



MateriApps Installer Documentation

リリース **1.0**

MateriApps Installer Development Team

2021 年 03 月 13 日

目次

第 1 章	MateriApps Installer とは？	1
1.1	MateriApps Installer の概要	1
1.2	開発の背景	1
1.3	目標	2
1.4	整備されているツールとアプリ	2
1.5	設計ポリシー	3
1.6	開発者	4
1.7	バージョン履歴	4
1.8	ライセンス	4
第 2 章	利用方法	5
2.1	ダウンロード	5
2.2	ディレクトリ構造	5
2.3	セットアップ	7
2.4	インストール	8
2.5	ツール・アプリの利用方法	10
第 3 章	チュートリアル	11
3.1	macOS 10.15 (Catalina) へのインストール	11
3.2	Linux へのインストール (gcc 利用)	13
第 4 章	ファイル形式	17
4.1	インストール関連ファイル	17
4.2	管理用ファイル (scripts/)	22
第 5 章	アプリ一覧	23
5.1	ALPS	23
5.2	ALPSCore	24
5.3	DSQSS	24
5.4	QUANTUM ESPRESSO	25
5.5	H Φ	26
5.6	K ω	26
5.7	LAMMPS	27

5.8	mVMC	28
5.9	OpenMX	28
5.10	RESPACK	29
5.11	TeNeS	30
第 6 章	お問い合わせ	31
第 7 章	謝辞	33

第 1 章

MateriApps Installer とは？

1.1 MateriApps Installer の概要

MateriApps Installer は様々な計算機環境に対応した計算物質科学アプリケーションのインストール補助を行うシェルスクリプト集である。MateriApps Installer には、オープンソースの計算物質科学アプリケーションやツール類を、macOSをはじめ、Linux PC やクラスワークステーション、さらには国内の主要なスパコンシステムにインストールするためのシェルスクリプトが含まれる。MateriApps Installer は、東京大学物性研究所の全国共同利用スパコンにおいて、主要アプリケーションのプレインストールにも利用されている（インストールされているソフトウェア一覧はアプリ一覧に記載されている）。

1.2 開発の背景

今日では物質科学の理論研究を推し進めていく上でコンピュータを用いた数値計算は必要不可欠な存在となっている。計算物質科学を進めていくためには、物質科学の方程式を効率的に解くためのアルゴリズムの開発が重要な役割を果たしており、効率的なアルゴリズム・優れた並列性能・最先端の機能などを有する優れたアプリケーションが数多く生まれている。我々は開発されたソフトウェアについての情報を実験家・企業研究者などにも広く伝えるため、2013年より物質科学シミュレーションのポータルサイト [MateriApps](#) を公開し、国内外で開発された物質科学アプリケーションの情報を発信してきた。

利用者が物質科学の公開アプリケーションを使い始めようとする際に障害となるのは、ソフトウェアのインストール作業である。そこで手持ちのノート PC などを用いて、気軽に計算物質科学アプリケーションを試せすことができる環境 [MateriApps LIVE!](#) の開発・公開も行っている。[MateriApps LIVE!](#)は、[MateriApps](#) アプリケーション、OS (Debian GNU/Linux)、エディタ、可視化ツールなど、チュートリアルを始めるのに必要な環境が [VirtualBox](#) の仮想ハードディスクイメージ (OVA) としてまとめられている。[MateriApps LIVE!](#)を用いることで、授業やソフトウェア講習会などで参加者の計算機環境を簡単に構築することが可能である。

しかしながら、実用向けの計算を行うには [MateriApps LIVE!](#)の計算環境では非力である。物性研スーパーコンピュータなどの国内の主要なスーパーコンピュータから各研究室のクラスター計算機、個人の PC など、幅広い計算環境において計算物質科学アプリケーションを利用者が手軽にインストールできるよう、[MateriApps Installer](#) の

開発が開始された。

1.3 目標

MateriApps Installer の目標は下記の通りである。

- 計算物質科学分野に共通するアプリ (cf. MateriApps) を国内 (外) の主要なスパコン全てにインストールする。
- 同様に CentOS (RedHat), Debian (Ubuntu) の標準的な環境、macOS 環境にもインストールできるスクリプトを整備する。

1.4 整備されているツールとアプリ

2021/3/13 現在、以下のツール・アプリが整備されています (config で default.sh 以外に用意されているものと一緒に記載しています)。

1. ツール

name	intel	macos	gcc
boost	o	o	x
cmake	x	x	x
eigen3	x	x	x
fftw	o	o	x
gcc10	x	x	x
gcc8	x	x	x
git	x	x	x
gsl	o	x	x
hdf5	x	x	x
julia	o	x	x
lapack	x	x	x
libffi	x	x	x
openmpi	o	x	o
openssl	x	x	x
python3	o	o	x
scalapack	x	x	x
tcltk	x	x	x
zlib	x	x	x

2. アプリ

name	intel	macos	gcc
ALPS	o x	o	
ALPSCore	o	o	o
DSQSS	o	x	o
QUANTUM ESPRESSO	o	x	x
HPhi	o	x	o
KOmega	o	x	x
LAMMPS	o	x	o
mVMC	o	x	o
OpenMX	o	x	x
RESPACK	o	x	x
TeNeS	o	x	x

