2023 Annual Conference & Exposition
Baltimore Convention Center, MD | June 25 - 28, 2023
The Harbor of Engineering Education for 130 Years
ASEE
Paper ID #36688

Development of an Online Phase-Field Theory Course for Mechanical Engineering Graduate Students

Dr. Yucheng Liu, South Dakota State University

Dr. Yucheng Liu (PhD, PE, FASME, FSAE) currently serves as the Department Head of Mechanical Engineering (ME) Department at South Dakota State University (SDSU) and holds the Sander Professor of Entrepreneurial Engineering in the Jerome J. Lohr College of Engineering at SDSU. Prior to joining SDSU in the fall of 2021, Dr. Liu was an Associate/Full Professor in the ME Department at Mississippi State University (MSU) since the fall of 2014. In that department, he served as Graduate Coordinator from 2016 to 2021 and held the Jack Hatcher Chair in Engineering Entrepreneurship in the Bagley College of Engineering from 2018 to 2021. Before joining MSU, Dr. Liu was an Assistant Professor of the ME Department at the University of Louisiana at Lafayette (UL Lafayette). Dr. Liu received his PhD degree from the University of Louisville in 2005 and bachelor's degree from the Hefei University of Technology in 1997, both in Mechanical Engineering. Dr. Liu's research has historically focused on the areas of multiscale material modeling and simulation, high strain rate performance of materials, vehicle systems design and analysis, and hydropower and wave energy technology. His current research interests and activities center on gaining a better understanding of the process-structure-property-performance relations of structural materials through advanced multiscale theoretical framework and integrated computational and experimental methods. To date, Dr. Liu has published nearly 250 peer reviewed publications, including more than 130 peer reviewed journal articles, and received 2 patents. He has been the PI and co-PI for over 40 research projects funded by NSF, DOD, DOE, NASA, FAA, Louisiana Board of Regents, and industry with a total amount over $15.5M. Dr. Liu has served on review panels for many NSF, DOD, NASA, and DOE programs. Dr. Liu received the Junior Faculty Researcher of the Year of the College of Engineering at UL Lafayette in 2013, the Outstanding Senior Faculty Research Award from the ME Department in 2016, MSU and BCOE Faculty Research Award in 2018; he was named to the Jack Hatcher Chair in Engineering Entrepreneurship in 2018 and promoted to Full Professor in 2019. In his professional societies, Dr. Liu was elected a Fellow of ASME in 2017, a Fellow of SAE in 2019, and received the SAE Forest R. McFarland Award in 2020. Dr. Liu is a Professional Engineer registered in Ohio State and also holds active membership in ASEE and AAAS.

DEVELOPMENT OF AN ONLINE PHASE-FIELD THEORY COURSE FOR MECHANICAL ENGINEERING GRADUATE STUDENTS

Abstract – A graduate level course, Phase-Field Theory, was designed and offered to the mechanical engineering (ME) M.S. and Ph.D. students at Mississippi State University (MSU) through online instruction. The significance of the phase-field method to students in materials science and the course development process are elaborated in this paper. In particular, since this class was taught in summer semester 2020, measures were taken to enable students to understand the essence of the phase-field method and master fundamental phase-field modeling techniques in a short period time despite the challenges of COVID-19 during the course development process. Positive student feedback and decent evaluation results were proofs that the course objectives were achieved.

Keywords: Phase-field method, mechanical engineering, graduate education, online teaching, computational approach

1. Introduction

Phase-field method has emerged as a powerful and flexible tool for simulating the formation and evolution of microstructures in a wide variety of physical processes in materials. A distinct advantage of this method is that it removes the need of explicitly tracking complex of interfaces and instead allows the prediction of the morphology evolution at micro- to mesoscale by following the growth kinetics of key microstructural features [1]. Because of such advantages, by now the phase-field method has been extensively applied to model different types of phase transformations in various materials, such as grain boundary migration and grain growth, precipitation, recrystallization, solidification of metallic materials, domain evolution in ferro-electric and ferro-magnetic materials, irradiation damage, and so on [2-8]. Thus, there is a compelling need to teach this method to graduate students who wish to study materials science as they get started in their research.

