximum_iteration…NS方程式の陰解法の最大反復回数.
- constitutive_eq.Navier_Stokes_Cahn_Hilliard_FDM.NS_solver.implicit_scheme.viscosity_change…ONを選ぶと粘度の異なる二成分相分離流体(A/B)のシミュレーションができる.
- constitutive_eq.Navier_Stokes_Cahn_Hilliard_FDM.NS_solver.implicit_scheme.ON.ETA_A…A成分流体の粘度.
- constitutive_eq.Navier_Stokes_Cahn_Hilliard_FDM.NS_solver.implicit_scheme.ON.ETA_B…B成分流体の粘度.
- constitutive_eq.Navier_Stokes_Cahn_Hilliard_FDM.CH_solver.type…CH方程式の解法をexplicit_scheme(陽解法)あるいはimplicit_scheme(陰解法)から選択する[*4].
- constitutive_eq.Navier_Stokes_Cahn_Hilliard_FDM.CH_solver.implicit_scheme.tolerance…CH方程式の陰解法の収束判定基準.
- constitutive_eq.Navier_Stokes_Cahn_Hilliard_FDM.CH_solver.implicit_scheme.maximum_iteration…CH方程式の陰解法の最大反復回数.
- constitutive_eq.Navier_Stokes_Cahn_Hilliard_FDM.DX…長さの単位量である格子幅Δ.
- constitutive_eq.Navier_Stokes_Cahn_Hilliard_FDM.RHO…流体の密度ρ.
- constitutive_eq.Navier_Stokes_Cahn_Hilliard_FDM.ETA…流体の粘度η[*5].
- constitutive_eq.Navier_Stokes_Cahn_Hilliard_FDM.kBT…粒子温度.
- constitutive_eq.Navier_Stokes_Cahn_Hilliard_FDM.alpha_v…粒子の並進温度に対する補正項.

[*3] KAPSELには，NS方程式の陰解法としてMAC陰解法[46]が実装されている．陰解法を用いると，計算時間は増えるものの粘度が異なる二成分相分離流体のシミュレーションを行うことができる．解法の詳細は付録D.1に示す． implicit_schemeを選ぶと，デフォルトで線形連立方程式の反復解法としてBiCGSTAB法[47]が選択されるとともに，表中に示す新たな設定項目が現れる．陰解法の反復解法では，残差のオーダーがtoleranceで設定した値より小さくなると，解が収束したと判定される．また，maximum_iterationよりも反復回数が多くなると解が得られなかったと判定され，シミュレーションは終了される． Lisライブラリ[48]と連携することで，BiCGSTAB法以外の反復解法を使って計算することもできる．詳細は付録Eに示す.

[*4] CH方程式の解法として，陽解法の他に，時間微分において後退差分，非線形のポテンシャル項についてはAdams–Bashforth法による外挿値を用いた半陰解法を実装している[49]．解法の詳細は付録Dに示す．また，CH方程式の半陰解法において，Lisライブラリ[48]と連携することでBiCGSTAB法以外の反復解法を使って計算することができる．詳細は付録Eに示す.

[*5] シミュレーションの時間刻みはここで設定した粘度をもとに設定される． viscosity_changeをONにしたときの二成分流体の粘度は時間刻みに影響を与えない．そのため，ここで設定したETAより極端に大きなETA_AやETA_Bを設定すると，シミュレーション精度が悪化したり計算不安定となる可能性があることに注意する．詳細は6.3.3節に示す.

- constitutive_eq.Navier_Stokes_Cahn_Hilliard_FDM.alpha_o...粒子の回転温度に対する補正項.
- constitutive_eq.Navier_Stokes_Cahn_Hilliard_FDM.Potential.type...二成分相分離流体の二重井戸型ポテンシャルを`Landau`(Landau型)と`Flory_Huggins`(FH型)から選択する.
- constitutive_eq.Navier_Stokes_Cahn_Hilliard_FDM.Potential.Landau.composition_ratio...Landau型ポテンシャル使用時の流体の組成比.
- constitutive_eq.Navier_Stokes_Cahn_Hilliard_FDM.Potential.Landau.initial_fluctuation...Landau型ポテンシャル使用時の初期濃度揺らぎ.
- constitutive_eq.Navier_Stokes_Cahn_Hilliard_FDM.Potential.Landau.a...Landau型ポテンシャル使用時の4次の項のパラメータa.
- constitutive_eq.Navier_Stokes_Cahn_Hilliard_FDM.Potential.Landau.b...Landau型ポテンシャル使用時の2次の項のパラメータb.
- constitutive_eq.Navier_Stokes_Cahn_Hilliard_FDM.Potential.Landau.d...Landau型ポテンシャル使用時の粒子内部の流体領域の組成を任意の値ψ_0に保つためのパラメータd.
- constitutive_eq.Navier_Stokes_Cahn_Hilliard_FDM.Potential.Landau.w...Landau型ポテンシャル使用時の粒子–流体間相互作用のパラメータw.
- constitutive_eq.Navier_Stokes_Cahn_Hilliard_FDM.Potential.Landau.z...Landau型ポテンシャル使用時の粒子と流体–流体界面間相互作用のパラメータz[*6].
- constitutive_eq.Navier_Stokes_Cahn_Hilliard_FDM.Potential.Landau.psi_0...粒子内部のψを任意の値ψ_0にするためのパラメータ.
- constitutive_eq.Navier_Stokes_Cahn_Hilliard_FDM.Potential.Landau.alpha...Landau型ポテンシャル使用時の流体–流体界面エネルギーのパラメータα.
- constitutive_eq.Navier_Stokes_Cahn_Hilliard_FDM.Potential.Landau.kappa...Landau型ポテンシャル使用時の易動度κ.
- constitutive_eq.Navier_Stokes_Cahn_Hilliard_FDM.Potential.Flory_Huggins.composition_ratio...FH型ポテンシャル使用時の流体の組成比.
- constitutive_eq.Navier_Stokes_Cahn_Hilliard_FDM.Potential.Flory_Huggins.initial_fluctuation...FH型ポテンシャル使用時の初期濃度揺らぎ.
- constitutive_eq.Navier_Stokes_Cahn_Hilliard_FDM.Potential.Flory_Huggins.na...A成分の重合度N_A.
- constitutive_eq.Navier_Stokes_Cahn_Hilliard_FDM.Potential.Flory_Huggins.nb...B成分の重合度N_B.
- constitutive_eq.Navier_Stokes_Cahn_Hilliard_FDM.Potential.Flory_Huggins.chi...Floryの相互作用パラメータχ.
- constitutive_eq.Navier_Stokes_Cahn_Hilliard_FDM.Potential.Flory_Huggins.d...FH型ポテンシャル使用時の粒子内部の流体領域の組成を任意の値ψ_0に保つためのパラメータd.
- constitutive_eq.Navier_Stokes_Cahn_Hilliard_FDM.Potential.Flory_Huggins.w...FH型ポテンシャル使用時の粒子–流体間相互作用のパラメータw.
- constitutive_eq.Navier_Stokes_Cahn_Hilliard_FDM.Potential.Flory_Huggins.z...FH型ポテンシャル使用時の粒子と流体–流体界面間相互作用のパラメータz[*7].
- constitutive_eq.Navier_Stokes_Cahn_Hilliard_FDM.Potential.Flory_Huggins.psi_0...FH型ポテンシャル使用時に粒子内部のψを任意の値ψ_0にするためのパラメータ.
- constitutive_eq.Navier_Stokes_Cahn_Hilliard_FDM.Potential.Flory_Huggins.alpha...FH型ポテンシャル使用時の流体–流体界面エネルギーのパラメータα.

[*6] 粒子と流体–流体界面間相互作用は，Landau型ポテンシャルを用いた場合だけ有効である.
[*7] 粒子と流体–流体界面間相互作用は，Landau型ポテンシャルを用いた場合だけ有効である.

- constitutive_eq.Navier_Stokes_Cahn_Hilliard_FDM.Potential.Flory_Huggins.kappa...FH型ポテンシャル使用時の易動度κ.
- constitutive_eq.Navier_Stokes_Cahn_Hilliard_FDM.Wall_Potential.type...ONを選ぶと平面壁が導入されたシミュレーションができる.
- constitutive_eq.Navier_Stokes_Cahn_Hilliard_FDM.Wall_Potential.ON.w...平面壁の流体の親和性を決めるパラメータ.
- constitutive_eq.Navier_Stokes_Cahn_Hilliard_FDM.Wall_Potential.ON.psi_0.magnitude...平面壁と流体の親和性の大きさを決めるパラメータ.
- constitutive_eq.Navier_Stokes_Cahn_Hilliard_FDM.Wall_Potential.ON.psi_0.profile...uniformを選ぶと平面壁面上で均一な流体の親和性が設定され，user_specifyを選ぶと手動設定できる.
- constitutive_eq.Navier_Stokes_Cahn_Hilliard_FDM.Wall_Potential.ON.psi_0.user_specify.PSI0[][]...平面壁面上の流体の親和性の値を手動設定する.
- constitutive_eq.Navier_Stokes_Cahn_Hilliard_FDM.Wall_Potential.ON.DRYING.type...ONを選ぶと乾燥のシミュレーションができる.
- constitutive_eq.Navier_Stokes_Cahn_Hilliard_FDM.Wall_Potential.ON.DRYING.ON.psi_dry...乾燥速度を決めるパラメータ.

せん断流下のシミュレーション設定

Shear_NS_LE_CH_FDM以下で二成分相分離流体の情報を入れる.

- constitutive_eq.Shear_NS_LE_CH_FDM.NS_solver.type...NS方程式の解法をexplicit_scheme(陽解法)あるいはimplicit_scheme(陰解法)から選択する[*8].
- constitutive_eq.Shear_NS_LE_CH_FDM.NS_solver.implicit_scheme.tolerance...NS方程式の陰解法の収束判定基準.
- constitutive_eq.Shear_NS_LE_CH_FDM.NS_solver.implicit_scheme.maximum_iteration...NS方程式の陰解法の最大反復回数.
- constitutive_eq.Shear_NS_LE_CH_FDM.NS_solver.implicit_scheme.viscosity_change...ONを選ぶと粘度の異なる二成分相分離流体(A/B)のシミュレーションができる.
- constitutive_eq.Shear_NS_LE_CH_FDM.NS_solver.implicit_scheme.ON.ETA_A...A成分流体の粘度.
- constitutive_eq.Shear_NS_LE_CH_FDM.NS_solver.implicit_scheme.ON.ETA_B...B成分流体の粘度.
- constitutive_eq.Shear_NS_LE_CH_FDM.CH_solver.type...CH方程式の解法をexplicit_scheme(陽解法)あるいはimplicit_scheme(陰解法)から選択する[*9].
- constitutive_eq.Shear_NS_LE_CH_FDM.CH_solver.implicit_scheme.tolerance...CH方程式の陰解法の収束判定基準.
- constitutive_eq.Shear_NS_LE_CH_FDM.CH_solver.implicit_scheme.maximum_iteration...CH方程式の陰解法の最大反復回数.
- constitutive_eq.Shear_NS_LE_CH_FDM.DX...長さの単位量である格子幅Δ.
- constitutive_eq.Shear_NS_LE_CH_FDM.RHO...流体の密度.
- constitutive_eq.Shear_NS_LE_CH_FDM.ETA...流体の粘度[*10].

[*8] 初期流れ場なしのシミュレーションと同様，NS方程式の陰解法としてMAC陰解法[46]が実装されている.

[*9] 初期流れ場なしのシミュレーションと同様，CH方程式の解法として陽解法の他に半陰解法が実装されている[49].

[*10] シミュレーションの時間刻みはここで設定した粘度をもとに設定される． viscosity_changeをONにしたときの二成分流体の粘度は時間刻みに影響を与えない．そのため，ここで設定したETAより極端に大きなETA_AやETA_Bを設定すると，シミュレーション精度が悪化したり計算不安定となる可能性があることに注意する．詳細は6.3.3節に示す.

- constitutive_eq.Shear_NS_LE_CH_FDM.kBT...粒子温度.
- constitutive_eq.Shear_NS_LE_CH_FDM.alpha_v...粒子の並進温度に対する補正項.
- constitutive_eq.Shear_NS_LE_CH_FDM.alpha_o...粒子の回転温度に対する補正項.
- constitutive_eq.Shear_NS_LE_CH_FDM.Potential.type...二成分相分離流体の二重井戸型ポテンシャルを`Landau`(Landau型)と`Flory_Huggins`(FH型)から選択する.
- constitutive_eq.Shear_NS_LE_CH_FDM.Potential.Landau.composition_ratio...Landau型ポテンシャル使用時の流体の組成比.
- constitutive_eq.Shear_NS_LE_CH_FDM.Potential.Landau.initial_fluctuation...Landau型ポテンシャル使用時の初期濃度揺らぎ.
- constitutive_eq.Shear_NS_LE_CH_FDM.Potential.Landau.a...Landau型ポテンシャル使用時の4次の項のパラメータa.
- constitutive_eq.Shear_NS_LE_CH_FDM.Potential.Landau.b...Landau型ポテンシャル使用時の2次の項のパラメータb.
- constitutive_eq.Shear_NS_LE_CH_FDM.Potential.Landau.d...Landau型ポテンシャル使用時の粒子内部の流体領域の組成を任意の値ψ_0に保つためのパラメータd.
- constitutive_eq.Shear_NS_LE_CH_FDM.Potential.Landau.w...Landau型ポテンシャル使用時の粒子–流体間相互作用のパラメータw.
- constitutive_eq.Shear_NS_LE_CH_FDM.Potential.Landau.z...Landau型ポテンシャル使用時の粒子と流体–流体界面間相互作用のパラメータz[*11]
- constitutive_eq.Shear_NS_LE_CH_FDM.Potential.Landau.psi_0...粒子内部のψの値を決めるパラメータ.
- constitutive_eq.Shear_NS_LE_CH_FDM.Potential.Landau.alpha...Landau型ポテンシャル使用時の流体–流体界面エネルギーのパラメータα.
- constitutive_eq.Shear_NS_LE_CH_FDM.Potential.Landau.kappa...Landau型ポテンシャル使用時の易動度κ.
- constitutive_eq.Shear_NS_LE_CH_FDM.Potential.Flory_Huggins.composition_ratio...FH型ポテンシャル使用時の流体の組成比.
- constitutive_eq.Shear_NS_LE_CH_FDM.Potential.Flory_Huggins.initial_fluctuation...FH型ポテンシャル使用時の初期濃度揺らぎ.
- constitutive_eq.Shear_NS_LE_CH_FDM.Potential.Flory_Huggins.na...A成分の重合度N_A.
- constitutive_eq.Shear_NS_LE_CH_FDM.Potential.Flory_Huggins.nb...B成分の重合度N_B.
- constitutive_eq.Shear_NS_LE_CH_FDM.Potential.Flory_Huggins.chi...Floryの相互作用パラメータχ.
- constitutive_eq.Shear_NS_LE_CH_FDM.Potential.Flory_Huggins.d...FH型ポテンシャル使用時の粒子内部の流体領域の組成を任意の値ψ_0に保つためのパラメータd.
- constitutive_eq.Shear_NS_LE_CH_FDM.Potential.Flory_Huggins.w...FH型ポテンシャル使用時の粒子–流体間相互作用のパラメータw.
- constitutive_eq.Shear_NS_LE_CH_FDM.Potential.Flory_Huggins.z...FH型ポテンシャル使用時の粒子と流体–流体界面間相互作用のパラメータz[*12].
- constitutive_eq.Shear_NS_LE_CH_FDM.Potential.Flory_Huggins.psi_0...FH型ポテンシャル使用時に粒子内部のψの値を決めるパラメータ.
- constitutive_eq.Shear_NS_LE_CH_FDM.Potential.Flory_Huggins.alpha...FH型ポテンシャル使用時の流体–流体界面エネルギーのパラメータα.
- constitutive_eq.Shear_NS_LE_CH_FDM.Potential.Flory_Huggins.kappa...FH型ポテンシャル使用時の易動

[*11] 粒子と流体–流体界面間相互作用は，Landau型ポテンシャルを用いた場合だけ有効である.
[*12] 粒子と流体–流体界面間相互作用は，Landau型ポテンシャルを用いた場合だけ有効である.

度κ.

- constitutive_eq.Shear_NS_LE_CH_FDM.External_field.type...DC(定常せん断流れ)または AC(振動せん断流れ)を選択する.
- constitutive_eq.Shear_NS_LE_CH_FDM.External_field.DC.Shear_rate...せん断速度$\dot{\gamma}$.
- constitutive_eq.Shear_NS_LE_CH_FDM.External_field.AC.Shear_rate...振動せん断速度の振幅$\dot{\gamma}_0$.
- constitutive_eq.Shear_NS_LE_CH_FDM.External_field.AC.Frequency...せん断速度の周波数ω.

6.3.2　オブジェクト（粒子）の設定

object_type.typeでは，spherical_particle(球状粒子)，chain(フレキシブル鎖)，rigid(剛体)を選択できる.

6.3.3　空間単位と時間単位

長さの単位として格子幅Δを採用している．時間の単位τ_0は，流体の密度ρと粘度ηと格子幅で決まる$\tau_0 = \rho\Delta^2/\eta$を採用している．ここで，粘度が異なる二成分相分離流体のシミュレーションの場合，時間の単位を計算する際に用いる粘度ηは，constitutive_eq.*.ETAで設定されたものであり，各成分の流体の粘度，constitutive_eq.*.NS_solver.implicit_scheme.ON.ETA_Aやconstitutive_eq.*.NS_solver.implicit_scheme.ON.ETA_Bでないことに注意する (ここで，*は，Navier_Stokes_Cahn_Hilliard_FDMとShear_NS_LE_CH_FDMである).

- 仮に入力udfファイルでRHO= A，ETA= B，DX= Cと入力した場合，最大波数k_{max}はCを用いて与えられ，時間刻みの上限は運動量の拡散時間より$T_{dump} = (A/B)/k_{max}^2$と与えられる．時間刻み$\Delta t$は$T_{dump} \times$ factorで調整する.
- 現実との対応を考察する．考察したい空間スケールとして，グリッドサイズを1 μmの長さを考える．また溶媒として水($\eta = 1 \times 10^{-3}$ Pa s，$\rho = 1 \times 10^3$ kg m^{-3})を対象とした場合，時間の単位は$\tau_0 = 1 \times 10^{-6}$ sとなる.