3. 今後追加予定のアプリ

abICS, DCore, TRIQS

1.5 設計ポリシー

MateriApps Installer の設計ポリシーは下記の通りである。

1. 特殊なツールに依存しないようにする (shell, make, tar などがあれば OK)。
2. スパコンはそれぞれ特殊なので、あまりユニバーサルなインストーラを作ることは目指さず、例外には個別のスクリプトを作成することで対処する。
3. ツールについては、Linux (RPM Package, Debian Package) や macOS (Homebrew, Fink, MacPorts) ですでにパッケージとして用意されている場合には、そちらを使う。
4. アプリのバージョンは、**MateriApps LIVE!** と揃える。必要なパッチも MateriApps LIVE!用の Debian Package と共通とする。
5. ツール・アプリ毎に別のフォルダを使う。その下にバージョン別のフォルダを準備する。

- cmake の場合の例

```
$PREFIX_TOOL/cmake/cmake-3.2.1-1
```

6. それぞれのツール・アプリ、バージョン用の環境変数設定スクリプトを準備する。

- cmake の場合の例

```
$PREFIX_TOOL/cmake/cmakevars-3.2.1-1.sh
```

7. ツールの環境変数設定スクリプトについては PREFIX_TOOL/env.d にリンクを張り、 PREFIX_TOOL/env.sh で一括設定されるように古いバージョンを残せるようにする。
8. インストールとリンクの張替えを分ける (リンクの張替えを行うまでは、他に影響を与えない)。
9. 一時的に違う場所にインストールしてテストできるようにする。

1.6 開発者

MateriApps Installer は以下のメンバーで開発している。

- ver. 0.1 -
 - 藤堂 眞治 (東京大学 大学院理学系研究科/物性研究所)
 - 本山 裕一 (東京大学 物性研究所)
 - 吉見 一慶 (東京大学 物性研究所)
 - 加藤 岳生 (東京大学 物性研究所)

1.7 バージョン履歴

- 2021/03/14 ver. 1.0 リリース.
- 2020/12/04 ver. 0.1 リリース.

1.8 ライセンス

MateriApps Installer の著作権は東京大学が所持しており、GNU General Public License version 3 (GPLv3) に準じて配布しています。ただし、インストールされるソフトウェアへのパッチファイルについては、それぞれのソフトウェアのライセンスに準ずるものとします。

(c) 2013-2021 The University of Tokyo. All rights reserved.

MateriApps Installer の開発については、2020 年度東京大学物性研究所ソフトウェア高度化プロジェクトから多大なる支援をいただきました。

第 2 章

利用方法

2.1 ダウンロード

- MateriApps Installer は以下の手順でダウンロードできる。

- リリース版のダウンロード

MateriApps Installer のリリースページに行き、zip ファイルをダウンロードした後に展開する。リリースページからダウンロードする zip ファイルには本マニュアルの pdf 版も同封されている。

- git を利用したダウンロード

以下のコマンドを打つことで、MateriApps Installer のダウンロードが可能である。:

```
git clone https://github.com/wistaria/MateriAppsInstaller
```

2.2 ディレクトリ構造

- 展開後に得られるディレクトリの構造は以下の通りである。

```
| - README.md
| - apps
| - check
| - check_prefix.sh
| - docs
| - list_maversion.sh
| - macos
| - scripts
| - setup
| - tools
```

- apps, tools 内にあるディレクトリは以下のような構成になっている。

```
-- software_name
  |- README.md
  |- download.sh
  |- link.sh
  |- setup.sh
  |- version.sh
  |- install.sh
  |- patch
  |- config
```

各ファイルおよびディレクトリの説明を以下に記載する。なお、* がついているファイルは、ディレクトリ内に必ず存在するファイルを示す。

- README.md (*)
 - * ソフトウェアの簡単な紹介や公式サイトの URL などが記載されている
 - download.sh (*)
 - * ソースコードアーカイブをダウンロードする
 - link.sh (*)
 - * インストールしたディレクトリや設定ファイルへのシンボリックリンクを作成する
 - setup.sh (*)
 - * 用意したソースコードアーカイブを展開し、(存在するなら)パッチを適用する
 - version.sh (*)
 - * ダウンロードするバージョンを指定する
 - install.sh (*)
 - * プログラムのビルドならびにインストールを行う
 - patch
 - * パッチを格納するディレクトリ
 - config
 - * Intel コンパイラを用いる場合など、デフォルト設定以外のインストールを行うための追加設定集
- また、上記以外にも以下のファイル・ディレクトリが用意されている。
- check_prefix.sh
 - * インストール先のトップディレクトリなど、各スクリプト共通で用いられる変数を表示するスクリプト

- list_maversion.sh
 - * 各ディレクトリ中にある version.sh の情報をまとめるスクリプト
- check ディレクトリ
 - * 各種ホストで複数のインストールスクリプトを順番に走らせるためのスクリプト
- docs ディレクトリ
 - * マニュアル及びそのソースコード一式が格納されているディレクトリ
- macos ディレクトリ
 - * Macports を用いて必要なツールをインストールするためのスクリプトが格納されているディレクトリ
- scripts ディレクトリ
 - * 管理用スクリプト一式が格納されているディレクトリ
- setup ディレクトリ
 - * ソフトウェアのインストールを行うための前準備を行うためのスクリプトが格納されているディレクトリ (詳細はこの後のセットアップを参照)

