Manual - MEUMAPPS-SS

Y. Song and B. Radhakrishnan

1. Introduction

This manual is for the MEUMAPPS-SS code that can simulate microstructure evolution during solid-state phase transformations in multi-component alloys including multiple crystallographic variants and multiple phases. Dr. Radhakrishnan and his colleagues have originally developed the code for an Ti-6Al-4V alloy [1] and later developed it for including simultaneous evolution of multiple phases. Dr. Song reorganized the code into its current modular form and also generalized the code to handle arbitrary number of phases and elements. Other theoretical background and details can be found in Refs. [1, 2, 3, 4, 5, 6, 7]. This manual provides a description of the code lay-out, required input files, and steps involved in compiling and running the code in a parallel environment.

2. Source files

The source files can be categorized into three groups: (1) input files, (2) compiling/execution, and (3) code files. The first group is for input parameters to perform a simulation. The second group is to compile *Makefile* and to execute the compiled program *summit.lsf*. The last group consists of Fortran codes of the main program *Multi_MEUMAPPS.f* and other subroutines/functions.

2.1 Input files

Two input files are *Input.txt* to set simulation parameters and *sym_var.inp* to calculate symmetrical crystallographic variants of a phase.

2.1.1 Input.txt

This file contains parameters to run a MEUMAPPS-SS simulation. The description below is an example file for three elements and three phases and should be suitably modified for arbitrary number of phases and elements.

- **line 1** A buffer line.
- **line 2** related to decomposition of the computational domain. The product of *dims(1)* and *dims(2)* is the number of processors used. If the first number is *ndim = 1*, the following two numbers become *dims(1) = 1* and *dims(2) = nprocs* (*nprocs* is the number of processors). If *ndim = 2*, the following two numbers specify the decomposition of the processor grid for the P3DFFT library. The product of the last two numbers should be the

same as the total number of processors. The simulations are carried out in a cubic domain *Nx*, *Ny* and *Nz*. These quantities must be divisible by *dims(1)* and *dims(2)*.

- **line 3** A buffer line.
- **line 4** related to the simulation domain. First three numbers are for domain sizes *Nx*, *Ny*, and *Nz* along the *x*, *y*, and *z* directions, respectively. The last number is for the grid spacing in meters.
- **line 5** related to time parameters. The first number *nrun* is a flag that denotes whether the run is new (*nrun*=1) or re-run (*nrun* > 1). The second number *step* is an initial time step. The third number *elastep0* shows the number of time steps before starting the elastic energy calculation. If a time step is less than this number *step* < *elastep0*, the elasticity calculation is ignored. This is useful if the elastic displacements fail to converge in the presence of a sharp discontinuity in the order parameter and concentration due to initial assignment of a nucleus. The fourth number *Nt* is the maximum iteration number. The fifth number *t_step* is a time step in seconds. The last number is *noise_step.* It denotes the number of time-steps for which the noise will be active, after which the noise is set to zero.
- **line 6** related to mobility parameters. The first value *b_mob* is for a boundary mobility. The second value *diff_coeff* is for a diffusion coefficient of an alloy element. In the current version, this value is same for all elements. The last value *m_star* is not being used in the current version.
- **line 7** related to noise parameters. The two numbers are linked to the Langevin noise term
- **line 8** related to output frequency. The first *ifreq* is the number frequency to write output files during a simulation. The code will generate those files every *ifreq* steps.
- **line 9** A buffer line.
- **line 10** related to nucleation status. The first word is for nucleation status *nucstat*. One of four status values *'single'*, *'multiple'*, *'random'* or '*none*' is used to initialize the order parameter in the nucleation step. This is applied only when *nrun* in line 5 is equal to 1. Otherwise, it reads output data from the previous run. The second number *targv* is a target variant number for a nucleation. If *targv = 0*, a new variant has its variant number of 1 for *nucstat = 'single'* or a randomly selected number for *nucstat = 'multiple'*. The third number *vartot* is the total number of variants. The fourth number *nmaxppt* is a total number of nuclei in the *'multiple'* nucleation case. The last number *iseed* is a seed number to generate random numbers that are used for selecting the location of the nuclei. This should be a negative integer.
- **Line 11** has two integers *var_lo* and *var_hi*. Th etwo numbers set the lowest and highest variant numbers to be used in the simulation. *var_lo* =0 indicates variant number 1. *Var_hi* cannot be greater than *var_tot* in line 10.
- **line 12** related to seed radii. The three numbers are radii along the *x*, *y*, and *z* axes of a new precipitate.
- **line 13** related to additional seed radii. The three numbers are additional radii along the *x*, *y*, and *z* axes of a new precipitate. Those are used when the seed size is not the same, and the maximum seed radius becomes *radppt* + *addrad*. This version does not use those numbers.
- **Line 14** *disc_ener* is the initial energy of dislocations in the matrix in J/m^3 – AM materials are known to have an initial dislocation density. The elastic energy due to dislocations is added to the Gibbs energy of the matrix