Several universities including Penn State, Northwestern University, McGill University, and Indian Institute of Technology Mumbai have taught phase-field method to their students and received very good responses. Recognizing a growing students' need for this method, the author developed a "Phase-Field Theory" course in the spring of 2020 and first taught it to the mechanical engineering (ME) graduate students at Mississippi State University (MSU) in Summer 2020. A course development approach [9-10] put forward by Liu was followed to establish a synopsis of this course and other course materials. Because of COVID-19, this course was intentionally designed as an online course and would be converted to a face-to-face course after the pandemic is over.

2. Course Description of "Phase-Field Theory"

Phase-Field Theory is a three-credit course about computational and theoretical methods in phase-field modeling and development of phase-field codes. The course is to be taken by graduate students in materials science who wish to delve into fundamental physics underlying various materials. The goal of this course is to provide students with the knowledge and skills necessary to develop efficient theoretical models to predict and manipulate the evolution of microstructure

and equip them with the tools to reveal the underlying mechanisms that define the functionality of materials and assist in the design of new materials. It is expected that after completing this course, students will be able to (1) gain the knowledge of the mathematics related to the phase-field method; (2) analyze phase transformations in metals when the metals are heated or during cooling from an elevated temperature; (3) recommend best practices for implementing the phase-field method given a specific scenario; and (4) Construct a phase-field model for a multi-phase physical system and simulate its microstructure evolution during a physical process.

As mentioned before, this course is designed as an online, summer class. This course is an 8-week series so it can be fit into a summer term and two 2-hour sessions per week (on Tuesdays and Thursdays) are scheduled for lectures, discussions, and phase-field code development. The difficulty of this course is such that a minimum weekly commitment of 10-12 outside study hours will be required.

3. Course Development

3.1 Challenges

Two challenges were faced by the author when developing this course. The first challenge was associated with teaching a summer class: how to make an easy-to-follow tutorial on phase-field modeling that would allow students to come up to speed with the details of this topic in a short period of time. The second challenge was caused by the COVID-19 pandemic, which was how to foster teacher-student interaction and keep the students engaged and on task in an online environment. Both challenges had to be addressed in order to achieve the course learning outcomes.

3.2 Textbooks and Topics

To overcome the first challenge, an easy-to-read textbook that transmits the essential content of phase-field method in a standard and economical way must be chosen for this class. After comparing several books on phase-field theory, the book *Phase-Field Methods in Materials Science and Engineering* written by Provatas and Elder [11] was selected as the textbook for this course. This book comprehensively covers basic principles of phase-field theory and explains numerical techniques for solving phase-field models in a fundamental way, which is most appropriate for learners who have the least experience in the phase-field method. Meanwhile, Biner's book *Programming Phase-Field Modeling* [12] was recommended as a reference book because this book contains a collection of codes that students can use as template to practice and develop their own phase-field codes. A combined usage of these two books provides a fast-track pathway from phase-field theory to numerical implementation of phase-field modeling.

Only fundamental topics that comprise the basic principles and fundamental knowledge of phase-field theory are covered in this 8-week course. A detailed course outline is listed in Table 1. In the phase-field modeling lectures (the two classes in week 7 and the first class in week 8), I demonstrated one example each class to show students how to create a phase-field model and run specific phase-field simulations using Abaqus/CAE. I also developed step-by-step guide to help the students independently complete the modeling and simulation assignments.

