

AFLOW V 3.1.111

```
*****
*
*           Aflow STEFANO CURTAROLO - Duke University 2003-2013
*           High-Throughput ab-initio Materials Discovery
*
*****
```

Thermal properties calculated using a quasi-harmonic Debye model.
This implementation in AFLOW written by Cormac Toher and Stefano Curtarolo,
and is described in the paper Phys. Rev. B 90, 90, 174107 (2014).
Based on the GIBBS program, described in the publications
Computer Physics Communications 158, 57-72 (2004) and
Journal of Molecular Structure (Theochem) 368, 245-255 (1996).
Please cite the 3 papers mentioned above in addition to the standard AFLOW papers
if you use results obtained using the AFLOW AGL code.

The current version of AGL can calculate the thermal conductivity, Debye temperature, Gruneisen parameter, heat capacity, vibrational free energy, and other thermal properties, as a function of temperature and pressure. The results are written in a variety of formats in different output files. The main output file is named `aflow.agl.out`, and contains the AGL results for the thermal conductivity at 300K, the traditional and acoustic Debye temperatures, the Gruneisen parameter, the heat capacities at constant volume and constant pressure (C_p) at 300K, the thermal expansion at 300K, and the static and isothermal bulk modulus at 300K. It also contains the results for the thermal conductivity, Debye temperature, Gruneisen parameter, heat capacity, expansion and bulk modulus as function of temperature; and the Gibbs free energy, vibrational Helmholtz free energy, vibrational internal energy, and vibrational entropy as a function of temperature. More information can be found in the other main output file `AGL.out`, which contains the results for the thermal properties and bulk moduli for all temperatures and pressures. The values for the vibrational internal energy, vibrational free energy, entropy, heat capacity, Debye temperature and Gruneisen parameter as a function of temperature at zero pressure are written to the file `AGL_THERMO`, in a similar format to that used by the `THERMO` file written by AFLOW APL. For further details on the output files written, see points (14) and (15) below. Results can be written and plotted by setting the appropriate values in the `aflow.in` file, as described in point (15) below.

AFLOW AGL expects a well relaxed structure on input. There is no additional relaxation inside the AFLOW AGL. So, you have to prepare such a structure by a separate AFLOW run using a relaxation mode for energy and/or forces.

When AFLOW AGL is run, it will first check for the existence of the required DFT calculations. If these exist, then it will extract the required information and proceed to calculate the thermal properties. If these do not exist, then it will create the required directories and `aflow.in` files, and then stop. By running AFLOW with the `--multi` flag, AFLOW will then automatically descend into these subdirectories and run the required electronic structure calculations, after which it will stop. The user will then have to rename or delete the `LOCK` file in the subdirectory, and then run AFLOW again to calculate the thermal properties. This can all be performed within a subshell script as in the following example:

```
aflow --multi -D $pathname/ >> log.$PBS_JOBID
mv $pathname/LOCK $pathname/LOCK.old
aflow --multi -D $pathname/ >> log.$PBS_JOBID
```

This procedure is similar to that required to run AFLOW APL.

1) To perform a AFLOW AGL calculation the following line has to be present in `aflow.in`:

```
[AFLOW_AGL]CALC
```

AFLOW checks for the existence of this line in `aflow.in`. If it does not exist, then the job is not an AFLOW AGL calculation and no other AGL settings are read.

AGL can also be run from within other parts of AFLOW, such as from APL, in order to obtain properties such as bulk modulus and equilibrium volume as a function of temperature to be used within these other parts of AFLOW. In this case, AGL will run and the other AGL settings in the `aflow.in` file will be read, even if the `"[AFLOW_AGL]CALC"` line is not present.

2) The AFLOW AGL method requires a set of energy vs. volume data, which are obtained from DFT

calculations of a set of strained structures. The user has the possibility to set the number of strained structures and the size of the strain step. These parameters can be set by including the following lines in the `aflow.in` file:

```
[AFLOW_AGL]NSTRUCTURES=number  
[AFLOW_AGL]STRAINSTEP=number
```

The default setting is 28 structures and a step size of 0.01. The strain step size corresponds to the value by which each lattice vector is shortened or lengthened for each new strained structure.