2.3 セットアップ

- ソフトウェアのインストールを行う前に setup/setup.sh を実行する必要がある
 - sh setup/setup.sh
 - このスクリプトは初期設定として、インストールディレクトリや作業用ディレクトリなどの作成を行う
- MateriApps Installer によって導入されるアプリケーションのインストール場所の設定
 - インストール場所は、次のように \$HOME/.maininstaller ファイルで設定可能 (使用する場合は自分で本ファイルを作成する必要あり)

```
# シェルスクリプトとして処理されるため、 = の前後に空白は置いてはいけない
MA_ROOT=$HOME/materiapps # ソフトウェアのインストール先
BUILD_DIR=$HOME/build    # インストール作業場所
SOURCE_DIR=$HOME/source  # ファイルダウンロード場所
```

オプション	デフォルト	説明
MA_ROOT	\$HOME/ materiapps	ソフトウェアのインストール先
BUILD_DIR	\$HOME/build	インストール作業場所
SOURCE_DIR	\$HOME/source	ソースコードアーカイブファイルのダウンロード場所

- このファイルがない場合は \$HOME/materiapps の下にソフトウェアがインストールされる
- (注) 実際のインストール場所は、setup.sh を実行した時の情報ではなく、以降で説明するインストール作業を行った時点での .maininstaller ファイルの内容が用いられる。

2.4 インストール

- 各ソフトウェアのディレクトリに移動し、install.sh を実行する。

```
sh install.sh
```

- このスクリプトを実行すると、ソースコードのダウンロード (download.sh)・展開 (setup.sh) を行った後に、ビルドおよびインストールが自動に行われる
- ソフトウェアによってはコンパイラやライブラリに対する設定が定義済みの場合があり、config ディレクトリ以下にサブディレクトリとして収められている

– sh install.sh intel のように、引数で与えることで使用可能

- * 存在しない設定ディレクトリを指定した場合、使用可能な設定の一覧を表示する

```
$ sh install.sh help
Error: unknown mode: help
Available list:
default
intel
```

– default

- * 引数を省略した場合に使用される、基本的な設定

– intel

- * Intel コンパイラ、Intel MKL、Intel MPI を使用するための設定

- シェル変数を用いてコンパイラなどの指定が可能

– (例) デフォルト設定を使いつつ C コンパイラとして Intel コンパイラを使いたい場合

```
CC=`which icc` sh install.sh
```

- 特に、MA_EXTRA_FLAGS を設定することでコンパイラオプションを追加可能

```
MA_EXTRA_FLAGS="-march=core-avx2" sh install.sh intel
```

- CMAKE を用いて cmake コマンドのパスを指定可能
- ISSP_UCOUNT は物性研スパコンにおける利用率測定スクリプトのパスであり、ほとんどのユーザは気にしなくて問題ない
- そのほか、利用可能な変数は install.sh のはじめの方を参照のこと
- sh runttest.sh で簡易テストを実行可能
 - インストールディレクトリの存在確認
 - 設定ファイルの有効性確認
 - ソフトウェアが実際に動作するかの確認
- ソフトウェアは \$MA_ROOT/NAME/NAME-VERSION-MA_REVISION ディレクトリにインストールされる
 - NAME, VERSION はそれぞれソフトウェア名とバージョンに置き換わる
 - * MA_REVISION は、ソフトウェアの同一バージョンに対して MateriApps Installer が改訂された場合に区別するための識別子
 - * 例: hphi/hphi-3.4.0-1
 - ソフトウェアと共に、環境変数などを設定する設定ファイル NAMEvars-VERSION-MA_REVISION.sh が \$MA_ROOT/NAME/ にインストールされる
 - * 例: hphivars-3.4.0-1.sh
 - * sh link.sh を実行することで、NAMEvars-VERSION.sh のシンボリックリンク NAMEvars.sh が作成される
 - ・ アプリの場合は NAME 以下に作成される
 - ・ ツールの場合は \$MA_ROOT/env.d 以下に作成され、\$MA_ROOT/env.sh 内で読み込まれる

2.5 ツール・アプリの利用方法

- ツール類 (cmake, hdf5, python 他)
 - 以下のコマンドを実行する (もしくは shell の初期化スクリプト (.bashrc など) に同じ内容を書いておく)

```
source $MA_ROOT/env.sh
```

- アプリケーション (alps, openmx, modylas 他)
 - アプリ毎に設定ファイルを読み込んで環境変数 (PATH など) を設定する

例) alps の場合:

```
source $MA_ROOT/alps/alpsvars.sh
```

- バージョンを固定したい場合は、そのバージョンの設定ファイルを用いる

```
source $MA_ROOT/alps/alpsvar-20201106-r7860-1.sh
```