6.4　計算事例

本節では，二成分相分離流体に分散した粒子のシミュレーション事例を示す． 6.4.1節では，短時間のシミュレーションを通して，KAPSELを使ったシミュレーションの手続き全体を概観する．ここでは，初期流れ場なしの二成分相分離流体中の粒子を扱う． 6.4.2節では，せん断流下の粒子が添加された二成分相分離流体のシミュレーションを行い，レオロジー特性を調べる． 6.4.3節では，Pickeringエマルション[35]のシミュレーション事例を紹介する．

6.4.1　二成分相分離流体に分散する粒子の運動

シミュレーションの実行

本節で実行するシミュレーションのUDFファイル一式は，Examples/11a/フォルダにある．下記のコマンドでExamples/11a/フォルダに移動し，シミュレーションを実行できる．

```
$ cd Examples/11a
$ ../../kapsel -Iinput.udf -Ooutput.udf -Ddefine.udf -Rrestart.udf
```

このシミュレーションでは，式(6.5)のLandau二重井戸型ポテンシャルにより相分離する二成分流体を扱う．各パラメータは，シミュレーション実行時に標準出力される:

```
#################################
# Landau potential is selected.
# Parameters
# Composition ratio   : 0
# Initial fluctuation : 0.05
# a      : 1
# b      : 1
# d      : 1
# w      : -1
# z      : 0
# psi0_p : 0
# alpha  : 1
# kappa  : 1
#
#################################
```

本シミュレーションでは，Composition ratioが0に設定されており，二成分流体がどちらも等しい量存在する系となっている．また，wが−1に設定されていることから，粒子は片方の成分(A成分)と親和性が高い系となっている． psi0_pは，粒子内部のψの値を表すパラメータである．

シミュレーションが正しく終了すると，以下のようにシミュレーションにかかった時間が表示される．

```
#Simulation has ended!
#Total Running Time (s):      11.89
#                   (m):       0.20
#                   (h):       0.00
#Average Step Time  (s):       0.01
#                   (m):       0.00
#                   (h):       0.00
```

計算時間は実行環境によって異なるが，本シミュレーションの場合数十秒程度で終了するはずである．シミュレーションが終了すると，Examples/11a/フォルダに下記のファイルが出力される:

- output.udf
- restart.udf
- fluid_phase.xmf

- flux_field.xmf
- particle_coordinate.xmf
- particle_phase.xmf
- velocity_field.xmf
- flux_*.h5
- orderparam_*.h5
- particle_*.h5
- particle_data_*.h5
- velocity_*.h5

ここで，*は連番を表す． output.udfはシミュレーション結果を含む出力UDFファイルであり，GOURMETで読み込むことができる． GOURMETでのデータの確認方法は後述する．また，restart.udfはシミュレーションを途中から再開するためのリスタート用UDFファイルである．拡張子h5の連番のファイル群には各出力ステップにおけるシミュレーションデータがHDF5バイナリ形式で格納されている．このファイル群は，時系列のシミュレーションデータを管理するeXtensible Data Model and Format(XDMF)ファイルにより，まとめて扱うことができる．XDMFファイルは拡張子xmfのファイルであり，KAPSELでは以下のように拡張子h5のファイル群と関連付けられる:

- fluid_phase.xmf...二成分流体の識別関数ψのデータ(orderparam_*.h5)
- flux_field.xmf...物質流束$\boldsymbol{J}$(式6.3)のデータ(flux_*.h5)
- particle_coordinate.xmf...粒子の座標データ(particle_data_*.h5)
- particle_phase.xmf...粒子識別関数ϕのデータ(particle_*.h5)
- velocity_field.xmf...流体の速度場$\boldsymbol{u}$のデータ(velocity_*.h5)

XDMFファイルはParaView [*13] やMayavi [*14] 等の可視化ソフトを使って表示することができる．次節でParaViewによる可視化の例を示す．また，拡張子h5のファイルに格納されたデータは，h5dumpコマンドで直接確認できる．以下は，orderparam_0.h5のデータを確認するコマンドである．

```
$ h5dump orderparam_0.h5
```

シミュレーションデータの可視化

本節では，GOURMETとParaViewを用いてシミュレーション結果を確認する．

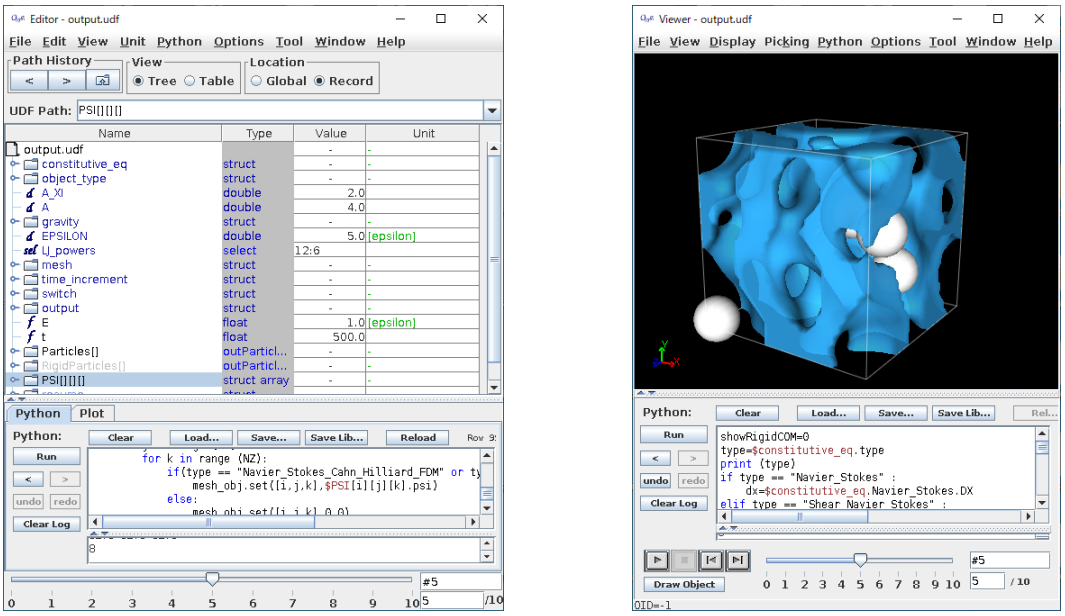
GOURMETを用いたシミュレーションデータの可視化

GOURMETで出力UDFファイル(output.udf)を読み込むと，シミュレーションで用いたパラメータの一覧や，各出力ステップのシミュレーション結果を確認することができる(図6.3(a))．また，Examples/11a/フォルダ内にあるgourmet_particle_field_show.pyをGOURMETに読み込むと，二成分流体の識別関数ψを可視化することができる．具体的には，GOURMETの下部のPythonパネルでこのスクリプトを読み込んで実行すると，新たな描画ウィンドウが開き，シミュレーション結果をアニメーションで確認できる(図6.3(b))．図中では，粒子が白の球，流体界面が青い等値面として表示されている．これらの表示はスクリプトを編集することで変更できる．

ParaViewを用いたシミュレーションデータの可視化

ParaViewでXDMFファイルを読み込むと，結果の可視化や解析を行うことができる． Examples/11a/フォルダにあるpv_sample.pvsmは，ParaView用のStateファイルである． ParaViewを起動し，左上のFilesからLoad

[*13] https://www.paraview.org/
[*14] http://docs.enthought.com/mayavi/mayavi/



(a) 出力UDFファイルに格納されたパラメータの表示.　**(b)** Pythonスクリプトによる二成分流体の識別関数ψの可視化.

図6.3: GOURMETによるシミュレーション結果の可視化.

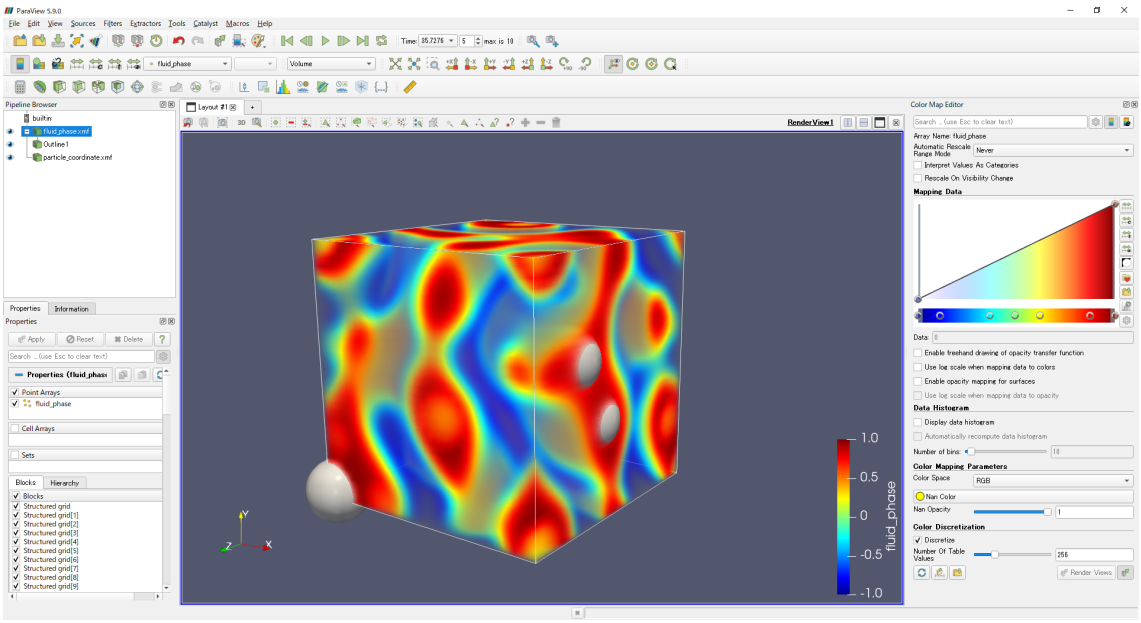


図6.4: Paraviewによるシミュレーション結果の可視化.

`State Files`を選択し，`pv_sample.pvsm`を読み込むと [*15]，二成分流体の識別関数ψを可視化できる(図6.4)．図中では，粒子が白の球として表示されており，二成分流体の一方の成分は赤色，もう一方の成分は透過して表示されている． ParaViewは，データを可視化できるだけでなく，強力な解析機能を多く有している．詳細は公式ドキュメント[50]を参照頂きたい．

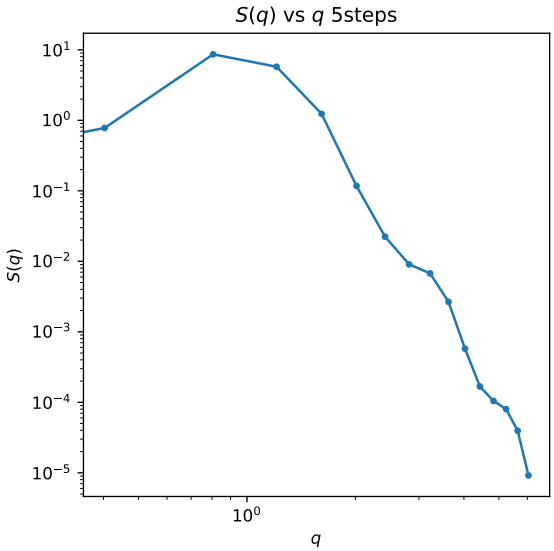


図6.5: Pythonスクリプトによる解析例: 静的構造因子$S(q)$を波数qに対してプロットしている(5ステップ目の結果を示す).

Pythonスクリプトによるシミュレーションデータの解析

KAPSELで出力されたシミュレーションデータは，Pythonスクリプトで読み込み，別途解析することもできる．`Examples/11a/`フォルダには，相分離構造の静的構造因子$S(q)$[51]を計算するPythonスクリプトが同梱されている． Pythonパッケージの`h5py`，`scipy`，`matplotlib`が導入された環境で以下のコマンドを実行することで，各時間ステップにおける$S(q)$のグラフが逐次得られる(図6.5).

```
$ python sq_analysis.py
```

6.4.2　せん断流下の二成分相分離流体に分散する粒子の運動

本節では`Examples/11/`フォルダの計算事例について紹介する．ここでは，定常せん断流下の粒子が添加された二成分相分離流体のシミュレーションを行う．シミュレーションは，以下のコマンドで実行できる:

```
$ cd Examples/11
$ ../../kapsel -Iinput.udf -Ooutput.udf -Ddefine.udf -Rrestart.udf
```

この例では，粒子半径$a = 4$，界面厚み$\xi = 2$の粒子が128個添加された二成分相分離流体にせん断速度$\dot{\gamma} = 0.01$の定常せん断流が印加されている．また，粒子温度は$k_{\mathrm{B}}T = 1$に設定されている．計算メッシュサイズは64×64×64である．二成分流体は式(6.5)のLandau二重井戸型ポテンシャルにより相分離する．各パラメータは，シミュレーション実行時に標準出力される:

```
################################
# Landau potential is selected.
# Parameters
# Composition ratio   : 0
# Initial fluctuation : 0.5
# a      : 1
# b      : 1
# d      : 1
# w      : 0
```

[15] ”Load State Data File Options”で“Search files under specified directory”を選択し，”Data Directory”に適切なパスを指定する．

```
# z      : 0
# psi0_p : 0
# alpha  : 1
# kappa  : 1
#
##################################
```

せん断流下でのシミュレーションでは，各ステップごとに以下のデータが1行ずつ標準エラー出力される:

- 1:time...経過時間
- 2:shear_rate...せん断速度
- 3:degree_oblique...斜交座標系の歪み [*16]
- 4:shear_strain_temporal...印加された歪み($\dot{\gamma}t$)
- 5:lj_dev_stress_temporal...粒子間ポテンシャル由来のせん断応力
- 6:shear_stress_temporal_old...粒子–流体間相互作用由来のせん断応力
- 7:shear_stress_temporal_new...粒子–流体間相互作用由来のせん断応力 [*17]
- 8:reynolds_stress...レイノルズせん断応力
- 9:fluid_stress...バルク流体由来のせん断応力
- 10:interfacial_stress...流体–流体界面由来のせん断応力
- 11:apparent_stress...懸濁液のせん断応力 [*18]
- 12:viscosity...懸濁液の粘度 [*19]

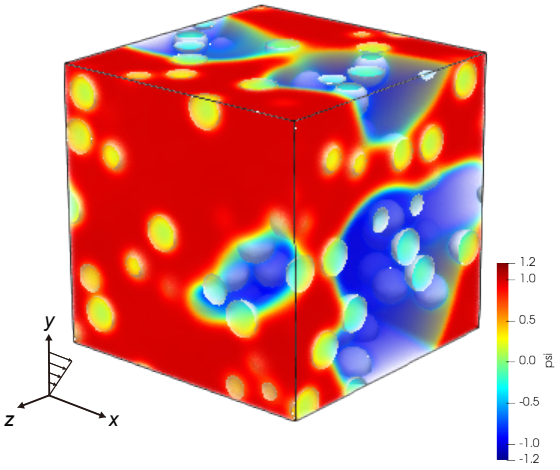


図6.6: 計算事例11(せん断流下の粒子が添加された二成分流体)の最終ステップで得られる流体界面関数ψのスナップショット.

図6.6は，本シミュレーションの最終ステップ(20,000ステップ)で得られる流体界面関数ψのスナップショットを表す． KAPSELでは，せん断流動方向をx軸，せん断勾配方向をy軸，中立方向(渦度方向)をz軸としてせん断流動が印加される． 2成分流体に対する粒子の親和性を決めるパラメータwを0に設定していることから，粒子が片方の相に偏在せず系全体に分散した結果が得られる.

[*16] KAPSELでは，一様なせん断流動を実現するため，時間と共にせん断方向に歪む斜交座標系上で計算を行っている．その際，計算安定性の観点から，斜交座標系の歪みがある一定値に到達したとき歪みをリセットし，もとの直交座標系へ値をリマップする操作を行っている． 3:degree_obliqueで表示される値は，このリマップを考慮した斜交座標系の歪みであり，系に印加された歪み($\dot{\gamma}t$)とは異なることに注意する．詳細は文献[23]に記されている.

[*17] 粒子–流体間相互作用由来のせん断応力はoldとnewの2種類の計算方法が実装されている．通常はnewを用いればよい.

[*18] 5:lj_dev_stress_temporal，7:shear_stress_temporal_new，8:reynolds_stress，9:fluid_stress，10:interfacial_stressの総和である.

[*19] 11:apparent_stressを2:shear_rateで割った値である.

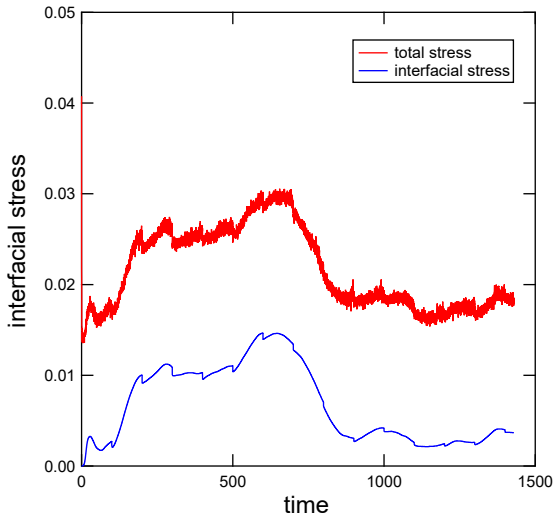


図6.7: 計算事例11(せん断流下の粒子が添加された二成分流体)で得られるせん断応力(xy成分)の時間変化．赤線は，懸濁液のせん断応力(`11:apparent_stress`)，青線は流体–流体界面由来のせん断応力(`10:interfacial_stress`)を示す．

図6.7は，せん断応力(xy)成分の時間変化を示す．図中の赤線は，懸濁液のせん断応力(標準エラー出力の`11:apparent_stress`)，青線は流体–流体界面由来のせん断応力(`10:interfacial_stress`)を示す．本シミュレーションでは，流体–流体界面由来のせん断応力が懸濁液のせん断応力に大きく寄与していることが分かる．また，懸濁液のせん断応力は粒子の熱運動に起因する細かい変動がみられるものの，流体–流体界面由来のせん断応力ではその変動はみられない．これは，KAPSELでは粒子の熱揺らぎを流体のNavier–Stokes方程式にではなく粒子のLangevin方程式を通して導入しているためである． KAPSELでの粒子温度の導入の詳細は3.1.1節を参照頂きたい．

6.4.3　Pickeringエマルション

本節では`Examples/13/`フォルダの計算事例について紹介する．ここでは，Pickering[35]により報告された微粒子添加によって安定化されるエマルション(Pickeringエマルション)のシミュレーションを行う． Pickeringエマルションは，流体–流体界面に微粒子が吸着されることによって安定化される． KAPSELでは，式(6.4)の第5項のパラメータzを通して粒子と二成分流体の界面との親和性を変えることができる．ここでは，二成分流体に対して中立な親和性($w = 0$)を持つ粒子(粒子半径$a = 4$，界面厚み$\xi = 2$)が200個添加された系において，粒子が二成分相分離流体の界面に親和性がある場合($z = -0.6$)について計算を行った．二成分流体は式(6.5)のLandau二重井戸型ポテンシャルにより相分離を起こし，パラメータは $a = b = d = \kappa = 1$，$\alpha = 3$に設定されている．

シミュレーションは，以下のコマンドで実行できる:

```
$ cd Examples/13
$ ../../kapsel -Iinput.udf -Ooutput.udf -Ddefine.udf -Rrestart.udf
```

`Examples/13`フォルダの`input.udf`は，粒子–二成分流体界面の親和性がある系($z = -0.6$)のシミュレーションとなっている．

図6.8は，本シミュレーションの最終ステップ(200,000ステップ)で得られる流体界面関数ψの粒子–二成分流体界面の親和性がある場合($z = -0.6$)のスナップショットを表す．図6.8 粒子–二成分流体界面の親和性がある場合(図6.8)，粒子は二成分流体界面に引き寄せられており，界面活性剤の役割を果たしていることがわかる．

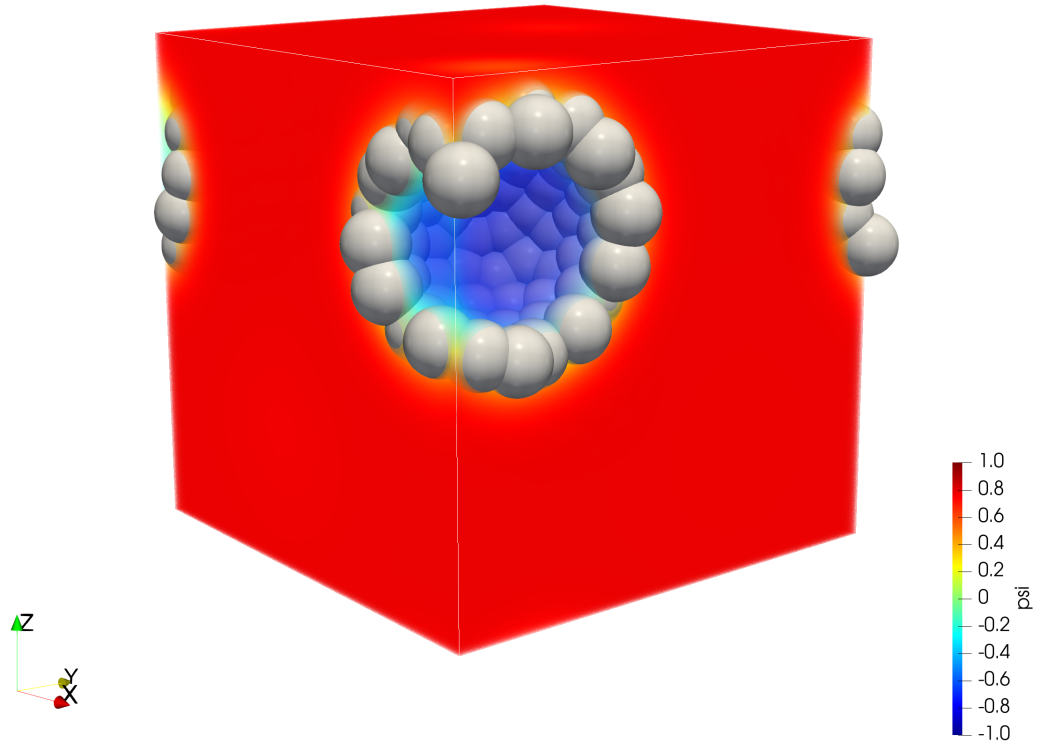


図6.8: 計算事例13(Pickeringエマルション)の最終ステップで得られる流体界面関数ψのスナップショット.