- 3) The AFLOW AGL method allows the user to select what type of AFLOW DFT runs are performed for each strained structure (see Calderon et al., *Comp. Mat. Sci.* 108A, 233-238 (2015) for a description of the parameters used for the different calculation types). The default option is to perform an AFLOW STATIC calculation for each strained structure, which can be set by including the following lines in the `aflow.in` file:

```
[AFLOW_AGL]STATIC=ON
```

The other option is to run a `RELAX_STATIC=2` calculation, which can be set by including the following lines in the `aflow.in` file:

```
[AFLOW_AGL]RELAX_STATIC=ON
```

- 4) The AFLOW AGL method allows the user to select which form of the equation of state is used to fit the (E, V) data. The options include numerical, Vinet (*J. Phys. Condens. Matter* 1, 1941 (1989)), Birch-Murnaghan and spinodal (Baonza, Caceres, and Nunez, *Phys. Rev. B* 51, 28 (1995)). These options can be selected by including the following line in the `aflow.in` file :

```
[AFLOW_AGL]IEOS=number
```

The options are as follows:

IEOS=0: Numerical equation of state (default setting)

IEOS=1: Vinet equation of state

IEOS=2: Birch-Murnaghan equation of state

IEOS=3: Vinet equation fitting, but numerical calculation to obtain thermal properties

IEOS=4: Birch-Murnaghan equation fitting, but numerical calculation to obtain thermal properties

IEOS=5: Spinodal (BCNT) equation of state

IEOS=6: Spinodal (BCNT) equation fitting, but numerical calculation to obtain thermal properties

IEOS=-1: Number equation of state with minimal output, no plots generated or files other than [system name].

For the spinodal equation of state, the user also has the option to optimize the non-linear parameter beta using the following:

```
[AFLOW_AGL]IOPT_G=number
```

The options are as follows:

IOPT_G=0: Beta is fixed to 0.85

IOPT_G=1: Beta is always optimized

IOPT_G=2: Beta is only optimized in the static calculation, and then this value is used for the finite temperature calculations.

- 5) The AFLOW AGL method allows the user to select whether or not to calculate the Debye temperature self-consistently. These options can be selected by including the following line in the `aflow.in` file:

```
[AFLOW_AGL]IDEBYE=number
```

The options are as follows:

IDEBYE=0: No self-consistent calculation of Debye temperature, Debye temperature calculated from static bulk modulus

IDEBYE=1: User contributed Debye temperatures

IDEBYE=2: Self-consistent calculation of Debye temperatures from adiabatic bulk moduli

IDEBYE=3: Debye temperature calculated from static bulk modulus at static equilibrium volume

IDEBYE=-1: Static calculation only, no finite temperature properties calculated

User contributed Debye temperatures should be given in the following format:

```
[AFLOW_AGL]DEBYE_TEMP_number=number
```

The first number is an integer corresponding to the number of the structure, while the second number is the value of the Debye temperature for this structure.

- 6) The AFLOW AGL method allows the user to choose the details of how the (E, V) curve is fitted by a polynomial particularly how convex points in the (E, V) data are accounted for. This allows the user to relax the concave restrictions on the (E, V) data being used. These options can be selected by including the following line in the aflow.in file:

```
[AFLOW_AGL]FITTYPE=number
```

The options are as follows:

FITTYPE=0: Fits only the first concave section of the (E, V) data. If this does not include the minimum, it fits the second concave section starting from the minimum, if such a section exists. This is the default setting.

FITTYPE=1: Fits only (E, V) data points where the energy is monotonically decreasing until the minimum energy is reached and points where the energy is monotonically increasing after the minimum energy is reached.

FITTYPE=2: Fits all (E, V) data points.

FITTYPE=3: Selects points using same method as for FITTYPE=1, but then fits these points by a polynomial which generates a new set of smoothly concave (E, V) data points.

FITTYPE=4: Fits all points by a polynomial which is used to generate a new set of smoothly concave (E, V) data points.

FITTYPE=5: The (E, V) data is read in from user-created file instead of being directly extracted from the original electronic structure calculations.