第 3 章

チュートリアル

3.1 macOS 10.15 (Catalina) へのインストール

本節では、macOS 10.15 (Catalina) で MateriApps Installer を利用する方法を記載します。

3.1.1 ツールのインストール

tools ディレクトリにはいつているツール類は、macOS のパッケージ管理ソフト (Homebrew, Fink, MacPorts など) を用いてインストールするほうが便利です。ここでは Homebrew を用いる方法を紹介します。アプリのインストール先はデフォルトのままであるとします。インストール先は\$HOME/materiapps, ビルドディレクトリは\$HOME/build になります。

まず <https://brew.sh/>にある情報に従ってインストールします。

```
$ /bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/
↪install.sh)"
```

次にツール類を Homebrew によってインストールします。他の方法でインストールがすでになされている場合は、Homebrew でインストールする必要はありません。以下のツールははじめにすべてインストールしてもいいですし、アプリのインストールで要求されたときにインストールするようにしても構いません。

```
$ brew install gcc
$ brew install boost
$ brew install cmake
$ brew install eigen
$ brew install fftw
$ brew install git
$ brew install gsl
$ brew install hdf5
$ brew install lapack
$ brew install libffi
```

(次のページに続く)

(前のページからの続き)

```
$ brew install openblas
$ brew install openmpi
$ brew install openssl
$ brew install python@3
$ brew install scalapack
$ brew install tcl-tk
$ brew install zlib
```

またアプリによっては、以下のツールもインストールしておく必要があります。

```
$ brew install svn
$ brew install boost-python
$ brew install boost-python3
$ brew install boost-mpi
$ brew install wget
```

さらにアプリによっては `python` のライブラリのインストールが必要な場合があります。以下のコマンドによってインストールしておきます。

```
$ pip3 install numpy --user
$ pip3 install scipy --user
$ pip3 install toml --user
```

3.1.2 アプリのインストール

まず最初のセットアップ (必要なディレクトリを作成するなど) を行います。MateriAppsInstaller のディレクトリにはいり、

```
$ sh setup/setup.sh
```

を実行します。次にインストールしたいアプリのディレクトリに入り、

```
$ CC=gcc-10 CXX=g++-10 FC=gfortran-10 CPP=cpp-10 sh install.sh
```

を実行すればインストールができるはずです。正しくインストールされているかどうかは、各アプリのディレクトリで

```
$ sh runtest.sh
```

を実行することで確認できます。(repack の runtest.sh を実行する際には、予め quantum ESPRESSO をインストールしておく必要があります。また次の節で述べる方法により、quantum ESPRESSO の実行環境を設定しておく必要があります。)

アプリのダウンロード・インストールの途中で一時停止すると、ソースファイルが残ったままになり、再度インス

ツールを行ったときにうまくいかないことがあります。その場合は、ソースファイル (デフォルトではホームディレクトリの `materiapps/source` 以下) にある対象アプリのディレクトリをすべて削除してください。

インストール時にエラーがでた場合は、エラーメッセージをよく見てください。多くの場合、必要なツール類がインストールされていないためにエラーが生じます。エラーメッセージから、必要となるツール類を読み取り、ツールのインストールを行ってください。

3.1.3 各アプリの実行環境の設定

アプリを実行するためには、実行環境の設定を行う必要があります。以下のコマンドを入力することにより、各アプリの実行環境を整え、すぐにアプリが実行できる環境を整備することができます。例えば、H Φ をインストールしてある場合 (インストールディレクトリがデフォルトの `materiapps` であったとする) は、

```
$ ls $HOME/materiapps/hphi
```

でファイルを表示させたときに、`hphivars.sh` という名前の設定ファイルがあるはずです。この設定ファイルを下記のように実行し、実行環境の設定を行います。

```
$ source $HOME/materiapps/hphi/hphivars.sh
```

3.2 Linux へのインストール (gcc 利用)

本節では、LinuxOS 上で MateriApps Installer を利用する方法を記載します。動作確認は CentOS 7, Ubuntu 20 で行っています。(他のディストリビューションでも同じ手順でインストール可能のはずです。) ツール類 (`gcc`, `cmake` など) はあらかじめ OS にインストールされているものを用いても構いませんが、万一何らかのトラブルが生じたときにエラーメッセージから原因を推測し適切に対応する必要があります。この作業に自信がない方は、すべてのツールを MateriApps Installer によってインストールすることをお勧めします。

3.2.1 初期セットアップ

まず最初のセットアップ (必要なディレクトリを作成するなど) を行います。MateriApps Installer のディレクトリにはいり、

```
$ sh setup/setup.sh
```

を実行します。この作業は最初に一回だけ行えば十分です。

3.2.2 gcc のインストール

まずコンパイラをインストールします。(あらかじめ LinuxOS にインストールされている gcc コンパイラを用いて以降のツール・アプリをインストールすることも可能ですが、一部のアプリは古い gcc コンパイラではうまくインストールできません。トラブルの処理に慣れていない方以外は、MateriAppsInstaller で gcc コンパイラを導入することをお勧めします。) 以下の手順で gcc10 をインストールします。

```
$ cd tools/gcc10
$ sh install.sh
```

gcc10 のインストールには長い時間がかかります。遠隔で計算機サーバなどにインストールする場合は、最後のコマンドは以下のように変更してバックグラウンドで実行するとよいかもかもしれません。

```
$ sh install.sh > log &
```