- **line 15** A buffer line.
- **line 16** related to phase and elements numbers. The first and the second numbers are for the number of phases and elements, respectively.
- **line 17** related to an interfacial energy between precipitates.
- **line 18** related to elastic energy. Three *e_mean* values are used to introduce external strains along the x, y and z axes of the domain.
- **line 19** A buffer line.
- **line 20** related to initialization of concentration status *constat*. If the word *constat = 'cppt'*, the concentrations inside a new precipitate are different from the ones in the matrix. Otherwise, the concentrations are the same everywhere inside the domain.
- **lines 21 and 22** describe concentration information. The example is for a simulation involving three alloying elements requiring two element concentrations to be specified. The first number *nelc* is for an element index. The second value *Cav* is an average concentration value. The third value *Cimax* is a value inside a precipitate. The last value *Cigrad* is a concentration slope from the precipitate to the matrix. The current version does not use this value. This input should be appropriately be extended for alloys with more that 3 elements.
- **line 23** A buffer line.
- **line 24** related to initial value of the order parameter in the domain. The value *phi_init* is the order parameter value that is used to initialize the field. It is typically set as $\varphi = 0$ for the matrix.
- **line 25** related to phase parameters. The first number *nphc* is the current phase number (*nphc = 1* for the matrix). The second number *Nvar* is the number of variants for the current phase. The last number *Ctype* is the crystal type.
- **line 26** related to the elasticity matrix for the matrix phase. The three values are for the (1,1), (1,2), and (4,4) values in the elasticity matrix. In this example, the matrix has cubic crystal structure. All elastic constants are in units of Pa (N/m^2)
- **line 27** related to Gibbs-free energy. Values in this line are parameters for a quadratic function of the Gibbs-free energy of the matrix.
- **line 28** A buffer line.
- **line 29** related to phase parameters. The first number *nphc* is the current phase number. The second number *Nvar* is the number of variants for the current phase. The last number *Ctype* is the crystal type.
- **line 30-31** related to the elasticity matrix. The values in line 27 are the (1,1), (1,2), (1,3), (2,2), (2,3), (3,3) in the elasticity matrix. The values in line 28 are the (4,4), (4,5), (4,6), (5,5), (5,6), (6,6) in the elasticity matrix.
- **line 32** related to the transformation strain for the current phase. The three values correspond to (100), (010), and (001) directions.
- **line 33** related to the interfacial energy. The three values correspond to (100), (010), and (001) directions in the unit of J/m^2.
- **line 34** related to Gibbs-free energy. Values in this line are parameters for a quadratic function of the Gibbs-free energy of the phase.
- **Repeat the phase section** For each precipitate, lines 28-34 are repeated.

2.1.2 sym_var.inp

This file is used to calculate crystallographically equivalent orientation variants of a precipitate. The current *sym_var.inp* is for a δ precipitate within a γ matrix.

2.2 Compiling and execution

Two files *Makefile* and *summit.lsf* are in this group.