Note that these fitting options should be used carefully, as they could give misleading results. They are meant for debugging, and for in depth exploration of certain properties of specific systems. For high-throughput data analysis, only the default option of FITTYPE=0, as this is the safest. Also, be sure to check for warnings about data points excluded from the calculation. If a lot of points are excluded, you may want to recheck the (E, V) from the original electronic structure calculations, and if it is very noisy it would be advisable to rerun these calculations with a denser k-point grid.

- 7) The AFLOW AGL method allows the user to set the value of the Poisson ratio to be used by including the following line in the aflow.in file:

```
[AFLOW_AGL]POISSON=number
```

The default value is 0.25.

The AFLOW AGL method can also call the AFLOW AEL method to calculate the Poisson ratio from the elastic constants by including the following line in the aflow.in file:

```
[AFLOW_AGL]AELPOISSONRATIO=ON
```

This option will run the AEL method in full, calculating the Poisson ratio, bulk and shear moduli, and other elastic properties. Note that running this method in conjunction with AGL will require an additional aflow run followed by moving the LOCK file to a different location. See the README file for AEL for more details on this option.

- 8) The AFLOW AGL method allows the user to set the number of temperature and pressure points to be used in the calculation, as well as the size of the increments between each temperature or pressure point. These values can be set by including the following lines in the aflow.in file:

```
[AFLOW_AGL]NPRESSURE=number
```

```
[AFLOW_AGL]SPRESSURE=number
```

```
[AFLOW_AGL]NTEMP=number
```

```
[AFLOW_AGL]STEMP=number
```

NPRESSURE is the number of pressure points and SPRESSURE is the increment between pressure points.

NTEMP is the number of temperature points and STEMP is the increment between temperature points.

The default values are as follows:

```
NPRESSURE=11
SPRESSURE=2.0
NTEMP=201
STEMP=10.0
```

- 9) The AFLOW AGL method allows the user to provide specific pressure values to be used in the calculations. These option can be selected by including the following line in the aflow.in file:

```
[AFLOW_AGL]SETPRESSUREVALUE=ON
```

Setting this option results in the specific pressure values being read in from the aflow.in file. The default value is SETPRESSUREVALUE=OFF so that no specific values are read in. User specified pressure values should be provided using the following format:

```
[AFLOW_AGL]PRESSURE_number=number
```

Note that the first pressure value should always be equal to 0.0. If this is not the case, AFLOW AGL will give a warning and insert a value of 0.0 as the first pressure value. Also, negative pressure values will result in AFLOW AGL exiting with an error.

- 10) The AFLOW AGL method allows the user to set the maximum number of convergence iterations for the Gauss-Legendre and self-consistent Debye functions. This limit prevents these functions from getting stuck in an infinite loop. This value can be set by including the following line in the aflow.in file:

```
[AFLOW_AGL]MAXLOOPS=number
```

The default value is 250.

- 11) The AFLOW AGL method allows the user to set the value of the energy for an infinite volume. The purpose of this is to set the zero of the energy scale. This value can be set by including the following line in the aflow.in file:

```
[AFLOW_AGL]EINF=number
```

The default value is 0.0.

- 12) The AFLOW AGL method allows the user to choose whether or not to write out the input file for the original, non-AFLOW version of GIBBS. This can be useful for debugging and testing. This option can be selected by including the following line in the aflow.in file:

```
[AFLOW_AGL]WRITEGIBBSINPUT=option
```

The AGL input will be written for WRITEGIBBSINPUT=ON. The default value is WRITEGIBBSINPUT=OFF, which does not write out the file.

- 13) The AFLOW AGL method allows the user to choose whether or not to check the concavity and location of the energy minimum of the (E, V) data calculated with VASP before starting the GIBBS method. In order to fit the (E, V) data by a polynomial, the data points should be concave and include a minimum. This option can be selected by including the following line in the aflow.in file:

```
[AFLOWGIBBS]CHECKEVCONCAVITY=option
```

A value of CHECKEVCONCAVITY=ON causes the concavity check to be performed. The default value is CHECKEVCONCAVITY=OFF so no concavity check is performed prior to the GIBBS method. The GIBBS method will perform another concavity check after the (E, V) data has been passed to it, and will return an error if the data is not concave.