Table 1. Course schedule of "Phase-Field Theory"

Week	Tuesday Lecture	Thursday Lecture
1 (6/9)	Introduction and overview of phase-field model	Phase separating and ordering systems; spinodal decomposition
2 (6/16)	Free energy; Gauss' theorem; group theory; variational calculus	Classical diffusion equation; Cahn-Hilliard equation
3 (6/23)	Solution of Cahn-Hilliard equation; interfacial energy	Phase transformation; Allen-Cahn equation
4 (6/30)	Coherent misfitting precipitates; Eshelby problem	Asymptotic analysis; grain growth phase-field model
5 (7/7)	Finite difference and spectral implementation of Allen-Cahn equation	Incorporating elastic stress effects into phase-field models
6 (7/14)	General framework of thermomechanics and energy dissipation for PF models	Energy-dissipative time-integration schemes
7 (7/21)	Phase-field modeling of grain growth	Phase-field modeling of dendritic solidification
8 (7/28)	Phase-field modeling of brittle fracture	Review and discussion

3.3 Communications in An Online Classroom

Several measures were adopted to effectively engage online students and promote a positive online learning environment. A course website was created through MSU's blackboard system for the instructor to post course materials and facilitate virtual chats and discussion boards between the instructor and students, and among the students. Online help and discussion sessions (twice a week) were introduced through the course website for the instructor to meet with the students online, answering their questions, clarifying confusing topics, explaining the instructor's expectations, etc. Such online sessions are an equivalent of traditional office hours and were later proved to be impactful on the academic success of the students who took this class in Summer 2020. In teaching this class, the author also stimulated informal and random communications with the students via emails and phone calls.

In addition, the students were asked to submit their notes after each class and the instructor would review and grade the class notes based on their completeness, neatness, structure and organization, and handwriting (for hand-written notes only). The notes of each lecture are a permanent record helping a student to identify the core of important ideas in the lecture, and to review and understand the lecture either. The notes were made worth 40% of the total grade to incentivize students to stay focused during each lecture.

3.4 Project Assignment

One important goal of this course is to help students to develop their skills to develop numerical codes for their own phase-field models so they can directly apply the programming skills to their research. To achieve that goal, the students needed to complete three code analysis reports and develop two phase-field codes throughout their course. The code analysis reports were required to ensure that the students understood the provided numerical codes for existing phase-field models. Those codes had mainly come from the instructor's previous research [2-8]. After that, the students

were assigned two projects and they would need to use the existing codes as templates to develop their own phase-field codes and run their codes on computer to generate results and plots. Midterm and final exams were not used because the goal of this course is not to sharpen students' problem-solving skills but to develop their programming and implementation skills. The projects were deliberately designed and assigned to the students so that there were no two students had the same two projects.

By adopting the project assignments, a project-based learning (PBL) approach was established in this class. As demonstrated by Foss and Liu and verified from their previous teaching practice [13-18], PBL is an effective approach to improve students' attitudes toward learning and make them more engagement and self-reliant. Therefore, the project assignments are indispensable for online courses. The examples of the phase-field codes provided by the instructor were written in Matlab and the students were encouraged (but not required) to use the Matlab programming package to develop their own codes. This is because had been implemented in several undergraduate and graduate courses in the ME curriculum at MSU [19, 20] and most of the ME students were adroit at using this programming language. It is worth mentioning that all the students enrolled in this course had previous experience in programming and computational modeling so they should have the capacity to complete the projects.

Due to the constraints associated with COVID-19, those projects were individual projects, but the students were encouraged to contact the instructor and/or join group discussions with their classmates whenever they needed help. After the pandemic is over, this course will be converted to a face-to-face course and group projects will be designed for the course following an approach presented by Liu [21, 22].

In this class, both the code analysis reports and phase-field codes were worth 30% of the final grade, and the lecture notes were worth 40%.

4. Student Work Samples

Examples of the results of phase-field simulations conducted by the students in their projects are displayed below. Fig. 1 illustrates a 3D phase-field simulation of γ" precipitation in a Ni-based superalloy during heat treatment. Fig. 2 displays grain morphologies of additively manufactured Ti-6Al-4V alloy. The evolution of the grain morphologies was simulated using a temperature-dependent grain growth phase-field model provided by the instructor. Fig. 3 exhibits simulated evolution of microstructure during solidification of IN718 alloy, which was obtained using a dendrite growth phase-field model. Main codes for the analysis illustrated in Fig. 1 are provided in Appendix as an example.