2.2.1 Makefile

Before compiling the code group, it is necessary to set a proper software environment. For the MEUMAPPS codes, LINPACK, MPI, FFTW and P3DFFT libraries are used. The system environment is required to be set properly. On the summit or summitdev, it can be easily achieved by loading the following command module before compiling (make):

```
> module load fftw
> module load pgi/19.9
> make
```

With the *make* command, the codes are compiled based on the Makefile file. Please note that since summit and summitdev environments are slightly different, the compilation is required to designate proper P3DFFT, fftw and LINPACK library locations. An example is provided for the Summit and Summitdev machines at ORNL for user "user" and for a user project number "proj_number

---

For the summit:
P3DFFT = -L /autofs/nccs-svm1 proj/proj_number/p3dfft summit/lib -I /autofs/nccs-

svm1 proj/proj_number/p3dfft summit/include -lp3dfft -L $(OLCF FFTW ROOT)/lib -l fftw3 -mp

LINPLIB = /ccs/home/user/linpack/linpack_summit/liblinpack.a

For the summitdev:
P3DFFT = -L /autofs/nccs-svm1 proj/proj_number/p3dfft summitdev/lib -I /autofs/nccs-

svm1 proj/proj_number/p3dfft summitdev/include -lp3dfft -L $(OLCF FFTW ROOT)/lib -l fftw3 -mp

LINPLIB = /ccs/home/user/linpack/linpack_summit dev/liblinpack.a

---

For other installations, make sure that the P3DFFT and LINPLIB are compatible with the local system requirement.

2.2.2 summit.lsf

This file is a batch script to execute a simulation on the summit and summitdev systems. The batch script contains specifying the scripts interpreter (first line), scheduler options (with *#BSUB*), and execution section (typically with shell commands). In the execution section, the compiled codes will be performed with the jsrun launcher. Detailed description of the batch script for Summit and Summitdev can be found at olcf.ornl.gov. Appropriate procedures for running in other computer systems should be followed.

2.3 Code group

Most files are in this code group. The main program is *Multi_MEUMAPPS.f*. It calls module files (*ModGparams.f*, *ModInitials.f*, *ModFunc.f*, and *ModElastic.f*) including subroutines and functions, and a function *ran_2.f* file during calculation. This section will review those files briefly.

2.3.1 Multi_MEUMAPPS.f

This is the main program. It allocates and deallocates array memories during a simulation. Some parameters are defined in *ModGparams.f*. It initializes the simulation setting with *ModInitials.f*. It performs main calculations with *ModFunc.f* and *ModElastic.f*. It writes simulation fields every *ifreq* steps.

2.3.2 ModGparams.f

Most global parameters are defined in this module. Most of them are determined using input parameters and do not change during a simulation.

2.3.3 ModInitials.f

This module contains subroutines: *eltensor*, *InitElasticMat*, *InitTransGrad*, *k_space*, *InitFields*, and *NucAdd*.

- **eltensor** This subroutine assembles an elasticity matrix using the input crytal symmetry.
- **InitElasticMat** This subroutine completes elasticity by calling *eltensor* and calculates its inverse matrix.
- **InitTransGrad** This subroutine calculates transformation strain and gradient coefficients for γ', and δ variants in Ni-base alloys 625, 718. For the δ phase, these are calculated in a general fashion using the rotation matrices that transform the transformation strain from the habit plane to the computational plane, and the symmetry matrices to generate the transformation strain matrix for all the crystallographically equivalent variants in the computational frame. A similar approach is used for the gradient coefficient matrix. For the γ' phase the transformation the values are simply assigned based on the simple alignment of the three variants along the primary axes of the computational frame. A more general approach for different variants in a consistent manner will be implemented in future versions of the code.
- **k_space** This subroutine calculates the k vectors in Fourier space and their square values.
- **InitFields** This subroutine initializes the phase fields and concentration fields.

- **NucAdd** This subroutine introduces precipitates when a simulation starts from a previous result.

2.3.4 ModFunc.f

This module consists of four subroutines *f_trans*, *inv_trans*, *invert*, and *langevin*.