第7章

マイクロスイマーのシミュレーション

7.1 理論的背景と基礎方程式

粘性流体中を自発的に泳動するマイクロスイマーの例は，精子・バクテリア・藻類など生物学やライフサイエンスにおいて至るところに存在する．また，自発的に運動するソフトマターであるアクティブマターの主な代表例の一つでもある．アクティブマターの研究は，非平衡系理論の開発・検証に貢献することが期待され，理論的な観点で非常にメリットが大きい．また応用的な側面においても，合成マイクロスイマーを導入することで，スマートドラッグデリバリーような最新医学の発展への貢献も期待できる．しかし，流体力学的相互作用が系の物性を決める上でどのような役割を果たすかは複雑であり，そのメカニズムについて理解することは依然として大きな課題である[52].

7.1.1 squirmersモデル

KAPSELでは，squirmersモデルと呼ばれる一般的なマイクロスイマーのモデルを採用している． squirmersモデルはLighthillとBlakeら[53, 54]によって開発されたモデルであり，球形マイクロスイマーの流体力学的相互作用を調べる際に一般的に用いられている．

KAPSELでは，半径 a の球状スイマーを考える． squirmersモデルでは，粒子表面での繊毛の動きを詳細にモデル化する代わりに，剛体粒子表面に特定の境界条件を課すことによって同様の推進効果を再現する．ここで，基準系を決定する正規直交基底ベクトルを $\{\tilde{\boldsymbol{e}}_i\}$ とし，粒子の泳動方向軸を $\tilde{\boldsymbol{e}}_3 = \tilde{\boldsymbol{e}}$ で固定する．粒子表面の点は，極角 θ と方位角 λ に対応する単位接ベクトル $\boldsymbol{\theta}$，$\boldsymbol{\lambda}$ と，動径方向の単位ベクトル $\boldsymbol{\varrho}$ ($\theta = \arccos \boldsymbol{\varrho} \cdot \tilde{\boldsymbol{e}}$) を用いて表される．動径速度を 0 とした場合，squirmerに印加されるslip境界条件は[55]

$$\boldsymbol{u}^{sq} = \sum_{n=1}^{\infty} \frac{2}{n(n+1)} B_n P_n'(\cos\theta)\sin\theta\boldsymbol{\theta} + \sum_{n=1}^{\infty} C_n P_n'(\cos\theta)\sin\theta\boldsymbol{\lambda} \tag{7.1}$$

である．ここで，P_n' は n 次のLegendre多項式の導関数であり， B_n と C_n はそれぞれ n 次の極モードおよび方位角モードである．方位角モードを無視し($C_n = 0$)，また2次より大きい次数での極モードも無視できる($B_n = 0$ where $n > 2$) とすると，pusher(後方で推進力が生じるスイマー)およびpuller(前方で推進力が生じるスイマー)の両方を表現できるシンプルかつ重要なモデルを構築できる．この場合，表面速度は $\alpha = B_2/B_1$ として

$$\boldsymbol{u}^{sq}(\theta) = B_1\left(\sin\theta + \frac{\alpha}{2}\sin 2\theta\right)\boldsymbol{\theta} \tag{7.2}$$

となる． B_1 は流体中を泳動する単一粒子の定常状態における速さ($U = 2/3B_1$)を決めるパラメータであり，B_2 はstresslet強度である．これらモードの比 α はスイマーの種類を決定し，$\alpha > 0$ ではpuller，$\alpha < 0$ ではpusher，$\alpha = 0$ ではいわゆるneutral swimmerとなる． pusher/pullerスイマーとその推進に伴う流体の流れの概略を図7.1に示す．泳動の性質は，その推進から生じる流れに対して流体力学的に大きく影響を与え，粒子の集団行動を考える上でも重要である．

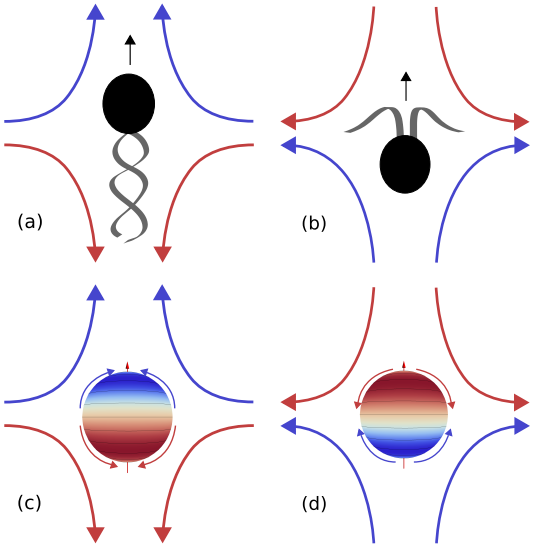


図7.1: (a)pusherスイマーおよび(b)pullerスイマーの推進メカニズムと流れの様子の概略図．これらスイマーをBlakeのsquirmingモデルで表現する，すなわち，詳細な推進メカニズムを粒子表面におけるslip境界条件で置き換えると，それぞれ(c)および(d)のように表される． Reproduced from Soft Matter 9, 4923-4936[4], Copyright 2013, with permission from the Royal Society of Chemistry.

7.1.2　マイクロスイマーについての基礎方程式

SP法を用いてマイクロスイマーの分散した系における流体力学を解くために，連続体方程式(3.13)に拘束力 $\phi \boldsymbol{f}_{sq}$ を新たに加え，粒子表面でのslip境界条件を実現する[4, 56].

$$\rho(\partial_t + \boldsymbol{u}\cdot\boldsymbol{\nabla})\boldsymbol{u} = \boldsymbol{\nabla}\cdot\boldsymbol{\sigma} + \rho\phi\boldsymbol{f}_p + \rho\phi\boldsymbol{f}_{sq} \tag{7.3}$$

上式を解くために，通常のSP法の1ステップに追加の手続きを組み入れる．移流項および粘性項を更新して $\boldsymbol{u}^\star$ を計算した後に，$\phi\boldsymbol{f}_p$ より剛性の拘束条件を実現するが，その間に新たに，squirmerの境界条件を満たす速度場 $\boldsymbol{u}^{**}$ を更新するため表面拘束力 $\phi\boldsymbol{f}_{sq}$ の計算を行う．これを実装するため，新しくSP関数 ϕ^{sq} を導入する．ϕ^{sq} は粒子界面領域内でのみ0以外の値をとり，

$$\phi_I^{sq} = (1-\phi_I)\frac{|\boldsymbol{\nabla}\phi_I|}{\max(|\boldsymbol{\nabla}\phi_I|)} \tag{7.4}$$

と表される．新しい速度場は次のような形で得られる．

$$\boldsymbol{u}^{**} = \boldsymbol{u}^* + \left[\int_{t_n}^{t_n+h} \mathrm{d}s\phi\boldsymbol{f}_{sq}\right] \tag{7.5}$$

$$\begin{aligned}\left[\int_{t_n}^{t_n+h} \mathrm{d}s\phi\boldsymbol{f}_{sq}\right] = &\sum_I^N \phi_I^{sq}\left(\boldsymbol{V}_I^\dagger + \boldsymbol{\Omega}_I^\dagger\times\boldsymbol{r}_I + \boldsymbol{u}_I^{sq} - \boldsymbol{u}^*\right) \\ &+ \sum_I^N \phi_I\left(\delta\boldsymbol{V}_I + \delta\boldsymbol{\Omega}_i\times\boldsymbol{r}_I\right) - \frac{h}{\rho}\boldsymbol{\nabla}p_{sq}\end{aligned} \tag{7.6}$$

ここで，第1項はslip境界条件を実際に固定するための項であり，第2項は局所的な運動量保存を保障するための項(つまりsquirmerが境界で流体を押すと反作用が働く)，第3項は最終的な流体速度場が非圧縮性となることを保障する項である．この段階では更新された粒子速度がまだわからないため，ここでは反復解を採用し，slip境界条件を $\boldsymbol{V}_I^\dagger$ $(\boldsymbol{\Omega}_I^\dagger)$ に対して実現する．流体力学的な力およびトルクは前述と同様に計算し($\boldsymbol{u}^*$ の代わりに $\boldsymbol{u}^{**}$ に用いる)，その計算結果を粒子速度を更新するために用いる．この計算手続きは，slip境界条件を実現するための速度 $\boldsymbol{V}_I^\dagger$ と

次の時間ステップにおける粒子速度$\boldsymbol{V}_l^{n=1}$が一致するまで繰り返される．この計算手法は単一スイマーの運動に対して検証済みであり，粒子速度および流体速度の両方が理論的に予測された値とよく一致することが確かめられている[4].

7.2　入力UDF について

7.2.1　流体の設定

constitutive_eqでは，以下を選択することができる．

- Navier_Stokes: Newton流体
- Shear_Navier_Stokes: ジグザクせん断流のNewton流体
- Navier_Stokes_FDM: Newton流体
- Navier_Stokes_Cahn_Hilliard_FDM: 二成分相分離流体

7.2.2　オブジェクト（粒子）の設定

object_type.typeでは，spherical_particle(球状粒子)のみ選択できる．

7.2.3　オブジェクト(平面壁)の設定

switch.wall.typeでは，NONEもしくはFLATを選択することができる．

7.3　計算事例

7.3.1　周期境界条件下のマイクロスイマーの運動

入力UDFとして，Examples/09/フォルダの squirm_single_a+2.udfおよびsquirm_phi0.1_a+2.udfを用いればよい． squirm_single_a+2.udfでは1粒子系での計算することができる．

```
$ ../../kapsel -Isquirm_single_a+2.udf -Ooutput.udf -Ddefine.udf -Rrestart.udf
```

この例ではメッシュサイズ64×64×64で計算している．このとき粒子体積分率は$\varphi = 0.002$であり，粒子数$N_p = 1$，粒子直径$D = 10$，界面厚さ$\xi = 2$である．マイクロスイマーは泳動速度$V = 2/3B_1 = 0.01$，泳動形態$B_2 = 2$である．

一方，squirm_phi0.1_a+2.udfでは多粒子系での計算をすることができる．

```
$ ../../kapsel -Isquirm_phi0.1_a+2.udf -Ooutput.udf -Ddefine.udf -Rrestart.udf
```

この例(Fig. 7.2)では粒子体積分率は$\varphi = 0.1$であり，粒子数$N_p = 50$，粒子直径$D = 10$，界面厚さ$\xi = 2$である．その他のパラメータは1粒子系と同じである．Fig. 7.2には，50個のマイクロスイマーが泳動しているスナップショットを表している．

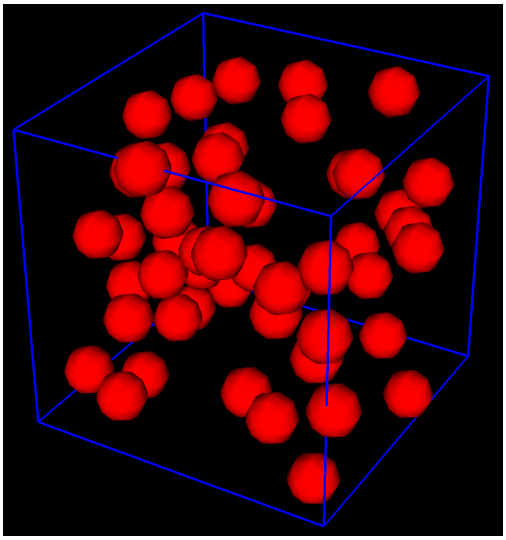


図7.2: 50個のマイクロスイマーが泳動しているスナップショット

7.3.2　2枚の平板間に拘束されたマイクロスイマーの運動

入力UDFとして，Examples/09/フォルダの squirm_wall.udfを用いればよい． squirm_wall.udfでは2枚の平板間に拘束されたマイクロスイマーを計算することができる．

```
$ ../../kapsel -Isquirm_wall.udf -Ooutput.udf -Ddefine.udf -Rrestart.udf
```

この例(Fig. 7.3)ではメッシュサイズ64×256×64で計算している．y軸に垂直な平面壁の厚みは4である．このとき粒子体積分率は$\varphi = 0.138$であり，粒子数$N_p = 4266$，粒子直径$D = 4$，界面厚さ$\xi = 2$である．マイクロスイマーは泳

動速度$V = 2/3B_1 = 0.25$，泳動形態$B_2 = 0.5$である．Gourmetでの可視化については，`particle_show_wall.py`を用いて行った．

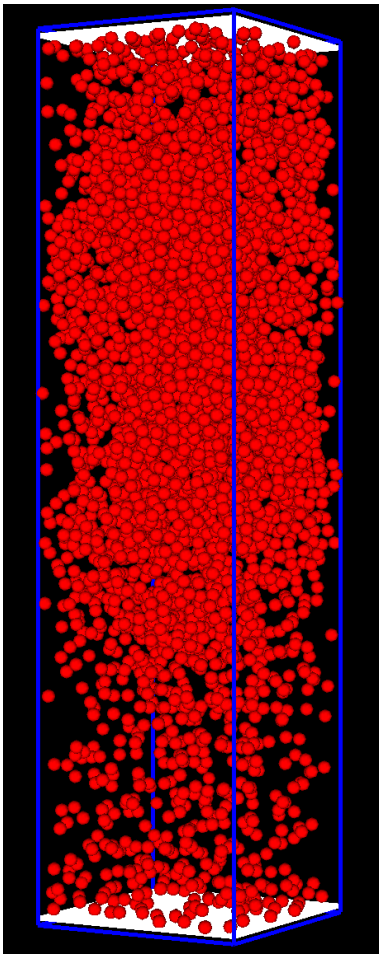


図7.3: 2枚の平板間に拘束されたマイクロスイマーの運動

第8章

クインケローラーのシミュレーション

8.1　理論的背景と基礎方程式

アクティブマターはバクテリアなどの生物の運動だけでなく，無生物もその対象である．自走コロイド粒子系はその代表例の一つであり，その研究は駆動機構や，集団ダイナミクスの理解といった基礎から，さらには，外場などによる制御といった応用まで幅広い利用が期待される．中でも，自走するコロイド粒子系における流体力学的相互作用がどのような影響をもたらすのかは複雑であり，その理解は重要な問題である．

8.1.1　クインケローラーとは

クインケローラーは，粘性流体中に分散された球状コロイド粒子が電極板上を転がることで，自走する系である(図8.1)[57]．自走の機構は，クインケ効果によりコロイドの自転を引き起こし，自転したコロイドが電極板の表面における滑りなしの境界条件により流体を介して力を受けることで生じる平板上の転がり運動によるものである．ここで，クインケ効果とは，電気伝導性を有する溶液中に誘電体コロイド粒子を分散させ，閾値以上の直流電場を印加すると，粒子が自発的に回転をし始める現象である[58]．このとき，自転の軸方向は常に直流電場に垂直な面上に固定され，自転の速さは電場の大きさに比例して決定される．クインケローラーの多体系では，流体力学はもちろん，静電相互作用，排除体積効果などは集団ダイナミクスに影響を与えることが期待され，その挙動は複雑である[57, 59]．

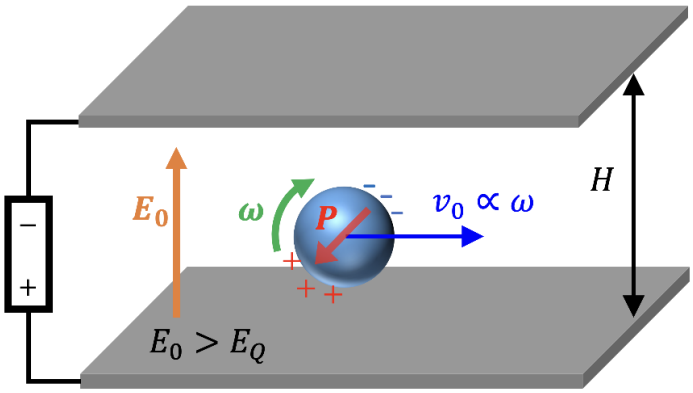


図8.1: クインケローラーの概略図．球状コロイド粒子に対して，一様な直流電場を印加すると，誘起された電気双極子の不安定性により，トルクが発生する．印加された電場E_0が閾値E_Qを超えるとコロイド球は自転する．自転するコロイド球は極板上の滑りなしの境界条件により，推進速度v_0は自転の角速度ωに比例する．

8.1.2　クインケローラーの基礎方程式

SP法を用いて，クインケローラーの運動を解析するため，運動方程式(3.5),(3.6)に対して，以下の2体間ポテンシャルを導入する；

$$U_{ij}^{DD}(r) = \frac{1}{4\pi\epsilon}\left(\frac{\boldsymbol{P}_i\cdot\boldsymbol{P}_j}{r^3} - 3\frac{(\boldsymbol{P}_i\cdot\boldsymbol{r})(\boldsymbol{P}_j\cdot\boldsymbol{r})}{r^5}\right), \tag{8.1}$$

$$U_{ij}^{LJ}(r) = 4\epsilon^{LJ}\left[\left(\frac{\sigma}{r}\right)^{36} - \left(\frac{\sigma}{r}\right)^{18}\right] + \epsilon, \tag{8.2}$$

$$U_{ij}^{EF}(r) = -\epsilon^{EF}\exp\left(-r/3\sigma\right)/r^2. \tag{8.3}$$

ここで，i, jは粒子ラベル，ϵは誘電率，U^{DD}, U^{LJ}, U^{EF}はそれぞれ電気双極子モーメント，排除体積効果，電気浸透流による相互作用ポテンシャルを表し，$\epsilon^{LJ}, \epsilon^{EF}$は排除体積効果および電気浸透流による相互作用ポテンシャルの大きさをそれぞれ表す．双極子モーメントは電場方向をz軸に取り，z成分P_zと，xy成分P_{xy}に分けられる．ここでは，簡単のため，クインケ効果による影響は定常的なものに限定する．コロイドの自転は電場方向に垂直な面にその軸方向をランダムに固定し，自転の速さは一定の値ωを入力する．自転の速さωの値は，回転のレイノルズ数$\mathrm{Re}_r \equiv \rho_f\omega\sigma^2/\eta$が指標となる．双極子モーメントの$xy$成分の向きは，電場と自転の角速度の向きの外積から決定される．コロイドの自転軸は平面上からずれないよう，平面からのずれを補正するトルクを調和振動子ポテンシャルの形で定義している．さらに，相互作用ポテンシャルの大きさは$\epsilon^{LJ} = \epsilon^{EF}$として設定している．

8.2　入力UDF について

8.2.1　流体の設定

constitutive_eqでは，以下を選択することができる．

- Navier_Stokes: Newton流体
- Navier_Stokes_FDM: Newton流体

8.2.2　オブジェクト（粒子）の設定

object_type.typeでは，rigid(剛体)のみ選択できる．

8.3　計算事例

8.3.1　単一のクインケローラーの運動

入力UDFとして，Examples/12/フォルダの a2_N1_Rer025.udfを用いればよい．

```
$ cd Examples/12
$ ../../kapsel -Ia2_N1_Rer025.udf -Ooutput.udf -Ddefine.udf -Rrestart.udf
```