Problems with the (E, V) data are usually the result of incompleteness of the electronic structure calculation basis set or k-point grid. If CHECKEVCONCAVITY=ON and if the (E, V) data is not concave, then AFLOW AGL doubles value of KPPRA to increase the number of k-points, and creates a new set of directories containing aflow.in files with this new value of KPPRA. The electronic structure calculations are rerun with the increased number of k-points, and the new set of (E, V) data is once again checked for concavity. This process is repeated until a concave data set is obtained or the maximum number of concavity check iterations is reached. The number of concavity check iterations can be set by including the following line in the aflow.in file:

```
[AFLOW_AGL]MAXCCITER=number
```

The default value for the maximum number of concavity check iterations is 1.

The AFLOW AGL method allows the user to choose whether or not to check that the (E, V) data set includes the energy minimum. In order to get a good fit, the (E, V) data should contain the minimum. This check can be initiated by including the following line in the aflow.in file:

```
[AFLOW_AGL]CHECKMIN=option
```

A value of CHECKMIN=ON causes the check of the location of the energy minimum to be performed. The default value is CHECKMIN=OFF so no check is performed.

If CHECKMIN=ON, the (E, V) data is sorted in order of increasing volume. If the lowest energy in the data set corresponds to the smallest or largest volume, then the (E, V) data set does not include the minimum. Additional strained structures are then created until the (E, V) data includes a minimum or the maximum number of minimum check iterations is reached. The number of minimum check iterations can be set by including the following line in the aflow.in file:

```
[AFLOW_AGL]MAXCMITER=number
```

The default value for the maximum number of minimum position check iterations is 5.

- 14) Since multiple different volume calculations are performed for a given system, there is a certain amount of failure of some of these calculations, and therefore options have been implemented within AGL to allow for one or more failed run. Failed runs can be detected automatically (by the absence of the appropriate LOCK or OUT) and the option of whether or not to skip these runs can be set by including the following line in the aflow.in file:

```
[AFLOW_AGL]AUTOSKIPFAILEDARUNS=option
```

The default is "OFF" so that no runs are skipped automatically; instead detection of a failed run will cause an error unless this particular run was explicitly listed to be skipped.

Specific runs can be listed for skipping by including the following line in the aflow.in file:

```
[AFLOW_AGL]SKIPFAILEDARUNS=directory_name_list
```

where "directory_name_list" is a string with the list of directory names containing the failed runs to be skipped by the fitting algorithm. If there is more than one failed run, the individual directory names should be listed on separate lines separated by commas.

AGL only permits a certain number of runs to be skipped. The maximum number of runs permitted to be skipped can be set by including the following line in the aflow.in file:

```
[AFLOW_AGL]SKIPARUNSMAX=number
```

The default value is 7, so that up to seven runs can be skipped.

- 15) The AFLOW AGL method will automatically calculate the thermal conductivity at the acoustic Debye temperature as a function of temperature using the equation originally derived by Slack (see Glen A. Slack, Solid State Physics 34, 1-75 (1979) for original method and C. Toher et al., Phys. Rev. B 90, 174107 (2014) for more details; these references also described the acoustic Debye temperature). This calculation is always performed.

The thermal conductivity can also be calculated with this equation as a function of temperature using the equilibrium volume as a function of temperature by including the following line in the aflow.in file:

```
[AFLOW_AGL]KAPPAVOLUME=option
```

The thermal conductivity will be calculated using the equilibrium volume as a function of temperature for KAPPAVOLUME=ON, and the results are written to the file "AGL_thermal_properties_temperature_vol.dat". The default value is KAPPAVOLUME=OFF, which only calculates the thermal conductivity using the equilibrium volume at the acoustic Debye temperature.

The user can also provide values to AGL for the acoustic Debye temperature and the Gruneisen parameter to be used when calculating the thermal conductivity by including the lines:

```
[AFLOW_AGL]THETA_COND=number
```

[AFLOW_AGL]GRUNEISEN=number

If both of these lines are present, the thermal conductivity is calculated with the Slack equation using the values provided by the user, and the results are written in the "AGL_thermal_conductivity_user_lib.dat". This can be useful for comparing AGL results to values in the literature for the purpose of debugging and testing the AGL code.