この場合はインストール作業の進行状況は同じディレクトリに生成されるログをみるとわかります (例えば cat log を実行する)。インストールが成功したら、以下のコマンドを実行します。

```
$ sh link.sh
```

これにより、ツールを利用する際の初期設定を行うスクリプトのリンクが所定の場所 (\$HOME/materiapps/env.d) にコピーされます。最後にこのコピーされた設定ファイルを実行し、以降のツール・アプリで利用可能な状態にします。

```
$ source $HOME/materiapps/env.sh
```

gcc10 がうまくインストールできた場合は、gcc8 はインストールする必要はありません。まれに古い Linux OS では gcc10 のインストールに失敗する場合があります。その場合にのみ、gcc8 をインストールしてください。インストール方法は最初に tools/gcc8 に入る以外は上述と同じです。

3.2.3 cmake のインストール

一部のアプリは cmake を利用してインストールを行います。cmake はあらかじめ Linux のディストリビューションに含まれることが多いですが、古いバージョンの cmake を用いると、一部アプリのインストールに失敗する場合があります。バージョン 3.6 以降であれば問題ありません。cmake をインストールするには、gcc コンパイラをインストールした直後の状況で以下のコマンドを実行します。

```
$ cd ../cmake
$ sh install.sh
$ sh link.sh
$ source $HOME/materiapps/env.sh
```

3.2.4 その他のツールのインストール

同様の方法で残りのツールのインストールも行います。例えば、`cmake` のインストール直後に、引き続いて `git` をインストールする場合は、以下のコマンドを実行します。

```
$ cd ../git
$ sh install.sh
$ sh link.sh
$ source $HOME/materiapps/env.sh
```

すべてのツールをインストールする場合には、まず `git`, `python3`, `libffi` を先にインストールし、残りのツールをアルファベット順 (`boost`, `eigen3`, `fftw`, `gsl`, `hdf5`, `lapack`, `libffi`, `openmpi`, `openssl`, `scalapack`, `tcltk`, `zlib`) にインストールするとよいでしょう。(`scalapack` のインストールを行う前に `openmpi` を先にインストールする必要があります。) コマンドは最初にはいるディレクトリ名をインストールしたツール名に変更する以外全く同じです。

3.2.5 アプリのインストール

ツール類のインストールが終了したら、下記のようにインストールしたいアプリのディレクトリに入り、ツールの実行環境設定を行ったあとに、インストールスクリプトを実行します。

```
$ sh $HOME/materiapps/env.sh
$ sh install.sh
```

正しくインストールされているかどうかは、各アプリのディレクトリで

```
$ sh runtest.sh
```

を実行することで確認できます。(`respack` の `runtest.sh` を実行する際には、あらかじめ `quantum ESPRESSO` をインストールしておく必要があります。また次の節で述べる方法により、`quantum ESPRESSO` の実行環境を設定しておく必要があります。)

アプリのダウンロード・インストールの途中で一時停止すると、ソースファイルが残ったままになり、再度インストールを行ったときにうまくいかないことがあります。その場合は、ソースファイル (デフォルトではホームディレクトリの `materiapps/source` 以下) にある対象アプリのディレクトリを削除してください。

インストール時にエラーがでた場合は、エラーメッセージをよく見てください。多くの場合、必要なツール類がインストールされていないためにエラーが生じます。エラーメッセージから、必要となるツール類を読み取り、ツールのインストールを行ってください。

テストが無事終了したら、最後に以下のコマンドでアプリの環境設定ファイルを適切なファイル名でリンクします。

```
$ sh link.sh
```

3.2.6 各アプリの実行環境の設定

アプリを実行するためには、実行環境の設定を行う必要があります。以下のコマンドを入力することにより、各アプリの実行環境を整え、すぐにアプリが実行できる環境を整備することができます。例えば、H Φをインストールしてある場合(インストールディレクトリがデフォルトの `materiapps` であったとする)は、

```
$ ls $HOME/materiapps/hphi
```

でファイルを表示させたときに、`hphivars.sh` という名前の設定ファイルがあるはずです。この設定ファイルを下記のように実行し、実行環境の設定を行います。

```
$ source $HOME/materiapps/hphi/hphivars.sh
```

第 4 章

ファイル形式

4.1 インストール関連ファイル

ここでは、ソフトウェアのインストールに必要な各ファイルに関するフォーマットやルールなどを説明します。

4.1.1 version.sh

ソフトウェアのバージョンを指定するためのスクリプト。

アンダースコア 2 つずつに挟まれた変数は他のスクリプト中で用いるための変数。

- `<NAME>_VERSION`
 - ソフトウェアのバージョン
- `<NAME>_MA_REVISION`
 - 同一バージョンに対して MAInstaller のスクリプトが改訂された場合、それを区別するための識別子
- `__NAME__`
 - ディレクトリ名などに用いられる、ソフトウェアの名前
 - MAInstaller では、スタイルとして小文字に統一している
- `__VERSION__`
 - `<NAME>_VERSION` を指定する
- `__MA_REVISION__`
 - `<NAME>_MA_REVISION` を指定する