この例ではメッシュサイズ64×64×32で計算している．このとき，粒子直径$\sigma = 4$，界面厚さ$\xi = 2$であり，回転のレイノルズ数$\mathrm{Re}_r = 0.25$，重力加速度$g = 1$である．流体場およびスナップショットの描画には，Examples/12/フォルダのstreamplot.pyを用いればよい．このとき，h5 フォーマットのデータを読み込んで描画を行う．

Fig. 8.2は，単一のクインケローラーの推進速度に関する潤滑理論との比較[60, 61]および，ローラー周りの流体場の様子を表している．

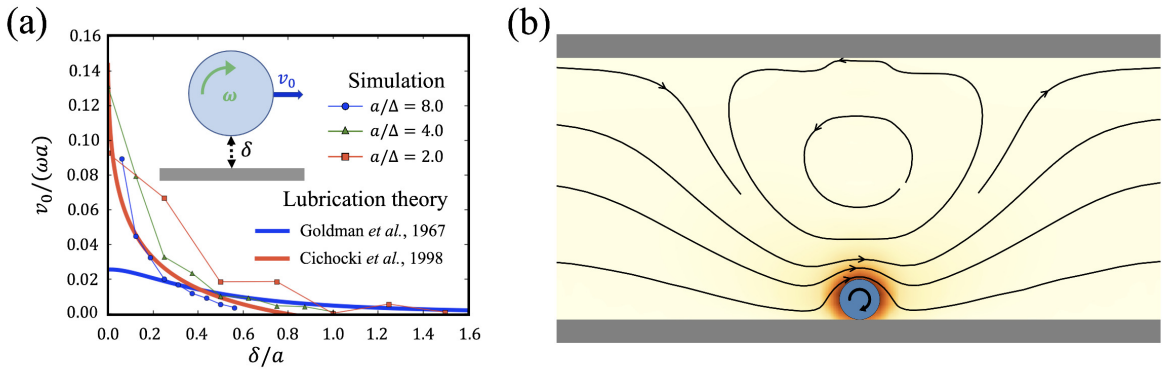


図8.2: 単一のクインケローラーの運動．(a) 粒子表面と壁面との距離δの関数としての粒子の推進速度v_0の関係．潤滑理論による解析と比較している．粒子半径aを大きく設定するほど，潤滑理論との良い一致を示す．(b) ローラー周りの流体場の様子．カラーは流速の大きさを表す．

8.3.2　多数のクインケローラーの運動

入力UDFとして，Examples/12/フォルダの a2_N200.udfを用いればよい．

```
$ cd Examples/12
$ ../../kapsel -Ia2_N200.udf -Ooutput.udf -Ddefine.udf -Rrestart.udf
```

この例(Fig. 8.3)ではメッシュサイズ128 × 128 × 32で計算している．このとき粒子体積分率は$\varphi = 0.00568$（面積分率は0.15）であり，粒子数$N_p = 200$である．また，相互作用ポテンシャルの大きさは$\epsilon^{LJ} = \epsilon^{EF} = 1.0$であり，双極子の大きさは$P_{xy}^2/4\pi = 6.0, |P_z|/|P_{xy}| = 3.0$としている．その他のパラメータは単一系と同じである．粒子の初期配置はExamples/12/フォルダのinit_loc_q.pyを用いることで生成できる．また，スナップショットの描画にはsnapshot.pyを用いればよい．

Fig. 8.3には，壁面上を駆動する多数のクインケローラーのスナップショットを表している．

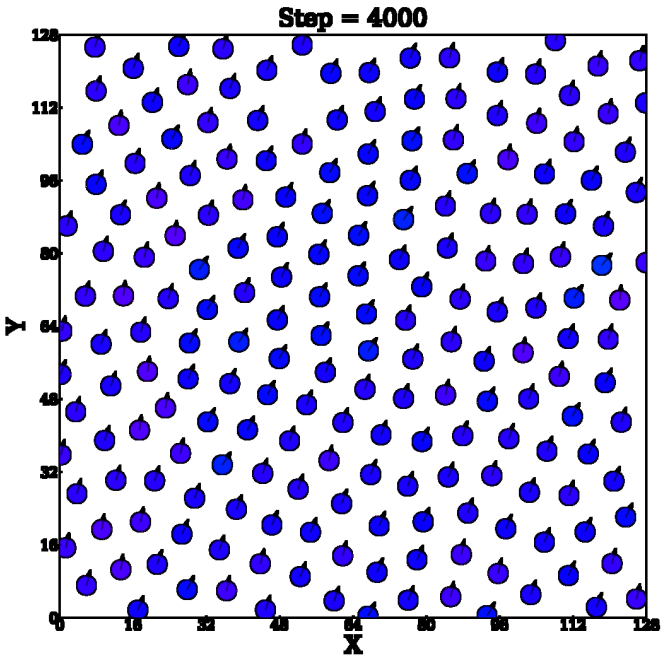


図8.3: 多数のクインケローラーのスナップショット．粒子のカラーおよび矢印は自転軸の方向を表している．

付録A

入力UDFの解説

本付録では，KAPSELでのシミュレーション条件を設定するための入力UDFを説明する．入力UDFの構造はdefine.udfによって定義されており，

- 流体の設定
- オブジェクトの設定
- 共通のシミュレーション設定
- 選択できる機能の設定
- データ出力設定
- UDF出力データクラスの定義
- リスタートの設定
- GOURMETでの表示設定

から構成されている．本節では各設定項目を順に概説する．詳細は対応する各章を参照頂きたい．

A.1　流体の設定

シミュレーションする流体の設定をconstitutive_eq構造体で行う．流体の種類は次のtypeセレクタで設定できる:

```
type: select {
  'Navier_Stokes',
  'Shear_Navier_Stokes',
  'Shear_Navier_Stokes_Lees_Edwards',
  'Electrolyte',
  'Navier_Stokes_FDM',
  'Navier_Stokes_Cahn_Hilliard_FDM',
  'Shear_Navier_Stokes_Lees_Edwards_FDM',
  'Shear_NS_LE_CH_FDM'
}
```

以降，本章では各設定を個別に説明する．

Navier_Stokes

Navier_Stokesを選ぶと，Newton流体中に分散した粒子のシミュレーションを行うことができる(第3章).
Navier_Stokes構造体では，流体に関する以下のパラメータを設定する:

```
DX:        double [L] "lattice spacing (=1), fixed for all directions"
RHO:       double [rho] "mass density of solvent"
ETA:       double [eta] "shear viscosity of solvent"
kBT:       double [epsilon] "temperature"
alpha_v:   double "correction coefficient of V"
alpha_o:   double "correction coefficient of Omega"
```

DXは格子幅，RHOは流体の密度，ETAは流体の粘度，kBTは粒子温度，alpha_vは粒子速度の調整パラメータ，alpha_oは粒子角速度の調整パラメータである．

Shear_Navier_Stokes

Shear_Navier_Stokesを選ぶと，ジグザグせん断流下でNewton流体中に分散した粒子のシミュレーションを行うことができる(本稿では触れない)． Shear_Navier_Stokes構造体では，流体に関する以下のパラメータを設定する:

```
DX:        double [L] "lattice spacing (=1), fixed for all directions"
RHO:       double [rho] "mass density of solvent"
ETA:       double [eta] "shear viscosity of solvent"
kBT:       double [epsilon] "temperature"
alpha_v:   double "correction coefficient of V"
alpha_o:   double "correction coefficient of Omega"
```

DXは格子幅，RHOは流体の密度，ETAは流体の粘度，kBTは粒子温度，alpha_vは粒子速度の調整パラメータ，alpha_oは粒子角速度の調整パラメータである．

また，External_field構造体では，ジグザグせん断流のパラメータを設定する:

```
External_field: {
  type: select {"DC","AC"}
  DC: {
    Shear_rate:    double [1/tau] "shear rate"
  }
  AC: {
    Shear_rate:    double [1/tau] "shear rate"
    Frequency:     double [1/tau] "alternating frequency"
  }
}
```

typeセレクタでDCを選ぶと定常せん断流となり，Shear_rateで定常せん断速度を設定できる．一方，ACを選ぶと振動せん断流となり，Shear_rateで最大振動せん断速度，Frequencyで周波数を設定できる．

Shear_Navier_Stokes_Lees_Edwards

Shear_Navier_Stokes_Lees_Edwardsを選ぶと，Lees–Edwards境界条件でのせん断流下でNewton流体中に分散した粒子のシミュレーションを行うことができる(第4章)． Shear_Navier_Stokes_Lees_Edwards構造体では，流体に関する以下のパラメータを設定する:

```
DX:        double [L] "lattice spacing (=1), fixed for all directions"
RHO:       double [rho] "mass density of solvent"
ETA:       double [eta] "shear viscosity of solvent"
kBT:       double [epsilon] "temperature"
alpha_v:   double "correction coefficient of V"
alpha_o:   double "correction coefficient of Omega"
```

DXは格子幅，RHOは流体の密度，ETAは流体の粘度，kBTは粒子温度，alpha_vは粒子速度の調整パラメータ，alpha_oは粒子角速度の調整パラメータである．

External_field構造体では，Lees–Edwards境界条件でのせん断流のパラメータを設定する:

```
External_field: {
  type: select {"DC","AC"}
  DC: {
    Shear_rate:   double [1/tau] "shear rate"
  }
  AC: {
    Shear_rate:   double [1/tau] "shear rate"
    Frequency:    double [1/tau] "alternating frequency"
```

```
  }
}
```

typeセレクタでDCを選ぶと定常せん断流となり，Shear_rateで定常せん断速度を設定できる． ACを選ぶと振動せん断流となり，Shear_rateで最大振動せん断速度，Frequencyで周波数を設定できる．

Electrolyte

Electrolyteを選ぶと，電解質溶液に分散した荷電コロイド粒子のシミュレーションを行うことができる(第5章)． Electrolyte構造体では，流体に関する以下のパラメータを設定する:

```
DX:         double [L] "lattice spacing (=1), fixed for all directions"
RHO:        double [rho] "mass density of solvent"
ETA:        double [eta] "shear viscosity of solvent"
kBT:        double [epsilon] "temperature"
alpha_v:    double "correction coefficient of V"
alpha_o:    double "correction coefficient of Omega"
Dielectric_cst:    double "dielectric constant"
INIT_profile: select {
  "Uniform",
  "Poisson_Boltzmann"
} "Initial condition for density profile of ions"
```

DXは格子幅，RHOは流体の密度，ETAは流体の粘度，kBTは粒子温度，alpha_vは粒子速度の調整パラメータ，alpha_oは粒子角速度の調整パラメータ，Dielectric_cstは溶媒の誘電率である． INIT_profileセレクタではイオン分布の初期設定をUniformとPoisson_Boltzmannから選択する．

Add_salt構造体では塩イオンに関する設定ができる:

```
Add_salt: {
  type:select {"saltfree","salt"}
  saltfree: {
    Valency_counterion:         double "valency of counterion"
    Onsager_coeff_counterion:   double "Onsager coefficient of counterion"
  }
  salt: {
    Valency_positive_ion:       double "valency of positive ion"
    Valency_negative_ion:       double "valency of negative ion"
    Onsager_coeff_positive_ion: double "Onsager coefficient of positive ion"
    Onsager_coeff_negative_ion: double "Onsager coefficient of negative ion"
    Debye_length:               double "Debye screening length in the unit of DX"
  }
}
```

typeセレクタでsaltfreeを選ぶと対イオンのみのシミュレーションとなり，saltを選ぶと対イオンに加えて正負2種類の塩イオンを設定できる． saltfreeを選ぶとき，Valency_counterionで対イオンの価数，Onsager_coeff_counterionで対イオンのOnsager輸送係数を設定する．一方，saltを選ぶとき，Valency_positive_ionで正イオンの価数，Valency_negative_ionで負イオンの価数，Onsager_coeff_positive_ionで正イオンのOnsager係数，Onsager_coeff_negative_ionで負イオンのOnsager係数，Debye_lengthでDebye遮蔽長を設定する．

Electric_field構造体では外部電場の設定ができる:

```
Electric_field: {
  type: select {"ON","OFF"}
  ON: {
    type: select {"DC","AC"}
    DC: {
      Ex:  double
      Ey:  double
      Ez:  double
    }
```

```
    AC: {
      Ex:  double
      Ey:  double
      Ez:  double
      Frequency:   double
    }
  }
}
```

typeセレクタでONを選ぶと外部電場が印加され，OFFでは外部電場を印加しないシミュレーションとなる．外部電場の種類としてDCを選択すると，Ex，Ey，Ezでx，y，z方向の電場の強さをそれぞれ設定できる．一方，ACを選択すると，Ex，Ey，Ezでx，y，z方向の電場の強さをそれぞれ設定できるのに加えて，Frequencyで交流電場の周波数が設定できる．

Navier_Stokes_FDM

Navier_Stokes_FDMを選ぶと，差分法によるNewton流体中に分散した粒子のシミュレーションを行うことができる(付録D)． Navier_Stokes_FDM構造体では，まずNS_solver構造体でNavier–Stokes方程式の解法に関する設定を行う:

```
NS_solver: {
  type: select {
    'explicit_scheme',
    'implicit_scheme'
  } "explicit_scheme: explicit MAC scheme, ON: implicit MAC scheme"
  implicit_scheme: {
    tolerance:         double "stopping criteria"
    maximum_iteration: int "number of maximum iteration"
  }
}
```

typeセレクタでexplicit_schemeを選ぶと陽解法が，implicit_schemeを選ぶと陰解法が用いられる．陰解法の場合，toleranceで収束判定基準，maximum_iterationで最大反復回数を設定する．

また，Navier_Stokes_FDM構造体では，流体に関する以下のパラメータを設定する:

```
DX:       double [L] "lattice spacing (=1), fixed for all directions"
RHO:      double [rho] "mass density of solvent"
ETA:      double [eta] "shear viscosity of solvent"
kBT:      double [epsilon] "temperature"
alpha_v:  double "correction coefficient of V"
alpha_o:  double "correction coefficient of Omega"
```

DXは格子幅，RHOは流体の密度，ETAは流体の粘度，kBTは粒子温度，alpha_vは粒子速度の調整パラメータ，alpha_oは粒子角速度の調整パラメータである．

Navier_Stokes_Cahn_Hilliard_FDM

Navier_Stokes_Cahn_Hilliard_FDMを選ぶと，二成分相分離流体中に分散した粒子のシミュレーションを行うことができる(第6章)． Navier_Stokes_Cahn_Hilliard_FDM構造体では，まずNS_solver構造体でNavier–Stokes方程式の解法に関する設定を行う:

```
NS_solver: {
  type: select {
    'explicit_scheme',
    'implicit_scheme'
  } "explicit_scheme: explicit MAC scheme, ON: implicit MAC scheme"
  implicit_scheme: {
    tolerance:         double "stopping criteria"
    maximum_iteration: int "number of maximum iteration"
    viscosity_change: select {'ON','OFF'}
    ON: {
      ETA_A:            double [eta] "shear viscosity of solvent A"
```

```
      ETA_B:            double [eta] "shear viscosity of solvent B"
    }
  }
}
```

typeセレクタでexplicit_schemeを選ぶと陽解法が，implicit_schemeを選ぶと陰解法が用いられる．陰解法では，toleranceで収束判定基準，maximum_iterationで最大反復回数を設定する．さらに，viscosity_changeセレクタでは，二成分相分離流体において異なる粘度を導入できる．ONにすると，ETA_AとETA_BでA成分とB成分の流体の粘度を個別に設定できる．OFFの場合，後述するETAの値が両方の流体の粘度として設定される．

また，CH_solver構造体では，Cahn–Hilliard方程式の解法に関する設定を行う:

```
CH_solver: {
  type: select {
    'explicit_scheme',
    'implicit_scheme'
  } "explicit_scheme: explicit Euler scheme, ON: implicit BDFAB scheme"
  implicit_scheme: {
    tolerance:         double "stopping criteria"
    maximum_iteration: int "number of maximum iteration"
  }
}
```

typeセレクタでexplicit_schemeを選ぶと陽解法が，implicit_schemeを選ぶと陰解法が用いられる．陰解法では，toleranceで収束判定基準，maximum_iterationで最大反復回数を設定する．

また，Navier_Stokes_Cahn_Hilliard_FDM構造体では，流体に関する以下のパラメータを設定する:

```
DX:       double [L] "lattice spacing (=1), fixed for all directions"
RHO:      double [rho] "mass density of solvent"
ETA:      double [eta] "shear viscosity of solvent"
kBT:      double [epsilon] "temperature"
alpha_v:  double "correction coefficient of V"
alpha_o:  double "correction coefficient of Omega"
```

DXは格子幅，RHOは流体の密度，ETAは流体の粘度，kBTは粒子温度，alpha_vは粒子速度の調整パラメータ，alpha_oは粒子角速度の調整パラメータである．

Potential構造体は相分離ポテンシャルの設定である．ポテンシャルは，typeセレクタにおいてLandauとFlory_Hugginsから選べる:

```
type: select {'Landau','Flory_Huggins'}
```

Landauを選ぶと，Landau型二重井戸型ポテンシャルとなり，さらに以下のパラメータを設定する:

```
composition_ratio:     double "composition ratio of A and B fluids"
initial_fluctuation:   double "initial fluctuation of concentration"
a:                     double "GL parameter (third order term)"
b:                     double "GL parameter (first order term)"
d:                     double "penalty factor in fictitious particle domain"
w:                     double "penalty factor on particle surface domain"
z:                     double "penalty factor on fluid interface"
psi_0:                 double "psi_0 for particle"
alpha:                 double "surface parameter on fluid-fluid surface"
kappa:                 double "mobility parameter"
```

composition_ratioは二成分相分離流体の組成比，initial_fluctuationは初期濃度揺らぎの大きさ，aは3次の項の係数，bは1次の項の係数，dは粒子部分の流体領域の組成をpsi_0に近づけるための項の大きさのパラメー

タ，wは粒子界面の親和性を決めるパラメータ，zは粒子界面と流体界面の親和性を決めるパラメータ，psi_0は粒子内部のψの値を指定するパラメータ，alphaは流体界面パラメータ，kappaは易動度である．

一方，Flory_Hugginsを選ぶと，Flory–Huggins型二重井戸型ポテンシャルとなり，さらに以下のパラメータを設定する:

```
composition_ratio:     double "composition ratio of A and B fluids"
initial_fluctuation:   double "initial fluctuation of concentration"
na:            double "(reduced) number of polymerization of A component"
nb:            double "(reduced) number of polymerization of B component"
chi:           double "Flory's interaction parameter (chi parameter)"
d:             double "penalty factor in fictitious particle domain"
w:             double "penalty factor on particle surface domain"
z:             double "penalty factor on fluid interface"
psi_0:         double "psi_0 for particle"
alpha:         double "surface parameter on fluid-fluid surface"
kappa:         double "mobility parameter"
```

composition_ratioは二成分相分離流体の組成比，initial_fluctuationは初期濃度揺らぎの大きさ，naはA成分の分子重合度，nbはB成分の分子重合度，chiはFloryの相互作用パラメータ，dは粒子部分の流体領域の組成をpsi_0に近づけるための項の大きさのパラメータ，wは粒子界面の流体の親和性を決めるパラメータ，zは粒子界面と流体界面の親和性を決めるパラメータ，psi_0は粒子内部のψの値を指定するパラメータ，alphaは流体界面パラメータ，kappaは易動度である．