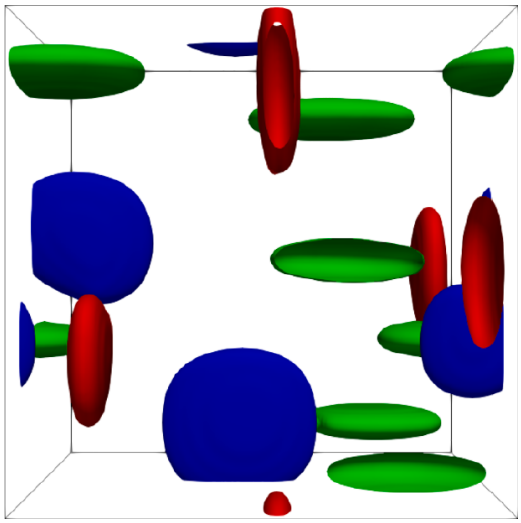

Figure 1. Simulated microstructure of γ” precipitates in Inconel 625 during heat treatment

Figure 2. Simulated grain morphologies of Ti-6Al-4V alloy

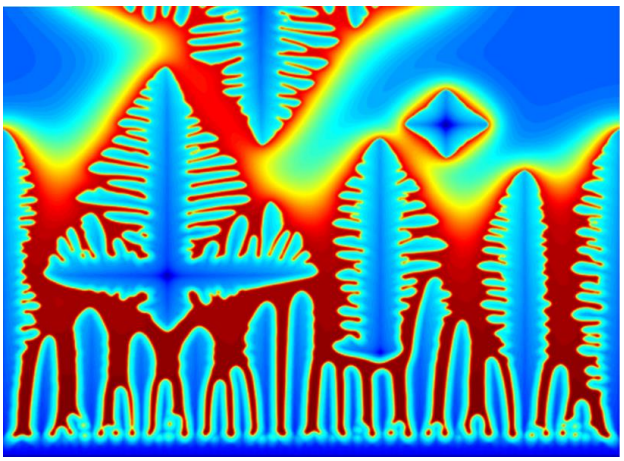

Figure 3. Simulated dendrite nucleation and growth of IN718 alloy during solidification

It is worth mentioning that these figures just represent a few examples of student work and all the students enrolled in this course had done very well. Sixteen graduate students (three PhD students and 13 Master's students) took this class, and the class average grade was 92.6%.

5. Student Feedback

A total of 16 students enrolled in this class in Summer 2020 and the final course evaluation score was 4.6 out of 5.0, a strong proof that the course goals were achieved. Student feedback further confirmed that through this class, the students gained a fundamental understanding of basic principles about phase-field method and developed appropriate phase-field modeling and simulation skills. The students would then apply those skills and knowledge in their research. Selected comments from the students include:

"*Amount of knowledge gained was tremendous. This class will be very helpful in my research!*"

"*The class was well organized as always. Dr. Liu presents a caring, considerate and open attitude toward us.*"

"*This is one of the toughest classes, but Dr. Liu is extremely effective at communicating difficult concepts through Webex!*"

"*I like this course and instructor so much. This course really arose my interest in mechanical engineering.*"

"*We are very interested in phase-field modeling. I think it's our best interest to truly learn what phase-field method is and how to develop and run a phase-field model.*"

Moreover, six of the 13 Master's students chose to continue pursuing a PhD degree either in mechanical engineering or materials science at MSU or other institutions after receiving their Master's degree. Among the three PhD students, one student already graduated in 2021 and is now a tenure-track faculty member at a national university in the United States. The phase-field theory and modeling method taught in this course are being intensively used by these students in their research.