例

```
TENES_VERSION="1.1.2"
TENES_MA_REVISION="0"

__NAME__=tenes
__VERSION__=${TENES_VERSION}
__MA_REVISION__=${TENES_MA_REVISION}
```

4.1.2 install.sh

ソフトウェアをインストールするスクリプト。

- 引数として設定名 (mode) をとる
 - 省略した場合は default
 - 存在しない設定名を与えた場合、利用可能な設定名の一覧を返す
- setup.sh を自動で呼び出し、ソースコードをダウンロード・展開する
- config/\$mode 以下にある preprocess.sh, build.sh, install.sh, postprocess.sh を自動で呼び出す
 - preprocess.sh
 - * configure や cmake など、ビルドの前処理をする
 - build.sh
 - * ソフトウェアをコンパイルする
 - install.sh
 - * ソフトウェアをインストールする
 - postprocess.sh
 - * ラッパーの作成やリネームなどの後処理
 - config/\$mode 以下にスクリプトがない場合、config/default 以下のものが実行される
 - * config/default にもない場合はスキップ

4.1.3 download.sh

ソースコードをインターネットから入手するためのスクリプト。

- \$SOURCE_DIR にダウンロードする
 - 保存されたファイルはフォーマットに従ってリネームしておく
 - * ソフトウェアごとの違いをできるだけ減らすため
 - 特に、バージョン番号は必ず付与し、区別可能にしておく
- すでにダウンロード済みの場合はスキップする
 - 外部アクセスできない環境の場合は、手動でファイルを置けるようにしてある

例

```
SCRIPT_DIR=$(cd "$(dirname $0)"; pwd)
. $SCRIPT_DIR/../../scripts/util.sh
. $SCRIPT_DIR/version.sh
set_prefix

if [ -f $SOURCE_DIR/${__NAME__}-${__VERSION__}.tar.gz ]; then ;; else
  check wget https://github.com/issp-center-dev/TeNeS/archive/v${__VERSION__}.tar.gz -
  ↪O $SOURCE_DIR/${__NAME__}-${__VERSION__}.tar.gz
fi
```

4.1.4 setup.sh

アーカイブファイルを展開するためのスクリプト。

- download.sh を実行してアーカイブファイルを取得する
 - すでにダウンロードされているかのチェックは download.sh が行う
- \$SOURCE_DIR 内にあるアーカイブファイルを \$BUILD_DIR に展開する
 - 展開後のディレクトリは、フォーマットに従ってリネームしておく
- patch ディレクトリ内にパッチファイルがある場合にはそれを適用する

例

```
SCRIPT_DIR=$(cd "$(dirname $0)"; pwd)
. $SCRIPT_DIR/../../scripts/util.sh
. $SCRIPT_DIR/version.sh
set_prefix
```

(次のページに続く)

```

sh ${SCRIPT_DIR}/download.sh

cd $BUILD_DIR
if [ -d ${__NAME__}-${__VERSION__} ]; then ;; else
  check mkdir -p ${__NAME__}-${__VERSION__}
  tarfile=$SOURCE_DIR/${__NAME__}-${__VERSION__}.tar.gz
  sc=`calc_strip_components $tarfile README.md`
  check tar zxf $tarfile -C ${__NAME__}-${__VERSION__} --strip-components=$sc
  cd ${__NAME__}-${__VERSION__}
  if [ -f $SCRIPT_DIR/patch/${__NAME__}-${__VERSION__}.patch ]; then
    patch -p1 < $SCRIPT_DIR/patch/${__NAME__}-${__VERSION__}.patch
  fi
fi

```

4.1.5 link.sh

設定ファイルのシンボリックリンクを作成するスクリプト。

- アプリケーションの場合とツールの場合でシンボリックリンクの作成先が異なる
 - アプリの場合は \$MA_ROOT/\${__NAME__}/\${__NAME__}vars.sh
 - ツールの場合は \$MA_ROOT/env.d/\${__NAME__}vars.sh

例

```

SCRIPT_DIR=$(cd "$(dirname $0)"; pwd)
. $SCRIPT_DIR/../../scripts/util.sh
. $SCRIPT_DIR/version.sh
set_prefix

. $MA_ROOT/env.sh

VARS_SH=$MA_ROOT/${__NAME__}/${__NAME__}vars-${__VERSION__}-${__MA_REVISION__}.sh
rm -f $MA_ROOT/${__NAME__}/${__NAME__}vars.sh
ln -s $VARS_SH $MA_ROOT/${__NAME__}/${__NAME__}vars.sh

```


4.1.6 README.md

ソフトウェアの情報について記載してあります。具体的には、

- アプリ名
- SUMMARY
- LICENSE
- OFFICIAL PAGE
- MateriApps URL

に関する情報が記載してあります。各情報については、物質科学シミュレーションのポータルサイト MateriApps より引用されています。

例

```

H Φ
==

SUMMARY
-----

並列計算機に対応した数値厳密対角化法による有効模型ソルバーパッケージ。
広汎な多体量子系の有効模型（多軌道ハバード模型、ハイゼンベルグ模型、近藤格子模型など）
の基底状態及び低励起状態の波動関数を並列計算によって求める。
ランチョス法による基底状態計算、熱的純粋量子状態を利用した
比熱・帯磁率の温度依存性計算が可能。さらに、シフト型クリロフ部分空間ライブラリ K ω を用いて
動的グリーン関数の計算が可能である。ver.3.0 からは実時間発展の機能も追加された。

LICENSE
-----

GNU GPL version 3

OFFICIAL PAGE
-----

http://www.pasums.issp.u-tokyo.ac.jp/hphi/

MateriApps URL
-----

https://ma.issp.u-tokyo.ac.jp/app/339