Wall_Potential構造体は平面壁の設定である:

```
type: select {'ON', 'OFF'}
ON:{
  w :        double "penalty factor on wall surface domain"
  psi_0: {
    magnitude:   double "psi_0 magnitude"
    profile : select {'uniform', 'user_specify'}
    user_specify:{
      PSI0[][]:{
        value :  double
      }
    }
  }
  DRYING:{
    type: select {'ON', 'OFF'}
    ON:{
      psi_dry : double "psi_dry"
    }
  }
}
```

typeセレクタでONを選ぶと平面壁が設定できる．このとき，wは平面壁の流体の親和性を決めるパラメータ，psi_0は平面壁内部のψの値を指定するパラメータである． psi_0では，magnitudeで平面壁と流体の親和性の大きさ，profileでuniform(平面壁面上で均一)，user_specify(平面壁面上の流体の親和性はユーザによって指定される)が設定できる． profileでuser_specifyを設定した場合，PSI0[][]にて値を手動設定できる．DRYINGのtypeセレクタでONを選ぶと乾燥のシミュレーションができる．このとき，psi_dryは乾燥速度を決めるパラメータである．

Shear_Navier_Stokes_Lees_Edwards_FDM

Shear_Navier_Stokes_Lees_Edwards_FDMを選ぶと，差分法によるLees–Edwards境界条件でのせん断流下でNewton流体中に分散した粒子のシミュレーションを行うことができる(付録D)． Shear_Navier_Stokes_Lees_Edwards_FDM構造体では，まずNS_solver構造体でNavier–Stokes方程式の解法に関する設定を行う:

```
NS_solver: {
  type: select {
    'explicit_scheme',
    'implicit_scheme'
  } "explicit_scheme: explicit MAC scheme, ON: implicit MAC scheme"
  implicit_scheme: {
    tolerance:         double "stopping criteria"
    maximum_iteration: int "number of maximum iteration"
  }
}
```

typeセレクタでexplicit_schemeを選ぶと陽解法が，implicit_schemeを選ぶと陰解法が用いられる．陰解法では，toleranceで収束判定基準，maximum_iterationで最大反復回数を設定する．

また，Shear_Navier_Stokes_Lees_Edwards_FDM構造体では，流体に関する以下のパラメータを設定する:

```
DX:       double [L] "lattice spacing (=1), fixed for all directions"
RHO:      double [rho] "mass density of solvent"
ETA:      double [eta] "shear viscosity of solvent"
kBT:      double [epsilon] "temperature"
alpha_v:  double "correction coefficient of V"
alpha_o:  double "correction coefficient of Omega"
```

DXは格子幅，RHOは流体の密度，ETAは流体の粘度，kBTは粒子温度，alpha_vは粒子速度の調整パラメータ，alpha_oは粒子角速度の調整パラメータである．

External_field構造体では，Lees–Edwards境界条件でのせん断流のパラメータを設定する．

```
External_field: {
  type: select {"DC","AC"}
  DC: {
    Shear_rate:    double [1/tau] "shear rate"
  }
  AC: {
    Shear_rate:    double [1/tau] "shear rate"
    Frequency:     double [1/tau] "alternating frequency"
  }
}
```

typeセレクタでDCを選ぶと定常せん断流となり，Shear_rateで定常せん断速度を設定できる． ACを選ぶと振動せん断流となり，Shear_rateで最大振動せん断速度，Frequencyで周波数を設定できる．

Shear_NS_LE_CH_FDM

Shear_NS_LE_CH_FDMを選ぶと，Lees–Edwards境界条件でのせん断流下で二成分相分離流体中に分散した粒子のシミュレーションを行うことができる(第6章)． Shear_NS_LE_CH_FDM構造体では，まずNS_solver構造体でNavier–Stokes方程式の解法に関する設定を行う:

```
NS_solver: {
  type: select {
    'explicit_scheme',
    'implicit_scheme'
  } "explicit_scheme: explicit MAC scheme, ON: implicit MAC scheme"
  implicit_scheme: {
    tolerance:         double "stopping criteria"
    maximum_iteration: int "number of maximum iteration"
    viscosity_change: select {'ON','OFF'}
      ON: {
        ETA_A:            double [eta] "shear viscosity of solvent A"
        ETA_B:            double [eta] "shear viscosity of solvent B"
      }
  }
}
```

typeセレクタでexplicit_schemeを選ぶと陽解法が，implicit_schemeを選ぶと陰解法が用いられる．

陰解法では，toleranceで収束判定基準，maximum_iterationで最大反復回数を設定する．さらに，viscosity_changeセレクタでは二成分相分離流体において異なる粘度を導入できる．ONにすると，ETA_AとETA_BでA成分とB成分の流体の粘度を個別に設定できる．OFFの場合，後述するETAの値が両方の流体の粘度として設定される．

また，CH_solver構造体では，Cahn–Hilliard方程式の解法に関する設定を行う:

```
CH_solver: {
  type: select {
    'explicit_scheme',
    'implicit_scheme'
  } "explicit_scheme: explicit Euler scheme, ON: implicit BDFAB scheme"
  implicit_scheme: {
    tolerance:         double "stopping criteria"
    maximum_iteration: int "number of maximum iteration"
  }
}
```

typeセレクタでexplicit_schemeを選ぶと陽解法が，implicit_schemeを選ぶと陰解法が用いられる．陰解法では，toleranceで収束判定基準，maximum_iterationで最大反復回数を設定する．

また，Shear_NS_LE_CH_FDM構造体では，流体に関する以下のパラメータを設定する:

```
DX:             double [L] "lattice spacing (=1), fixed for all directions"
RHO:            double [rho] "mass density of solvent"
ETA:            double [eta] "shear viscosity of solvent"
kBT:            double [epsilon] "temperature"
alpha_v:         double "correction coefficient of V"
alpha_o:         double "correction coefficient of Omega"
```

DXは格子幅，RHOは流体の密度，ETAは流体の粘度，kBTは粒子温度，alpha_vは粒子速度の調整パラメータ，alpha_oは粒子角速度の調整パラメータである．

Potential構造体は相分離ポテンシャルの設定である．ポテンシャルは，typeセレクタにおいてLandauとFlory_Hugginsから選べる:

```
type: select {'Landau','Flory_Huggins'}
```

Landauを選ぶと，Landau型二重井戸型ポテンシャルとなり，さらに以下のパラメータを設定する:

```
composition_ratio:     double "composition ratio of A and B fluids"
initial_fluctuation:   double "initial fluctuation of concentration"
a:     double "GL parameter (third order term)"
b:     double "GL parameter (first order term)"
d:     double "penalty factor in fictitious particle domain"
w:     double "penalty factor on particle surface domain"
z:     double "penalty factor on fluid interface"
psi_0: double "psi_0 for particle"
alpha: double "surface parameter on fluid-fluid surface"
kappa: double "mobility parameter"
```

composition_ratioは二成分相分離流体の組成比，initial_fluctuationは初期濃度揺らぎの大きさ，aは3次の項の係数，bは1次の項の係数，dは粒子部分の流体領域の組成をpsi_0に近づけるための項の大きさのパラメータ，wは粒子界面の親和性を決めるパラメータ，zは粒子界面と流体界面の親和性を決めるパラメータ，psi_0は粒子内部のψの値を決めるパラメータ，alphaは流体界面パラメータ，kappaは易動度である．

一方，Flory_Hugginsを選ぶと，Flory–Huggins型二重井戸型ポテンシャルとなり，さらに以下のパラメータを設定する:

```
composition_ratio:     double "composition ratio of A and B fluids"
initial_fluctuation:   double "initial fluctuation of concentration"
na:    double "(reduced) number of polymerization of A component"
nb:    double "(reduced) number of polymerization of B component"
chi:   double "Flory's interaction parameter (chi parameter)"
d:     double "penalty factor in fictitious particle domain"
w:     double "penalty factor on particle surface domain"
z:     double "penalty factor on fluid interface"
psi_0: double "psi_0 for particle"
alpha: double "surface parameter on fluid-fluid surface"
kappa: double "mobility parameter"
```

composition_ratioは二成分相分離流体の組成比，initial_fluctuationは初期濃度揺らぎの大きさ，naはA成分の分子重合度，nbはB成分の分子重合度，chiはFloryの相互作用パラメータ，dは粒子部分の流体領域の組成をpsi_0に近づけるための項の大きさのパラメータ，wは粒子界面の流体の親和性を決めるパラメータ，zは粒子界面と流体界面の親和性を決めるパラメータ，psi_0は粒子内部のψの値を決めるパラメータ，alphaは流体界面パラメータ，kappaは易動度である．

External_field構造体では，Lees–Edwards境界条件でのせん断流のパラメータを設定する:

```
External_field: {
  type: select {"DC","AC"}
  DC: {
    Shear_rate:    double [1/tau] "shear rate"
  }
  AC: {
    Shear_rate:    double [1/tau] "shear rate"
    Frequency:     double [1/tau] "alternating frequency"
  }
}
```

typeセレクタでDCを選ぶと定常せん断流となり，Shear_rateで定常せん断速度を設定できる． ACを選ぶと振動せん断流となり，Shear_rateで最大振動せん断速度，Frequencyで周波数を設定できる．

A.2　オブジェクトの設定

流体に分散させるオブジェクトは，object_type構造体で設定される． typeセレクタでオブジェクトの種類を選ぶ:

```
type: select {'spherical_particle','chain','rigid'}
```

spherical_particle

spherical_particleを選ぶと，球状粒子が分散したシミュレーションを行うことができる．この場合，さらに以下のパラメータを設定する:

```
Particle_spec[]:{
  Particle_number: int "number of colloidal particles"
  MASS_RATIO: double "mass density ratio colloid/solvent"
  Surface_charge: double "surface charge of colloid"
  janus_axis: select {'NONE', 'X', 'Y', 'Z'} "janus axis in body_fixed frame"
  janus_propulsion: select{'OFF', 'TUMBLER', 'SQUIRMER', 'OBSTACLE'}
  janus_force: Vector3d "self-propulsion force"
  janus_torque: Vector3d "self-propulsion torque"
  janus_slip_vel: float "Slip velocity coeff B1"
  janus_slip_mode: float "Blake squirmer mode B2/B1"
  janus_rotlet_C1: float "rotlet coefficient C1"
  janus_rotlet_dipole_C2: float  "rotlet dipole C2"
}
```

Particle_numberでは粒子数，MASS_RATIOでは粒子の流体に対する密度比を設定する． Surface_chargeは粒子表面電荷の設定であり，constitutive_eq構造体のtypeセレクタで，Electrolyteを選んだときに有効なパラメータである．

janus_*はJanus粒子に関する設定である． janus_axisセレクタでは粒子に固定された座標系におけるJanus軸の方向をNONE，X，Y，Zから選ぶ． janus_propulsionセレクタではJanus粒子の推進運動の指定ができ，OFF(無効)，TUMBLER(一定の外力で推進軸方向に推進する粒子)，SQUIRMER(Slip境界条件により推進軸方向に推進するスクワマー粒子)，OBSTACLE(固定された障害物粒子)から選択する．janus_force.x，janus_force.y，janus_force.zはそれぞれJanus粒子の推進力のx，y，z成分の設定である．また，janus_torque.x，janus_torque.y，janus_torque.zはそれぞれJanus粒子の推進トルクのx，y，z成分の設定である． janus_slip_velはJanus粒子の表面滑り速度を決めるパラメータB_1，janus_slip_modeはJanus粒子の泳動形態を決定するパラメータB_2/B_1，janus_rotlet_C1は双極子型の力を流体に及ぼすJanus粒子のパラメータC_1，janus_rotlet_dipole_C2はrotlet双極子型の力を流体に及ぼすJanus粒子のパラメータC_2である．

chain

chainを選ぶと，フレキシブルな粒子鎖が分散したシミュレーションを行うことができる．フレキシブルな粒子鎖では，さらに以下のパラメータを設定する:

```
chain:{
  Chain_spec[]:{
    Beads_number: int "number of beads in a chain"
    Chain_number: int "number of chains"
    MASS_RATIO: double "mass density ratio chain/solvent"
    Surface_charge: double "surface charge of colloid"
    janus_axis: select {'NONE', 'X', 'Y', 'Z'} "janus axis in body_fixed frame"
  }
}
```

Beads_numberは1本の鎖に属するビーズの数，Chain_numberは鎖の本数，MASS_RATIOはビーズの流体に対する密度比である．また，Surface_chargeは粒子表面電荷の設定であり，constitutive_eq構造体のtypeセレクタで，Electrolyteを選んだときに有効なパラメータである． janus_axisセレクタでは，ビーズに固定された座標系におけるJanus軸の方向をNONE，X，Y，Zから選ぶ.

rigid

rigidを選ぶと，球形のビーズで構成された剛体が分散したシミュレーションを行うことができる．剛体では以下のパラメータを設定する必要がある．

```
rigid:{
  Rigid_spec[]:{
    Beads_number: int "number of beads in a rigid"
    Rigid_number: int "number of rigid bodies"
    MASS_RATIO: double "mass density ratio rigid body/solvent"
    Surface_charge: double "surface charge of particle"
    Rigid_motion: select {'fix','free'}
    Rigid_velocity: Vector3d "speed of translation ### fix only ###"
    Rigid_omega: Vector3d "angular velocity ### fix only ###"
  }
}
```

Beads_numberは1つの剛体に属するビーズの数，Rigid_numberは剛体の数，MASS_RATIOはビーズの流体に対する密度比である．また，Surface_chargeは粒子表面電荷の設定であり，constitutive_eq構造体のtypeセレクタで，Electrolyteを選んだときに有効なパラメータである． Rigid_motionでは，剛体の運動に関する6つの自由度の全てをfree (自由運動)とするかfix (指定した速度と角速度に固定)とするかを選ぶ．Rigid_velocity.x，Rigid_velocity.y，Rigid_velocity.z，はそれぞれ剛体の速度のx，y，z成分（ラボフレーム）の設定である．また，Rigid_omega.x，Rigid_omega.y，Rigid_omega.z，はそれぞれ剛体の角速度

のx，y，z成分（ラボフレーム）の設定である． 6つの自由度を個別に解放/固定するには，「選択できる機能の設定」の項で説明する switch.free_rigid オプションを同時に使用します。

A.3　共通のシミュレーション設定

A.1章とA.2章で，それぞれ流体とオブジェクトに関する設定を行った．本章では，それらに共通のシミュレーション条件の設定について説明する．

KAPSELで扱うオブジェクトは粒子から構成される．その一次粒子の共通パラメータとして，以下を設定する:

```
A_XI: double "interface thickness in the unit of DX"
A: double "colloid radius in the unit of DX"
```

A_XIとAは，それぞれ粒子界面の厚さと粒子半径を格子幅DXを単位として設定する．

系に印加される重力はgravity構造体で設定される:

```
gravity: {
  G: double [L*tau^-2] "gravitational acceleration constant"
  G_direction: select {'-X','-Y','-Z'} "direction of gravitational acceleration"
}
```

Gは重力加速度である．また，G_directionセレクタは，重力が印加される方向を-X，-Y，-Zから選ぶ．

次に，粒子間に働くポテンシャルを設定する．設定項目は以下の通りである:

```
EPSILON: double [epsilon] "Lennard-Jones depth"
LJ_powers: select {'12:6','24:12','36:18','DLVO','electro_osmotic_flow'} "set of power
 ↪  exponents of LJ potential"
```

EPSILONでは，Lennard-Jonesポテンシャルのエネルギーの単位を指定する． LJ_powersでは，粒子間ポテンシャルを，12:6，24:12，36:18 (Lennard-Jonesポテンシャル)，DLVO (DLVOポテンシャル)あるいは，electro_osmotic_flow (クインケローラ以外では使用しない)から選ぶ．

mesh構造体では，流体計算の計算点を定義するメッシュサイズの指定を行う:

```
mesh: {
  NPX: int "number of mesh in x-direction = 2^NPX"
  NPY: int "number of mesh in y-direction = 2^NPY"
  NPZ: int "number of mesh in z-direction = 2^NPZ"
}
```

NPX，NPY，NPZはそれぞれx，y，z方向におけるメッシュ数を決めるパラメータであり，これにより各方向のメッシュ数は$L_x = 2^{\mathrm{NPX}}$，$L_y = 2^{\mathrm{NPY}}$，$L_z = 2^{\mathrm{NPZ}}$と設定される．

時間刻みの設定

time_increment構造体では，シミュレーションの時間刻みを設定する:

```
time_increment: {
  type: select {"auto","manual"}
  auto: {
    factor: double "delta_t = factor * h(determined by system parameters)"
  }
  manual: {
    delta_t: double [tau]
  }
}
```

typeセレクタでautoを選べば時間刻みの上限である$T_{dump} = \rho/\eta k_{max}^2$が時間刻みとして自動設定される．ここ

でk_{max}は格子幅Δで決まる最大波数である．またこのとき，factorで時間刻みの係数を設定でき，時間刻みは$\Delta t = T_{dump} \times$ factorとなる．一方，manualを選べば，delta_tで設定した値が時間刻みとして設定される．

A.4　選択できる機能の設定

本章では，switch構造体で定義される選択できる機能の設定について，順に説明する．

ROTATIONセレクタでは，粒子の回転を考慮するかどうかをONかOFFで選ぶ:

```
ROTATION: select {'ON','OFF'} "OFF: not solve rotation, ON: solve rotation"
```

LJ_truncateセレクタでは，粒子間に働くLennard-Jonesポテンシャルによる力について，引力項を含む通常の形OFF，引力項を含まない斥力のみON，まったく力を加えないNONEから選ぶ:

```
LJ_truncate: select {'ON','OFF','NONE'} "OFF:normal LJ, ON:WCA, NONE: no-interaction
 ↪  at all"
```

INIT_distribution構造体では粒子の初期配置を設定できる．

```
INIT_distribution: {
  type: select {
    'uniform_random',
    'random_walk',
    'FCC',
    'BCC',
    'user_specify'
  }
  "uniform_random:distributed uniformly in box, random_walk:distributed uniformly in
   ↪  box, FCC:distributed on FCC lattice, BCC:distributed on BCC lattice,
   ↪  user_specify:configuration and velocity specified by user"
  random_walk: {
    iteration: int
  }
  user_specify: {
    Particles[]: Particle
  }
}
```

typeセレクタで初期粒子配置をuniform_random (ランダム)，random_walk (正方格子上からランダムにずれている)， FCC (FCC格子上)，BCC (BCC格子上)，user_specify (座標と速度はユーザによって指定される)から選ぶ．random_walkを選択した場合，iterationでランダムウォークの試行回数を設定する．また，user_specifyを選択した場合，Particles[]で粒子位置と速度を設定する．

INIT_orientationセレクタでは，粒子の初期配向を，user_specify(座標と速度はユーザによって指定される)，random(ランダム)，space_align(1方向に配向)から選ぶ:

```
INIT_orientation: select {'user_specify', 'random', 'space_align'}
```

user_specifyを選んだ場合，粒子の初期配向をParticles[]に設定する．

SLIP_tolは，Janus粒子界面における接戦方向の滑り速度を導入する際の，流体流れ場の反復計算の収束判定基準の設定である:

```
SLIP_tol: float "Tolerance for iterative slip convergence"
```

また，SLIP_iterは，Janus粒子界面における接戦方向の滑り速度を導入する際の，流体流れ場の反復計算の最大反復回数の設定である:

```
  SLIP_iter: int "Maximum number of iterations for iterative slip convergence"
```

FIX_CELL構造体はシミュレーションセルに関する設定である:

```
FIX_CELL: {
  x: select{'ON','OFF'}"OFF:w/o DC current, ON:with DC current"
  y: select{'ON','OFF'}"OFF:w/o DC current, ON:with DC current"
  z: select{'ON','OFF'}"OFF:w/o DC current, ON:with DC current"
}
```

x，y，zセレクタでは，それぞれx，y，z方向の全速度の直流成分を0とするかどうか，ONとOFFから選ぶ.

pin構造体では，粒子運動の自由度の設定ができる:

```
pin: {
  type: select{"NO","YES"}
  YES:{
    pin[]: int
    pin_rot[]: int
  }
}
```

typeセレクタでYESを選ぶと粒子を固定できる．このとき，並進運動させない粒子番号と回転運動させない粒子番号をそれぞれpin[]とpin_rot[]で設定する.