- **F_trans** This subroutine performs the forward FFT, i.e., *p3dfft_ftran_r2c* function in the P3DFFT library, and the results are divided by the simulation domain size ($Nx \times Ny \times Nz$).
- **Inv_trans** This subroutine carries out the inverse FFT, i.e., *p3dfft_btran_c2r* function in the P3DFFT library.
- **invert** This subroutine inverts an input matrix using functions in the LINPACK library (http://www.netlib.org/linpack/).
- **langevin** This subroutine calculates the Langevin noise terms. The current version is not supported.

2.3.5 ModElastic.f

This module has two subroutines of *eldis* and *elener*. They calculate elastic energy values based on Refs. [4, 5, 6, 7].

2.3.6 ran_2.f

This function generates a random number (output) based on an input seed number.

2.4 Functions

We briefly review other functions in the MPI and P3DFFT libraries.

2.4.1 MPI

This library is imposed by including *'mpi.h'*.

- **MPI_Init** This function initializes the MPI execution environment. It is closed by calling MPI Finalize.
- **MPI_Comm_rank** This function determines the rank of the calling process in the communicator.
- **MPI_Comm_size** This function determines the size of the group associated with a communicator.
- **MPI_Finalize** This function terminates MPI execution environment. The MPI environment was initialized with either *MPI_Init* or *MPI_Init_thread*.
- **MPI_Bcast** This function broadcasts a message (i.e., a value) from the root processor to all other processors in the MPI group.

- **MPI_Reduce** This function combines the elements provided in the input buffer of each process in the MPI group.
- **MPI_Allreduce** This function combines values from all processes and distributes the result back to all processes.
- **MPI_Barrier** This function completes after all MPI processors have entered here. Hence, all processors are aligned at this point.

2.4.2 P3DFFT

This library is for the 3D FFT (url: https://www.p3dfft.net).

- **p3dfft_setup** It sets the 2D array decomposition up. By calling p3dfft clean, the P3DFFT environment is closed.
- **p3dfft_get_dims** It uses as *p3dfft_get_dims(S, E, SZ, IP)*. *S* is for an array containing 3 integers, defining the beginning indices of the local array for the given task within the global grid. *E* is for an array containing 3 integers, defining the ending indices of the local array within the global grid. *SZ* is for an array containing 3 integers, defining the local array's dimensions. *S*, *E*, and *SZ* are the output values. *IP* is an input value. *IP=1* is for an original physical space array of real numbers. local in *x*, distributed among *P1* tasks in y dimension and *P2* tasks in z dimension, where *P1* and *P2* are processor grid dimensions defined in the call to *p3dfft_setup*. *IP=2* is a transposed wavenumber space array of complex numbers, local in *z*, distributed among *P1* tasks in *x* dimension, *P2* tasks in *y* dimension.

3. Outputs

This section reviews output files and error messages.

3.1 Output files

If the simulation is successfully completed, it writes *fort** files for simulation parameters, log files, and field information files.

3.1.1 fort files

The code writes simulation parameters in four files starting with *fort*.

- **fort.93** It shows the elasticity matrices of the matrix phase and the first precipitate phase (see the *InitElasticMat* function in *ModInitials.f*).
- **fort.95** It shows the transformation strain and gradient coefficients for all variants (see the main program *Multi_MEUMAPPS.f*).
- **fort.98** It shows energy values and minimum/maximum concentrations at each step. Each column is for (1) *step*, (2) total interfacial energy *fint_tot*, (3) total energy-well potential *fw_tot*, (4) total chemical energy *fch_tot*, (5) elastic energy *elen_tot*, (6) total energy *e_tot*

of the system, (7) minimum concentration *glob_min*, and (8) maximum concentration *glob_max*. (see the main program and the *eldis* function in *ModElastic.f*).

- **fort.99** It shows average concentrations at each step. The first column is for a step, and the rest columns are the average concentrations of the elements.

3.1.2 Log files

Because the log file is designated in the batch script, two log files (*jobname.i6* and *run.log*) are generated.