6. Conclusion

An online course about phase-field method and modeling was designed and offered to the ME graduate students at MSU in Summer 2020. Topics, references, and assessment tools of this course were deliberately determined to ensure that the most wanted computational and theoretical methods in phase-field modeling would be covered and the student learning outcomes would be closely assessed. Positive student feedback and other course outcomes are strong testament to the achievement of the course goals and student engagement in an online environment. The student work samples have shown that by taking this class, the students were able to use the existing phase-field codes as templates to develop their own codes and produce all calculations. One student in that class received her PhD and is now a tenure-track faculty member at a U.S. national university and eight students in that class are still pursuing their PhD degree. These students are heavily using the phase-field theory and modeling approach taught in this course as a powerful tool in their research. In the future, this course can be converted to a face-to-face class and permanently implemented into ME curriculum as a core graduate course taught during regular semesters. Students from any engineering or science department who need to predict microstructure

evolutions or understand process-structure-property relationships of materials will benefit from this class. The present course development approach can also be adjusted for the development of other online numerical modeling and analysis courses.

References

1. L.-Q. Chen, "Phase-field models for microstructure evolution", *Annual Review of Materials Research*, 32(1), 2002, 113-140.
2. L. Chen, C.O. Yenusah, Y.-Z. Ji, Y.-C. Liu, T.W. Stone, M.F. Horstemeyer, and L.-Q. Chen, "Three-dimensional phase-field simulation of γ" precipitation kinetics in Inconel 625 during heat treatment", *Computational Materials Science*, 187, 2021, 110123.
3. P.-W. Liu, Z. Wang, Y.-H. Xiao, R.A. Lebensohn, Y.-C. Liu, M.F. Horstemeyer, X.-Y. Cui, and L. Chen, "Integration of phase-field model and crystal plasticity for the prediction of process-structure-property relation of additively manufactured metallic materials", *International Journal of Plasticity*, 128, 2020, 102670.
4. X. Wang, P.-W. Liu, Y.-Z. Ji, Y.-C. Liu, M.F. Horstemeyer, and L. Chen, "Investigation on microsegregation of IN718 alloy during additive manufacturing via integrated phase-field and finite element modeling", *Journal of Materials Engineering and Performance*, 28(2), 2019, 657-665.
5. C.O. Yenusah, T.W. Stone, N.R. Morgan, R.W. Robey, Y.-C. Liu, and L. Chen, "Incorporating performance probability and data-oriented design in phase-field modeling", IDETC2022-89513, ASME 2022 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference (IDETC/CIE 2022), August 14-17, 2022, St. Louis, MO, USA.
6. C.O. Yenusah, Y.-Z. Ji, Y.-C. Liu, T.W. Stone, M.F. Horstemeyer, and L. Chen, "Investigation of precipitation kinetics and hardening effects of γ" in Inconel 625 using a combination of meso-scale phase-field simulations and macro-scale precipitates strengthening calculations", IMECE2020-23328, ASME 2020 International Mechanical Engineering Congress & Exposition, virtual conference, November 16-19, 2020.
7. C. Jouhari, Y.-C. Liu, and D. Dickel, "Phase-field modeling of aluminum foam based on molecular dynamics simulations", TMS 2023 152nd Annual Meetings & Exhibition Supplemental Proceedings, 2023, 632-641.
8. C. Jouhari, Y.-C. Liu, and D. Dickel, "Molecular dynamics study of gradient energy coefficient and grain-boundary migration in aluminum foam", TMS 2023 152nd Annual Meetings & Exhibition Supplemental Proceedings, 2023, 623-631.
9. Y.-C. Liu, "Teaching continuum mechanics in mechanical engineering program", *Journal of STEM Education: Innovations and Research*, 12(1/2), 2011, 17-23.
10. Y.-C. Liu, "A programming course including C# and MATLAB for mechanical engineering students", *ASEE Computers in Education Journal*, 2(3), 2011, 106-112.
11. N. Provotas and K. Elder, *Phase-Field Methods in Materials Science and Engineering*, John Wiley & Sons, 2011.
12. S.B. Biner, *Programming Phase-Field Modeling*, Springer, 2017.
13. M. Foss and Y.-C. Liu, "Creating solutions through project-based and experimental learning: A case study of the Concept Center", *International Journal of Engineering Education*, 37(6), 2021, 1630-1642.