free_rigid構造体では，剛体の並進/回転運動の6つの自由度を個別に自由運動としたり，指定値に固定することができる．このオプションは，並進/回転運動の6つの自由度のすべてをまとめて自由にするか指定値に固定するobject_type.rigid.Rigid_spec オプションと同時に使用する必要がある.

```
free_rigid:{
  type: select{'NO', 'YES'}  "Free rigid degrees of freedom"
  YES:{
    DOF[]:{
      spec_id: int           "Rigid body species id"
      vel:{
        x:select{'NO', 'YES'},
        y:select{'NO', 'YES'},
        z:select{'NO', 'YES'}
      } "Free velocity components"
      omega:{
        x:select{'NO', 'YES'},
        y:select{'NO', 'YES'},
        z:select{'NO', 'YES'}
      } "Free omega components"
    }
  }
}
```

typeセレクタでYESを選ぶと，DOF[]で各剛体の運動の自由度を個別に設定できる． spec_idでは設定する剛体番号を指定する． vel構造体のx，y，zセレクタでは，それぞれx，y，z方向の並進運動の自由度をYES (自由運動)，NO (指定値に固定)から選択できる．また，omegaでは構造体の，x，y，zセレクタで，それぞれx，y，z方向の回転運動の自由度をYES (自由運動)，NO (指定値に固定)から選択できる.

ns_solver構造体は，Navier–Stokes方程式の解法に関する設定である:

```
ns_solver:{
  OBL_INT: select {'linear', 'spline'} "interpolation scheme for Oblique/Rectangular
   ↪  transform"
}
```

OBL_INTセレクタでは，せん断流下のシミュレーションにおける座標系変換時の近似関数をlinear(線形近似)とspline(スプライン近似)から選ぶ.

wall構造体は，平面壁の設定である:

```
wall:{
  type: select{'NONE', 'FLAT'}
  FLAT: {
    axis: select{'X', 'Y', 'Z'}"perpendicular axis to flat parallel walls"
    DH: int "wall thickness in number of grid points"
    LJ_Params: select{"AUTO", "MANUAL"}
    MANUAL:{
      truncate: select{'ON', 'OFF', 'NONE'} "Truncate OFF: attractive force, ON: no
       ↪  attractive force"
      powers: select{'12:6','24:12','36:18'} "type of LJ potential"
      EPSILON: double "LJ parameter, default (Basically, a large value.) "
    }
  }
}
```

typeセレクタでFLATを選ぶと，平面壁を設定できる．このとき，axisセレクタで平面壁の法線方向をX，Y，Zで指定する．また，DHで平面壁の厚みを格子点の数で指定する． LJ_Paramsセレクタは平面壁に働くポテンシャルの設定である． AUTOを選ぶと粒子間ポテンシャルと同じポテンシャルが設定される． MANUALを選ぶと，粒子間ポテンシャルとは異なるポテンシャルを設定できる． truncateセレクタでは，平面壁での引力の有無を設定する． ONにすると斥力のみが働くポテンシャルとなり，OFFにすると引力が働くポテンシャルとなる． powersセレクタでは，平面壁でのLennard-Jonesポテンシャルのべき指数を12:6，24:12，36:18から選べる． EPSILONでは，平面壁でのLennard-Jonesポテンシャルのエネルギーの単位を設定する．

quincke構造体はクインケローラーの設定である:

```
quincke:{
  type: select{'ON','OFF'}
  ON: {
    e_dir: select{'X', 'Y', 'Z'}"constraint axis(the direction of external electric
     ↪  field E)"
    w_dir: select{'X', 'Y', 'Z'}"the direction of constant angular velocity vector(body
     ↪  frame)"
    torque_amp: double "the amplitude of constraint torque"
  }
}
```

typeセレクタでONを指定すると，クインケローラーのシミュレーションができる．このとき，e_dirセレクタでは外部電場を印加する方向をX，Y，Zで指定する． w_dirセレクタではクインケ効果による回転の角速度ベクトルの方向をX，Y，Zで指定する． torque_ampではトルクの大きさを設定する．

multipole構造体は，Ewald法によるシミュレーションの設定である:

```
multipole:{
  type: select{'ON', 'OFF'}
  ON:{
    Dipole:{
      type : select{'ON', 'OFF'}
      ON:{
        magnitude: double "the dipole strength"
        type     : select{'FIXED', 'QUINCKE'} "type of dipole"
        FIXED:{
          dir   : select{'X', 'Y', 'Z'} "direction of dipolar axis"
        }
        QUINCKE:{
          type : select{'with_mirror_image','no_mirror_image'} "Mirror image component
           ↪  of electrode surface"
          Pz_factor : double "Pz strength += Pz_factor*magnitude"
        }
      }
    }
    EwaldParams:{
```

```
      alpha   : double "Ewald screening parameter"
      delta   : double "Tolerance parameter, used to determine k_max"
      converge: double "Convergence parameter (fraction of k vectors to consider)"
      epsilon : double "Permittivity at boundary ( if  negative, set to tinfoil)"
    }
  }
}
```

typeセレクタでONを指定すると，Ewald法によるシミュレーションができる． Dipole構造体は双極子に関する設定である．双極子を扱う場合，typeセレクタをONに設定する． magnitudeでは双極子モーメントの大きさを設定する．双極子の種類を，typeセレクタでFIXED (固定)とQUINCKE (クインケ回転)から選ぶ． FIXEDを選ぶ場合，dirにおいて双極子を固定する方向をX，Y，Zで指定する．一方，QUINCKEを選ぶ場合，電極表面の鏡像成分の有無をwith_mirror_image (有り)，no_mirror_image (無し)から選ぶ．また，Pz_factorは双極子モーメントのZ成分の大きさを与えるパラメータである． EwaldParams構造体ではEwald法のパラメータを設定する． alphaは遮蔽パラメータ，deltaは$k_{\max}$を決定するための収束判定基準，convergeは収束パラメータ，epsilonは，境界での誘電率である．

A.5　データ出力設定

本章では，KAPSELのデータ出力設定について説明する．これはoutput構造体で設定される:

```
output: {
  GTS: int "interval between snapshots"
  Num_snap: int "number of snapshots"
  AVS: select {"ON","OFF"}
  ON:{
    Out_dir: string "directory name"
    Out_name: string "prefix name for data file"
    FileType: select {"BINARY","ASCII","EXTENDED"} "output data type"
    EXTENDED:{
      Driver:{
        Format: select {"HDF5"}
      }
      Print_field:{
        Crop:select{"YES", "NO"}       "Crop Field Data to Hyperslab"
        YES:{
          Slab_x: SlabSelection
          Slab_y: SlabSelection
          Slab_z: SlabSelection
        }
        Vel: select{"YES", "NO"}       "Print velocity field"
        Phi: select{"YES", "NO"}       "Print phi field"
        Charge: select{"YES", "NO"}    "Print charge fields (surface & solute charge &
         ↪  potential)"
        Pressure: select{"YES", "NO"}  "Print pressure field"
        Tau: select{"YES", "NO"}       "Print stress tensor"
      }
    }
  }
  UDF: select {"ON","OFF"}
}
```

GTSでは，データ出力のインターバルのステップ数を設定する． Num_snapでは，データ出力の回数を設定する．つまり全ステップ数はGTS × Num_snapで決まる． AVSセレクタでONを選ぶと，AVS形式のデータを出力できる． ONのとき，Out_dirでAVS形式のデータを出力するディレクトリを指定する．また，Out_nameでAVS形式の出力データのファイル名の接頭語を指定する． FileTypeセレクタでは，AVSデータファイルのフォーマットをBinary (バイナリ)，ASCII (アスキ)，EXTENDED (拡張データ形式)から選ぶ． EXTENDEDを選ぶと，KAPSELでは拡張データ形式としてHDF5が選択される． Print_field構造体はフィールドデータを出力する際の設定である． CropセレクタでYESを選ぶと，フィールドデータを間引いて出力する． Slab_x，Slab_y，Slab_z構造体は，

それぞれx，y，z方向における間引きの設定である． start，stride，countで，出力する最初の格子点番号，出力する格子点間隔，出力する格子点数をそれぞれ設定する． Velセレクタでは，YESを選ぶと，速度場を出力する． Phiセレクタでは，YESを選ぶと，SP関数ϕを出力する． Chargeセレクタでは，YESを選ぶと，電荷密度分布を出力する(constitutive_eqのtypeセレクタでElectrolyteを選択したときのみ有効)． Pressureセレクタでは，YESを選ぶと，圧力場を出力する(未実装)． Tauセレクタでは，YESを選ぶと，応力テンソルを出力する．また，UDFを出力する場合はUDFセレクタでONを選ぶ．

A.6　UDF出力データクラスの定義

define.udfでは，次に示すUDFデータ出力用のデータクラスが定義されている．これらは設定項目ではないので，詳細は割愛する．

```
\begin{def}
  class outParticle:{
    R:Vector3d [L],
    R_raw:Vector3d [L],
    v:Vector3d [L/tau],
    q:Quaternion,
    omega:Vector3d,
    f_hydro:Vector3d [mass*L*tau^-2],
    torque_hydro:Vector3d,
    f_r:Vector3d [mass*L*tau^-2],
    torque_r:Vector3d,
    f_slip:Vector3d [mass*L*tau^-2],
    torque_slip:Vector3d
  }
  E: float [epsilon] "total kinetic energy of the system"
  t: float "total time"
  Particles[]: outParticle
  RigidParticles[]: outParticle
  PSI[][][]: {
    psi: float
  }
\end{def}

\begin{def}
  class sParticle:{
    R:Vector3d [L],
    R_raw:Vector3d [L],
    v:Vector3d [L/tau],
    v_old:Vector3d [L/tau],
    f_hydro:Vector3d [mass*L*tau^-2],
    f_hydro_previous:Vector3d  [mass*L*tau^-2],
    f_hydro1:Vector3d  [mass*L*tau^-2],
    f_slip:Vector3d  [mass*L*tau^-2],
    f_slip_previous:Vector3d  [mass*L*tau^-2],
    fr:Vector3d  [mass*L*tau^-2],
    fr_previous:Vector3d  [mass*L*tau^-2],
    omega:Vector3d,
    omega_old:Vector3d,
    torque_hydro:Vector3d,
    torque_hydro_previous:Vector3d,
    torque_hydro1:Vector3d,
    torque_slip:Vector3d,
    torque_slip_previous:Vector3d,
    torque_r:Vector3d,
    torque_r_previous:Vector3d,
    q:Quaternion,
    q_old:Quaternion
  }
  class Matrix3d:{
    xx: float,
    xy: float,
    xz: float,
    yx: float,
```

```
    yy: float,
    yz: float,
    zx: float,
    zy: float,
    zz: float
  }
  class CTime:{
    ts:int
    time:float [tau]
  }
\end{def}
```

A.7　リスタートの設定

resume構造体は，中断した計算を再計算するためのリスタート用の設定である．以下に示すCalculationセレクタで計算条件を選択する．

```
Calculation: select {
  'NEW',
  'CONTINUE',
  'CONTINUE_FDM',
  'CONTINUE_FDM_PHASE_SEPARATION'
} "flg in order to specify resumed simulation or not"
```

NEWを選ぶと新規の計算ができる． CONTINUE，CONTINUE_FDM，CONTINUE_FDM_PHASE_SEPARATIONはいずれも前回計算終了したときの情報を読み込んで計算を再開するための設定であり，constitutive_eqの設定によって使い分ける必要がある． CONTINUEは，スペクトル法を用いるシミュレーション(constitutive_eqにおいて，Navier_Stokes，Shear_Navier_Stokes，Shear_Navier_Stokes_Lees_Edwards，Electrolyteを選択している場合)を再開する設定である． CONTINUE_FDMは，差分法による一成分流体のシミュレーション(Navier_Stokes_FDM，Shear_Navier_Stokes_Lees_Edwards_FDMを選択している場合)を再開する設定である． CONTINUE_FDM_PHASE_SEPARATIONは，差分法による二成分流体のシミュレーション(Navier_Stokes_Cahn_Hilliard_FDM，Shear_NS_LE_CH_FDMを選択している場合)を再開する設定である．各設定ごとにSaved_Data以下に再計算に必要な結果が保存される:

```
CONTINUE:{
  Saved_Data:{
    jikan: CTime
    Particles[] : sParticle
    GR_body[]   : Vector3d
    GR_masses[] : float
    GR_moments_body[]: Matrix3d
    Zeta[][][]:{
      zeta0: float
      zeta1: float
    }
    uk_dc: Vector3d
    Concentration[][][][]: {ck:float}
    oblique: {
      degree_oblique: float
    }
  }
}
CONTINUE_FDM: {
  Saved_Data: {
    jikan: CTime
    Particles[]: sParticle
    GR_body[]: Vector3d
    GR_masses[]: float
    GR_moments_body[]: Matrix3d
    U[][][]: {
```

```
      u0: float
      u1: float
      u2: float
    }
    U_OLD[][][]: {
      u_old_0: float
      u_old_1: float
      u_old_2: float
    }
    oblique: {
      degree_oblique: float
    }
  }
}
CONTINUE_FDM_PHASE_SEPARATION: {
  Saved_Data: {
    jikan: CTime
    Particles[]: sParticle
    GR_body[]: Vector3d
    GR_masses[]: float
    GR_moments_body[]: Matrix3d
    U[][][]: {
      u0: float
      u1: float
      u2: float
    }
    U_OLD[][][]: {
      u_old_0: float
      u_old_1: float
      u_old_2: float
    }
    PSI[][][]: {
      psi: float
    }
    PSI_OLD[][][]: {
      psi_old: float
    }
    STRESS_OLD[][][]: {
      stress_old_0: float
      stress_old_1: float
      stress_old_2: float
    }
    oblique: {
      degree_oblique: float
    }
  }
}
```

これらの項目は，手動で設定する項目ではないので詳細は割愛する．

A.8　GOURMETでの表示設定

Unit_Parameter構造体では，GOURMETでの表示に関する設定である．シミュレーション結果への影響はない．

```
Unit_Parameter:{
  Name: string            "Name"
  Comment:string          "Comment"
  Temperature:double  [K] "Temperature"
  Length:double [nm]      "Grid spacing"
  rho:double  [g/cm^3]    "Fluid density"
} "Parameters for unit conversion"
```

NameではUDFファイルの名前を設定できる．　CommentではUDFファイルのコメントを設定できる．TemperatureではGOURMETでシミュレーションパラメータを実次元表示するための基準となる単位温度を指定

する．同様にLengthでは基準となる単位長さを，rhoでは，基準となる単位密度を指定する．

付録B

粒子計算

粒子のコンフィグレーション(位置，配向，速度，角速度)は2ステップAdams-Bashforth法を用いて更新される．ただし，初期ステップではEuler法を用いる．注目の動的変数を$\boldsymbol{Y}_I$とし，$\boldsymbol{Y}_I^n = \boldsymbol{Y}_I(t_n)$とすると，

$$\boldsymbol{Y}^{n+1} = \boldsymbol{Y}^n + \frac{h}{2}\left(3\dot{\boldsymbol{Y}}^n - \dot{\boldsymbol{Y}}^{n-1}\right) \tag{B.1}$$

Euler方程式は，立体(主軸)基準座標系(ここでは慣性モーメント$\widetilde{\boldsymbol{I}}$が対角成分となる)において，

$$\begin{pmatrix} \dot{\widetilde{\omega}}^1 \\ \dot{\widetilde{\omega}}^2 \\ \dot{\widetilde{\omega}}^3 \end{pmatrix} = \begin{pmatrix} \widetilde{\tau}^1 + \widetilde{\omega}^2\widetilde{\omega}^3(\widetilde{I}^{22} - \widetilde{I}^{33}) \\ \widetilde{\tau}^2 + \widetilde{\omega}^3\widetilde{\omega}^1(\widetilde{I}^{33} - \widetilde{I}^{11}) \\ \widetilde{\tau}^3 + \widetilde{\omega}^1\widetilde{\omega}^2(\widetilde{I}^{11} - \widetilde{I}^{22}) \end{pmatrix} \tag{B.2}$$

を用いて解かれる[8]．さらに数値計算の精度を上げるために，配向は，回転行列$\boldsymbol{R}$の代わりに(毎ステップごとに標準化した)回転四元数qを用いて更新される．それにより配向に対する動的方程式は，

$$\dot{\mathsf{q}} = \frac{1}{2}\mathsf{w}_I \circ \mathsf{q} = \frac{1}{2}\mathsf{q} \circ \widetilde{\mathsf{w}} \tag{B.3}$$

で与えられる[62]．ここで，$\circ$は四元数の乗算を表し，$\widetilde{\mathsf{w}} = (0, \widetilde{\boldsymbol{\omega}})$ と $\omega = (0, \boldsymbol{\omega})$はそれぞれ実験系での角速度$\boldsymbol{\omega}$および立体基準座標系$\widetilde{\boldsymbol{\omega}} = R^t \cdot \boldsymbol{\omega}$に相当する四元数を表す．

付録C

スペクトル法による流体計算

シミュレーション方法のセクションで示した通り，KAPSELではFractional Step法を採用している．この方法ではまず最初に，$\boldsymbol{u}^*$を決定するために，$\boldsymbol{\phi f}_p$項のない，移流項および粘性応力項だけを考慮した変形Navier-Stokes方程式を時間発展させる． KAPSELでは，計算を単純化するために，直接$\boldsymbol{u}$について解かずに，$\boldsymbol{\omega} = \boldsymbol{\nabla} \times \boldsymbol{u}$について解く．固定されたデカルト座標系を考慮すると，周期境界条件下において，非圧縮性条件($\boldsymbol{\nabla} \cdot \boldsymbol{u} = 0$)を満たす渦度の方程式は，

$$\begin{aligned} \partial_t \boldsymbol{\omega} &= -\boldsymbol{\nabla} \times \boldsymbol{\nabla} \cdot (\boldsymbol{uu}) + \rho^{-1} \boldsymbol{\nabla} \times \boldsymbol{\nabla} \cdot \boldsymbol{\sigma}, \\ \partial_t \widehat{\boldsymbol{\omega}} &= \boldsymbol{k} \times [\boldsymbol{k} \cdot \mathcal{F}_{\boldsymbol{k}} (\boldsymbol{uu})] - \rho^{-1} \boldsymbol{k} \times [\boldsymbol{k} \cdot \widehat{\boldsymbol{\sigma}}] \end{aligned} \tag{C.1}$$

で与えられる．ここで，$\mathcal{F}_{\boldsymbol{k}}(f) = \widehat{f}(\boldsymbol{k})$はFourier変換を表し，$\boldsymbol{k}$は波数ベクトルである．流体の速度場は，渦度の定義と非圧縮性条件から，

$$\widehat{\boldsymbol{u}} = i \frac{\boldsymbol{k} \times \widehat{\boldsymbol{\omega}}}{|\boldsymbol{k}|^2} \tag{C.2}$$

と得られる．渦度$\boldsymbol{\omega}$についての方程式を解くことの利点は，非圧縮性を満たすための圧力Poisson方程式を解く必要がないことにある。非圧縮性は，Fourier空間において

$$\widehat{\boldsymbol{u}} \longrightarrow \left[\boldsymbol{I} - \frac{\boldsymbol{kk}}{|\boldsymbol{k}|^2} \right] \cdot \widehat{\boldsymbol{u}} \tag{C.3}$$

のような射影を行うことにより満たされる．

さらにKAPSELでは，プログラムコードのメモリ必要量を減らすために，回転の3成分が線型独立ではなくその内の2成分のみ考慮すればよいという利点を活用する．ここで，$\widehat{\boldsymbol{a}} = \mathcal{F}_{\boldsymbol{k}}(\boldsymbol{\nabla} \times \boldsymbol{A}) = i\boldsymbol{k} \times \widehat{\boldsymbol{A}}$を任意のベクトル場$\boldsymbol{A}$での回転に対するFourier変換とし，全ての波数ベクトル(ただし$\boldsymbol{k} \neq 0$)について

$$(\widehat{\boldsymbol{a}})^\dagger = (\widehat{a}_1, \widehat{a}_2, \widehat{a}_3)^\dagger : \begin{cases} (\widehat{a}_2, \widehat{a}_3) & k_1 \neq 0 \\ & \widehat{a}_1 = -(k_3 \widehat{a}_3 + k_2 \widehat{a}_2)/k_1 \\ (\widehat{a}_3, \widehat{a}_1) & k_1 = 0, k_2 \neq 0 \\ & \widehat{a}_2 = -k_3 \widehat{a}_3 / k_2 \\ (\widehat{a}_1, \widehat{a}_2) & k_1 = k_2 = 0, k_3 \neq 0 \\ & \widehat{a}_3 = 0 \end{cases} \tag{C.4}$$