- 16) The main output file for AFLOW AGL is aflow.agl.out. It contains the AGL results for the thermal conductivity, the traditional and acoustic Debye temperatures, the Gruneisen parameter, the heat capacities at constant volume and constant pressure (Cp) at 300K, the thermal expansion at 300K, and the static and isothermal bulk modulus as well as recording the source and value of the Poisson ratio used to calculate the Debye temperature. It also contains the results for the thermal conductivity, Debye temperature, Gruneisen parameter, heat capacity, thermal expansion and bulk modulus as function of temperature; and the Gibbs free energy, vibrational Helmholtz free energy, vibrational internal energy, and vibrational entropy as a function of temperature.
- 17) The AFLOW AGL method allows the user to choose whether or not to automatically create plots of the results for the Debye temperature, heat capacity, Gruneisen parameter, and vibrational free energy. These plots are saved as different formats depending on the input given by the user. The plots are stored in the files debye_temperature.*, heat_capacity.*, gruneisen_parameter.* and vibrational_free_energy.* where "*" is dependent on the format in which the files are saved. These results can be plotted by including the following line in the aflow.in file:

[AFLOW_AGL]PLOTRESULTS=option

If PLOTRESULTS=ON, then the plots are generated with gnuplot. The default option is PLOTRESULTS=OFF so that no plots are generated. Which files are kept and which formats are used to save the plots depends on the option used to run AFLOW. In particular, the "--keep" option controls whether or not the gnuplot script ("--keep=gnuplot") and the eps file ("--keep=eps") are saved instead of deleted. The plots can be saved in other file formats such as png, pdf, jpeg and gif using the options "--keep=png", "--keep=pdf", "--keep=jpg" and "--keep=gif", respectively.

If [AFLOW_AGL]IEOS=-1 or [AFLOW_AGL]IDEBYE=-1, then the plotting of these properties is skipped.

The AFLOW AGL method also allows the user to choose whether to write additional output files. These additional output files include the file Debye_temperature.dat, which writes the Debye temperature, heat capacity, vibrational free energy and Gruneisen parameter as a function of temperature in a format which is easy to plot using commonly available graphing software such as xmgrace. This file can be written by including the following entry in the aflow.in file:

[AFLOW_AGL]WRITEFULLRESULTS=option

The additional output files are written if WRITEFULLRESULTS=ON. The default value is WRITEFULLRESULTS=OFF, so that no additional output files are written. When this option is active, the values of the Debye temperature, heat capacity, vibrational free energy and Gruneisen parameter at certain temperatures are written in the file Debye_summary.dat. These temperature values correspond to methods most often used in the literature to determine which temperature is used for the Debye temperature or heat capacity being reported. If [AFLOW_AGL]IEOS=-1 or [AFLOW_AGL]IDEBYE=-1, then the writing of these files is skipped.

The specific temperature values are determined as follows:

- Best fit of Debye temperature to heat capacity:

The first temperature value is determined from the value of the Debye temperature which produces the best fit to the heat capacity curve calculated using AGL. The heat capacity curve is fitted using the following equation:

$$c_V = 9 n k_B \left(\frac{T}{\Theta_D} \right)^3 \int_0^{\Theta_D / T} \frac{x^4 e^{-x}}{(e^x - 1)^2} dx$$

Here, c_V is the heat capacity at constant volume, n is the number of atoms in the unit cell, k_B is Boltzmann constant, T is the temperature, and Θ_D is the Debye temperature. This equation is the same as that given in equation 23.26 in Solid State Physics, Ashcroft and Mermin (1976). The temperature value at which this value of c_V is produced by AGL is also written, along with the heat capacity, vibrational free energy and Gruneisen parameter at this temperature.

- Temperature where heat capacity is half Dulong-Petit value

At high temperatures, the heat capacity tends to a limit given by $3 n k_B$, where n is the number of atoms per unit cell and k_B is the Boltzmann constant. This is known as the Dulong-Petit value. The second temperature value is determined from the point at which the heat capacity is equal to half of this value, i.e. $1.5 n k_B$. This method is used to determine the Debye temperatures listed in table 23.3 in Solid State Physics, Ashcroft and Mermin (1976).