14. M. Foss, Y.-C. Liu, and S. Yarahmadian, “Project-based learning in a virtual setting: A case study on materials and manufacturing process and applied statistics’, *International Journal of Engineering Education*, 39(5A), 2022, 1377-1388.
15. M. Foss and Y.-C. Liu, “Promoting sustainable development goals through project-based learning: A case study of the concept center”, 1st International Academy Conference on the Sustainable Development Goals, Utah Valley University, Orem, UT, USA, October 5-7, 2022.
16. M. Foss and Y.-C. Liu, “Lessons learned from 5 years of parent daughter engineering outreach: Using project-based learning to introduce families to engineering disciplines”, 2022 ASEE Annual Conference and Exposition, Minneapolis, MN, USA, June 26-29, 2022.
17. M. Foss and Y.-C. Liu, “Developing creativity through project-based learning”, Wasatch Aerospace and Systems Engineering Mini-Conference, virtual conference, April 15-16, 2021.
18. M. Foss and Y.-C. Liu, “Project-based learning center to bridge to students with technology”, 2020 Intermountain Engineering Technology and Computing (IETC), Orem, UT, USA, September 2020.
19. Y.-C. Liu, “A programming course including C# and MATLAB for mechanical engineering students”, *ASEE Computers in Education Journal*, 2(3), 2011, 106-112.
20. Y.-C. Liu, “Implementation of MATLAB/Simulink into a vibration and control course for mechanical engineering students”, ASEE SE Annual Conference, Auburn University, Auburn, AL, USA, March 8-10, 2020.
21. Y.-C. Liu, “Renovation of a mechanical engineering senior design class to an industry-tied and team-oriented course”, *European Journal of Engineering Education*, 42(6), 2017, 800-811.
22. Y.-C. Liu and Y.-Q. Dou, “Design of an industry-tied and team-oriented course for mechanical engineering seniors”, ASEE SE Annual Conference, University of Florida, Gainesville, FL, USA, April 12-14, 2015.