のような可逆的な写像($\dagger : C^3 \to C^2$)を定義する．ただし，全ベクトル場の体積積分に対応する$\widehat{\boldsymbol{a}}(\boldsymbol{k} = 0)$は，$\widehat{\boldsymbol{a}}^\dagger$から求めることができず，独立に計算される．本写像を式(C.1)の2式に適用することで，$\widehat{\boldsymbol{\zeta}} = \widehat{\boldsymbol{\omega}}^\dagger$についての式が得られ，速度場を更新するために必要な計算量を1/3削減することができる．

積分される式の形式は，用いられる応力テンソル$\boldsymbol{\sigma}$に依存する． $\boldsymbol{\nabla} \cdot \boldsymbol{\sigma} = \eta \nabla^2 \boldsymbol{u}$となるNewton流体の場合，渦度$\boldsymbol{\omega}$の方程式は

$$\partial_t \widehat{\boldsymbol{\omega}} = -\nu k^2 \widehat{\boldsymbol{\omega}} + \boldsymbol{k} \times [\boldsymbol{k} \cdot \mathcal{F}_{\boldsymbol{k}}(\boldsymbol{uu})] \tag{C.5}$$

となる．これは，

$$\partial_t a = \mathcal{L}a - \mathcal{N}(t, a(t)) \tag{C.6}$$

の形で表されている．ここで，$\mathcal{L}$は時間に依存しない線形演算子，$\mathcal{N}$は非線形演算子である．本式は一般解が

$$a(t_n + h) = e^{\mathcal{L}h}a(t_n) - e^{\mathcal{L}h}\int_0^h \mathrm{d}\tau e^{-\mathcal{L}\tau}\mathcal{N}(t_n + \tau, a(t_n + \tau)) \tag{C.7}$$

で与えられる．ここで線形部分は厳密に解くことが可能である．非線形部分の積分は様々な方法で近似することが可能である[63, 64]．特に，一次近似$\mathcal{N}(t_n + \tau, a(t_n + \tau)) = \mathcal{N}(t_n, a(t_n))$を用いると

$$a(t_n + h) = e^{\mathcal{L}h}\left[a(t_n) - \mathcal{L}^{-1}\left(1 - e^{-h\mathcal{L}}\right)\mathcal{N}(t_n, a(t_n))\right] \tag{C.8}$$

$$= a(t_n) + \left(e^{-h\mathcal{L}} - 1\right)\left(a(t_n) + \mathcal{L}^{-1}\mathcal{N}(t_n, a(t_n))\right) \tag{C.9}$$

が得られ，$|\mathcal{L}| \to 0$の極限でEuler法となる．

付録D

有限差分法(FDM)による流体計算

第6章に示す二成分相分離流体に分散した粒子のシミュレーションでは，流体計算に差分法が適用される．本付録では，KAPSELで実装されている差分法による流体計算手法について説明する．

D.1　Navier–Stokes方程式の解法

KAPSELでは，式(6.1)に示されるNavier–Stokes (NS)方程式を，陽的及び陰的MAC(Marker And Cell)法[43, 46]で解く．計算格子における変数配置は，速度変数と圧力変数がそれぞれが格子点と格子中心に配置されるsemi-staggered Arakawa B 格子となっている[42]． MAC陽解法と陰解法の詳細を以下に示す．

D.1.1　MAC陽解法

まず，式(6.1)から，粒子が流体に及ぼす体積力項を除いた式において，圧力項を陰的に，それ以外の項を陽的に離散化する:

$$\frac{\tilde{\boldsymbol{u}}^{n+1}-\boldsymbol{u}^n}{\Delta t}+(\boldsymbol{u}^n\cdot\boldsymbol{\nabla})\boldsymbol{u}^n+\frac{1}{\rho}\boldsymbol{\nabla}p^{n+1}-\nu\nabla^2\boldsymbol{u}^n+\frac{\psi^n}{\rho}\boldsymbol{\nabla}\mu_\psi^n+\frac{\phi^n}{\rho}\boldsymbol{\nabla}\mu_\phi^n=0. \tag{D.1}$$

ここで，上付きの添字$^{n+1}$のついた物理量は未知の量であり，添字^{n}のついた物理量はnステップにおける既知の量である．また，未知流速についたチルダは，ここで求めた速度場が予測速度であり，後で修正が必要であることを明示するために記している．未知流速に関して，連続の式が満たされると仮定すると，以下の圧力に関するPoisson方程式が得られる:

$$\frac{\Delta t}{\rho}\nabla^2p^{n+1}=\boldsymbol{\nabla}\cdot\boldsymbol{u}^n-\Delta t\boldsymbol{\nabla}\cdot\left[(\boldsymbol{u}^n\cdot\boldsymbol{\nabla})\boldsymbol{u}^n-\nu\nabla^2\boldsymbol{u}^n+\frac{\psi^n}{\rho}\boldsymbol{\nabla}\mu_\psi^n+\frac{\phi^n}{\rho}\boldsymbol{\nabla}\mu_\phi^n\right]. \tag{D.2}$$

右辺第1項は本来0となることが期待されるが，サイクル誤差自己調整のため残している．このPoisson方程式を解くことで未知の圧力p^{n+1}が得られ，これを式(D.1)に代入して未知流速$\tilde{\boldsymbol{u}}^{n+1}$を求める．ここで得られた$\tilde{\boldsymbol{u}}^{n+1}$は，計算格子が流体・粒子領域のいずれの場所にあるかを区別せずに求められたものであり，これをそのまま次ステップの値とすることはできない．そこで，粒子領域の速度場を粒子運動と対応させるため，粒子領域に拘束力を印加することで速度場を修正する:

$$\boldsymbol{u}^{n+1}=\tilde{\boldsymbol{u}}^{n+1}+\phi\boldsymbol{f}_{\mathrm{p}}\Delta t. \tag{D.3}$$

ここで印加する拘束力は，ダイバージェンス・フリーな力であり，得られた速度場も連続の式を満たす．この$\boldsymbol{u}^{n+1}$を次ステップの速度場とする．

D.1.2　MAC陰解法

以下のように速度と圧力を分離したまま陰解法で解くことで，陽解法より安定に計算できる[46].

まず，NS方程式を次のように離散化する:

$$\frac{\tilde{\boldsymbol{u}}^{n+1}-\boldsymbol{u}^{n}}{\Delta t}+\boldsymbol{u}^{*}\cdot\boldsymbol{\nabla}\boldsymbol{u}^{n+\frac{1}{2}}+\frac{1}{\rho}\boldsymbol{\nabla}p^{n+1}-\boldsymbol{\nabla}\cdot\frac{\eta^{*}}{\rho}\left(\boldsymbol{\nabla}\boldsymbol{u}^{n+\frac{1}{2}}+\boldsymbol{\nabla}\boldsymbol{u}^{\mathrm{T}n+\frac{1}{2}}\right)+\frac{\psi^{*}}{\rho}\boldsymbol{\nabla}\mu_{\psi}^{*}+\frac{\phi^{n+1}}{\rho}\boldsymbol{\nabla}\mu_{\phi}^{*}=0. \tag{D.4}$$

ここで，上付きの添字$^{n+1/2}$のついた物理量は，Crank–Nicolson法による$n+1/2$ステップでの未知の量であり，添字*のついた物理量はAdams–Bashforth (AB)法により外挿で求めた既知の量である．例えば，流速については，それぞれ

$$\boldsymbol{u}^{n+\frac{1}{2}}=\frac{1}{2}(\boldsymbol{u}^{n}+\tilde{\boldsymbol{u}}^{n+1}), \tag{D.5}$$

$$\boldsymbol{u}^{*}=\frac{1}{2}(3\boldsymbol{u}^{n}-\boldsymbol{u}^{n-1}) \tag{D.6}$$

と求めることができる．また，ここでは粘度ηが場所に依存して変化する変数としている．式(D.4)において，未知流速$\boldsymbol{u}^{n+1}$について連続の式が成り立つとすると，次の圧力に関するPoisson方程式が得られる:

$$\frac{\Delta t}{\rho}\nabla^{2}p^{n+1}=\boldsymbol{\nabla}\cdot\boldsymbol{u}^{n}-\Delta t\boldsymbol{\nabla}\cdot\left[\boldsymbol{u}^{*}\cdot\boldsymbol{\nabla}\boldsymbol{u}^{*}-\boldsymbol{\nabla}\cdot\frac{\eta^{*}}{\rho}\left(\boldsymbol{\nabla}\boldsymbol{u}^{*}+\boldsymbol{\nabla}\boldsymbol{u}^{\mathrm{T}*}\right)+\frac{\psi^{*}}{\rho}\boldsymbol{\nabla}\mu_{\psi}^{*}+\frac{\phi^{n+1}}{\rho}\boldsymbol{\nabla}\mu_{\phi}^{*}\right]. \tag{D.7}$$

ただし，ここでは右辺の未知項を全てAB法による既知項で置き換えており，このPoisson方程式は独立に解くことができる．これを解くことで得られた圧力を式(D.4)に代入し，得られる線形連立方程式を解くことで未知流速を求めることができる． KAPSELでは，この線形連立方程式の解法として前処理なしBiCGSTAB法[47]を採用している．付録Eに記すように，並列反復解法ライブラリのLisを導入することで，他の前処理や反復解法も利用できる．

D.1.3　Lees–Edwards境界条件でのせん断流下でのNavier–Stokes方程式

これらのMAC陽解法及び陰解法は，次に示すテンソル解析による座標変換によって印加されるLees–Edwards境界条件でのせん断流れの計算[23]にも適用できる．本手法の詳細は第4章に記されている．ここでは，直交座標系の基底ベクトル$\boldsymbol{e}_x$方向に印加されるせん断速度$\dot{\gamma}(t)$の流れを考える．このときせん断勾配方向$\boldsymbol{e}_y$方向の座標yにおけるせん断速度は，$\boldsymbol{U}=\dot{\gamma}(t)y\boldsymbol{e}_x$と表せる．上記条件のせん断流れについて，時間発展に伴い変化する斜交座標(共変基底)への座標変換を考える．斜交座標系の基底ベクトルは，式(4.2)の上の文中に示されている．KAPSELのLees–Edwards境界条件でのせん断流下のシミュレーションでは，斜交座標系において，流速からせん断流れの寄与を差し引いた速度場$\hat{\boldsymbol{\xi}}=\boldsymbol{u}-\boldsymbol{U}$の反変成分$\hat{\xi}^{i}$に関するNS方程式

$$\partial_t\hat{\xi}^{i}+\hat{\xi}1^{j}\hat{\partial}_j\hat{\xi}^{i}=-\rho^{-1}G^{ij}\hat{\partial}_j\hat{p}+\nu G^{jk}\hat{\partial}_j\hat{\partial}_k\hat{\xi}^{i}-2\dot{\gamma}\hat{\xi}^{2}\delta_{i1}, \tag{D.8}$$

$$\hat{\partial}_j\hat{\xi}^{j}=0 \tag{D.9}$$

を解く．ここで，$\hat{p}$は斜方座標系で定義される圧力，式(D.8)(D.9)は，Einstein縮約記法で記述している．また，∂_tは時間微分，$\hat{\partial}_i$は斜交座標系(共変基底)での空間微分を表し，$\hat{\partial}_i=\partial/\partial\hat{x}^{i}$である． G^{ij}は反変成分表記の計量テンソルであり，せん断流れにおいては，

$$G^{ij}=\begin{pmatrix}1+(\dot{\gamma}t)^{2} & -\dot{\gamma}t & 0\\ -\dot{\gamma}t & 1 & 0\\ 0 & 0 & 1\end{pmatrix} \tag{D.10}$$

である[23]．詳細は省略するが，KAPSELでは，式(D.8)(D.9)について直交座標系における計算と同様の手順で時間方向の離散化を行い，斜交座標系における陽的及び陰的MAC法を実装している．

D.2　Cahn–Hilliard方程式の解法

次に，式(6.2)に表されるCahn–Hilliard (CH)方程式の解法について説明する． KAPSELでは，CH方程式についても陽解法及び半陰解法を実装している．

D.2.1　陽解法

CH方程式の陽解法はEuler法が実装されている:

$$\frac{\psi^{n+1}-\psi^n}{\Delta t}+\boldsymbol{\nabla}\cdot(\psi^n\boldsymbol{u}^n)=\kappa\nabla^2\left[f'(\psi^n)-\alpha\nabla^2\psi^n+w\xi|\boldsymbol{\nabla}\phi^{n+1}|^2+2d\phi^{n+1}(\psi^n-\bar{\psi})-2z\xi_\psi\nabla\phi^{n+1}\cdot\nabla\psi^n\right]. \tag{D.11}$$

ここで，ψの保存性を良くするために，CH方程式の対流項は保存形で記述している．ここで，ϕ^{n+1}は，粒子識別関数の次ステップの値であるが，弱連成解法によりψに先立って時間発展させているため，既知である．

D.2.2　陰解法

CH方程式の数値計算では最大で4階の微分演算が必要になるので，一般に計算不安定となりやすい．そこで，KAPSELでは，以下の半陰解法も実装している[49]:

$$\begin{aligned}\frac{3\psi^{n+1}-4\psi^n+\psi^{n-1}}{2\Delta t}+\boldsymbol{\nabla}\cdot(\psi^{n+1}\boldsymbol{u}^{n+1})=\kappa\nabla^2\Big[&2f'(\psi^n)-f'(\psi^{n-1})-\alpha\nabla^2\psi^{n+1}+w\xi|\boldsymbol{\nabla}\phi^{n+1}|^2+2d\phi^{n+1}(\psi^{n+1}-\bar{\psi})\\&-2z\xi_\psi\nabla\phi^{n+1}\cdot\nabla\psi^{n+1}\Big].\end{aligned} \tag{D.12}$$

この定式化では，時間微分項については後退差分を適用し，また，右辺の非線形のポテンシャル項についてはAB法による外挿値を用いて式を線形化することで陰的に解いている．この線形連立方程式の解法においても，KAPSELでは前処理なしBiCGSTAB法を採用している．また，付録Eに記すように，並列反復解法ライブラリのLisを導入することで，他の前処理や反復解法も利用できる．

D.2.3　Lees–Edwards境界条件でのせん断流下でのCahn–Hilliard方程式

斜交座標系でのCH方程式の表式は

$$\partial_t\hat{\psi}+\hat{\partial}_i(\hat{u}^i\hat{\psi})=\kappa G^{ij}\hat{\partial}_i\hat{\partial}_j\left\{f'(\hat{\psi})-\alpha G^{kl}\hat{\partial}_k\hat{\partial}_l\hat{\psi}+w\xi G^{kl}\hat{\partial}_k\hat{\phi}\hat{\partial}_l\hat{\phi}+2d\hat{\phi}(\hat{\psi}-\bar{\psi})-2z\xi_\psi G^{kl}[\hat{\partial}_k\hat{\phi}\hat{\partial}_l\hat{\psi}+\hat{\phi}\hat{\partial}_k\hat{\partial}_l\hat{\psi}]\right\} \tag{D.13}$$

である．ここで，$\hat{\phi}$，$\hat{\psi}$は，それぞれ斜交座標系で定義される粒子識別関数，流体界面識別関数である．斜交座標系においても前節と同様に陽解法及び半陰解法のソルバーが実装されている．

付録E

Lisライブラリとの連携

KAPSEL内の有限差分法（FDM）計算における線形連立方程式の解法として，BiCGSTAB法[47]がデフォルトで用いられている．BiCGSTAB法は安定かつ高速な解法として幅広く流体計算に利用されており，KAPSELを用いた計算でもほとんどのケースで最良の選択肢となりえるが，ごく稀に計算が破綻してしまう可能性も完全には否定できない．そのような場合により堅牢な手法(例えばGMRES法[65]など)を選択できるよう，KAPSELには汎用反復解法ライブラリLis[48]との連携機能も実装されている． KAPSELの実行ファイルをビルドする際に，MakefileでLISオプションをONにすると，自動的にNavier–Stokes(NS)方程式とCahn–Hilliard(CH)方程式の陰解法ソルバーでLisが利用され，様々な前処理と反復解法を組み合わせた計算が可能である（つまりLisを使用するためにはKAPSELの再コンパイルが必須である）.

使用する前処理と反復解法はプログラム内で設定する． fdm_matrix_solver.cxxのInit_lis関数に，lis_solver_set_optionの設定箇所が2箇所存在する．それらはNS方程式とCH方程式の陰解法ソルバーの設定である． Lisの設定については，該当箇所のコメントに例を示しているが，詳細は公式ユーザーガイド[66]を参照いただきたい.

付録F

KAPSELのコンパイル

KAPSELversion5はバイナリの実行ファイルで配布されるので，ユーザーがソースコードからコンパイルする必要はない．ここでは開発者向けの情報として，KAPSELのコンパイル手順について解説する．以下の説明ではルートユーザとしての作業を仮定している．管理者権限を持つユーザーアカウントで作業する場合は，適宜sudoコマンドを用いること．最新の情報はKAPSEL5-HP[*1]を参照していただきたい．

F.1　KAPSELの動作環境

KAPSELの動作環境について，Windows (+ Cygwin)，Linux，Macにおける注意点を以下に示す．

Linuxの場合

Linux上でKAPSELをコンパイルする場合，F.2節へと移動.

Windowsの場合

Windows上でKAPSELをコンパイルする場合，Cygwin[*2]の基本パッケージに加え， Select Package section で“Category” viewにセットし，以下のパッケージをインストールに含めることが必要である．

- all Packages in the `Devel` category
- all Packages related to `fftw3` in the `Libs` category
- all Packages related to `hdf5` in the `Libs` category
- `python3` Package in the `Python` category

その後，F.2節へと移動.