- Minimum value of Debye temperature
The third temperature value used it that at which the Debye temperature calculated using the AGL method is :

- Maximum value of Debye temperature
The fourth temperature value used it that at which the Debye temperature calculated using the AGL method is :

- 300K (~room temperature)
The final temperature used is 300K, which corresponds approximately to room temperature.

Finally, the values of the Debye temperature, Gruneisen parameter and vibrational free energy for the temperature corresponding to the best fit to the heat capacity, the half Dulong-Petit value of the heat capacity, and 300K (~room temperature), respectively, are written in a script-parsable format in the file agllib.dat. The format is the same as that used for the file aflowlib_entry.out.

18) EXAMPLE: CaO FCC with default AGL settings (note that this will give the same results if all of the "[AFLOW_AGL]CALC" lines except "[AFLOW_AGL]CALC" are removed or commented out)::

```
[AFLOW] *****
[AFLOW] Test file for AGL calculation of CaO (fcc)
[AFLOW] *****
[AFLOW] SYSTEM=Ca101_ICSD_26959
[AFLOW] *****
[AFLOW] input file for aflow
[AFLOW_MODE=VASP]
[AFLOW] *****
[AFLOW_MODE_ZIP=bzip2]
[AFLOW_MODE_BINARY=vasp46s]
[AFLOW] *****
[AFLOW] *****
[AFLOW_MODE_MPI]
[AFLOW_MODE_MPI_MODE]NCPUS=MAX
[AFLOW_MODE_MPI_MODE]COMMAND ="mpirun -np"
[AFLOW_MODE_MPI_MODE]AUTOTUNE
[AFLOW_MODE_MPI_MODE]BINARY="mpivasp46s"
[AFLOW] *****
[AFLOW_SYMMETRY]CALC
#[AFLOW_SYMMETRY]SGROUP_WRITE
#[AFLOW_SYMMETRY]SGROUP_RADIUS=7.77
[AFLOW] *****
#[AFLOW_NEIGHBOURS]CALC
[AFLOW_NEIGHBOURS]RADIUS=7.7
[AFLOW_NEIGHBOURS]DRADIUS=0.1
[AFLOW] *****
#[AFLOW_APL]CALC // README_AFLOW_APL.TXT
[AFLOW_APL]ENGINE=DM // README_AFLOW_APL.TXT
[AFLOW_APL]DISMAG=0.015 // README_AFLOW_APL.TXT
[AFLOW_APL]MINATOMS=100 // README_AFLOW_APL.TXT
#[AFLOW_APL]SUPERCELL=3x3x3 // README_AFLOW_APL.TXT
[AFLOW_APL]DC=y // README_AFLOW_APL.TXT
[AFLOW_APL]DPM=y // README_AFLOW_APL.TXT
[AFLOW_APL]ZEROSTATE=y // README_AFLOW_APL.TXT
[AFLOW_APL]DOS=y // README_AFLOW_APL.TXT
[AFLOW_APL]TP=y // README_AFLOW_APL.TXT
[AFLOW_APL]TPT=0:2000:10 // README_AFLOW_APL.TXT
[AFLOW] *****
#[AFLOW_QHA]CALC // README_AFLOW_APL.TXT
[AFLOW_QHA]GP_VOL_DISTORTION_PERCENTAGE=0.03 // README_AFLOW_APL.TXT
[AFLOW_QHA]DISPLACEMENTS=y // README_AFLOW_APL.TXT
[AFLOW_QHA]PROJECTION_DIR=1:1:1 // README_AFLOW_APL.TXT
[AFLOW_QHA]EOS=n // README_AFLOW_APL.TXT
[AFLOW_QHA]EOS_VOLRANGE_DIST=-2:4:0.5 // README_AFLOW_APL.TXT
```

```

[AFLOW_QHA]EOS_KPOINTS_MODE=32768:10000:20:100000 // README_AFLOW_APL.TXT
[AFLOW] *****
#[AFLOW_AAPL]CALC // README_AFLOW_APL.TXT