Appendix

Main codes for the analysis shown in Fig. 1

```
!start timing. first call to system_clock
!call system_clock(count1, count_rate)
start_time = dclock()


!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
! Definition of reference frames        !
! (x'y'z'): variant reference frames    !
! (xyz): grain reference frame in 3D    !
! (XYZ): global reference frame         !
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!


!k-values
CALL Receiprocal(kpow2,kpow4,kx,ky,kz,nx,ny,nz,dx,dy,dz)


!Define the dimension of the system
IF (nx==1 .or. ny==1 .or. nz==1) THEN
  dimen=2
ELSE
  dimen=3
ENDIF

if (initflag == 1) then
  write (*,*) "   "
  write (*,*) "Getting the results from the previous calculation!"
  write (*,*) "   "
endif

write (*,*) "   "
do i=1,nc
  write (*,*) "Overall composition (X",i,"):",c0(i)
enddo
write (*,*) "   "
!unit conversions
!K/min to K/sec
heating_rate = heating_rate/60.0


!Initialize needed data
call Initialilzation

!initialization for time marching
dt = dt1
sim_time=0.0
!-----------------------------------time marching
timemarch: DO iter = 1,numsteps
           global_iter = iter + initcount  !global_iter is for debuging codes
           niter = iter + initcount

           if (niter <= stepstochange) then
             pstep = stepst1
           elseif (niter > stepstochange) then
             pstep = stepst2
           endif

           !Time step for integration 2
           if (niter > stepstochange) then
             dt = dt2
           endif

           !Calculate Temperature dependent parameters
           CALL Temp_Dependent

           !Dim.less simulation time
           sim_time = sim_time+dt
           !Real simulation time [secs]
           real_sim_time = real_sim_time + real_dt

           !Get heating_start_time
           !Uncomment if non-isothermal
           !if (niter .eq. heating_start_iter) then
           !**************************************************
           IF ( (restart_file_written == .TRUE.) .OR. (iter == numsteps) ) THEN
               deallocate(kpow2,kpow4)
               deallocate(eta,comp)
               deallocate(dfdc,dfdeta,muelast)
               deallocate(equi_compa_temp,equi_compb_temp,equi_compc_temp)

               !***********
               STOP
               !***********
           ENDIF
           !**************************************************
ENDDO timemarch

CONTAINS
!---------------------------Initialization
SUBROUTINE Initialilzation

    if(initflag == 0)then
      initcount = 0
      real_sim_time = 0.0
      num_of_nuclei = 0
      num_dens = 0.0
    elseif(initflag == 1)then
      !read previous_simulation info from file = 'previous_sim_info.txt'
      call previous_simulation
    endif

    !Get temperature dependent parameters
    CALL Temp_Dependent

    !initialization of eta and comp
    CALL iniconf(initflag,icase)

    !checking if restart_file has been written
    restart_file_written = .FALSE.




      !seed drand
      call srand(iseed)

      !initialize  global_iter
      global_iter = 0

  END SUBROUTINE Initialilzation


END SUBROUTINE Evolve


!Subroutine of ksquare
SUBROUTINE Receiprocal(kpow2,kpow4,kx,ky,kz,nx,ny,nz,dx,dy,dz)
    implicit none

    integer, intent(in) :: nx,ny,nz
    real, intent(in) :: dx,dy,dz
    real, intent(out) :: kx(nx/2+1),ky(ny),kz(nz)
    real, intent(out) :: kpow2(nx/2+1,ny,nz),kpow4(nx/2+1,ny,nz)

    integer i,j,k,ti,tj,tk
    real, parameter :: twopi=6.28318531
    real fksq

    do k=1,nz
      tk=k-1
      if (tk.GT.nz/2) tk=tk-nz
    do j=1,ny
      tj=j-1
      if (tj.GT.ny/2) tj=tj-ny
    do i=1,nx/2+1
      ti=i-1
      fksq=float(ti**2+tj**2+tk**2)
      if (fksq.GT.1E-12) then
        kx(i)=float(ti)*twopi/(nx*dx)
        ky(j)=float(tj)*twopi/(ny*dy)
        kz(k)=float(tk)*twopi/(nz*dz)
        kpow2(i,j,k)=kx(i)**2+ky(j)**2+kz(k)**2
```

```
        !Get heating_start_time
        !Uncomment if non-isothermal
        !if (niter .eq. heating_start_iter) then
        !  heating_start_time = real_sim_time
        !endif

        !Classical Nucleation Theory
        if ( (icase .eq. 8) ) then
