



# VASP ユーザーマニュアル



# 目次

1	VASP について .....	2
2	VASP 実行ファイル .....	3
2.1	vasp_std .....	3
2.2	vasp_gam .....	3
2.3	vasp_ncl .....	3
2.4	vasp_w90 .....	3
2.5	vasp_beef .....	4
3	VASP 実行例 .....	5
3.1	VASP 実行環境の設定 .....	5
3.2	シリアル(1CPU コア)の実行例 .....	8
3.3	MPI 並列の実行例 .....	8
4	既知の問題点 .....	9
5	可視化ツール .....	10
5.1	VESTA .....	10
5.2	VMD .....	10
付録 A	.....	11
A.1	HPC システムズ お問い合わせ先 .....	11

# 1 VASP について

---

VASP は、密度汎関数法による平面波・擬ポテンシャル基底を用いた第一原理電子状態計算プログラムパッケージです。結晶や無機固体の周期境界条件を用いた計算を行います。VASP はウィーン大学で開発された有償のソフトウェアです。VASP を使用する際は開発元とライセンス契約を行う必要があります。ライセンス契約の詳細については VASP ホームページ(<http://www.vasp.at/>) の FAQs をご確認ください。

VASP のパッケージはライセンス契約をしているユーザーであれば VASP ホームページからダウンロードをすることが可能です。当社では VASP のファイルをお借りして計算機へインストールをしています。

本マニュアルでは VASP の実行方法をご案内します。VASP の詳細については VASP Wiki をご確認ください。

[https://www.vasp.at/wiki/index.php/The\\_VASP\\_Manual](https://www.vasp.at/wiki/index.php/The_VASP_Manual)

## 2 VASP 実行ファイル

---

VASP の実行ファイルは、標準の `vasp_std`、`vasp_gam`、`vasp_ncl` と、`makefile.include` のパラメータを修正して外部ソースコード由来のライブラリを加えてビルドした `vasp_w90`、`vasp_beef` の 5 種類があります。

### 2.1 `vasp_std`

ビルド時に `DNGZhalf` の宣言をつけてビルドを行ったものです。基本的な VASP の実行ファイルです。

### 2.2 `vasp_gam`

ビルド時に `DwNGZhalf` の宣言をつけてビルドを行ったものです。ガンマ点のみの計算をする VASP の実行ファイルです。

VASP Wiki に記載されておりますので、詳しくは以下 URL をご参照下さい。

[https://www.vasp.at/wiki/index.php/Installing\\_VASP](https://www.vasp.at/wiki/index.php/Installing_VASP)

### 2.3 `vasp_ncl`

INCAR 中の `LNONCOLLINER` タグや `LSORBIT` タグを使用して計算を行う機能です。

VASP Wiki に記載されておりますので、詳しくは以下 URL をご参照下さい。

[https://www.vasp.at/wiki/index.php/Installing\\_VASP](https://www.vasp.at/wiki/index.php/Installing_VASP)

### 2.4 `vasp_w90`

Wannier90 対応の VASP で、`DVASP2WANNIER90v2` を付けて `wannier90` のライブラリとリンクしてビルドした VASP です。使用している Wannier90 のバージョンは 2.1.0 です。`LWANNIER90` タグが使用可能です。

VASP Wiki に記載されておりますので、詳しくは以下 URL をご参照下さい。

[https://www.vasp.at/wiki/index.php/Installing\\_VASP](https://www.vasp.at/wiki/index.php/Installing_VASP)

Wannier90 は以下の URL に詳細が記載されているので、ご参照下さい。

<https://www.wannier90.org>

## 2.5 vasp\_beef

Dlibbeef を付けて、BEEF-vdW 汎関数を使用可能にする libbeef に対応した VASP です。使用バージョンは 0.1.1 です。

VASP Wiki に記載されておりますので、詳しくは以下 URL をご参照下さい。

[https://www.vasp.at/wiki/index.php/Installing\\_VASP](https://www.vasp.at/wiki/index.php/Installing_VASP)

libbeef に関する詳細は GitHub の以下 URL をご参照下さい。

<https://github.com/vossjo/libbeef/blob/master/README>

## 3 VASP 実行例

### 3.1 VASP 実行環境の設定

弊社出荷の RHEL/AlmaLinux 8 系 OS においては、VASP の実行環境の設定を、VASP を実行したいタイミングで、[Environment Module](#) と呼ばれる環境設定ユーティリティを用いて実施ください。Environment Module ではアプリケーション等のソフトウェア部品の環境設定を「モジュール」と呼び、`module load`・`module unload` というコマンドによりモジュールの有効化・無効化を実施できます。モジュールが有効化されているとき、そのアプリケーション等は、PATH 等の環境設定が済んで実行可能な状態になっています。

Environment Module で VASP を使用可能とするモジュール定義ファイルは次の場所に設置してあります。

```
/home/.common/modulefiles/oneAPI/oneAPI バージョン/VASP/VASP バージョン
```

以下では Environment Module を用いた VASP 実行環境設定の方法を示します。

(1) 使用できるモジュールの一覧表示 : `module avail`

`module avail` コマンドにより、使用可能なモジュールの一覧を表示します。次は VASP 6.4.2 の場合の例です。

```
$ module_ avail
----- /home/.common/modulefiles/oneAPI/2023.2.0 -----
VASP/6.4.2
(後略)
```