```

4.2 管理用ファイル (scripts/)

4.2.1 util.sh

スクリプト内で使う関数を定義しているファイル

4.2.2 check_prefix.sh

インストール先 (MA_ROOT)、ソースコードダウンロード先 (SOURCE_DIR)、ビルド作業用ディレクトリ (BUILD_DIR) を表示するスクリプト

4.2.3 fix_dylib.sh

macOS で共有ライブラリの名前などを修正するスクリプト

4.2.4 list_maversion.sh

MateriApps Installer でサポートされている各種アプリのバージョンを列挙するスクリプト

第 5 章

アプリ一覧

5.1 ALPS

5.1.1 SUMMARY

磁性体や相互作用電子系などの強相関格子模型を取り扱う数値計算ライブラリ。モンテカルロ法、厳密対角化法、密度行列繰り込み群などの代表的な強相関系ソルバーが利用できる。相互作用スピン系の比熱・帯磁率や磁化過程、強相関電子系の状態密度計算などを行うことができる。高効率の並列計算用スケジューラも整備されている。

5.1.2 LICENSE

ALPS Licence

5.1.3 OFFICIAL PAGE

<http://alps.comp-phys.org>

5.1.4 MateriApps URL

<https://ma.issp.u-tokyo.ac.jp/app/213>

5.2 ALPSCore

5.2.1 SUMMARY

計算物理アプリ構築のためのオープンソースソフトウェア。量子モンテカルロ法などの多くの計算プログラムで必要となる共通の C++ 補助モジュールが整備されている。再利用可能なコード構築を補助し、複雑な計算科学アプリケーションの開発時間の短縮が可能。MPI および OpenMP の並列プログラミングにも対応している。

5.2.2 LICENSE

The GNU General Public License (GPL), Version 2

5.2.3 OFFICIAL PAGE

<https://alpscore.org/>

5.2.4 MateriApps URL

<https://ma.issp.u-tokyo.ac.jp/app/2871>

5.3 DSQSS

5.3.1 SUMMARY

格子上で定義された量子多体系の計算プログラム。連続虚数時間向き付きループアルゴリズムに基づく量子モンテカルロ法により、量子多体系の各種物理量を計算する。格子の形状、相互作用、スピンの大きさ、磁場、温度などを入力する事ができ、量子スピン系とボーズ系のシミュレーションが可能。

5.3.2 LICENSE

GNU General Public License (GPL) Version 3 に基づく。

5.3.3 OFFICIAL PAGE

<https://github.com/issp-center-dev/dsqss>

5.3.4 MateriApps URL

<https://ma.issp.u-tokyo.ac.jp/app/171>

5.4 QUANTUM ESPRESSO

5.4.1 SUMMARY

擬ポテンシャル法と平面波基底を用いた第一原理計算ライブラリ。広範な物理系に対して、密度汎関数法に基づく電子状態計算を高精度で行うことができる。基本プログラムのほかに多数のコアパッケージ・プラグインが含まれ、無償ながら研究・開発に利用できる多くの充実した機能を持つ。MPI による並列計算にも対応している。

5.4.2 LICENSE

GNU GPL v2

5.4.3 OFFICIAL PAGE

<http://www.quantum-espresso.org/>

5.4.4 MateriApps URL

<https://ma.issp.u-tokyo.ac.jp/app/720>

5.5 H Φ

5.5.1 SUMMARY

並列計算機に対応した数値厳密対角化法による有効模型ソルバーパッケージ。広汎な多体量子系の有効模型 (多軌道ハバード模型、ハイゼンベルグ模型、近藤格子模型など) の基底状態及び低励起状態の波動関数を並列計算によって求める。ランチョス法による基底状態計算、熱的純粋量子状態を利用した比熱・帯磁率の温度依存性計算が可能。さらに、シフト型クリロフ部分空間ライブラリ $K \omega$ を用いて動的グリーン関数の計算が可能である。ver. 3.0 からは実時間発展の機能も追加された。

5.5.2 LICENSE

GNU GPL version 3

5.5.3 OFFICIAL PAGE

<http://www.pasums.issp.u-tokyo.ac.jp/hphi/>

5.5.4 MateriApps URL

<https://ma.issp.u-tokyo.ac.jp/app/339>

5.6 K ω

5.6.1 SUMMARY

シフト型クリロフ部分空間解法を用いた大規模疎行列向け並列化数値計算ルーチン。ハミルトニアン行列と状態ベクトルを入力として動的相関関数を効率よく計算することが可能。汎用な数値ライブラリであり、多体量子模型ソルバー H Φ での動的グリーン関数の計算機能の実装などに利用されている。また、Heisenberg 模型などの量子格子模型の動的相関関数を計算するミニアプリが付属している。