            CALL Nucleation

        endif


num_dens_if: if ( (num_dens .gt. 0.0) .or. (icase .ne. 8) ) then
        !Normalize all parameters
        CALL Normalization

        !Setting of elastic properties, SFTS and Gradient coefficients
        CALL Set_up(dimen,eigenstrain_0_v,Chom,Chom_v,SS_v)

        !---------------this section beginning here is the bottleneck of the code
        !timing bottleneck of code
        !call cpu_time(tic)
        tic = secnds(0.0)

        !Elasticity solver -
        !elasticity solver for homogeneous/inhomogeneous case - Voigt notation
        CALL Elastic_Solver(kx,ky,kz,eigenstrain_0_v,Chom,Chom_v,SS_v,muelast)

        !Set Up of local-free-energy functions
        CALL local_free_energy(dfdeta,dfdc)
        !print*,dfdeta
        !stop

        !1. Cahn-Hilliard Equation - Semi-implicit Fourier-spectral method
        CALL Solving_CH(dfdc,kx,ky,kz,kpow2)

        !Solving Gorvening Equations:
        !2. Allen-Cahn Equation - Semi-implicit Fourier-spectral method for
```

```
        !Solving Gorvening Equations:
        !2. Allen-Cahn Equation - Semi-implicit Fourier-spectral method for
        !single crystal and Explicit Euler method with k-space Laplacian calculation for polycrystals
        CALL Solving_AC(kx,ky,kz,kpow2,dfdeta,muelast)

        !timing bottleneck of code
        !call cpu_time(tic)
        toc = secnds(0.0)
        if (global_iter .le. 20) write(5050,*)'evolve.f90 =', toc-tic, 'seconds'
        if (global_iter .eq. 20) close(5050)
        !----------------------------end of bottleneck of the code
endif num_dens_if

        !Heat Treatment Scheme (temperature as a function of time)
        !Uncomment if non-isothermal
        !if ( (niter .ge. heating_start_iter) .and. (TT .lt. hold_temp) ) then
        !  CALL Temp_History
        !endif

        !second call to system_clock
        !call system_clock(count2, count_rate)
        stop_time = dclock()

        !total_elapsed_time = float((count2 - count1))/count_rate !in seconds
        total_elapsed_time = stop_time-start_time !in seconds

        !------------------------------OUTPUT
        if ( (mod(niter,pstep)==0) .OR. (iter == numsteps) ) then
          CALL write_vtk(niter, total_elapsed_time)
        endif


        !Writing the final results of all variants for restarting
        IF (((total_elapsed_time>(3600.0*float(walltime)-60.0*float(timecutoff))) &
           .AND. (mod(niter,repstep)==0)) .OR. (iter == numsteps)) THEN

            CALL write_restart(niter, total_elapsed_time, restart_file_written)
        ENDIF
```

```
        fksq=float(ti**2+tj**2+tk**2)
        if (fksq.GT.1E-12) then
          kx(i)=float(ti)*twopi/(nx*dx)
          ky(j)=float(tj)*twopi/(ny*dy)
          kz(k)=float(tk)*twopi/(nz*dz)
          kpow2(i,j,k)=kx(i)**2+ky(j)**2+kz(k)**2
          kpow4(i,j,k)=kpow2(i,j,k)**2
        else
          kx(i)=0.0
          ky(j)=0.0
          kz(k)=0.0
          kpow2(i,j,k)=0.0
          kpow4(i,j,k)=0.0
        endif
      enddo
      enddo
      enddo

END SUBROUTINE Receiprocal
```

```
!------------------------Interpolation function: H_func(h)
  FUNCTION H_func(h)
      implicit none
      real, intent(in) :: h
      real :: H_func

      !(1.)
      H_func = 2.0*h**2 - h**4

      !(2.)
      !H_func = -2.0*h**3 + 3.0*h**2

      !(3.)
      !H_func = abs(h)**3*(10.0 -15.0*abs(h) +6.0*h**2)
```

```
    !(2.)
    !H_func = -2.0*h**3 + 3.0*h**2

    !(3.)
    !H_func = abs(h)**3*(10.0 -15.0*abs(h) +6.0*h**2)
    RETURN
END FUNCTION
```

```
------------------------First derivative of H_function: dH_func(h)
FUNCTION dH_func(h)
    implicit none
    real, intent(in) :: h
    real :: dH_func

    !(1.)
    dH_func = 4.0*h - 4.0*h**3

    !(2.)
    !if (h .NE. 0.0) then
      !dH_func = -(6.0*h**3)/abs(h) + 6.0*h
    !else
      !dH_func = -(6.0*h**3)/almost_zero + 6.0*h
    !endif

    !(3.)
    !if (h .NE. 0.0) then
      !dH_func = (30.0*h**5 - 60.0*h**3*abs(h) + 30.0*h**3)/abs(h)
    !else
      !dH_func = (30.0*h**5 - 60.0*h**3*abs(h) + 30.0*h**3)/almost_zero
    !endif
    RETURN
END FUNCTION
```