赤文字で示したように、使用可能なモジュール（ソフトウェア名/バージョン）が表示されます。

(2) モジュールの有効化・無効化 : `module load` ・ `module unload`

モジュールを有効にするコマンドは `module load` です。その後ろにモジュール名を付けて実行します。例えば VASP 6.4.2 のモジュールを有効化する場合、具体的なコマンドは次となります。

```
$ module load VASP/6.4.2
```

逆に、有効化したモジュールを無効化するコマンドは `module unload` です。同様に、後ろにモジュール名を付けて実行します。

```
$ module unload VASP/6.4.2
```

`module unload` は、有効化したモジュールによる環境設定が別の作業に悪影響を及ぼす際などに使用ください。有効化したモジュールによる環境設定が特に悪影響を及ぼしていない場合には、わざわざ無効化する必要はありません。

正常にモジュールを読み込めたかどうかについては、次の `module list` コマンドで確認できます。

(3) 有効になっているモジュールの表示 : `module list`

`module list` により、Environment Module で有効化されているモジュールの一覧を表示します。

```
$ module list
```

例えば VASP/6.4.2 のモジュールが有効になっている場合、次のように Currently Loaded Modulefiles: 行より下に表示されます。

```
# module list
Currently Loaded Modulefiles:
 1) tbb/2021.10.0          4) mk1/2023.2.0
 2) compiler-rt/2023.2.1 5) mpi/2021.10.0
 3) compiler/2023.2.1    6) VASP/6.4.2
```

#### (4) ユーザーシェルログイン時にモジュールを自動的に有効化する方法

ユーザーログイン時に、自動的にモジュールの有効化を行いたい場合、ユーザーのシェル環境設定ファイルに `module load` コマンドを追記してください。

VASP 6.4.2 の場合の具体的な修正例を次に示します。

- ☞ Bash をお使いの場合  
ホームディレクトリの `.bashrc` の最終行に以下の追記を行います。
  
- ☞ Tcsh をお使いの場合  
ホームディレクトリの `.cshrc` の最終行に以下の追記を行います。

```
# ----- 任意のコメント -----  
# module_load_VASP/6.4.2
```

モジュールの有効化が他のアプリケーションに悪影響を及ぼすようなケースも存在いたしますので、ログイン時のシェル環境における自動有効化については、有効時の影響に十分注意した上で行うようお願いいたします。

## 3.2 シリアル(1CPU コア)の実行例

入力ファイルがあるディレクトリに移動して vasp コマンドを実行します。次は VASP として vasp\_std を用いる例です。

```
$ cd _jobdir
$ ls
INCAR KPOINTS POSCAR POTCAR
$ vasp_std
```

## 3.3 MPI 並列の実行例

入力ファイルがあるディレクトリに移動して mpirun を用いて vasp コマンドを実行します。次は VASP として vasp\_std を用いる例です。

```
$ cd _jobdir
$ ls
INCAR KPOINTS POSCAR POTCAR
$ mpirun -np 16 /usr/local/VASP-6.4.2/bin/vasp_std
```

※ -np 以降に並列数を指定します。-np の値は 2 のべき乗の数字でない場合、VASP が動作しないことがあるのでご注意ください。

※ mpirun 以降の実行ファイルは絶対パスで指定する必要があります。実行ファイルがあるディレクトリにパスが通っていても同様です。

## 4 既知の問題点

---

2024 年 1 月 30 日現在、VASP 5.4.4p1、VASP 6.4.2 で明らかになっている問題点はありません。

VASP のビルドの際に動作確認の際に使用したテスト、およびその実行結果については、`/opt/hpcs/app_doc` 以下に PDF 文書としてまとめてありますので、そちらをご参照ください。

## 5 可視化ツール

---

VASP はテキスト形式で入力用データファイルを作成し、計算結果もテキスト形式で出力されます。VASP のユーザーはこうした形式に慣れていますが、VASP をこれから始めるという場合や、VASP の計算結果を可視化したい場合、簡便なツールはないかとお問い合わせを頂く場合があります。VASP の可視化に使えるソフトウェアとして知られているものを以下に紹介いたします。セットアップについては弊社までお問い合わせください。

### 5.1 VESTA

結晶構造、電子・核密度等の三次元データ、及び結晶外形の可視化プログラムです。

配布元 : <https://jp-minerals.org/vesta/jp/>

### 5.2 VMD

結晶構造や電子密度分布を可視化することができます。

配布元 : <https://www.ks.uiuc.edu/Research/vmd/>

# 付録A

---

## A.1 HPC システムズ お問い合わせ先



弊社ホームページ [http://www.hpc.co.jp/support\\_index.html](http://www.hpc.co.jp/support_index.html)

サポート案内やお問い合わせの多い内容など様々な情報を掲載しております。  
是非ご活用ください。

### HPC システムズ株式会社

〒108-0022 東京都港区海岸 3-9-15 LOOP-X 8 階

### HPC 事業部



【営業】 03-5446-5531    【サポート】 03-5446-5532

お電話によるサポート受付は祝日、弊社指定休日を除く月曜日から金曜日の 9:30～17:30  
とさせていただきます。



【FAX】 03-5446-5550



【電子メール】 [hpcs\\_support@hpc.co.jp](mailto:hpcs_support@hpc.co.jp)