[AFLOW_AAPL]TDMAG=0.015 // README_AFLOW_APL.TXT
[AFLOW_AAPL]CUT_SHELL=4 // README_AFLOW_APL.TXT
[AFLOW_AAPL]CUT_RAD=4.5 // README_AFLOW_APL.TXT
[AFLOW_AAPL]SUMRULE=1E-5 // README_AFLOW_APL.TXT
[AFLOW_AAPL]BTE=FULL // README_AFLOW_APL.TXT
[AFLOW_AAPL]THERMALGRID=21x21x21 // README_AFLOW_APL.TXT
[AFLOW_AAPL]ISOTOPE=y // README_AFLOW_APL.TXT
[AFLOW_AAPL]CUMULATIVEK=y // README_AFLOW_APL.TXT
[AFLOW_AAPL]BOUNDARY=n // README_AFLOW_APL.TXT
[AFLOW_AAPL]NANO_SIZE=100 // README_AFLOW_APL.TXT
[AFLOW_AAPL]TCT=200:700:20 // README_AFLOW_APL.TXT
[AFLOW] *****
#[VASP_RUN]RELAX_STATIC_BANDS=2 // GENERATE | STATIC | RELAX=N | RELAX_STATIC=N | STATIC
#[VASP_FORCE_OPTION]NEGLECT_NOMIX
#[VASP_FORCE_OPTION]CHGCAR=OFF // ON | OFF (default ON)
#[VASP_FORCE_OPTION]KPOINTS=keyword[,keyword] // EVEN | ODD | KSHIFT_GAMMA_EVEN | KSHIFT_GAMMA_ODD | KS
#[VASP_FORCE_OPTION]SYM=ON // ON | OFF (default ON)
#[VASP_FORCE_OPTION]AUTO_PSEUDOPOTENTIALS=potpaw_PBE // pot_LDA | pot_GGA | potpaw_LDA | potpaw_GGA | potpaw_P
#[VASP_FORCE_OPTION]NBANDS // Estimate Bands (better than VASP)
#[VASP_FORCE_OPTION]SPIN=ON,REMOVE_RELAX_1 // (ON | OFF (default ON)), REMOVE_RELAX_1 | _2
#[VASP_FORCE_OPTION]AUTO_MAGMOM=ON // ON | OFF (default OFF)
#[VASP_FORCE_OPTION]RELAX_MODE=ENERGY // ENERGY | FORCES | ENERGY_FORCES | FORCES_ENERGY (defau
#[VASP_FORCE_OPTION]PREC=HIGH // (LOW | MEDIUM | NORMAL | HIGH | ACCURATE), PRESERVED (
#[VASP_FORCE_OPTION]ALGO=FAST // (NORMAL | VERYFAST | FAST | ALL | DAMPED), PRESERVED (
#[VASP_FORCE_OPTION]NOTUNE
#[VASP_FORCE_OPTION]TYPE=DEFAULT // METAL | INSULATOR | SEMICONDUCTOR | DEFAULT (default D
#[VASP_FORCE_OPTION]CONVERT_UNIT_CELL=SPRIM // SPRIM, SCONV, NIGGLI, MINK, INCELL, COMPACT, WS, CART,
#[VASP_FORCE_OPTION]VOLUME+=10.0
#[VASP_FORCE_OPTION]VOLUME*=1.05
[AFLOW] *****
[AFLOW] *****
[VASP_INCAR_MODE_EXPLICIT]START
SYSTEM=Ca101_ICSD_26959
NELM = 120
NELMIN=2
LPLANE=.TRUE.
LREAL=.FALSE.
LSCALU=.FALSE.
#NBANDS=XX # for hand modification
#IALGO=48 # for hand modification
[VASP_INCAR_MODE_EXPLICIT]STOP
[AFLOW] *****
[VASP_KPOINTS_MODE_IMPLICIT]
[VASP_KPOINTS_FILE]KSCHEME=M
[VASP_KPOINTS_FILE]KPPRA=8000
[VASP_KPOINTS_FILE]STATIC_KSCHEME=M
[VASP_KPOINTS_FILE]STATIC_KPPRA=10000
[VASP_KPOINTS_FILE]BANDS_LATTICE=FCC
[VASP_KPOINTS_FILE]BANDS_GRID=20
[AFLOW] *****
[AFLOW_AGL]CALC
[AFLOW_AGL]AELPOISSONRATIO=OFF
[AFLOW_AGL]POISSON=0.25
[AFLOW_AGL]NTEMP=201
[AFLOW_AGL]NPRESSURE=11
[AFLOW_AGL]NSTRUCTURES=28
[AFLOW_AGL]STRAINSTEP=0.01
[AFLOW_AGL]WRITEGIBBSINPUT=OFF
[AFLOW_AGL]WRITEFULLRESULTS=OFF
[AFLOW_AGL]PLOTRESULTS=OFF
[AFLOW_AGL]IEOS=0
[AFLOW_AGL]IDEBYE=0