Macの場合

MacOS上でKAPSELをコンパイルする場合，Xcodeとcommand line toolsの環境ではgccコマンドがclangコマンドを呼び出す仕様になっている．計算機の性能をより引き出すためには，Homebrew等を用いて別途gcc-12をインストールすることが望ましい．

F.2　OCTAのインストール

KAPSELは, ソフトマテリアルに対する統合的なシミュレータとして開発されたOCTA内部にあるGourmetとよばれるユーザインターフェースと連携し入力パラメータの管理や出力データの可視化をおこなっている. そこで, KAPSELを使用するためにはOCTAがインストールされていることが前提となっている．

[*1] https://kapsel-dns.com/v5
[*2] http://cygwin.com/

インストールの手順として，まずOCTAホームページ http://octa.jp/ にアクセスし，適当なインストーラをダウンロードする．その後，インストーラを実行し，OCTAをインストールする[*3]．以降，OCTA8.# が /usr/local/OCTA8# (Linux/Mac) または C:\OCTA8.# (Windows) にインストールされているとして手順を説明する[*4]．

F.3　libplatformのビルド

libplatformは，OCTAで用いられる，UDFファイルに保存されたデータにアクセスするための I/Oライブラリである． libplatformをビルドする際，環境に応じてOpenJDK[*5]やPython3[*6]などのソフトウェアを追加でインストールする必要がある．

Windowsの場合

管理者権限でWindowsにログオンし，Cygwinのターミナルウィンドウを開く．ターミナル上で以下を実行．

```
$ ln -s /cygdrive/c/OCTA8.# /usr/local/OCTA8#
$ cd /usr/local/OCTA8#/GOURMET/src
$ make distclean
$ ./configure --with-python
$ make
$ make install
```

Linuxの場合

- gccを用いる場合

```
$ cd /usr/local/OCTA8#/GOURMET/src
$ make distclean
$ ./configure --with-python
$ make
$ make install
$ mv ../lib/linux64/libplatform.a ../lib/linux64/libplatform_gcc.a
```

- icc[*7]を用いる場合

```
$ cd /usr/local/OCTA8#/GOURMET/src
$ make distclean
$ ./configure CC=icc CXX=icpc --with-python
$ make
$ make install
$ mv ../lib/linux64/libplatform.a ../lib/linux64/libplatform_icc.a
```

Macの場合

- clang[*8]を用いる場合

```
$ cd /usr/local/OCTA8#/GOURMET/src
$ make distclean
$ ./configure --with-python
$ make
```

[*3] OCTA/GOURMETに関する質問は，OCTA–BBSのメンバーになることで可能である．
[*4] インストールしたOCTAのバージョンに応じて，”#” に適当な数字を入力する．
[*5] AdoptOpenJDK: https://adoptopenjdk.net / For arm64 (Apple silicon): https://www.azul.com/downloads/
[*6] Anaconda: https://www.anaconda.com/
[*7] Intel oneAPI Toolkits
[*8] macOSのデフォルトCコンパイラ

```
$ make install
$ mv ../lib/macosx/libplatform.a ../lib/macosx/libplatform_clang.a
```

- gcc[*9]を用いる場合

```
$ cd /usr/local/OCTA8#/GOURMET/src
$ make distclean
$ ./configure CC=gcc-12 CXX=g++-12 --with-python
$ make
$ make install
$ mv ../lib/macosx/libplatform.a ../lib/macosx/libplatform_gcc.a
```

libplatformのビルド後，何らかの理由でGROUMETが正しく起動しない可能性がある．その場合は，以下のどちらかのコマンドを実行する．

```
$ cd /usr/local/OCTA8#/GOURMET
$ ./Make-All.sh
```

```
$ cd /usr/local/OCTA8#/GOURMET/src
$ ./build-gourmet
```

libplatformに関するより詳しい情報は，OCTAのマニュアルを参照していただきたい．

F.4　FFTWのインストール

Windowsの場合

ソースコードをダウンロードし，手動でインストールする必要がある．

```
$ ./configure --prefix=/opt/fftw/latest.gcc CFLAGS="-O3" FFLAGS="-O3"
↪ --enable-openmp --enable-threads --enable-shared --disable-fortran
$ make
$ make install
```

Linuxの場合

ソースコードをダウンロードし，手動でインストールする必要がある．

- gccを用いる場合

```
$ ./configure --prefix=/opt/fftw/latest.gcc CFLAGS="-O3" FFLAGS="-O3"
↪ --enable-openmp --enable-threads --enable-shared --disable-fortran
$ make
$ make install
```

- iccを用いる場合

```
$ ./configure --prefix=/opt/fftw/latest.gcc CC=icc CXX=icpc CFLAGS="-O3"
↪ FFLAGS="-O3" --enable-openmp --enable-threads --enable-shared
↪ --disable-fortran
$ make
$ make install
```

Macの場合

[*9] brew install gcc-12 等でインストール可能

- clangを用いる場合

```
brew install fftw
```

- gccを用いる場合

```
    $ ./configure --prefix=/opt/fftw/latest.gcc CC=gcc-12 CXX=g++-12 CFLAGS="-O3"
↪   FFLAGS="-O3" --enable-openmp --enable-threads --enable-shared
↪   --disable-fortran
    $ make
    $ make install
```

F.5　HDF5のインストール

Windowsの場合

ソースコードをダウンロードし，手動でインストールする必要がある．

```
    $ ./configure --prefix=/opt/hdf5/latest.gcc CFLAGS="-O3" FFLAGS="-O3"
↪   --enable-fortran --enable-cxx
    $ make
    $ make install
```

Linuxの場合

ソースコードをダウンロードし，手動でインストールする必要がある．

- gccを用いる場合

```
    $ ./configure --prefix=/opt/hdf5/latest.gcc CFLAGS="-O3" FFLAGS="-O3"
↪   --enable-fortran --enable-cxx
    $ make
    $ make install
```

- iccを用いる場合

```
    $ ./configure --prefix=/opt/hdf5/latest.icc CC=icc CXX=icpc CFLAGS="-O3"
↪   FFLAGS="-O3" --enable-fortran --enable-cxx
    $ make
    $ make install
```

Macの場合

- clangを用いる場合

```
    $ brew install hdf5
```

- gccを用いる場合

```
    $ ./configure --prefix=/opt/hdf5/latest.gcc CC=gcc-12 CXX=g++-12
↪   --enable-fortran --enable-cxx
    $ make
    $ make install
```

F.6　LISのインストール (任意)

Windowsの場合

ソースコードをダウンロードし，手動でインストールする必要がある．

```
$ $./configure --prefix=/opt/lis/latest.gcc
$ make
$ make install
```

Linuxの場合

ソースコードをダウンロードし，手動でインストールする必要がある．

- gccを用いる場合

```
$ $./configure --prefix=/opt/lis/latest.gcc
$ make
$ make install
```

- iccを用いる場合

```
$ ./configure --prefix=/opt/lis/latest.icc CC=icc CXX=icpc
$ make
$ make install
```

Macの場合

ソースコードをダウンロードし，手動でインストールする必要がある．

- clangを用いる場合

```
$ ./configure --prefix=/opt/lis/latest.clang CC=clang CXX=clang++
$ make
$ make install
```

- gccを用いる場合

```
$ ./configure --prefix=/opt/lis/latest.gcc CC=gcc-12 CXX=g++-12
$ make
$ make install
```

F.7　KAPSEL実行ファイルのビルド

最新バージョンのKAPSELソースコード `kapsel#.#.zip`[*10] をダウンロードし，圧縮ファイルを解凍する．

```
$ unzip kapsel#.#.zip
$ cd kapsel#.#/src
```

F.3節でビルドしたライブラリ `libplatform.a` と正しくリンクされるために，添付されている `Makefile` 中の `GOURMET_HOME_PATH` を修正する必要がある．また環境に応じて，`-I` (インクルードファイルのパス) および `-L` (ライブラリファイルのパス) も変更する必要がある．正しくパスが設定されたら，まず作業ファイルを削除する．

[*10] バージョンに応じて ”#” に適当な数字を入力する

```
$ make clean
```

その後，以下に示す make コマンドの中から1つを実行し，利用する計算環境に適切なKAPSELバイナリ実行ファイルをビルドする．

Windowsの場合

Cygwinを用いる．

```
$ make ENV=CYGWIN
$ make ENV=CYGWIN FFT=FFTW
$ make ENV=CYGWIN_OMP FFT=FFTW
```

Linuxの場合

- gccを用いる場合

```
$ make ENV=GCC
$ make ENV=GCC FFT=FFTW
$ make ENV=GCC_OMP FFT=FFTW
```

- iccを用いる場合

```
$ make ENV=ICC
$ make ENV=ICC FFT=IMKL
$ make ENV=ICC_OMP FFT=IMKL
```

Macの場合

- clangを用いる場合

```
$ make ENV=CLANG
$ make ENV=CLANG FFT=FFTW
$ make ENV=CLANG_OMP FFT=FFTW
```

- gccを用いる場合

```
$ make ENV=GCC_MAC
$ make ENV=GCC_MAC FFT=FFTW
$ make ENV=GCC_MAC_OMP FFT=FFTW
```

全ての場合

作成した実行ファイルkapselをbinにコピーしてKAPSELのインストールディレクトリにシンボリックリンクする．

```
$ cd ..
$ cp ./src/kapsel ./bin/kapsel_self_made
$ ln -s ./bin/kapsel_self_made ./kapsel
```

環境変数を適切に設定してKAPSELを実行する．

```
$ export DYLD_LIBRARY_PATH = "/opt/fftw/latest.gcc/lib: /opt/hdf5/latest.gcc/lib:
↪ /opt/lis/latest.gcc/lib: DYLD_LIBRARY_PATH"
$ ./kapsel
Usage:
> ./kapsel -I[input UDF] -O[output UDF] -D[define UDF] -R[restart UDF]
```

謝辞

この成果は，国立研究開発法人科学技術振興機構(JST)さきがけ研究，JST-CREST研究，科学研究費助成事業，国立研究開発法人新エネルギー・産業技術総合開発機構(NEDO)からの援助，及び産業技術総合研究所(AIST)，京都大学，九州大学の協力により得られたものです.

参考文献

[1] R. Yamamoto, J. J. Molina, and Y. Nakayama, "Smoothed profile method for direct numerical simulations of hydrodynamically interacting particles", Soft Matter **17**, 4226 (2021).

[2] J. N. Israelachvili, 分子間力と表面力第*2*版 (朝倉出版, 1996).

[3] J. R. Blake, "A spherical envelope approach to ciliary propulsion", Journal of Fluid Mechanics **46**, 199 (1971).

[4] J. J. Molina, Y. Nakayama, and R. Yamamoto, "Hydrodynamic interactions of self-propelled swimmers", Soft Matter **9**, 4923 (2013).

[5] Y. Nakayama and R. Yamamoto, "Simulation method to resolve hydrodynamic interactions in colloidal dispersions", Physical Review E **71**, 036707 (2005).

[6] Y. Nakayama, K. Kim, and R. Yamamoto, "Simulating (electro)hydrodynamic effects in colloidal dispersions: Smoothed profile method", The European Physical Journal E **26**, 361 (2008).

[7] J. J. Molina and R. Yamamoto, "Direct numerical simulations of rigid body dispersions. I. Mobility/friction tensors of assemblies of spheres", The Journal of Chemical Physics **139**, 234105 (2013).

[8] J. V. José and E. J. Saletan, *Classical Dynamics: A Contemporary Approach* (Cambridge University Press, Aug. 1998).

[9] H. Tanaka and T. Araki, "Simulation Method of Colloidal Suspensions with Hydrodynamic Interactions: Fluid Particle Dynamics", Physical Review Letters **85**, 1338 (2000).

[10] R. Yamamoto, "Simulating Particle Dispersions in Nematic Liquid-Crystal Solvents", Physical Review Letters **87**, 075502 (2001).

[11] X. Luo, M. R. Maxey, and G. E. Karniadakis, "Smoothed profile method for particulate flows: Error analysis and simulations", Journal of Computational Physics **228**, 1750 (2009).

[12] T. Iwashita, Y. Nakayama, and R. Yamamoto, "A numerical model for brownian particles fluctuating in incompressible fluids", Journal of the Physical Society of Japan **77**, 10.1143/JPSJ.77.074007 (2008).

[13] T. Iwashita, Y. Nakayama, and R. Yamamoto, "Velocity autocorrelation function of fluctuating particles in incompressible fluids: toward direct numerical simulation of particle dispersions", Progress of Theoretical Physics Supplement **178**, 10.1143/PTPS.178.86 (2008).

[14] T. Iwashita and R. Yamamoto, "Short-time motion of brownian particles in a shear flow", Physical review. E, Statistical, nonlinear, and soft matter physics **79**, 031401 (2009).

[15] D. A. S. W. B. Russel and W. R. Schowalter, *Colloidal dispersions* (Cambridge University Press, Cambridge, England, 1989).

[16] Y. Otsubo, "The present status and prospect of suspension rheology", Nihon Reoroji Gakkaishi **31**, 15 (2003).

[17] T. Matsumoto, "Rheology of colloidal disperse systems", Nihon Reoroji Gakkaishi **32**, 3 (2004).

[18] I. M. Krieger, "Rheology of monodisperse latices", Advances in Colloid and Interface Science **3**, 111 (1972).

[19] C.-H. Hsueh and P. Becher, "Effective viscosity of suspensions of spheres", Journal of the American Ceramic Society **88**, 1046 (2005).

[20] A. Onuki, "Phase transitions of fluids in shear flow", Journal of Physics: Condensed Matter **9**, 6119 (1997).

[21] S. Toh, K. Ohkitani, and M. Yamada, “Enstrophy and momentum fluxes in two-dimensional shear flow turbulence”, Physica D: Nonlinear Phenomena **51**, 569 (1991).

[22] H. Kobayashi and R. Yamamoto, “Implementation of Lees–Edwards periodic boundary conditions for direct numerical simulations of particle dispersions under shear flow”, The Journal of Chemical Physics **134**, 064110 (2011).

[23] H. Kobayashi and R. Yamamoto, “Implementation of lees–edwards periodic boundary conditions for direct numerical simulations of particle dispersions under shear flow”, The Journal of Chemical Physics **134**, 064110 (2011).

[24] K. Kim and R. Yamamoto, “Efficient simulations of charged colloidal dispersions: a density functional approach”, Macromolecular Theory and Simulations **14**, 278 (2005).

[25] K. Kim, Y. Nakayama, and R. Yamamoto, “Direct numerical simulations of electrophoresis of charged colloids”, Phys. Rev. Lett. **96**, 208302 (2006).

[26] H. Tanaka and T. Araki, “Simulation method of colloidal suspensions with hydrodynamic interactions: fluid particle dynamics”, Phys. Rev. Lett. **85**, 1338 (2000).

[27] T. Kajishima, S. Takiguchi, H. Hamasaki, and Y. Miyake, “Turbulence structure of particle-laden flow in a vertical plane channel due to vortex shedding”, JSME International Journal Series B Fluids and Thermal Engineering **44**, 526 (2001).

[28] J.-L. Barrat and J.-P. Hansen, *Basic concepts for simple and complex liquids* (Mar. 2003).

[29] 北原文雄・古澤邦夫・尾崎正孝・大島広行, ゼータ電位微粒子界面の物理化学 (サイエンティスト社, 1995).

[30] H. Ohshima, T. W. Healy, and L. R. White, “Accurate analytic expressions for the surface charge density/surface potential relationship and double-layer potential distribution for a spherical colloidal particle”, Journal of Colloid and Interface Science **90**, 17 (1982).

[31] R. O’Brien and L. White, “Electrophoretic mobility of a spherical colloidal particle”, Journal of The Chemical Society, Faraday Transactions 2 **74**, 1607 (1978).

[32] H. Ohshima, T. W. Healy, and L. R. White, “Approximate analytic expressions for the electrophoretic mobility of spherical colloidal particles and the conductivity of their dilute suspensions”, J. Chem. Soc., Faraday Trans. 2 **79**, 1613 (1983).

[33] 土井正男・滝本淳一, 物理仮想実験室 (名古屋大学出版会, 2004).

[34] W. Ramsden, “Separation of solids in the surface-layers of solutions and ‘suspensions’ (observations on surface-membranes, bubbles, emulsions, and mechanical coagulation).—preliminary account”, Proceedings of the royal Society of London **72**, 156 (1904).

[35] S. U. Pickering, “CXCVI.-Emulsions”, Journal of the Chemical Society, Transactions **91**, 2001 (1907).

[36] F. Fenouillot, P. Cassagnau, and J.-C. Majesté, “Uneven distribution of nanoparticles in immiscible fluids: morphology development in polymer blends”, Polymer **50**, 1333 (2009).

[37] Y. Yang, Z. Fang, X. Chen, W. Zhang, Y. Xie, Y. Chen, Z. Liu, and W. Yuan, “An overview of pickering emulsions: solid-particle materials, classification, morphology, and applications”, Frontiers in pharmacology **8**, 287 (2017).

[38] M. E. Cates and P. S. Clegg, “Bijels: a new class of soft materials”, Soft Matter **4**, 2132 (2008).

[39] M. N. Lee and A. Mohraz, “Bicontinuous macroporous materials from bijel templates”, Advanced Materials **22**, 4836 (2010).

[40] P. C. Hohenberg and B. I. Halperin, “Theory of dynamic critical phenomena”, Reviews of Modern Physics **49**, 435 (1977).

[41] 土井正男・小貫明, 高分子物理・相転移ダイナミクス (岩波書店, 1992).

[42] A. Arakawa and V. R. Lamb, “Computational design of the basic dynamical processes of the ucla general circulation model”, General circulation models of the atmosphere **17**, 173 (1977).

[43] F. H. Harlow and J. E. Welch, “Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface”, The physics of fluids **8**, 2182 (1965).

[44] J. H. Ferziger and M. Peric, *Computational methods for fluid dynamics* (Springer Science & Business Media, 2012).

[45] D. Jacqmin, “Calculation of two-phase navier–stokes flows using phase-field modeling”, Journal of Computational Physics **155**, 96 (1999).

[46] A. Maruoka, J. Matsumoto, and M. Kawahara, “A fractional step finite element method for incompressible navier–stokes equations using quadrilateral scaled bubble function”, Journal of Structural Engineering A **44**, 383 (1998).

[47] H. A. Van der Vorst, “Bi-cgstab: a fast and smoothly converging variant of bi-cg for the solution of nonsymmetric linear systems”, SIAM Journal on scientific and Statistical Computing **13**, 631 (1992).

[48] *Lis: Library of Iterative Solvers for Linear Systems*, `https://www.ssisc.org/lis/index.ja.html`, Accessed: 2019-01-28.

[49] Y. He, Y. Liu, and T. Tang, “On large time-stepping methods for the Cahn–Hilliard equation”, Applied Numerical Mathematics **57**, 616 (2007).

[50] *ParaView Documentation*, `https://docs.paraview.org/en/latest/`, Accessed: 2021-09-30.

[51] C. Domb, *Phase transitions and critical phenomena* (Elsevier, 2000).

[52] M. C. Marchetti, J. F. Joanny, S. Ramaswamy, T. B. Liverpool, J. Prost, M. Rao, and R. A. Simha, “Hydrodynamics of soft active matter”, Reviews of Modern Physics **85**, 1143 (2013).

[53] M. J. Lighthill, “Hydromechanics of Aquatic Animal Propulsion”, Annual Review of Fluid Mechanics **1**, 413 (1969).

[54] J. R. Blake, “A spherical envelope approach to ciliary propulsion”, Journal of Fluid Mechanics **46**, 199 (1971).

[55] O. S. Pak and E. Lauga, “Generalized squirming motion of a sphere”, Journal of Engineering Mathematics **88**, 1 (2014).

[56] N. Oyama, J. J. Molina, and R. Yamamoto, “Simulations of Model Microswimmers with Fully Resolved Hydrodynamics”, Journal of the Physical Society of Japan **86**, 101008 (2017).

[57] A. Bricard, J.-B. Caussin, N. Desreumaux, O. Dauchot, and D. Bartolo, “Emergence of macroscopic directed motion in populations of motile colloids”, Nature **503**, 95 (2013).

[58] G. Quincke, “Ueber rotationen im constanten electrischen felde”, Annalen der Physik **295**, 417 (1896).

[59] A. Mauleon-Amieva, M. Mosayebi, J. E. Hallett, F. Turci, T. B. Liverpool, J. S. Van Duijneveldt, and C. P. Royall, “Competing active and passive interactions drive amoebalike crystallites and ordered bands in active colloids”, Physical Review E **102**, 032609 (2020).

[60] A. J. Goldman, R. G. Cox, and H. Brenner, “Slow viscous motion of a sphere parallel to a plane wall—i motion through a quiescent fluid”, Chemical engineering science **22**, 637 (1967).

[61] B. Cichocki and R. Jones, “Image representation of a spherical particle near a hard wall”, Physica A: Statistical Mechanics and its Applications **258**, 273 (1998).

[62] M. P. Allen and D. J. Tildesley, *Computer Simulation of Liquids* (Oxford Science Publications, 1987).

[63] S. M. Cox and P. C. Matthews, “Exponential time differencing for stiff systems”, Journal of Computational Physics **176**, 430 (2002).

[64] M. Hochbruck and A. Ostermann, “Exponential integrators”, Acta Numerica **19**, 209 (2010).

[65] Y. Saad and M. H. Schultz, “Gmres: a generalized minimal residual algorithm for solving nonsymmetric linear systems”, SIAM Journal on scientific and statistical computing **7**, 856 (1986).

[66] *Lis* ユーザガイド, `https://www.ssisc.org/lis/lis-ug-ja.pdf`, Accessed: 2019-01-28.