- **jobname.i6** The file name is jobname that is defined in the batch script and i6 for six identification numbers when the job submitted. It shows system related outputs including the initial and final times of the simulation.
- **run.log** It writes calculated simulation parameters (see the main program). If the file name is not designated when executing the program, the results will be shown in the other log file *jobname.i6*.

3.1.3 Field information

The code generates a folder of '*step_i7*' using the current step. The current step number consists of seven numbers *i7*. One file *data_extract_multi.in* shows the simulation and processor information. Other files of data *save.i4*, where *i4* is for four numbers, saves field information.

- **Data_extract_multi.in** This file has eight numbers in two lines. In the first line, the first three numbers correspond to the simulation domain size of *Nx*, *Ny*, and *Nz*. The fourth number is for the number of processors *nprocs*. The last two numbers are for the total number of variants *vartot* and elements *numel*. In the second line, two numbers are for the FFT decomposition using the P3DFFT library.
- **Data_save.i4** Each processor generates a binary file for the field data. The *i4* of the tail in a file name corresponds to the processor id. The lists of those files are *ist*, *ien*, *Cons*, *phi*, *el_en*, *u1*, *u2*, *u3*, *grad_1_u1*, *grad_1_u2*, *grad_1_u3*, *grad_2_u1*, *grad_2_u2*, *grad_2_u3*, *grad_3_u1*, *grad_3_u2*, and *grad_3_u3* in order. The numbers *ist* and *ien* are for the dimension information. The arrays of *Cons*, *phi*, and *el_en* correspond to concentration fields, phase fields, and elastic energy fields, respectively. Other arrays are used elastic energy calculation.

3.2 Error flags

There are two types of error messages. The first type is to check initial input parameters. The second type is shown during the calculations.

3.2.1 Type I

The Type I error is shown before the main calculation is initiated. It checks initial parameters, and, if an error is notified (initerr $\neq 0$), the main calculation is not initiated.

- **initerr = 1** shows when the total number of variants is not matched to the summation of the variant number.
- **initerr = 2** shows when an *inp* file for crystallographically equivalent orientation variants is not detected.
- **initerr = 3** shows when files about input field data have not been detected.
- **initerr = 4** shows when the total number of phases is not matched to the last phase number.
- **initerr = 5** shows when the total number of alloy elements is not matched to the summation of elements.

3.2.2 Type II

This Type II error occurs during the calculation. The error message comes with an error flag number.

- **err_flag = 2** shows when the determinant of the Z matrix is less than 10^{-16} in the main program *Multi_MEUMAPPS.f*.
- **err_flag = 3** shows when the determinant of a matrix is less than 10^{-16} in the *eldis* subroutine (0^{th} order) in *ModElastic.f*.
- **err_flag = 4** shows when the determinant of a matrix is less than 10^{-16} in the *eldis* subroutine (a higher order) in *ModElastic.f*.
- **err_flag = 6** shows when the determinant of a matrix is less than 10^{-16} in the *elener* subroutine in *ModElastic.f*.
- **err_flag = 20** shows when the determinant of an elasticity matrix is less than 10^{-16} in the *InitElasticMat* subroutine in *ModInitials.f*.

References

[1] B. Radhakrishnan, S. Gorti, and S. S. Babu, Metall. Mater. Trans. A (2016).

[2] N. Moelans, B. Blanpain and P. Wollants, CALPHAD (2008).

[3] S. G. Kim, W. T. Kim, and T. Suzuki, Phys. Rev. E (1999).

[4] I. Steinbach and M. Apel, Physica D. (2006).

[5] S. Y. Hu and L. Q. Chen, Acta Mater. (2001).

[6] A. Durga, N. Moelans, and P. Wollants, In Proceedings of the ISRS–International Symposium for Research Scholars in Metallurgy, Materials Science and Engineering (2010).

[7] A. Durga, P. Wollants, and N. Moelans, Modelling Simul. Mater. Sci. Eng. (2013).