```

```

[AFLOW_AGL]CHECKEVCNCAVITY=OFF
[AFLOW_AGL]CHECKMIN=OFF
[AFLOW_AGL]FITTYPE=0
[AFLOW] *****
[VASP_POSCAR_MODE_EXPLICIT]START
Ca101 #225 - (Ca101_ICSD_26959) - Ca101 [Ca101] cF8 Fm-3m Ca0 225 Ca101_ICSD_26959 ICSD_26959 (icsd library) (
1.224745
  0.00000000000000  1.95061033183634  1.95061033183634
  1.95061033183634  0.00000000000000  1.95061033183634
  1.95061033183634  1.95061033183634  0.00000000000000
1 1
Direct(2) [A1B1]
  0.00000000000000  0.00000000000000  0.00000000000000  Ca
  0.50000000000000  0.50000000000000  0.50000000000000  0
[VASP_POSCAR_MODE_EXPLICIT]STOP
[AFLOW] *****
[VASP_POTCAR_MODE_IMPLICIT]
[VASP_POTCAR_FILE]Ca
[VASP_POTCAR_FILE]0
[AFLOW] potpaw_PBE: Ca_sv 0
[AFLOW] COMPOSITION_PP=|Ca_sv1|01|
[AFLOW] COMPOSITION=|Ca1|01|
[AFLOW] VOLUME(A^3)=|0|0|
[AFLOW] MASS(amu)=|40.08|15.9994|
[AFLOW] *****
[AFLOW] Aflow automatically generated (aflow_avasp.cpp)
[AFLOW] *****
[AFLOW] aflow/aconvasp/ae/apennsy v(30793) in ICSD/FCC/Ca101_ICSD_26959
[AFLOW] *****
[AFLOW] SELF-MODIFICATION
[AFLOW] Recycling CONTCAR of relax1
[AFLOW] *****
[VASP_POSCAR_MODE_EXPLICIT]START
Ca101 #225 - (Ca101_ICSD_26959) - Ca101
1.224745
  0.00000000000000  1.97560216103176  1.97560216103176
  1.97560216103176  0.00000000000000  1.97560216103176
  1.97560216103176  1.97560216103176  -0.00000000000000
1 1
Direct(2) [A1B1]
  0.00000000000000  -0.00000000000000  -0.00000000000000
  0.50000000000000  0.50000000000000  0.50000000000000
[VASP_POSCAR_MODE_EXPLICIT]STOP
[AFLOW] *****
[VASP_FORCE_OPTION]SPIN=OFF // Self Correction
[AFLOW] SELF-MODIFICATION
[AFLOW] Recycling CONTCAR of relax2
[AFLOW] *****
[VASP_POSCAR_MODE_EXPLICIT]START
Ca101 #225 - (Ca101_ICSD_26959) - Ca101
1.224745
  0.00000000000000  1.97560216103176  1.97560216103176
  1.97560216103176  0.00000000000000  1.97560216103176
  1.97560216103176  1.97560216103176  0.00000000000000
1 1
Direct(2) [A1B1]
  0.00000000000000  0.00000000000000  0.00000000000000
  0.50000000000000  0.50000000000000  0.50000000000000
[VASP_POSCAR_MODE_EXPLICIT]STOP
[AFLOW] *****

```