5.6.2 LICENSE

GNU LESSER GENERAL PUBLIC LICENSE Version 3

5.6.3 OFFICIAL PAGE

<https://github.com/issp-center-dev/Komega>

5.6.4 MateriApps URL

<https://ma.issp.u-tokyo.ac.jp/app/534>

5.7 LAMMPS

5.7.1 SUMMARY

オープンソースの汎用古典分子動力学アプリケーション。ソフトマター、固体、メソスコピック系などの多くの系で動力学計算を行うことができる。原子の動力学計算や一般的な粒子のシミュレーターとしても利用可能で、空間分割を用いた並行計算にも対応する。GPL ライセンスを採用し、コードは変更や拡張が容易となるようにデザインされている。

5.7.2 LICENSE

GPLv2

5.7.3 OFFICIAL PAGE

<https://lammmps.sandia.gov>

5.7.4 MateriApps URL

<https://ma.issp.u-tokyo.ac.jp/app/596>

5.8 mVMC

5.8.1 SUMMARY

広汎な多体量子系の有効模型 (多軌道ハバード模型、ハイゼンベルグ模型、近藤格子模型など) の基底状態の高精度な波動関数を変分モンテカルロ法によって数値的に求める有効模型ソルバーパッケージ。グッツヴィラー・ジャストロー、ダブロン-ホロン束縛因子の相関因子を取り扱うことが可能であり、一万以上の変分パラメータを最適化することが可能である。また、量子数射影によって量子数を指定することで低エネルギー励起状態も求めることが可能である。

5.8.2 LICENSE

GNU General Public License version 3

5.8.3 OFFICIAL PAGE

<http://www.pasums.issp.u-tokyo.ac.jp/mvmc/>

5.8.4 MateriApps URL

<https://ma.issp.u-tokyo.ac.jp/app/81>

5.9 OpenMX

5.9.1 SUMMARY

原子局在基底と擬ポテンシャルを用いた第一原理計算プログラム。結晶・界面・溶液などの広範な物理系に対して、局在基底を利用した高速の電子状態計算を行う。大規模系に対して分子動力学計算や構造最適化を速やかに実行でき、ハイブリッド並列も実装する。ノンコリニア磁性や非平衡グリーン関数法による電気伝導計算にも対応。

5.9.2 LICENSE

GNU-GPL

5.9.3 OFFICIAL PAGE

<http://www.openmx-square.org/>

5.9.4 MateriApps URL

<https://ma.issp.u-tokyo.ac.jp/app/594>

5.10 RESPACK

5.10.1 SUMMARY

物質の相互作用パラメータを評価する第一原理計算ソフトウェア。最局在ワニエ関数、RPA 応答関数、周波数依存電子間相互作用パラメータが計算可能。ノルム保存型擬ポテンシャル + 平面波基底を用いるバンド計算に対応しており、xTAPP および Quantum ESPRESSO に関しては、ファイルの変換スクリプトが用意されている。金属、半導体、遷移金属化合物、有機化合物など広範な物質群を計算できる。OpenMP / MPI に対応。

5.10.2 LICENSE

GNU GPL v3

5.10.3 OFFICIAL PAGE

<https://sites.google.com/view/kazuma7k6r>

5.10.4 MateriApps URL

<https://ma.issp.u-tokyo.ac.jp/app/767>

5.11 TeNeS

5.11.1 SUMMARY

テンソルネットワーク法、特に PEPS 波動関数状態と角転送行列くりこみ群法を用いた、二次元量子格子模型の基底状態ソルバー。

5.11.2 LICENSE

GNU GPLv3

5.11.3 OFFICIAL PAGE

<https://github.com/issp-center-dev/TeNeS>

5.11.4 MateriApps URL

<https://ma.issp.u-tokyo.ac.jp/app/2290>

第 6 章

お問い合わせ

MateriApps Installer に関するお問い合わせはこちらにお寄せください。

- バグ報告

MateriApps Installer のバグ関連の報告は [GitHub](#) の [Issues](#) で受け付けています。

バグを早期に解決するため、報告時には次のガイドラインに従ってください。

- 使用している MateriApps Installer のバージョンを指定してください。
- インストールに問題がある場合には、使用しているオペレーティングシステムとコンパイラの情報についてお知らせください。
- 実行に問題が生じた場合は、実行に使用した入力ファイルとその出力を記載してください。

- その他

研究に関連するトピックなど [GitHub](#) の [Issues](#) で相談しづらいことを問い合わせる際には、以下の連絡先にコンタクトをしてください。

E-mail: mainstaller-dev__at__issp.u-tokyo.ac.jp ([_at_](#)を@に変更してください)

第 7 章

謝辞

このソフトウェアの開発は、以下のプロジェクトとコンピューター資源の提供によりサポートされてきました。この場を借りて感謝します。

- 文部科学省 HPCI 戦略プログラム (SPIRE) 分野 2「新物質・エネルギー創成」
- 文部科学省 ポスト「京」重点課題 7「次世代の産業を支える新機能デバイス・高性能材料の創成」

また、本ソフトウェアは東京大学物性研究所 ソフトウェア高度化プロジェクト (2020 年度) の支援を受け開発されました。この場を借りて感謝します。